Andrzej Kwiecień
Piotr Gaj
Piotr Stera (Eds.)

Communications in Computer and Information Science 79

Computer Networks

17th Conference, CN 2010
Ustroń, Poland, June 2010
Proceedings

Springer

Communications
in Computer and Information Science 79

Andrzej Kwiecień Piotr Gaj
Piotr Stera (Eds.)

Computer Networks

17th Conference, CN 2010
Ustroń, Poland, June 15-19, 2010
Proceedings

Springer

Volume Editors

Andrzej Kwiecień
Piotr Gaj
Piotr Stera

Silesian University of Technology
Institute of Informatics
Gliwice, Poland

E-mail: {andrzej.kwiecien, piotr.gaj, piotr.stera}@polsl.pl

Library of Congress Control Number: Applied for

CR Subject Classification (1998): C.2, H.4, D.2, H.3.4-5, D.4.6, K.6.5

ISSN 1865-0929
ISBN-10 3-642-13860-8 Springer Berlin Heidelberg New York
ISBN-13 978-3-642-13860-7 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 06/3180 5 4 3 2 1 0

Preface

The continuous and intensive development of computer science results in the fast progress of computer networks. Computer networks, as well as the entire computer science field, are subject to regular changes caused by the general development of technology, and also the influence of new computer science technology. This progress refers to the methods as well as the tools of designing and modeling computer networks. Particularly, the range of using computer networks permanently is extended thanks to the results of new research and new applications, which were not even taken into consideration in the past. These new applications stimulate the development of scientific research, because the wider use of system solutions based on computer networks results in both theoretical and practical problems. This book is the evidence of the above considerations, with particular chapters referring to the broad spectrum of issues and problems.

This book is the result of the research of scientists from many remarkable scientific research centers. It was created as a collection of articles presented during the 17th edition of the International Conference 'Computer Networks', which took place in Ustroń (Poland) during June 15–19, 2010. This conference, organized continuously since 1994 by the Institute of Informatics of Silesian University of Technology, is the oldest event of this kind organized in Poland, having an international status for three years. This year's edition like last year, took place under the auspices of IEEE Poland Section. As usual, the papers, after insightful reviews, were thematically divided into following groups:

- Molecular networks
- New technologies related to nano and quantum technology
- New technologies related to the computer networks
- Fundamentals of computer networks architecture and programming
- Internet networks
- Data security in distributed systems
- Industrial computer networks
- Applications of computer networks

In each of the groups there are very interesting studies that may interest a wide readership.

In conclusion, on behalf of the Program Committee, we would like to express our acknowledgments to all the authors for sharing their research results and for their assistance in creating this monograph, which in our opinion, represents a valuable reference on computer networks.

April 2010

Andrzej Kwiecień
Piotr Gaj

Organization

CN 2010 is organized by the Institute of Informatics, Silesian Univeristy of Technology (SUT) with support by the Committee of Informatics of the Polish Academy of Sciences, Section of Computer Network and Distributed Systems in technical cooperation with IEEE Poland Section.

Institute of Informatics
Silesian University of Technology
ul. Akademicka 16
44-100 Gliwice, Poland
e-mail: siecikomputerowe@polsl.pl
web: http://cn.polsl.pl

Executive Committee

All members of the Executing Committee are from the Silesian University of Technology, Poland.

Honorary Member:	Halina Węgrzyn
Organizing Chair:	Piotr Gaj
Technical Volume Editor:	Piotr Stera
Technical Support:	Aleksander Cisek
Technical Support:	Arkadiusz Jestratjew
Technical Support:	Jacek Stój
Office:	Małgorzata Gładysz
WEB Support:	Piotr Kuźniacki
IEEE PS Coordinator:	Jacek Izydorczyk

Program Committee

Program Chair

Andrzej Kwiecień	Silesian University of Technology, Poland

Honorary Members

Klaus Bender	TU München, Germany
Zdzisław Duda	Silesian University of Technology, Poland
Andrzej Karbownik	Silesian University of Technology, Poland
Jerzy Rutkowski	Silesian University of Technology, Poland
Stefan Węgrzyn	IITiS PAN Gliwice, Poland
Bogdan M. Wilamowski	Auburn University, USA

Program Committee Members

Tülin Atmaca	Institut National de Télécommunication, France
Win Aung	National Science Foundation, USA
Leszek Borzemski	Wrocław University of Technology, Poland
Markus Bregulla	University of Applied Sciences Ingolstadt, Germany
Tadeusz Czachórski	Silesian University of Technology, Poland
Andrzej Duda	INP Grenoble, France
Jean-Michel Fourneau	Université de Versailles, France
Natalia Gaviria	Universidad de Antioquia, Colombia
Jerzy Klamka	IITiS PAN Gliwice, Poland
Demetres D. Kouvatsos	University of Bradford, UK
Stanisław Kozielski	Silesian University of Technology, Poland
Henryk Krawczyk	Gdańsk University of Technology, Poland
Kevin McNeil	Engineering Fellow, BAE Systems, USA
Nihal Pekergin	Université de Versailles, France
Piotr Pikiewicz	Academy of Business, Poland
Bolesław Pochopień	Silesian University of Technology, Poland
Frank Schiller	TU München, Germany
Mirosław Skrzewski	Silesian University of Technology, Poland
Kerry-Lynn Thomson	Nelson Mandela Metropolitan University, South Africa
Sylwester Warecki	Freescale Semiconductor, USA
Tadeusz Wieczorek	Silesian University of Technology, Poland
Bane Vasic	University of Arizona, USA
Grzegorz Zaręba	University of Arizona, USA

Referees

Tülin Atmaca	Andrzej Kwiecień
Leszek Borzemski	Kevin McNeil
Markus Bregulla	Piotr Pikiewicz
Tadeusz Czachórski	Bolesław Pochopień
Jean-Michel Fourneau	Frank Schiller
Natalia Gaviria	Mirosław Skrzewski
Jerzy Klamka	Kerry-Lynn Thomson
Demetres D. Kouvatsos	Sylwester Warecki
Stanisław Kozielski	Tadeusz Wieczorek
Henryk Krawczyk	Grzegorz Zaręba

Sponsoring Institutions

Technical cosponsor of conference: IEEE Poland Section.

Table of Contents

Structures of Molecules and of Molecular Systems of Informatics

Stefan Węgrzyn

Polish Academy of Sciences, Gliwice

Abstract. The terms contents and structure need to be discerned. For example, the molecule of water consists of one atom of oxygen and two atoms of hydrogen. But, the structure of the water molecule requires to give mutual distribution, within the three dimensional space, of the three atoms that build that molecule. In such a case, the most frequently used form is the so called structural model of a molecule. In this paper, structural models of methane and ethane are discussed in more detail. In this work, also same problems of nanotechnology and genetic engineering are discussed, which require the knowledge about the models of structures of molecules as well.

Keywords: structure, molecular genetic engineering, chirality.

1 Introduction

With the term molecular systems of informatics, we denote such systems of informatics, that were either made or bad built themselves, thanks to mutual interconnections of molecules. Such molecular systems of informatics are biological systems of informatics, which shaped themselves for milliards of years and thanks to which living organisms exist, develop and replicate.

The present work is dedicated to fundamentals of technical molecular systems of informatics, which could be shaped using laboratory methods.

2 Atomic Structure of Matter

Supposition, that the structure of matter is granular, began to form already in ancient times. It was discusses by Democritus, who lived circa 400 years BC. Indivisible grains of matter were names atoms by him, from a Greek word *atmos* – indivisible. The science returned to these ideas in the 18th century, when a lot of observed facts in chemical reactions and in properties of gases could not be explained without the hypothesis about atomic structure of matter. The second half of the 19th century brought the conviction, that atoms exist and have a complex structure. They consist of particles, which have positive and negative electrical charges. Then it has also been stated, that electrons have a mass, being 2000 times smaller that the mass of a lightest ion, that is, of the atom H devoid of its electron.

A. Kwiecień, P. Gaj, and P. Stera (Eds.): CN 2010, CCIS 79, pp. 1–14, 2010.

In 1904 – Thomson presented the first model of an atom. According to the model, an atom is a ball of the radius of a nanometer (10^{-9} mm). Inside the ball exist a nucleus and randomly moving electrons.

In 1909 – Rutherford reported, that most of the atom's mass is concentrated in a sphere of 10^{-12} m. It has already been a nuclear model of an atom, inspired by the structure of Solar system, but without the phenomenon of radiation, despite of the movement of electrical charges.

In 1913 – Bohr devised a model of an atom, based on the principles of quantum mechanics, but he has not explained why the principles of classical mechanics can not be applied to such microparticles as atoms, but one has to rely on the assumptions of quantum mechanics.

1. Electrons in an atom revolve only along certain orbits, for whose angular momentum is an integer multiple of the Planck constant h, that is,

$$mvr = nh \tag{1}$$

where m – electron's mass, v – electron's velocity, r – radius of the electron's orbit, h – Planck's constant, n – integer positive number. An electron moving in such an orbit does not emit an energy – Fig. 1.

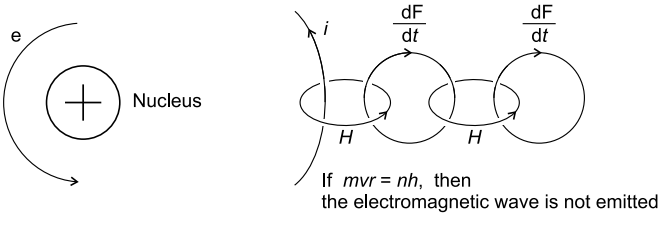


Fig. 1. An illustration of allowed orbits of electrons around nuclei of atoms, in the case when electromagnetic radiation is not emitted

2. An atom can either absorb or emit radiation in the form of an energy quantum $E = hv$, when changing the orbit from one to another – Fig. 2.

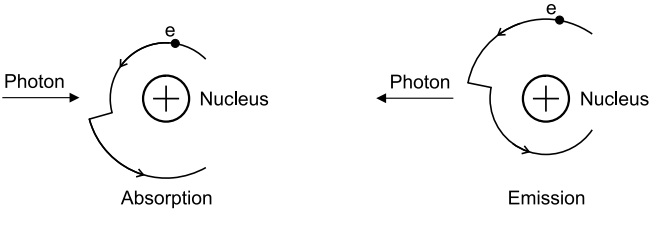


Fig. 2. An illustration of the phenomenons of absorption and emission in an atom

The explanation, why principles of classical mechanics can not be used in analysis of the structure of an atom, were not given until the works of Heisenberg and Schrodinger in 1926. Especially, until the Heisenberg uncertainty principle which states, that the position of an electron can be determined only with a given probability.

3 Contents and Structure of Molecules

Molecules are collections of atoms, connected with internal bounds. Chemistry determines the contents of the molecules, that is, the number of atoms in the molecules, for example the formula H_2O shows, that the molecule of water contains two atoms of hydro gen and one atom of oxygen.

Stereochemistry analyzes the configuration, created by atoms inside a molecule, that is, the structure of molecules. A structure of the molecule of water, for example, is typically presented as shown in Fig. 3.

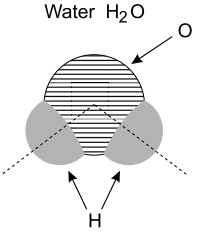


Fig. 3. Structure of the molecule H_2O

To denote subsequent atoms within a molecule, the following international colors were accepted:

carbon – black	hydrogen – white	nitrogen – blue
oxygen – white	sulfur – yellow	phosphorus – violet

In the model in the latest figure, because of the lack of possibility of using a color, oxygen has been presented an a striped region. The presented type of model of the structure of water is called a space-filling model. Besides the space-filling form, there are other frequently found types of models, for example a ball-and-stick model as seen in Fig. 4, and a model that uses atom symbols and sticks as found in Fig. 5.

Sometimes it is bandy to use ribbon types of models, that show the molecules in a very schematic way, or surface types of models, if the surface properties of a molecule are important.

In general, it can be said that the type of model is chosen, which presents best the analyzed properties of an atomic structure.

3.1 Structure of the Molecule of Methane CH_4

The molecule of methane has a fixed geometry. It is a molecule with a rigid structure. The angle H–C–H in all cases is $109° 28'$ (see Fig. 6).

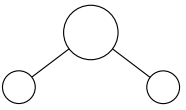


Fig. 4. Ball-and-stick form of a molecular

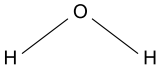


Fig. 5. A form that uses atom symbols and sticks structure of water. to show the structure of water

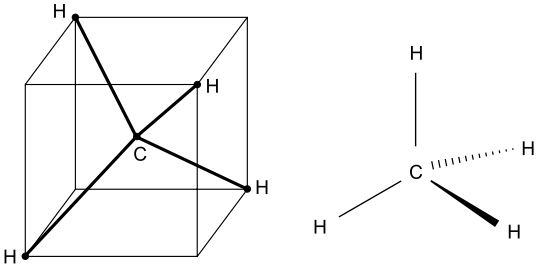


Fig. 6. Configuration of atoms of hydro gen around the atom of carbon in the molecule of methane CH_4

3.2 Structure of the Molecule of Ethane C_2H_6

The molecule of ethane can internally rotate around its C–C bound. Thus, the conformations can differ in the angle of rotation of a configuration CHHH, in respect to the other configuration CHHH (see Fig. 7).

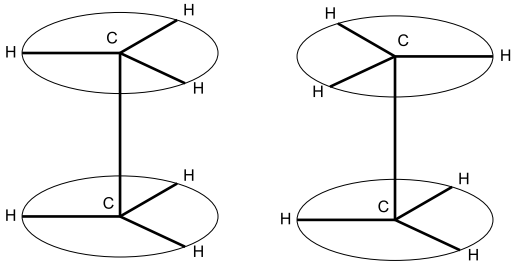


Fig. 7. Conformations opposite (left), alternating (right) of a configuration CHHH in the molecule of ethane

3.3 Properties of the Molecules of Methane and Ethane

By analyzing the molecules of methane and ethane (see Fig. 8) it can be found, that they are shaped in such a way, that the sum of distances between atoms of hydrogen is maximal, within given volume, and that the sum of distances between the atom of carbon and the atoms of hydrogen is minimal, what can be expressed using the following formulas:

$$\sum_{i=1}^{n} O(\mathrm{C}, \mathrm{H}_i) = Min \tag{2}$$

$$\sum_{i=1}^{n} O(\mathrm{H}_i, \mathrm{H}_j) = Max \tag{3}$$

where $O(a, b)$ means the distance in a three-dimensional metric space between the points a and b.

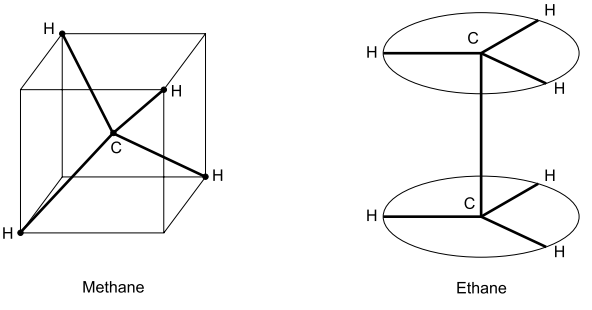


Fig. 8. Properties of structures of molecules of methane (left) and ethane (right)

4 Organization of the Process of Development of an Organism

Organization of the process of development of an organism is presented in Fig. 9.

The organizer of the process of development of an organism is a DNA molecule, which initializes the process of production of proteins needed for that development, as well as inspires their transport, according to the signals that come from the organism.

4.1 Production of Proteins

Proteins are chains of amino acids. There are 22 different molecules of amino acids, as presented in Table 4.

Let, for example, a required type of protein has a structure

Met—Ala—Lys—His—Gly .

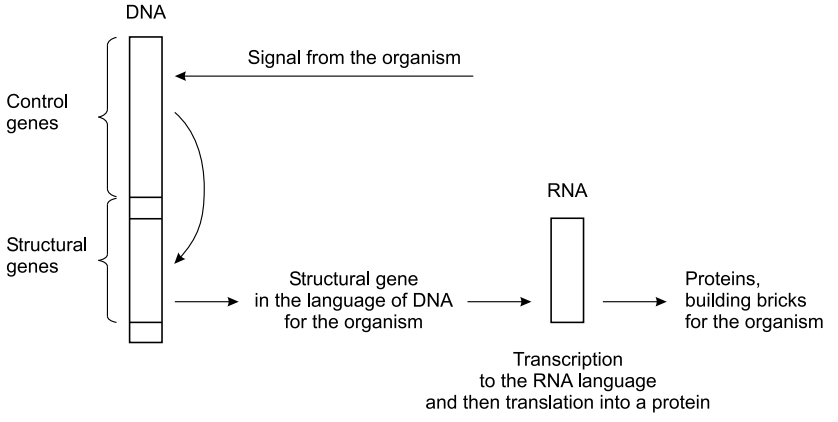


Fig. 9. Organization of the process of development of an organism

Table 1. Names and symbols of amino acids

Amino acid	Symbol	Amino acid	Symbol
Alanine	Ala	Aspargine	Asp
Cysteine	Cys	Pyrrolysine	Pyl
Aspartic acid	Asp	Proline	Pro
Glutamic acid	Glu	Glutamine	Gln
Phenylalanine	Phe	Arginine	Arg
Glycine	Gly	Serine	Ser
Histidine	His	Threonine	Thr
Isoleucine	Ile	Selenocysteine	Sec
Lysine	Lys	Valine	Val
Leucine	Leu	Tryptophan	Trp
Methionine	Met	Tyrosize	Tyr

To obtain such a protein, it is needed to enter into the collection of various amino acids an information about the required ordering, written in the language of genetic code, which has the property, that suitable molecules of amino acids will begin to adhere to the places, where occur respective notes about them in the genetic code.

4.2 Structure of a Molecular System of Informatics

The structure of a molecular system of informatics in living organisms in presented in Fig. 10.

4.3 Ordering of Unordered Molecules

Proteins are build out of a set of 22 amino acids, whose structure is presented in Fig. 11.

After the introduction to the collection of unordered molecules of the information, written in the language of the genetic code, about the required ordering, begins the process of forming of subsets being products, as illustrated in Fig. 12.

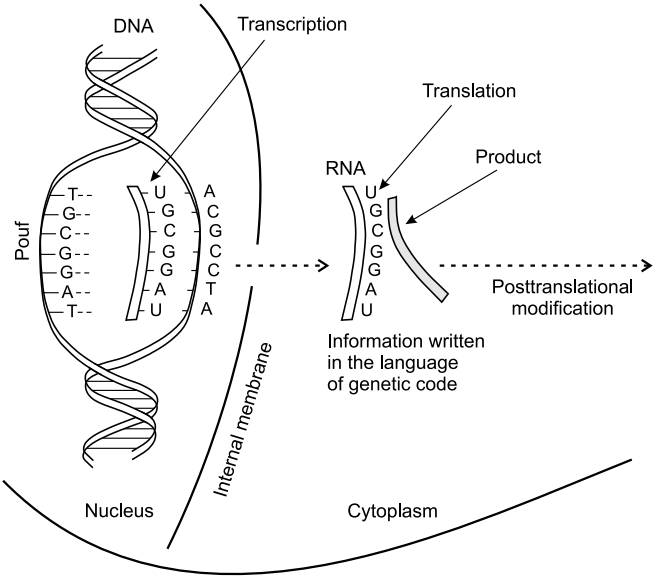


Fig. 10. Structure of a molecular system of informatics in a living organism

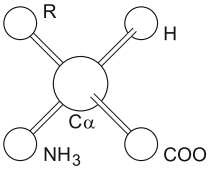


Fig. 11. Structure of the molecule of an amino acid, where Cα – atom of carbon, NR – amine group, COO – carboxylic group, H – atom of hydrogen, R – the so called side chain

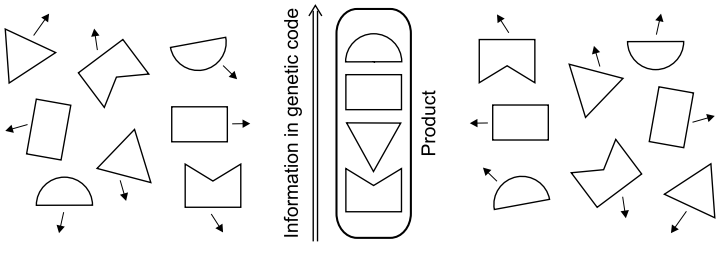


Fig. 12. Illustration of the organization of the process of ordering of unordered molecules of amino acids

4.4 Technology of Gluey Matrix

The technology of the gluey matrix is presented in Fig. 13.

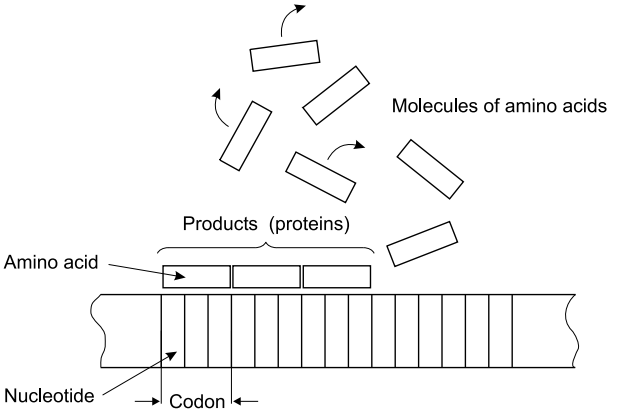


Fig. 13. Technology of the gluey matrix

4.5 Language of the Genetic Code

The alphabet of the language of the genetic code is a tour-letter alphabet. The letters are molecules of adenine, cytosine, guanine and uracil, which we will denote as A, C, G, U. Below is the notation of amino acids in the language of the genetic code. There are 22 amino acids. Each amino acid has at least one three-letter word in the language of the genetic code, called a *codon* (see Table 2).

Table 2. Codons

Amino acid	**Codons**
Alanine	GCU, GCC, GCA, GCG
Cysteine	UGU, UGC
Aspartic acid	GAU, GAC
Glutamic acid	GAA, GAG
Phenylalanine	UUU, UUC
Glycine	GGU, GGC, GGA, GGG
Histidine	CAU, CAC
Isoleucine	AUU, AUC, AUA
Lysine	AAA, AAG
Leucine	UUA, UUG, CUU, GUC, CUA, CUG
Methionine	AUG (start)
Asparagine	AAU, AAC
Pyrrolysine	UAG
Proline	CCU, CCC, CCA, CCG
Glutamine	CAA, CAG
Arginine	CGU, CGC, CGA, CGG, AGA, AGG
Serine	UCU, UCC, UCA, UCG, AGU, AGC
Threonine	ACU, ACC, ACA, ACG
Selenocysteine	UGA
Valine	GUU, GUC, GUA, GUG
Tryptophan	UGG
Tyrosine	UAU, UAC
Stop codon	UAA, UAG, UGA

We will present the process of synthesis of a given product by an example of synthesis of a protein having the following structure:

methionine, arginine, glycine, proline, glycine, alanine

what in the language of the genetic code would be written as

AUG CGU GGU CCU GGU GCU

which can be treated as a catalyst in the process of synthesis of the required protein.

4.6 Structure of the Biological Cycle of Life

The biological cycle of life consists of: self-replication, transcription, translation and posttranslational modification. The subsequent operations are realized by:

- DNA – replication and transcription;
- RNA – translation;
- PROTEINS – posttranslational modification, folding, proteins biologically active, organism.

It is schematically illustrated in Fig. 14.

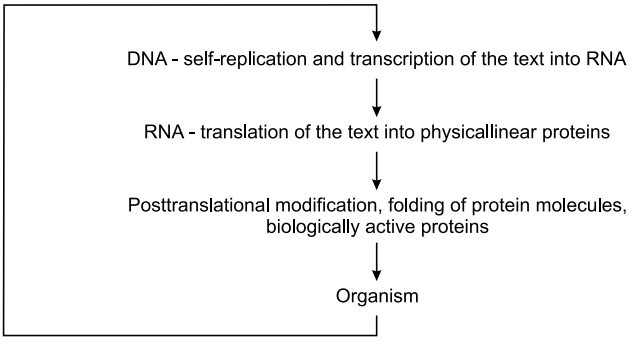


Fig. 14. Cycle of life

5 Chirality

The term chiral object origins from a Greek word cheir – a hand, and means an object, which can not be superimposed on its mirror image. An example is the structure of a left band, shown in Fig. 15.

It can be easily found out, that the structures of the methane molecule CH_4 and of the DNA helix are chiral as well.

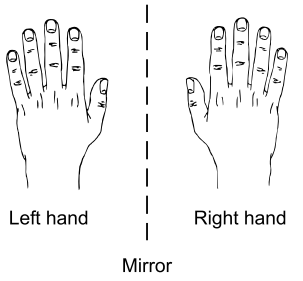


Fig. 15. Illustration of testing of chirality

5.1 Chirality of Structures of Methane and Helix

Chirality of structures of methane and helix is presented in Figs. 16 and 17.

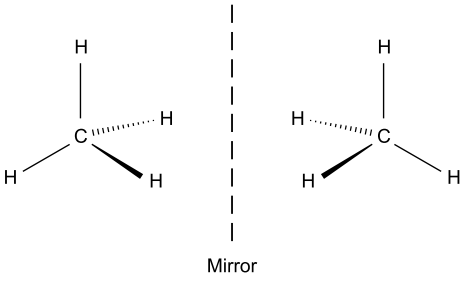


Fig. 16. Chiral structure of methane CH_4

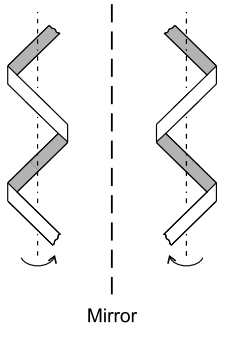


Fig. 17. Chiral structure of a helix

5.2 Chiral Addressing

Chiral addressing relies on the property, that the place of an element in an object is unambiguously determined by the structure of the element (see Fig. 18).

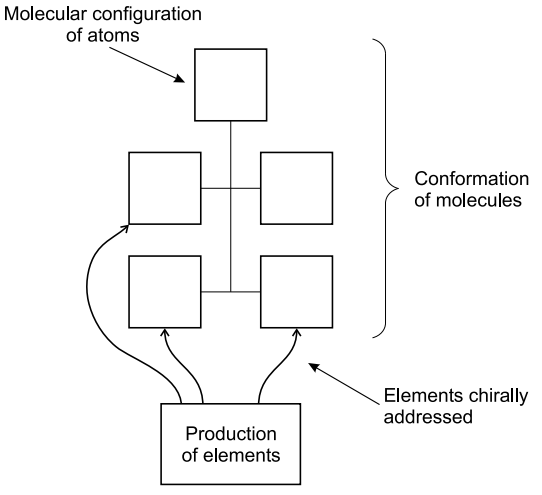


Fig. 18. Structure of chiral addressing

6 Technical and Molecular Systems of Informatics

The basic difference between technical and biological systems of informatics is the size of letter symbols. In the technical systems of informatics these symbols are the states of same systems, where in the biological systems of informatics, these symbols are directly the molecules.

Figure 19 presents formats of texts in the technical and molecular systems of informatics.

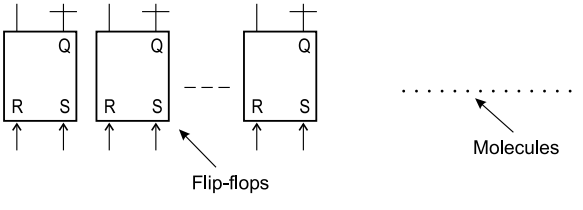


Fig. 19. Text forms in the: technical (left) and molecular systems of informatics (right)

7 Molecular Genetic Engineering

The road to the development of molecular genetic engineering has been created by a successful cut of a DNA chain into individual genes, as illustrated in Fig. 20.

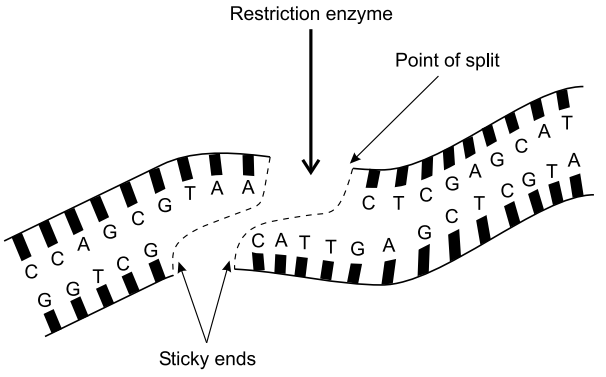


Fig. 20. Illustration of dissection of a DNA chain

Cutting of DNA is possible thanks to the use of specific chemical substances known as enzymes, which recognize specific sequences of bases in a double-stranded DNA helix and split these strands in strictly defined points. These enzymes are indispensable in sequencing of very long DNA molecules, in isolation of genes and in the creation of new DNA molecules. Restrictive enzymes are used to split DNA molecules into specific fragments, which then undergo analysis and manipulations of different kinds.

7.1 Manipulation with Genes Cut Out of DNA

Examples of operations on fragments of cut DNA are illustrated in Figs. 21 and 22.

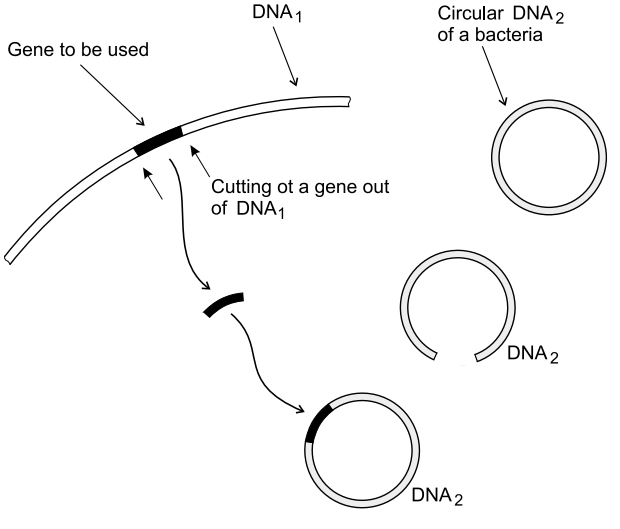


Fig. 21. Cloning in DNA_2 of a gene cut out of DNA_1

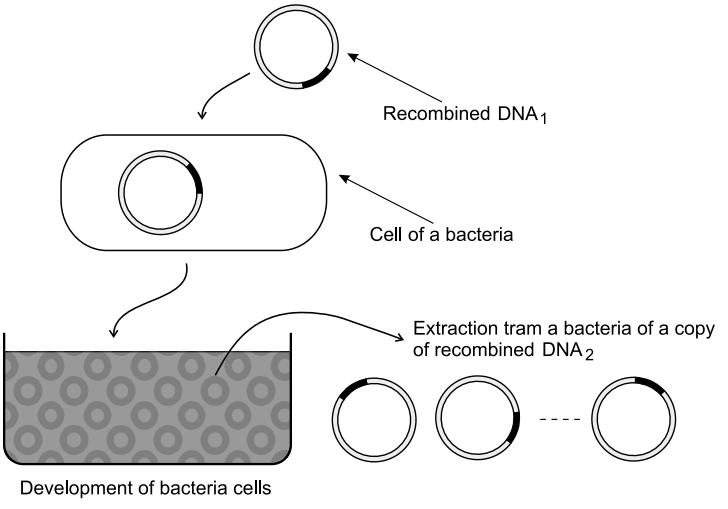


Fig. 22. Cloning of to foreign structural genes in host cells

7.2 Molecular Genetic Engineering – A Discussion

Genetic engineering relies on developing and insertion into DNA of such programs, which would lead to desired shaping of structures in products or objects to be produced. Methods of genetic engineering can be split into:

- matching and joining genes extracted out of existing DNA chains of living organisms;
- introducing changes into texts of programs of existing genes;
- undertaking of laboratory tries of technical study and production of artificial DNA chains.

Technical systems of informatics and molecular systems of informatics differ in structure and size of the fields, which they occupy. To make the structure of current technical systems of informatics and the structure of molecular systems of informatics more similar, the following changes would need to be introduced:

- replace the bit, an elementary unit of information, very convenient in realizing computational processes, with four – or more valued nanobits, depending on the way of, coding of structures of products to be manufactured;
- in the place of arithmetic logic unit, currently existing in technical systems of informatics, a system should be introduced, that uses the technology of gluey matrix, that f makes it possible to bring together and in effect join simple molecules into complex molecules, which would be the basic elements of the products to be manufactured;
- accept as the symbols of elementary units of informations not the states of same systems, but instead, directly the presence of chosen molecules.

After application of these changes, local fragments of technical systems of informatics would exhibit the course of processes similar to that, which occurs in local fragments of biological systems of informatics.

Yet, there would arise a fundamental problem, namely, if the collection of local fragments, joined into homogeneous, global entity, would accomplish the same functions, which currently are realized by biological systems of informatics?

When answering the question, one may say, that it depends on the way, in which these fragments would be joined, and if would also be reconstructed the means of both the communication of the joined fragments between each other, and the communication of the joined fragments with the environment, as it happens in biological systems of informatics. It is a problem of such a difficulty, that same biologists are inclined to accept as an axiom the claim, that a living cell must origin only from another living cell, and that a technical realization of such a cell is impossible. But there also exist such geneticist, that claim, that such a technical realization is possible, and they even specify the number of technically produced genes, which would be needed for the purpose.

The problem is not ultimately explained.

8 Consciousness

The phenomenon of consciousness, which occurs in biological systems of informatics, is still unexplained. Will it be possible to find the clarification of the yet not unfathomable phenomenon of consciousness, relying on the laws of physics and chemistry?

The beginning of an embryonic cycle is initiated by a mother cell, which in turn gives rise to the process of self-replication of cells, after which begins the next process – of differentiation and development of tissues, organs and of the structure of an organism (see Fig. 23). An unexplained problem is the phenomenon of development, in the forming organism, of consciousness.

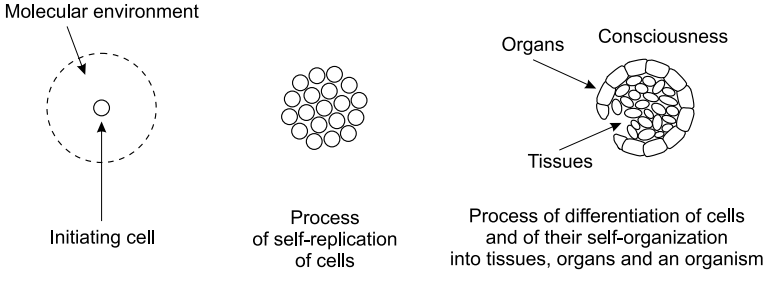


Fig. 23. Structure of an embryonic cycle

References

1. Morris, D.G.: Stereochemistry. Royal Society of Chemistry, New York (2002)
2. Berg, J.M., Tymoczko, J.L., Stryer, L.: Biochemistry, 5th edn. W. H. Freeman and Company, New York (2002)
3. Węgrzyn, S.: Nanosystems of informatics. International Journal of Systems Science 32(12), 1389–1392 (2001)

Entanglement in General Multipartite Quantum Systems and Its Role in Quantum Information Processing Tasks

Roman Gielerak

Institute of Control & Computation Engineering
University of Zielona Góra, ul. Podgórna 50, Zielona Góra 65-246, Poland
R.Gielerak@issi.uz.zgora.pl

Abstract. A major role playing by entanglement of quantum states in several, present day applications of genuine quantum technologies is briefly reviewed. Additionally, the notion and classification of multipartite entanglement has been presented. A new, monotone under (S)LOCC-operations measures of many-partite entanglement are defined and discussed briefly.

Keywords: quantum key distribution,teleportation protocols, multipartite entanglement,LOCC monotone functions.

1 Introduction

The opportunity of using genuine quantum behaviour of real matter at small scale has been considered for quite a long period as an interesting albeit purely academic challenge for developing a new computation model to perform effective computations of classically hardly computable functions [1]. The first real breakthrough in this area happened in the course of famous 1981 MIT 1st Conference on Physics and Computations [2] when Feynman has been suggested to use another quantum system to simulate a genuine quantum behaviour of a given quantum system as the only possibility to compute effectively the arising quantum effects. Soon, after this Conference the research activity in this narrow, academic area has been significantly increased. However the real eruption of activity and interest can be observed after discovery by P.Shor [3] of factorising big integers into prime factors algorithm working in polynomial time. As is widely known this is the only possibility to break the security of most popular cryptography protocols introduced in present day electronic communication industry. But, the point here is that in order to implement the Shor algorithm to factorise a big, 1024 bits integer in an acceptable time a quantum machine of a rather large scale (with quantum registers composing of thousands of quantum logical units) is necessary. Despite to unprecedented activity of all over the world scientific society (presumably, today it is hardly to find a developed country in the world where there are no several research groups that are working on this challenge) the present day existing and working machines looks to be

A. Kwiecień, P. Gaj, and P. Stera (Eds.): CN 2010, CCIS 79, pp. 15–28, 2010.

rather like small toys for kids. See [4,5,6] for a report on the present status of project of constructing Quantum Computer. A several, major breakthroughs in the fundamental Quantum Physics have to be achieved until the final, ultimate goal which is scalable Quantum Computing Machine working at our Macroscopic (and therefore classical) Reality will start to do an useful work.

The observed in last decades,exponential growth of computational power of all kinds of classical computers, PC including,sometimes summarised as Moor's Law is achieved mainly due to the technology miniaturisation process. Smaller and smaller chips like transistors consume less power, can be packed more densely and switch faster and faster. However. already today there are some technological processes that are to be controlled on atomic scales,dioxide insulating layers within each transistors produced is a good example of this. Providing the miniaturisation will progress with speed as observed in last decades a typical for genuine quantum effects domination scale (it is theoretically predicted that the border in between classical and quantum is of size of 10 nanometers) will be reached soon. Then the necessity, not only the opportunity (as today), of understanding deeper the quantum behaviour and the use of it to perform a useful calculations on this quantum scale should be stressed. Otherwise a serious slowing down of the progress in the computational power of computer technology do appear and the only today predictable possibility to breakthrough the forthcoming crisis is to build quantum computing machines. There are a number of quantum computing candidates, among those: superconductor-based quantum computers (including SQUID-based quantum computers), trapped ion quantum computer, optical lattices, topological quantum computer, quantum dot on surface (e.g. the Loss-DiVincenzo quantum computer), nuclear magnetic resonance on molecules in solution (liquid NMR), solid state NMR Kane quantum computers, electrons on helium quantum computers, cavity quantum electrodynamics (CQED), molecular magnet, fullerene-based ESR quantum computer, optic-based quantum computers, diamond-based quantum computer, BoseEinstein condensate-based quantum computer, transistor-based quantum computer – string quantum computers with entrainment of positive holes using an electrostatic trap, spin-based quantum computer, adiabatic quantum computation, rare-earth-metal-ion-doped inorganic crystal based quantum computers.

Such a large number of candidates shows explicitly that the topic, in spite of rapid progress, is still in its infancy. But at the same time there is also a vast amount of flexibility.

There is no doubts that quantum entanglement is this genuine quantum property that is the key resource of many fruitful applications of quantum information processing technology. For example, the realistic applications of entangled states include theirs major applications for quantum key distribution protocols (Quantum Cryptography, [7,8,6]). A still futuristic but fascinating visions of using quantum channels for communication purposes applies entanglement as the main tool for teleportation protocols implementations, see [9] and the so called dense coding protocols, see [10]. It is not excluded that the use of quantum channels will play more and more important role in future communication

technologies. The vision of global communications networks with an appropriate quantum communication channels present inside them seems to be a very realistic vision of a near future.

2 Some Applications of Quantum Entanglement to the Future Quantum Communication Technologies

Some introductory material for reading this section, at least on the level for beginners, can be found in [11] and [12].

2.1 Quantum Cryptography Protocols Based on Quantum Key Distribution (QKD)

At present the security of the widely used RSA and connected, theory of numbers based encryption protocols relies on the computational complexity of finding period of an element in multiplicative group Z_N^*, for N a sufficiently big integer. In particular, the difficulty of factorization can be overcomed providing a suitable algorithm for computing periods of elements of Z_N^* is known. RSA is used to establish secure connections over Internet, enabling the transition of sensitive data such as passwords,credit cards details and online banking sessions. RSA also forms the heart of secure messaging protocol PGP (Pretty Good Privacy). As we have pointed out RSA is only conditionally secure and the construction of sufficiently large scale quantum computer will destroy this kind of security completely.

Quantum cryptography sometimes called quantum key distribution (QKD) offers absolute security of the corresponding encryption protocols as opposite to the only conditional security offered by all present day available classical technologies. Let us recall that already in 1917 Vernam [13] proposed one-time pad encryption scheme the absolute security of which was proven by Shannon in 1947 [14]. However the serious drawbacks of one-time pad among which the most serious is the question of session key distribution seems to be the most crucial ones for makings them insecure and impractical.

Quantum technology offers instead a very promising solution to all of the mentioned drawbacks of one-time pad protocol. There are mainly two types of QKD schemes. The first type of QKD schemes is based on quantum measurement process applied to earlier prepared and distributed states and includes several protocols among which the best known are BB84 protocol [15], B92 [16] protocol and others as well [17]. The other types of QKD schemes are entanglement based QKD schemes, the Ekert91 [18] and BBN92 [19] are the best known representatives of this family.

At the heart of any of entanglement based QKD protocol is the quantum entanglement of the used quantum states and Bell theorem known also as the violation of Bell inequalities in presence of entanglement [20]. In particular case of Ekert91 scheme the sender (traditionally called as Alice) and the receiver (Bob) are assumed to share a rather big reservoir of maximally entangled pairs

of photons. The photons are distributed in such a way that Alice and Bob each end up with one photon from each pair. We have to assume that Alice and Bob are space like separated (in accordance with the locality principle in 4D Minkowski space-time and the assumptions for applying Bell inequalities) and that they both measure polarisations of theirs photons. Although the particular results of their measurements are completely random certain correlations among theirs particular outcomes obtained are to be hold as predicted by Bell theorem. It follows that any attempt at eavesdropping (by Eve) will disturb these correlations in such a way that Alice and Bob immediately will be able to detect them. However, the disturbances of quantum correlations might have its origin not only in eavesdropping but also may be caused by imperfections in the transmission line and detectors. Therefore the presented scheme of QKD must be complemented with two additional steps. The first step should eliminate the erroneous bits caused by noise of channels used and this becomes to be possible by the so called information reconciliation step. Information reconciliation is a form of error correction carried out between Alice and Bob's keys, in order to ensure both keys are identical. It is conducted over the public channel and as such it is vital to minimise the information sent about each key, as this can be read by Eve. A common protocol used for information reconciliation is the cascade protocol, proposed in 1994 [21]. This operates in several rounds, with both keys divided into blocks in each round and the parity of those blocks compared. If a difference in parity is found then a binary search is performed to find and correct the error. If an error is found in a block from a previous round that had correct parity then another error must be contained in that block; this error is to be found and corrected as before. This process is repeated recursively, which is the origin of the cascade name for it. After all blocks have been checked, Alice and Bob both reorder their keys in the same random way, and a new round begins. At the end of multiple rounds Alice and Bob have identical keys with high probability, however Eve has gained some additional information about the key from the parity information exchanged. The second step necessary in the presence of eavesdropper is to reduce the amount of catched by him information about the session key below some minimal level (in fact reduction to any low level is possible) from which the reconstruction of the session key will be impossible at all. For this purpose the so called Privacy Amplification procedure has been formulated. Privacy Amplification is a method for reducing (and effectively eliminating) Eve's partial information about common Alice and Bob's key session. This partial information could have been gained both by eavesdropping on the quantum channel during key transmission (thus introducing detectable errors), and on the public channel during information reconciliation (where it is assumed Eve gains all possible parity information). Privacy amplification uses Alice and Bob's key to produce a new, shorter key, in such a way that Eve has only negligible information about the new key. This can be done using a universal hash function, chosen at random from a publicly known set of such functions, which takes as its input a binary string of length equal to the key and outputs a binary string of a chosen shorter length. The amount by which this new key is

shortened is calculated, based on how much information Eve could have gained about the old key (which is known due to the errors this would introduce), in order to reduce the probability of Eve having any knowledge of the new key to a very low value. See [6] for details concerning both additional steps.

The complete mathematical proof of unconditional, absolute security of QKD protocols was appeared to be not an easy problem which was solved only in the last few years [17,19,22]. However, in real applications of QKD realistic devices used for that purpose contain many imperfections comparing with mathematically idealized QKD protocols to which the above mentioned proofs are referring. Summarising: the question of absolute security of realistic performance of QKD is still under active debate. Although there is still a big gap in between the theory and practise of QKD a lot of experimental work has been done successfully in the last years. In particular successful implementations of QKD protocols based on entanglement have been realised by free-space transfer of entangled pairs of photons over 144 km distance [23]. Concerning the present day realistic implementations of QKD schemes we note the longest distance over which QKD has been demonstrated using optic fibre is about 148.7 km (BB84 protocol) [24]. As it concerns free space the best result as it was already mentioned is 144 km ,the distance in between two of the Canarian Islands (Ekert91) [23] and (BB84 protocol) [25]. Attempts to extend QKD schemes to satellite communication level are also started with. At present there are at least four companies offering commercial quantum cryptography equipement: idQuantque (Geneva), MagiQ Technologies (New York), SmartQuantum (France) and QuintessenceLabs (Australia). It is interesting to recall that already in 2008 the first computer network protected exlusively by QKD protocols has been successfully tested during some scientific conference in Vienna. The name of this network is SECOQC (Secure Communication Based on Quantum Cryptography) and EU funded this project. The network used 200 km of standard fibre optic cable to interconnect six locations across Vienna and the town of St. Poelten located 69 km to the west [26].

Despite to powerfull progress in QKD implementations still, some crucial from the point of view of commercial applications problems, like the rather low bit rate and the problem of amplifications of quantum signals (the quantum repeater problem) making the transmission possible on a relatively small distances only has to be solved before QKD will start to dominate in computer security technology. The highest bit rate system currently demonstrated exchanges secure keys at 1 Mbit/s (over 20 km of optical fibre) and 10 kbit/s (over 100 km of fibre), achieved by a collaboration between the University of Cambridge and Toshiba and using the BB84 protocol with decoy pulses [27].

2.2 Quantum Teleportation Protocols and Dense Coding Protocols

It is not excluded, even more it is clear that the future communication technologies will use intensively quantum channels for transferring quantum and classical information purposes together with classical channels based technology. Especially interesting seems to be a vision of using quantum channels for performing the process of teleportation of classical and quantum as well information

and sending classical information through quantum channels using dense coding based protocols fot this purpose. Quantum teleportation, or entanglement-assisted teleportation, is a technique used to transfer quantum information from one quantum system to another. It does not transport the system itself as it is described by sciences-fiction writers, nor does it allow communication of information at superluminal (faster than light) speed as it seems to us today. Neither does it concern rearranging the particles of a macroscopic object to copy the form of another object. Its distinguishing feature is that it can transmit the information present in a quantum superposition, useful for quantum communication and computation. Again entanglement of quantum states used lies in the very origin of this way of communication. To start with let us explain in details a general teleportation protocol by which we can send both quantum and classical information encoded in quantum states. Reversing teleportation protocol we obtain the dense coding protocol by which we can send much more classical information then in any of previously known and used classical protocols.

The standard teleportation scheme involve three-partite quantum system $A \wedge B \wedge C$ with the corresponding Hilbert spaces of states $\mathcal{H}_A$, $\mathcal{H}_B$ and $\mathcal{H}_C$. It is assumed that a general mixed state ρ is given in the sector A and the problem is to "send it" to the spacely separated sector C. In the standard teleportation scenario it is assumed that an additional and sufficiently entangled (in most cases the maximally entangled state is used) bipartite state $\omega_{BC} \in E(\mathcal{H}_B \otimes \mathcal{H}_C)$ is at disposal of both AB and C. The teleportation based transfer of information contained in the state ρ allows to perform only local with respect to the decomposition $AB + C$ and physical operations (i.e. only completely positive and local operations are allowed) together with classical communication channel through which certain amount of supplementary (to conclude the teleportation process) classical information can be send. In most cases the above mentioned local operations are measurement operations which in general are realised through the Positive Operator Valued Maesure (POVM) $\mathbb{F}$ with a discrete spectrum $\sigma(F)$, see [28]. The first step of the protocol is to perform the measurement of $\mathbb{F}$ and if $x \in \sigma(\mathbb{F})$ is resulted then this information is sent through the adjusted classical channel to the receiver C. The receiver C must have a suitable library of keys $\{K(x), x \in \sigma(\mathbb{F})\}$ which are local (acting in C sector only) physical operations and such that if he applies the proper key $K(x)$ to its end of the state ω_{BC} i.e. $(\mathbb{I}_B \otimes K_x)(\omega_{BC})$ then the result of this operation should be exactly equal to ρ. All this can expressed by the following equality:

For any observable $A = A^\dagger \in \mathcal{H}_A$

$$\mathrm{Tr}_{\mathcal{H}_A}(\rho A) = \sum_{x \in \sigma(\mathbb{F})} \mathrm{Tr}\,(\rho \otimes \omega_{BC})\,(F_x \otimes K_x(A)) \tag{1}$$

where it is assumed (for simplicity only) that the space $\mathcal{H}_B$ is isomorphic to $\mathcal{H}_C$ i.e. $\dim(\mathcal{H}_B) = \dim(\mathcal{H}_C)$. A graphical exposition of the described teleportation process is presented in Fig. 1.

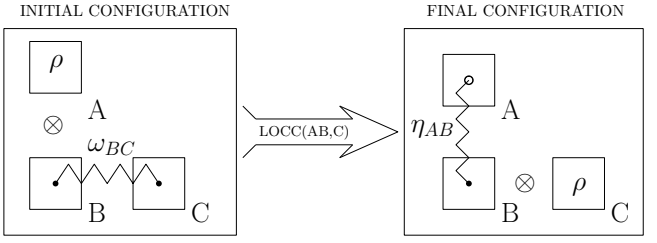


Fig. 1. The general scheme of teleportation

The fundamental question is of course whether teleportation processes as described are possible at all and both on theoretical level and experimental as well.

Example 1. Teleportation of pure states of qubit protocol [29].

This was historically the first teleportation protocol discovered for teleporting pure states of qubit. In this case the corresponding Hilbert spaces are given by 2d complex Euclidean spaces C^2 and the corresponding entangled state (which is maximally entangled state in fact) is given by:

$$|\Psi_{BC}\rangle = \frac{1}{\sqrt{2}}\left(|00\rangle + |11\rangle\right) \quad . \tag{2}$$

Let $|\Psi_A\rangle$ be any pure state of qubit that will be teleported. The measurement arrangement here reduces to the orthodox von Neumann measurement in the of Bell vectors composed orthonormal basis of $C^2 \otimes C^2 = C^4$ which are defined as:

$$\begin{array}{l} |\phi_{\pm}\rangle = \frac{1}{\sqrt{2}}\left(|00\rangle \pm |11\rangle\right) \\ |\psi_{\pm}\rangle = \frac{1}{\sqrt{2}}\left(|01\rangle \pm |10\rangle\right) \end{array} \quad . \tag{3}$$

The only possible outcomes of such an experiment are given as pairs of numbers (00), (01), (10) and (11) and all can arise with the same probability equal to 0.25. The resulted pair of bits is then send to receiver C through classical channel and the corresponding library of keys used by him in order to obtain a state $|\Psi_A\rangle$ in his register is given by $\{\mathbb{I}, \sigma_Z, \sigma_X, i\sigma_Y\}$. This is depicted graphically in Fig. 2.

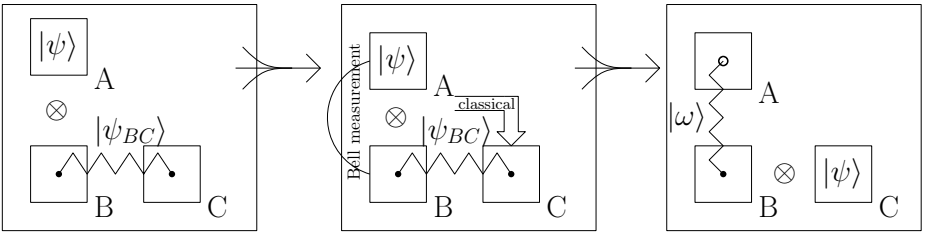


Fig. 2. Teleporting of qubit state using maximally entangled state $|\psi_{BC}\rangle$ and orthogonal measurement in Bell basis together with classical channel

It is interesting to observe that from mathematical point of view the teleportation process just described is nothing more then the following algebraic identity:

$$|\Psi\rangle|\Psi_{BC}\rangle = \frac{1}{2}|\phi_+\rangle(\mathbb{I}|\Psi_A\rangle) + \frac{1}{2}|\phi_-\rangle(\sigma_Z|\Psi_A\rangle) + \\ + \frac{1}{2}|\psi_+\rangle(\sigma_X|\Psi_A\rangle) + \frac{1}{2}|\psi_-\rangle(-i\sigma_Y|\Psi_A\rangle) \ . \tag{4}$$

It follows from the anylysis of Werner [30] that the following general result can be proven:

Theorem 1. *Let us assume that* $\dim(\mathcal{H}_A) = \dim(\mathcal{H}_B) = \dim(\mathcal{H}_C) < \infty$. *Then there exist infinitely many unitarly unequivalent scenarios of teleportation protocols as desribed in Fig. 1.*

Concerning experimental demonstrations of teleportation we referr to [31] (photonic states) and [32] (atomic states).

3 The Mathematical Notion of Entanglement

Let $\mathcal{G}$ be a quantum system composed from n smaller parties $\mathcal{G}_i$ which we assume are separated spatially, however, a classical communication in between them is still allowed. From quantum physics we know that to subsystems $\mathcal{G}_i$ there are associated in an unique (up to unitary isomorphisms) Hilbert spaces $\mathcal{H}_i$ of states and the states of composite system $\mathcal{G}$ are associated with the space $\mathcal{H} = \mathcal{H}_1 \otimes \mathcal{H}_2 \otimes \ldots \otimes \mathcal{H}_n$, where $\otimes$ means the (completed if dimensions of $\mathcal{H}_i$ are infinite) tensor product operation. The pure states of $\mathcal{G}$ are then given by (rays) normalised vectors $|\psi\rangle \in \mathcal{H}$ and the corresponding space of pure states will be denoted as $\partial E(\mathcal{H})$. The corresponding mixed states are given by linear, of trace class semi-positive maps of $\mathcal{H}$ and form a compact and convex subset $E(\mathcal{H})$ the topological boundary of which is exactly the set $\partial E(\mathcal{H})$. The source of all of problems connected with entanglement is that $E(\mathcal{H})$ is not a Choquet simplex and therefore there do exist infinitely many different decompositions of a given state $\rho \in E(\mathcal{H})$ into pure states from $\partial E(\mathcal{H})$.

Let $Par(n)$ be a set of all partitions of a given n-element set $X = \{x_1, \ldots, x_n\}$, the number of elements of $Par(n)$ is given by the corresponding Bell number $B(n)$. For a given partition $\pi \in Par(n)$ of the form $\pi = (X_1, \ldots, X_k)$ the number k stands for the length $|\pi|$ of π. The following partial order $\prec_f$ will be used in the following: we say that a partition $\pi = (X_1, \ldots, X_k)$ is finer that a partition $\pi' = (Y_1, \ldots, Y_l)$ iff for each $i \in \{1, \ldots, k\}$ there exists $l \in \{1, \ldots, l\}$ such that $X_i \subseteq Y_l$. This will be denoted as $\pi \prec_f \pi'$. Then, in the poset $(Par(X), \prec_f)$, there exists a maximal element $\pi_{\max} = (X)$ and minimal $\pi_{\min} = (\{x_1\}, \ldots, \{x_n\})$ as well.

Let $|\psi\rangle \in \partial E(\mathcal{H})$. We will say that the vector state $|\psi\rangle$ is π-separable state, where $\pi = (X_1, \ldots, X_k)$, iff $|\psi\rangle$ can be written as

$$|\psi\rangle = \bigotimes_{i=1}^{k} |\psi_{X_i}\rangle, \text{ where } |\psi_{X_i}\rangle \in \partial E(\bigotimes_{\alpha \in X_i} \mathcal{H}_\alpha) \tag{5}$$

and π is the finest partition for which (5) hold.

In particular a state $|\psi\rangle \in \partial E(\mathcal{H})$ is called completely separable iff $|\psi\rangle$ is $\pi_{\min}$-separable. The dual notion is that of partial entanglement. A state $|\psi\rangle \in \partial E(\mathcal{H})$ is called completely entangled iff $|\psi\rangle$ is $\pi_{\max}$-separable and is π-entangled iff it is π-separable and $\pi \notin \{\pi_{\max}, \pi_{\min}\}$. If $|\psi\rangle$ is π-separable state, $\pi = (X_1, \ldots, X_k)$, then for any $i : |X_i| \geq 2$ the state $|\psi_{X_i}\rangle$ is completely entangled on X_i.

A similar definition can be introduced to treat the case of general quantum states $\rho \in E(\mathcal{H})$. A state $\rho \in E(\mathcal{H})$ is called π-separable state iff there exists $\pi \in Par(\{1, \ldots, n\})$ for which the following representation exists:

$$\rho = \sum_{\alpha} P_\alpha \otimes_{i \in \{1,\ldots,|\pi|\}} \rho^\alpha_{X_i}, P_\alpha \geq 0, \sum_{\alpha} P_\alpha = 1, \rho^\alpha_{X_i} \in E(\bigotimes_{j \in X_i} \mathcal{H}_j). \tag{6}$$

Having a particular state $|\rho\rangle \in E(\mathcal{H})$ we want to answer the question whether this state is entangled or not. In the two-partite case certain useful criterions have been developed in the past twenty years due to activity of many peoples. For an extensive and up to date well written reviews see [33,34].

In the case of two-partite systems and for pure states the complete answer to the question of entanglement and the amount of it is provided by the Schmidt decomposition theorem [28]. In the case of general two-partite systems (however the finite dimensional situations are mainly explored till now) several techniques checking whether a given mixed state ρ is entangled or not have been worked out and certain methods for definite answering this question (based on Linear Programming Algorithms) are at our disposal at present. Although the definite answer to the question of entanglement can be obtained the important point is that this problem is known to be NP-HARD problem [35]. However, a plenty of polynomially complex methods can be used for this purpose also. However, the price for a use of a reasonable calculationaly complex method is that all of them are not giving definite answer to the question of entanglement. The so called PPT-criterion of Peres [28,36,33] the cross norm criterion [33,34] and several witness constructions are known examples [34,33].

So, we can assume that a suitable procedure enabling us to check whether a given state of 2-partite system $\mathcal{H} = \mathcal{H}_1 \otimes \mathcal{H}_2$ is separable or entangled is given.

2p-Oracle: Is entangled?

input : $\mathcal{H} = \mathcal{H}_A \otimes \mathcal{H}_B$, $\rho \in E(H)$, the 2p decomposition
output: YES, if ρ is entangled, NO if ρ is separable

For further use let us recall what the Schmidt decomposition theorem tells us. For any $|\psi\rangle \in \mathcal{H}_1 \otimes \mathcal{H}_2$ there exists a sequence of strictly positive numbers

$(\lambda^1_{|\psi\rangle}, \ldots, \lambda^k_{|\psi\rangle}) \equiv \lambda_{|\psi\rangle}$ called a Schmidt coefficients of $|\psi\rangle$ and where $k = \min(\dim(\mathcal{H}_1), \dim(\mathcal{H}_2))$ is the so called Schmidt rank of a vector $|\psi\rangle$ and a pair of ON bases $\{\Theta^1_i\}, \{\Theta^2_i\}$ in $\mathcal{H}_1$ and respectively in $\mathcal{H}_2$ and such that the following eguality holds:

$$|\psi\rangle = \sum_{i=1}^{k} \lambda^i_{|\psi\rangle} |\Theta^1_i\rangle |\Theta^2_i\rangle \ . \tag{7}$$

SD-Function

input : $\mathcal{H} = \mathcal{H}_A \otimes \mathcal{H}_B$, $|\psi\rangle \in \partial E(H)$
output: $[r_{|\psi\rangle}, \lambda_{|\psi\rangle}, \{\Theta^1_i\}, \{\Theta^2_i\}]$

where $r_{|\psi\rangle}$ is the Schmidt rank of $|\psi\rangle$, $\lambda_{|\psi\rangle} = (\lambda^1_{|\psi\rangle}, \ldots, \lambda^k_{|\psi\rangle})$, $k = r_{|\psi\rangle} \leq \min(\dim \mathcal{H}_1, \dim \mathcal{H}_2)$, $\{\Theta^i_\alpha\}, i = 1, 2$ are the corresponding ON-bases in $\mathcal{H}_i$

In the paper [37] certain algorithms for answering the question of partial separability (and thus of entanglement) in the case of n-partite systems have been presented. We summarise the results of [37].

n-partite-Oracle: Is entangled?

input : $\mathcal{H} = \mathcal{H}_1 \otimes \mathcal{H}_2 \otimes \ldots \otimes \mathcal{H}_n$, the decomposition of the system, $\rho \in E(H)$
output: $X = (X_1, \ldots, X_k) \in Par(I_n)$ such that ρ is X-separable

However, the serious drawback of methods introduced in [37] is that they are all of nonpolynomial (in n) calculational complexity.

One of the important corollaries of [37] is that in the case of n-partite systems several incomparable (here we mean the comparison by the use of (S)LOCC class of operations and appropriately adopted to the case discussed, see below) kinds of partial entanglement do appear.

Let $\rho \in E(\mathcal{H})$ and let $X = (X_1, \ldots, X_k)$ be the corresponding partition given by the use of previous n-partite Oracle. Any reasonable measure $\mathcal{M}_n$ of amount of entanglement included in ρ should obey the following requirements:

($\mathcal{E}_n(1)$) $\mathcal{M}_n$ is separating the subset of complete separable states from the remaining (open) part of $E(\mathcal{H})$,

($\mathcal{E}_n(2)$) $\mathcal{M}_n$ must be invariant under $LU(X)$-operations, when $LU(X) = U(X_1) \otimes \ldots \otimes U(X_k)$ for $X = (X_1, \ldots, X_k) \in Par(I_n)$ and $U(X)$ is the group of unitaries acting on $\mathcal{H}_X$,

($\mathcal{E}_n(3)$) $\mathcal{M}_n$ must be monotone under $(S)LOCC(X)$ class of operations, which means that for any $\rho \in E(\mathcal{H})$ and with $X = (X_1, \ldots, X_k)$ as above, for

any $X^{'} \prec_{finer} X$ and the action of local physical operations Θ, local with respect to the partition $X^{'} = (Y_1, \ldots, Y_i)$:

$$\mathcal{M}_n(\Theta(\rho)) \leq \mathcal{M}_n(\rho) \ ,$$

$(\mathcal{E}_n(4))$ $\mathcal{M}_n$ should obey natural continuity requirements.

Even in the case of 2-partite systems it is not an easy task to construct the corresponding function, although some definition and proceduras are given the real drawback of all of them is that they are hardly to be effectively calculable [38]. A mathematical proof of the generic NP-hardness of computing such functions is known [35].

Any function $\mathcal{M}_n$ obeying $\mathcal{E}_n(2)$ – $\mathcal{E}_n(3)$ (and $\mathcal{E}_n(4)$) will be called n-partite entanglement monotone (resp. continuous n-partite entanglement monotone) function.

Now, we present an example of such n-partite monotone that will be called weakest k-clique entangled. For this we assume that $\rho \in E(\mathcal{H})$ is completely entangled state.

For any $k \in \{1, \ldots, \frac{n}{2}\}$ let $C_k(n) \equiv \{$the set of all k-elements subsets of $I_n = \{1, 2, \ldots, n\}\}$. The elements of $C_k(n)$ will be called k-cliques in the following. Any $X \in C_k(n)$ definies 2-partition $(X, I_n \setminus X)$ of I_n. Let $\mathcal{M}_2$ stands for any 2-partite monotone. With the use of $\mathcal{M}_2$ we define the strength of entanglement in between X and $I_n \setminus X$ that is contained in the state ρ:

$$\widetilde{es}_k(X, X^c)(\rho) = \mathcal{M}_2(\rho, \mathcal{H}_X, \mathcal{H}_{X^c}) \tag{8}$$

Then the weakest k-clique entangled is defined as

$$wes_k(\rho) = \inf\{\widetilde{es}_k(X, X^c)(\rho)\}, X \in C_k(n) \ . \tag{9}$$

Our main result of this contribution is formulated now:

Theorem 2. *Let $\mathcal{H} = \mathcal{H}_1 \otimes \mathcal{H}_2 \otimes \ldots \otimes \mathcal{H}_n$ be a n-partite system and let $\rho \in E(\mathcal{H})$ be completely entangled state. Then for any 2-partite continous monotone $\mathcal{M}_2$ and any $k \in \{1, \ldots, \frac{n}{2}\}$ the function wes_k as defined in (9) is (continous) n-partite entanglement monotone.*

A detailed discussion of the introduced n-partite entanglement monotone measures shall be presented in a separate, forthcoming publications where also certain numerical examples obtained by the use of Zielona Gora Quantum Computing System [39] will be presented as well.

4 Summary

The quantum entanglement is this genuine quantum property by use of which, otherwise impossible tasks may be achieved. Among the best known applications of entanglement are: quantum key disribution offering an absulute secure

exchange of information, superdense coding, quantum state teleportation, information exchanges through time and many ohers as well, most of them is even hardly to predict today. Quantum entanglement already now has found many interesting applications in present day state of art in the emerging technologies of quantum computing and quantum cryptography, for example it has been used to realize practically quantum teleportation tasks. At the same time, it prompts some of the more philosophically oriented discussions concerning quantum theory. The correlations predicted by quantum mechanics, and observed in experiment, reject the principle of local realism, which is that information about the state of a system can only be mediated by interactions in its immediate surroundings and that the state of a system exists and is well-defined before any measurement [40]. Efforts to quantify the amount of entanglement included in a typically entangled quantum states are therefore very important to understand better a very nature of quantum world.The present paper contains both the very condensed review of the actual applications of quantum entanglement in the existing technology and also presents the main ideas of our very recent attempts to understand qualitatively the phenomenon of multipartite entanglement [37].

References

1. Deutsch, D.: Quantum Theory the Church-Turing principle and the universal quantum computer. Proceed. R. Soc. 400, 97–117 (1985)
2. Feynman, R.P.: Keynote talk by R.P. Feynman, 1st Conference on Physics and Computations. MIT, Cambridge (1981); International Journal of Theoretical Physics 21, 467–488 (1982)
3. Shor, P.: Algorithms for quantum computations: discrete log and factoring. In: Goldwasser, S. (ed.) Proceedings of the 35th Annual Symposium on the Foundations of Computer Sciencem, pp. 124–134. IEEE Computer Society Press, Los Alamitos (1994)
4. Petit, C.: Quantum Computer Simulates Hydrogen Molecule Just Right. Science News (January 2010)
5. DiCarlo, L.: Demonstration of two-qubit algorithms with a superconducting quantum processor. Nature 460, 240–244 (2009)
6. Bennett, C.H., Bessette, F., Brassard, G., Salvail, L., Smolin, J.: Experimental Quantum Cryptology 5(1), 3–28 (1992)
7. Gisin, N., Ribordy, G., Tittel, W., Zbinden, H.: Quantum cryptography. Rev. Mod. Phys. 74, 145–194 (2002)
8. Lo, H.-K., Lutkenhaus, L.: Quantum Cryptography: from theory to practise. Physics in Canada 63, 191 (2007)
9. Vaidman, L.: Teleportation of Quantum States. Phys. Rev. A 49, 1473–1476 (1994)
10. Peres, A.: What is actually teleported? IBM Journal of Research and Development 48(1)
11. Bugajski, S., Klamka, J., Wegrzyn, S.: Foundation of quantum computing. Part 1. Archiwum Informatyki Teoretycznej i Stosowanej 13(2), 97–142 (2001)
12. Bugajski, S., Klamka, J., Wegrzyn, S.: Foundation of quantum computing. Part 2. Archiwum Informatyki Teoretycznej i Stosowanej 14(2), 93–106 (2002)
13. Vernam, G.S.: Cipher printing telegraph systems for secrets wire and racho telegram communications. J.AIEE, 109 (1926)

14. Shannon, C.: Communication theory of secrecy systems. Bell System technical Journal 28(4), 656 (1949)
15. Bennett, C.H., Brassard, G.: Quantum Cryptography: Public key distribution and coin tessing. In: Proc. IEEE Computer Systemsand Signal Processing, pp. 175–179. IEEE, Bangalore (1984)
16. Bennett, C.H.: Quantum cryptography using any two nonorthogonal states. Phys. Rev. Lett. 68, 3121 (1992)
17. Lo, H.-K., Chau, H.-F.: Unconditional security of quantum key distribution over arbitrary long distances. Science 283, 2050
18. Ekert, A.K.: Quantum cryptography based on Bell theorem. Phys. Rev. Lett. 67, 661 (1991)
19. Acin, A., Gisin, N., Masanes, L.: From Bell theorem to secure QKD. Phys. Rev. Lett. 97, 120405 (2006)
20. Bell, J.S.: Speakable and unspeakable in quantum mechanics. Cambridge University Press, Cambridge (1987)
21. Brassard, G., Salvail, L.: Secret key reconcilation by public discussion. In: Helleseth, T. (ed.) EUROCRYPT 1993. LNCS, vol. 765, pp. 410–423. Springer, Heidelberg (1994)
22. Mesanes, L., Winter, A.: Unconditional security of key distribution from causality constrains. http://arxiv.quant/ph/06066048 (2006)
23. Ursin, R.: + 17 coauthors: Entanglement based quantum communication over 144 km. Nature 3, 481 (2007)
24. New Jornal of Physics 8, 193 (2006)
25. Schmitt-Manderbach et al.: Experimental demonstration of the free space decoy-state quantum key distribution over 144 km. Phys. rev. lett. 98, 1010504 (2007)
26. http://news.bbc.co.uk/1/hi/sci/tech/7661311.stm
27. Dixon, A.R., Yuan, Z.L., Dynes, J.F., Sharpe, A.E., Shields, A.J.: Optics Express 16(23), 18790–18979
28. Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information. Cambridge University Press, Cambridge (2000)
29. Bennett, C.H., et al.: Teleporting an Unknown Quantum State via Dual Classical and Einstein-Podolsky-Rosen Channels. Phys. Rev. Lett. 70, 1895–1899 (1993)
30. Werner, R.F.: All teleportation and Dense coding schemes, http://arXiv.quant/ph/003070v1
31. Barrett, M.D., et al.: Deterministic Quantum Teleportation of Atomic Qubits. Nature 429, 737 (2004)
32. Riebe, M., et al.: Deterministic Quantum Teleportation with Atoms. Nature 429, 734–737 (2004)
33. Bengtsson, I., Życzkowski, K.: Geometry of Quantum States: An Introduction to Quantum Entanglement. Cambridge University Press, Cambridge (2006)
34. Horodecki, R., Horodecki, P., Horodecki, M., Horodecki, K.: Quantum entanglement. Rev. Mod. Phys (2007), http://arxiv.org/abs/quant-ph/0702225
35. Gurvits, L.: Classical deterministic complexity of Edmonds' Problem and quantum entanglement. In: Proceedings of the thirty-fifth annual ACM symposium on Theory of computing (October 2003)
36. Gruska, J.: Quantum Computing. McGraw-Hill, New York (1999)
37. Gielerak, R., Sawerwain, M.: Twopartite, combinatorial approach to partial k-separability problem for general multipartite states, http://arXiv.quant/ph/1003.0103v1
38. Plenio, M.B., Virmani, S.: An introduction to entanglement measures. Quant. Inf. Comp. 7, 1 (2007)

39. Sawerwain, M., Gielerak, R.: Natural quantum operational semantics with predicates. Int. J. Appl. Math. Comput. Sci. 18(3), 341–359 (2008)
40. Jaeger, G.: Entanglement, Information and the Interpretation of Quantum Mechanics. Springer, Heidelberg (2009)
41. Bouwmeester, D., Pan, J.W., Mattle, K., Eibl, M., Weinfurter, H., Zeilinger, A.: Experimental Quantum Teleportation. Nature 390(6660), 575–579 (1997)
42. Boschi, D., et al.: Experimental Realization of Teleporting an Unknown Pure Quantum State via Dual classical and Einstein-Podolsky-Rosen channels. Phys. Rev. Lett. 80(6), 1121–1125 (1998)
43. Marcikic, I., et al.: Long-Distance Teleportation of Qubits at Telecommunication Wavelengths. Nature 421, 509 (2003)

GPGPU Based Simulations for One and Two Dimensional Quantum Walks

Marek Sawerwain and Roman Gielerak

Institute of Control & Computation Engineering
University of Zielona Góra, ul. Podgórna 50, Zielona Góra 65-246, Poland
{M.Sawerwain,R.Gielerak}@issi.uz.zgora.pl

Abstract. Simulations of standard 1D and 2D quantum walks have been performed within Quantum Computer Simulator (QCS system) environment and with the use of GPGPU (General Purpose Graphics Processor Unit) supported by CUDA (Compute Unified Device Architecture) technology. In particular, simulations of quantum walks may be seen as an appropriate benchmarks for testing computational power of the processors used. It was demonstrated by a series of tests that the use of CUDA based technology radically increases the computational power compared to the standard CPU based computations.

Keywords: one and two dimensional quantum walks, simulation of quantum walks on gpgpu, CUDA technology.

1 Introduction

Recently the concept of quantum walks attracted a big attention as they provide us with a very promising source of ideas for constructing new quantum algorithms [1], [2], [3]. In particular exponential speedups of some classical problems have been discovered again (like in the Shor's algorithm case), the exponentially faster hitting described in [4] and [5] seems to be a good example for this. Additionally, certain although not so particular as exponential speedups are, speedups of some classical problems like k-distinction problem, triangle and k-clique algorithms are examples of them [6],[1],[7] have been obtained by the use of quantum walks concept. Another inspirations for studying quantum walks based applications may come from the observation [8], [9] that quantum walks might play the role of universal quantum calculation tool.

It is the main purpose of the present contribution to test some particular properties of quantum walks by using certain simulating tools. The main tools used for our simulations are the Zielona Góra Quantum Computer Simulator (as the main tool) and the GPGPU equipped with new computational technology offered by CUDA.

It is one of the main result of this note the demonstration of powerful incrementation of calculational abilities if we use GPGPU of new generation compared with standard CPU based computations.

A. Kwiecień, P. Gaj, and P. Stera (Eds.): CN 2010, CCIS 79, pp. 29–38, 2010.

Organisation of this note is as follows: in Sect. 2 we provide a reader with the basic definitions and constructions connected with quantum walks on general graphs. Numerical implementations of quantum walks on certain lattice structures coming from that of $\mathbb{Z}^1$ and $\mathbb{Z}^2$ together with the corresponding numerical algorithms are presented in Sect. 3. Additionally several examples of our simulations will be presented and discussed briefly there.

2 Some Mathematical Preliminaries

Let $G = (\mathbb{V}, \mathbb{E}, l)$ be a graph and where as obviously $\mathbb{V}$ stands for the set of vertices of G the number of elements $|\mathbb{V}|$ of which is equal to N, $\mathbb{E}$ stands for the set of (undirected) edges of G and $l : \mathbb{E} \to \{\mathbb{V}, \mathbb{V}\}$ is the edges labelling function. The corresponding incidence matrix of G will denoted as M_G. With the use of M_G the continuous time Markov walk on G can be described by the corresponding, canonical Markovian transition semi-group $T_t = \exp(-tM_G)$ and (eventually) an initial distribution π_0. For any $v \in \mathbb{V}$ we denote by $d(v)$ the corresponding degree of vertex v and let then $\mathbb{P}(v) = (P_1, P_2, \ldots, P_{d(v)})$ be a discrete probability measure assigning a probability that the walker jump with probability p_i by the use of i-th edge e_i connecting the vertex v with $l(e_i)(2) = \omega$. The system $(\mathbb{P}, \pi_0)$, where π_0 is an initial distribution gives rise to the discrete step random walk on G.

The corresponding quantum walks on G can be constructed in the following way.

The continuous time quantum walk on G: by the very definition starting from the vertex v_0 (with probability π_0) after time t we arrive at the vertex obtained by the action of the unitary group $U_t^G = \exp(itH_G)$ (where H_G stands for the corresponding graph Hamiltonian) acting in the Hilbert space $\mathcal{H}^G = \oplus_{v\in\mathbb{V}}|v\rangle$. However only the discrete time processes will be discussed in this note.

2.1 The Discrete Time Markovian Quantum Walks

Let$\mathcal{H}^G = \oplus_{v\in\mathbb{V}}|v\rangle$ be the canonical Hilbert space associated with G, obviously $\mathcal{H}^G \simeq \mathbb{C}^N$. For any $v \in \mathbb{V}$ let $d(v)$ be degree of v. Then the local Hilbert space $\mathcal{H}_v$ is defined as a space isomorphic to $\mathbb{C}^{d(v)}$, explicitly $\mathcal{H}_v = \oplus_{e_u}|e_u\rangle$, e_u runs over all edges connecting the vertex v with others. A collections $\mathbb{C} = (C_v, v \in \mathbb{V})$ of unitary maps acting on the spaces $\mathcal{H}_v$, $v \in \mathbb{V}$ and fulfilling additionally certain natural coincidence conditions, see i.e. [5]; will be called a "coin flip transformation sequence". In other words, for any $v \in \mathbb{V}$:

$$\begin{array}{c} C_v : \ |v\rangle \otimes \mathcal{H}_v \to |v\rangle \otimes \mathcal{H}_v \\ \text{where } |v\rangle \otimes |\omega\rangle \to |v\rangle \otimes C_v|\omega\rangle \ . \end{array} \tag{1}$$

The global Hilbert space $\mathcal{H}$ is defined as $\mathcal{H} = \oplus_{v\in\mathbb{V}}|v\rangle\otimes\mathcal{H}_v$ and the corresponding discrete quantum walk on G, providing the family $\mathbb{C}$ is given, can be defined by its one step transformation:

$$U = S(I \otimes \mathbb{C}) \tag{2}$$

where the shift transformation S is defined as:

$$S|v,e\rangle = |v',e\rangle \quad \text{if} \quad l(e) = \{v,v'\} \ . \tag{3}$$

Several questions (with analogy to the classical case, especially the problems connected to the mixing and hitting times on a large class of graphs have been studied intensively) can be studied, the question about limiting probability distributions and hitting times are amoung the most popular one [10].

Although intensive simulations of quantum walks on many complex graphs are planned to be done we have concentrated first on some simplest quantum walks on the infinite (the finite amount of with appropriate boundary condition are considered in real simulation tasks of course) graphs $\mathbb{Z}^1$ and $\mathbb{Z}^2$ that we describe now.

2.2 Quantum Walks on Lattice $\mathbb{Z}^1$ and on Lattice $\mathbb{Z}^2$

With the lattice $\mathbb{Z}^1$ we associate the Hilbert space $l_2(\mathbb{Z}) = \oplus_{n\in\mathbb{Z}}|n\rangle$ and the coin flip transformation $\mathbb{C}$ acting in $\mathbb{C}^2 \equiv |R\rangle \oplus |L\rangle$ symbolising the possible steps in the right ($|R\rangle$) or left direction ($|L\rangle$) is given. Then the corresponding Hilbert space $l_2(\mathbb{Z}) \otimes \mathbb{C}^2$ can be seen as a space of infinite sequences $|\psi\rangle \approx ((\alpha_{jk}), j \in \mathbb{Z},\ k \in \{R,L\})$, i.e. any vector $|\psi\rangle \in l_2(\mathbb{Z}) \otimes \mathbb{C}^2$ can be given by:

$$|\psi\rangle = \sum_{j\in\mathbb{Z},k=L,R} \alpha_{jk}|j,k\rangle \quad \text{where} \quad \sum_{j,k} |\alpha_{jk}|^2 = 1 \ . \tag{4}$$

Different choices of $\mathbb{C}$ and shift operators lead to different models of quantum walks on the lattice $\mathbb{Z}^1$.

The graph Hilbert space $\mathcal{H}^{\mathbb{Z}^2}$ for the 2D lattice $\mathbb{Z}^2$ is defined as

$$\mathcal{H}^{\mathbb{H}_2} = \oplus_{j,m\in Z}|j,m\rangle = l_2(\mathbb{Z}^2) \ . \tag{5}$$

The degrees of all vertices are equal to 4 and therefore the local Hilbert spaces are isomorphic with $\mathbb{C}^4$ to be identified with R,L,U,D (right, left, up, down) steps on the lattice. The total space $\mathcal{H} = l_2(\mathbb{Z}^2) \otimes \mathbb{C}^4$ and the typical vector $|\psi\rangle \in \mathcal{H}$ can be decomposed as

$$|\psi\rangle = \sum_{j,k=1}^{4} \sum_{m,n=-\infty}^{+\infty} \alpha_{j,k,m,n}|j,k\rangle|m,n\rangle \quad \text{with} \quad \sum_{j,k=1}^{4} \sum_{m,n=-\infty}^{+\infty} |\alpha_{j,k,m,n}|^2 = 1 \ . \tag{6}$$

A different versions of the corresponding coin flip transformation $\mathbb{C}$ and the shift operators (reflecting some additional topological constraints) then lead to different quantum walk models on $\mathbb{Z}^2$ lattice. Some of them will be presented for simulations performed in the next section including some quantum walk models on $\mathbb{Z}^1$ as well.

3 The Algorithm for Simulating Quantum Walks on GPGPU

The calculation routine for simulation of quantum walks can be build directly from the definition of state of quantum walker walking on the lattice $\mathbb{Z}^2$. In general a state of the quantum walker at time t is given in the following way:

$$|\psi(t)\rangle = \sum_{j,k=0}^{1} \sum_{m,n=-\infty}^{+\infty} \alpha_{j,k,m,n}(t)|j,k\rangle|m,n\rangle \tag{7}$$

where $\alpha_{j,k,m,n}(t) \in \mathbb{C}$ and $\sum_{j,k}\sum_{m,n}|\alpha_{j,k,m,n}(t)|^2 = 1$. The evolution of the quantum walker system over time t is expressed by following unitary operator

$$U = S(C \otimes I) \tag{8}$$

where S is the shift operator, I represents the identity operator and C is the coin operator (in most cases we can assume that the coin is represented by Hadamard operator, but there exist other representations of the coin operator e.g. Fourier and Grover coins) which acts on the local $\mathcal{H}_2 \otimes \mathcal{H}_2$ subspace of whole walker Hilbert space system $l_2(\mathbb{Z}^2) \otimes \mathcal{H}_2 \otimes \mathcal{H}_2$.

In this contribution we propose rather special definition of shift operator for two-dimensional quantum walks. A comparison of our definition with those used frequently can be found in [11], [12] and [13]. We also use the random broken links (termed RBL) technique, first developed for one dimensional quantum walks and introduced in [14]. The RBL technique was generalised for two-dimensional case in [11].

The used definition of shift operator which coincides with physical and mathematical lattice is the following

$$S = \sum_{j,k=0}^{1} \sum_{m,n=-\infty}^{+\infty} |j,k\rangle\langle j,k| \otimes |m+(-1)^j(1-\delta_{j,k}), n+(-1)^j\delta_{j,k}\rangle\langle m,n| \tag{9}$$

where $\delta_{j,k}$ is Dirac discrete delta function.

The following function includes the possibility of appearance of the broken line in a path in between site (j,k) and (m,n) in example depicted on Fig. 1:

$$\mathcal{RBL}(j,k,m,n) = \begin{cases} (-1)^j & \text{– if link to } m+(-1)^j(1-\delta_{j,k}), \\ & \quad n+(-1)^j\delta_{j,k} \text{ is closed} \\ 0 & \text{– if link is open} \end{cases} \tag{10}$$

where $j,k \in \{0,1\}$.

After applying shift operator (9) to state (7) the evolution can be summarised in following way:

$$\psi_{(1-j,1-k,m,n)}(t+1) = \sum_{j',k'=0}^{1} C_{j+\mathcal{RBL}(j,k,m,n),k\oplus\mathcal{RBL}(j,k,m,n),j',k'} \cdot \psi_{(j',k',m+\mathcal{RBL}(j,k,m,n)(1-\delta_{j,k}),n+\mathcal{RBL}(j,k,m,n))\delta_{j,k}}(t) \tag{11}$$

where $\oplus$ represents addition modulo two.

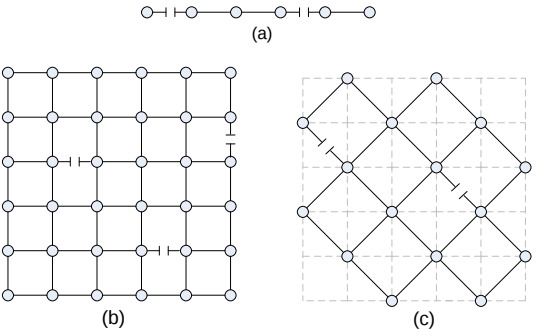


Fig. 1. The idea of broken links for one-dimensional quantum walk (a) where we show broken links between sites, part (b) and part (c) represent examples of natural and diagonal lattices for two-dimensional quantum walk

The algorithm to simulate two-dimensional quantum walks is directly basing on the evolution given by Equation (11). It can be implemented on the traditional architecture for standard CPU and of course for GPGPU based solutions.

Figure 2 shows a fragment of calculation routine for GPGPU which task is to compute values for the next iteration of quantum walk in the segment case. In each iteration all points in the segment attain a new values. This means that the all available GPGPU cores are fully used. Additionally, the efficiency can be increased by better usage of threads available in CUDA architecture.

```
__global__ void one_iteration_segment(
            cuFloatComplex *A0, cuFloatComplex *A1,
            cuFloatComplex *Atemp0, cuFloatComplex *Atemp1, int *RBL0,
            int *RBL1, cuFloatComplex *C, int N) {
    int m = blockIdx.x * blockDim.x + threadIdx.x; int L, cidx1, cidx2;

    if (m<N) {
        L = RBL0[m]; cidx1=(L*2); cidx2=(L*2)+1;
        Atemp1[m] = cuCaddf(cuCaddf(Atemp1[m], cuCmulf(C[cidx1], A0[m+L])),
                            cuCaddf(Atemp1[m], cuCmulf(C[cidx2], A1[m+L])));

        L = RBL1[m]; cidx1=((1+L)*2); cidx2=((1+L)*2)+1;
        Atemp0[m] = cuCaddf(cuCaddf(Atemp0[m], cuCmulf(C[cidx1], A0[m+L])),
                            cuCaddf(Atemp0[m], cuCmulf(C[cidx2], A1[m+L])));
    }
}
```

Fig. 2. The kernel function to compute trajectories of one-dimensional quantum walks on segment

The function to calculate the probability distribution for one-dimensional quantum walk on the line is very similar to the segment case with one important difference. In the i-th iteration the quantum walker cannot be farther than i sites from its initial position. The necessary change in GPGPU routine is expressed as:

```
int m = blockIdx.x * blockDim.x + threadIdx.x; int left, right;
left = max(midpoint - extra - iteration, 1);
right = min(midpoint + extra + iteration, N-1);
if (m>=left && m<=right) { ... }
```

The probability distributions connected to a calculated trajectory for segment and line (and also for two-dimensional cases) are not the same, which is illustrated on the Fig. (3).

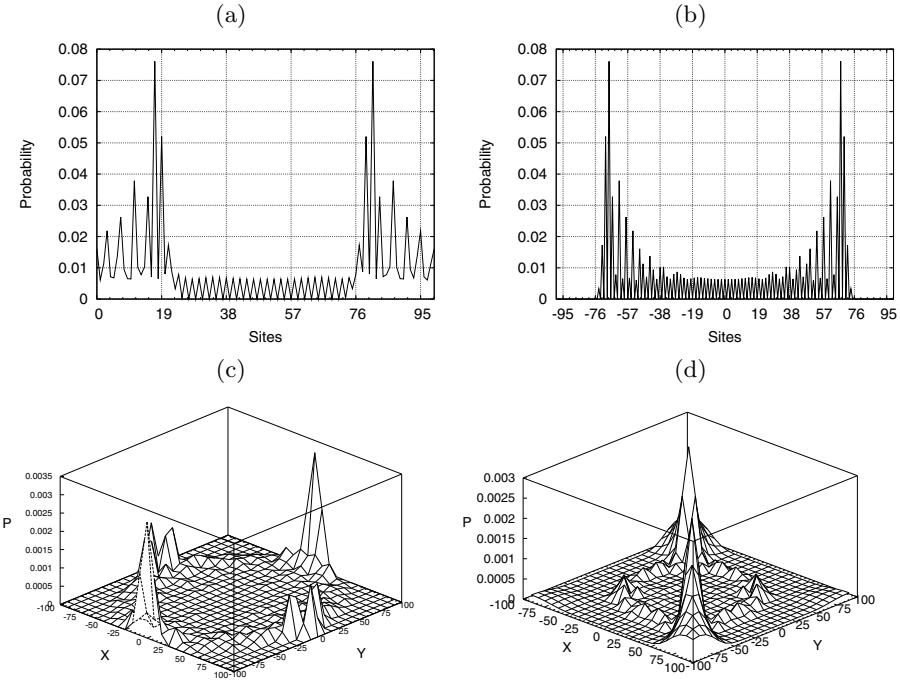


Fig. 3. The one-dimensional quantum walk on segment (a) and on line (b). In both cases number of iterations are the same and equal to 100 and probability distribution of two-dimensional quantum walks with Fourier walker and Fourier's initial state on diagonal lattice depicted on plot (c) and natural lattice depicted on plot (d).

3.1 Complexity Analysis Briefly

The computational complexity T^{1d} of simulation of quantum walk for one-dimensional case strictly depends on the size of segment or line used. Let N be a size of segment for arising in i-th iteration, then:

$$T_i^{1d}(N) = \sum_{m=0}^{N-1} \sum_{j=0}^{1} \left(T_{\mathcal{RBL}_m} \sum_{k=0}^{1} (T_{\mathrm{OP}_m}) \right) = 4 \cdot T_{\mathcal{RBL}_m} \cdot T_{\mathrm{OP}_m} \cdot N \ . \qquad (12)$$

It is possible to write Equation (12) in such a form because operations $\mathcal{RBL}_m$ and OP_m need the same constant time of work which is independent on the number of iteration. The symbol $T_{\mathcal{RBL}_m}$ stands for amount of time necessary to process information about broken links, the second symbol T_{OP_m} stands for amount of time necessary to process the probability amplitude of selected point. The use of GPGPU means, that the processing time can be divided by the number N_c of available cores, because operations $\mathcal{RBL}_m$ and OP_m are independent for each point in segment or line (as well as in two-dimensional case):

$$T^{1d}_{\mathrm{GPGPU}}(n_i) = T^{1d}_i(N)/N_c \ . \tag{13}$$

The simulation of quantum walk on line shows one important difference comparing to segment case. The point is that the new values of points are calculated only in part and the size of simulated portion strictly depends on the number of iteration. The complexity can be denoted as (where $T_{\mathcal{RBL}_m}$ and T_{OP_m} have the same sense as in previous, segment case):

$$T^{1d}_i(N) = \sum_{m=\mathrm{l}_i}^{\mathrm{r}_i} \sum_{j=0}^{1} \left(T_{\mathcal{RBL}_m} \sum_{k=0}^{1} (T_{\mathrm{OP}_m}) \right) = 4 \cdot T_{\mathcal{RBL}_m} \cdot T_{\mathrm{OP}_m} \cdot ((\mathrm{r}_i - \mathrm{l}_i) + 1) \tag{14}$$

where the l_i and r_i are given by:

$$\mathrm{l}_i = \max(\mathrm{MP} - \mathrm{ES} - \mathrm{i}, 1), \quad \mathrm{r}_i = \min(\mathrm{MP} + \mathrm{ES} + \mathrm{i}, N - 1) \tag{15}$$

where MP is the midpoint index of line, the value ES is used to pad and therefore prevents from range errors. However, these values are constant so the difference of r_i and l_i for i-th iteration can be expressed as

$$(r_i - l_i) = (2 \cdot i) + 1 \ . \tag{16}$$

In the case of small systems (ten, twenty or fifty points), Equation (16) shows that the most of available cores in GPGPU are not fully utilised. In the case of bigger systems this problem is not arising. The difference in (16) for higher iteration number is bigger and what is more the values of this difference for all iterations form the arithmetic sequence, which means that in sense of complexity theory that only linear speedup is achieved, however for segment with size of 10000 points the obtained speedup is nearly hundredfold. The corresponding times have been depicted in Tables 1, 2 and 3.

The use of Equation (11) allows to estimate computational complexity of two-dimensional quantum walk trajectory calculations in i-th iteration:

$$\begin{aligned} T^{2d}_i(N) = \sum_{m=\mathrm{lb}_i}^{\mathrm{rb}_i} \sum_{n=\mathrm{lb}_i}^{\mathrm{rb}_i} \sum_{j=0}^{1} \sum_{k=0}^{1} T_{\mathcal{RBL}_{(m,n,j,k)}} 4 \cdot T_{\mathrm{OP}_{(m,n,j,k)}} = \\ > 16 \cdot T_{\mathcal{RBL}_{(m,n,j,k)}} \cdot T_{\mathrm{OP}_{(m,n,j,k)}} \cdot ((\mathrm{rb}_i - \mathrm{lb}_i) + 1)^2 \ , \end{aligned} \tag{17}$$

where N means the length of trajectory calculated.

Table 1. The measured times of calculations of one-dimensional quantum walks trajectories for segments with different sizes (without broken links)

	Core 2 Duo 8400 (1 core)	Geforce 9600 GT (64 cores)	Geforce 280 (240 cores)
Size	Time in ms	Time in ms	Time in ms
100	6.00	4.51	2.48
1000	663.00	40.186	24.171
5000	22685.00	612.67	281.57
10000	96362.00	2324.3002	884.297

Table 2. The measured times of calculations of two-dimensional quantum walks trajectories for diagonal lattice without broken links. The measured times for the case of Core 2 Duo and two-thread computational routine are presented in bracket.

	Core 2 Duo 8400 (1 core)	Geforce 9600 GT (64 cores)	Geforce 280 (240 cores)
Size	Time in ms	Time in ms	Time in ms
100	700 (413)	195	64
200	5480 (2945)	1003	363
300	19140 (10512)	3423	1137
400	47230 (25785)	8123	3127
500	92530 (50274)	14462	4706

Table 3. The measured times of calculations of two-dimensional quantum walks trajectories for diagonal lattice without broken links. The simulation was performed on two Intel Xeon E5420 2.50 Ghz processors, the multi-threaded calculation subroutine was compiled with GCC compiler with "-O3" option.

	(1-thread)	(2th)	(4th)	(8th)
Size	Time in ms	Time in ms	Time in ms	Time in ms
100	770	409	299	206
200	6740	3281	2190	1382
300	22910	12029	7130	4848
400	56610	29853	17699	12069
500	114060	65292	33275	24602

The variables lb_i and rb_i have the same meaning as l_i, r_i introduced before and are calculated in the following way:

$$\mathrm{lb_i} = \max(\mathrm{MP} - \mathrm{ES} - \mathrm{i}, 1), \quad \mathrm{rb_i} = \min(\mathrm{MP} + \mathrm{ES} + \mathrm{i}, 2 \cdot \mathrm{MP} - 1) \ . \tag{18}$$

The expression $((\mathrm{rb}_i - \mathrm{lb}_i)+1)^2$ in Equation (17) can be expressed in the following way

$$((\mathrm{rb}_i - \mathrm{lb}_i) + 1)^2 = ((2 \cdot i) + 1)^2 \ . \tag{19}$$

Speedups obtained for 2D quantum walks presented in Fig. 4 are based on results presented in Tables 2 and 3.

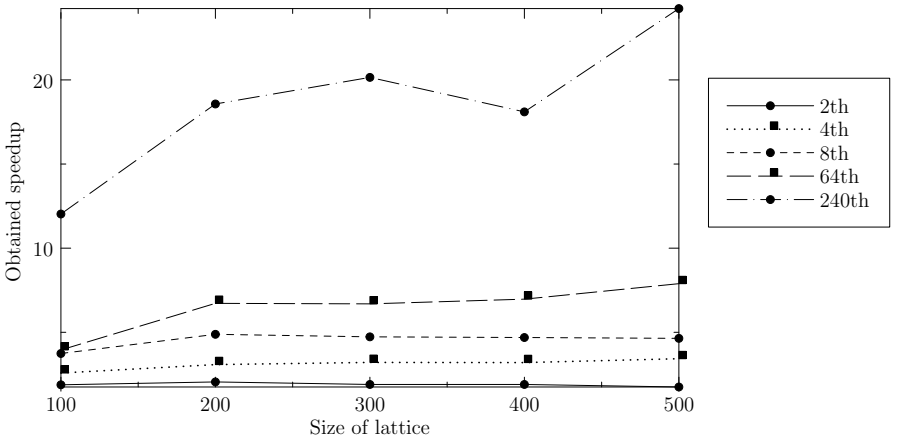


Fig. 4. The obtained values of speedup for simulations of two dimensional quantum walks on diagonal lattice. Graphs labelled as 2th, 4th and 8th are compared to one-thread computational routine. Graphs marked as 64th and 240th represent the speedup obtained by computational routine executed on the Geforce video card 9600 GT and GTX 280 respectively.

4 Conclusions and Further Work

The specialised software to simulate one and two dimensional random quantum walks without and with broken links has been presented in this article. The used software is a part of the Quantum Computing Simulator presented in [15], [16].

A significant speedup of the simulations process comparing to previous paper [13] have been achieved. The used technologies enables to simulate effectively much more complex quantum walks then previously known. Additionally, certain more deeper notions connected to the analysis of quantum walks behaviour can be analysed using computer simulations as an appropriate tool.

Acknowledgments. We acknowledge useful discussions on the QCS with the *Q-INFO* group at the Institute of Control and Computation Engineering of the University of Zielona Góra, Poland.

References

1. Szegedy, M.: Quantum Speedup of Markov Chain Based Algorithms. In: Proc. of 45th Annual IEEE Symposium on Foundations of Computer Science, pp. 32–41 (2004)
2. Aharonov, D., Ambainis, A., Kempe, J., Vazirani, U.: Quantum walks on graphs. In: Proceedings of 33th STOC, pp. 50–59. ACM, New York (2001)
3. Nayak, A., Vishwanath, A.: Quantum Walk on the Line, http://arxiv.org/abs/quant-ph/0010117
4. Childs, A.M., Cleve, R., Deotto, E., Farhi, E., Gutmann, S., Spielman, D.A.: Exponential algorithmic speedup by quantum walk. In: Proc. 35th ACM Symposium on Theory of Computing, pp. 59–68 (2003)

5. Kempe, J.: Quantum random walks hit exponentially faster. In: Proceedings of 7th Intern. Workshop on Randomization and Approximation Techniques in Computer Science. LNCS, pp. 354–369. Springer, Heidelberg (2003)
6. Ambainis, A.: Quantum walks and their algorithmic applications. International Journal of Quantum Information 1(4), 507–518 (2003)
7. Wocjan, P., Abeyesinghe, A.: Speedup via quantum sampling. Phys. Rev. A 78, 042336 (2008)
8. Oliveira, A., Portugal, R., Donangelo, R.: Simulation of the single- and double-slit experiments with quantum walkers (2007), http://arxiv.org/abs/0706.3181
9. Childs, A.M.: Universal computation by quantum walk. Phys. Rev. Lett. 102, 180501 (2009)
10. Kempe, J.: Quantum random walk algorithms. Contemp. Phys. 44(3), 302–327 (2003)
11. Oliveira, A.C., Portugal, R., Donangelo, R.: Decoherence in two-dimensional quantum walks. Phys. Rev. A 74, 012312
12. Kendon, V.: Decoherence in quantum walks: a review. Math. Struct. in Comp. Sci. 17(6), 1169–1220 (2006)
13. Marquezino, F.L., Portugal, R.: The QWalk Simulator of Quantum Walks. Computer Physics Communications 179(5), 359–369 (2008)
14. Romanelli, A., Siri, R., Abal, G., Auyuanet, A., Donangelo, R.: Decoherence in the quantum walk on the line. Physica A 347C, 137–152 (2005)
15. Sawerwain, M.: Parallel algorithm for simulation of circuit and one-way quantum computation models. In: Wyrzykowski, R., Dongarra, J., Karczewski, K., Wasniewski, J., et al. (eds.) PPAM 2007. LNCS, vol. 4967, pp. 530–539. Springer, Heidelberg (2008)
16. Sawerwain, M., Gielerak, R.: Natural quantum operational semantics with predicates. Int. J. Appl. Math. Comput. Sci. 18(3), 341–359 (2008)

Negative Difference Resistance and Its Application to Construct Boolean Logic Circuits

Maciej Nikodem, Marek A. Bawiec, and Tomasz R. Surmacz

Institute of Computers, Control and Robotics, Wroclaw University of Technology
ul. Wybrzeże Wyspiańskiego 27, 50-370 Wrocław

Abstract. Electronic circuits based on nanodevices and quantum effect are the future of logic circuits design. Today's technology allows constructing resonant tunneling diodes, quantum cellular automata and nanowires/nanoribbons that are the elementary components of threshold gates. However, synthesizing a threshold circuit for an arbitrary logic function is still a challenging task where no efficient algorithms exist. This paper focuses on Generalised Threshold Gates (GTG), giving the overview of threshold circuit synthesis methods and presenting an algorithm that considerably simplifies the task in case of GTG circuits.

Keywords: Generalised Threshold Gate, NDR, gate synthesis.

1 Introduction

CMOS is a very powerful technology of constructing complex circuits implementing various Boolean logic functions. One of the most important features of CMOS is the ability to construct switching circuits that theoretically do not consume power when in steady state. However, constant need to develop faster and more complex circuits leads to increased integration scale and brings new issues, such as small size of single electronic devices (close to the technology limits), increasing current leakages, or difficulties with clock synchronization. These parasitic effects lead to increased power consumption, thermal runaways and other undesirable effects, showing that technology is already reaching the limits and that new solutions are required.

International Technology Roadmap for Semiconductors [1] proposes several alternatives for CMOS technology, focusing on improved technology of CMOS circuit design (e.g. 193 nm wavelength lithography) and new technologies (e.g. nanowires, carbon nanotubes or graphene devices) that take advantage of quantum effects. One of the alternatives are the devices that feature Negative Differential Resistance (NDR) property [2].

Circuits with NDR property were first studied in 1960s and 1970s when several CMOS based circuits have been proposed [3]. CMOS based NDR devices are still under development and up to date CMOS technologies are used to improve properties of the devices [4,5]. These works focus on constructing NDR devices using currently available technologies while it is believed that the future belongs to resonant tunneling diodes (RTDs), nanowires, nanoribbons or carbon nanotubes (CNT).

A. Kwiecień, P. Gaj, and P. Stera (Eds.): CN 2010, CCIS 79, pp. 39–48, 2010.

2 Previous Work

Advances in nanotechnology and NDR devices in the last 15 years have restored interest in NDR based circuits for implementing Boolean functions. Attention is focused on threshold gates, since the proposed circuits structures (namely MTTG and GTG) implement a threshold function.

Threshold circuits were extensively analysed in 1950s and 1960s and some methods on circuit synthesis have been presented at that time. The power of threshold gates and threshold logic arises from inherent ease of implementation of complex logic functions, when compared to standard Boolean logic gates. For example, important functions – such as multiplication, multiple addition or division – can be implemented by a polynomial size threshold circuit of small depth [6]. Despite their advantages, threshold circuits did not receive much attention in 60's, mainly due to the difficulties with physical realization of threshold gates and circuits at that time. Synthesis was also focused on one-to-one mapping of each Boolean gate to a threshold one. Today, various algorithms exist which improve threshold synthesis [7,8], thanks to algebraic and Boolean factorization and Boolean network simplification, but threshold circuit synthesis is still a challenging task. This is mainly due to heuristics used and relatively complex procedures to collapse Boolean functions and split non-threshold functions into appropriate sets of threshold functions.

Zhand et al. [8] proposed a recursive algorithm that takes algebraically-factored multi-output combinatorial network and outputs functionally equivalent threshold network. Algebraically-factored form is used, since it is more likely that its nodes are unate, thus simplifying the whole synthesis. The algorithm itself operates on each output of the network, utilizing integer linear programming, node splitting and combining. The proposed algorithm succeeds to minimize the number of gates and levels as well as the area of the threshold circuit, when compared to simple one-to-one mapping with fan-in set to 3.

Multivalued Decision Diagrams (MDDs) are used in the synthesis algorithm proposed by Avedillo et al. [7]. These use a partitioning algorithm that, according to authors, can be optimized for targeted threshold network implementations thus improving the synthesis. Our work focuses on constructing threshold gates that implement complex Boolean functions rather then on algorithms for building networks of threshold gates.

Recent works on threshold logic synthesis have focused on general purpose algorithms while no interest was in nanoelectronic devices that enable constructing threshold gates and circuits. The rest of this paper focuses on the structure of GTG circuits and presents a dedicated synthesis algorithm for implementing complex Boolean logic functions in a single threshold gate.

3 Generalised Threshold Gates

Avedillo et al. [10,11] proposed Multi Threshold Threshold Gates (MTTGs) and Generalised Threshold Gates (GTGs) that are current-driven devices utilizing

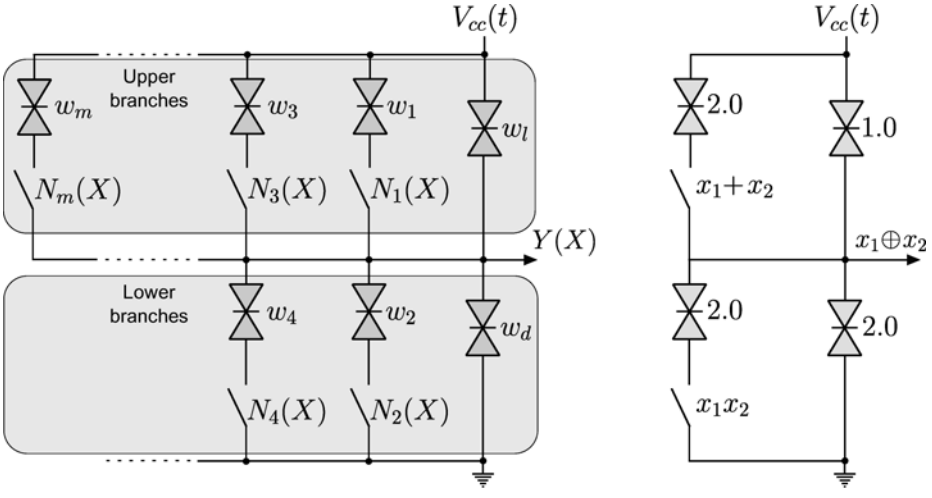


Fig. 1. General structure of GTG (left) and a GTG circuit implementing EXOR sum of two inputs (right). Two rightmost NDRs are called the *load* and the *driver* respectively.

nanodevices with NDR property (e.g. Resonant Tunneling Diodes – RTDs). The GTG is composed of parallel branches, each of them consisting of a single NDR element serially connected to a serial-parallel (SP) transistor network (see Fig. 1). SP transistor network implements unate switching functions ($N_i(\mathbb{X})$, where $\mathbb{X}$ denotes vector of n input signals x_i) that control currents in upper and lower branches of the circuit. Depending on whether the sum of maximum currents in upper branches is larger or smaller then in lower branches, the circuit switches to one of the bistable states that correspond to logic zero and one (see Fig. 2). As a result, the GTG circuit implements function

$$Y(\mathbb{X}) = \begin{cases} 1 \text{ iff } \sum_i^m w_i N_i(\mathbb{X}) \geq w_\mathrm{d} - w_\mathrm{l} \\ 0 \text{ otherwise} \end{cases}, \tag{1}$$

where weights w_i depend on the parameters of NDR devices and are positive for devices located in the upper level and negative for NDRs in the lower level. The difference $w_\mathrm{d} - w_\mathrm{l}$ is a threshold value that depends on parameters of two NDR devices – the load and the driver. Function implemented by a GTG circuit can be represented in a recursive way [12]:

$$Y_i(\mathbb{X}) = \begin{cases} Y_{i-1}(\mathbb{X}) + N_i(\mathbb{X}), & \text{if } i \text{ is odd} \\ Y_{i-1}(\mathbb{X})\overline{N_i(\mathbb{X})}, & \text{if } i \text{ is even} \end{cases}, \tag{2}$$

with $Y_0(\mathbb{X}) = 0$. This model assumes that switching functions with odd/even indices control the upper/lower branches of the GTG accordingly. Even though this model is still impractical for synthesis purposes, its application has made possible enumeration of all possible functions that can be implemented in four branches GTG circuit [12], with a restriction of fan-in of each switching function set to 4

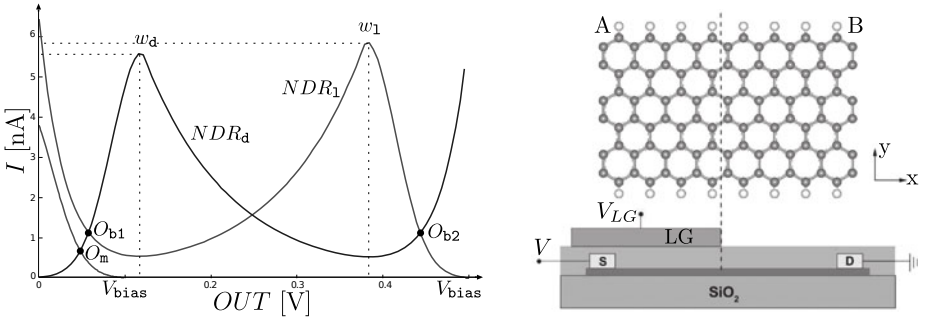


Fig. 2. Graphical solution for the simplest NDR-based circuit, presenting monostable (O_m) and bistable states (O_{b1}, O_{b2}) that correspond to logic 0 and 1. The exact state depends on the relation of peak currents w_d and w_l (left). Atomic structure of the zigzag Graphene Nanoribbon and side view of device proposed by [9] that may serve as a basis for NDR devices (right).

inputs. Exhaustive verification of all possible switching functions has shown that a 4-branch GTG circuit can implement all Boolean functions of up to 4 variables. When analyzing this exhaustive verification it becomes obvious that:

- a Boolean function can be implemented in multiple ways with several different sets of switching functions,
- there is a particular order of switching functions in which they must appear in the GTG model.

The first observation may be verified very easily: For any two successive functions $N_i(X)$ and $N_{i+1}(X)$ the resulting function $Y_{i+1}(X)$ will not change if the same term $F(X)$, such that $F(X) \cdot (Y_{i-1}(X) + N_i(X)) = 0$ is added to both of them. For example, if i is odd, then from (2) we first get:

$$\begin{aligned} Y_{i+1}(X) &= Y_i(\mathbb{X})\overline{N_{i+1}(\mathbb{X})} \\ &= (Y_{i-1}(\mathbb{X}) + N_i(\mathbb{X}))\,\overline{N_{i+1}(\mathbb{X})} \end{aligned} \tag{3}$$

then we still get the same result if we add $F(X)$ both to N_i and N_{i+1}:

$$\begin{aligned} Y_{i+1}(X) &= (Y_{i-1}(\mathbb{X}) + N_i(\mathbb{X}) + F(\mathbb{X}))\,\overline{N_{i+1}(\mathbb{X}) + F(\mathbb{X})} \\ &= (Y_{i-1}(\mathbb{X}) + N_i(\mathbb{X}) + F(\mathbb{X}))\,\overline{N_{i+1}(\mathbb{X})} \cdot \overline{F(\mathbb{X})} \\ &= (Y_{i-1}(\mathbb{X}) + N_i(\mathbb{X}))\,\overline{N_{i+1}(\mathbb{X})} \cdot \overline{F(\mathbb{X})} + \underbrace{F(\mathbb{X}) \cdot \overline{N_{i+1}(\mathbb{X})} \cdot \overline{F(\mathbb{X})}}_{=0} \\ &= (Y_{i-1}(\mathbb{X}) + N_i(\mathbb{X}))\,\overline{N_{i+1}(\mathbb{X})} \end{aligned} \tag{4}$$

Similar equation holds if i is even. Second observation comes from the fact that all switching functions $N_i(\mathbb{X})$ share the same set of input signals. It follows that they output 1 in a predefined order determined by the actual functions – e.g.

$N_i(\mathbb{X}) = x_1x_2$ outputs 1 only when $N_j(\mathbb{X}) = x_1$ outputs 1. This means that the order of switching functions is important and cannot be set freely.

Based on these observations we take the following assumption:

Assumption 1. *In a GTG circuit successive switching functions $N_j(\mathbb{X})$ are absorbed by any previous functions $N_i(\mathbb{X})$:*

$$N_i(\mathbb{X}) + N_j(\mathbb{X}) = N_i(\mathbb{X}), \text{ where } i < j \ . \tag{5}$$

The following corollary follows immediately:

Corollary 1. *For any three functions $N_i(\mathbb{X})$, $N_j(\mathbb{X})$, $N_k(\mathbb{X})$, where $i < j < k$, the following holds:*

$$N_i(\mathbb{X})\overline{N_j(\mathbb{X})} = N_i(\mathbb{X}) \oplus N_j(\mathbb{X}) \ , \tag{6}$$

and

$$N_i(\mathbb{X}) \oplus N_j(\mathbb{X}) + N_k(\mathbb{X}) = N_i(\mathbb{X}) \oplus N_j(\mathbb{X}) \oplus N_k(\mathbb{X}) \ . \tag{7}$$

Both equations can be easily verified, since it follows from the assumption that $N_i(\mathbb{X})N_{i+1}(\mathbb{X}) = N_{i+1}(\mathbb{X})$ and $\overline{N_i(\mathbb{X})}N_{i+1}(\mathbb{X}) = 0$, which gives

$$N_i(\mathbb{X})\overline{N_j(\mathbb{X})} + \overline{N_i(\mathbb{X})}N_j(\mathbb{X}) = N_i(\mathbb{X}) \oplus N_j(\mathbb{X}) \ . \tag{8}$$

Concerning equation (7) we have:

$$\begin{aligned} N_i(\mathbb{X}) \oplus N_j(\mathbb{X}) \oplus N_k(\mathbb{X}) = \\ &= \overline{N_i(\mathbb{X}) \oplus N_j(\mathbb{X})}N_k(\mathbb{X}) + (N_i(\mathbb{X}) \oplus N_j(\mathbb{X}))\,\overline{N_k(\mathbb{X})} \\ &= \overline{N_i(\mathbb{X})\overline{N_j(\mathbb{X})}}N_k(\mathbb{X}) + N_i(\mathbb{X}) \cdot \overline{N_j(\mathbb{X})} \cdot \overline{N_k(\mathbb{X})} \\ &= N_k(\mathbb{X}) + N_i(\mathbb{X})\overline{N_j(\mathbb{X}) + N_k(\mathbb{X})} \\ &= N_k(\mathbb{X}) + N_i(\mathbb{X})\overline{N_j(\mathbb{X})} = N_k(\mathbb{X}) + N_i(\mathbb{X}) \oplus N_j(\mathbb{X}) \ . \end{aligned} \tag{9}$$

Equations (6) and (7) allow us to simplify model (2) into a simple EXOR sum of switching functions:

$$Y(\mathbb{X}) = \bigoplus_{i}^{m} N_i(\mathbb{X}) \ , \tag{10}$$

which is much more suitable for GTG synthesis.

Model (10) also allows a formal analysis of the GTG circuit to be performed.

Theorem 2. *Any Boolean function $Y(\mathbb{X})$ can be represented as an EXOR sum of unate functions $N_i(\mathbb{X})$ satisfying assumption (5).*

With no loss of generality, let us assume that $Y(0^n) = 0$ – i.e. the function outputs 0 for all zero input. It is obvious that for any Boolean function $Y(\mathbb{X})$, the smallest unate function $N_1(\mathbb{X})$ such that $N_1(\mathbb{X}) + Y(\mathbb{X}) = N_1(\mathbb{X})$ can be found (at least $N_1(\mathbb{X}) = 1$). If $Y_1(X) = N_1(\mathbb{X}) \oplus Y(X)$ is unate then we are done.

Otherwise, there exists the smallest unate function $N_2(\mathbb{X})$ such that $N_2(\mathbb{X}) + Y_1(\mathbb{X}) = N_2(\mathbb{X})$ and $Y_2(\mathbb{X}) = N_2(\mathbb{X}) \oplus Y_1(\mathbb{X})$. By repeating this process for $Y_2(\mathbb{X})$ and so on we finally get $Y_m(\mathbb{X})$ that is unate. This procedure converges in at most n steps, since successive $N_i(\mathbb{X})$ functions get smaller and satisfy (5). If $Y(0^n) = 1$ then we simply change the relation between weights w_l and w_d and synthesize function $Y(\mathbb{X}) \oplus 1$.

Since each iteration outputs one switching function and there is at most n iterations of the above procedure, thus the corresponding GTG circuit, composed of at most $n+2$ branches, can implement any Boolean function $Y(\mathbb{X})$. The next section presents how to find $N_i(\mathbb{X})$ for a given Boolean function.

3.1 GTG Synthesis

We have proposed two synthesis algorithms that output switching functions for the GTG circuit when given an arbitrary Boolean function as input. The first algorithm takes a function in Reed-Muller form and groups some of its elements to construct the unate switching functions for each branch of the circuit. The drawback of this algorithm is the number of branches in the resulting circuit which may be much larger then $n+2$. The actual number of branches depends on the order in which terms of Reed-Muller representation are grouped and transformed (Fig. 3). Since there is no way to determine proper and improper grouping during the algorithm execution, one has to generate circuits for all possible groups and choose the best solution. Other way is to find one solution and try to simplify the circuit through removing surplus branches. This, however, is troublesome.

The second algorithm (Algorithm 3.1) takes a Boolean function represented in a sum of products form and recursively finds the smallest unate functions that satisfy assumption (5). Precisely, the relation between weights w_l and w_d is determined first, depending on whether $Y(0)$ equals to 0 or 1 for all zero input. Next, the algorithm finds $N_1(\mathbb{X})$ that is the smallest unate function, such

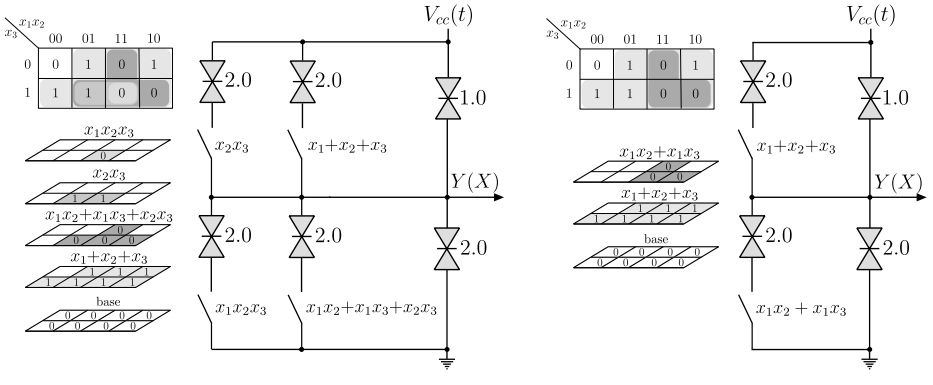


Fig. 3. Two GTG circuits implementing function $Y(\mathbb{X}) = x_1 \oplus x_2 \oplus x_2 \oplus x_2x_3$ using different number of switching functions

that $N_1(\mathbb{X}) + Y(\mathbb{X}) = N_1(\mathbb{X})$. As presented earlier, such function always exists and can be found easily by removing all complemented variables from the SOP representation of $Y(\mathbb{X})$. Then $Y_1(\mathbb{X})$ is calculated and the next iteration begins (i.e. searching for successive $N_i(\mathbb{X})$ and calculating the corresponding $Y_i(\mathbb{X})$). The algorithm terminates when $Y_i(\mathbb{X}) = 0$, after n iterations at most.

Algorithm 1. AND/OR/NOT based GTG circuit synthesis

Require: n–variable Boolean function $Y_0(\mathbb{X})$
Ensure: w_l vs. w_d relation, and set of $N_i(\mathbb{X})$ functions
1: **if** $Y_0(0) = 0$ **then**
2: $w_\mathrm{l} > w_\mathrm{d}$,
3: **else**
4: $w_\mathrm{l} < w_\mathrm{d}$,
5: $Y_0(\mathbb{X}) = 1 \oplus Y_0(\mathbb{X})$,
6: **end if**
7: set $i = 0$,
8: **while** $Y_i(\mathbb{X}) \neq 0$ **do**
9: find the smallest unate function $N_{i+1}(\mathbb{X})$ such that $N_{i+1}(\mathbb{X}) + Y_i(\mathbb{X}) = N_{i+1}(\mathbb{X})$,
10: calculate $Y_{i+1}(\mathbb{X})$ such that $Y_{i+1}(\mathbb{X}) = N_{i+1}(\mathbb{X}) \oplus Y_i(\mathbb{X})$,
11: set $i = i + 1$,
12: **end while**

The algorithm based on AND/OR/NOT representation of Boolean function has several advantages over the previous one. The most important one is the number of switching functions which is less then or equal to n and thus, the synthesized circuit consists of $n + 2$ branches at most. There is also no need to transform the Boolean function into the Reed-Muller form, as the procedure for finding the smallest unate functions is simple and effective.

We have verified correctness and efficiency of our algorithm for a number of Boolean functions. Figure 4 presents a circuit implementing function $Y(\mathbb{X}^3) = x_1x_2 \oplus x_3$ and simulation results obtained with PSpice software.

3.2 Algorithm Improvements

The synthesis algorithm outputs at most n switching functions for GTG circuit, satisfying assumption (5). This assumption was based on the analysis of circuit operation and allowed us to simplify its mathematical model and to carry out formal analysis. However, this also restricts the number of possible circuits implementing a given Boolean function, as only one set of possible unate switching functions satisfies the assumption (5).

Let us consider function $Y(\mathbb{X}) = x_1\overline{x_2} + x_2x_3$. When synthesized, the GTG circuit consists of three branches with the following switching functions: $N_1(\mathbb{X}) = x_1 + x_2x_3$, $N_2(\mathbb{X}) = x_1x_2$, and $N_3(\mathbb{X}) = x_1x_2x_3$ (Fig. 5). However, if we get rid of assumption (5) and use the previous GTG model (2), then switching functions

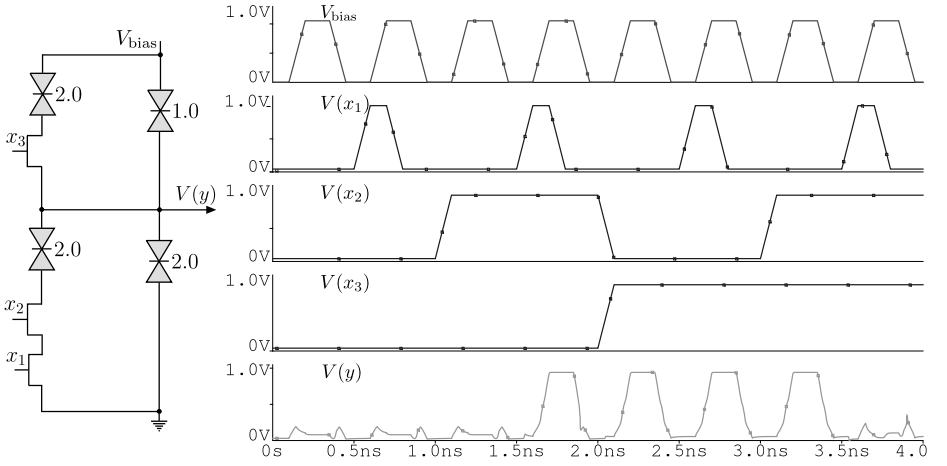
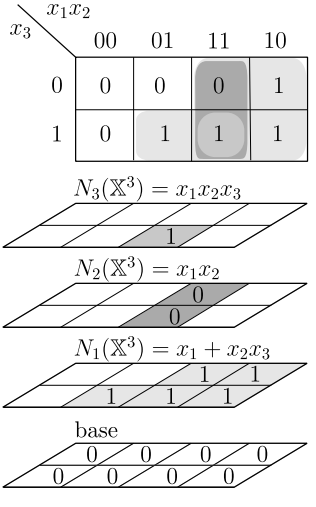


Fig. 4. Circuit structure implementing function $Y(\mathbb{X}) = x_3 \oplus x_1 x_2$ (left) and simulation results obtained with PSpice (right)

$$Y(X) = \bigoplus_i N_i(X) = (x_1 + x_2 x_3) \oplus (x_1 x_2) \oplus x_1 x_2 x_3$$

$$Y(X) = Y_3(X) = x_1\overline{x_1 x_2} + x_2 x_3, \texttt{ where:}$$

$$Y_i(X) = \begin{cases} Y_{i-1}(X) + N_i'(X) & \texttt{for odd } i \\ Y_{i-1}(X)\overline{N_i'(X)} & \texttt{for even } i \end{cases}$$



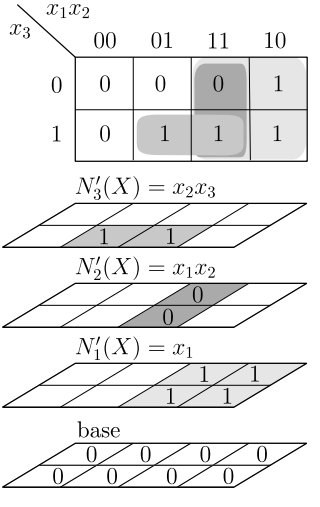


Fig. 5. Example of two circuits implementing the same Boolean function using the same number of branches but different number of transistors – 8 (left) vs. 5 (right)

may be substituted with: $N_1'(\mathbb{X}) = x_1$, $N_2'(\mathbb{X}) = x_1x_2$, and $N_3'(\mathbb{X}) = x_2x_3$. Even though function $N_i'(\mathbb{X})$ does not satisfy the absorption rule, it can be easily verified, that the corresponding GTG circuit correctly implements the given function

$$N_1'(\mathbb{X})\overline{N_2'(\mathbb{X})} + N_3'(\mathbb{X}) = x_1\overline{x_1x_2} + x_2x_3 = \overline{x_1}x_2 + x_2x_3 \ . \tag{11}$$

The difference between these two sets of functions is the number of transistors required to implement them. The previous one requires 8 transistors, while the latter one only 5. Reducing the number of transistors has a direct impact on circuit size and its properties, so finding effective methods to simplify the switching functions is very important.

The procedure for simplifying function $N_i(\mathbb{X})$ requires checking whether there are some minterms of $N_{i-2}(\mathbb{X}) \oplus N_{i-1}(\mathbb{X})$ that may be added to $N_i(\mathbb{X})$ while still keeping it unate. Function $N_i(\mathbb{X})$ is then extended with such minterms if the resulting function $N_i'(\mathbb{X})$ requires less transistors for its implementation. It also follows that the first two switching functions are unchanged, thus $N_1'(\mathbb{X}) = N_1(\mathbb{X})$ and $N_2'(\mathbb{X}) = N_2(\mathbb{X})$. Figure 5 presents the above example where either minterm $M_1(\mathbb{X}) = \overline{x_1}x_2x_3$ or $M_2(\mathbb{X}) = x_1\overline{x_2}x_3$ can be added to $N_3(\mathbb{X})$. Both of them allow expansion of the switching function $N_3(\mathbb{X})$ and, as a result, reducing the number of transistors required for its implementation. It is advisable however to use $M_1(\mathbb{X})$, since in that case $N_3'(\mathbb{X}) = x_2x_3$ has a prime implicant common with $N_1(\mathbb{X})$. This is advantageous, since if the same prime implicant exists in $N_{i-2}'(\mathbb{X})$ and $N_i'(\mathbb{X})$, and does not exist in $N_{i-1}'(\mathbb{X})$, then it may be removed from $N_{i-2}'(\mathbb{X})$. For our example this condition is satisfied for prime implicant x_2x_3, so it can be removed from $N_1'(\mathbb{X})$ with no change to the resulting function $Y(\mathbb{X})$. Removing common implicants leads to further reduction of the overall number of transistors required to implement the given Boolean function.

The above example shows that circuits synthesized with the proposed algorithm can be optimized further. Procedure for reducing the number of transistors requires exhaustive verification of all possibilities. This is inefficient but acceptable for functions with only few input variables.

4 Conclusions and Future Work

Todays technology is still too immature to produce NDR devices as an off-the-shelf technology. However, several researches are under way and it is believed that various CNT and graphene devices will reach qualification/pre-production state around year 2015. Therefore, theoretical work on circuit synthesis have to run in parallel to the technology research. The presented algorithm provides a circuit synthesis method for an arbitrary boolean function by decomposing a complex non-threshold boolean function into a set of threshold functions to be used as switching functions in a single gate. With some restrictions, this may be extended to provide a generalized algorithm for synthesis of threshold gate networks. Removing these restrictions is a work in progress. The algorithm presented in this paper is dedicated to Generalised Threshold Gates, focusing

on providing simple and effective way of synthesising the given functionality. As mentioned in previous sections, several improvements are still possible that will be addressed in future work.

Acknowledgements. This work was partially supported by grant no. N N516 451538 from the Polish Ministry of Science and Higher Education.

References

1. ITRS: Process integration, devices and structures. In: Report on international technology roadmap for semiconductors, 2009 edition. International Technology Roadmap for Semiconductors (2009)
2. ITRS: Emerging research devices. In: Report on international technology roadmap for semiconductors, 2009 edition. International Technology Roadmap for Semiconductors (2009)
3. Wu, C.Y., Lai, K.N.: Integrated Λ-type differential negative resistance MOSFET device. IEEE Journal of Solid-State Circuits 14(6), 1094–1101 (1979)
4. Guo, W.L., Wang, W., Niu, P.J., Li, X., Yu, X., Mao, L., Liu, H., Yang, G., Song, R.: CMOS-NDR transistor. In: 9th International Conference on Solid-State and Integrated-Circuit Technology, ICSICT 2008, October 2008, pp. 92–95 (2008)
5. Kwang-Jow, G., Cher-Shiung, T., Dong-Shong, L.: Design and characterization of the negative differential resistance circuits using the CMOS and BiCMOS process. Analog Integrated Circuits and Signal Processing 62(1), 63–68 (2010)
6. Siu, K.Y., Roychowdhury, V.P., Kailath, T.: Depth-Size Tradeoffs for Neural Computation. IEEE Trans. Comput. 40(12), 1402–1412 (1991)
7. Avedillo, M.J., Quintana, J.M.: A Threshold Logic Synthesis Tool for RTD Circuits. In: DSD 2004: Proceedings of the Digital System Design, EUROMICRO Systems, pp. 624–627 (2004)
8. Zhang, R., Gupta, P., Zhong, L., Jha, N.K.: Synthesis and Optimization of Threshold Logic Networks with Application to Nanotechnologies. In: DATE 2004: Proceedings of the conference on Design, automation and test in Europe, p. 20904. IEEE Computer Society, Washington (2004)
9. Wang, Z.F., Zheng, H., Shi, Q.W., Chen, J.: Emerging nanodevice paradigm: Graphene-based electronics for nanoscale computing. J. Emerg. Technol. Comput. Syst. 5(1), 1–19 (2009)
10. Avedillo, M.J., Quintana, J.M., Pettenghi, H.: Logic Models Supporting the Design of MOBILE-based RTD Circuits. In: ASAP 2005: Proceedings of the 2005 IEEE International Conference on Application-Specific Systems, Architecture Processors, pp. 254–259. IEEE Computer Society Press, Washington (2005)
11. Pettenghi, H., Avedillo, M.J., Quintana, J.M.: Using multi-threshold threshold gates in RTD-based logic design: A case study. Microelectron J. 39(2), 241–247 (2008)
12. Berezowski, K.S.: Compact binary logic circuits design using negative differential resistance devices. IET Electr. Lett. 42(16), 902–903 (2006)

Improved Estimation of Success Probability of the Shor's Algorithm

Piotr Zawadzki

Institute of Electronics,
Silesian University of Technology,
Akademicka 16, 44-100 Gliwice, Poland
Piotr.Zawadzki@polsl.pl

Abstract. The quantum factorization is probably the most famous algorithm in quantum computation. The algorithm succeeds only when some random number with an even order relative to factorized composite integer is fed as an input to the quantum period finding algorithm. Moreover, post processing of the quantum measurement recovers the correct order only for some subset of possible values. It is well known that numbers with even orders are found with probability not less than 1/2. However, numerical simulation proves that probability of such event exhibits grouping on some discrete levels above that limit. Thus, one may conclude that usage of the common bound leads to underestimation of the successful factorization probability. Empirical formulas on expected success probability introduced in the paper give rise to the more profound analysis of classic part behaviour of the Shor's algorithm. The experimentally observed grouping still awaits for theoretical explanation.

Keywords: quantum computation, factorization.

1 Introduction

Theoretical study of quantum systems serving as computational devices has achieved tremendous progress in the last several years. There exist strong theoretical evidence that quantum computers are able to break all presently used asymmetrical algorithms whose security is based on computational complexity. Qualitative progress results from the massive parallel computation realized as the quantum system controlled evolution. The efficient function period finding seems to be one of the most stimulating development in the field, as it provides efficient solution for factorization problem, which seems to be the Holy Grail of the classic algebra. Presently, interest in the factoring problem is especially great for composite integers being a product of two large prime numbers – the ability to factor such integers is equivalent to the ability to read information encoded via the RSA cryptographic system [1]. Thus, quantum computers, if built, pose a serious challenge for the security of today's asymmetric cryptographic systems. However, the quantum factorization is a probabilistic process, unlike the classical one, and profound analysis of its efficiency is required. The algorithm random

A. Kwiecień, P. Gaj, and P. Stera (Eds.): CN 2010, CCIS 79, pp. 49–57, 2010.

behavior comes both from the inherent nature of quantum measurement and specific construction of supporting classic calculations. The researchers in the field are mainly interested in the probability of success of the quantum part and proposed many modifications improving that aspect of the original Shor's proposal [2,3]. It is reported that quantum devices are extremely reliable if multiple measurements and sophisticated post processing are employed [4]. Much less attention was devoted the classic part, although it has large impact on the overall factorization procedure efficiency. It was proved in [5] that lower bound on probability of finding parameter suitable for factorization is equal to $1/2$. The classic parameter selection failure causes entire algorithm repetition that is very undesirable as the repetitive runs of quantum device are undoubtedly costly both in terms of time and money. The fine grained formulas on the above mentioned probability are vital to overall cost estimation of the factorization procedure.

The aim of this paper is to provide an analysis of the randomness introduced by that classical part of the algorithm. The influence of that step on the overall algorithm performance one can find in [6]. Next two sections describe quantum and classic part of the quantum factorization and provide some hints about expected success probability. The methodology of the quantum factorization with classic computer is introduced in Sect. 4. The Section 5 presents simulation results and novel analytical expressions on the expected success probability.

2 Quantum Period Finding

Consider the circuit from Fig. 1. It processes the control register and the targed one composed from K and L qubits, respectively. Both registers are initialized to the state $|0\rangle$ on the start of the device. The symbol $H^{\otimes K}$ denotes the tensored product of the operator which is, in fact, an independent application of the transformation to each of the K qubits which are forming the register. The Hadamard operator applied to a single qubit $|0\rangle$ results in the state with equally likely qubits $|0\rangle$ and $|1\rangle$

$$H|0\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \frac{1}{\sqrt{2}} \left(|0\rangle + |1\rangle\right) \tag{1}$$

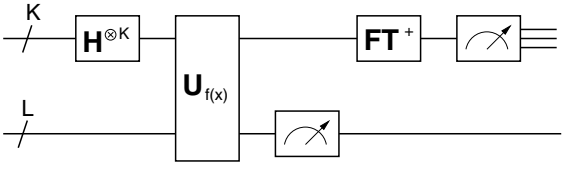


Fig. 1. Quantum circuit for period finding

The tensored product of Hadamard operators transforms the input state into equiprobable superposition of all possible configurations of the control register

$$|\psi\rangle = H^{\otimes K}|0\rangle|0\rangle = \sum_{x=0}^{M-1} |x\rangle|0\rangle \tag{2}$$

where $M = 2^K$. The unitary transformation $U_{f(x)}$ calculates the value of the function under consideration for all x

$$|\psi\rangle = \sum_{x=0}^{M-1} |x\rangle|f(x)\rangle \ . \tag{3}$$

Let the function $f(x)$ have the period r and be injective within the period. The measurement of the target register selects randomly one of the possible function values and quantum state collapses to

$$|\psi\rangle = \frac{1}{\sqrt{Q}} \sum_{k=0}^{Q-1} |x_0 + kr\rangle|f(x_0)\rangle \tag{4}$$

where x_0 is some random offset and $Q = \lfloor M/r \rfloor$ is the number of periods in the observed domain. The inverse Fourier transform applied to the control register extracts the random offset to the phase factor

$$\begin{aligned} |\psi\rangle = FT^{\dagger} \frac{1}{\sqrt{Q}} \sum_{k=0}^{Q-1} |x_0 + kr\rangle = \frac{1}{\sqrt{MQ}} \sum_{k=0}^{Q-1} \sum_{l=0}^{M-1} e^{-2\pi j \frac{l(x_0+kr)}{M}} |l\rangle = \\ = \frac{1}{\sqrt{MQ}} \sum_{l=0}^{M-1} e^{-2\pi j \frac{lx_0}{M}} \left(\sum_{k=0}^{Q-1} e^{-2\pi j \frac{lkr}{M}} \right) |l\rangle = \\ = \frac{1}{\sqrt{MQ}} \sum_{l=0}^{M-1} \frac{\sin\left(\pi \frac{lr}{M} Q\right)}{\sin\left(\pi \frac{lr}{M}\right)} e^{-2\pi j \frac{l}{M}(x_0 + r(Q-1)/2)} |l\rangle \ . \end{aligned} \tag{5}$$

The subsequent measurement of that register selects the state $|l\rangle$ with probability

$$p(l) = \frac{1}{MQ} \left(\frac{\sin\left(\pi \frac{lr}{M} Q\right)}{\sin\left(\pi \frac{lr}{M}\right)} \right)^2 \ . \tag{6}$$

When M is a multiple of r the numerator always vanishes. The probability is different from zero only when denominator also vanishes – number lr/M is an integer for $l \in [0, rM - 1]$ (Fig. 2). If M is not the multiple of r, then probability distribution has sharp peaks when lr/M is close to an integer k. Thus, from measurement output l and target register size M one has to find estimate for the fraction k/r, where k is some integer and r is unknown period. It is proved that continued fraction expansion leads to correct k/r from estimate l/M if

$$\left| \frac{l}{M} - \frac{k}{r} \right| < \frac{1}{r^2} \tag{7}$$

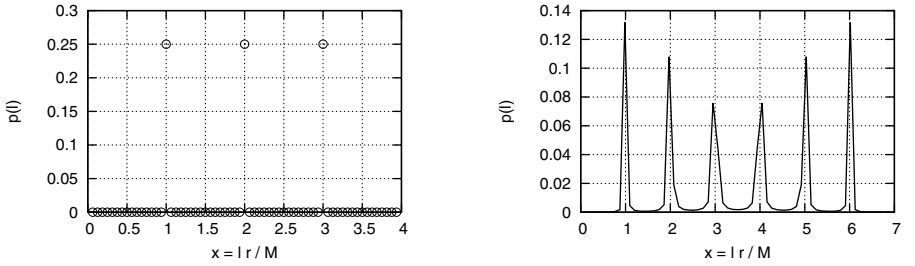


Fig. 2. Probability distribution for $r = 4$ (left) and $r = 7$ (right) with target register size $M = 2^6 = 64$

that means that one can recover k/r if measured l is sufficiently close to the local maximum. The failure probability is directly related to the width of peaks that in turn depends on the control register size [7]. Thus the confidence of quantum measurement may be enlarged to arbitrary accuracy by appropriate selection of the control register size. However, for k not being coprime to r, the estimated period would be divided by a common factor and period recovery fails.

The finding of the function period seems to be not too exciting application of quantum computers. However, the security of the RSA and ElGamal cryptosystems is based on the difficulty of exponential function ($f(x) = a^x \bmod N$) period finding. The knowledge of the function period allows for efficient modulus factorization or discrete logarithm calculation. In effect, an polynomial time period finding algorithm opens a way to breaking those asymmetrical cryptosystems.

3 Shor's Algorithm

At the core of the RSA cryptosystem security lays the assumption that factorization of the composite number formed as the product of two large primes cannot be performed in polynomial time. Presently, the best classic algorithm performs factorization in time proportional to $e^{n^{1/3}(\log n)^{2/3}}$, where n is the composite number size expressed in bits. However, the assumption of factorization ineffectiveness is purely empirical – there is no theoretical proof that this property holds even for classic computers.

The algorithm proposed in [8] completely changed the view on factorization complexity. It exploits reduction of factorization to order finding based on the following observation. Let N be the composite number and $a < N$ is coprime to N. The order of a is the smallest number such that $a^r \bmod N = 1$, thus r is the period of exponential function. If the order of a is even then one may write

$$\left(a^{r/2} - 1\right)\left(a^{r/2} + 1\right) \bmod N = 0 \tag{8}$$

and

$$p = \gcd\left(a^{r/2} - 1, N\right) \tag{9}$$

is a nontrivial factor of N provided that

$$a^{r/2} \bmod N \neq N - 1 \ . \tag{10}$$

Thus factorization of composite number N is reduced to the order finding of some number a. Classical order finding gives no advantage over other factorization algorithms as its complexity is also exponential. However, it was shown in Sect. 2 that the order of a may be determined in polynomial time with the help of the quantum computer.

4 Numerical Simulation

The following steps summarize the Shor's algorithm for quantum factorization of the composite number N:

1. Select a random number a coprime to N (otherwise $\gcd(a, N)$ is a factor of N). Only some a are good candidates as the order of a determined by the next step has to be even and condition (10) must be fulfilled.
2. Find the order of a with the quantum computer. The order is successfully recovered only for some subset of valid quantum measurements.
3. Calculate divisor p from the Equation (9) and return to the point 1) with $N = N/p$.

It is clear, that the nature of the above algorithm is probabilistic. The sources of uncertainty are twofold: the randomness of the quantum period finding and the random selection of the number a with desired properties (10). The success

```
success(N) {
  coprime=1 ;
  lucky=0 ;
  a=2 ;
  while (a<N) {
    if (gcd(a,N)==1) {
      ++coprime ;
      x=a ;
      r=1 ;
      while ( x != 1 ) {
        x=x*a mod N ;
        ++r ;
      }
      if ( r mod 2 == 0  )
        if ( a^(r/2) mod N != N-1 )
          ++lucky   ;
    }
    ++a ;
  }
return lucky/coprime ;
}
```

Fig. 3. Pseudocode for calculation of Shor's factorization success probability

probability of a quantum step is well known. However, the influence of the second factor requires further investigation. The literature on the second subject provides only a lower bound on its success probability [5]

$$p(a) \geq 1 - 2^{-(k-1)} \tag{11}$$

where k is the number of prime factors of N. That lower bound has maximal value when composite number is a product of only two prime numbers, what in fact represents the most interesting situation.

The relatively simple code is required for numeric calculation of $p(a)$ (see Fig. 3). The classic order finding algorithm has exponential complexity and very quickly becomes a daunting task for typical PC architectures. Moreover, exponentiation of a very quickly leads to register overflow of constant length integer representation. To overcome that problem one have to take advantage of library for efficient computations in arbitrary precision, what in turn, additionally slows down the program execution.

5 Results

The probability of selection of the parameter a, suitable for factorization, was calculated for composite numbers of the form $N = pq$, where p, q are taken from the list of the first 500 prime numbers. The calculated probability is shown on the Fig. 4, but due to clarity only composites not exceeding 5000 are included on the plot. The observed probabilities are always greater than $1/2$ as predicted by the bound (11). However, presented simulation results exhibit a deeper structure in probability distribution. The grouping of points around some discrete levels is evident and that indicates the existence of a class of composites less resistant to quantum factorization than others. Observed behavior comes from the factors' properties, but there is no satisfactory theoretical explanation of the obtained results in the literature known to the author.

The introduction of empirical formulas on the observed probability levels is the main contribution of the paper. The factors are primes numbers, so they have to be odd (the trivial case of factor 2 is excluded) and may be expressed as

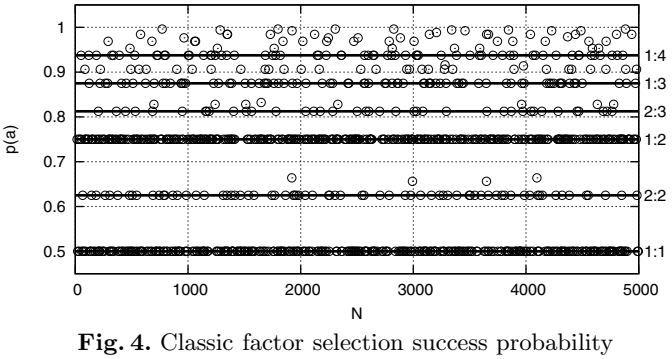


Fig. 4. Classic factor selection success probability

$$p = 2^{\alpha}\mu + 1 \ , \tag{12}$$

$$q = 2^{\beta}\nu + 1 \ . \tag{13}$$

Lets name α and β as the *parity levels* of the factors. The probability of "lucky" parameter a selection for the composite $N = pq$ is then given by the expression

$$p(a) = f(\alpha, \beta) = 1 - \frac{1 + \sum_{\delta=1}^{\min\{\alpha,\beta\}} 4^{\delta-1}}{2^{\alpha+\beta}} \ . \tag{14}$$

The levels predicted by (14) are shown on Fig. 4 as solid lines with respective values of α and β labeled on the right. The perfect matching of simulation results and theoretical predictions is visible. The points not included in the marked levels come from less probable combinations of parity levels that are not placed on figure because of clarity.

The lack of correlation between bits in the prime number representation is the one of its most useful cryptographic properties. It is also very helpful in counting prime numbers with the given parity level. The least significant bit of the prime number representation is always set to "1", as the prime number have to be odd and parity level $\alpha \geq 1$. The prime numbers with parity level $\alpha \geq 2$ have the second least significant bit set to "0", and numbers with $\alpha \geq 3$ have the second and third least significant bit set to "0", and so on. Thus probability that randomly selected prime number has parity level not less than α equals to $2^{-(\alpha-1)}$. Probability that randomly selected prime number has parity level exactly equal to α is given by

$$P(\alpha) = 2^{-(\alpha-1)} - 2^{-\alpha} = 2^{-\alpha} \ . \tag{15}$$

The above considerations have been verified experimentally. The parity level probability density function $P(\alpha)$ was computed for primes less than 10^7. The comparison of numerical experiment and theoretical consideration (15) is presented on Fig. 5.

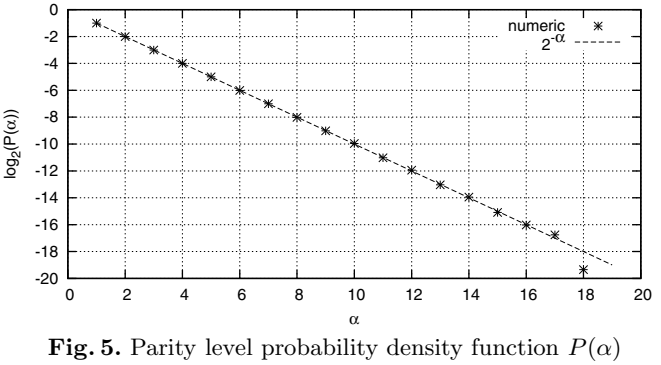


Fig. 5. Parity level probability density function $P(\alpha)$

Some interesting conclusions follow from (14) and (15). The lower bound $p(a) = 1/2$ is reached only when $\alpha = \beta = 1$. Only $P(\alpha = 1)P(\beta = 1) = 25\,\%$ composite numbers of the RSA form fulfill this condition. Thus, for $75\,\%$ factorization cases one may expect faster algorithm convergence than the one estimated from the commonly used bound (11). On average, the selection probability of the lucky parameter is equal to

$$\sum_{\alpha=1}^{\infty}\sum_{\beta=1}^{\infty} f(\alpha,\beta)P(\alpha)P(\beta) = 0.736 \ . \tag{16}$$

Also from (14) and (15). one can numerically estimate the percentage of composite numbers for which lucky parameter selection probability is above some threshold, for instance, more than $20\,\%$ composites have $p(a) > 0.9$

$$\sum_{\alpha,\beta:f(\alpha,\beta)>0.9} P(\alpha)P(\beta) = 20.5\,\% \ . \tag{17}$$

6 Conclusion

The quantum factorization algorithm represents a breakthrough in complexity theory and modern cryptography. The Shor's algorithm owes its fame to polynomial time breaking of virtually all presently used public key algorithms. Unfortunately, the practical breaking is out of reach yet because factorization of the number 15 is still one of the most complicated quantum computations [9]. However, very rapid progress in that field is observed so it is difficult to estimate the time horizon when practical computation will be in scientists' reach. The quantum factorization was analyzed many times and several modifications were proposed to the original algorithm version improving its speed and efficiency. However, the researchers were concentrated so far on the probabilistic aspect of the quantum measurement. The randomness introduced by the classical parts of the algorithm still requires further investigation. The computer simulation results presented herein expose that success rate of the algorithm is usually underestimated. Theoretical considerations are required to validate empirical formulas introduced in the paper.

References

1. Gerjuoy, E.: Shor's factoring algorithm and modern cryptography. An illustration of the capabilities inherent in quantum computers. Am. J. Phys. 73(6), 521–540 (2005)
2. Knill, E.: On Shor's quantum factor finding algorithm: Increasing the probability of success and tradeoffs involving the Fourier Transform modulus. Technical Report LAUR-95-3350, Los Alamos National Laboratory (1995)
3. McAnally, D.: A refinement of Shor's algorithm (2001)
4. Bourdon, P.S., Williams, H.T.: Probability estimates for Shors algorithm. Quant. Inf. Comput. 7(5&6), 522–550 (2007)

5. Ekert, A., Jozsa, R.: Quantum computation and Shor's factoring algorithm. Rev. Mod. Phys. 68(3), 733–753 (1996)
6. Zawadzki, P.: A numerical simulation of quantum factorization success probability. In: Tkacz, E., Kapczyski, A. (eds.) Internet – Technical Developments and Applications. Advances in Intelligent and Soft Computing, vol. 64, pp. 223–231. Springer, Heidelberg (2009)
7. Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information. Cambridge University Press, Cambridge (2000)
8. Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM J. Sci. Statist. Comput. 26, 1484–1509 (1997)
9. Vandersypen, L.M.K., Steffen, M., Breyta, G., Yannoni, C.S., Sherwood, M.H., Chuang, I.L.: Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance. Nature 414, 883–887 (2001)

FPGA Based "Intelligent Tap" Device for Real-Time Ethernet Network Monitoring

Rafał Cupek, Piotr Piękoś, Marcin Poczobutt, and Adam Ziębiński

Silesian University of Technology,
Institute of Informatics
rcupek@polsl.pl, piotr.piekos@pidi.pl,
marcin.poczobutt@gmail.com, adam.ziebinski@polsl.pl

Abstract. This paper describes an "Intelligent Tap" – hardware device dedicated to support real-time Ethernet networks monitoring. Presented solution was created as a student project realized in Institute of Informatics, Silesian University of Technology with support from Softing A.G company. Authors provide description of realized FPGA[1] based "Intelligent Tap" architecture dedicated for Real-Time Ethernet network monitoring systems. The practical device realization and feasibility study conclusions are presented also.

Keywords: real-time Ethernet networks, network diagnosis, packet-analysis methods, FPGA, RISC soft-core architecture.

1 Introduction

The real-time industrial network area is now changing from solutions based on standard serial communication into area of dedicated real-time (RT) Ethernet based solutions. Another fact is the increasing size of newly created industrial networks. In fact the network structures used for the distributed industrial control systems have become more and more complicated. New functional requirements for horizontal and vertical communication are given. Despite these facts there are very few solutions which can help engineers to startup and maintain industrial networks based on real-time Ethernet. There are some general purpose network analyzers, but very few solutions are dedicated to low level Ethernet based industrial networks [1].

The traffic structure in the real-time Ethernet network consists of large number of deterministically exchanged small Ethernet packets. To support RT Ethernet network monitoring the preliminary filtration of data is necessary due to large amounts of data incoming in one instance of time [2]. Current paper contains "Intelligent Tap" hardware device presentation which is dedicated to support industrial network monitoring.

[1] FPGA – Field Programmable Gate Array – an integrated circuit which internal structure can be modified (programmed) after manufacturing.

A. Kwiecień, P. Gaj, and P. Stera (Eds.): CN 2010, CCIS 79, pp. 58–66, 2010.

Due to large number of transmitted packets real-time Ethernet network diagnosis using packet-analysis methods becomes a task, which requires high computational power and memory consumption [3]. One of solutions that allow to reduce amount of data, received by dedicated network-analysis software, is a filtering device implemented as a dedicated hardware. This paper clarifies the "Intelligent Tap" hardware filtering device concept, together with its principle of the operation and presents the realized solution for real-time Ethernet traffic filtration embedded into FPGA structure as a configurable IP core[2] named Ethernet Frame Filter. This idea is presented in Fig 1.

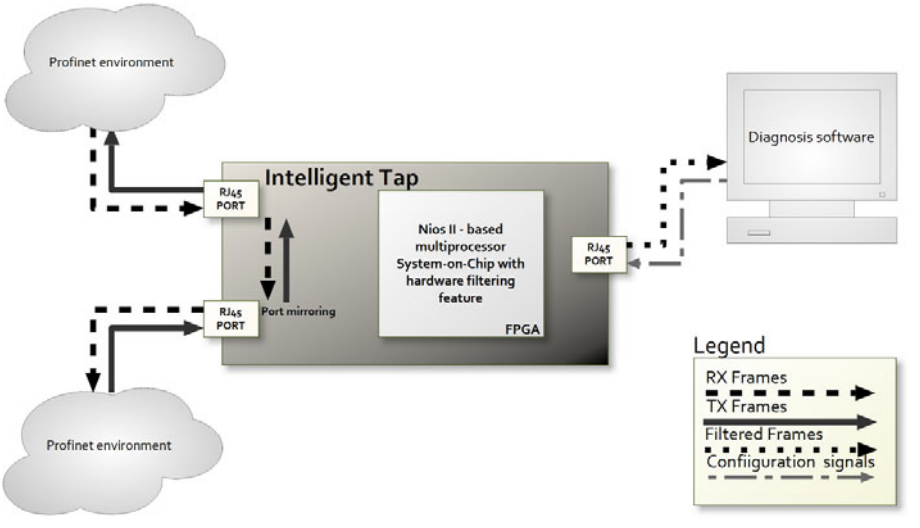


Fig. 1. "Intelligent Tap" hardware device idea

Three main goals for considered the "Intelligent Tap" architecture were formulated:

- Design of a sufficient method of Ethernet packets filtration in hardware, being able to process up to 100 Mb/s full duplex data flow.
- Development of a specification of hardware filtering module that could be embedded into Altera's family FPGA chips as an IP core.
- Delivery of a working prototype of a filtering device on particular platform: Altera's Cyclone III Development Board.

2 "Intelligent Tap" System on Chip Structure

The network monitoring process based on "Intelligent tap" filtering device consists of two main parts. First one is the software level: filtering schema preparation. The

[2] IP core – reusable unit of logic that is the intellectual property (IP) of one party. IP cores are commonly used as building blocks of FPGA designs.

user of the monitoring system is responsible for the filtering pattern preparation. For this task a dedicated filter description language and the related software components were created. This software allows for user defined mask preparation and is responsible for its translation into proper low-level register configuration of the IP core. Second part of realized solution is FPGA based "Intelligent Tap" hardware architecture presented in Fig. 2. Each rectangle in Fig. 2 embodies single System on Programmable Chip Module. Presented system comprises of NIOS II embedded processor, an IP core responsible for filtration, two Ethernet MAC's[3] and a number of standard system components. Those include the following: SDRAM controller that allows access to external SDRAM chip used as CPU instruction container as well as data storage. CFI Flash controller connected through a tri-state bridge, and EPCS controller[4] – both capable of initial FPGA configuration and program startup. Sysid component ensures safe compilation of software projects, pll (phase locked loop) divides external crystal oscillator frequency into several clock domains.Timer provides an interval-timer for Avalon-based processor systems and jtag uart module allows USB connectivity during design process.

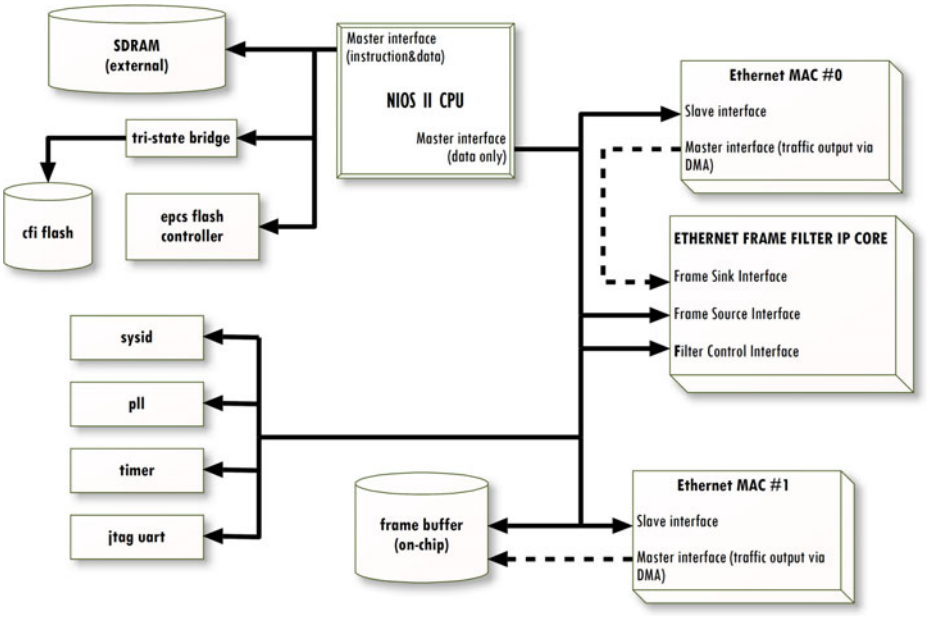


Fig. 2. The FPGA embedded "Intelligent tap" hardware structure

[3] MAC: Medium Access Controller. A piece of hardware that implements data communication protocol sub-layer specified in OSI model. In FPGA design commonly occurs as an IP core.

[4] EPCS chip controller – an IP core controller for AlteraâĂŹs Serial Configuration Device a.k.a. EPCS device.

Three different data channels can be distinguished in the Fig. 2:

1. *Instruction and data Avalon buses* represented by solid arrows indicates the communication between instruction master port of the Nios II CPU and peripherals that are able to store processor's instructions (program). Data master port is also connected to those peripherals to allow accessing data located on those mediums or to write to them. This includes following modules: EPCS chip controller, CFI Flash (through tri-state Bridge) and SDRAM. Last memory device (SDRAM) was used during testing. It stored both program and data information. Initial FPGA configuration loaded from EPCS chip should be considered overwritten.
2. *Data Avalon buses* (solid arrows) show typical Avalon Memory Mapped Interface's interconnections between NIOS II and those of its peripherals that do not consist program memory. Media Access Controllers, Ethernet Frame Filter, Frame Buffer memory and standard SOPC components (PLL, Sysid, timer, jtag uart) are accessed this way by NIOS II CPU.
3. *Direct Memory Access buses*[5] represented by dashed arrows define those interconnections that are omitting CPU and define communication between system's modules. Ethernet packets are sent from MAC using DMA interface. Two DMA sinks are present in the system: Ethernet Frame Filter module and Frame Buffer.

Filtration schema – once translated from the script query into register compatible sum of minterms form – is downloaded into FPGA structure (Ethernet Frame Filter (EFF) IP Core). This Ethernet Frame Filter preparation phase have to be done before actual network monitoring. Second phase starts after the IP core configuration is uploaded – data capturing begins. Media Access Controllers Ethernet MAC 0, ETHERNET MAC 1 insert the packets directly inside internal frame buffer of the Tap. This is done via DMA channel. If the packet arrived successfully to the buffer, MAC asserts an interrupt, informing CPU about its arrival. This causes an update (software controlled) of Tap status registers and starts the filtering process. Depending on the filtering result, an interrupt is asserted, informing that the frame matched filter's current configuration and can be sent to external buffer (IP stack, SDRAM etc.). If the filtering result was positive, packet is copied. This concludes data capturing phase, which repeats itself in the loop until filtering process is terminated.

Depending on network's payload and reference clock driving IP core, another frame might be copied into EFF buffer, while the previous one is being sent. In order to provide fast and "transparent" filtering, buffer space is paged – internal controller alternating inserts frames into upper or lower part of the buffer. Since dual-port memory is implemented, it allows parallel reading from lower part, while the upper is being overwritten by next incoming frame and vice versa. Presented Tap device is implemented on Cyclone III FPGA (40k logical elements) using only peripherals located on DBC3C40 [4] development board.

[5] Direct Memory Access (DMA): feature that allows access to certain hardware subsystems independently of the central processing unit.

Two Ethernet ports are functional, however only one of them is "equipped" with hardware filtering capability. Second Ethernet port serves as monitor port. Filtering output is redirected onto PHY[6] responsible for this port. Although not saved on any mass storage medium, filtering results (number of frames on filtering output) can also be viewed on seven segment display located on board. On-board reference clock generator provides 50 MHz signal. It is later divided by Phase-locked loop module to generate 75 MHz reference clock for all peripherals included in System on Programmable Chip design.

3 Performance Tests

Researches of the "Intelligent Tap" performance under heavy load conditions was performed in simplified Ethernet topology. Tap Ethernet port (the one capable of hardware filtering) was connected directly to packet generator soft-ware running on PC. Ethernet Frame Filter module was configured so that every incoming frame was passed to the external monitor port. Such EFF setup simulated maximum CPU load condition, since it resulted in execution of Interrupt Service Routine every time the packet arrived to EFF.

This assured execution of maximal number of instructions per time unit. As a packet generator AnetTest software was chosen. It allows creation of packets with different lengths varying from smallest possible valid Ethernet packet to largest ones defined by IEEE 802.3 standard. Generated packets can also be sent with arbitrary defined time intervals between them. In every tested scenario no time intervals were added, therefore maximum effective payload was determined by overall system performance – a PC computer equipped with Intel's Core2 Duo CPU and 100 Mb/s compatible Ethernet network card, governed by Microsoft Windows operating system. Such configuration allowed generation of following bandwidth utilization efficiency: 7.8 Mb/s for smallest possible Ethernet packet, up to 83.5 Mb/s for maximal Ethernet packet size. This radical change in payloads occurred due to system's limited ability to generate high packet per second rate (it did not exceeded 14 000 packets/s for AnetTest generator). The results of high payload testing are presented in Fig. 3.

Since maximal bandwidth utilization efficiency for 64 bytes packet size equals 84.2%, and AnetTest generated approximately ten times lower payload, different generator was used to provide more comprehensive results. This flooding software executed on UNIX based server station with 100 Mb/s compatible Ethernet network card. Generated traffic achieved levels close to maximal (approx. 77 Mb/s), using 60-bytes ARP[7] packets. Test results present that for larger packets (exceeding 600 bytes) number of packets were lost (they did not shown up on the monitor port of Technology Demonstrator). It appears that the maximal supported payload dangles around 41 Mb/s for frames larger than 600 bytes. This problem was not explicitly identified within internal system structure; however

[6] PHYceiver is a device that operates at physical layer of OSI network model.
[7] ARP – Address Resolution Protocol.

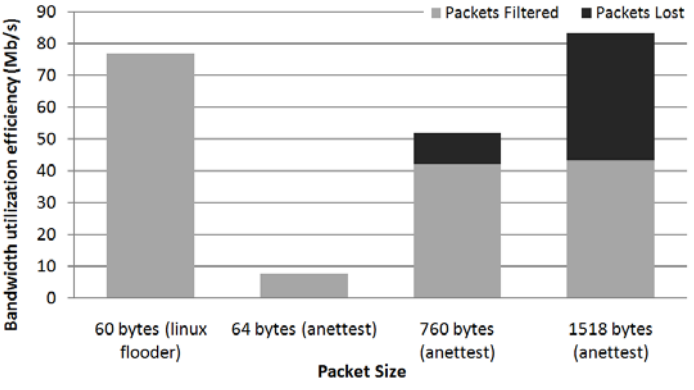


Fig. 3. Results of high payload testing

few factors that causing it were identified. Since packet size is the variable causing the problem, such threshold is most likely defined by ineffective copying operations (the bigger the frame is, the more clock cycles are needed to transfer the packet from one buffer to another within system on chip). Copying operation from the Ethernet Frame Filter IP core's internal storage to frame buffer on-chip memory takes place during execution of interrupt service routine with priority higher than MAC IRQ[8] priority. Accordingly, it cannot be halted by an incoming packet event (announced as MAC interrupt request), which results in packet loss. Additionally, software loaded to FPGA did not support multi-threading, so despite of DMA channel established between MAC and filter, and filter's support for multi-threading is impossible to receive new packet from MAC while the previous one is being sent to the monitor port of the Tap device. Running an optimized version of the software and increasing the reference clock frequency resolved the problem. This leads to the conclusion that system in presented form oscillates on the border of theoretical magnitude of network traffic. It does not completely fulfill 100 Mb/s constraint for software compiled in debug mode and reference clock of 75 MHz, while it obtains 100 Mb/s functionality after the increase of reference clock to 100 MHz and execution of optimized software in release mode.

4 "Intelligent Tap" Behavior in Profinet I/O Environment

Although presented "Intelligent Tap" may be used for any 10/100 Mb/s Real-time Ethernet frame filtering the practical tests were made in reference Profinet I/O environment that consisted of three Profinet compatible IO devices and Siemens PLC (Programmable Logic Controller) (S300 Series). All of mentioned devices were connected to Profinet compatible 8-port Siemens switch. During normal

[8] IRQ – Interrupt ReQuest.

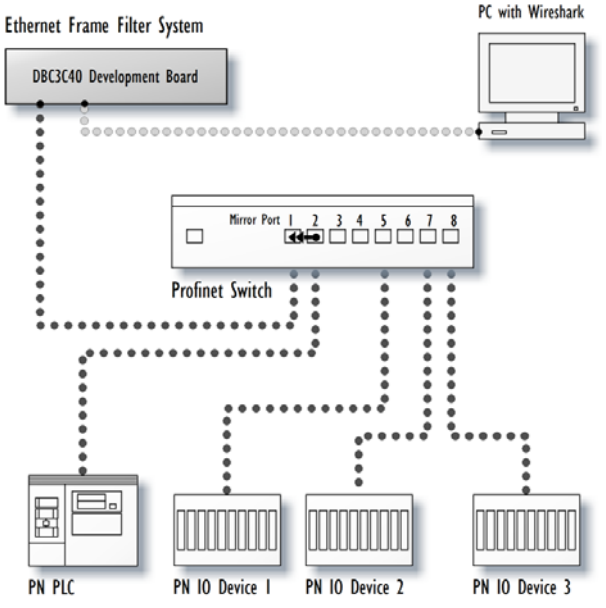


Fig. 4. The Profinet I/O "Intelligent Tap" test environment

operation, reference environment generated some (around 15–20%) VLAN[9] tagged frames [5,6]. The Profinet I/O test environment is presented in Fig. 4.

The Ethernet Frame Filter System gathered outgoing and incoming data from the PLC device through the mirror port of the switch. Testing procedures were common for following presented below PROFINET I/O communication scenarios:

- Startup of Profinet devices. Communication establishment between IO Supervisor and IO Device was investigated. Traffic consisted mainly of broadcast/multicast, LLDP[10] and DCP[11] frames. Network load did not exceed 0.01 Mb/s.
- Normal network operation where vast majority of the traffic was generated by RT frames. Bandwidth utilization in this case was evaluated to be approximately 2 Mb/s.
- Malfunctions imitation of different elements from reference topology by disconnecting or powering off some of the devices. Different elements were forced to produce alarm type frames and reestablish the communication after plugging them back into the network.

[9] VLAN – Virtual Local Area Network, is a group of hosts with a common set of requirements that communicate as if they were attached to the same broadcast domain, regardless of their physical location.
[10] LLDP – Link Layer Discovery Protocol.
[11] DCP – Device Control Protocol.

Various filtering schema were uploaded to the "Intelligent Tap" device in order to investigate its behavior under three scenarios presented above:

- all packets transmitted to PC with Wireshark software,
- no packets transmitted to PC with Wireshark software,
- only alarm frames were allowed to pass to the PC,
- bidirectional communication between only one of the IO devices and PLC,
- unidirectional communication between only one of the IO devices and PLC,
- multicast and broadcast frames only,
- real time traffic only.

More sophisticated scenarios utilizing all features of "Intelligent Tap" device were tested also for example: combination of different filtration schema uploaded to "Intelligent Tap" device. Scenarios investigated had exploited most of the cases that can occur in practical applications. Positive results were obtained in every investigated set of circumstances; however one must remember that reference topology assembled using only few Profinet devices.

5 Conclusions

This article describes "Intelligent Tap" student's project results and practical tests of network monitoring hardware device realized in Institute of Informatics Silesian University of Technology with cooperation with Softing A.G. The Tap hardware support for software real-time Ethernet networks monitoring was presented. Described solution was implemented and tested as a preliminary study for a commercial device that would compete with several other available network monitors, such as ProfiTAP device from Grid Connect Inc.

Although realized in Profinet I/O environment tests didn't find any data losing the heavy load performance tests had shown that in some extreme conditions transmitted frames may be lost. Lost packets were caused by very have network load and long frame transmitted. Because as the principle industrial networks use short packets and lower traffic load planned in the network design process [7] such condition weren't observed. Because of the more and more complicated network structure we can't guaranty that such conditions will not appear in real installation.Additional restriction is the fact that second Ethernet port serves as monitor port only. Such configuration allows for one directional network traffic monitoring or requires additional switch device with mirroring port as it was done in realized test environment presented in Fig 4.

Presented "Intelligent Tap" device has shown that embedded FPGA architectures based on one processor may by insufficient for real-time Ethernet network monitoring. Presented architecture may be improved by multi-processor parallel architecture which will offer parallel processing of receiving and sending RT data and filtering process procedures. Although presented solution can't be used for critical data registration many network monitoring applications may use it for statistical real-time network analysis and for network maintenance and startup tasks as well.

References

1. Cupek, R., Bregulla, M., Huczala, L.: PROFINET I/O network analyzer. In: Kwiecień, A., Gaj, P., Stera, P. (eds.) 16th Conference on Computer Networks, CN 2009, Wisła, Poland. CCIS, vol. 39, pp. 242–251. Springer, Heidelberg (2009)
2. Cupek, R., Huczala, L.: Passive PROFINET I/O OPC DA Server. In: 14th IEEE International Conference on Emerging Technologies and Factory Automation (2009)
3. 802.1AB Station and Media Access Control Connectivity Discovery. IEEE Computer Society, New York (2005)
4. Nios II Hardware Development Tutorial. Altera Corporation, San Jose (2007)
5. Popp, M., Weber, K.: The Rapid Way to PROFINET. PROFIBUS Nutzerorganisation e.V., Karlsruhe (2004)
6. PROFINET Technology and Application In: Siemens information materials, Karlsruhe (2005)
7. Kleines, H., Detert, S., Drochner, M., Suxdorf, F.: Performance Aspects of PROFINET IO. IEEE Transactions on nuclear science 55(1) (February 2008)

Throughput Upper Limit for IEEE 802.11 Networks with Block Acknowledge and Frame Aggregation

Bartłomiej Zieliński

Silesian University of Technology, Institute of Computer Science
ul. Akademicka 16, 44-100 Gliwice, Poland
Bartlomiej.Zielinski@polsl.pl

Abstract. The paper presents new transmission procedures present in IEEE 802.11 standard and compares them to the "traditional" frame exchange method. On the basis of their operating rules, analytical equations are derived that allow calculate network efficiency or effective throughput. The throughput upper limit is also calculated and compared for all considered frame exchange procedures.

Keywords: IEEE 802.11 standard, block acknowledge, frame aggregation, throughput upper limit.

1 Introduction

IEEE 802.11 standard [1] is currently the leading solution in the range of wireless local area networks. Peripherials compatible with this standard are available for numerous types of devices, not necessarily being computers. For example, they can be embedded into laptops, palmtops, but also high-class digital photo cameras (like Canon EOS 1D), DVD players, digital satellite receivers etc. In addition, 802.11 has now several modifications necessary for QoS (802.11e) and high throughput (802.11n) support.

The standard architecture, in a brief, consists of data link and physical layers specification. While data link layer is uniform, there are several physical layer definitions, applying various modulation and encoding methods that allow achieve multiple data rates. However, in conformance with ISO/OSI network model, physical layer details must be transparent to higher layers. Thus, necessary mechanisms must be included in physical layer signaling rules that are not visible to data link; unfortunately, such approach introduces additional protocol overhead that does not remain without influence on overall network performance.

Several paper exist that analyse the efficiency of 802.11 protocol, e.g., [2,3]. In [3] a good explanation of transmission procedures and their influence on effective throughput is given using analytical methods. Using similar approach, it has been proved [2], that "traditional" frame exchange rule has a throughput upper limit (TUL) of about 75 Mbps even when transmission rate is infinite. However, new transmission methods have been introduced in 802.11 standard since then. In

A. Kwiecień, P. Gaj, and P. Stera (Eds.): CN 2010, CCIS 79, pp. 67–75, 2010.

this paper, using methods similar to those shown in [2,3], these transmission procedures are analyzed and compared to the "traditional" one. Thus, the paper extends some results presented in [2].

2 Data Transmission in IEEE 802.11 Standard

In IEEE 802.11 standard, data transmission may proceed according to few frame exchange procedures. For many years, only a single procedure, has been defined. Further in this paper it is referred to as basic frame exchange. Later, together with QoS enhancements, block acknowledge has been proposed in IEEE 802.11e in order to reduce protocol overhead by reducing number of acknowledge frames. Finally, 802.11n [4] introduces frame aggregation which allows even further overhead reduction by merging frames into long frame sequences.

2.1 Basic Frame Exchange

In basic frame exchange using DCF (*Distributed Coordination Function*) protocol, Data and Ack frames alternate. Each frame must be preceded by PLCP (*Physical Layer Convergence Protocol*) preamble and header. Thus, frame exchange process runs as presented on Fig. 1.

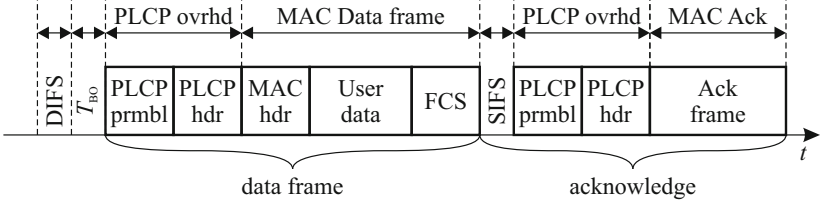


Fig. 1. Basic frame transmission exchange

Bearing in mind frame exchange elements shown on the diagram, transmission cycle duration might be expressed as:

$$T_{\mathrm{p}} = T_{\mathrm{DIFS}} + T_{\mathrm{BO}} + T_{\mathrm{SIFS}} + 2T_{\mathrm{PLCP}} + T_{\mathrm{Data}} + T_{\mathrm{Ack}} \ , \tag{1}$$

where: T_{DIFS} and T_{SIFS} are DIFS (*Distributed Inter-Frame Space*) and SIFS (*Short Inter-Frame Space*) duration, respectively, while T_{PLCP} – duration of PLCP preamble (T_{prmbl}) and header (T_{hdr}). These values are defined in specifications of individual physical layers and collected in table 1. In turn, T_{BO} represents backoff period duration, which, under perfect conditions and according to explanations given in [3], may be simplified to

$$T_{\mathrm{BO}} = \frac{CW_{\min}}{2} T_{\mathrm{slot}} \ . \tag{2}$$

T_{slot} is a slot time [s] (Table 1), while $CW_{\min}$ (*Contention Window*) – minimum number of contention slots for a given physical layer. In turn, bearing in mind MAC frames formats,

$$T_{\text{Data}} = \frac{8(28+L)}{R_{\text{wl}}} \qquad \text{and} \qquad T_{\text{Ack}} = \frac{8 \cdot 14}{R'_{\text{wl}}} \ , \tag{3}$$

where: L – data field capacity (often referred to as payload) in bytes, R_{wl} – Data frame transmission rate [bps], and R'_{wl} – Ack frame transmission rate [bps]. Within a single transmission cycle, exactly L bytes are transmitted. During calculation of frame transmission times, we must take into account any additional overhead resulting from modulations used in a given physical layer, e.g., 32/33 encoding in FHSS (*Frequency Hopping Spread Spectrum*) as well as tail and pad bits in OFDM (*Orthogonal Frequency Division Multiplexing*) and ERP (*Enhanced Rate Physical*).

Table 1. PLCP-dependent parameters of IEEE 802.11 standard

PLCP	$CW_{\min}$	$CW_{\max}$	T_{SIFS}	T_{slot}	T_{prmbl}	T_{hdr}	Additional overhead
DSSS	31	1023	10	20	144	48	—
FHSS	15	1023	28	50	96	32	32/33 encoding
Ir (1 Mbps)	63	1023	10	8	16	41	—
Ir (2 Mbps)	63	1023	10	8	20	25	—
HR-DSSS (sp)	31	1023	10	20	72	24	—
OFDM	15	1023	16	9	20	4	$\geq$ 22 bits
ERP-DSSS (lp)	15 or 31	1023	10	9 or 20	144	48	18 μs
ERP-DSSS (sp)	15 or 31	1023	10	9 or 20	72	24	18 μs
HT	15	1023	16	9	16	4	—
HT-mixed	15	1023	16	9	16	16 to 40	—
HT-GF	15	1023	16	9	16	12 to 36	—

All times given in [μs]; sp – short preamble, lp – long preamble

2.2 Block Acknowledge

Block acknowledge mechanism allows for transmission of series of multiple data frames which are then commonly acknowledged. The acknowledge itself may be immediate or delayed; the first one is assumed to support higher transmission efficiency [1]. Block Acknowledge procedure must be set up prior to transmission and torn down after the transfer is finished. Assuming that the information to be transmitted is sufficiently long, these initial and final frame exchanges do not play an important role from the point of view of protocol efficiency and thus they will not be further considered.

When using Block acknowledge, transmission cycle consists of multiple (but no more than 64) Data frames. The latest of them is followed by the BlockAckReq frame, after which BlockAck frame appears. All frames are separated by SIFS period and preceded by PLCP preamble and header. On the data link layer

level, BlockAckReq frame is 24 bytes long. BlockAck is even longer by 128 bytes as it carries fragmentation-specific information for every acknowledged frame. Information exchange process with Block acknowledge is explained on Fig. 2.

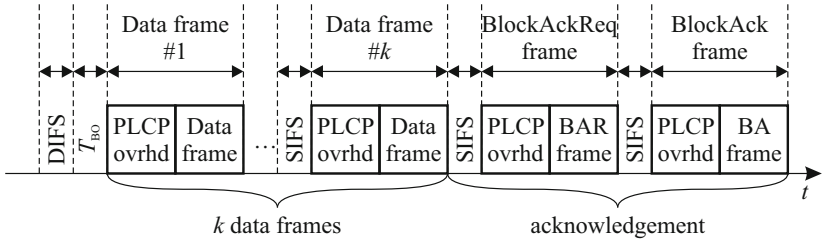


Fig. 2. Frame transmission in Block Acknowledge procedure

Bearing in mind transmission course described above, transmission cycle length may be expressed as:

$$T_{\mathrm{p}} = T_{\mathrm{DIFS}} + T_{\mathrm{BO}} + (k+1)T_{\mathrm{SIFS}} + (k+2)T_{\mathrm{PLCP}} + kT_{\mathrm{Data}} + T_{\mathrm{BAR}} + T_{\mathrm{BA}} \ , \quad (4)$$

where T_{BAR} – transmission time of a BlockAckReq frame, equal to

$$T_{\mathrm{BAR}} = \frac{8 \cdot 24}{R'_{\mathrm{wl}}} \quad (5)$$

and T_{BA} – transmission time of a BlockAck frame, equal to

$$T_{\mathrm{BA}} = \frac{8 \cdot (24 + 128)}{R'_{\mathrm{wl}}} \ . \quad (6)$$

Assuming constant Data frame length, $L_{\mathrm{D}} = k\mathrm{L}$ data bytes are sent within a single transmission cycle.

2.3 Frame Aggregation

Frame aggregation is introduced to reduce the PLCP overhead. As the PLCP frame format is set, the only way to reduce the overhead is using a single preamble and header for multiple Data frames. It is especially important for high transmission rates, because PLCP overhead is always transmitted at the lowest rate defined for a given physical layer. Thus, we can say that preamble and header transmission time is constant, while that of PSDU (*Physical layer Service Data Unit*) decreases with increasing transmission rate. Therefore, protocol overhead increases, while its efficiency – decreases. In order to avoid it, in IEEE 802.11n standard two aggregation methods are proposed: A-MSDU (*MAC Service Data Unit*) and A-MPDU (*MAC Protocol Data Unit*).

A-MSDU Aggregation. A-MSDU aggregation, similarly to Block acknowledge, allows for transmission of a series of Data frames, which are then commonly acknowledged. However, while Block acknowledge requires that each frame was an individual unit containing PLCP preamble and header, frame aggregation allows precede the entire series with a single preamble and header, which are common for all the Data frames. MAC header is also common for all these frames. Each of them is completed by a short, individual header.

When using A-MSDU aggregation, transmission cycle consists of a series of subframes containing individual headers. They are preceded by PLCP preamble and header and typical MAC header. All this information is protected by a common FCS (*Frame Check Sequence*) and acknowledged with a single Ack frame. Information exchange process with A-MSDU aggregation is explained on Fig. 3.

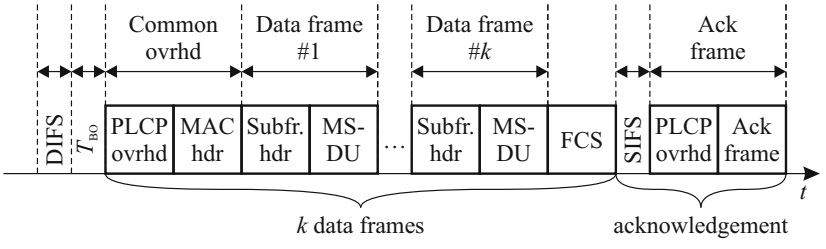


Fig. 3. Frame transmission with A-MSDU frame aggregation

Bearing in mind transmission course described above, transmission cycle length may be expressed as:

$$T_{\mathrm{p}} = T_{\mathrm{DIFS}} + T_{\mathrm{BO}} + T_{\mathrm{SIFS}} + 2T_{\mathrm{PLCP}} + T_{\mathrm{MAC}} + kT_{\mathrm{SubFr}} + T_{\mathrm{Ack}} \ , \tag{7}$$

where k – data block size, while T_{MAC} and T_{SubFr} – transmission times of MAC header and a subframe with its header, respectively. Bearing in mind their formats,

$$T_{\mathrm{MAC}} = \frac{8 \cdot 28}{R_{\mathrm{wl}}} \tag{8}$$

and

$$T_{\mathrm{SubFr}} = \frac{8 \cdot 4 \left\lceil \frac{14+L}{4} \right\rceil}{R_{\mathrm{wl}}} \ . \tag{9}$$

The length of an aggregated frame is limited to 3839 or 7935 bytes, depending on capabilities of communicating stations. This limit may alter the number and length of subframes in two ways.

In the first method, the sender collects MSDU units of a constant size until the remaining buffer space is not sufficient to place a new unit. In this case, the number of aggregated frames equals to

$$k = \left\lfloor \frac{L_{\max}}{4 \left\lceil \frac{14+L}{4} \right\rceil} \right\rfloor \ , \tag{10}$$

thus, the number of data bytes transmitted within a transmission cycle equals to

$$L_{\mathrm{D}} = kL = L \left\lfloor \frac{L_{\max}}{4 \left\lceil \frac{14+L}{4} \right\rceil} \right\rfloor \quad . \tag{11}$$

In the second method, the sender collects MSDU units, and when the remaining buffer space is not sufficient to place a new unit, a shorter unit is added. Its length is selected so that the limit of an aggregated frame is utilised entirely. This variant is less real because of possible difficulties in its implementation. However, as it allows utilise frame length limit more efficiently, it should support higher efficiency. The number of aggregated MSDU units equals to

$$k = \left\lceil \frac{L_{\max}}{4 \left\lceil \frac{14+L}{4} \right\rceil} \right\rceil \quad , \tag{12}$$

while the number of data bytes transmitted within a transmission cycle equals to $L_{\max}$ decreased by organisation information (subframe headers and stuff bytes). As a result,

$$L_{\mathrm{D}} = L_{\max} - \left\lfloor \frac{L_{\max}}{4 \left\lceil \frac{14+L}{4} \right\rceil} \right\rfloor \left(4 \left\lceil \frac{14+L}{4} \right\rceil - L \right) - 14 \ . \tag{13}$$

Regardless of A-MSDU data field capacity, the total length of the aggregated frame equals always to $L_{\max}$.

A-MPDU Aggregation. When using A-MPDU Aggregation, transmission cycle consists of a series of Data frames, preceded with only a single PCLP preamble and header. The Data frames are transmitted immediately one after another, without even a SIFS gap. The cycle ends with a slightly modified block acknowledge. In A-MPDU aggregation, BlockAckReq frame is not used because aggregation forces the use of block acknowledge. Besides, A-MPDU aggregation does not allow fragmentation, thus, BlockAck frame is substantially shorter than that of Block Acknowledge mechanism. The number of aggregated frames may not exceed 64, and the total length of an aggregated frame may not exceed 65535 bytes. Information exchange process with A-MPDU aggregation is explained on Fig. 4.

Bearing in mind transmission course described above, transmission cycle length may be expressed as:

$$T_{\mathrm{p}} = T_{\mathrm{DIFS}} + T_{\mathrm{BO}} + T_{\mathrm{SIFS}} + 2T_{\mathrm{PLCP}} + kT'_{\mathrm{Data}} + T'_{\mathrm{BA}} \ , \tag{14}$$

where k – data block size, T'_{Data} represents Data frame transmission time with aggregation overhead:

$$T'_{\mathrm{Data}} = \frac{8\left(4 + 28 + 4\left\lceil \frac{L}{4} \right\rceil\right)}{R'_{\mathrm{wl}}} \tag{15}$$

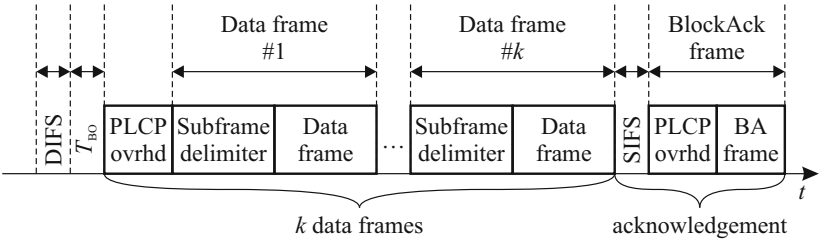


Fig. 4. Frame transmission with A-MPDU frame aggregation

T'_{BA} represents transmission time of a modified BlockAck frame, equal to

$$T'_{\mathrm{BA}} = \frac{8(24+8)}{R'_{\mathrm{wl}}} \quad . \tag{16}$$

3 Throughput Upper Limit

It has been proved [2], that throughput upper limit (TUL) exists for 802.11 networks with basic frame exchange. TUL is calculated, assuming perfect operating conditions (no collisions or transmission errors) and infinite transmission rate. In this case, transmission time of all data link layer frames (such as Data, Ack and others) is zero. Thus, during calculation of transmission cycle duration, only PLCP overhead counts, namely, T_{SIFS}, T_{DIFS}, T_{prmbl} and T_{hdr}. Transmission cycle duration does not therefore depend on Data frame length, or – to be more precise – on data field capacity. However, TUL does depend on it, because it influences on number of data bytes transmitted within a cycle.

Calculated TUL values for OFDM physical layer (compatible with IEEE 802.11a and 802.11g standards) are collected in Table 2. Calculations were performed for payload of 2304, 1500, 256 and 48 bytes. The following data exchange methods were considered:

- basic (DCF),
- block acknowledge with block length set to $k = 64$ frames,
- A-MSDU aggregation with aggregated frame length limit set to $L_{\max} = 3839$ bytes (4k),
- A-MSDU aggregation with aggregated frame length limit set to $L_{\max} = 7935$ bytes (8k),
- A-MPDU aggregation with aggregated frame length limit set to 65535 bytes and block size limited to $k = 64$ frames.

As shown in Table 2, basic data exchange method in 802.11 standard limits TUL to about 118 Mbps, but only when longest possible frames are used. Decrease of frame length to 1500 bytes causes TUL fall down to about 75 Mbps. It

can be easily seen that this method does not ensure effective use of HT (*High Throughput*) physical layer capabilities. Even OFDM physical layer, in some cases, cannot be effectively utilised.

When block acknowledge is used, TUL is almost 4 times as high as for basic method. Using longest frames, we can get TUL of 434 Mbps, while with 1500-bytes frames – about 283 Mpbs. We can therefore assume that block acknowledge allows effectively utilise OFDM layer capabilities, however, for HT layer it is not sufficient.

Table 2. Throughput upper limit for OFDM physical layer [Mbps]

Payload [B]	Basic (DCF)	BlockAck	A-MSDU (4k)	A-MSDU (8k)	A-MPDU
2304	117.78	434.25	111.37	184.12	3119.12
1500	76.68	282.72	145.02	183.35	3093.61
256	13.09	48.25	173.24	174.07	791.98
48	2.45	9.05	136.89	139.26	148.50

Unlike expected, A-MSDU aggregation for the longest frames not only does not bring advantages, but it can even make network achievements worse – when $L_{\max} = 3839$ bytes, TUL is even lower than for basic method. It is caused by larger overhead resulting from aggregation, but, despite aggregation, only a single maximum-size frame (2304 bytes) can be sent. However, for shorter frames, e.g., 1500-bytes long, TUL is twice as high as for besic method. When $L_{\max}$ increases to 7935 bytes, network performance is much better, but still below the capabilities of block acknowledge. Performance does not practically depend on payload size.

A-MSDU aggregation, however, shows high efficiency for shorter frames – TUL is about 135 to 140 Mbps for 48-bytes frames and 173 to 174 Mbps for 256-bytes ones. Similar results for both basic method and block acknowledge are much below these numbers. Thus, we can say that A-MSDU aggregation allows increase network efficiency while transmitting short frames. Nevertheless, it does not ensure effective usage of HT physical layer capabilities, despite they correspond to each other as both are defined in 802.11n standard.

A-MPDU aggregation shows the best performance of all considered transmission methods – TUL for 256-bytes frames exceeds 700 Mbps, while for the longest ones reaches over 3 Gbps. Even for the shortest frames this method is most effective. We can therefore say that A-MPDU aggregation allows effectively utilise transmission rates defined for HT physical layer. It could possibly allow effectively utilise future solutions with even higher transmission rates.

Calculated TUL values for HT physical layer (compatible with IEEE 802.11n standard) are collected in Table 3. The results are slightly worse than those for OFDM, because physical layer preamble and header are longer than in OFDM. It increases protocol overhead and decreases its efficiency.

Table 3. Throughput upper limit for HT physical layer [Mbps]

Payload [B]	Basic (DCF)	BlockAck	A-MSDU (4k)	A-MSDU (8k)	A-MPDU
2304	106.85	363.58	101.55	167.89	2844.16
1500	69.57	236.71	132.23	167.18	2820.89
256	11.87	40.40	157.97	158.72	722.16
48	2.23	7.57	124.83	126.99	135.40

4 Summary and Conclusions

Presented results show that the modifications of data link layer in IEEE 802.11 standard was necessary. In fact, without presented enhancements, the effective throughput would still be limited to relatively low value that would make physical layer utilisation ineffective, and its deployment – useless.

It must be noticed however that the calculations are done for perfect conditions that are far from the real network operating conditions. Nevertheless, such conditions are not impossible, e.g., in a small home network. It would be interesting how the frame exchange procedures that have been analysed in the paper behave in a real network. Such results can be achieved using computer simulations, but still more accurate results could be obtained in an experimental network. However, the results may depend on network hardware and software used for tests. For example, today's access points and network adapter compatible with 802.11n draft standard typically do not allow reach transmission rates higher than 300 Mbps, which is only a half of what is defined in the standard.

References

1. IEEE Std 802.11™–2007: IEEE Standard for Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. IEEE, New York (2007)
2. Xiao, Y., Rosdahl, J.: Throughput and delay limits of IEEE 802. IEEE Commun. Lett. 6(8), 355–357 (2002)
3. Qiao, D., Choi, S., Shin, K.G.: Goodput Analysis and Link Adaptation for IEEE 802.11a Wireless LANs. IEEE Trans. Mobile Comput. 1(4), 278–292 (2002)
4. IEEE Std 802.11n™–2009: IEEE Standard for Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. Amendment 5: Enhancements for Higher Throughput. IEEE, New York (2009)

OPC UA Based Solutions for Integrated Operations

Olav Sande[1], Marcin Fojcik[2], and Rafał Cupek[3]

[1] Sogn og Fjordane University College
olav.sande@goodtech.no
[2] Sogn og Fjordane University College
marcin.fojcik@hisf.no
[3] Silesian University of Technology, Institute of Informatics
rcupek@polsl.pl

Abstract. In all big, industrial systems standardization is necessary. It concerns also offshore systems. There are some ways to realize it: use existing standards and join them together in new functionality or find something new. OPC UA seems to be one of new possibilities. This article presents different offshore standards and how OPC UA can be used in this case, not only in control and supervisory systems, but in sensor and MES/ERP levels as well.

Keywords: OPC UA, industrial system integration, industrial networks.

1 Integrated Operations

Integrated Operations (IO) is a complex IT system used to transfer and process information between offshore and onshore oil and gas exploration and production industry. IO includes broadband connections for data and video of the platform and information related to remotely monitoring and controlling processes in a safe and environmentally friendly way in order to maximize the life value of the field. All these activities together are called Integrated Operations.

In this paper IO aspects of real time data exchange between local and remote sensors through communication network, used to steering and visualization are analyzed. All this data flow includes not only industrial data as signals to/from sensors, actuators, remote control but additional information as audio and video both from control systems and social services as well. These are important challenges to define common definitions of standards in integrated operations which are designed to meet all challenges from both onshore and offshore.

In the beginning IO were limited to some sets of independent systems, specialized onshore units and periodic onshore support. Nowadays in IO we have integrated onshore and offshore centres and continuous onshore support. In future all data exchange should be done in IO centres with continuous integration of all onshore and offshore activities. This integration should be possible also across companies, which is not often executed now. Different companies are not interested in cooperation with other. This is the reason why it is often difficult

A. Kwiecień, P. Gaj, and P. Stera (Eds.): CN 2010, CCIS 79, pp. 76–83, 2010.

to use information from different systems in one, big control system. One of solutions could be using open standards for communication instead of proprietary protocols. But first, it is necessary to find what type of information should be used, transferred, saved and processed [1].

Integrated Operations centres should connect types of activity as monitoring of all sensors, steering of production, data integration, security, redundancy, communication with onshore, remote steering and business level (MES, ERP). This means that is necessary to join completely different types of information as: real-time data from sensors (field data, environment, seismic, drilling, completion, reservoir, production, maintenance, logistic), processing data (collecting data, saving data, common format and standard interface) and using data (visualization, analysis, decisions, steering). All these data could have different time requirements, priorities, amount, protocols and description.

2 Integrated Operations Communication Infrastructure

There are some important activities in IO: production, maintenance of technical conditions, maintenance of Health, Safety and Environment (HSE) and reducing costs. From the structural point of view (industry processes levels) all activities can be divided into different types of communication:

- between devices and controllers: wireless sensor network, AS-I, DeviceNet, Profibus, Foundation Fieldbus, Hart IO,
- between controllers and workstations: Ethernet (cable, radio or fiber),
- between IO centre and onshore and IO centre and remote objects: Ethernet (cable, radio or fiber).

Each level has precise definition of interfaces and parameters. For example [1]:

- Level 0 and 1: ISA S88 and many IEC and ISO standards,
- Level 2: OPC UA, MIMOSA, many IEC and ISO standards,
- Level 3, 4: ISA S95.

As it was mentioned, IO activities were started before some standardizations processes and as set of different activities. This is why all processes was divided and were tried to be standardized separately. Now there are many different standards including different activities:

- Environment: ISO 14040,
- Drilling: WITSML,
- Developments projects: ISO/IEC 11179, IEC 61346, ISO 13628, ISO10303,
- Production: PRODML,
- Operations: IEC/ISO 6264, IEC 61512, IEC 61131, ISO13374,
- Logistics: ISO 15000, ISO 9735.

To make it possible to integrate all these data flows in one system it was necessary to try to make new or use existing standards of description of all the data. This standard is composed of three parts:

- ISA S88 and ISO S95 for assets and physical hierarchy,
- ISO 15926 for asset life cycle management,
- PRODML/WITSML for data acquisition, operation monitoring and reporting.

ISA S88 is a standard in batch process control. It is a design philosophy for describing equipment and procedures.

ISA S95 standard specifies interfaces to the business and logistics levels. It also specifies interfaces to control levels, typically programmable logic controllers (PLCs) and distributed control systems (DCSs) [2].

ISO 15926 is a standard for integration of life-cycle data for process plants including oil and gas production facilities and it is promoted to a standard for data integration, sharing, exchange, and hand-over between computer systems. ISO 15926 gives possibility to build common data model for the whole infrastructure. It is built upon ISO 10303. ISO 15926 defines a format for the representation of information about a process plant. The basis for ISO 15926 is a record of the physical objects that exist within a process plant (identifications, properties, classifications, connection) and changing (maintenance) [2].

Another standard is PRODML – standard method to realize the exchange of information between applications in the domain of oil and gas production based on IT industry Web Services standards. This standard is based on WITSML which is a standard for data exchange between different operators and vendors, and their software systems. WITSML was created mainly for drilling processes, but its success made further development and researching – into PRODML. The main task for PRODML is going to define a data transmission standard between two elements, no matter how different communications structure they have (description of data, frequency of data transfer, amount of sent information).

All above standards are open standards; they are built on established Internet standards such as XML and SOAP.

One of the biggest problems in one system is to create an automatic translation of description of data from one standard to another. One of the solutions is to use the Semantic Web technologies to enable automatic information sharing and conversion across applications. There are many tries to define rules of conversion with using terminologies or ontologies. A terminology can describe what a Person is and ontology can, in addition describe how persons can be related to each other [3,4].

3 OPC UA Communication for Integrated Operations

In an Integrated Operation environment it is vital that information from all conceivable information sources is available for reading in real time for all types information consumers. Real time in this case means seconds. The Figure 1 below shows how the control system can be placed in the information hierarchy in an Integrated Operation environment.

In integrated operations it is crucial that communication is secure, that information sources are secure, and that information interchange is standardized and

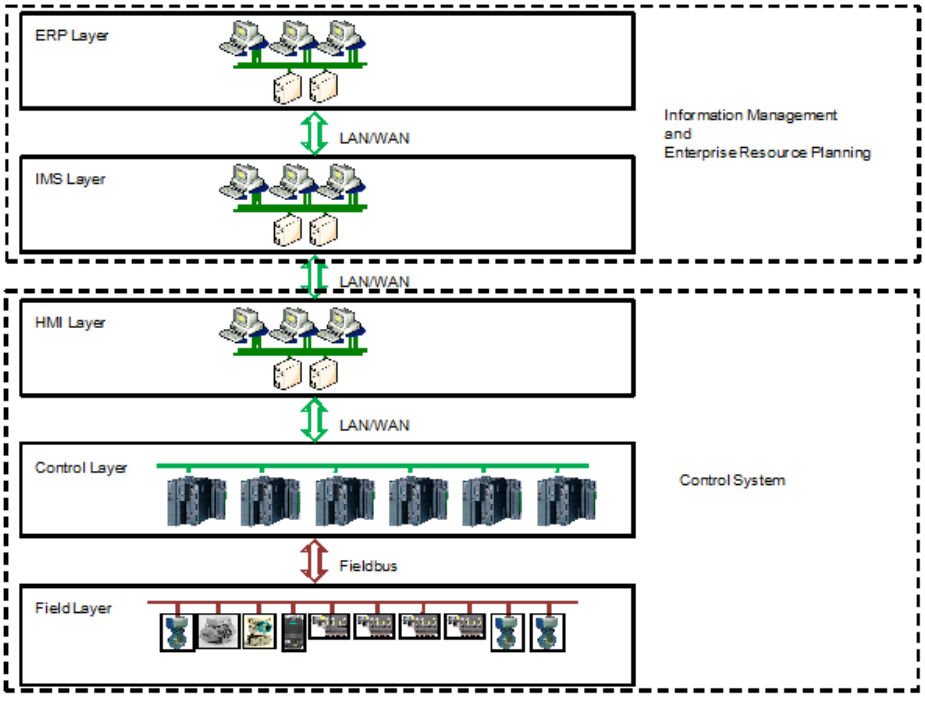


Fig. 1. Layered Integrated Operation system. The control system is the interface to the plant, the real operation. On the layers above the three control system layers and Information Management System and an Enterprise Resource Planning system is installed. All parts are vital in an Integrated Operation environment.

uniform throughout the entire system comprising an Integrated Operation environment. The Figure 2 below depicts a sample Integrated Operation environment involving a number of information sources such as control systems, information management systems and enterprise resource planning systems. All these systems interchange information and the aggregated system is a prerequisite for integrated operations.

The Figure 2 shows an Integration Operation Environment "blob" where information publishers and information subscribers share information. There can be more than one type of IMS, ERP and Control System involved in an integrated operation. In an integrated operation it is also important that all information in the entire system is more or less readily available in all parts of the system. The information is not necessarily used by all parts of the system, but it should be available in case it has to be used in an application. An Integrated Operation environment can also involve other information types than pure process data, such as video conferencing and other types of employer collaboration.

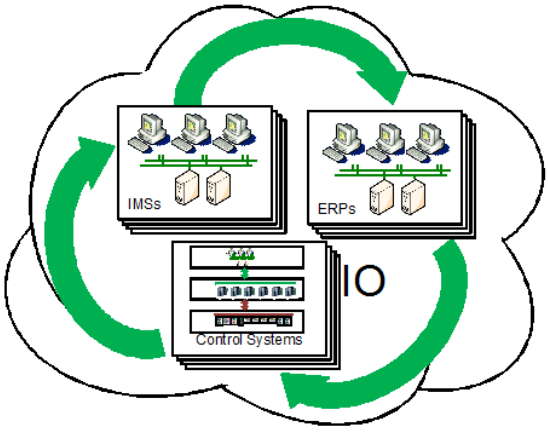


Fig. 2. Integrated Operation Information flow

OPC Unified Architecture (UA) seems to become a standard for interaction with control systems and as a mechanism for communication within a control system. This standard addresses all issues concerning secure data transport and data access security for interaction with control systems. Moreover, the standard would enforce and object (component) based approach with respect to data exchange with control systems. The OPC UA uses, the open and standardized service-oriented architecture (SOA), XML and Web Services based mechanism for data transport between computers and provide a single, coherent, object data model [5,6].

Additionally, there are some researches in using OPC UA on other levels: In the field of process automation, in integration with electronic device description language (EDDL) [7] (see Fig. 3). This needs only implementation of a TCP/IP stack in a connected device. TCP/IP is used as standard in network communication and it is independent of operating system. Nowadays it is also very popular in industry level and many PLCs, sensors and actuator have this functionality built in [8].

Integration with other applications like MES, ERP, HMI. There are different systems with big amount of data, they do not guarantee real-time processing and even quick confirmation of transmission.

Unfortunately, there is currently no standard available for communication between the other systems (IMS/ERP) comprising the Integrated Operation environment. OPC UA does not address this type of communication, and the most promising (and emerging?) technology in this area is Message Queuing (MQ). Message queuing offers secure publisher-subscriber communication suitable for Integrated Operation information exchange. MQ would also facilitate object oriented clustering of information from any information source in the Integrated Operation system by using structured naming convention in the message queue, offering some of the benefits from OPC UA technology.

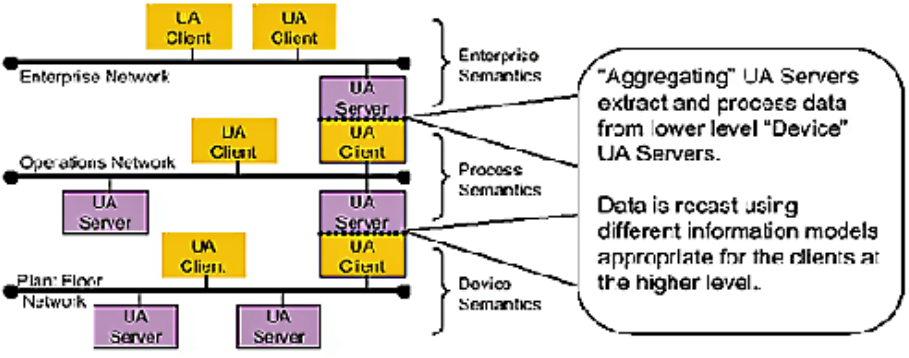


Fig. 3. Aggregating UA Servers in different levels [9]

Assuming OPC UA to be used by all suppliers of Control System technology a bridging system between the OPC UA based Control Systems and the information handling systems using Message Queuing techniques is necessary. The natural interface between the systems would be on the Control Systems HMI layer, but other control system layers could also be interfaced. The Figure 4 shows a system "bridging" the IMS/ERP layer with the control systems HMI layer. It should be noted that the control system is the most important information publishing system in an IO environment.

There is an OPC UA to MQ bridge converting information between the control systems and the other systems comprising an Integrated operation environment. Dependent on the nature of the IMS and ERP systems bridging some of these systems to a Message Queue might also be required. Such a system would enable bidirectional communication between all components in an Integrated Environment independent of the various systems "flavor" encountered.

It should also be noted that if the higher layer systems (e.g. IMS and ERP systems) supports OPC UA the bridge could also be used to interface these to the message queue. This principle is shown in the figure 5.

There is an IMS system supporting OPC, and an OPC UA to MQ bridge converting information from this IMS system to the MQ system.

The bridge could, of course, support other interfaces than OPC UA. Currently such a bridge is not commercially available, but it would be relatively simple to create a bridge supporting for instance OPC UA and the market leading IBM WebSphere MQ since both systems offers APIs which can conveniently be used within a variety of programming environments [10].

Such a bridge could, and should really, be seen as a temporarily solution to the communication challenges in an Integrated Operation system. What is really called for is standardization of communication mechanisms in the IMS and ERP layers, much the same way as OPC UA standardizes communication between and within control systems.

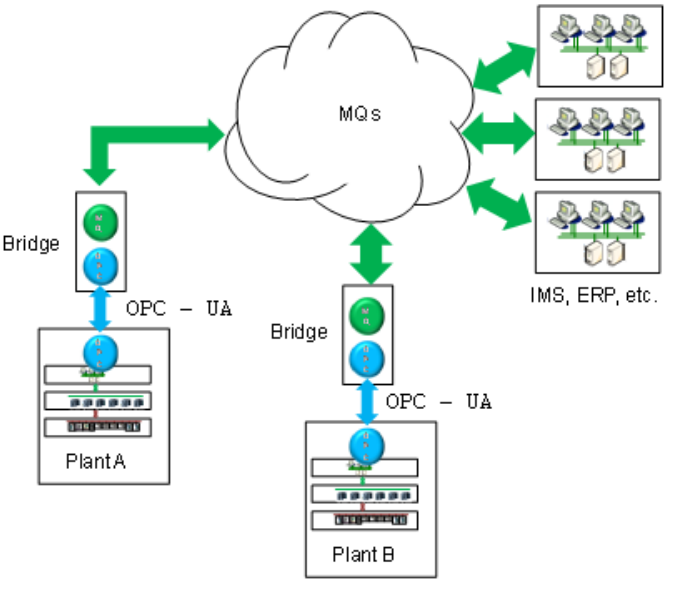


Fig. 4. OPC UA and MQ bridge

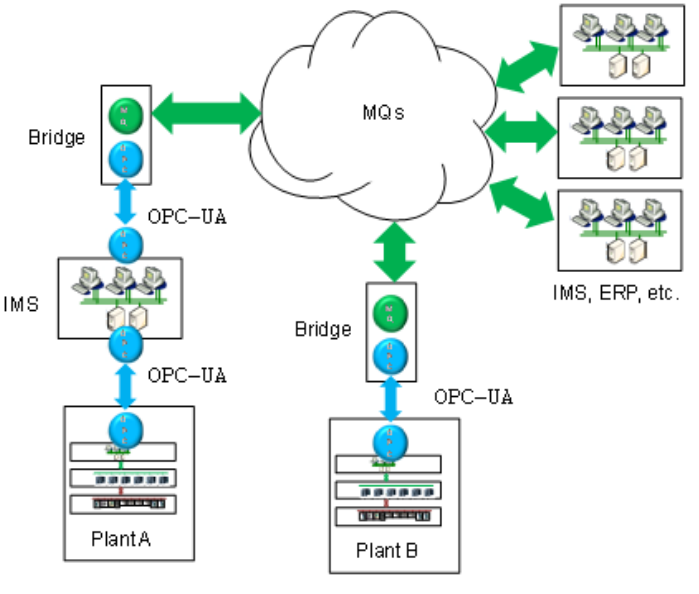


Fig. 5. IMS and MQ bridge

4 Conclusions

The idea of using OPC UA communication standard gives some new possibilities. One of the biggest is unifying all the data in one memory model. Instead of sending two or three transmissions (value, value changing, alarm, event) it is enough to send one object with all the parameters. In this way the client receives consistent data prepared in convenient format on server's side. Next thing is that the model makes it possible to get better resolution of Time Stamping. It can be realized in changing all cyclical exchanges into client-server model (Service Oriented Architecture). Another advantage of such communication model is the data transfer reduction on underlying communication network also. It is particularly important that the idea of OPC UA can be used on all levels of industry processes, from sensors through SCADA systems to MES/ERP systems. It gives possibility of complete unification of data. But here it is necessary to have methods to translate and adjust data structure between different equipment and communication levels.

References

1. Koronios, A., Nastasie, D., Chanana, V., Haider, A.: Integration Through Standards – An Overview Of International Standards For Engineering Asset Management. In: Fourth International Conference on Condition Monitoring, Harrogate, United Kingdom, June 11–14 (2007)
2. ISO 15926, ISA-S88, ISA-S95 standards
3. Kari, A., Haaland, T., Chunming, R.: Towards Dataintegration from WITSML to ISO 15926. In: Sandnes, F.E., Zhang, Y., Rong, C., Yang, L.T., Ma, J. (eds.) UIC 2008. LNCS, vol. 5061, pp. 626–635. Springer, Heidelberg (2008)
4. Batres, R., West, M., Leal, D., Price, D., Masaki, K., Shimada, Y., Fuchino, T., Naka, Y.: An Upper Ontology based on ISO 15926. In: Selected Papers from the 15th European Symposium on Computer Aided Process Engineering held in Barcelona, Spain, May 29–June 1. Elsevier, Amsterdam (2005, 2006)
5. Leitner, S.H., Mahnke, W.: OPC UA – Service-oriented Architecture for Industrial Applications. ABB Corporate Research Center
6. Information Model
7. Huang, R., Liu, F.: Research on OPC UA based on Electronic Device Description. In: ICIEA 2008. 3rd IEEE Conference on Industrial Electronics and Applications (2008)
8. Hannelius, T., Shroff, M., Tuominen, P.: Embedding OPC Unified Architecture. In: Automaatio XVIII Seminari, Helsinki (2009)
9. Burke, T.J.: The magic of OPC Unified Architecture. Industrial Ethernet Book Issue (30/99)
10. Davies, S., Broadhurst, P.: WebSphere MQ V6 Fundamentals, IBM (November 2005)
11. Gregovic, R.M., Foreman, R., Forrester, D., Carroll, C.: A Common Approach to Accessing Real-Time Operations Data: Introducing Service Oriented Architecture to E&P. In: SPE Annual Technical Conference and Exhibition, Dallas, Texas, USA, October 9-12, SPE 96441, Society of Petroleum Engineers (2005)

Cloud Service Solving N-Body Problem Based on Windows Azure Platform

Dariusz Rafał Augustyn and Łukasz Warchał

Institute of Informatics
Silesian Technical University
Akademicka 16, 44–100 Gliwice, Poland
{draugustyn,lukasz.warchal}@polsl.pl

Abstract. This paper shows how to use cloud computing to solve N-body problem. It presents an idea and implementation of cloud service based on Windows Azure Platform. Clients can access cloud service via Internet over HTTP protocol. They create computation tasks supplying simulation parameters such as number of steps, time step and XML file with body definitions (initial position, mass and velocity). Presented solution uses Barnes-Hut Algorithm (based on adaptive oct tree) to reduce computation complexity form $N \times N$ to $N \log N$. All body interactions are computed in parallel, on worker nodes in cloud.

Keywords: distributed processing, cloud service, Microsoft Windows Azure Platform, continuous dynamical systems simulation, N-body problem.

1 Introduction

N-body simulation methods are fundamental in domains of modeling of complex systems e.g. astrophysics (gravity interactions), molecular dynamics (electrostatics – Biot-Savart and van der Waals interaction), computer graphics (radiocity metods - computer image generation). Modern N-body simulation applications allow to observe an evolution of complex N bodies systems for huge values of N.

The most simple and naive method is based on an assumption that bodies interact with each other. For this method so-called all-paired one, a complexity of computational work scales asymptotically as N^2.

There are many methods which decrease this complexity. Most of them use one of two approaches: Barnes-Hut algorithm [1] or FMM (Fast Multipole Method) [2]. Those methods are efficient but they give approximate results.

In this paper besides all-paired method the Barens-Hunt algorithm is considered. In the BH method the 2-dimensional space or 3-dimensional one is divided using an adaptive quad tree or oct one (both called BH-tree). Calculation of interaction (i.e. a force vector) for a selected body is preformed using only BH-tree's nodes (no calculating components of the force for all $N - 1$ bodies). Because of the height BH-tree (significant less than N) this reduces the problem complexity so it can be computed in time as $O(N \log N)$.

A. Kwiecień, P. Gaj, and P. Stera (Eds.): CN 2010, CCIS 79, pp. 84–95, 2010.

Because a task of calculation for a selected single body is independent of other tasks, all those tasks can be executed separately. This gives a possibility of efficient parallel implementations of N-body simulation methods using e.g. distributed calculation environments.

There was many approaches to solve problem of parallel execution of N-body simulation (e.g. [1,2,3,4,5,6,7]) but this paper shows the application of cloud computing by using the newest Microsoft distributing calculation environment – the Windows Azure Platform [8].

Nowadays cloud computing becomes more and more popular. Growing number of developers deploy their applications in Internet-accessible data centers and never again worry about infrastructure, availability or data security. Windows Azure Platform gives flexible environment for creating service based applications accessed by large number of users via Internet. This solution can be easily scale to fit particular requirements. It offers Data Storage capabilities that allow to handle huge binary data files and tables containing billions of entities (with simple REST [12] interfaces). Communication between different application components is realized by queues (also accessed RESTfully). Windows Azure allows to focus on solving main problem rather than building complex infrastructure (software and hardware) for application.

2 Applied N-Body Methods – The Theoretical Background

N-bodies problem will be explained by presenting some example of evolution an astrophysical star system in the 3-dimensional space – a spiral galaxy.

The system can be described as a set of bodies. m – mass, x coordinate, y one, z one, v_x – x-component of velocity vector, v_y – y-component of velocity, v_z – z-component of velocity are 7 properties of each body form the given system.

The movement of i body in t_{k+1} (the next moment of time) using values in t_k (the current moment) can be approximately described by following obvious forms (based on the second Newton's law of motion):

$$t_{i,k+1} = t_{i,k} + dt, \tag{1}$$

$$\begin{bmatrix} x_{i,k+1} \\ y_{i,k+1} \\ z_{i,k+1} \end{bmatrix} = \begin{bmatrix} x_{i,k} \\ y_{i,k} \\ z_{i,k} \end{bmatrix} + \begin{bmatrix} v_{x,i,k} \\ v_{y,i,k} \\ v_{z,i,k} \end{bmatrix} dt, \tag{2}$$

$$\begin{bmatrix} v_{x,i,k+1} \\ v_{y,i,k+1} \\ v_{z,i,k+1} \end{bmatrix} = \begin{bmatrix} v_{x,i,k} \\ v_{y,i,k} \\ v_{z,i,k} \end{bmatrix} + \frac{1}{m_i} \begin{bmatrix} F_{x,i,k} \\ F_{y,i,k} \\ F_{z,i,k} \end{bmatrix} dt \tag{3}$$

where $[F_{x,i}\ F_{y,i}\ F_{z,i}]^T = \boldsymbol{F_i}$ is the resultant force vector.

For selected i body the resultant force $\boldsymbol{F_i}$ is calculated using gravity interactions with all $N-1$ remaining bodies.

The gravity interaction between i body and j one can by expressed as follows (the Newton's law of gravitation):

$$\boldsymbol{F_{ij}} = \begin{bmatrix} F_{x,ij} \\ F_{y,ij} \\ F_{z,ij} \end{bmatrix} = G\frac{m_i m_j}{r_{ij}^3} \begin{bmatrix} x_j - x_i \\ y_j - y_i \\ z_j - z_i \end{bmatrix} \quad , \tag{4}$$

$$r_{ij}^3 = (x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2 \tag{5}$$

where G is the gravitational constant and r_{ij} is the distance between i body and j one.

So the resultant force vector for single i body can be obtained as follows:

$$\boldsymbol{F_i} = \sum_{i \neq j} \boldsymbol{F_{ij}} \quad . \tag{6}$$

Based on rules presented above (Equations (1)–(6)) the application for simulating gravity interactions was designed and developed. For example it was used for obtaining the simulation result of an evolution of a small spiral galaxy containing 100 bodies. A startup bodies configuration (position and velocity vectors of bodies) is shown on Fig. 1. The final bodies positions after 20 000 steps of the simulation (with $dt = 0.001$) is shown on Figs. 2 and 3. The system is unstable – some bodies "escaped from the galaxy" (Fig. 2). But most of bodies remained in the space region determined at startup (Fig. 3). Startup positions, final ones and trajectories of bodies are presented on Fig. 3.

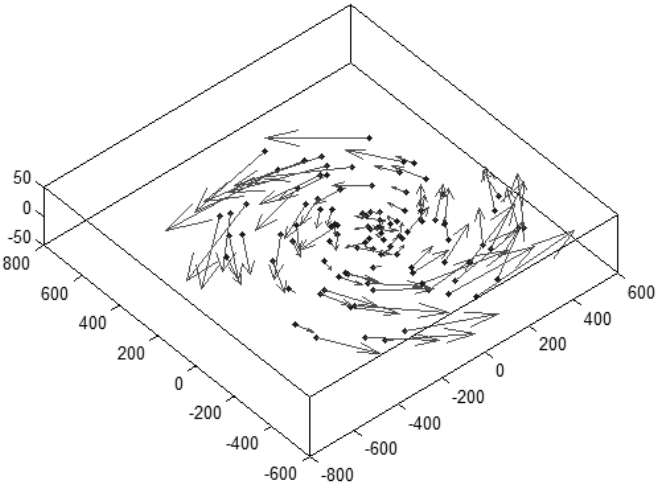


Fig. 1. View of startup position and velocity distribution for the sample spiral galaxy

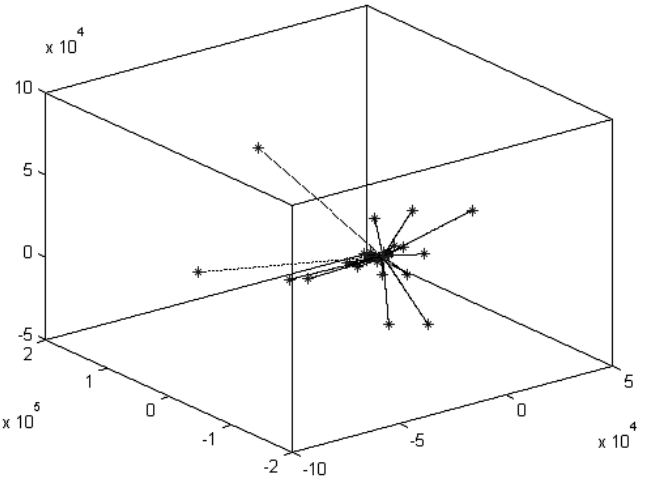


Fig. 2. Simulation results: escape of some bodies form the sample spiralgalaxy; final positions denotes by asterisks

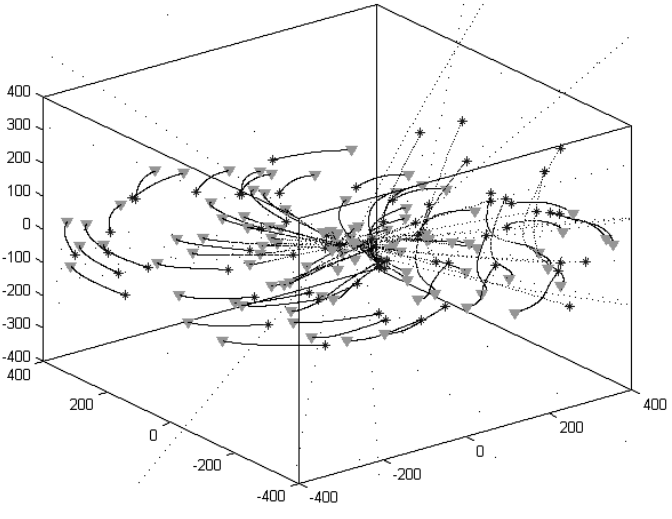


Fig. 3. Simulation results: bodies startup positions (triangles), final positions (asterisks) and trajectories in space located near the center of the sample galaxy

Because of efficiency reason the known approach [1] based on hierarchical tree data structures was considered too. For simplicity of presentation, the 2-dimensional case with a quad tree is shown at the first. The sample set of 5 bodies A,. . . ,D is located in nested squares (Fig. 4b). The space decomposition is finished when only one body is placed in a square. The process of this space division is illustrated on Fig. 4a.

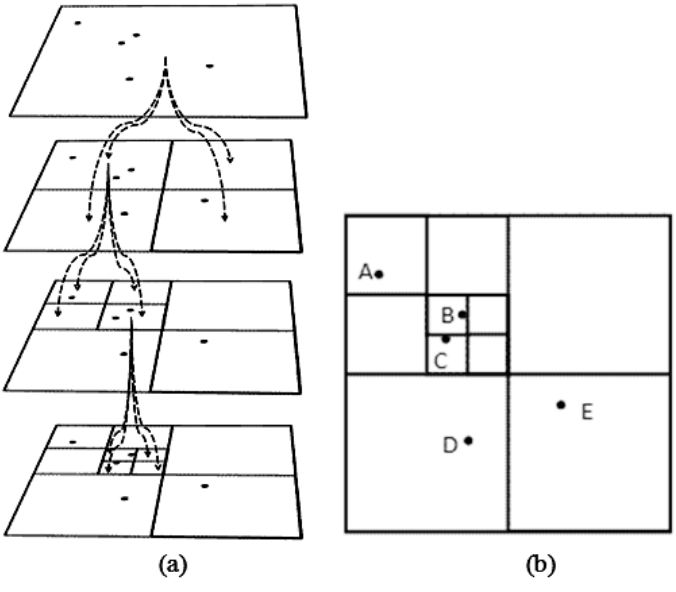


Fig. 4. Division of 2-dimensional space for sample set of bodies (division process – (a), result of division – (b))

The structure of sample adaptive quad tree is shown on Fig. 5a. Each tree node is identified by unique binary code. The method of coding nodes at every level of the tree is presented on Fig. 5b.

A similar method can be applied for a division of the 3-dimenasional space using an oct tree. Each node of the oct tree represents a cube. Each cube is decomposed on eight sub-cubes at the lower level of the tree. The method of identifying nodes in oct tree is shown on Fig. 6.

Advantages of the BH method results from applying some approximate method of calculation the resultant force. If a given body is distant enough from a node (square or cube) we can assume some simplification [1]. Let d denotes the edge of a node (side of a square or edge of a cube). Let r denotes the distance between the given body and center of mass (CM) of bodies subsystem nested in a node (see Fig. 7).

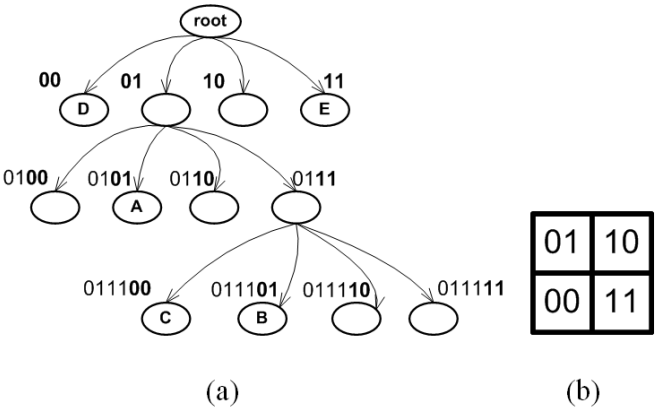


Fig. 5. Adaptive quad tree for sample bodies set with coded nodes – (a). Method of binary clockwise coding of quad tree's nodes – (b)

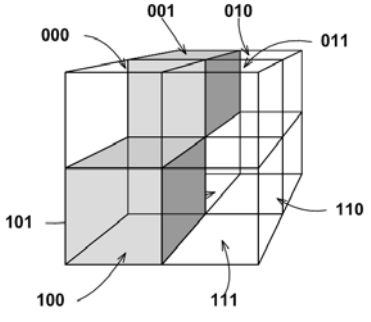


Fig. 6. Method of coding sub-cubes in the oct tree-based process of decomposition of 3-dimensional space

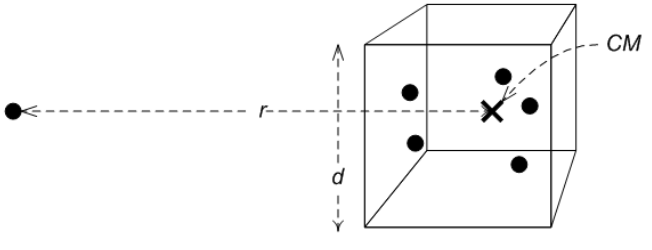


Fig. 7. Calculation of approximate resultant force between a single body and all bodies nested in a node

If formula presented below is satisfied (the long distant condition):

$$\frac{d}{r} < \theta \tag{7}$$

we can approximate the interaction between the given body and the subsystem nested in the node, by calculation of the resultant force as an interaction between only 2 bodies: the given body and a substitutionary body (with the mass equals to total mass of bodies nested in the node).

The θ is a parameter of the BH algorithm which determines an accuracy of the force approximation. It determines also the efficiency of the algorithm. Commonly values of θ belong to $[0.7, 1]$.

The method of calculation of the resultant force between the given i body and j node can be described by the following pseudocode:

```
function ResultantForce(i,j) {
    if (j is a leaf )
        return (force between i body and j one)
    elseif ( i body is distant enough from j node)
        return (approximate force between i body and j node)
    else {
        foreach ( l in sub-nodes(j)){
           if (l is not empty)
              aggregate ResultantForce(i,l)
        }
        return (aggregated resultant force)
    }
}
```

Calculation the interaction between a single i body and all bodies of system is equivalent to invoking `ResultantForce (i, root of the tree)`.

3 Azure Platform Overview

Microsoft Azure Platform is an environment for running cloud applications and services. It has three main parts: *Compute* service, in which applications run, *Storage* service, which provides data storage and *Fabric* that manages and integrates all of them.

In Windows Azure each application can be made of two different instance types: Web role or/and Worker role. First one is responsible for interaction with end users. It can accept incoming HTTP or HTTPS requests. Typically, this is an ASP.NET web application or some kind of Web Service exposed by WCF [13]. Worker role instances are mainly considered as background processing components communicating with other workers or web role instances via queues.

Each Worker or Web instance runs inside its own Virtual Machine thus it is completely isolated. Number of instances of particular type is set at the stage of configuring application and can be change at runtime to avoid overloading. Each application needs to store its data in secure, reliable place, so Windows Azure

provides mechanisms for storing and retrieving data in cloud-based environment. There are three kinds of Windows Azure storage: blobs, tables and queues, all accessed over standard HTTP protocol (using GET, PUT, DELETE methods).

3.1 Azure Table Storage

Azure Table Storage is a structured storage that supports scalable tables, that can contain large number of entities (rows) and terabytes of data [9]. It is efficiently scaling out by spreading to many servers as traffic grows.

Entity stored in a Table contains a set of user defined Properties (columns) and a PartitionKey and RowKey Properties that are used as primary key (both strings). PartitionKey is also used in scaling out. If there are many queries for entities from the same partition (the same PartitionKey), load balancing algorithm implemented in Windows Azure can activate dedicated node for serving only this partition, as shown on Fig. 8. Choosing proper way of creating partition key values is necessary to achieve good scalability.

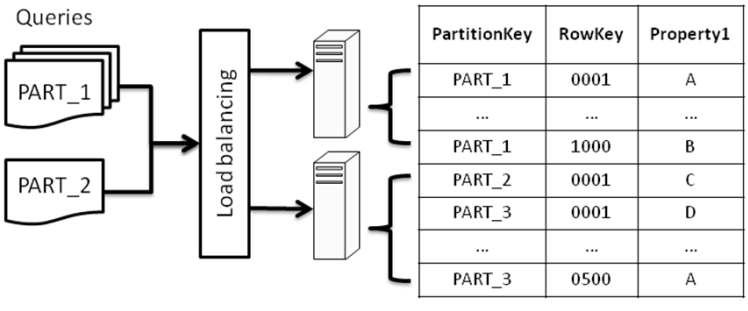


Fig. 8. Scaling out with PartitionKey

All Tables in Windows Azure Table Storage have flexible schema, so two entities stored in one table can have different set of properties. This allows to avoid problems when entity definition is extended.

3.2 Azure Blob Storage

As mention above Azure Table Storage allows to keep terabytes of data in tables containing entities. But each entity can have maximum 255 different properties and only 1MB of combined size. Fortunately Windows Azure provides another mechanism for storing large binary data – Azure Blob Storage [11].

In Azure Blob each stored object can have up to 50 GB, it is replicated at least 3 times and strong consistency is provided. Blobs can be accessed from any place in any time using HTTP/REST protocol. The URI for particular object has following form:

```
http://<account>.blob.core.windows.net/<container>/<blobname>
```

where <account> is an account name in Windows Azure service, <container> is a name of container grouping a set of blobs (i.e. movies, pictures, etc.), and <blobname> is name of requested object.

3.3 Azure Queues

Azure Queues provides efficient and reliable message delivering mechanism [10]. It allows separated Worker and Web roles to communicate and exchange some data. Monitoring queue length can help deciding to increase number of processing instances.

All operations on queues can be done using HTTP REST protocol. An example of a GET request listing messages in a queue is presented below:

```
http://sampleAccount.queue.core.windows.net/nbodyqueue/messages?
timeout=30&numofmessages=32&peekonly=true
```

4 N-Body Cloud Service Architecture

Cloud service described in this paper is composed of Web and Worker role instances communicating over queues and storing data in Data Storage (Table and Blob) as shown in Fig. 9.

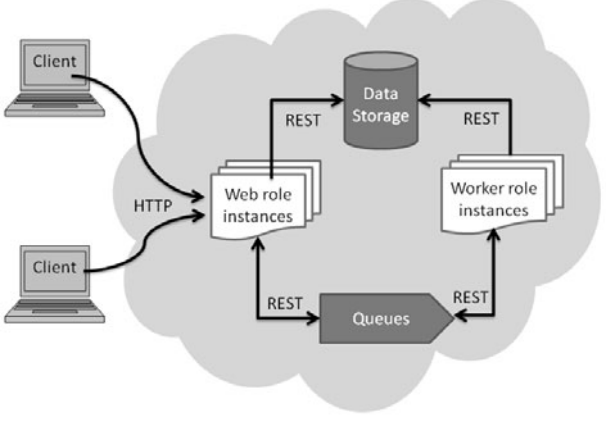


Fig. 9. Cloud Service Architecture Overview

Web role is a simple ASP.NET application. On main web page user defines simulation parameters such as: number of steps, time step, θ value and uploads XML file with bodies definitions (initial position, body mass and velocity), which is saved in Azure Blob Storage with an unique file name. After starting simulation, web application creates a *Task* entity (containing user supplied parameters) and stores it in *Tasks* table in Azure Table Storage. This entity is identified by

unique string (Task Identifier) which is also sent back to client. After this, web application sends message to queue that worker instances can consume and start computation.

Worker role instances are responsible for some background processing and computation. Each instance reads a message from queue and depending of its type performs some task. The single message is a XML serialized object of class *NBodyWorkItem*, where value of *Type* property tells worker what to do. There are 4 main tasks that worker instances perform.

The presented cloud service works according to algorithm shown on an activity diagram from Fig. 10.

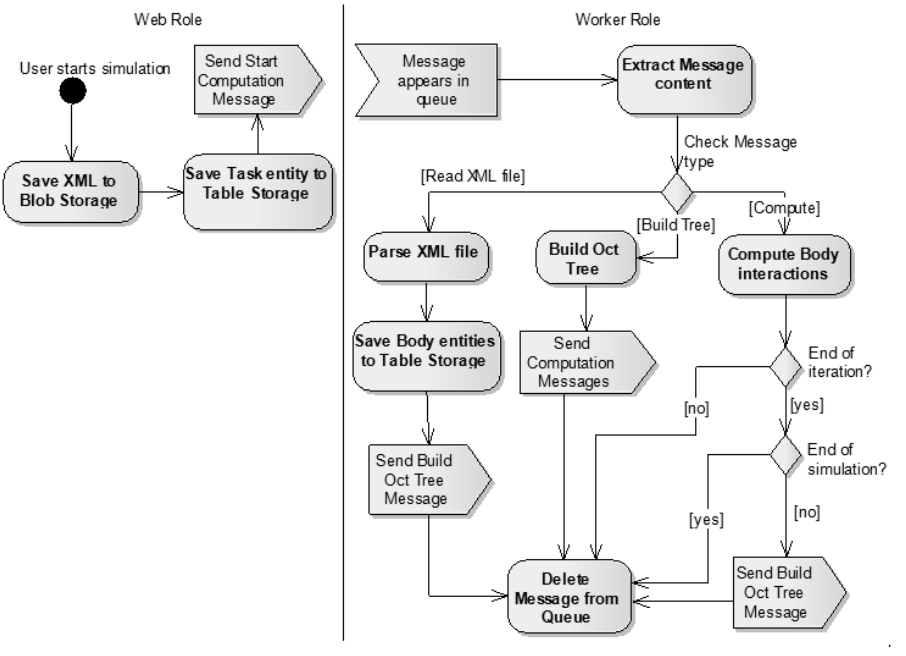


Fig. 10. Activity diagram presenting algorithm of N-body cloud service

When simulation starts, the first task is dedicated to parse uploaded by the user XML file. It was saved (by web application) in Azure Blob Storage. One of worker instances reads this file (its name is passed as a part of queue message), parses it and creates all *Body* entities that are persisted in Azure Table Storage in *Bodies* table. Task Identifier is used as a PartitionKey, so all *Body* entities from one simulation are in the same partition so they can be accessed in an efficient way. RowKey is a concatenation of iteration number (1 in this case) and some generated body identifier, so it is possible to distinguish between entities from different iterations.

Next one of worker instances is responsible for building oct tree. Each node in a tree is an *OctTreeNode* entity and it is stored in Azure Table Storage in

OctTreeNodes table. Oct tree is a hierarchical data structure but in this case it is persisted in flat table. To provide efficient way to access child nodes from parent one, special nodes identifying was implemented. This method of node coding is different from the one described in previous section because it is adapted to Azure's specificity. Each node has an unique identifier (RowKey property) which is a number converted to 20 characters length string padded with "0" to its left. Root has always RowKey equals to "0". Child nodes identifiers are generated according to this procedure written in C#:

```
for (int i = 1; i <= 8; i++) {
 // ...
 newId = String.Format("{0:D20}",Convert.ToInt64(parentId+i.ToString()));
 // ...
}
```

Such node coding makes that preparing LINQ query [14] against Azure Table Storage that finds all child nodes of given parent node is simple:

```
Int64 value = Convert.ToInt64(node.RowKey + "1");
string startOfLevel = String.Format("{0:D20}", value);
value += 7;
string endOfLevel = String.Format("{0:D20}", value);

var q = from n in DataContext.OctTreeNodes
        where n.PartitionKey == TaskId &&
              n.RowKey.CompareTo(startOfLevel) >= 0 &&
              n.RowKey.CompareTo(endOfLevel) <= 0
        select n;
```

When oct tree is ready, Worker sends N messages (requests of computation tasks) to queue, one for each body. Other workers consume one of this messages and perform computation of body interaction using the oct tree (in parallel). It is obvious that increasing number of Workers (up to N) reduces the time needed to calculate particular iteration.

When new position and velocity of a body is known, new *Body* entity with this values is created and stored in *Bodies* table. When all bodies have been processed next iteration is started in similar order, unless it was the last one.

In the presented solution *Bodies* table contains for each body its "snapshots" from any iteration. This allows a user to download (as XML file) any step of simulation. The result XML file can be used in future as an input file for another simulation (or it can be presented to user as it was shown on Fig. 1).

5 Conclusions

This paper presents an idea and implementation of cloud service for N-body problem running on Windows Azure Platform. It covers various Windows Azure aspects and capabilities. Azure Storage is described in details. Presented service solves N-Body problem using all-pairs method or adaptive oct tree (Barnes-Hut Algorithm). Interaction for single body is now computed in parallel by Worker

instances, however building oct tree is done by only one of them. Implementing parallel tree construction algorithm can obliviously help improving overall efficiency.

During the tests, we found that accessing Azure Storage is time expensive (it is foreseeable, because it is done with HTTP REST). Better efficiency can be achieved by reducing number of storage access, i.e. during body interaction calculation, each Worker can process more than one body at one time (one storage access returning many *Body* entities). For this reason we provided two implementations of building oct tree procedure. If number of bodies is not large, whole tree is first constructed in operating memory and then persisted in Table Storage in a single request. If number of bodies is very large, each successive oct tree nodes are immediately persisted to storage.

References

1. Barnes, J., Hut, P.: A hierarchical $O(N \log N)$ force calculation algorithm. Nature 324 (1986)
2. Greengard, L., Rokhlin, V.: A Fast Algorithm for Particle Simulations. Journal of Computational Physics 73(325) (1987)
3. Warren, M.S., Salmon, J.K.: A parallel hashed Oct-Tree N-body algorithm. In: Proceedings of the 1993 ACM/IEEE conference on Supercomputing, pp. 12–21. ACM, New York (1993)
4. Dubinski, J.: A Parallel Tree Code, Santa Cruz (1996)
5. Cruza, F.A., Barba, L.A., Knepley, M.G.: Fast Multipole Method for particle interactions: an open source parallel library component. In: The 20th Parallel Computational Fluid Dynamics conference, Lyon (2008)
6. Izaguirre, J.A., Hampton, S.S., Matthey, T.: Parallel multigrid summation for the N-body problem. Journal of Parallel and Distributed Computing 65(8), 949–962 (2005)
7. Nyland, L., Harris, M.: Fast N-Body Simulation with CUDA. In: GPU Gems, vol. 3. Addison-Wesley Professional, Reading (2007)
8. Chappell, D.: Introducing Windows Azure. David Chappell & Associates (2009)
9. Haridas, J., Nilakantan, N., Calder, B.: Windows Azure Table (2008)
10. Calder, B.: Windows Azure Queue (2008)
11. Calder, B., Wang, T., Mainali, S., Wu, J.: Windows Azure Blob (2009)
12. Richardson, L., Ruby, S.: Restful Web Services. O'Reilly Media, Sebastopol (2007)
13. Peiris, C., Mulder, D., Cicoria, S., Bahree, A., Pathak, N.: Pro WCF: Practical Microsoft SOA Implementation. Springer, New York (2007)
14. Rattz, J.C.: Pro LINQ: Language Integrated Query in C# 2008 (Windows.Net). Apress (2008)

An Experimental Wireless Mesh Network Node Based on AVR ATmega16 Microcontroller and RFM12B Radio Module

Remigiusz Olejnik

West Pomeranian University of Technology, Szczecin
Faculty of Computer Science and Information Systems
ul. Żołnierska 49, 71-210 Szczecin, Poland
r.olejnik@ieee.org

Abstract. The paper describes a simple node designed to act as part of an experimental wireless mesh network. The node is based on Atmel AVR ATmega16 microcontroller and Hope Microelectronics RFM12B radio module. Along with technical details of the node, the network layer (addressing scheme and routing algorithms) is presented. Fundamental principles on wireless mesh networks are also presented. Experimental results summarize the paper and prove limited usability of the project.

Keywords: wireless mesh networks, routing protocol, network layer.

1 Introduction

Wireless mesh networks (WMN) are quite new way of building data communication systems. It's main goal is to achieve higher level of reliability, which is done mainly through special routing protocols. WMNs are still in development stage – there are not enough simple ways of building such networks for masses, due to lack of software support [1]. The field seems to be still in experimental stage. Following that direction, this paper presents an Experimental Wireless Mesh Network (EWMN) node: its hardware and software architecture, including network layer.

The paper consists of three main parts. First part presents basics of wireless mesh networks. Second part describes details of the EWMN node: its hardware and software, including network addressing scheme and routing protocol. Third part shows experimental results. Summary containing final conclusions and possible future works ends the paper.

2 Wireless Mesh Networks' Basics

2.1 Wireless Mesh Networks

Unlike in most common network topologies – bus or star – in mesh topology data is forwarded between the nodes, until it will reach destination, without

A. Kwiecień, P. Gaj, and P. Stera (Eds.): CN 2010, CCIS 79, pp. 96–105, 2010.

help of central node (hub/switch/access point). Such communication is direct and depends on radio range of the nodes – if destination node is not within radio range, it is forwarded according to routing tables. The nodes of mesh networks act as routing devices, which decide about path of data [2]. An example of mesh network is depicted on Fig. 1.

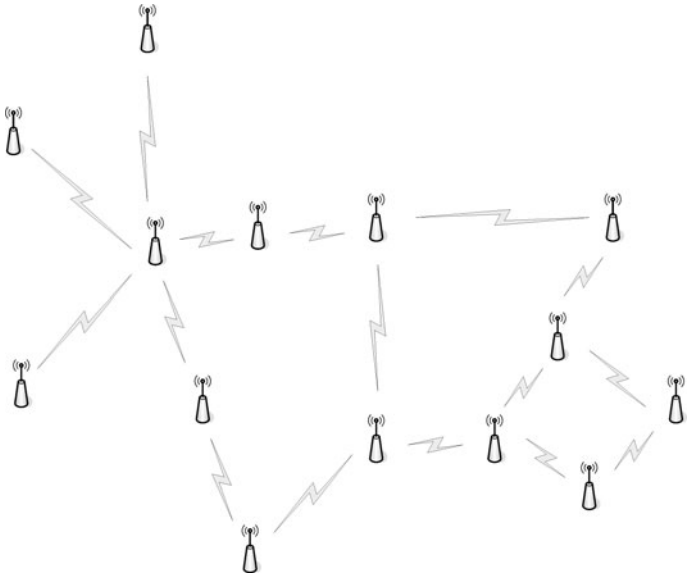


Fig. 1. An example of mesh network

Advantages of mesh networks are numerous: higher redundancy, easy reconfiguration capability, flexibility and easy upgrade possibility. Dynamic reconfiguration and network self-healing may be useful in case of network fail or nodes' location change. The network have to retain its main goal – forwarding data between pairs of source and destination nodes.

2.2 Routing in Wireless Mesh Networks

Performace of wireless mesh networks highly depends on routing techniques. There are many known approaches, that can be classified either as proactive (they maintain paths for every pair of nodes) or reactive (the route paths are discovered in "on demand" way) [2,3]. In this paper one of the modified traditional algorithms is used – B.A.T.M.A.N. (**B**etter **A**pproach **T**o **M**obile **A**d-hoc **N**etworking) [4,5] behaviour is inspired by termites, which leave feromonal traces along optimal paths to the food. None of the nodes keep full topological information about the network. Every node has only information about its neighbouring nodes, which could handle data for destination node, associated

with some kind of metric. Active neighbouring nodes are discovered periodically by sending `HELLO` datagram in broadcast manner.

3 An Experimental Wireless Mesh Network Node

3.1 Architecture of the Node

Experimental Wireless Mesh Network (EWMN) has been built of quite simple nodes. The node's architecture is based on cheap and wide known Atmel AVR ATmega16 microcontroller [6] as CPU. Communication between the nodes is provided in one of ISM (industrial, scientific, medical) bands with Hope Microelectronics RFM12B [7] radio module. External communication with PC is done via MAX232 chip and serial port of the computer. Schematic of the node is shown on Fig. 2 and physical design of the node is shown on Fig. 3.

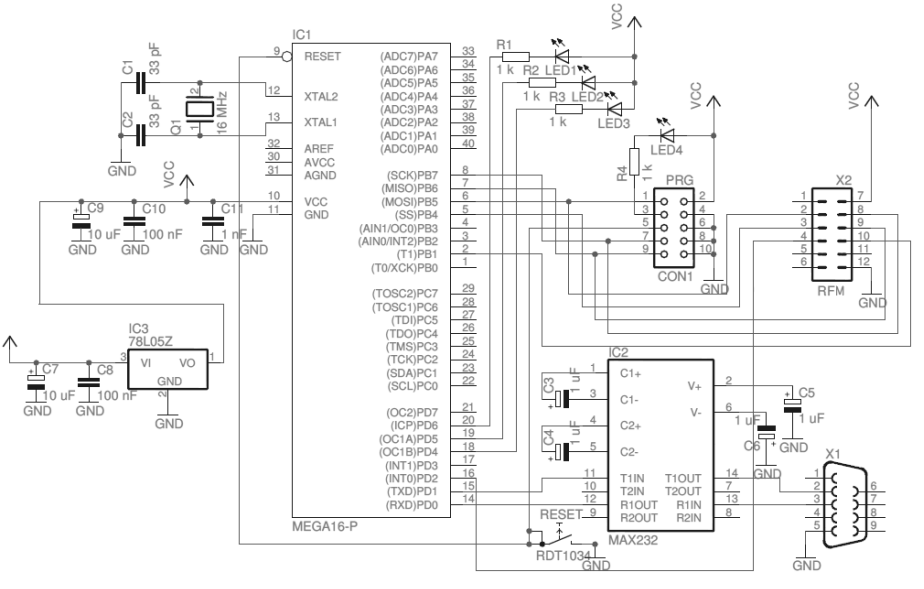


Fig. 2. EWMN node schematic

3.2 Network Layer

Addressing Scheme. Addresses of the nodes have length of one octet. Excluding `0x00` (empty address/address not assigned) and `0xFF` (broadcast address) capacity of address pool is 254. The node has random address after initialization. Moreover default destination address value is set to broadcast address `0xFF`. Broadcast mode is always local, within radio range of the node, so the datagrams are not forwarded by neighbours to the rest of the network.

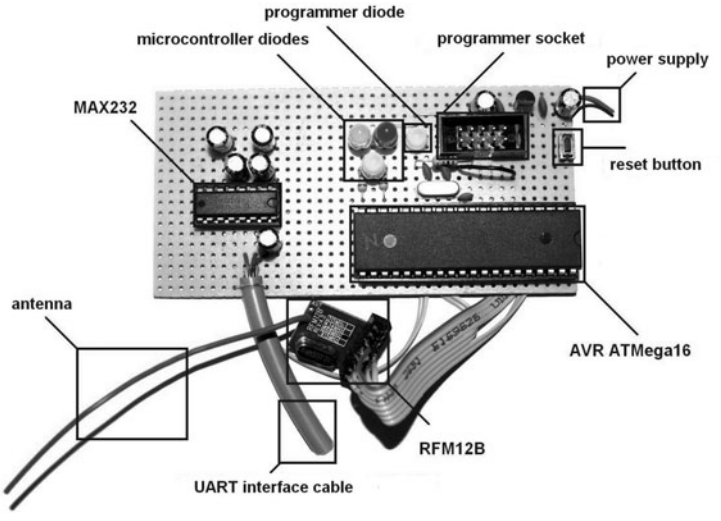


Fig. 3. EWMN node physical design

Datagram Specification. Datagrams have fixed length of 64 bits. The header is 24 bits long and data part is 40 bits long. Structure of the datagram is depicted on Fig. 4.

type	src address	dst address	data	data	data	data	data

Fig. 4. Datagram structure

The header consists of three fields, each of 8 bits length:

- **type** – with value `HELLO` for broadcast datagram or `REGULAR` in case of ordinary data;
- **source address** – with correct value in range `0x01`–`0xFE`;
- **destination address** – with correct value in range `0x01`–`0xFF`.

Broadcast datagram `HELLO` uses two data fields carrying originator address and sequential number. Three remaining data fields are unusued at this time. Structure of broadcast datagram is depicted on Fig. 5.

HELLO	src address	dst address	originator	seq no.	*	*	*

Fig. 5. `HELLO` datagram structure

Routing Algorithm. Routing algorithm is based on the B.A.T.M.A.N. algorithm [4,5] and implements addressing, splitting data into datagrams, broadcasting and data forwarding between the nodes to provide routing routines. Key concept of the B.A.T.M.A.N. algorithm is to inform other nodes of node's presence in cyclical manner. `HELLO` datagrams allow building a routing table at the level of every node. Creating new datagrams is rather simple, but decision mechanism used after datagram's arrival is more complex. That algorithm is shown on Fig. 6.

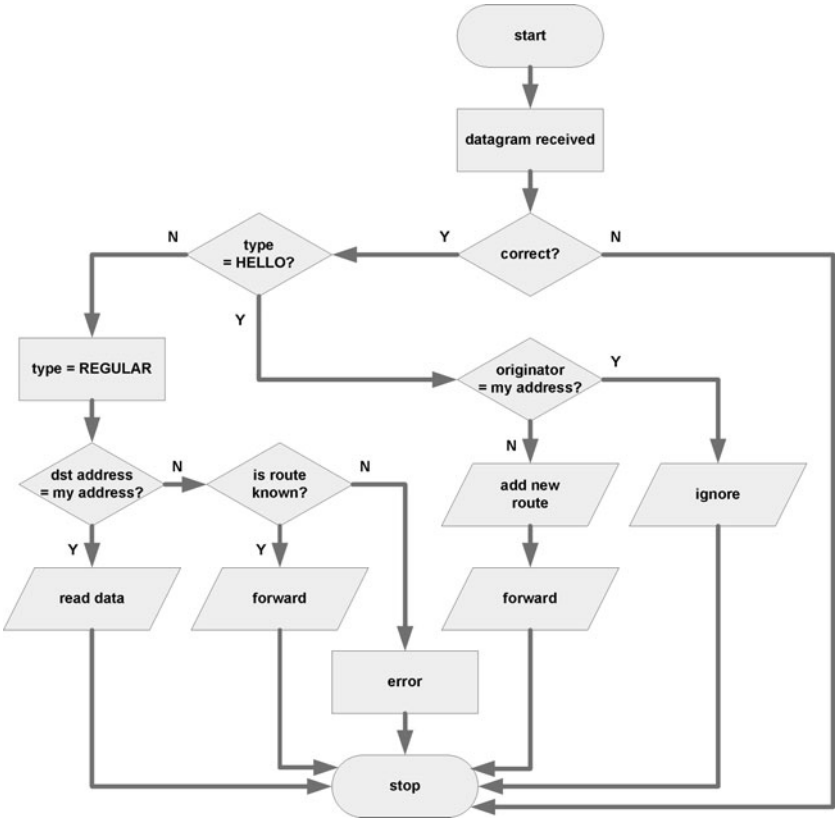


Fig. 6. Decision algorithm after datagram's arrival

At first, datagram type is checked against its correctness. If type is not valid (`HELLO` or `REGULAR`) it is discarded. In case of `REGULAR` datagram, destination address is checked. If it is equal to the node's own address, data fields are used by higher layer. If the datagram is not directed to the node, the routing table is checked to find a route to destination node. If the route is succesfully found, the datagram is forwarded using that route. In case of `HELLO` datagram, originator address is compared with the node's own address – if they are equal, datagram

is ignored because it came back to its source; if originator address is not equal to the node's own address, new route (or update) is added to the routing table and datagram is forwarded do all of the neighbouring nodes.

Routing Table. Routing table consists of four fields: destination address, intermediate node address, metric (number of hops) and sequential number. Every node knows only directions (neighbouring nodes), which could be used as routing paths. Data will be forwarded through them until it will reach its destination. The metric is used to find most reliable path and sequential number is used when updating routing tables. Example of routing table is presented in Table 1.

Table 1. Example of routing table

Destination	Direction	Metric	Seq No.
0x4C	0x4C	1	17
0x2A	0x3C	2	6
0x5D	0xFA	5	95
0xD4	0x12	3	25
0xD4	0xFA	2	104

Routing Table Update. Routing table update is done in two stages. First stage is preparing temporary table with new routes. During second stage, synchronously with sending `HELLO` datagrams, new routes are added to routing table. Algorithm of preparing new routes is presented on Fig. 7.

3.3 Program Architecture

Program code of the node has been divided into functional blocks (Fig. 8):

- **main**: main part of the program;
- **BATMAN**: routing algorithm implementation with routing tables and mechanisms;
- **RFM12B**: radio module's routines;
- **RS232**: serial interface communication;
- **datagram**: datagram coding and decoding;
- **tools**: auxiliary routines.

3.4 Terminal Interface – Communication with PC

The node is connected to external computer using node's RS232 MAX232 interface with 9600 bps 8N1 settings. Eight terminal based commands are hidden in ATMega16 program and they are combined of ~ and abbreviated commands:

- (**a**)ddress of the node (view/change),
- (**d**)estination address (view/change),
- (**r**)outing table (view),

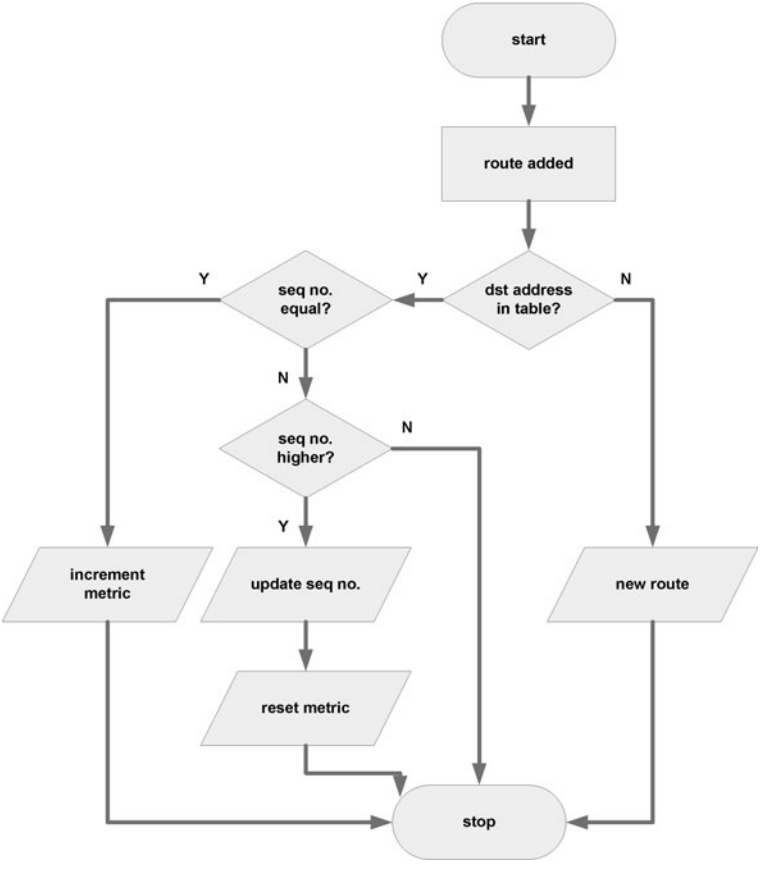


Fig. 7. Algorithm of preparing new routes for routing table

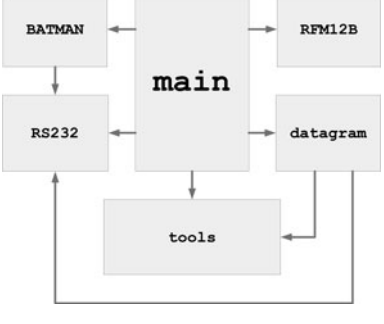


Fig. 8. Functional blocks of the node's program code

– (**n**)ode state toggle (Fully Online, Locally Online, Offline),
– (**p**)ower of TX change (min/med/max),
– (**v**)erbose mode on/off toggle,
– (**c**)lear terminal screen,
– (**e**)cho on/off toggle.

Typical information in verbose mode are as follows:

```
# Hello msg from: 0x71; origin: 0x71; seq num: 0x1B
# New Route to 0x71 via 0x71
# Sending Hello
# Regular datagram from 0x71
# Data field: dzien
# Regular datagram from 0x71
# Data field: dobry
# Sending Hello
# Hello msg from: 0x71; origin: 0x47; seq num: 0x1C
# My own hello returned
```

4 Experimental Results

After initial design of Experimental Wireless Mesh Network hardware and software, some preliminary experiments have been conducted.

At first, simple test of datagram transmission has been made. 10 datagrams has been sent 10 times to find possible data loss ratio for one node working in Fully Online state. Results are shown on Fig. 9.

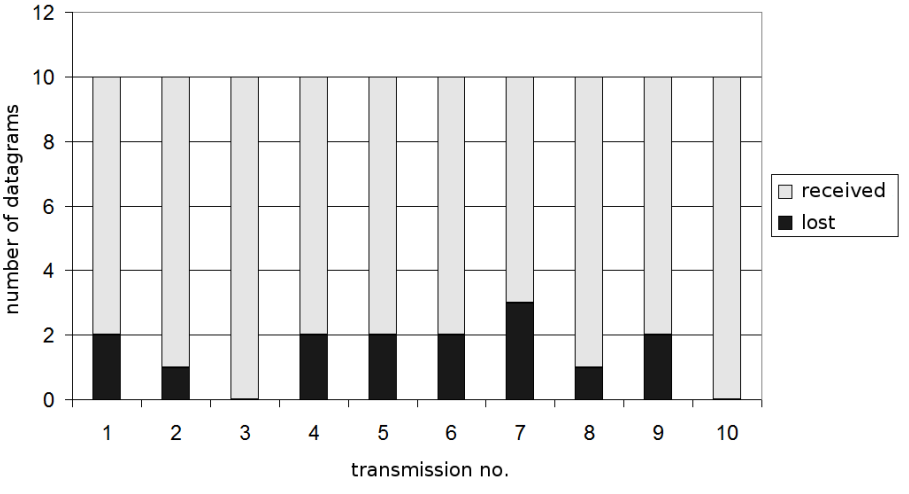


Fig. 9. Lost datagrams

Unfortunatelly, data loss ratio is rather high, about 15 %, so it can be said that data integrity is not assured at this time. More nodes mean more hops and possible multiplication of data loss ratio. Probable reason of such high datagram loss ratio is long main program loop and insufficient length of FIFO queue of RFM12B radio module (only 16 bits long). When the node had been in Locally Online state, data loss ratio was measured at 1 % level. Possible solution for that issue is redesign of main program loop and/or add network layer that would facilitate communication between nodes.

Theoretical network range, that can be computed using Friis equation [8], has been also confirmed. Theoretically the range is less than 150 m in free space [9]. It has been practically proven than network's capability of proper communication is kept at distances 120–130 m, even in office environment, without direct line-of-sight requirement met and with very simple antennas, supplied by radio modules manufacturer.

5 Summary

Presented Experimental Wireless Mesh Network node has been successfully deployed as hardware and software design using Atmel AVR ATMega16 microcontroller and Hope Microelectronics RFM12B radio module. The paper presents details on the design, especially network layer and other network-related issues. Experimental results prove limited usability of that system, although general specific aspects of wireless mesh networks (such as selfconfiguring capability and auto-updating of routing tables) and regular data transmission between nodes has been shown.

Possible enhancements could be done with:

- redesigning of main loop of the node program,
- adding external RAM allowing bigger routing tables (1 KB SRAM yields limit of 32 records now),
- longer addresses allowing bigger network (only 254 usable addresses now),
- adding local keyboard and display to the nodes allowing direct text messages exchange,
- changing flooding algorithm (used for `HELLO` datagrams propagation) into edge forwarding algorithm,
- using more efficient antennas helping radio communication reliability,
- increasing network throughput (RFM12B modules allow maximum throughput 115.2 kbps),
- designing higher layers, especially network layer, helping datagram loose problems.

Acknowledgments. I would like to thank my graduate student, Marek Korniowski, who developed my initial idea of the network as his master dissertation [10]. Parts of his work are used here with author's permission.

References

1. Olejnik, R.: Przegląd rozwiązań sprzętowo-programowych pozwalających na budowę bezprzewodowych sieci o topologii kratowej (in Polish). Metody Informatyki Stosowanej 1(19), 69–74, Komisja Informatyki PAN, Szczecin (2009)
2. Zhang, Y., Luo, J., Hu, H. (eds.): Wireless Mesh Networking: Architectures, Protocols and Standards. Auerbach Publications, Boca Raton (2007)
3. Hossain, E., Leung, K. (eds.): Wireless Mesh Networks: Architectures and Protocols. Springer, London (2008)
4. BATMAN routing protocol, `http://www.open-mesh.net/`
5. Johnson, D., Ntlatlapa, N., Aichele, C.: A simple pragmatic approach to mesh routing using BATMAN. In: 2nd IFIP International Symposium on Wireless Communications and Information Technology in Developing Countries (WCITD 2008), Council for Scientific and Industrial Research, Pretoria (2008)
6. Atmel AVR ATmega16 datasheet, `http://www.atmel.com/dyn/resources/prod_documents/doc2466.pdf`
7. Hope Microelectronics wireless FSK transceiver module RFM12B, `http://www.hoperf.com/rf_fsk/rfm12b.htm`
8. Golio, M. (ed.): The RF and Microwave Handbook. CRC Press, Boca Raton (2001)
9. Olejnik, R.: A concept of emergency communication system based on ISM bands. In: Węgrzyn, S., Czachórski, T., Kwiecień, A. (eds.) Contemporary Aspects of Computer Networks, vol. II, pp. 275–280. Wydawnictwa Komunikacji i Łączności, Warszawa (2008)
10. Korniowski, M.: A concept of a failsafe computer network based on the licence-free bands (in Polish). Master thesis, West Pomeranian University of Technology, Szczecin (2009)

The Cost of Redundancy in Distributed Real-Time Systems in Steady State

Andrzej Kwiecień and Jacek Stój

Silesian University of Technology, Institute of Informatics
Akademicka 16, 41-100 Gliwice, Poland
{andrzej.kwiecien,jacek.stoj}@polsl.pl
http://www.polsl.pl

Abstract. The paper presents considerations that refer to introducing the term "cost of redundancy", and indicates general assumptions that allow defining this term and its quantitative determination. However, "the cost of redundancy" does not apply in this case to financial expenses but only time loses (temporal cost) associated with the use of redundancy in industrial real-time distributed systems. Several computer systems architectures are analyzed from the point of view of that temporal redundancy cost.

Keywords: distributed real-time system, data flow, redundancy, temporal cost of redundancy.

1 Introduction

As commonly known, redundancy allows to increase the reliability of computer systems, but also entails significant cost. On one hand, there are the obvious material cost associated with the necessity of purchasing additional equipment, hardware and software installation, making additional wiring needed primarily for the implementation of redundancy of communication bus or synchronization of the computer units. On the other hand, there are time overheads brought into the system by every additional redundant component that requires service. These overheads worsen time parameters of real-time systems and may even lead to the violation of time limitations which are determined for the system. Therefore, it is necessary to know the size of these overheads while designing and building real-time systems with redundancy and during development of already existing systems by interposing a redundancy. Unfortunately, authors of the article have not found any literature on this subject (which does not mean it does not exist), where the impact of redundancy on time dependencies in industrial real-time systems is discussed. It is worth emphasizing that the subject of considerations is the time cost (overheads) associated with the usage of redundancy. The ability to define the cost rate is important in redesigning or modernization of existing systems, because it allows to answer the basic question: whether the use of redundancy in the new system meets the real-time constraints?

A. Kwiecień, P. Gaj, and P. Stera (Eds.): CN 2010, CCIS 79, pp. 106–120, 2010.

For the needs of the research work the term *steady state* has been defined. The authors are aware that this term is commonly applied to electronic circuits or system automation. However, in this case, the term *steady state* refers to the state of distributed system with no breakdowns which should force the activation of redundant units. After a breakdown occurs the state is considered as transient and it lasts until the proper procedures are executed and duplicate (redundant) circuits are activated.

The further part of the paper concerns the impact of redundancy on the time parameters of real-time information systems in the *steady state*. It discusses the influence of redundancy in the subsystem of computer units for four basic architectures [1,2]:

- architecture with redundancy of communication bus,
- architecture with redundancy of communication link, that is network coprocessor and communication bus redundancy,
- architecture with redundancy of computer unit,
- architecture with redundancy of computer unit and communication link.

Moreover, sources of redundancy time overheads in inputs and outputs subsystems are also mentioned. Furthermore, issues referring to time cost of the implementation of redundant computer units synchronization are also indicated.

2 Redundancy in Input and Output Subsystem

Redundancy in input and output subsystems (without bus redundancy) is mainly associated with an increase of number of subscribers in a communication network caused by additional remote input/output stations. As a result, the network operation cycle lengthens as it depends on the following parameters:

- time needed to gain access to the communication medium by subscribers in a communication network – time to transfer the rights for transmission,
- time of additional data transmission among remote input/output modules and other network subscribers (for transmission in both directions: *to* as well as *from* remote input/output station).

In case of time overheads associated with the access to the communication bus and the transfer of the right to transmit, their size and the manner of designation depends on what communication protocol is used. For example: in the Token Bus network the time overheads are associated with the transfer of a token to the additional subscribers. Whereas, in Master-Slave networks they results from the extension of exchanges scenario – increased number of exchanges triggered by the Master station. In some communication networks, where the right for transmission is exchanged with the use of timeslots (TDMA – Time Division Multiple Access) [3], this cost may be less than 1 ms for all the subscribers in total. It occurs, for example in the Genius network for transmission speed 153.6 kb/s, both during correct communication with the network subscribers, and at the break of communication with one or even all network subscribers.

In other communication networks the time for transferring the right for transmission may be more significant. It occurs in a very popular Modbus network. Time for transmitting a communication frame, transferring the right to transmission in the form of a request for read or write data, may be indeed very small (approximately 3.3 ms for the transmission speed 19 200 kb/s), but more important are the delays caused by the lack of communication with subscribers of the network associated with the need to count down time limit (timeout) (the sizes of these delays depend on configuration; they are usually defined at the level of tens or even hundreds of milliseconds). In other words, when one subscriber is disconnected from the previously mentioned Genius network, then the rest of subscribers receive the right to transmit data more frequently, because the expected time for the right for transmission will be reduced approximately to data transmission time due to disconnected subscriber. Whereas, in Modbus network the subscribers in the same situation are able to transmit data less frequently because of the necessity of counting down the time limit (timeout) by the Master subscriber, when it tries to communicate with the disconnected Slave subscriber.

The time needed for additional data transmission also depends on communication protocol and the parameters of communication network that is used. However, it is worth mentioning that in some networks a broadcasting data transmission is possible. In such a case, the redundancy of output modules may not require a multiple transmission of the same information regarding the status of digital and analogue redundant outputs, if the redundant modules have the same addressable outputs. It is due to the fact that a single data package may be received by a number of output modules working in the redundancy system. Hence, the influence of the redundant output modules on the communication system operation cycle length may be negligible.

Broadcasting data transmission in case of digital inputs redundancy may in some circumstances also allows to minimize the time needed for data transmission. It occurs when the input data from modules are to be transmitted to more than one subscriber. Broadcasting data transmission of that data enables sending a single frame with current output state and status to all subscribers of the network, instead of a separate frame for each of the subscribers awaiting the input data individually.

The elements mentioned above – transferring the right for transmission and transmission of additional data to and from remote redundant input/output stations cause the length of communication system operation cycle to be increased. In case of appearing one redundant input/ output station, the cycle increases by:

$$T_{\mathrm{RIOBus}i} = T_{\mathrm{MA}i} + T_{\mathrm{TR}i} \ , \tag{1}$$

where:

$T_{\mathrm{RIOBus}i}$ elongation of the time of a cycle with redundancy of a single remote input/output station,
$T_{\mathrm{MA}i}$ time to transfer the right for transmission for a single input/output station,
$T_{\mathrm{TR}i}$ time of data transmission among the additional, redundant input/output station and the other communication network devices.

It is necessary to notice that time for data transmission $T_{\mathrm{TR}i}$, should include not only the transmission from the remote input/output station for the other subscribers on a network, but also time for transferring additional data to this redundant input/output station.

Time, which will extend the operation cycle of a given network with redundancy of input/output remote station, can be expressed as follows:

$$T_{\mathrm{RIOBus}} = \sum_{i=1}^{n} T_{\mathrm{RIOBus}i} \ , \tag{2}$$

where, according to (1), $T_{\mathrm{RIOBus}i} = T_{\mathrm{MA}i} + T_{\mathrm{TR}i}$ determines the time overhead of redundancy of the i-th station input/output.

Presuming that the time overheads associated with the transferring the right for transmission is constant for all redundant input/output stations and equals T_{MA}, then formula (2) can be transformed into the following form:

$$T_{\mathrm{RIOBus}} = n \cdot T_{\mathrm{MA}} + \sum_{i=1}^{n} T_{\mathrm{RI}i} \ , \tag{3}$$

where n is the number of redundant input/output stations.

Additionally, redundancy in the input/output subsystems contributes to the system time overhead associated with the necessity to process additional data of the redundant modules. The mentioned processing consists of execution of many voting algorithms and data transmission from the redundant input/output between the network coprocessor and the module of a central unit in the computer unit. This overhead causes the elongation of the time of automaton cycle duration of a computer unit, and its size depends on the complexity of the voting algorithms' calculation and the number of data that comes from the redundant modules. In addition, it depends on hardware platform used for building the applied system. Thus, it can be said that the time overhead for the length of automaton cycle is primarily a function of the number of input/output signals:

$$T_{\mathrm{RIOCPU}} = f_{\mathrm{RIOCPU}}(n_{\mathrm{DIO}}, n_{\mathrm{AIO}}) \ , \tag{4}$$

where:

T_{RIOCPU} elongation of the time of automaton cycle using redundancy in input/output subsystems,
f_{RIOCPU} function of time-based cost of the redundancy in input/output subsystem,
n_{DIO} number of discrete inputs and outputs subjected to redundancy,
n_{AIO} number of analogue inputs and outputs subjected to redundancy.

The total time cost of redundancy in subsystem input/output will change in the function of time overhead of redundancy for the length of network operation cycle T_{RIOBus}, and automaton cycle T_{RIOCPU}. The cycle of T_{A} automaton is the time of execution of the basic loop in the distributed system (diagnostics → sampling of imputs → execution of application → system operation → updating of outputs → diagnostics ...).

3 Redundancy of Communication Bus

The determined time overheads, which cause deterioration of the real-time system parameters, accompany also the redundancy of communication bus. The exemplary system of the redundancy of communication bus is presented in the Fig. 1.

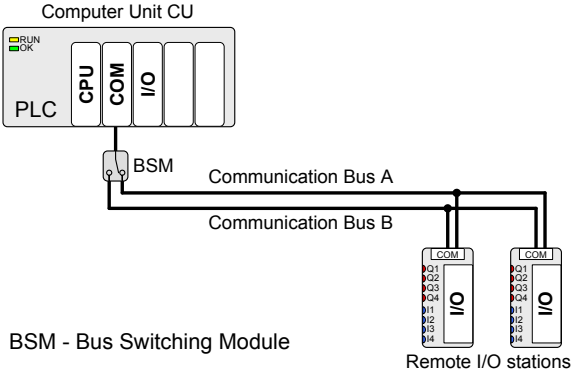


Fig. 1. Architecture with a redundant communication bus

The redundancy of communication bus causes the time of automaton operation cycle of a computer unit to be extended. In each cycle there is a necessity to choose a bus trough which the data is to be transmitted. However, this requires the diagnostic of the bus operation and switching of the buses in case of a breakdown.

The time cost of using redundancy of communication bus in the steady state can be expressed as follows:

$$T_{\mathrm{RBus}} = T_{\mathrm{RBusCPU}} = T_{\mathrm{DiagBus}} + T_{\mathrm{CHBus}} \ , \tag{5}$$

where:

T_{RBus} time cost of using redundancy of communication bus,
T_{RBusCPU} time elongation of automaton cycle duration of a computer unit caused by redundancy of communication bus,
T_{DiagBus} time overhead associated with bus diagnostic,
T_{CHBus} time overhead associated with the choose of a bus.

Both elements in the formula (5) depend on the way of implementation of diagnostic procedures and the way of choosing the communication bus, by which communication is to be maintained. Diagnostic refers usually to some system functions that are called, and which enable to read communication module status. Moreover, the choice a bus may also be associated with calling of system functions, or it can be executed for example, via setting up one of the local digital outputs of the computer unit.

It is necessary to mention that the cost of using bus redundancy does not depend on the size of the system or on the data amount (number of signals, numbers of variables) transmitted with the use of the communication network. The use of such type of redundancy does not cause additional subscribers to connect; hence the number of operations to transfer the right for transmission does not change. Similarly, the amount of transmitted information via communication network also is unchanged. The redundancy of communication bus may require utilization of additional elements necessary for bus switching. Nevertheless, such elements are usually passive – they do not participate in the process of communication, and as a consequence they do not contribute any time expenses.

The size of time cost of using redundancy of communication bus is not significant and may even amount to less than 1 millisecond, but this refers only to the cost in system steady state. More important are the time overheads of redundancy in transient state, which is not the subject of the this paper. In the transient state, the time cost of redundancy of communication bus may be critical for the system operation.

4 Communication Link Redundancy

After having extended the system with bus redundancy to the redundancy of network coprocessor, the system with communication link redundancy is obtained. The redundancy of communication bus and the network coprocessor does not modify the configuration of a communication network, but only the configuration and application program of a computer unit. This, on the other hand, causes the automaton operation cycle to be elongated as the consequence of execution of diagnostic routines and communication tasks in the communication coprocessor in every cycle. Moreover, this type of redundancy is associated with the increase of variable count (digital, analogue signals, etc.), which have to be processed by computer unit. In case of system with bus redundancy and without coprocessor redundancy, there was no such need because, as an example, one input signal corresponded to one variable, but the value of that variable could be transmitted via two independent communication buses, while communication via basic bus excluded the possibility to communicate through a backup bus. However, using redundancy of bus and network coprocessor requires the ability to distinguish the data which is received via particular communication links, or pairs (communication bus, network coprocessor). In case of double redundancy each digital input signal state should correspond to two variables. It is obvious then, that increasing the number of variables causes the time for application program of computer unit be extended. The total time overhead associated with communication bus redundancy and network coprocessor is illustrated by formula:

$$T_{\mathrm{RL}} = T_{\mathrm{RLCPU}} = T_{\mathrm{DiagL}} + T_{\mathrm{SVCM}} + T_{\mathrm{RIOCPU}} \ , \tag{6}$$

where:

T_{RL} time-based cost of using bus redundancy and communication coprocessor,
T_{RLCPU} time elongation of the automaton cycle resulting from using redundancy of communication link,

T_{DiagL} time overhead associated with diagnosing of additional communication link,
T_{SVCM} time overhead associated with additional coprocessor operation (elongation of communication window in automaton cycle),
T_{RIOCPU} time overhead associated with increasing the variable numbers in the application program of a computer unit .

The value T_{DiagL} in the formula depends on the kind of device that is used, and should be defined by its producer. Similarly to the redundancy of communication bus, the diagnostic of the redundant communication module in the redundant communication link may be associated with the necessity to execute additional diagnostic function.

The operation time of the communication module T_{SVCM}, includes diagnostic of this module which is done by operating system of the computer unit, and most of all the data transmission time referring to input/output states of the system between the network coprocessor and the CPU module. The time for that data transmission depends on the system size and the number of inputs and outputs.

Similar dependence appears when referring to the time needed for additional data transmission that come from redundant communication link T_{RIOCPU} (see: formula (4)). It is required here to execute the voting algorithms in case of discrete inputs, and to determine the value of analogue inputs (for example, counting the average state of analogue inputs received via two communication links). The necessity for implementation of additional routines may also involve redundant analogue outputs. The routines may be implemented either by an individual computer's operating system or an application.

The value T_{SVCM} (like T_{RIOCPU}) depends on the size of the system expressed in the number of input and output signals. This dependency is usually linear or close to linear.

$$T_{\mathrm{SVCM}} = f_{\mathrm{SVCM}}(n_{\mathrm{DIO}}, n_{\mathrm{AIO}}) \ , \tag{7}$$

where:

f_{SVCM} time-based cost function of an additional communication module of a redundant communication link,
n_{DIO} the number of discrete inputs and outputs being redundant,
n_{AIO} the number of analogue inputs and outputs being redundant.

It is worth mentioning that redundancy of communication link may be accompanied by redundancy of input/output subsystem. In such case additional redundant input/output remote stations are connected to the backup communication bus. An example of such a system is presented in Fig. 2. It consists of a single computer unit with a communication redundant link and two redundant input/output remote stations.

If the bus redundancy is not accompanied by redundancy in input/output subsystem, then each remote station is connected to both buses. It must then have a dual network interface or a system for switching between the buses (station communicates with the computer unit with the use of one bus, and in case of lack of communication it switches to the backup bus). However it has no essential impact on the time dependencies in the system.

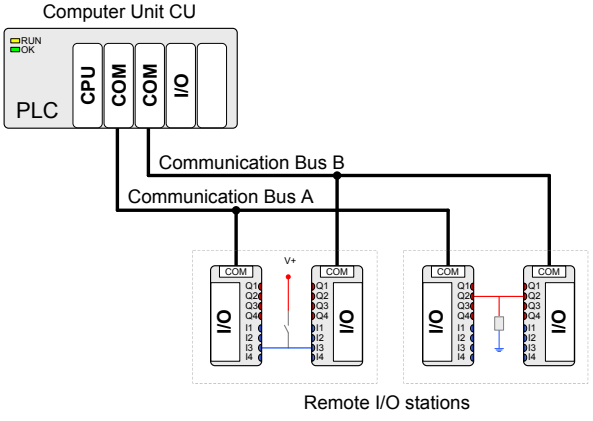


Fig. 2. Architecture with redundant communication link and two pairs of redundant input/output stations

5 Redundancy of a Computer Unit

Redundancy of a computer unit is associated with the connecting of a redundant unit to the communication network, and hence with increasing the number of subscribers. An example of a system with the redundancy of the computer unit is illustrated in Fig. 3.

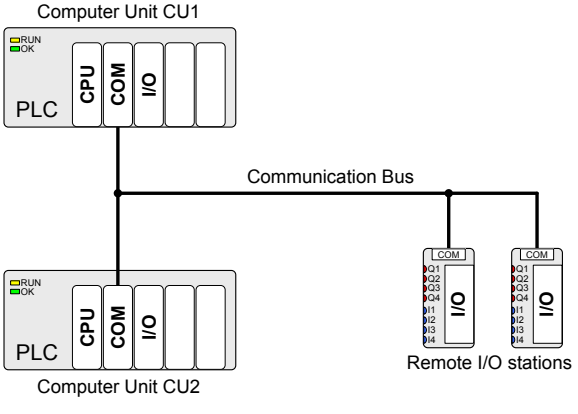


Fig. 3. Architecture with redundant computing unit

Connecting additional subscribers to the communication network constitutes the main time cost of using redundancy of the computer unit. The operation cycle of the communication network will be extended by the value:

$$T_{\mathrm{RCUBus}} = T_{\mathrm{MACU}} + T_{\mathrm{TRCU}} \ , \tag{8}$$

where:

T_{RCUBus} elongation of network operation cycle resulting from redundancy of a computer unit,
T_{MACU} access time (passing the right for transmission) of a redundant computer unit to the communication bus,
T_{TRCU} time of data transmission among the redundant computer unit and the other subscribers of the communication network.

The values T_{MACU} and T_{TRCU} being the elements of formula (8) are analogue to the T_{MA} and T_{TR} components of formula (2) and they have already been discussed referring to the redundancy in input/output subsystem. However, it should be noted that the time of T_{TRCU} data transmission may relate not only to data transmission between remote input and output stations, but also the additional data exchanged among the redundant computer units. This data may include, among others, so-called "word of life" that is used for mutual diagnostic of CPU modules in computer units. Execution of its diagnostic algorithms might be needed to perform switching of source control in the system with redundancy of the computer units at the time of failure of the main unit, or the unit that controls operation of the object (backup unit).

Data transmitted among computer units extends additionally the operation cycle of the network and increases the size of the T_{TRCU} parameter. This data must also be prepared and processed by the computer unit. This, in turn causes the automaton cycle to be extended according to the following formula:

$$T_{\mathrm{RCUCPU}} = T_{\mathrm{DP}} + T_{\mathrm{DEX}} \ , \tag{9}$$

where:

T_{RCUCPU} time overhead associated with the time of the automaton cycle duration due to the additional exchange of data between computer units,
T_{DP} time of additional data processing in the program of the controller,
T_{DEX} time of the additional data exchange between the central unit module and the network coprocessor module of the computer unit.

T_{DP} time value is dependent on the number of additional data, the manner of processing it and the parameters of the central unit (such as, for example, its performance), which have influence on execution time of particular operations on this additional data. These operations are executed in the application program implemented in the computer unit. Therefore, T_{DP} parameter defines how much the duration of the application program will be extended.

T_{DEX} also depends on the number of additional data but, what is more important, on the architecture of the computer unit. The size of time for a specific data number should be determined by the manufacturer of the implemented devices in their technical and maintenance documentation. Total time overhead of redundancy of the computer units will be the time function defined in formulas (8) and (9):

$$T_{\mathrm{RCU}} = f_{\mathrm{RCU}}(T_{\mathrm{RCUBus}}, T_{\mathrm{RCUCPU}}) \ , \tag{10}$$

where:

T_{RCUBus} elongation the time of operation cycle of network resulting from the use of redundancy of computer unit,

T_{RCUCPU} time overhead associated with the time of automaton cycle duration resulting from the exchanging of the additional data among the computer units.

It is worth mentioning that the time overhead is very important for the duration of operation cycle of the communication network – T_{RCUBus}. This overhead is usually much larger than T_{RCUCPU} and has more impact on the system response time to enforcement.

6 Redundancy of a Computer Unit and a Communication Link

System with the redundancy of computer unit can be additionally extended to provide redundancy of communication link. An exemplary system with the redundancy of computer unit and communication link containing two remote outputs/inputs stations is presented in Fig. 4.

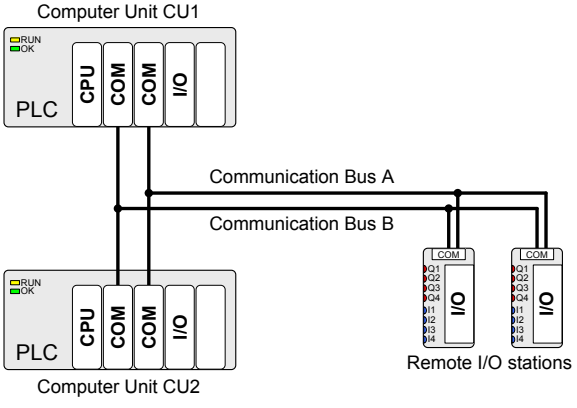


Fig. 4. Architecture with redundant computing unit and redundant communication link

The extension of the system to the redundancy of the communication link result in an extra time overheads connected with the operation of this link. Therefore, the time of automaton cycle duration of redundant units will be elongated (time T_{RLCPU}). Taking into account the cost associated with the redundancy of computer unit that cause elongation of the network operation cycle (see: formula (10), time T_{RCU}), then the total elongation of automaton cycle equals:

$$T_{\mathrm{RCUL}} = T_{\mathrm{RCU}} + T_{\mathrm{RLCPU}} \ , \tag{11}$$

where:

T_{RCUL} time overhead of redundancy of the computer unit and communication link,
T_{RCU} time overhead of redundancy of the computer unit,
T_{RLCPU} time overhead of redundancy of the communication link.

In case of the control object, for which a non-impact switching of the control source is significant, the synchronization of redundant computer units is very important. Its purpose is to synchronize their internal state and the time of execution of computer units' program. The impact of synchronization on time dependencies in real-time systems is discussed in the next section.

7 Synchronization of Computer Units

Realization of computer units synchronization has fundamental impact on the automaton cycle duration of computer units. It results from the necessity of diagnosing and operating the synchronization modules in every cycle of the controller. The operation of synchronization modules consists of data transmission time from the central unit (synchronizing) to backup unit (synchronized). The total time overhead of synchronization can be expressed as follows:

$$T_{\mathrm{S}} = T_{\mathrm{DiagS}} + T_{\mathrm{SVCS}} = T_{\mathrm{DiagS}} + T_{\mathrm{STRI}} + T_{\mathrm{STRO}} \ , \tag{12}$$

where:

T_{S} time cost of synchronization of computer units,
T_{DiagS} time overhead associated with the diagnostics of a module for synchronization,
T_{SVCS} time overhead associated with the operation of a module for synchronization,
T_{STRI} time of inputs synchronization,
T_{STRO} time of outputs synchronization.

It is worth emphasizing that additional redundancy of synchronization link is used very frequently. In such a case, in each unit there are two redundant synchronization modules that form independent links. However, it does not mean that each channel must cause the same extension of automaton cycle (totally by $2 \cdot T_{\mathrm{S}}$). Data transfer in the automaton cycle can be processed once in this operation cycle, using only one pair of synchronization modules – one synchronization link. Moreover, it can be also done simultaneously at two synchronization links. The time cost of synchronization, in such a case equals $2 \cdot T_{\mathrm{DiagS}} + T_{\mathrm{SVCS}}$. Considering the fact, that the values of time T_{DiagS} is usually considerably smaller than the value T_{SVCS}, it can be claimed, that the redundancy of synchronization link contributes to the system negligibly small time overheads.

8 The Meaning of Time Overheads Associated with Redundancy

Redundancy of any type may be associated with time overheads that influence the network operation cycle duration and the automaton cycle duration. Various redundant architectures may influence either both or one of the mentioned parameters. Thus, it should be noted that the time cost of using redundancy is not determined by difference in the length of mentioned values – automaton cycle or network operation cycle, before and after applying the reservation to the system. Modification of these parameters, in itself does not explain the importance of time overheads.

The meaning of time overheads of redundancy can be determined by setting of one of the fundamental parameters characterizing any real-time system. This parameter is the system response time to enforcement. It allows to determinate whether the system is able to meet the time demands or not. Therefore, the determination of differences between the length of time response before and after using the redundancy may give the evidence of the cost of using it.

System response time is the maximum time, in which the computer unit controlling the operation of the object is able to react to an event taking place in this object. Thus, it is the time needed to detect the fact that the event occurred in the object by collecting the input vector of control system (acquisition of inputs), executing application program running in the computer unit, and then driving the appropriate output signals of the system (outputs update). It is the response to the event which took place in the control object (sampling-processing-responding) [4].

In case of a centralized system (made of one computer unit equipped with local input/output modules) working in a synchronous way, the system response time is equivalent to two automaton cycles T_{A}. Maximum duration of one automaton cycle in this case is needed to detect event in the control object. Another cycle of automaton is required to generate the system response to the enforcement – execution of application program and updating the outputs of the system.

Determination of the response time of a distributed system is a bit more complex. The time of the response is still the time needed for inputs' acquisition, execution of application program and updating of outputs, however acquisition of inputs and updating of outputs refer to the remote input/output, which are connected to the computer unit via communication network (Fig. 5).

Acquisition of inputs in the distributed system concerns collecting input signals via remote input/output station and sending information about the state of these signals to the computer unit through the communication network. Then a computer unit, as in the centralized system, through the execution of the application program running in the computer, generates a response of the control system to a given input vector in the form of the corresponding value of the output vector. Output vector is sent to the remote input/output station, where both appropriate updating of outputs and setting the object executive devices are made. The information flow associated with the generation of the system response to enforcement is presented in Fig. 6 [4].

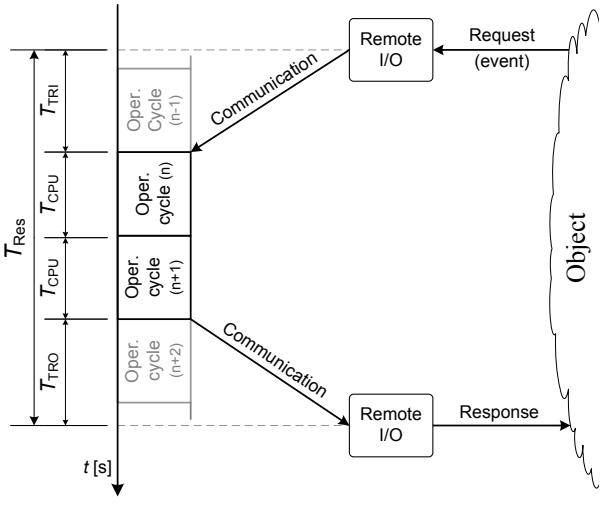


Fig. 5. Response time of a distributed system

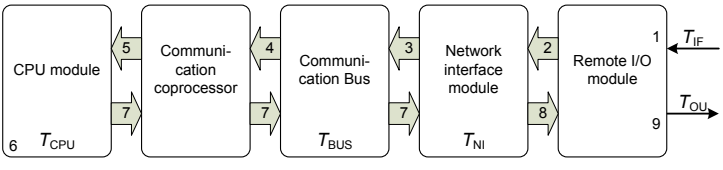


Fig. 6. Information flow associated with generating response of a computer system to an event

The following sections/stages illustrated in Fig. 6, which consists of generation of the system response to the enforcement refer to:

1. Acquisition of the input signal from the object.
2. Saving the input state into the network memory module interface (network coprocessor of the remote input/output station).
3. Sending data using the communication network.
4. Receiving data by network coprocessor of computer unit.
5. Rewriting the input data to the computer unit memory.
6. Generating corresponding value of the output vector based on the input data.
7. Rewriting the input vector into the memory of the communication coprocessor and then sending the data into the network interface module of the remote input/output station.
8. Rewriting data from the internal memory of the network interface of the remote input/output station into the memory of the output module.
9. Outputs updating in accordance with the newly obtained value of the output vector.

When assume that the computer unit, the communication network, the network interface module and the input/output module operation cycle is accordingly $T_{\mathrm{CPU}}, T_{\mathrm{Bus}}, T_{\mathrm{NI}}$, and the time required for acquisition of input state equals T_{IF}, and the time required for updating of the outputs equals T_{OU}, then the response time of distributed system T_{Res} to enforcement can be determined as follows:

$$T_{\mathrm{Res}} = 2 \cdot T_{\mathrm{CPU}} + k_{\mathrm{Bus}} \cdot T_{\mathrm{Bus}} + 4 \cdot T_{\mathrm{NI}} + T_{\mathrm{IF}} + T_{\mathrm{OU}} \ , \tag{13}$$

where:

T_{Res} the response time to enforcement in the distributed system,
T_{CPU} time of the automaton cycle,
T_{Bus} time of the operation cycle of the communication network,
k_{Bus} factor determining the number of the network cycles required for sending the input vector (with the enforcement) and the output (with the response to enforcement) of the system,
T_{NI} time of network interface operation cycle of the remote input/output station,
T_{IF} time of filtration of the inputs,
T_{OU} time of updating of the outputs.

The time required for operating the communication coprocessor of the computer unit is contained in T_{CPU} component. In addition, a component specifying the time required for the data transmission with the use of communication bus, and more precisely k_{Bus} factor, may have a different value depending on the type of the communication network. For example, in the Genius network (according to the manufacturer documentation [5]) it equals 3, which means that in order to generate the system response to the enforcement the execution of three network cycles will be necessary. In the majority of other networks two cycles will be sufficient ($k_{\mathrm{Bus}} = 2$).

Based on the formula (13) it can be concluded that extending the automaton cycle by time t_{A} will increase the time of the system response to enforcement by $2 \cdot t_{\mathrm{A}}$ time. Additionally, elongation of the network operation cycle by t_{S} will increase the size of response time by k_{Bus}-fold higher ($k_{\mathrm{Bus}} \cdot t_{\mathrm{S}}$).

9 Final Notes

Redundancy on one hand involves some additional financial cost connected with the purchase of additional elements, their configuration and programming, delivery, installation, wiring, setup and maintenance. On the other hand, it involves some temporal cost, too. The temporal cost of redundancy are any delays introduced into the computer system by the redundant elements. Every additional element needs some time to be serviced and that makes the whole system to be slower – its response time to be longer.

This paper concerned the redundancy of different elements of a computer systems and its influence on temporal characteristics of that system. The source

of any delays and their importance was analyzed. In future papers the delays will be calculated for specific systems and then measured in order to check the validity of the calculation. Moreover, a transient state needs also to be considered. The transient state starts whenever one or more of the redundant elements in the systems breaks down. The state lasts until the failure is serviced and appropriate steps are taken in order to control the failure (e.g. a switchover of the redundant controller). The most weighty temporal cost/delays are connected with the transient state. They all must be taken into consideration while doing the WCET analysis (Worst Case Execution Time).

References

1. Kwiecień, A., Gaj, P., Cupek, R.: Redundancja w przemysłowych przestrzennie rozległych systemach czasu rzeczywistego. In: Współczesne Problemy Systemów Czasu Rzeczywistego, WNT, Gliwice (2004)
2. Stój, J., Kwiecień, A., Sidzina, M.: Analiza wybranych architektur redundantnych z zastosowaniem sieci Modbus RTU. In: Sieci komputerowe, Aplikacje i zastosowania, WKŁ, Warszawa, vol. 2, pp. 359–368 (2007)
3. Wandeler, E., Thiele, L.: Optimal TDMA Time Slot and Cycle Length Allocation for Hard Real-Time Systems. In: Asia and South Pacific Conference on Design Automation (January 2006)
4. Kwiecień, A.: Analiza przepływu informacji w komputerowych sieciach przemysłowych. Wydawnictwo Politechniki Śląskiej, Gliwice (2002)
5. Genius I/O System and Communications. GE Fanuc Automation, Programmable Control Products, document nr: GEK-90486F (November 1994)

Fuzzy-Neural Controller in Service Requests Distribution Broker for SOA-Based Systems

Mariusz Fras[1], Anna Zatwarnicka[2], and Krzysztof Zatwarnicki[2]

[1] Wroclaw University of Technology, Institute of Informatics,
Wybrzeze Wyspianskiego 27, 50-370 Wroclaw, Poland
mariusz.fras@pwr.wroc.pl
[2] Opole University of Technology, Institute of Automatic Control
and Computer Science, ul. Mikołajczyka 5, 45-271 Opole, Poland
a.zatwarnicka@po.opole.pl, k.zatwarnicki@gmail.com

Abstract. The evolution of software architectures led to the rising importance of the Service Oriented Architecture (SOA) concept. This architecture paradigm support building flexible distributed service systems. In the paper the architecture of service request distribution broker designed for use in SOA-based systems is proposed. The broker is built with idea of fuzzy control. The functional and non-functional request requirements in conjunction with monitoring of execution and communication links are used to distribute requests. Decisions are made with use of fuzzy-neural network.

Keywords: Service Oriented Architecture, service request distribution, fuzzy-neural network.

1 Introduction

One of the most important issues for contemporary computer network service providers is high quality of delivery of network services. Very important subject studied now in many research projects is the evaluation of service quality based on the knowledge on client-to-server network path and service execution system efficiency. Users want to complete requested service without inconvenience and as fast as possible with low latency and high throughput. According to those users activity development of Quality of Web Service (QoWS) sensitive Internet applications raises a need for building Web site services supporting quality and differentiation of services [1]. Nowadays, Service Oriented Architecture (SOA) paradigm gives us more universal approach to composing and building Internet network applications.

The information systems designed with SOA paradigm are working both in local and wide area networks, particularly built with Internet and GRID recourses. The clients don't need to carry about the localization and execution meanings of the service it want to use. The set of available services can change, however this process is not frequent, and usually the set of localizations of given service is constant for some period [2,3].

A. Kwiecień, P. Gaj, and P. Stera (Eds.): CN 2010, CCIS 79, pp. 121–130, 2010.

The paradigm of SOA says that given service can be achievable from different service providers. Because the parameters of provider's servers and communication links to them (communication costs) can be different, the quality of service delivery (especially the time of delivery) can be significantly different too [4]. Taking into account distribution of the services and characteristic of Internet network the proper architecture of service delivery based on service request distribution may significantly increase the quality of services built with SOA paradigm.

In this work the architecture of service request distribution broker built with fuzzy-neural control idea and designed for use in SOA-based systems is proposed.

2 The Request Broker for SOA-Based Systems

The idea of the proposed service request distribution system is to distribute clients requests via service broker B to known execution systems of service providers sp_m (Fig. 1).

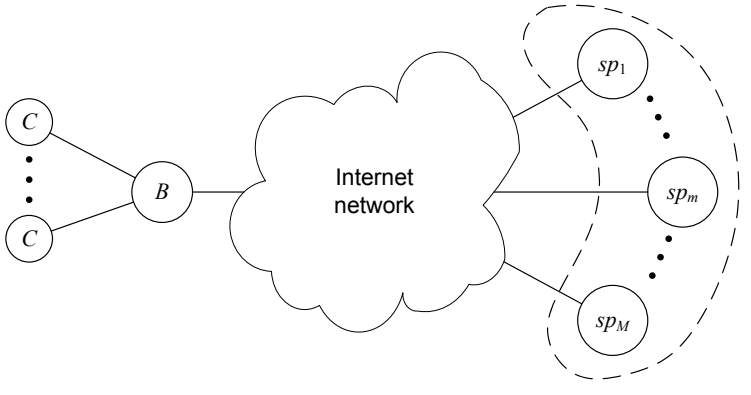


Fig. 1. General infrastructure of service request distribution system

The assumptions for the system are the following:

- the client of the broker is a system delivering complex services $cs(i) \in CS$, where CS is a set of possible services, $i \in \langle 1, \ldots, I \rangle$, aggregated from atomic services $as(j)$, $j \in \langle 1, \ldots, J \rangle$, available in execution systems distributed in the Internet network,
- there is known the set $SP = \{sp_1, \ldots, sp_m, \ldots, sp_M\}$, $m \in \langle 1, \ldots, M \rangle$, of execution systems and it is constant for some considered period,
- there is known the set of atomic services $AS = \{as(1), \ldots, as(j), \ldots, sp(J)\}$ that can be requested by the client and that are available in execution systems $sp_m \in SP$,

- the given atomic service $as(j)$ can be available in several different localizations (execution systems) – the concrete atomic service at given localization is called instance of atomic service,
- the considered resources (from the broker point of view) that affect service execution parameters, are communication links and execution systems.

The goal of the broker is to distribute requests for atomic services that constitute requested complex services (allocate communication resources) in order to fulfill formulated demands for execution of services. Knowing demands on service execution it should distribute request to proper execution system.

Let $CS = \{cs(1), \ldots, cs(i), \ldots, cs(I)\}$, $i \in \langle 1, \ldots, I\rangle$ be the set of possible complex services supplied by the complex service delivery system. The complex service $cs(i)$ is composed with more basic services, especially atomic services $as(k)$, $k \in \langle 1, \ldots, K_i\rangle$, i.e. the ones that can't be partitioned thereafter. The complex service can be described as a directed graph of atomic services $GA\langle V, A\rangle$, where $V = \{as(1), \ldots, as(k), \ldots, cs(K)\}$, and $A \subset V \times V$ is a set of pairs defining edges of execution dependencies between atomic services (vertices) – so called implementation graph of complex service.

The atomic services $as(j), j \in \langle 1, \ldots, J\rangle$ are available (localized) at the constant set SP of execution systems (service providers). The execution system sp_m delivers the set $IS(sp_m) = \{is(m_1), \ldots, is(m_p), \ldots, is(m_P)\}$ of instances of atomic services, where: P – the number of different services.

The instance of the service $is(m_p)$ is characterized by the non-functional parameters $\Psi(is(m_p)) = [\Psi^1_{m_p}, \Psi^2_{m_p}, \ldots, \Psi^f_{m_p}, \ldots, \Psi^F_{m_p}]$, that may be different for two different instances of the same atomic service. An examples of non-functional parameters are: the level of security, the completion time of service execution, etc.

Let $GA_n\langle AS_n, ES_n\rangle$ be the implementation graph of complex service requested at the moment n. The set AS_n is a subset of the set AS containing all atomic services necessary to accomplish complex service cs_n requested at the moment n: $AS_n = \{as(n_1), as(n_2), \ldots, as(n_k), \ldots, as(n_{K_n})\} \subset AS$, $k \in \langle 1, \ldots, K_N\rangle$, where K_n is the number of atomic services (number of vertices in the implementation graph). For each implementation graph it is an equivalent graph $G_n\langle IS_n, ES_n\rangle$, where $IS_n = \{IS_{n1}, IS_{n2}, \ldots, IS_{nk}, \ldots, IS_{nK_n}\} \subset IS$, IS is the set of all instances of atomic services. The set IS_{nk} is the set of all instances of atomic service as_{nk}, i.e. $IS_{nk} = \{is(n,k,1), is(n,k,2), \ldots, is(n,k,m), \ldots, is(n,k,M_{nk})\}$. IS_{nk} corresponds to the set $SP(as(n_k)) = \{sp_m : as(n_k) \in AS(sp_m)\}$, i.e. the set of M_{nk} execution systems that deliver instances of atomic service $as(n_k)$. The graph $G_n\langle IS_n, ES_n\rangle$ defines different execution graphs for the implementation graph $GA_n\langle AS_n, ES_n\rangle$.

Suppose, there is an complex service composed from two subsequent atomic services ($as(1)$, $as(2)$), and $as(1)$ is delivered in two different localizations, $as(2)$ is delivered in three different localizations, i.e. $IS_{n1} = \{is(n,1,1), is(n,1,2)\}$ and $IS_{n2} = \{is(n,2,1), is(n,2,2), is(n,2,3)\}$. The complex service can be executed in six different ways what illustrates paths of execution in Fig. 2 (the index n is omitted).

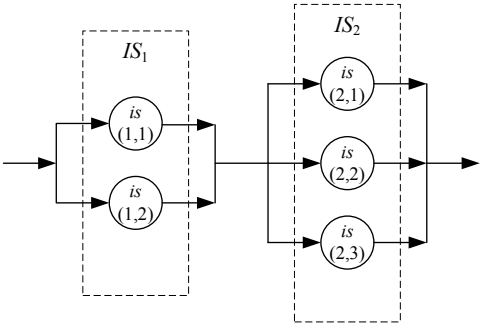


Fig. 2. Possible paths of execution of an example complex service

The choice of specific execution path is a problem that is involved with specifying demands $SLA_{nf,n}$ on non-functional parameters $\Psi(cs_n)$ of the complex service cs_n, that can be specified in request $X_n = \langle GA_n\langle AS_n, ES_n\rangle, SLA_{nf,n}\rangle$ at the moment n. The non-functional parameters of complex service $\Psi(cs_n)$ correspond to non-functional parameters of instances of atomic services $\{\Psi(is(n,k,m)\}$. Knowing $\{\Psi(is(n,k,m)\}$ it is possible to perform different procedures of fulfilling $SLA_{nf,n}$. The general goal of service request distribution system is to determine execution graph of complex service that fulfills $SLA_{nf,n}$:

$$GA_n\langle AS_n, ES_n\rangle \rightarrow G_n^*\langle IS_n^*, ES_n\rangle \ ,$$

where: $IS_n^* \leftarrow min_{k=1,\ldots,K_n} h(\Psi(is(n,k,m))$, what denotes the problem of finding appropriate instances of atomic services that should be requested in order to fulfill formulated demands on non-functional parameters, $h(\Psi(is(n,k,m))$ is formulated criterion of fulfilling $SLA_{nf,n}$.

One of the most important non-functional parameters is completion time of delivery of requested service. The impact on this time have two components communication links and execution systems, specifically completion time of data transfer and completion time of service execution at the execution system. Thus, one of the main functionality of service request distribution broker must to be estimation (forecasting) of non-functional parameters $h(\Psi(is(n,k,m))$ at the moment n for all instances of atomic service, $j = 1,\ldots,J$, $m = 1,\ldots,M$. Especially estimation of service execution time at execution systems and service request/reply transfer time.

To determine for the request arriving at the moment n destination server which should serve this request on the basis on time criteria, the broker has to support the following functionality:

- monitoring of servicing client requests,
- prediction of selected communication link parameters,
- estimation of transfer times and execution times of all instances of atomic services, on the basis of adaptive models of communication links and execution systems.

In the Figure 3 we propose the general architecture of the broker that fulfills all these functionalities.

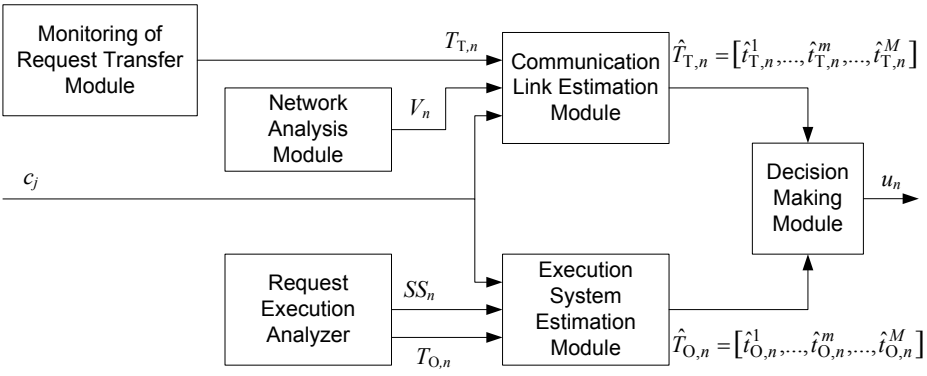


Fig. 3. General architecture of the service request broker

The decision u_n of the choice of one of M execution systems is performed on the basis of the vector $\hat{T}_{T,n}$ of estimated transfer times $\hat{t}^m_{T,n}$ of the request to each server m, and vector $\hat{T}_{O,n}$ of estimated execution times $\hat{t}^m_{O,n}$ of the request at each execution system m. The key components of the broker are estimation modules of the communication links and execution systems. Modules are built of M models of links or execution systems respectively that works as controllers.

The next section describes proposed method for estimation of completion times of data transfers t_T for requested atomic services.

3 The Fuzzy-Neural Controller

The input of the communication link model is identifier c_j of requested atomic service that arrives at the moment n, measured real transfer times $t^m_{T,n}$ of prior (before the moment n) requests to the system m and the vector V_n of two communication link parameters: forecasted link throughput $\hat{th}_n$ and forecasted link latency $\hat{t}_{TCPC,n}$ (namely TCP Connect Time). These two parameters are derived from periodic measurements of latency and link throughput, and using time series analysis based prediction algorithms. The output of each model is estimated transfer time $\hat{t}^m_{T,n}$ of the request.

Each communication link model is built for each destination and for each atomic service (for each instance of atomic service). It is designed as fuzzy-neural controller (Fig. 4) based on [5,6,7]. Hereafter, the indexes of link and service are omitted.

The inputs a_n and b_n of the fuzzyfication block are forecasted link latency $\hat{t}_{TCPC,n}$ and forecasted link throughput $\hat{th}_n$ at the moment n. The outputs are the grades of membership of latency μ_A in defined fuzzy sets $Z^a_k, k = 0, \ldots, K$ for

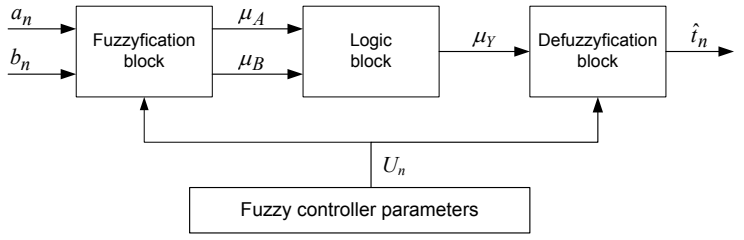


Fig. 4. Communication link model as a fuzzy-neural controller

latency, and grades of membership of throughput μ_B in defined fuzzy sets $Z_l^b, l = 0, \ldots, L$ for throughput. The sets are described with membership functions as follows:

$$\mu_{Z_1^d}(d) = \begin{cases} \frac{\delta_0 - d}{\delta_0} & \text{for } 0 \le d < \delta_0 \\ 0 & \text{for } \delta_0 \le d \end{cases}$$

$$\mu_{Z_h^d}(d) = \begin{cases} \frac{d-\delta_{h-1}}{\delta_h - \delta_{h-1}} & \text{for } \delta_{h-1} < d \le \delta_h \\ \frac{d-\delta_{h+1}}{\delta_h - \delta_{h+1}} & \text{for } \delta_h < d < \delta_{h+1} \\ 0 & \text{for } \delta_{h+1} \le d \end{cases} \tag{1}$$

$$\mu_{Z_H^d}(d) = \begin{cases} 0 & \text{for } 0 \le d \le \delta_{H-1} \\ \frac{d-\delta_{H-1}}{\delta_H - \delta_{H-1}} & \text{for } \delta_{H-1} < d < \delta_H \\ 1 & \text{for } \delta_H \le d \end{cases} .$$

Variable d corresponds to input a or b. Parameters $\delta_0, \ldots, \delta_h, \ldots, \delta_H$ correspond to parameters of fuzzy sets of input a: $\alpha_0, \ldots, \alpha_k, \ldots, \alpha_K$, or parameters of fuzzy sets of input b: $\beta_0, \ldots, \beta_l, \ldots, \beta_L$ respectively (H is equal K or L, h is the index of fuzzy set (k or l). Functions $\mu_{Z_h^d}(d), h = 1, \ldots, H$, are membership functions of fuzzy sets for parameter a i.e. $\mu_{Z_k^a}(a), k = 1, \ldots, K$, or parameter b i.e. $\mu_{Z_l^b}(b), l = 1, \ldots, L$.

The logic block inferences output. It computes grades of membership of consequence μ_Y of the rules that are of the format proposed in [8]:

$$a = Z_k^a \text{ AND } b = Z_l^b \text{ THEN } y = Z_r^y \ , \tag{2}$$

where: Z_r^y is a fuzzy set for output y (linguistic variable for estimated $\hat{t}_n$) $r = 1, \ldots, R, R = K * L$. The membership function for Z_r^y is described by formula:

$$\mu_{Z_r^y}(y) = \begin{cases} 1 \text{ for } y = y_r \\ 0 \text{ for } y \ne y_r \end{cases} . \tag{3}$$

Assuming Mamdani model of inference and Larsen form of T-norm for fuzzy implication, the membership function for consequence of fuzzy rule r is described by:

$$\mu_{Y_r}(y) = \begin{cases} \mu_{R_r}(a^*, b^*) & \text{for } y = y_r \\ 0 & \text{for } y \ne y_r \end{cases} , \tag{4}$$

where $\mu_{R_r}(a^*, b^*) = \mu_{Z_k^a}(a^*) * \mu_{Z_l^b}(b^*)$, a^* and b^* are spice values of inputs. The rule is fired when $\mu_{R_r}(a^*, b^*) > 0$.

After the defuzzyfication (using height method) the estimated transfer time of the request is derived according to formula (5):

$$\hat{t}_n = \sum_{r=1}^{R} \mu_{R_r}(a^*, b^*) * y_r \quad . \tag{5}$$

In order to satisfy the need of working of the controller in adaptive mode it is built as a 3-layered neural network similar to [8,9,6] (Fig. 5). The first layer of the network constitutes the fuzzyfication block. Each neuron transform sharp value of the input into fuzzy value. The second layer computes grades of membership for consequences of fuzzy rules. In the third layer and aggregation neuron the sharp value of the input is derived. A, B and Y are vectors of parameters α_k, β_l and y_r.

The parameters α_k and β_l of fuzzy sets for inputs a and b, as well as y_r for output are tuned during learning process using back propagation method [10] on the basis on measured actual request transfer times t_n. The computation formula for output is following (g is the moment of adjustment of parameters after completion of request x_n):

$$y_{r(g+1)} = y_{r(g)} + \gamma_y(\hat{t}_n - \hat{t}_n) * \mu_{R_r(g)}(a^*, b^*) \quad , \tag{6}$$

where γ_y is learning constant – scaling factor for parameter y used in learning process.

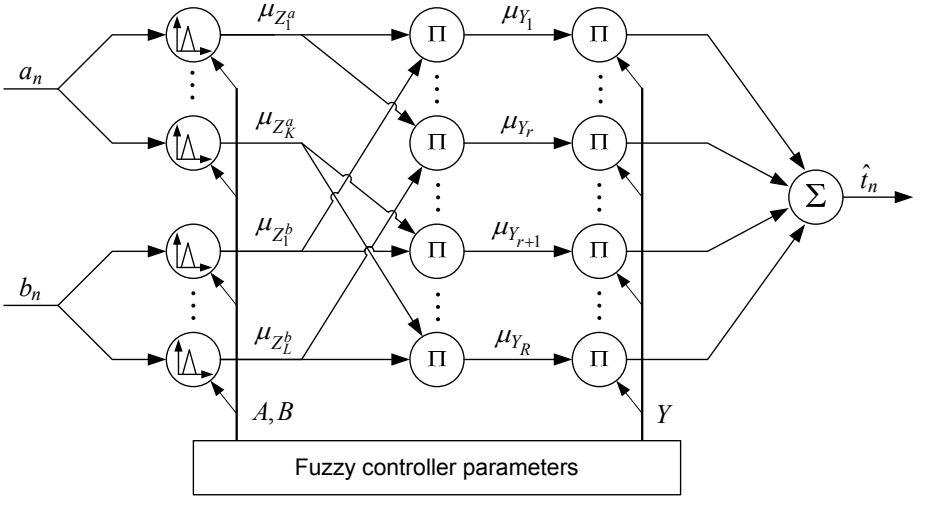


Fig. 5. Fuzzy-neural controller as fuzzy-neural network

The parameters for inputs a and b are tuned according to formulas (7):

$$\alpha_{k(g+1)} = \alpha_{k(g)} + \Delta^{\alpha}_{k(g)} \ , \qquad (7)$$
$$\beta_{k(g+1)} = \beta_{k(g)} + \Delta^{\beta}_{k(g)} \ ,$$

where:

$$\Delta^{\alpha}_{k(g)} = \gamma_a \cdot (\hat{t_n} - t_n) \cdot \sum_{r=1}^{R} \left[y_{r,n} \cdot \left(\mu_{Z^b_l(g)}(a^*) \right)^{(r)} \right] \cdot \sum_{l=1}^{L} \left(\frac{\partial \mu_{Z^a_k(g)}}{\partial \alpha_{k(g)}} \right) \ , \qquad (8)$$

and:

$$\Delta^{\beta}_{k(g)} = \gamma_b \cdot (\hat{t_n} - t_n) \cdot \sum_{r=1}^{R} \left[y_{r,n} \cdot \left(\mu_{Z^a_k(g)}(a^*) \right)^{(r)} \right] \cdot \sum_{l=1}^{L} \left(\frac{\partial \mu_{Z^b_l(g)}}{\partial \beta_{l(g)}} \right) \ , \qquad (9)$$

where $\left(\mu_{Z(.)(g)}(.)\right)^{(r)}$ is the value of grade of membership for input for interconnection r (see Fig. 5). Parameters a and b are learning constants.

4 Experimental Measurements for Request Transfer Time Estimation

The components of the request broker that concern estimation of transfer time including fuzzy-neural controller, were implemented as complex set of tools in Linux system. The network analysis module performs active measurements in Internet network and forecast TPC Connect time (latency of communication) and TCP/HTTP throughput on the basis of time series analysis method – Exponentially Weighted Moving Average (EWMA) algorithm. The monitoring module is able to measure transfer times of clients HTTP requests. The fuzzy-neural controller works for several values of scaling factors (learning constant used in learning process for parameters of fuzzy sets, for inputs: throughput and latency) simultaneously.

The preliminary experiments were performed as follows:

- at the client side (localized at Wroclaw University of Technology) the HTTP requests to given set of HTTP servers distributed in Internet network were generated,
- according to preliminary nature of the experiment there were selected 8 servers, for which communication links were the most stable,
- the service requests for transfer static HTTP object (2 MB size) were generated every 5 min. during about 30 hours,
- there were measured HTTP throughput and TCP Connect time for every request,
- the prediction of above parameters were performed using EWMA method,
- the fuzzy-neural controller has estimated transfer times with use of scaling factor $\gamma_a = \gamma_b = \gamma$ equal 0.1, 0.2,. . . , 0.9 simultaneously.

In the Figure 6 the aggregated Mean Average Percentage Error (MAPE) for 150 last measurements is presented. The graph shows data for learning constant γ used in fuzzy-neural controller equal 0.1,..., 0.9. This data shows effectiveness of the estimation when controller works after first phase of learning process. The results suggest that during stable phase of work more optimal parameter of the controller is $\gamma = 0.5$.

In the Figure 7 the cumulated MAPE of request transfer time prediction versus number of request n is shown for learning constant γ equal 0.5. The results shows proper learning of the system. However the error doesn't decline as fast and regularly as we would like. The more detailed analysis of measurements results suggests, that more accurate prediction method of link throughput should improve the work of the controller.

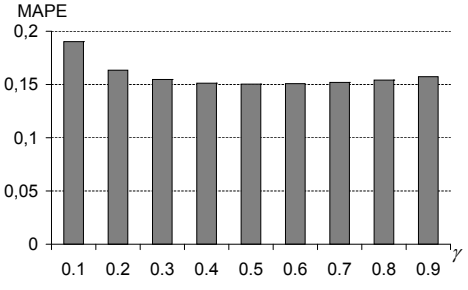


Fig. 6. Cumulative MAPE versus number of request n for learning constant $\gamma = 0.5$

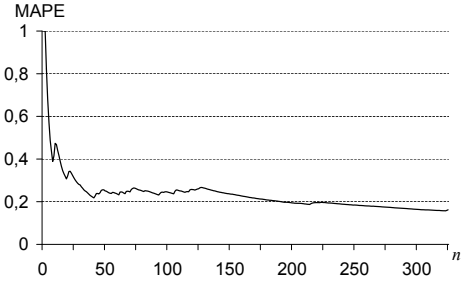


Fig. 7. Aggregated MAPE for 150 last measurements, for learning constant $\gamma = 0.1, \ldots, 0.9$

5 Conclusion

The new approach to build systems in compliance with SOA paradigm raises new challenges in delivery of Internet network services. In this work the architecture of broker designed for use in SOA-based systems has been proposed. The broker support efficient aggregation of complex services composed with atomic ones

that are localized in execution systems distributed in the Internet network. The broker performs requests distribution using fuzzy-neural network thus trying to deliver needed service to the client in the fastest way. The estimation of service transfer time is done with use of designed fuzzy-neural controller.

Presented distribution system was implemented in Linux system. The preliminary experiments based on the real measurements in Internet network show that such approach could help delivering of services in SOA-based systems. However, it should be done by appropriate prediction (or measurement) of communication link parameters. Further research on fuzzy-neural network parameters should also be continued.

Acknowledgements. The research presented in this paper has been partially supported by the European Union within the European Regional Development Fund program no. POIG.01.03.01-00-008/08.

References

1. Cardellini, V., Casalicchio, E., Colajanni, M., Mambelli, M.: Web switch support for Differentiated Services. ACM Perv. Eval. Rev. 29(2), 14–19 (2009)
2. Brawn, P.C.: Implementing SOA. Pearson Education, London (2008)
3. Mabrouk, M.: SOA fundamentals in a nutshell. IBM Corp. (2008)
4. Williams, W., Herman, R., Lopez, L.A., Ebbers, M.: Implementing CICS Web Services. IBM Readbook (2007)
5. Driankov, D., Hellendoorn, H., Reinfrank, M.: An Introduction to Fuzzy Control. Springer, Berlin (1993)
6. Mamdani, E.H.: Application of Fuzzy Logic to Approximate Reasoning Using Linguistic Synthesis. IEEE Trans. on Comp. C-26 (1997)
7. Yager, R.R., Filev, D.P.: Essentials of Fuzzy Modeling and Control, New York (1994)
8. Borzemski, L., Zatwarnicka, A., Zatwarnicki, K.: Global distribution of HTTP requests using the fuzzy-neural decision-making mechanism. In: Proc. of 1st Int. Conf. on Comp. Collective Intelligence. Lect. Notes in AI. Springer, Heidelberg (2009) (in press)
9. Jain, L.C., Martin, N.M.: Fusion of neural networks, fuzzy sets and genetic algorithms industrial applications. CRC Press LLC, London (1999)
10. Keong-Myung, L., Dong-Hoon, K., Hyung, L.: Tuning of fuzzy models by fuzzy neural networks. Fuzzy Sets and System 76 (1995)

Linking Simulation with Formal Verification and Modeling of Wireless Sensor Network in TLA+

Jerzy Martyna

Institute of Computer Science, Jagiellonian University, ul. Prof. Łojasiewicza 6, 30-348 Cracow, Poland
martyna@softlab.ii.uj.edu.pl

Abstract. In this paper, we present the results of the simulation of a wireless sensor network based on the flooding technique and SPIN protocols. The wireless sensor network was specified and verified by means of the TLA+ specification language [1]. For a model of wireless sensor network built this way simulation was carried with the help of specially constructed software tools. The obtained results allow us to predict the behaviour of the wireless sensor network in various topologies and spatial densities. Visualization of the output data enable precise examination of some phenomenas in wireless sensor networks, such as a hidden terminal, etc.

Keywords: SPIN protocol, flooding protocol, wireless sensor network, formal verification, TLA+ specification language, simulation.

1 Introduction

Wireless sensor networks [2] are a new generation of distributed systems of communication over radio links. These networks are used for environment monitoring [3], for remote monitoring of physiological data in health applications [4], in technological process automation [5], in building automation [6], etc. Among others, these networks are used in intelligent systems to find the source of fire and detect explosives at standoff ranges [7,8].

The TLA specification language (Temporal Logic of Actions) was introduced by Leslie Lamport [9], [10] for the specification of distributed, asynchronous systems. This language joins the linear temporal logic and the classical set theory providing of specifiction and verification of the software and hardware system. The application of the TLA for specification of real-time systems is natural. One of the first specifications in the TLA was realized as a model of a real-time system [11]. The improved version of the TLA language called TLA+ was introduced by L. Lamport in 1998 [12,1]. It was used for the specification and verification of designed concurrent processes [13]. With the help of a TLA+ full verification of Alpha processor 21354 EV7 was possible [14]. The verification of the Estelle technique proposed by ISO organization was given in the paper by Janowska [15].

The specification of radio link protocols was given in the paper by Mokkeden [16] and Sacuta [17]. In the first of them the specification of RLP1 protocol

A. Kwiecień, P. Gaj, and P. Stera (Eds.): CN 2010, CCIS 79, pp. 131–140, 2010.

which is based on TDMA was made. A formal specification of the mobile cellular network was given in the [18]. The formal verification of the protocol for IEEE 802.16 WiMAX was given in the paper by Narayama [19].

The main goal of the paper is to link the formal verification with the simulation of wireless sensor networks which are based on the SPIN protocol and the flooding technique. The obtained results allow us to examine this protocol and compare it to other solutions. The simulation checks the behaviour of the SPIN protocol after a given time. It is a practical examination not of a single node, but of the total sensor network after a defined amount of time units.

In Section 2 a short description of the SPIN protocol is given. Section 3 provides the specification and verification in the TLA+ of the SPIN protocol. In Section 4, we give the obtained simulation results. In the Section 5 we conclude with a summary and by some future work directions.

2 The SPIN Protocol and Its Activity

In this section, we present a short description of the SPIN protocol.

The SPIN protocol was introduced by Heinzelman in the paper [20] as an alternative solution to the flooding protocol [21,22]. In the flooding protocol all the nodes send the messages to their neighbours in the broadcast mode.

The flooding protocol generates some difficulties in use. Among others, the communication between nodes generates the broadcast storm – the sending of the same message to each node. There also occurs the so-called hidden terminal problem specifically for the class of the Carrier Sense Multiple Access (CSMA) method [23], where a node senses the medium before starting to transmit a packet. If the medium is found to be busy, the node defers its packet to avoid a collision and a subsequent retransmission. To overcome these problems the negotiation technique was introduced.

The negotiation technique is composed of three phases in which the message will be sent only to the receiver requested this message. The negotiation technique is as follows:

1. A node that has obtained new data *advertises* (packet ADV) the name of these data to its neighbors. These are the so-called *metadata*, that is a description of the data.
2. The receiver of an advertisement can compare it with its local knowledge and, if the advertised data are yet unknown, the receiver *requests* (packet REQ) the actual data. If the advertisement gives the already known data (for example, because it has been received via another node), the advertisement is ignored.
3. Only when a request for data is received, the data (packet DATA) is transmitted.

The basic operations of the SPIN protocol are given in Fig. 1. For the described SPIN protocol we can give a state diagram. It consists of four states, namely: *READY*, *CHECK*, *WAIT*, *RUN* (see Fig. 2). In the first state (*READY*) a node

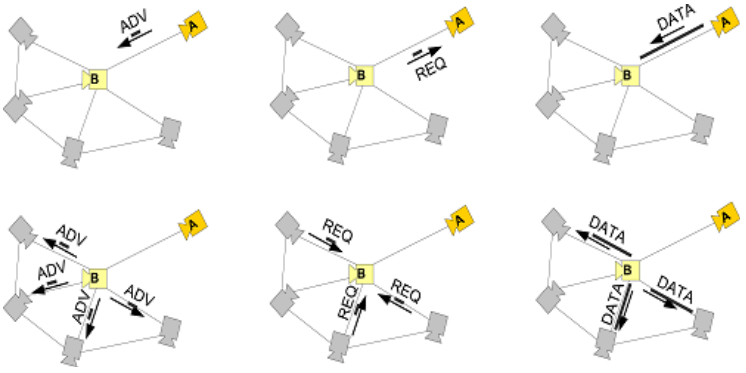


Fig. 1. The basic operations of the SPIN protocol

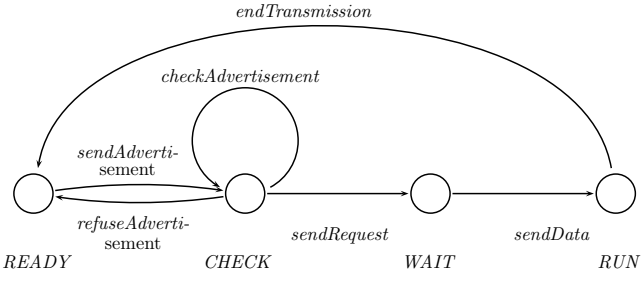


Fig. 2. State diagram for the SPIN protocol

waits for the end of the transmission or for the rejection of the advertisement. If the node has the requested data, it sends the ADV packet. Otherwise, a node remains in the *WAIT* state. When a node receives the ADV packet, it checks it (the *CHECK* state). It sends the REQ packet when the data are needed. After receiving the obtained data a node passes into the *run* state.

A full specification of the above given activity of the sensor node is presented in the section **actions** in TLA+ language (see Figs. 3 and 4). To this description a temporal formula is added, which guarantees that the actions are performed when they are accessible.

The activity of the SPIN protocol thus specified determines the formula which defines that each node capable of sending the data (transmitter set) can perform the action: *sendAdvertisement(n)*, *sendRequest(n)*, *refuseData(n)* and *sendData(n)*. The verification of this protocol is made with use the formula of liveness. This formula is defined in the TLA+ language with the help of the $WF\langle A\rangle$. It is equivalent to the theorem

$$\Box(ENABLED\langle A\rangle_v \Rightarrow \Diamond\langle A\rangle_v) \tag{1}$$

where $ENABLED\langle A\rangle$ is a predicate which is true if and only if action A is accessible. This condition assures that if action A is accessible, this action follows.

module

actions

$sendAdvertisment(n) \triangleq$
$\wedge\ n \in transmitters$
$\wedge\ node(n).state = DATA$
$\wedge\ node[n].energy \geq MessageTransmissionEnergy$
$\wedge\ node[n]' = [node[n]\ \text{EXCEPT}$
$!.energy = @ - MessageTransmissioEnergy$
$!.state = READY$
$!.destNode = \{\}$
$]$
$\wedge\ \forall r \in node[n].destNode :$
$\wedge\ node[r].state = READY$
$node[r]' = [node[r]\ \text{EXCEPT}$
$!.state = ADV$
$]$

$sendRequest(n) \triangleq$
$\wedge\ n \in transmitters$
$\wedge\ node[n].state = ADV$
$\wedge\ node[n].energy \geq MessageTransmissionEnergy$
$\wedge\ node[n]' = [node[n]\ \text{EXCEPT}$
$!.energy = @ - MessageTransmissionEnergy$
$!.state = WAITDATA$
$]$
$\wedge\ next \triangleq \text{CHOOSE}\ s : s \in node[n].destNodes$
$\wedge\ node[next]' = [node[next]\ \text{EXCEPT}$
$|.destNode = @ \cup n$
$!.state = REQ$
$]$

$refuseData(n) \triangleq$
$\wedge\ n \in transmitters$
$\wedge\ node[n].state = ADV$
$\wedge\ node[n].energy > MessageTransmissionEnergy$
$\wedge\ node[n]' = [node[n]\ \text{EXCEPT}$
$!.energy = @ - MessageTransmissionEnergy,$
$!.state = READY,$
$!.destNode = \{\}$
$]$

Fig. 3. The specification of the **action** section in TLA+ with the temporal formula (part 1)

module

$sendData(n) \triangleq$
$\quad \wedge\ n \in transmitters$
$\quad \wedge\ node[n].state = REQ$
$\quad \wedge\ node[n].energy > DataTransmissionEnergy$
$\quad \wedge\ node[n]' = [node[n]$ EXCEPT
$\qquad !.energy = @ - DataTransmissionEnergy,$
$\qquad !.state = READY,$
$\qquad !.destNode = \{\}$
$\qquad]$
$\quad \wedge\ \forall r \in node[n].destNode :$
$\qquad \wedge\ node[r].state = WAITDATA,$
$\qquad \wedge\ node[r]' = [node[r]$ EXCEPT
$\qquad\quad node[r].state = DATA$
$\qquad\quad]$

$program \triangleq \forall n \in transmitters :$
$\qquad \vee sendData(n)$
$\qquad \vee refuseData(n)$
$\qquad \vee sendAdvertisment(n)$
$\qquad \vee sendRequest(n)$
$fairness \triangleq$
$\quad \wedge\ \forall n \in Nodes :$
$\qquad \wedge\ node[n].state = DATA$
$\qquad \wedge\ WF_{var}(sendAdvertisment(n))$
$\quad \wedge\ \forall n \in Nodes :$
$\qquad \wedge\ node[n].state = METADATA$
$\qquad \wedge\ WF_{var}(sendRequest(n))$
$\qquad \wedge\ WF_{var}(sendRequest(n))$
$\quad \wedge\ \forall n \in Nodes :$
$\qquad \wedge\ node[n].state = REQ$
$\qquad \wedge\ WF_{var}(sendData(n))$

$Spec \triangleq Init \wedge \Box[program]_{var} \wedge fairness$

THEOREM $Spec \Rightarrow \Box Invariant$

Fig. 4. The specification of the **action** section in TLA+ with the temporal formula (part 2)

This condition is known as weak fairness and it guarantees that the specified system will be not stop in this state.

The states of this model have variables for the state of individual nodes and the state of the neighbouring nodes. All state variable have finite domains. The transition system defined by the sensor network's actions is also a finite state machine. This finite state machine specification is executable and can be used for simulation. An execution of this machine specification in given time of simulation provides answer to the model correctness.

3 The Simulation Study of the Wireless Sensor Network Specified by the TLA+ Language

In this section, we give the results of the simulation of the wireless sensor network specified by TLA+ language.

In our simulation the program in Java written by P. Downey [24] was used. This program was applied by R. Cardell-Oliver to the study of the performance evaluation of the wireless sensor network with the flooding protocol [25].

In our simulation scenario we studied the details of the behaviour of our network with both protocols, e.g. the flooding and the SPIN ones, for varied topologies of the network. The term "behaviour" defines here the temporal correctness of the improvement of the data transmission from the source to the destination node under different circumstances. The packet sizes were fixed at 38 bytes for each simulation run.

In the simulation we used the following topologies: mesh, which can be sparsely or thick, spine and band. These topologies are shown in Fig. 5. We also considered two densities of nodes in our landscape. The first one has 512 nodes randomly placed on a square sensor field of 1 000 m $\times$ 1 000 m ("sparsely" topology) at the begining of a simulation. The second sensor field consists of 1 024 nodes placed on the same square field ("thick" topology).

In the simulation we took into considerations various values of parameters of the CSMA method. It concerns the maximum backoff time, the base retry time, after which there will be a retransmission in the situation of the packet lack, and the maximum retry time – the maximum time after which take place a retransmission.

The obtained results of the simulation are given for all the studied topologies and parameters in Tables 1 and 2. The subject of our simulation was to determine the number of the sent messages, the number of the received messages, etc.

We conclude that the flooding protocol behaves correctly regardless of the network topology. In particular, we can see it in the thick mesh topology. In the flooding protocol the number of generated messages is equal to the number of nodes. For the SPIN protocol the number of generated messages is three times larger than the number of nodes (1 220 messages in the network with 400 nodes). It means that the overlay time of negotiation in the SPIN protocol grows if the radio coverage of each node raises higher. Thus interferences of radio signals are observed. In consequence the nodes must send the retransmission of the

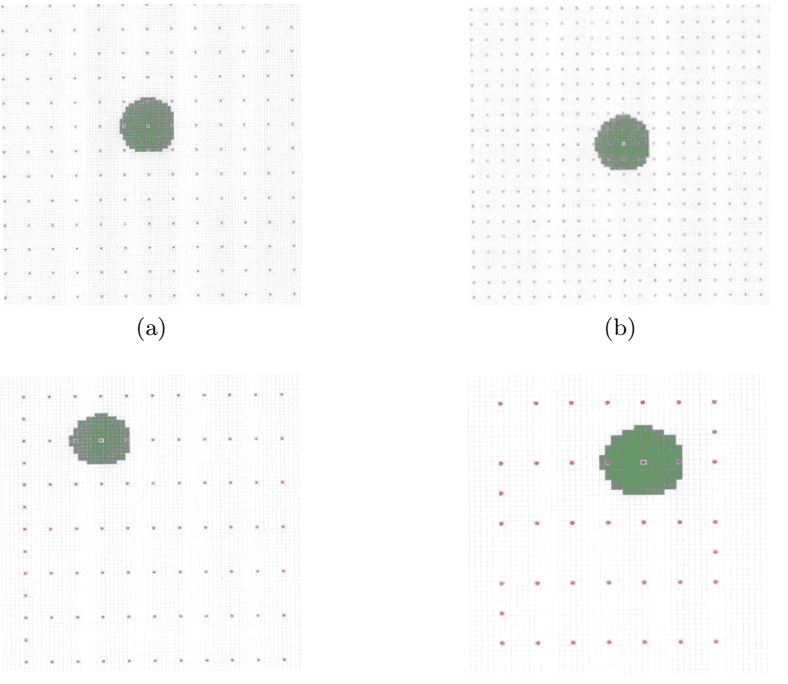


Fig. 5. (a) The "sparsely" mesh topology; (b) The "thick" mesh topology; (c) The "spine" topology; (d) The "band" topology

Table 1. Simulation results for sensor networks with the "sparsely" and "thick" mesh topology

	"Sparsely" mesh topology		"Thick" mesh topology	
	Flooding protocol	SPIN protocol	Flooding protocol	SPIN protocol
Width of mesh	100	100	100	100
Height of mesh	100	100	100	100
Thickness	0.125	0.125	0.2	0.2
Transmit power	79	79	79	79
Max backhoff	10	10	30	30
Base retry time	10	10	100	100
Max retry time	50	50	300	300
No. of nodes with data	169	169	400	400
No. of cycles	17	541	60	1287
No. of sent packets (messages)	0	577	0	1220
No. of sent packets (data)	91	173	400	411
No. of received packets	90	641	399	537
No. of discarded packets	1	109	1	1094

Table 2. Simulation results for sensor networks with the "spine" and "band" topology

	Spine topology		Band topology	
	Flooding protocol	SPIN protocol	Flooding protocol	SPIN protocol
Transmi power	59	59	59	59
Max backhoff	10	10	2	2
Base retry time	10	10	5	5
Max retry time	30	30	5	5
No. of nodes with data	250	250	39	39
No. of cycles	24	279	40	123
No. of nodes with data	0	188	0	78
No. of sent packets (data)	83	83	39	38
No. of received packets	82	126	38	63
No. of discarded packets	1	145	1	53

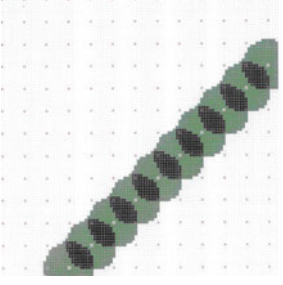


Fig. 6. The simulation of hidden terminal phenomenon in the sensor network with flooding protocol. The dark footprints are due to the interference.

ADV messages and the REQ messages. This phenomenon is not observed in the flooding protocol, in which in the case of the thick mesh the probability that the sent data are delivered farther grows.

The SPIN protocol guarantees that the data are sent to all the nodes of the network. Instead, in the flooding protocol the data are delivered only to 54% of all nodes. Moreover, in the flooding protocol the hidden terminal phenomenon is observed, which is characterized reciprocal by transmission between the nodes (see Fig. 6).

We also observed that in the flooding protocol the radio noise effectiveness of network increases in the case of a data sending. It is caused by the fact that the radio noise is overlaped on the transmission signal. Thus, the number of the disturbed signals is decreased and a greater number of nodes are not strayed. This phenomenon is shown in Fig. 7 in which it is observed that the additional noise does not exclude the transmission between the main nodes in the network.

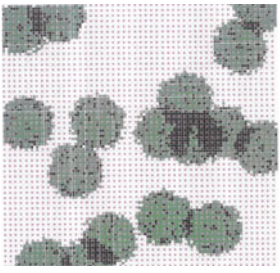


Fig. 7. The simulation of the sensor network with flooding protocol in case of interference. The dark footprints are due to the interference.

4 Conclusion

The simulation study of the behaviour of the wireless sensor network using the TLA+ language shows that this methodology can provide much valuable informations about the wireless sensor network. Here, we obtained the parameters of two protocols used in these networks with two protocols: the SPIN and the flooding protocols for given topologies and defined spatial density of nodes. It showed and also visualized two phenomena, namely hidden the terminal and the flooding of nodes by the messages. Also the overlay of noise temporarily increased the effectiveness of transmission in the wireless sensor network.

It is obvious that the above study can be substantial providing carried out with the help of the formal methods. In our future work will add other protocols of the wireless sensor network. We will also simulate other characteristics of these networks, such as mobility, throughput capacity, etc.

References

1. Lamport, L.: Specifying Systems. In: The TLA+ Language and Tools for Hardware and Software Engineers. Addison-Wesley, Boston (2003)
2. Akyildiz, I.F., Su, W., Sankarasubramaniam, Z., Cayirci, E.: Wireless Sensor Networks. Computer Networks 38, 393–422 (2002)
3. Xia, D., Vlajic, N.: Near-optimal Node Clustering in Wireless Sensor Networks for Environment Monitoring. In: Canadian Conference on Electrical and Computer Engineering 2006, Ottawa, Ont., Canada (2007)
4. Welsh, M., Chen, B., et al.: CodeBlue: Wireless Sensor Networks for Medical Care. In: Division of Engineering and Applied Sciences. Harvard University (2006)
5. Neumann, P.: Wireless Sensor Networks in Process Automation – Survey and Standardisation Activities. Automatisierungstechnische Praxis 50(3), 61–67 (2007)
6. Gutierrez, J.A.: On the Use of IEEE 802.15.4 to Enable Wireless Sensor Networks in Building Automation. In: Proc. of the IEEE 15th Int. Symp. on Personal Indoor and Mobile Radio Communications, vol. 3, pp. 1865–1869 (2004)
7. Osborn, K.: Sensor Finds. In: Ids Sources of Fire, Defense News, Technology Watch, February 12 (2007)

8. Schechter, E.: Sensing Trouble. In: Novel Approaches to Detecting Explosives at Standoff Ranges, Defense News, Technology Watch (2007)
9. Lamport, L.: The Temporal Logic of Actions. ACM Transactions on Programming Languages and Systems 16(4), 872–923 (1994)
10. Lamport, L.: TLA WWW pages (1996), http://www.research.digital.com/SRC/tla/tla.html
11. Abadi, M., Lamport, L.: An Old-Fashion Recipe for Real-Time. ACM Trans. on Programming Languages and Systems 16(5), 166–178 (1994)
12. Lamport, L., Paulson, L.C.: Should Your Specification Language Be Typed? ACM Trans. Programming Languages and Systems 16(3), 872–923 (1999)
13. Batson, B., Lamport, L.: High-Level Specifications: Lessons from Industry. In: de Boer, F.S., Bonsangue, M.M., Graf, S., de Roever, W.-P. (eds.) FMCO 2002. LNCS, vol. 2852, pp. 242–261. Springer, Heidelberg (2003)
14. Tasiran, S., Yu, Y., Batson, B.: Linking Simulation with Formal Verification at a Higher Level. IEEE Design and Test of Computers, 472–482 (2004)
15. Janowska, A., Janowski, P.: Verification of Estelle Specifications Using TLA+ (1998)
16. Mokkedem, A., Ferguson, M.J., deB Johnston, R.: A TLA Solution to the Specification of the RLP1 Retransmission Protocol. In: Fitzgerald, J.S., Jones, C.B., Lucas, P. (eds.) FME 1997. LNCS, vol. 1313, pp. 398–417. Springer, Heidelberg (1997)
17. Sacuta, A.D.: PN-3306: Radio Link Protocol 1 (ballot resolution draft). TLA Draft Standard TR45.3.2/95.02.28.03, Data Services Task Group of ANSI Accredited TIA TR45-3 (1995)
18. Kapus, T., Brezocnik, Z.: TLA-Style Specification of a Mobile Network. In: Proceedings 23rd Euromicro Conference: New Frontiers of Information Technology (EUROMICRO 1997), pp. 440–447. IEEE Computer Society, Los Alamitos (1997)
19. Narayama, P., Ruiming, C., Yao, Z., Zhi, F., Hai, Z.: Automatic Vulnerability Checking of IEEE 802.16 WiMAX Protocols through TLA+. In: 2nd IEEE Workshop on Secure Network Protocols, pp. 44–49 (2006)
20. Heinzelman, W.R., Kulik, R., Balakrishnan, H.: Adaptive Protocols for Information Dissemination in Wireless Sensor Networks. In: Proc. of the 5th Annual International Conference on Mobile Computing and Networking, pp. 174–185. ACM, Seatle (1999), http://citeseer.nj.nec.com/heinzelman99adaptive.html
21. Carzaniga, A.: Architectures for an Event Notification Service Scalable to Wide-Area Networks. Ph.D. Thesis, Politecnico di Milano, Milano, Italy (1999)
22. Carzaniga, A., Rosenblum, D.S., Wolf, A.L.: Design and Evaluation of a Wide-Area Event Notification Service. ACM Trans. on Computer Systems (TOCS) 19(3), 332–383 (2001)
23. Glisic, S.G.: Advanced Wireless Networks. 4G Technologies. John Wiley and Sons, Chichester (2006)
24. Downey, P., Cardell-Oliver, R.: Evaluating the Impact of Limited Resource on the Performance of Flooding in Wireless Sensor Networks. In: Proc. of the 2004 International Conference on Dependable Systems and Networks (2004)
25. Cardell-Oliver, R.: http://www.csse.uwa.edu.au/~rachel/

Interpretation of Dual Peak Time Signal Measured in Network Systems

Stanisław Wideł, Jarosław Flak, and Piotr Gaj

Silesian University of Technology, Institute of Informatics,
ul. Akademicka 16, 44-100 Gliwice, Poland
{jaroslaw.flak,piotr.gaj,stanislaw.widel}@polsl.pl
http://www.polsl.pl

Abstract. During a computer network activity there are many events which occur in certain moments of time with a given likelihood. The statistical aspect of this can be described by a probability density function of a random variable connected with the event time. The function in many cases has a specific feature which is not covered by any well known probability distributions. It has two maxima, in particular when time signal in network systems is measured. It is suggested that the function is a convolution of other ones. However, in order to simplify the probability modeling, a special distribution named *if* can be defined, especially for discrete variables. Interpretation of dual peak.

Keywords: distribution, pdf, pmf, if, multi-maximal, time, ping, peek.

1 Introduction

In this article the authors generally examine the time of execution of various processes in computer systems. When measuring these times, two phenomena were found during further analysis of the results. The time of execution of a given process is not the same in consecutive trials. It can be said that this is a 'distribution' of these times. The second phenomenon found by the authors is that in many cases the time of execution is not concentrated around one value but there are two peaks in these distributions. This is the effect of various execution paths of programs due to the execution of conditional statements. Distribution with this feature is called by authors an *if* distribution.

Such distribution can be used to better describe a computer networking system, which software is naturally based on caching or conditional program statements. Additionally, some examples of a simple sequence of command statements were found, e.g. writing the data to a system log or performing a ping transaction.

2 Industry Standard, Statistical Evaluation of Measurement Data

The industrial standard used for verification of server systems is a benchmark. It is the way of system verification by one synthetic factor. Such a measurement

A. Kwiecień, P. Gaj, and P. Stera (Eds.): CN 2010, CCIS 79, pp. 141–152, 2010.

method allows for comparison systems but has many imperfections. The first one is standardized system workload prepared to collect a stable result. It is expected that if the system A achieved a better result than the system B, the relation of the introduced measure will be kept during every next repetition of the test. However, if the server is exposed to real workload within a real computer network, one cannot exclude that the system B will be more efficient in the particular real application, in spite of a clear test indication pointing to the server A. One tries to resolve this defect of the tests based on a synthetic factor calculated from the mean value by extending the family of simulations. This, in the authors' opinion, complicates the problem of performance estimation instead of helping. After performing the research there are many results for the A and B systems, which are not correlated with the system's architecture until it is better recognized.

Many questions regarding server's performance [1,2] remain without answers. Based for instance on the test result, one is unable to claim how the modification of the B system, let us say the server's CPU exchange, will influence the network system performance. In time-limited systems the maximal measurement data values are used for analysis. Typically to estimate the performance, the measurements of the mean value of system parameters are used. Because of that, the measurement error is averaged out and the method becomes resistant to any measurement errors. The mean value is representative for the measurement data when the system is of a stationary type, but from the measurement practice point of view it turns out that it is not an assumption, which is simple to fulfill.

Let us assume for a given network system that there is a necessity of measure the response time R. A basic example of such a measurement using a ping command is presented in Fig. 1. The ping utility (iputils software packet) provides

```
ping -c 5 192.168.16.11
PING 192.168.16.11 (192.168.16.11) 56(84) bytes of data.
64 bytes from 192.168.16.11: icmp_seq=1 ttl=64 time=1.38 ms
64 bytes from 192.168.16.11: icmp_seq=2 ttl=64 time=0.241 ms
64 bytes from 192.168.16.11: icmp_seq=3 ttl=64 time=0.257 ms
64 bytes from 192.168.16.11: icmp_seq=4 ttl=64 time=0.265 ms
64 bytes from 192.168.16.11: icmp_seq=5 ttl=64 time=0.253 ms
--- 192.168.16.11 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4001ms
rtt min/avg/max/mdev = 0.241/0.480/1.388/0.454 ms
```

Fig. 1. Example of performance measurement using a ping command

the following basic statistical values (last line of output results in Fig. 1): `min` is the minimal value of the RTT time ($x_{\min}$), `avg` is the mean value (x), `max` is the maximal value ($x_{\max}$), as described in equations:

$$x_{\min} = \min(x_i) \tag{1}$$

$$\bar{x} = \frac{1}{n} \sum_{n}^{i=1} x_i \tag{2}$$

$$x_{\max} = \max(x_i) \tag{3}$$

where x_i are measured values of time. The values comes from basic statistical processing. Let us note that `mdev` (Fig. 1) is not the standard deviation (4) but is the mean deviation (5). The standard deviation σ is defined as the square root of the variance σ^2.

$$\sigma = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2} \tag{4}$$

The mean deviation (also called mean absolute deviation) is the mean of the absolute deviations of a set of data which contains mean values of the data. For a sample of the size n, the mean deviation is defined as [1]:

$$md = \frac{1}{n} \sum_{n}^{i=1} |x_i - \bar{x}| \ . \tag{5}$$

3 System Log as a Source Data

A typical method of collecting data about server's performance is a log analysis. In network systems usually a log file is a source of data to analyze the behavior of various processes. A network server log is treated as a measurement device for data acquisition of the execution time of a chosen operation. In the research [3] the authors have an existing networking system, which writes necessary parameters to the system data log. The authors want to acquire the execution time of data processed by a selected operation in the existing system. Let us define the system log as:

$$\begin{aligned} t_1 - o_1 \\ t_2 - o_2 \\ \vdots \\ t_L - o_L \end{aligned} \tag{6}$$

Let us define the difference between two time stamps in the following way:

$$\Delta_l = t_1 - t_{l-1} \tag{7}$$

A log is a pair of a time-stamp and operation name that was executed by the system. In the observation time T, the L number of log lines was collected in the order from 1 to L. Each operation can be classified as one set of operation type. The number of operation types from 1 to K is finite in the system. The

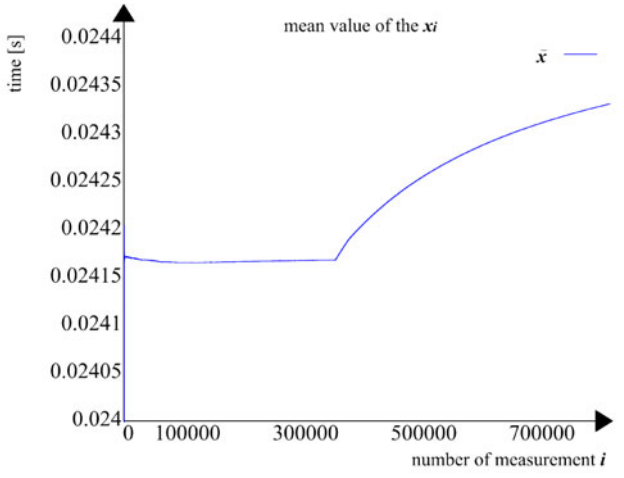


Fig. 2. Change of the mean value of the execution time x_i during the selected measurement process

assumption is that the log is well-detailed and descriptive, so the log entry of a given type obtains service only from one resource.

During the research conducted by the authors [4], measurement errors important for the analysis, entered by log records, were noticed on the network server. It turns out (Fig. 2, 3) that the server's activity, without any commands except writing timestamps to the log file, seems to be non-stationary. In a set up range (from $i = 0$ to about 360 000) the mean value varies and after that it should enter a steady state. Except of this, it can be noticed in the pictures 3 and 4 that the mean value jumps when iteration i is about 360 000 and 380 000 during the experiment. The second jump is better visible in Fig. 3. Because of the fact that floating of the mean value in the measurement process can be noticed (Fig. 2), it is necessary to better investigate the non-stationary nature of this process. During the analysis of the time of execution of processes (time-stamps) from the log file (Fig. 3) the problem of distribution of the characteristic *if* often occurs. Let us notice that for the description of the stationary process x_s a histogram was used, which indicates two maxima. Following the observation, a decomposition of the observed process can be done into the stationary process x_s as well as a shift of the process by the constant value A, which is changing during the experiment (Fig. 4). This is a way which describes the system work more precisely. The time signal $x(t)$ can be written as a multiplication of the function of the stationary process x_s and the constant value.

$$x(s) = A(t)x_s(t) \ . \tag{8}$$

By gathering data of the server's performance from the log file one can not take into consideration the writing procedure to the log. However, in this case all the data used during the analysis will be burdened with an error.

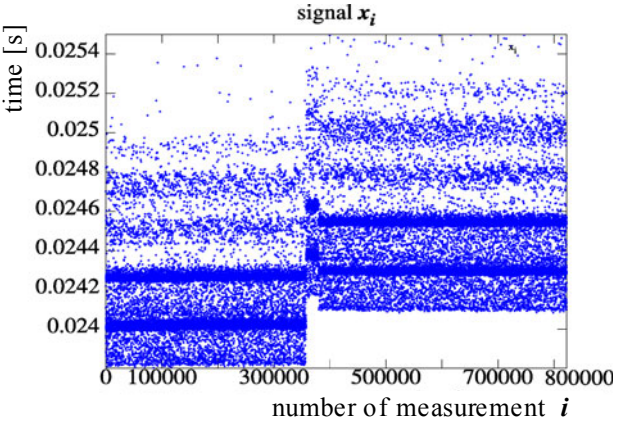


Fig. 3. Data value of the execution time x_i during measurement process

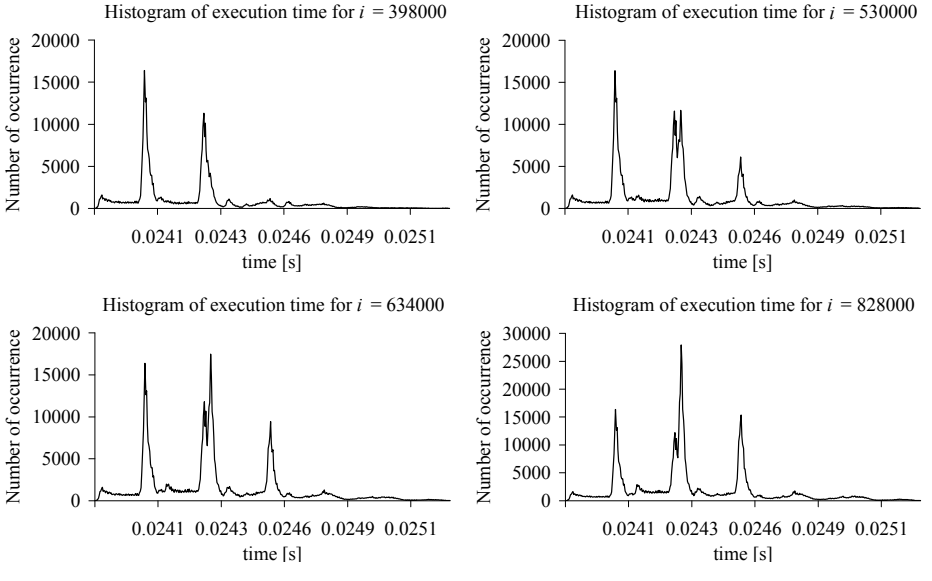


Fig. 4. Histogram of PrintToDebugLog execution time x_i for selected number of samples i

4 Applying Probability Mass Function for Computer Measurement

Let us extend the numerical characteristic that is computed from a sample of observations. Beside these basic statistical parameters (1)–(5), it is possible to

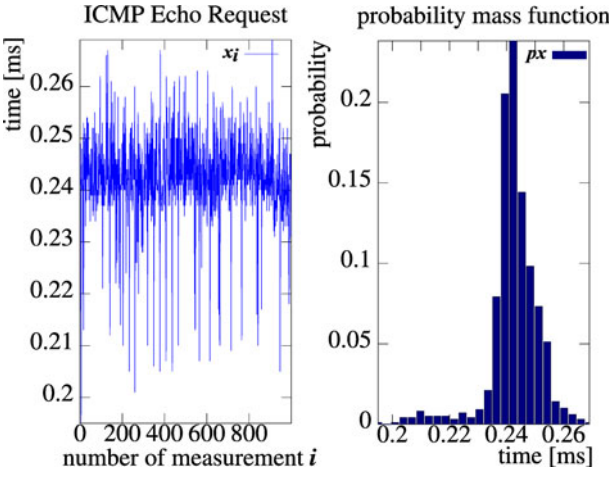


Fig. 5. ICMP Echo_Request measurement time series and its histogram

collect data and build histograms [5]. Examples of a measurement time series and its histogram are shown in Fig. 5. If a continuous $x(t)$ function exists, the probability density function can be defined as:

$$p_x \lfloor a \leq x(t) \leq b \rfloor = \int_a^b x(t)dt \ . \tag{9}$$

Sometimes the distribution function is used [6] to designate the probability density function (pdf). However, this term can be considered as the probability distribution function, or as the cumulative distribution function, or it may be the probability mass function rather than the density. So it is necessary to be aware of the meaning of this name. The probability mass function (pmf) specifies the probability that a discrete random variable is exactly equal to some value [7]. The pmf and the pdf are not the same. The values of the pdf which are defined only for continuous random variables do not represent the probabilities. Alternatively, the integral of the pdf over a range of possible values can be used, and thanks to that the probability of a random variable falling within that range can be shown.

5 *if* Distribution Basic Definition Form

Let us assume the existence of a statistically independent conditional statement x and operation y (Fig. 6). Operation y is responsible for writing data to a disk. If the buffer, filled with the data, is full, $x^{(2)}$ operation writes the data to the disk, else $x^{(1)}$ operation writes the data to the buffer in RAM. The times of execution

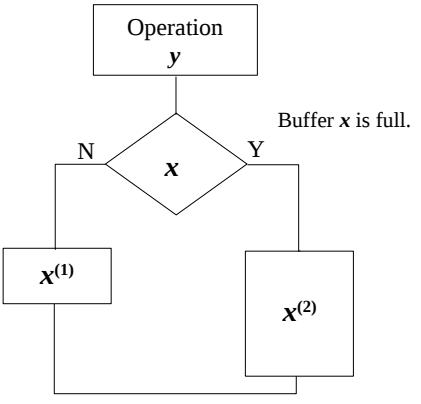


Fig. 6. Algorithm of program execution with a conditional statement x

of operations $x^{(1)}$ and $x^{(2)}$ are different. The time in the measurement process is defined only for discrete values of the independent variable, so the time is an independent variable. The signal belonging to the discrete time domain is called a discrete-time signal [8]. The independent variable of a mathematical object (sequence) that represents a discrete-time signal belongs to the set of integer numbers.

In a computer performance analysis [9,10], a discrete-time signal can be seen as generated by the system and defined in the discrete time domain, rather than seen as discrete-time signals obtained by sampling continuous-time signals. A discrete-time signal is mathematically represented (10) by a sequence of values x_i:

$$x(t) \rightarrow x_i, i \leq \infty \ . \tag{10}$$

The i-th sample of the sequence is the i-th value x_i in the sequence. The independent variable i belongs to the set of integer numbers, so x_i is mathematically defined only for integer values of i.

Let us define a probability mass function for a conditional statement x (Fig. 6). The value of the measurement $x^{(1)}(t)$ is the time of execution of an $x^{(1)}$ operation and it is a discrete signal $x_i^{(1)}$. Analogously, $x^{(2)}(t)$ is a discrete signal $x_i^{(2)}$. For a basic *if* distribution form, let us assume that all the results of the measurement are the same. This means that:

$$x_i^{(1)} = const \tag{11}$$

$$x_i^{(2)} = const \ . \tag{12}$$

Probability of the execution of the $x^{(1)}$ path can be denoted as $px^{(1)}$. The measurement value signal $x_i^{(2)}$ in the path $x^{(2)}$ has the probability $1 - px^{(1)}$. For $x^{(1)}$, $x^{(2)}$, the *if* distribution is defined by $px^{(1)}$, $px^{(2)}$, which are the probabilities of the execution of operations $x^{(1)}$, $x^{(2)}$ respectively (13).

for measurement $x_i^{(1)}, x_i^{(2)}$ *if* distribution is defined by
$px^{(1)}$, $px^{(2)}$ where $px^{(2)} = 1 - px^{(1)}$ (13)

The case when the formula (13) follows the conditions (11), (12), is a basic form of the *if* distribution. In other words, when the basic form signals $x_i^{(1)}$ and $x_i^{(2)}$ follow the formulas (14), (15), the basic form of the *if* probability mass function has two peaks, as shown in Fig. 7.

$$x_i^{(1)} = t_1 \tag{14}$$

$$x_i^{(2)} = t_2 \ . \tag{15}$$

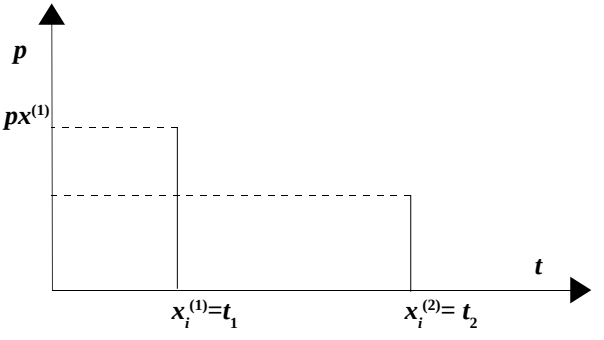


Fig. 7. Basic form of *if* probability mass function

In the example (Fig. 6) the operation y and the conditional statement x are executed together. Let z be a program executing both x and y operation. If a measurable result of the execution of $x(t)$ and $y(t)$ exists as independent discrete functions x_i and y_i, what will the z execution time be? If the z execution time is presented as the probability mass function pz, it will be a convolution [8] of the probability mass function px and py (16):

$$pz_j = px * py = \sum_{j=1}^{i} px_j py_{i-j} \tag{16}$$

where: px, py, pz are time probability mass functions (Fig. 5) as results of the execution x, y, z, px_i, py_i, pz_i are probabilities for the given interval i of px, py, pz. $x^{(1)}$ and $x^{(2)}$ are disjointed events and are dependent. However, if operation x consists of $x^{(1)}$, $x^{(2)}$ then the event is complete event. The x cumulative distribution function (CDF) is equal 1. The convolution pmf function $px * py$ are defined if x, y are independent. The example of the convolution is shown in Fig. 8.

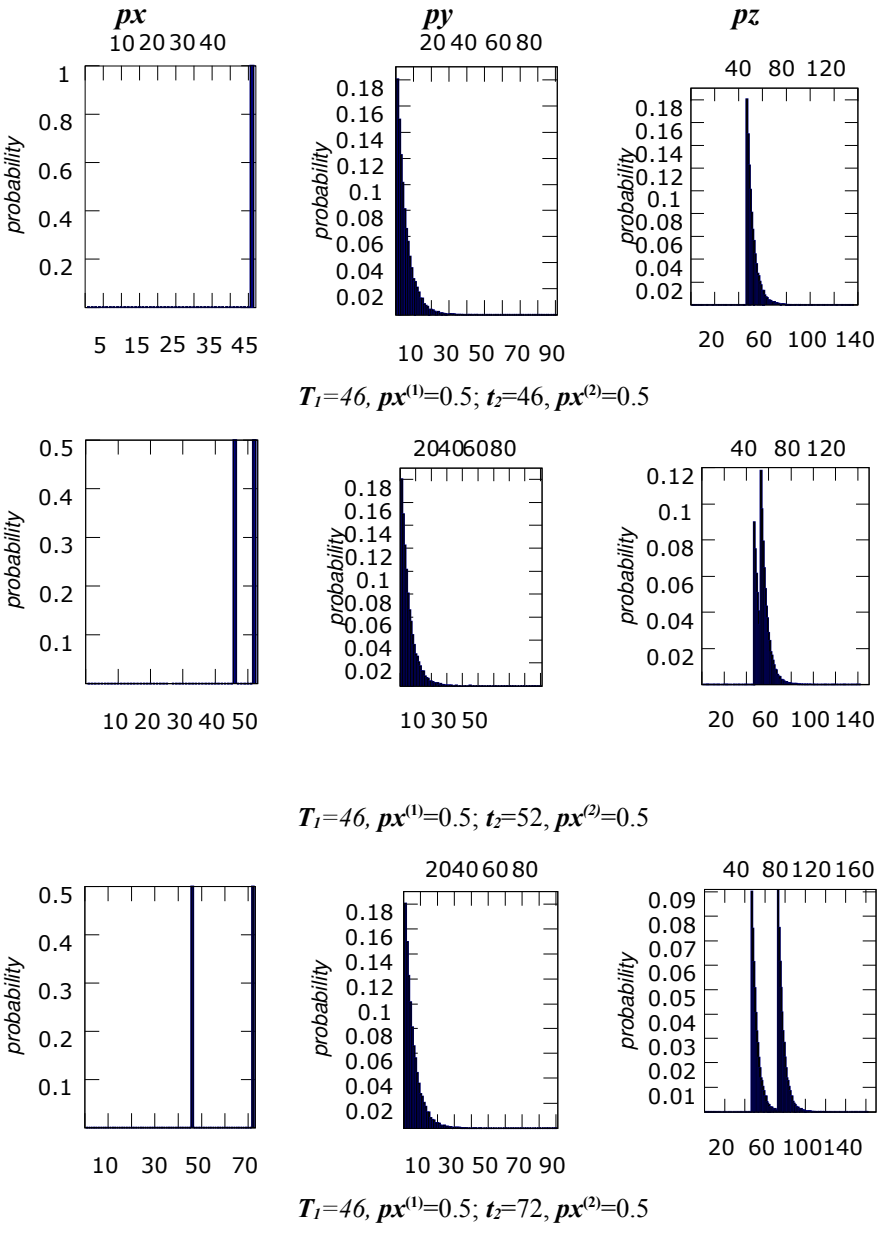


Fig. 8. Example of the convolution of probability mass functions *if* and exponential distribution

6 Extended Definition of the *if* Distribution

The $px^{(1)}$, $px^{(2)}$ are time probability mass functions of measurements that are results of the execution of $x^{(1)}$, $x^{(2)}$. In general $px^{(1)}$, $px^{(2)}$ can receive any probability mass function distribution.

$$\begin{gathered}\text{for measurements } x_i^{(1)},\, x_i^{(2)} \\ \textit{if} \text{ distribution is defined by } px^{(1)},\, px^{(2)} \text{ where} \\ \sum_{i=1}^{m} px_i^{(2)} = 1 - \sum_{i=1}^{j} px_i^{(1)} \quad .\end{gathered} \tag{17}$$

The $px_i^{(1)}$, $px_i^{(2)}$ are probabilities for a given interval i of $px^{(1)}$, $px^{(2)}$.

In case the number pt_1 of the $x^{(1)}$ operation is known, the probability of the execution of the $x^{(1)}$ operation can be calculated. Then, for defining the px function it is possible to use pt_1 probability and a given g distribution instead of $px^{(1)}$.

For a signal $x_i^{(1)}$ it is possible to find a function g, which, after multiplication by pt_1, gives $px^{(1)}$ (17). It is not necessary to use a convolution because pt_1 is a single constant value.

$$px^{(1)} = pt_1 \cdot g \quad . \tag{18}$$

It is obvious that:

$$pt_2 = 1 - pt_1 \quad . \tag{19}$$

Analogously, for $x_i^{(2)}$ it is possible to find a function h, which multiplicated by pt_2, gives $px^{(2)}$ (20).

$$px^{(2)} = pt_2 \cdot h \quad . \tag{20}$$

7 No Software-Based Process with *if* Distribution

The dual peak function in this article was defined as one related to the execution of a conditional statement – this is why its name is the *if* distribution. However, it must be noted, that the physical interpretation of the process by means of the *if* function is not only confined to systems based on the programming execution of conditional statements.

As a practical example, let us analyze the result of the measurement of round trip time on the empty network from one server and two workstations (Fig. 9). For this measurement experiment three hardware identical computers with the same CentOs (Red Hat) operating system were prepared. From one computer (server) the 'ping' command was executed and the round-trip time (RTT) result was logged to two other computers (workstations).

The real estimation of $x^{(1)}$ and $x^{(2)}$ can be changing over the time and is depended on various parameters. The values depend on load of a network. In the example we want to show that characteristic distribution (named *if* is not only connected with execution of conditional command). In the measurements

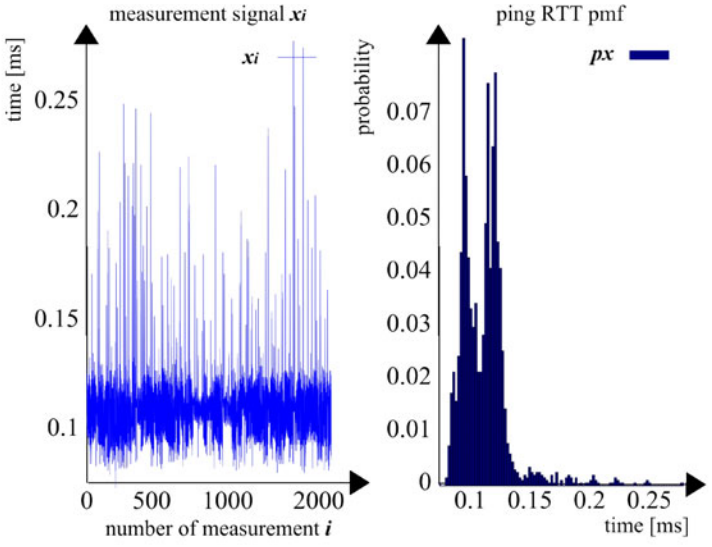


Fig. 9. Example of the measurement of round-trip time in a network with one server and two workstations

with normal load, we expected the same results for both stations. However, the difference was constant for all experiments. In this case the diference between $x^{(1)}$ and $x^{(2)}$ does not describe the process but rather depicts its variation area.

8 Conclusions

The performed research of network systems show that the measurement values of system activity, which before the experiment appeared to be simple and have the character of a constant value, have a much more complex statistical description. Writing to the server's log file was selected as the analysis base, because of set of data registered as the real workload and replies of the server. The process measurements were performed on the network server. The probability mass function was used instead of the mean value. The mean value was changing during the experiment and its usage was less accurate than the probability mass function. A description by the probability mass function is more accurate in case of the experiments made, because the process which was non-stationary can be described, in some cases, by a stationary process and a constant factor value, changes of which are easy to detect. The analysis of the stationary element x_s leads to a description which is defined as a conditional statement. So a simplified and complex process definition was introduced and it was named the *if* distribution. A statistical description by means of the *if* distribution can be used not only for systems implemented with conditional statements in programs, but also in many hardware based cases in the analysis of networks' behavior.

References

1. Czachórski, T.: Modele Kolejkowe Systemów Komputerowych. Wydawnictwo Politechniki Śląskiej, Gliwice (1999)
2. Lazowska, E.D., Zahorjan, J.G., Graham, S., Sevcik, K.C.: Quantitative System Performance. In: Computer System Analysis Using Queueing Network Models. Prentice-Hall, Inc., Englewood Cliffs (1984)
3. Wideł, S., Machniewski, J., Fiuk, M.: Wyznaczanie czasu wykonania operacji na podstawie dziennika systemowego serwera. In: Wysokowydajne sieci komputerowe. Nowe technologie, WKŁ (2005)
4. Wideł, S., Machniewski, J.: Measurement and data acquisition of execution time from application log. In: 16th Polish Teletraffic Symposium, Łódź, Poland, September 24-25 (2009)
5. D'Antona, G., Ferrero, A.: Digital signal processing for measurement systems: theory and applications. Springer, Milano (2006)
6. Mathworld: Mean deviation, http://mathworld.wolfram.com
7. Mason, R.L., Gunst, R.F., Hess, J.L.: Statistical Design Analysis of Experiments, 2nd edn. Wiley, Hoboken (2003)
8. Salkind, N.J.: Encyclopedia of Measurement and Statistics. SAGE Publications, Kansas (2007)
9. Lipsky, L.: Queuing Theory: A Linear Algebraic Approach. Springer, Connecticut (2009)
10. Menasce, D.A., Almeida, V.A.F., Dowdy, L.W., Dowdy, L.: Performance by design: computer capacity planning by example. Prentice Hall PTR, New Jersey (2004)
11. de Sá, J.P.M.: Applied statistics: using SPSS, Statistica, MATLAB and R. Springer, Heidelberg (2007)
12. Mattuck, A.: Introduction to analysis. Prentice Hall, Cambridge (1998)

Evaluation of IP Transmission Jitter Estimators Using One-Way Active Measurement Protocol (OWAMP)

Przemysław Skurowski[1], Robert Wójcicki[1], and Zbigniew Jerzak[2]

[1] Silesian University of Technology, Department of Computer Science, ul. Akademicka 16, 44-100 Gliwice, Poland
{przemyslaw.skurowski,robert.wojcicki}@polsl.pl
http://inf.polsl.pl
[2] SAP Research Dresden, SAP AG
zbigniew.jerzak@sap.com
http://www.sap.com

Abstract. Network transmission with real-time constraints appears to be very sensitive to the jitter – short time packet delivery delay. In this paper we describe several methods for estimating that phenomenon. We perform quantitative and qualitative analyses for the gathered data using modern measurement tool called OWAMP (one way ping) and stratum 1 Network Time Protocol (NTP) time server.

Keywords: jitter estimation, network delay, estimator, multimedia transmission, distributed systems.

1 Introduction

1.1 Background

Reliability of the data transmission plays crucial role in many areas of computer networks. Transmission delay and its variability called 'jitter' is one of the most important parameters characterizing networks' transmission. Precise estimation of these parameters might be especially important in areas such as publish/subscribe systems or multimedia transmission (VoIP).

Transmission of multimedia content in its most advanced technological solutions is performed using streaming paradigm. The data is not stored locally but consumed as it comes with a small delay protecting against jitter which might appear as transmission 'hiccup'. In case of non-interactive transmission like Video-On-Demand (VoD) or even live broadcasts the problem is less troublesome due to large de-jittering buffers (up to several seconds) that protect well against jitter. More demanding case is the VoIP (Video or Voice over IP) interactive transmissions. Such one way transmission delay (latency) depending on the target application and a class of service quality should be less than 150 ms for high quality services (class 0) and less than 400 ms for worse cases (class 1) [1]. In

A. Kwiecień, P. Gaj, and P. Stera (Eds.): CN 2010, CCIS 79, pp. 153–162, 2010.

such cases precise estimation of jitter is important: its results should be conservative enough to buffer enough packets to support uninterrupted streaming and on the other side it should be small enough to ensure interactive transmission.

In this study authors decided to verify usability of jitter estimators using OWAMP (*One-Way Active Measurement Protocol*) [2] tool that allows for precise estimation of packets one-way transmission delays. Prior to OWAMP reliable measurement of transmission delays was both technological and organizational problem. Due to the impossibility of precise synchronization of distributed clock it was necessary to provide GPS calibrated clocks to both remote and local machines. There was also a need to prepare special software just for sending packets containing only timestamps. Therefore, ICMP echo (ping) requests were used to gather data for the analysis of jitter estimators. As the influence of transmission delays is proportional to the number of packet hops and due to the fact that ICMP echo is often treated by the routers as second class traffic the ICMP echo approach could not be treated as a real network delay measurement. Nowadays, there exist several publicly available [3] OWAMP PMPs (*Performance Measurement Point*) therefore it is possible to gather real network characteristics.

1.2 Jitter Classification

According to [4] in packet switching networking terms jitter definition is just a simple packet delay variation. Clark [5] identified three basic kinds of jitter: contant, transient and short term delay variation. There also exists one more phenomenon – slow delay change which is not jitter in fact but it is an obstacle that dejitter buffers has to deal with. These types and their root causes are as follows (see Fig. 1):

- constant jitter – present in flawless transmission with roughly constant packet to packet delay variation,
- transient jitter – where single packet is significantly delayed to the others in stream. It is observed in numerous cases and it has numerous reasons like routing table up-dates, LAN congestion, router packet scheduling, route flapping and others,
- short term delay variation – occurring when a burst of packets has increased trans-mission delay. It is usually connected with access link congestion or route change,
- slow delay change – appearing in graph as a ramp like characteristics – Clark also connects this phenomenon with access link congestion.

2 Jitter Estimators

There are various estimators described in documents of two interested institutions IETF and ITU. They can be divided into two main groups PDV (packet delay variation) related to some absolute reference value and IPDV (inter packet delay variation) where as a reference packet there is used preceding one. Additionally there is also used simple moving range statistics or a moving distance between selected percentiles.

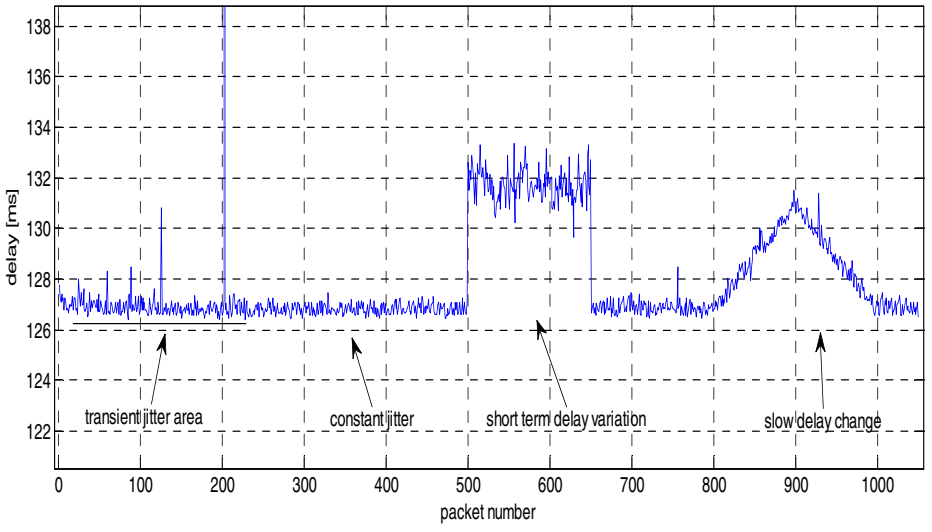


Fig. 1. Types of jitter

2.1 Inter-packet Delay Variation

The basic form of this measure appears in two variants: simple IP Delay Variation, IPDV [6] and Mean Packet to Packet Delay Variation, MPPDV [5]:

$$IPDV(n) = t(n) - t(n-1) \tag{1}$$

$$MPPDV(n) = \text{mean}\left(|t(n) - t(n-1)|\right) \tag{2}$$

where: $t(n)$, $t(n-1)$ is transmission delay of n^{th} and n^{th-1} packet, in case of $MPPDV$ mean value is computed for the most recent 16 packets.

2.2 RTP/RTCP Inter-packet Delay Estimator

In the family of IETF multimedia protocols transmission of audiovisual data is done using a pair of protocols RTP/RTCP [7]. The RTCP protocol reports among the others current jitter of RTP transmission. The calculated jitter is based on exponentially weighed moving average (EWMA) inter packet delay difference and it is computed according to the following formula:

$$J(n) = \frac{1}{16}\left|D(n)\right| + \frac{15}{16}J(n-1) \tag{3}$$

where: $J(n)$, $J(n-1)$ is current and previous jitter estimate, $D(n)$ is current packet delay change computed as difference of transmission time of two packets:

$$D(n) = t(n) - t(n-1) = (R(n) - S(n)) - (R(n-1) - S(n-1)) = \\ (R(n) - R(n-1)) - (S(n) - S(n-1))$$

where: $t(n)$, $t(n-1)$ is transmission delay, $R(n)$, $R(n-1)$ is arrival time and $S(n)$, $S(n-1)$ is timestamp (time of sending) of n^{th} and n^{th-1} packet.

2.3 Packet Delay Variation

There exist two variants of measure that refers to estimated local mean [5], [8]. Simple Mean Absolute Delay Variation, MAPDV, (Equation (4)) and its more sophisticated version MAPDV2 (Equation (5)).

$$MAPDV = \text{mean}\left(|t(n) - a(n)|\right) \tag{4}$$

where: $t(n)$ is current packet transmission delay, $a(n)$ is nominal (average) transmission time. One can easily notice that this approach requires prior knowledge of default transmission time, which means using local mean or median estimate to adopt it to short term jitter evaluation. The more complicated measure is MAPDV2 regarding explicitly short term average transmission delay. It is computed according to following formula: estimate a mean delay $a(n)$ using Jacobson's estimator [9] with gain set to 1/16 as a first step

$$a(n) = \frac{1}{16}t(n-1) + \frac{15}{16}a(n-1)$$

where: $t(n-1)$ transmission delay of recent packet; $a(n)$, $a(n-1)$ new and former estimate of mean transmission delay. In next step such approximate value is used in following computations given below as meta code for last 16 packets:

```
for (i=n-16; i<n; i++)       // n - current packet number
  if t(i)>a(i)                 // i - former packet numbers
    P(i)=P(i)+(t(i)-a(i))    // positive deviation
  elseif t(i)<a(i)
    N(i)=N(i)+(a(i)-t(i))    // negative deviation
  end if                     // if t(i)==a(i) do nothing !!!
end for
```

In a third step MAPDV2 is calculated as sum of mean values of positive and negative delays of 16 recent packets:

$$MAPDV2 = \text{mean}(P(i)) + \text{mean}(N(i)) \ . \tag{5}$$

2.4 IPDV and Relatives

Another existing solution for estimating a jitter is moving range statistic. To the author's knowledge, there are two similar solutions: one defined by ITU [10] called also IPDV (IP Delay Variation) given with formula:

$$IPDV_{\text{ITU}} = IPTD_{\text{up}} - IPTD_{\text{min}} \tag{6}$$

where: $IPTD_{\text{min}}$ is minimum IP transmission delay, $IPTD_{\text{up}}$ is some upper percentile (99.9^{th} is commonly used) thus $IPDV_{\text{ITU}}$ represents variability range of transmission delay. The other approach $IPDV_{\text{OWAMP}}$ to this proposal is used in OWAMP [3] implementation where client evaluates jitter as difference between

median and upper percentile but there are used $IPTD_{\min}$ – median value of IP transmission delay and as $IPTD_{\text{up}}$ transmission delay 95^{th} percentile.

Results obtained by using such estimators are affected by the moving window length for which the statistics are calculated. If the length of window is too short, the resulting values would be bound to few recent packets so predictor wouldn't exhibit any generalization. Otherwise if it is too long, it couldn't be used for predicting momentary value but would give general overview characteristics. Window of 200 ms used in VoIP applications seems to be a good compromise.

2.5 SMPDV

The SMPDV is an acronym of Switched Measure Packet Delay Variation estimator [11]. The concept is somewhat similar to classical EWMA AR-like estimator (Equation (3)). It is given using formula:

$$J(n) = A|D(n)| + B|D(n-1)| + RJ(n-1) \tag{7}$$

where $J(n)$, $J(n-1)$ are current and preceding SMPDV estimate, and $D(n)$, $D(n-1)$ are current and previous packet delay values, A, B, R are model coefficients fit experimentally having values 1/2, 1/16, 7/16 respectively.

The key fact is that the estimating of jitter is based on larger value chosen from two packet delay variation values as given below:

$$D(n) = \max\{D_{\text{ap}}(n), D_{\text{pp}}(n)\} \tag{8}$$

where: $D(n)$ is current delay value, $D_{\text{pp}}(n)$ is inter-packet delay, $D_{\text{ap}}(n)$ is absolute packet delay with reference to some estimate of base transmission time. They are described with following equations:

$$D_{\text{pp}}(n) = |t(n) - t(n-1)|\,,\ D_{\text{ap}}(n) = |t(n) - a_{\text{m}}(n)|$$

where: $t(n)$, $t(n-1)$ are transmission delays of current and previous packets, $a_{\text{m}}(n)$ is base transmission time estimate over some of past transmission delays which is median estimator over last 16 packets:

$$a_{\text{m}}(n) = \text{median}\,\{t(n), t(n-1), \ldots, t(n-15)\}$$

where: t(n-1), t(n-2), t(n-15) are transmission delay values of 16 previous packets.

3 OWAMP

OWAMP protocol and server are modern tools designed to measure network latencies. Since common ICMP echo packets (ping) is often classified by routers as low priority traffic so it cannot be used as reliable network performance measurement tool. Also conclusions about one way transmission on the round trip delay basis seems to be questionnable due to route asymmetry and changes [12]. General idea of OWAMP is based on sending as packet contents 64 bit long unsigned

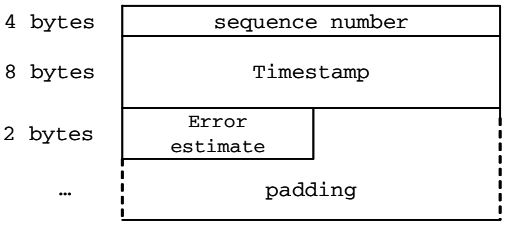


Fig. 2. Basic OWAMP unencrypted testing packet format

timestamps (see Fig. 2) derived from sender's well synchronized clock. Receiver complements packets with timestamps of arrival times from its own clocks that should be also well calibrated. It is obvious that the clock calibration is crucial for the precise OWAMP measurements. It is therefore recommended to deploy within the same computer a NTP server having at least 4 peer synchronization sources (or good reference, e.g., atomic clock) that would continuously calibrate the clock. The 64 bit long timestamp consists of two parts – integer and fractional. High-order 32 bits are integer part counting seconds since January 1st, 1900, low-order 32 bits are fractional part of second with theoretic precision up to $1/(232-1) \approx 2.33\ 10^{-10}$ s. Additional noteworthy information is that basic OWAMP packet as depicted in Fig. 2 has minimal size of 14 bytes meanwhile its encrypted form has 32 bytes.

In OWAMP system there are two parties involved in the testing procedure (Fig. 3) – the client and the server which results in two main usage scenarios that differ in the direction of the test packets transmission: the test packets are sent from client (owping) to server (owampd) – after the test transmission the server sends reports through the control connection containing send and arrival timestamps for each packet (among the others); test packets are sent on

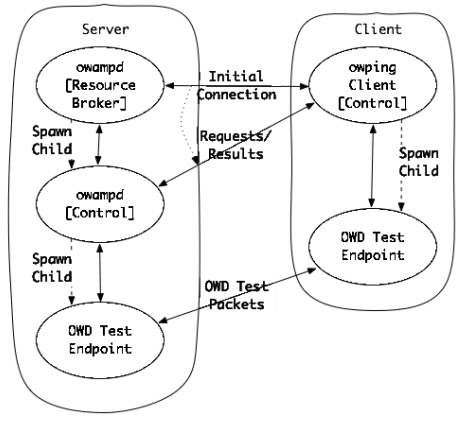


Fig. 3. OWAMP architecture [3]

client's request from server so it can compare their arrival time with the original timestamp included with the packet.

In opposite to usual ping, OWAMP is not commonly installed at the Internet connected hosts as it is rather specialized measurement tool but according to OWAMP promoting website [3] there are numerous PMPs available in the Internet. Very few of them allow for a free access, it is usually required to obtain AES keys or set up a ser account to be created on the OWAMP server. So usually it is necessary to have contact to the PMP administrator to get access to it.

4 Experimental Setup and Data Sets

To collect data sets to deal with jitter estimators the Compaq PIII host running FreeBSD OS has been used. The machine was running GPS driven NTP server. OWAMP client and server was installed on that machine as well. To simulate real traffic we decided to model VoIP transmission using following parameters: the very basic G.711 codec was used where the data is sampled at 8 kHz and it is packetized into 20 ms frames so there are transmitted 50 RTP packets per second, each containing 160 bytes of data plus additional overhead related to IP/UDP/RTP protocols stack headers – overall packet size is 196 bytes. To emulate real network workload it was necessary to set the OWAMP packet padding to fit the size of real transmission, which was straightforward as OWAMP uses the same lower layer protocols as VoIP transmission. Since IP/UDP/OWAMP packet size is 38 bytes we used 158 bytes padding.

Traces are 60 seconds (3000 packets) logs of two way OWAMP transmissions from and to the servers – as most of PMPs were inaccessible to us, we used the only open one in Korea (134.75.29.10) and we set up one in France (188.165.195.68) by courtesy of one of the hosting companies. The data was gathered on four successive days Saturday-Tuesday (twice during workdays) so there were 24 data sets collected. The data sets were manually classified (see Table 1) on a basis of observed episodes of types of jitter and relation between whole set $IPDV_{ITU}$ to the target inter packet offset (ΔT=20 ms). We observed the following classes:

- low variance constant jitter ($IPDV_{\mathrm{ITU}}/\Delta T < 0.05$) with no or few transient jitter episodes,
- low variance constant jitter ($IPDV_{\mathrm{ITU}}/\Delta T < 0.05$) with several episodes of transient jitter,
- moderate variance constant jitter ($0.05 < IPDV_{\mathrm{ITU}}/\Delta T < 0.2$) with numerous episodes of transient jitter,
- heavy variance ($IPDV_{\mathrm{ITU}}/\Delta T > 0.2$) with short term delay changes,
- low and moderate variance constant jitter sets with few episodes (1–3) of transmission jamming – packets bursts delayed above 400 ms and arriving almost at once after lag were considered to be lost so data sets were then reclassified on basis of their proper transmission.

Table 1. Test sets classification

	from Korea	to Korea	from France	to France
Saturday	B	A	A	A
Sunday	A	A	A	A
Monday 9:30	A	E(A)	A	A
Monday 12:30	D	C	D	D
Tuesday 9:30	C	E(C)	E(B)	C
Tuesday 12:30	C	C	C	B

5 Estimators Evaluation

As a fundamental criterion for judging efficiency of estimator we assumed the difference between real and predicted transmission delay. Such approach might be too arbitrary so we decided to consider also a number of underestimated packets and mean gap between underestimated and real delay value. The best results was highlighted in each table.

5.1 Predictor Precision

Predicted delay was calculated using formula:

$$P(n) = J(n) + a(n) \tag{9}$$

Mean absolue difference was calculated as:

$$E = \frac{1}{N} \sum_{n=1}^{N} |P(n) - D(n+1)| \tag{10}$$

where: $P(n)$ – predicted delay value, $J(n)$ – predicted jitter value, $D(n+1)$ – corresponding real transmission delay, $a(n)$ – running central tendency measure. In our case as $a(n)$ we decided to use moving median over last 16 packets as in our former research [11] it appeared to outperform simple moving average or EWMA. Results grouped by dataset class are presented in the Table 2.

Table 2. Mean absolute prediction error [ms] per data set in group

Group	MPPDV	RTCP	MAPDV	MAPDV2	$\text{IPDV}_{\text{ITU}}/2$	$\text{IPDV}_{\text{OWAMP}}$	SMPDV
A	0.410	0.410	**0.360**	0.543	0.555	0.742	0.446
B	0.475	0.476	**0.416**	0.612	0.645	0.884	0.517
C	1.368	1.374	**1.250**	1.960	1.872	2.724	1.502
D	3.802	3.774	**3.508**	5.090	4.849	6.105	4.199
E	7.619	7.076	10.007	11.881	10.494	12.363	**6.378**
All	1.987	**1.920**	2.196	2.907	2.689	3.401	1.936

Surprising fact is that relatively simple predictor MAPDV appeared to have better performance than sophisticated ones in all but three cases. The only

notable difference is for those sets where transmission jams were observed in such cases SMPDV appeared to be the most exact one. One should remember that simple mean absolute difference gives just rough view on efficiency of the predictors so there were required in-depth considerations on the underestimated packet delay.

5.2 Negative Cases Analysis

Since absolute difference between predicted and real delay value give just brief view to estimators performance we decided to analyze also performance in negative cases where predicted delay was underestimated. We measured it in two ways: as a number of underestimated packets and an average difference for underestimated packet. Number of underestimated packets is shown in Table 3.

Table 3. Mean number of underestimated packet delays per data set in group

Group	MPPDV	RTCP	MAPDV	MAPDV2	$IPDV_{ITU}/2$	$IPDV_{OWAMP}$	SMPDV
A	440.444	485.667	668.778	312.000	330.778	**261.556**	512.333
B	430.500	472.500	636.000	319.500	339.000	**265.000**	531.500
C	510.667	555.167	685.833	366.167	423.500	**280.333**	598.500
D	557.667	626.333	741.333	358.333	424.000	**304.333**	558.000
E	**214.333**	512.667	612.667	328.000	407.000	280.667	579.667
All	436.750	519.542	669.708	331.625	374.667	**274.292**	547.625

Besides the number of incorrectly estimated packets there is important question how 'wrong' was the prediction therefore we calculated mean prediction difference for underestimated packets. The resulting values are shown in Table 4. The best (minimal) values are highlighted.

Table 4. Mean difference for underestimated packet delays per data set in group [ms]

Group	MPPDV	RTCP	MAPDV	MAPDV2	$IPDV_{ITU}/2$	$IPDV_{OWAMP}$	SMPDV
A	0.1290	0.1113	**0.1011**	0.1222	0.1177	0.1229	0.1048
B	0.1477	0.1414	**0.1293**	0.1556	0.1466	0.1459	0.1303
C	0.5108	0.4606	0.4341	0.4712	0.4484	0.4701	**0.4131**
D	0.9764	0.9304	0.9300	0.9088	0.8775	**0.8739**	0.8782
E	19.1802	7.2951	5.0090	6.9569	7.7705	10.4780	**2.0139**
All	2.7212	1.2066	0.9070	1.1737	1.2614	1.6081	**0.5237**

6 Conclusions

We proposed the testing procedure and evaluated the performance of a number of jitter estimators. The test was performed against the test data gathered using OWAMP server using the scenario as close to real life application as it was possible to achieve.

Obtained results are ambiguous. The minimal mean absolute differences are observed for simple $MAPDV$ predictor except the situation where occurs transmission jamming – in such cases $SMPDV$ seems to be the most precise predictor. Overall best performance was achieved for EWMA used in RTCP. The worst results were obtained for $IPDV_{\text{OWAMP}}$ but on the other hand it seems to be the most protective as it underesti-mates the least number of packets but the cost is in the fact that its precision is the lowest one due to relatively large and protective overestimation margin. Finally, when taking into account not only the number of underestimated packets but also the size of that underestimation then it appears that $SMPDV$ seems to be the most reliable.

Summarizing, its hard to point a universally best estimator. Choice should be made depending on the application area. Both the data and procedure are reusable it is possible to perform such experiment using other data and other estimators. Our further research in this area will focus on scenario based analyses using synthetic data sets to test behavior and properties of estimators in case of specfific jitter cases.

Acknowledgments. This work has been partially supported by the Polish Ministry of Science and Higher Education grant number N N516 375034.

References

1. ITU-T Recommendation G.114: One-way transmission time. ITU (2003)
2. Shalunov, S., et al.: A one-way Active Measurement Protocol (OWAMP). IETF RFC4656 (2006)
3. One-Way Ping, http://www.internet2.edu/performance/owamp/
4. Morton, A., Claise, B.: Packet Delay Variation Applicability Statement, IETF RFC5481 (2009)
5. Clark, A.: Analysis, measurement and modelling of Jitter. ITU-T Delayed Contribution COM 12 - D98, ITU, Geneva (2003)
6. Demichelis, C., Chimento, P.: IP Packet Delay Variation Metric for IP Performance Metrics (IPPM). IETF RFC3393 (2002)
7. Schulzrinne, S., et al.: RTP: A Transport Protocol for Real-Time Applications. IETF RFC3350 (2003)
8. ITU-T Recommendation G.1020: Performance Parameter Definitions for Quality of Speech and other Voiceband Applications Utilising IP Networks. ITU (2003)
9. Jacobson, V.: Congestion Avoidance and Control. In: Proc. ACM SIGCOMM 1988, pp. 314–329 (1998)
10. ITU-T Recommendation Y.1540: Internet protocol data communication service – IP packet transfer and availability performance parameter. ITU (2007)
11. Skurowski, P., Gruca, A.: SMPDV – A New Jitter Estimator Proposal. Studia Informatica 29(4B), 37–47 (2008)
12. Anagnostakis, K.G., Greenwald, M., Ryger, R.S.: cing: Measuring Network-Internal Delays using only Existing Infrastructure. In: Proc. IEEE INFOCOM 2003 (2003)

QoS Predictability of Internet Services

Tomasz Bilski

Poznań University of Technology
tomasz.bilski@put.poznan.pl

Abstract. The paper[1] presents problems of QoS (Quality of Service) predictability of network services (mainly in WAN environment). In the first part we present general remarks on QoS predictability problem, mentioning some research projects and available resources. The main part of the paper deals with QoS predictability in long-term as well as short-term viewpoints. We will try to answer a question: is it possible to predict network QoS/performance level with a use of statistical data from the past? The term quality of service has many meanings ranging from the user's qualitative perception of the service to a set of quantitative connection parameters (RTT (Round Trip Time), throughput, loss packet rate) necessary to achieve particular service quality. In the paper we will mostly use the second meaning of the term based on RFC 2386 [1]. Analyzed, statistical data on Internet performance are taken from the IEPM (Internet End-to-end Performance Measurement) database.

Keywords: QoS, throughput, RTT, lost packet rate.

1 Introduction

The number of interactive, real time services (e.g. Voice over IP) and critical network-oriented applications (such as MRP (Material Requirement Planning), SAP (Systems Applications and Products)) is growing. The services impose strict requirements on network QoS. On the other hand QoS in a given communication channel is dependent on many diverse factors. Not all of them are manageable and predictable. The question is: is it possible to find regression trend approximating statistical data, is it possible to forecast/guess QoS and network performance level with a use of data from the past? We will try to answer the question from 2 viewpoints: long-term (years) and short-term (days). In general, IT performance parameters (processor power, data bandwidth) improve in time. From the predictability point of view we may divide the parameters into two types: with or without practical limits. For example RTT for any network has minimum value that could not be crossed due to physical limitations such as signal propagation speed limit (e.g. electrical signal travels through copper wire at 182 000 km/s). Other parameters (e.g. throughput) also have limitations but in practice they are very distant from current values of the parameters.

[1] The research project is scheduled for years 2010–2013 and supported by scientific grant from the Polish Ministry of Education and Science.

A. Kwiecień, P. Gaj, and P. Stera (Eds.): CN 2010, CCIS 79, pp. 163–172, 2010.

For example typical throughput available for common user on Internet connection is measured in Mbit/s while optical fiber bandwidth reaches Tbit/s. The term QoS is defined in the paper as a set of quantitative connection parameters (delay, jitter, throughput, loss packet rate) with throughput and RTT/delay recognized as most important. In the case of such services as VoIP it is important if values of the connection parameters are not below some thresholds (e.g. for speech coded with ITU G.711 throughput should not drop below 90 kbit/s).

2 Research Projects and Resources

Network performance and quality of Internet services is an important research topic. Many different tools and services have been designed and implemented, for example: IEPM at Stanford Linear Accelerator Center, MWING at Wroclaw University of Technology [2]. The Software Predictability Group[2] from Department of Computer Science on The University of Virginia is working on such topics as: QoS Control, QoS Portability and architectures for QoS on the Web. Internet performance measurement is also business area with commercial tools and services, for example Keynote's test and measurement products and services[3]. Furthermore, number of protocols for QoS optimization on different TCP/IP layers are introduced and tested (e.g. [3]). There are also forecasts estimating Internet growth rates, e.g. Cisco Visual Networking Index predicts that in 2008–2013 overall, IP traffic will grow at a compound annual growth rate of 40% [4]. Analysis presented here is based mainly on statistical data taken from the IEPM database operating since 1995 and collecting Internet performance measurements since 1998 [5].

3 Long Term QoS Predictability

Predicting average QoS level of Internet services in months and years perspective is a complex, multidimensional problem. One has to take into account many technological, economic, social factors, such as number of hosts, services, users, cable deployment, bandwidth, router performance. The status quo is roughly known but future is hardly predictable, e.g. Moore's law is no more satisfied.

As an example we will analyze changes in average QoS factors for Europe to North America connections. Transatlantic Internet connections between Europe and North America are based on submarine cables. With growing number of hosts and Internet services the traffic increases quickly. In general, theoretical fiber capacity growth as well as router performance growth roughly keep pace with growing user demands [6]. Nevertheless new submarine cables deployment is sluggish and irregular (e.g. last new transatlantic cable was deployed in 2003) and no new cables are planned for next two years[4]. Data on transatlantic cables are based on ICPC (International Cable Protection Committee) resources [www.iscpc.org].

[2] http://www.cs.virginia.edu/qos/
[3] http://www.keynote.com
[4] http://www.atlantic-cable.com/Cables/CableTimeLine/index2001.htm

It may be assumed that total capacity of North Atlantic cables operated in 1998 was equal roughly to 170 Gbit/s. Several new cables were deployed between 1999 and 2003. The total capacity grew to about 19 Tbit/s. So we have roughly 110-fold increase in available bandwidth[5].

Throughput. From the user point of view total bandwidth is less important than data throughput. One may assume that data throughput (available for a given user) growth would be equivalent to bandwidth growth. Statistical data show that the two rates of progress are dissimilar (Table 1) (the transatlantic communication channels are asymmetric, the parameters for two directions (USA to Europe and Europe to USA) are different. USA to Europe throughput is worse than Europe to USA, the table presents average values for both directions and between 150 and 550 different connections[6]). From 1998 to 2009 the total capacity of submarine cables increased about 110 times, while throughput increased just 6-fold. Despite fast and numerous advances in physical layer, link layer and new telecommunication cables deployment network performance measures are hardly progressing. Furthermore, the average yearly throughput is hardly predictable since it is not growing monotonously (Fig. 1). There are periods (1998–2005) of approximately linear growth (+0.32 Mbit/s/year)). On the other hand in 2005–2007 we may observe decrease from 2.5 to 1.8 Mbit/s and in 2008–2009 average throughput is constant. Year to year changes are in the range from -0.5 Mbit/s (2006/2007) to +0.9 Mbit/s (2007/2008).

Table 1. Changes in performance parameters for transatlantic connections in 1998–2009 [http://www-wanmon.slac.stanford.edu/cgi-wrap/pingtable.pl]

Year	Total capacity of cables [Tbit/s]	Average throughput [Mbit/s]	Average RTT [ms]	Median jitter [ms]	Packet loss rate [%]
1998	0.17	0.46	228	0	5.2
1999	0.21	0.48	223	0	4.1
2000	0.85	0.72	188	2.0	2.6
2001	16	1.2	202	3.4	1.4
2002	16	1.4	189	3.0	1.0
2003	19	1.8	166	1.8	1.0
2004	19	2.3	157	1.5	0.6
2005	19	2.5	156	1.0	1.6
2006	19	2.3	163	0.8	1.4
2007	19	1.8	163	0.8	3.3
2008	19	2.7	165	1.1	0.7
2009	19	2.7	162	1.1	1.7

[5] It may be observed that Ethernet bandwidth is also growing exponentially from 10 Mbit/s in 1989 to 100 Gbit/s in 2010 – this means 10 000-fold increase in 20 years.

[6] The number of tested connections changes in time.

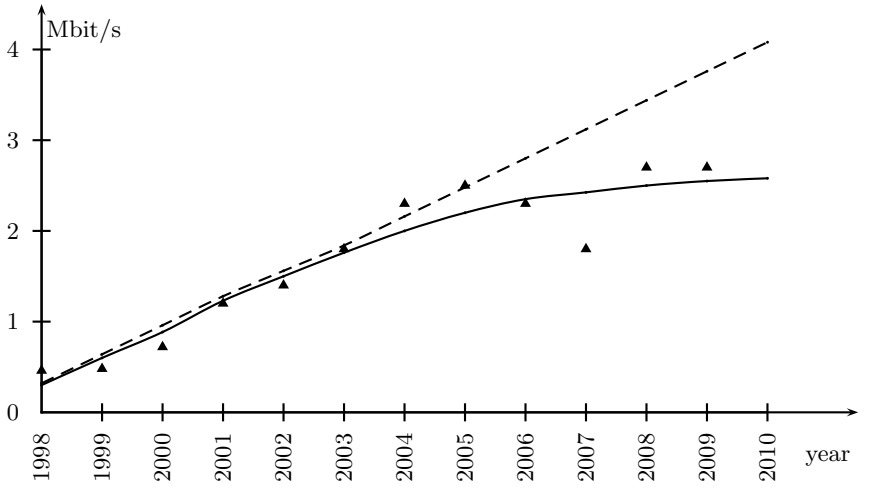


Fig. 1. Statistical data on throughput with linear regression line (based on data for 1998–2005) and polynomial regression line (based on data for 1998–2009)

Going back to 2005 and trying to predict average throughput for year 2009 we have only data from 1998–2005. Assuming best approximation with linear regression (Fig. 1, dashed line),

$$y = 0.3193x - 0.0793 \tag{1}$$

where $x = year - 1998$, in 2009 throughput should be at the level of 4.08 Mbit/s. It is obvious that value predicted this way is far from accurate. On the other hand going back to 2008, using data from 1998–2008 and assuming best approximation with polynomial regression (Fig. 1, solid line)

$$y = 0.0151x^2 + 0.4107x - 0.1573 \tag{2}$$

in 2009 throughput should be at the level of 2.58. The value is slightly below the measured one (Fig. 1), so the approximation is well fitted. However, observing statistical data from the past one is not able to guess if this polynomial trend will hold on in next years.

RTT and Jitter. Another important QoS parameter is RTT. It is rather loosely related to number of submarine cables and their capacity. For a given communication channel RTT is dependent on routers' performance and their load. The router performance, which takes advantage of the Moore's Law[7] increases at approximately the same rate as processor power. For transatlantic connections, in 1998–2004 average RTT drops from 228 ms to 157 ms (about -11 ms/year), in

[7] For years it was assumed that, according to Moore's law [7], processing power doubles every 18 months. Nowadays the assumption is negated even by the author of the law [http://www.techworld.com/opsys/news/index.cfm?NewsID=3477].

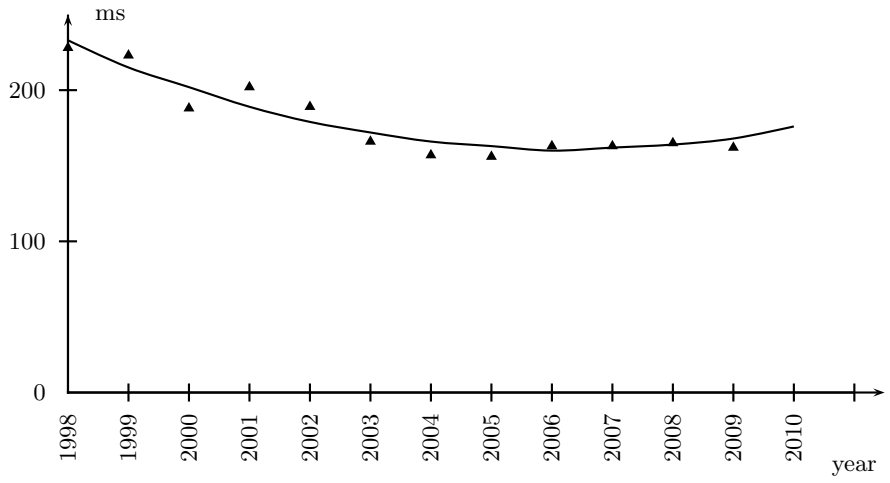


Fig. 2. Statistical data on RTT with polynomial regression line

the period 2004–2009, average yearly RTT is more or less predictable and constant. There are only minor fluctuations (±5 ms). Using data from 1998–2008 and assuming polynomial regression (Fig. 2, solid line)

$$y = 1.0932x^2 - 19.973x + 251.37 \tag{3}$$

where $x = year - 1998$, in 2009 RTT should be at the level of 167 ms (the value is near the measured one) (Fig. 2). It may be assumed that 100–110 ms is a minimum value of RTT (in the case of hosts topologically near submarine transmit and receive terminals and a minimum number of routers) and no further improvement is possible because of physical limitations. Typical length of North Atlantic cable is between 6000 and 7500 km. The speed of infrared waves in a fiber is close to 220 000 km/s, so the signal travels one way through the cable in 27–34 ms. In the worst case transmit terminal in submarine fiber system adds 13 ms delay and receive terminal 10 ms [8]. So one way signal propagation time (excluding delays in routers) is about 40–50 ms.

Average jitter in the given period fluctuates highly due to temporary problems on some tested connections (see Sect. 4). Median of jitter is much more stable, in the last 5 years it stays in the range of 0.8–1.1 ms.

In the last 10 years average lost packet rate fluctuates between 0.6 and 3.3% It must be noted that lost packet rate above 2.5% means bad quality of real time voice transmission.

4 Short Term QoS Predictability

Predicting QoS level in short term (days) perspective should be easier assuming there are no extraordinary incidents. In normal operation average values of network parameters do not change rapidly.

4.1 Throughput

Table 2 presents average monthly (December 2009) and yearly (2009) values of throughput together with daily throughput values from January 2010 for North America to Europe Internet connections (average values for 184–202 connections). In the case of N. America to Europe connections minimum value of daily throughput in January 2010 drops significantly below average minimum values for 2009 and December 2009. Average and maximum values in January 2010 are relatively constant (average – 1.88–2.06 Mbit/s, maximum – 3.40–3.60) but much worse than average and maximum for December 2009. So predicting January data with a use of December statistics give us overestimated numbers.

Table 2. Exemplary fluctuations of throughput in January 2010 for N. America to Europe connections [http://www-wanmon.slac.stanford.edu/cgi-wrap/pingtable.pl]

Date	Minimum [Mbit/s]	Average [Mbit/s]	Maximum [Mbit/s]	Std deviation [Mbit/s]
2009	1.12	2.61	8.70	1.74
Dec 2009	0.06	3.89	18.0	3.25
Jan 1, 2010	0.21	2.06	3.52	0.64
Jan 2, 2010	0.32	2.05	3.41	0.55
Jan 3, 2010	0.30	2.06	3.40	0.57
Jan 4, 2010	0.18	2.05	3.48	0.68
Jan 5, 2010	0.08	2.01	3.41	0.68
Jan 6, 2010	0.06	2.01	3.52	0.67
Jan 7, 2010	0.06	1.94	3.40	0.63
Jan 8, 2010	0.07	1.88	3.48	0.63
Jan 9, 2010	0.03	1.94	3.60	0.67
Jan 10, 2010	0.05	2.00	3.60	0.68

Table 3. Exemplary fluctuations of throughput in January 2010 for Europe to N. America connections [http://www-wanmon.slac.stanford.edu/cgi-wrap/pingtable.pl]

Date	Minimum [Mbit/s]	Average [Mbit/s]	Maximum [Mbit/s]	Std deviation [Mbit/s]
2009	0.96	2.84	6.80	1.64
Dec 2009	1.09	4.11	11.5	2.34
Jan 1, 2010	1.17	2.45	3.66	0.64
Jan 2, 2010	0.58	2.44	3.68	0.71
Jan 3, 2010	0.55	2.44	3.68	0.70
Jan 4, 2010	0.55	2.30	3.58	0.76
Jan 5, 2010	0.32	2.18	3.68	0.84
Jan 6, 2010	0.28	2.28	3.68	0.80
Jan 7, 2010	0.73	2.40	3.65	0.74
Jan 8, 2010	0.98	2.43	3.67	0.68
Jan 9, 2010	0.87	2.43	3.69	0.67
Jan 10, 2010	0.69	2.40	3.64	0.80

Table 3 presents average monthly (December 2009) and yearly (2009) values of throughput together with daily throughput from January 2010 for N. America to Europe Internet connections (average values for 47–50 connections). In the case of Europe to N. America connections daily minimum throughput in January 2010 drops to roughly 1/4 of average minimum values for 2009 and December 2009. Average and maximum values in January 2010 are relatively constant (average in the range of 2.18–2.45 Mbit/s, maximum – 3.58–3.69) but worse than average and maximum for December 2009 and year 2009. So predicting January data with a use of December statistics would gives us overestimated numbers.

4.2 RTT

Table 4 presents average monthly (December 2009) and yearly (2009) values of RTT together with daily RTT from January 2010 for North America to Europe Internet connections (average values for 184–202 connections). Table 5 presents average monthly (December 2009) and yearly (2009) values of RTT together with daily RTT from January 2010 for N. America to Europe Internet connections (average values for 47–50 connections). In both cases minimum as well as average values of RTT in the first days of January 2010 are next to average values for December 2009 and for year 2009 – the differences are not more than 10% with standard deviation not exceeding 40 ms. The same is true also for maximum RTT in Europe to N. America connections. Nevertheless maximum RTT for reverse direction changes highly up to 170% of December 2009 value and 340% of 2009 value with standard deviation exceeding 110 ms (January 9, 2010).

4.3 Weekdays Fluctuations of Network Performance Level

Some minor fluctuations are visible in the average QoS values of particular days of week. Throughout a given period of time (e.g. year) Saturdays and Sundays

Table 4. Exemplary fluctuations of RTT in January 2010 for N. America to Europe connections [http://www-wanmon.slac.stanford.edu/cgi-wrap/pingtable.pl]

Date	Minimum [ms]	Average [ms]	Maximum [ms]	Std deviation [ms]
Year 2009	105	171	249	29
Dec 2009	104	179	504	59
Jan 1, 2010	103	177	411	44
Jan 2, 2010	103	172	224	27
Jan 3, 2010	103	171	229	27
Jan 4, 2010	104	172	338	37
Jan 5, 2010	106	180	527	61
Jan 6, 2010	105	184	660	80
Jan 7, 2010	106	182	673	85
Jan 8, 2010	103	179	541	66
Jan 9, 2010	102	188	840	115
Jan 10, 2010	103	183	782	99

Table 5. Exemplary fluctuations of RTT in January 2010 for Europe to N. America connections [http://www-wanmon.slac.stanford.edu/cgi-wrap/pingtable.pl]

Date	Minimum [ms]	Average [ms]	Maximum [ms]	Std deviation [ms]
Year 2009	102	154	203	30
Dec 2009	101	150	201	30
Jan 1, 2010	101	147	198	30
Jan 2, 2010	100	147	198	30
Jan 3, 2010	100	148	197	30
Jan 4, 2010	102	148	203	30
Jan 5, 2010	100	156	222	37
Jan 6, 2010	100	155	218	37
Jan 7, 2010	101	148	198	29
Jan 8, 2010	101	149	198	30
Jan 9, 2010	100	148	199	30
Jan 10, 2010	102	148	198	30

have the best average QoS (the highest average throughput and the lowest average RTT) Mondays have the worst. Nevertheless in a given week of the year it is not possible to predict a day with the highest QoS level [9].

4.4 Unpredictability of QoS in the Case of Disasters

QoS is disrupted by many, intentional and unintentional factors: malicious software, spam, hackers and disasters (e.g. earthquakes). First 3 factors are widespread and hardly predictable. Damages imposed are usually short-lived and restricted to single service. On the other hand disasters are uncommon, almost unpredictable, their impact is long-lived (up to several months of degraded QoS). They disrupt Internet services, telephone calls and ATM transactions. Intercontinental cable faults are relatively infrequent (in 2003 annual fault rate was at the level of 1 fault per 10 000 km of cable [10]). In the Atlantic, cable breaks happen repeatedly (above 50 cable repairs are yearly). Submarine cables are prone to being affected by earthquakes, storms, fishing equipment and anchors (Table 6).

The disaster's impact on QoS is: significant, long-lived and widespread (due to rerouting of damaged connections). The performance parameters during the accidents decrease in some cases to unacceptable (for interactive applications) levels. Internet performance is changing in unpredictable way. The effects of submarine cable cut accident are notably different from the effects of hacker attack on Internet server. Similar accidents in the future should be expected. Generally we are not able to predict time and place of the accidents (e.g. earthquakes are hardly predictable). Case study of January 2008 accident is presented in [11].

Table 6. Recent disasters with submarine cable faults

Date	Damaged cables	Cause of cable fault	Time of QoS degradation
May 2003	5 cables near Algiers	earthquake	6 weeks
Nov 2003	TAT-14 (USA-Europe)	technical fault	few days
Jun 2005	SMW 3 (Pakistan)	unknown	12 days
Dec 2006	CANTAT-3 (Iceland-Europe)	unknown	7 months
Dec 2006	7 cables near Taiwan	earthquake	7 weeks
Jan 2008	SMW 4, FLAG, FALCON	anchors	2 weeks
Dec 2008	SMW 3, SMW 4, FLAG	earthquake	unknown

5 Conclusions

It is easy to conclude that QoS prediction is a hard problem especially in long-term perspective. How to determine future if such basic and general principles as Moore's Law are no more fulfilled. Intercontinental connections performance is upgrading very slowly and irregularly. QoS is related to a few network performance parameters. Some of them are easier to predict than others. Such parameters as average RTT are relatively stable, on the other hand throughput fluctuations are large.

Compound growth of throughput for exemplary transatlantic connections in 10 years is just 6-fold on average. The increase rate is lagging behind other computer and network performance indicators. At the same time total North Atlantic bandwidth increased approximately 110 times. This is related to fiber transmission capacity and DWDM link speed, which grow by a factor of about 200 in the decade. Similarly, computer power increase in the same period is 100-fold. The router capacity increases at approximately the same rate. Predicting average parameters (in particular throughput) in long-term perspective is hardly possible, even with a use of great number of statistical data. Many factors should be taken into account. The values of the parameters (e.g. throughput) slowly improve in time but the improvement is not monotonous and very loosely related to more or less predictable factors such as communication channel bandwidth or processor (router) performance. Approximating statistical data with regression trends is hardly possible. We have shown that the best approximation for some sets of older data is linear regression. Unfortunately this regression type is useless if we take into account more recent data.

In short-term perspective it is much easier to predict QoS level, assuming there are no extraordinary incidents. In normal operation average values of network parameters do not change rapidly. Nevertheless irregular disasters with submarine cable faults make this predictability more complicated. Significant QoS deterioration together with high fluctuations of performance parameters may last weeks and even months.

References

1. Crawley, E., Nair, R., Rajagopalan, B., Sandick, H.: A Framework for QoS-based Routing in the Internet RFC 2386, IETF (1998)
2. Borzemski, L., Cichocki, Ł., Fraś, M., Kliber, M., Nowak, Z.: MWING: A Multiagent System for Web Site Measurements. In: Nguyen, N.T., Grzech, A., Howlett, R.J., Jain, L.C. (eds.) KES-AMSTA 2007. LNCS (LNAI), vol. 4496, pp. 278–287. Springer, Heidelberg (2007)
3. Huston, G.: Next Steps for the IP QoS Architecture RFC 2990, IETF (2000)
4. Cisco Visual Networking Index: Forecast and Methodology, 2008–2013, Cisco, San Jose (2009)
5. Cottrell, L., Matthews, W., Logg, C.: Tutorial on Internet Monitoring and PingER at SLAC, SLAC (2007), http://www.slac.stanford.edu/comp/net/wanmon/tutorial.html
6. Bilski, T.: Inconsistency in QoS factors improvement rates, Information Systems Architecture and Technology. In: Grzech, A., Borzemski, L., Świątek, J., Wilimowska, Z. (eds.) Information Systems and Computer Communication Networks. Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław (2009)
7. Moore, G.E.: Cramming more components onto integrated circuits. In: Electronics, April 19, vol. 38(8) (1965)
8. ITU-T Recommendation G.114. One-way transmission time, ITU-T 05/2003 (2003)
9. Bilski, T.: Fluctuations and Lasting Trends of QoS on Intercontinental Links. In: QSHINE 2009, The 6th Int. ICST Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness. ICST, Las Palmas (2009)
10. Kordahi, M.E., Shapiro, S.: Worldwide Trends in Submarine Cable System Faults, SubOptic (2004), http://www.suboptic.org/Viewdocument.aspx?id=381
11. Bilski, T.: Disaster's Impact on Internet Performance – Case Study. In: Kwiecień, A., Gaj, P., Stera, P. (eds.) Proceedings of 16th conference on Computer Networks, CN 2009, Wisła, Poland, June 16-20, pp. 210–217. Springer, Heidelberg (2009)

Instruction Prediction in Microprocessor Unit Based on Power Supply Line

Michał Maćkowski and Krzysztof Skoroniak

Silesian University of Technology, Institute of Computer Science,
Akademicka 16, 44-100 Gliwice, Poland
{michal.mackowski,krzysztof.skoroniak}@polsl.pl
http://www.polsl.pl/

Abstract. This paper illustrates the research results concerning the character of signals emitted by the selected microcontroller via the power supply lines. The main purpose of the study is to determinate the instruction that is currently realized by the microprocessor, based on the conducted emission in the power supply lines.

The research results presented in the study indicate that there are the differences in the spectrum of the signal emitted by the processor, depending on the program being executed. Thus, it can be presumed that there is a partly possibility to determinate the instruction currently realized by a microprocessor based on the conducted disturbances emitted by the power supply lines. It can be suggested that there is a risk for programs written in microprocessor memory, which should be protected from copying.

Keywords: conducted emission, electromagnetic compatibility (EMC), electromagnetic disturbances, microcontroller, program code.

1 Introduction

The source of electromagnetic field diffusing in the space is each device supplied with electric energy. This kind of field can disturbs the work of various electric and electronic devices. However, the disturbances are not the only problem that accompanies electromagnetic emission. It is possible, in some cases, to get the information about signals appearing inside the device if the emission from the device is recorded, and the received signals are decoded and interpreted by the appropriate methods. Such situation is especially value in the case of digital devices because the remote reconstruction of signals inside the device may enable to regenerate information processing by this device.

The high integration scale and permanent increase of the frequency of microprocessor circuit, entail that current peaks are generated with higher amplitudes and shorter rise times on the power supply lines of the electronic circuits. These impulses are generated by simultaneous switching of millions of transistors inside of the integrated unit. The total power drawing by all gates that execute a single instruction may indicate the kind of instruction which is already executed. Moreover, the propagation of these currents through lines and paths on

A. Kwiecień, P. Gaj, and P. Stera (Eds.): CN 2010, CCIS 79, pp. 173–182, 2010.

PCB board into other electronic devices may cause problems with their proper functioning.

The network controller placed in the network card can be also considered as a microprocessor unit, which is responsible for data processing of transmitted frames. Such unit can be also the source of electromagnetic disturbances that can interfere the work of other electronic devices.

The previous article [1] aimed mainly at determining the spectrum of signal emitted by the processor during realization of a particular single instruction. Nevertheless, this instruction was realized in the loop. This paper presents the methodology and the research, that allow determining and comparing the spectrum of a specific single instruction. This issue is opposite to the one discussed in the publications [2] and [3], where authors intended to predict the emission of electromagnetic disturbances emitted by the microprocessor under different program behavior.

The main purpose of the research is to determine the instruction currently realized by the microprocessor based on the emission of conducted disturbances on the power supply lines. The test bench constructed for the research is discussed in the Sect. 2. Determining the instruction currently executed, which is based on the disturbances on the power supply lines may lead to unauthorized reconstruction of a program without any directly interference into the microprocessor memory. Protection of information is a very essential issue; otherwise the cost of so far, very expensive systems of confidential information increases. Moreover, an inappropriate attitude to this issue may expose the protected information to be used in an unauthorized way.

2 Test Bench

The research used a microprocessor PIC16F84A produced by Microchip. The test bench illustrated in the Fig. 1 consists of microprocessor to which the power supply and 2 MHz external square waveform generator were connected. In order to exclude conducted disturbances that can get into microprocessor when using an external AC/DC power supply, a complete of batteries was used instead.

The oscilloscope was connected to the power supply lines to monitor the decreases of voltage. With the using batteries as an unstable source of power supply, the voltage decreases were equivalent to the current in the power supply lines. Such solution was dictated by the difficulties in finding a proper current probe that could have measured 1 milli- or even 1 microampere current with the approximate 1 GHz sampling frequency. The microchip programmer used in the research, was connected to the microprocessor only for the period of programming, in order to avoid conducted disturbances that can get from the programmer to the supplying lines.

The microprocessor PIC16F84A used in the research has one executive pipeline, which means that during one instruction cycle only one command is realized. One instruction cycle (2 μs) consists of four machine cycles (500 ns):

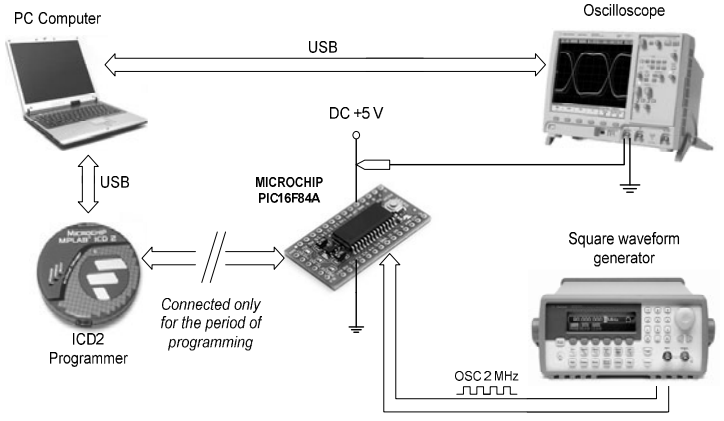


Fig. 1. The schema of research position

- Q1 – instruction decode cycle,
- Q2 – instruction read data cycle,
- Q3 – process the data,
- Q4 – instruction write data cycle and fetching the next instruction from the program memory.

Due to the fact that in the last cycle (Q4) another instruction is fetched from the program memory, thus not only currently realized instruction but also next instruction has the influence on the current flow (shape). Microcontroller PIC16F84A contains totally 35 instructions, among others:

- 1-cycle instructions ($2\,\mu$s): ADDWF, ANDWF, CLRF, CLRW, COMF, DECF, INCF, IORWF, MOVF, MOVWF, NOP, RLF, RRF, SUBWF, SWAPF, XORWF, BCF, BSF, ADDLW, ANDLW, CLRWDT, IORLW, MOVLW, SLEEP, SUBLW, XORLW,
- 2-cycle instructions ($4\,\mu$s): DECFSZ, INCFSZ, BTFSC, BTFSS, CALL, GOTO, RETFIE, RETLW, RETURN.

In the research presented in this work only 1-cycle instructions (24 instructions) were taking into account in order to simplify the analysis. However, the authors intend to test also 2-cycle instruction in the further research.

3 The Research Procedure

First, the two researches were conducted for all 24 one-cycle instructions, and next similarities among particular instructions were determined. The research procedure is as follows:

1. Determination of the time flow of the microprocessor power supply voltage during whole program (Fig. 2 and 3a).

2. Excision of part of the time flow which refers to the currently tested instruction (Fig. 3b).
3. Passing from time domain to frequency domain for the removed part of time flow using FFT calculation algorithm of Fourier Transform (Fig. 3c).
4. Comparison of amplitude spectra determined for particular instructions based on the least squares method.

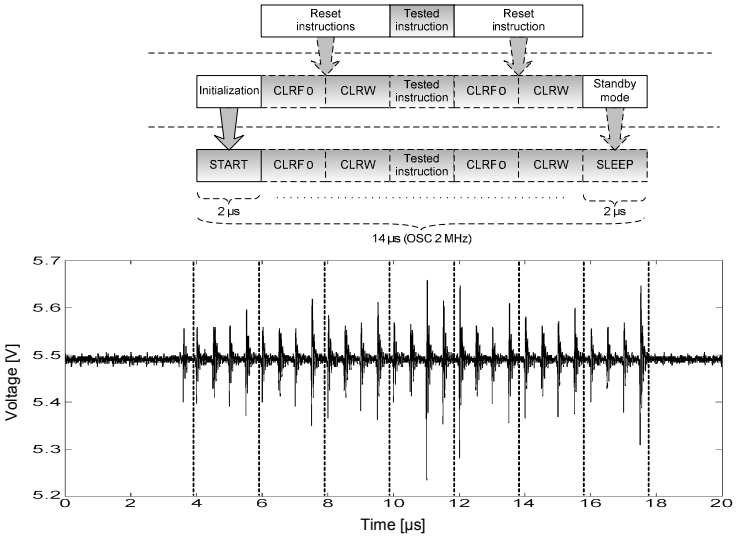


Fig. 2. The research procedure – microcontroller program construction

It was necessary, during the test of each instruction to record a proper program for the microprocessor (Fig. 2). Thus, each program consists totally of 7 instructions:

- the instruction that initiates the microprocessor, realized implicitly,
- the instructions that reset the value of accumulator and the first working register,
- the tested instruction,
- the instructions that reset the value of accumulator and the first working register,
- the SLEEP instruction that aim was to switch the microprocessor into standby mode.

In order to ensure the same conditions for realization of each instruction, the working registers that were used by a particular instruction, were reset each time. During the test the parameters of a particular instruction were set to 0.

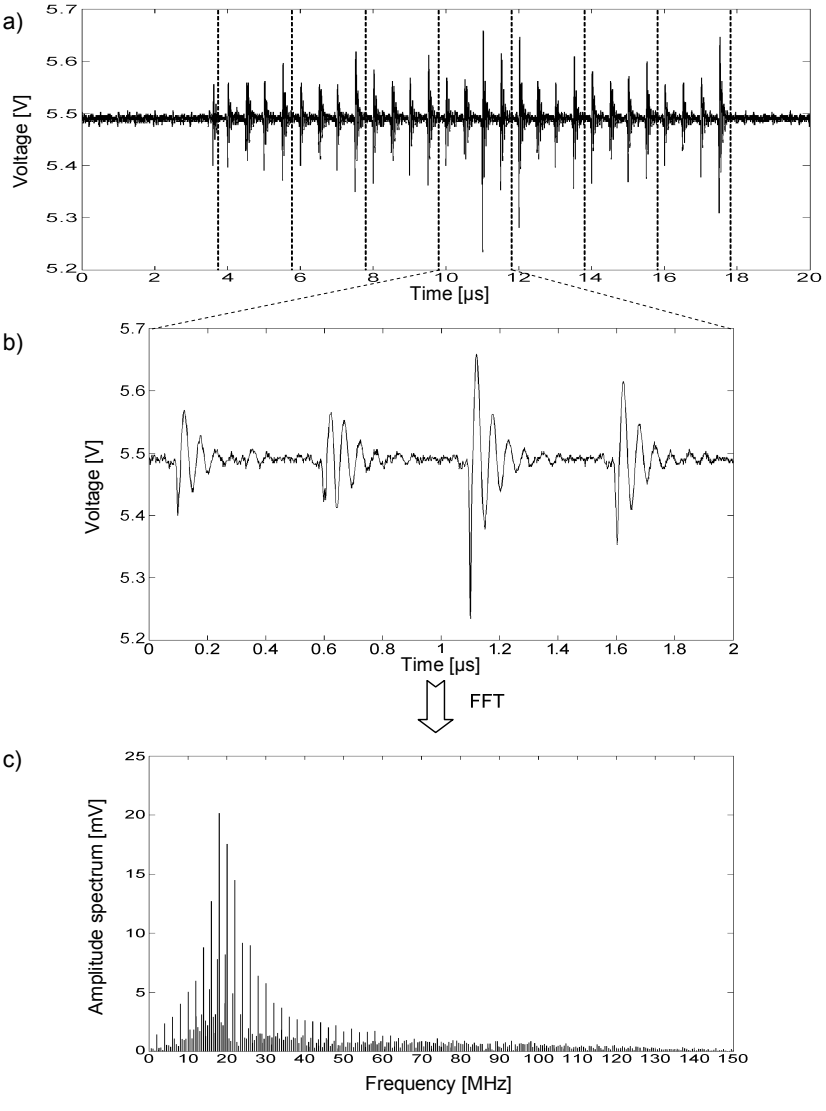


Fig. 3. The research procedure: (a) Voltage on power supply line – test 1, program 6, (b) Voltage on power supply line – test 1, program 6, DECF instruction (c) Amplitude spectrum – test 1, program 6, DECF instruction

4 The Research Results

As the test results illustrate 19 of 24 instructions were correctly recognized thanks to analysis of microprocessor supply voltage flow (Table 1).

Table 1. The results of amplitude spectra comparison determined for particular instructions using the least squares method

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
ADDWF	1	0.858	2.043	0.726	7.531	27.253	29.260	2.102	0.885	0.827	0.749	1.516	0.740	0.840	5.536	0.744	0.868	3.346	1.746	3.059	4.627	1.829	1.517	10.428	2.004
ANDWF	2	1.257	0.784	2.049	3.138	20.196	21.375	1.811	1.526	2.030	1.834	0.832	2.199	2.106	1.797	1.627	1.737	0.930	1.523	1.545	1.719	1.252	1.308	4.843	1.416
CLRF	3	1.230	2.639	0.556	7.982	29.847	31.935	2.716	1.145	0.843	0.681	2.170	0.950	1.028	6.513	0.966	1.048	3.756	1.818	3.988	5.802	2.770	2.013	12.204	2.705
CLRW	4	4.813	3.040	6.598	0.685	12.870	13.666	3.393	5.324	6.439	6.073	3.737	6.838	6.541	1.931	5.632	5.856	2.191	3.566	4.160	3.373	4.673	5.204	3.952	4.733
COMF	5	23.864	21.301	28.627	13.043	0.759	0.852	18.961	25.624	27.151	28.003	21.329	27.706	27.376	15.675	25.771	25.683	18.385	20.272	19.486	17.791	21.236	22.890	13.174	21.961
DECF	6	25.298	22.470	30.418	13.238	0.810	0.743	20.022	26.890	28.725	29.441	22.410	29.406	29.074	16.266	27.230	27.208	19.392	21.452	20.518	18.848	22.880	24.458	13.682	23.198
INCF	7	1.254	1.983	1.911	4.231	18.432	20.101	0.736	1.611	1.666	1.914	1.488	1.742	1.669	4.086	1.398	1.485	2.704	0.814	3.081	4.142	2.245	2.115	8.409	2.456
IORWF	8	0.632	1.631	0.919	5.900	24.516	26.426	1.649	0.577	0.680	0.797	0.985	0.922	0.895	4.445	0.615	0.592	2.621	1.334	2.543	3.852	1.536	0.946	9.194	1.661
MOVF	9	0.754	2.188	0.707	7.411	27.156	29.204	2.008	0.821	0.554	0.727	1.583	0.756	0.684	5.865	0.647	0.719	3.641	1.689	3.439	5.104	2.071	1.506	11.142	2.186
MOVWF	10	0.952	2.036	0.671	6.950	28.031	30.001	2.301	0.772	0.746	0.537	1.603	0.923	0.934	5.437	0.868	0.806	3.051	1.666	3.312	4.821	2.215	1.570	10.618	2.211
NOP	11	1.223	1.163	2.381	4.174	19.882	21.323	1.832	1.625	2.019	2.040	0.596	2.210	2.009	2.487	1.500	1.686	1.561	1.785	1.312	1.843	0.891	0.801	5.488	0.982
RLF	12	0.796	2.053	0.601	7.173	26.624	28.603	1.848	0.741	0.609	0.744	1.448	0.575	0.607	5.668	0.605	0.673	3.350	1.533	3.273	4.789	1.965	1.375	10.691	2.083
RRF	13	0.773	1.714	0.813	6.265	25.400	27.327	1.635	0.719	0.710	0.812	1.174	0.836	0.836	4.884	0.686	0.691	2.849	1.343	2.804	4.176	1.742	1.223	9.658	1.746
SUBWF	14	3.662	1.886	5.407	2.139	14.981	15.709	3.690	4.244	5.290	4.848	2.369	5.591	5.313	0.839	4.550	4.583	1.093	3.354	2.057	1.033	2.553	3.067	1.949	2.701
SWAPF	15	0.658	1.619	0.804	6.294	25.244	27.187	1.592	0.708	0.617	0.760	1.099	0.673	0.641	4.711	0.571	0.693	2.776	1.313	2.679	4.081	1.596	1.090	9.554	1.629
XORWF	16	0.673	1.571	1.031	5.753	24.260	26.221	1.511	0.983	0.757	0.869	0.973	0.854	0.857	4.303	0.647	0.690	2.497	1.226	2.369	3.783	1.650	1.050	9.003	1.413
BCF	17	2.867	1.395	4.092	2.200	18.008	18.837	3.191	3.188	4.160	3.534	1.830	4.549	4.216	1.022	3.516	3.634	0.624	2.660	1.796	1.189	2.043	2.447	3.016	2.151
BSF	18	1.011	1.662	1.235	4.701	20.552	22.179	1.139	1.279	1.384	1.264	1.214	1.355	1.405	3.907	1.124	1.230	2.169	0.686	2.664	3.646	1.712	1.599	8.029	1.912
ADDLW	19	2.020	1.616	3.452	4.399	18.701	19.946	2.876	2.696	3.177	3.184	1.337	3.523	3.088	2.156	2.569	2.658	1.530	2.629	0.680	1.029	0.841	1.166	3.722	0.868
ANDLW	20	3.446	1.985	5.287	3.187	17.121	17.956	3.952	4.113	5.056	4.701	2.150	5.561	5.137	1.080	4.301	4.338	1.128	3.722	1.123	0.532	1.766	2.334	1.824	1.812
IORLW	21	1.555	1.674	2.996	4.857	20.527	22.027	2.575	1.976	2.425	2.492	0.877	2.843	2.598	2.521	1.877	2.051	1.757	2.295	0.841	1.494	0.852	0.713	5.029	0.696
MOVLW	22	1.029	1.563	1.918	5.903	23.214	24.861	2.301	1.297	1.515	1.677	0.772	1.776	1.573	3.550	1.120	1.256	2.041	1.919	1.179	2.236	0.691	0.470	6.764	0.695
SUBLW	23	7.847	5.181	10.709	3.781	13.244	13.559	7.753	8.695	10.291	9.845	5.711	10.961	10.399	2.018	9.113	9.231	3.295	7.481	3.497	1.865	5.107	6.096	0.704	5.123
XORLW	24	1.524	1.357	2.730	4.749	20.049	21.483	2.571	1.942	2.454	2.432	0.917	2.820	2.418	2.369	1.926	1.991	1.548	2.158	0.790	1.317	0.668	0.835	4.550	0.727

The Figures 4, 5 and 6 present an exemplary microprocessor supply voltage flow for entire program. Figure 4 comes from program 6 executed during the first test, whereas Fig. 5 comes from program 6 which was executed in the test 2. As it can be seen there is a similarity between the two charts. However, Fig. 6 shows that the flow of voltage for program 7 differs a lot from the two others. That can indicate the differences in the microprocessor supply voltage which depend on a particular executed program or a single instruction. The Figures 7, 8 and 9 present the same programs, but only the parts of voltage that refer to the tested instruction were specified. Whereas Figs. 10, 11 and 12 present amplitude spectra for Figs. 7, 8 and 9 that were determined by the FFT algorithm. The spectra determined for the whole instructions were afterwards compared using the method of the least squares. The results are presented in Table 1.

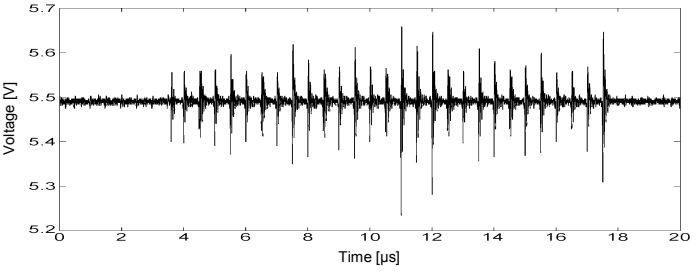


Fig. 4. Voltage on power supply lines – test 1, program 6

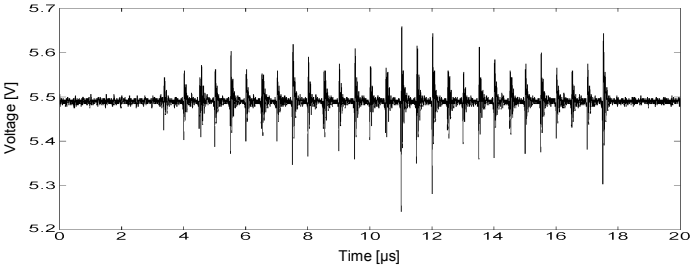


Fig. 5. Voltage on power supply lines – test 2, program 6

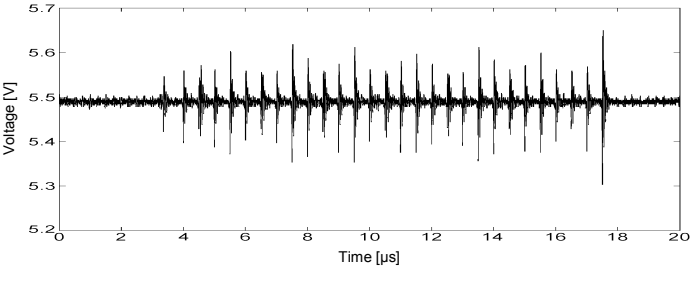


Fig. 6. Voltage on power supply lines – test 2, program 7

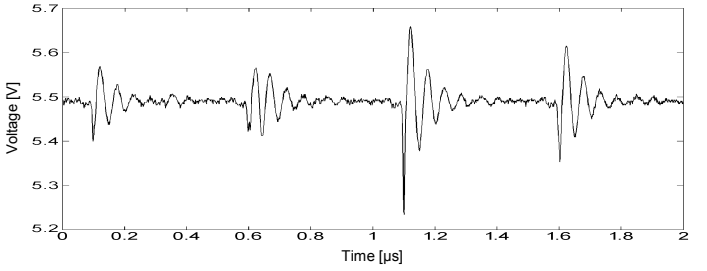


Fig. 7. Voltage on power supply lines – test 1, program 6, DECF instruction

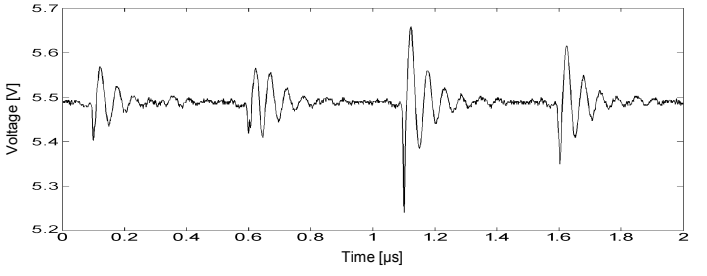


Fig. 8. Voltage on power supply lines – test 2, program 6, DECF instruction

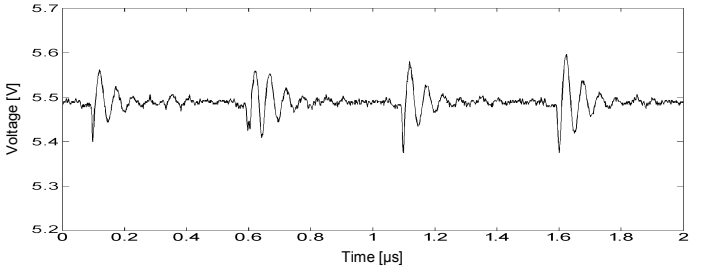


Fig. 9. Voltage on power supply lines – test 2, program 7, INCF instruction

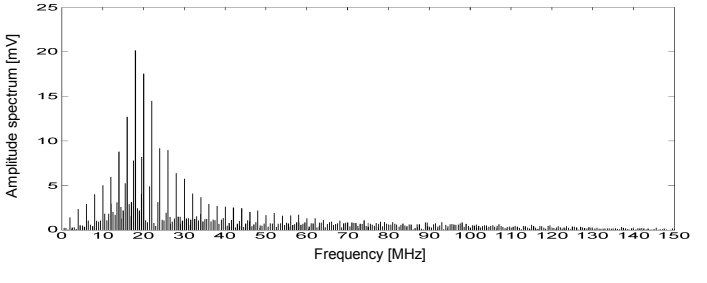


Fig. 10. Amplitude spectrum – test 1, program 6, DECF instruction

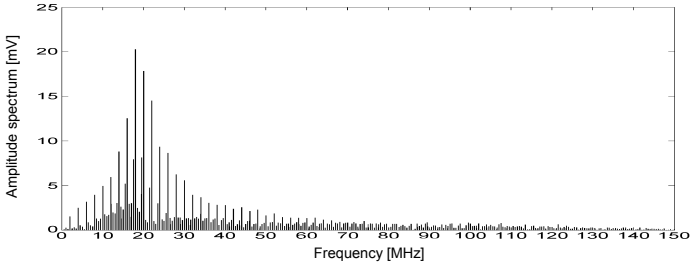


Fig. 11. Amplitude spectrum – test 2, program 6, DECF instruction

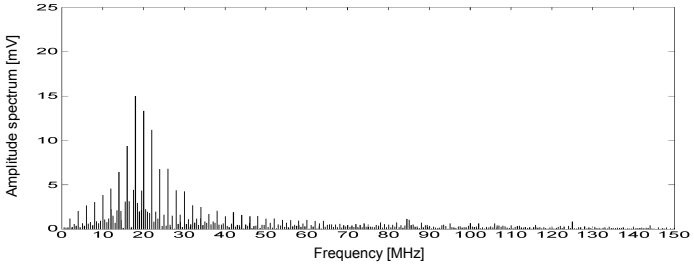


Fig. 12. Amplitude spectrum – test 2, program 7, INCF instruction

The fields in the Table 1 present the result of the least squares method between two particular instructions. Because of the fact that the method of the least squares was used, thus the smallest value is in a particular field the more similarity is between two instructions. Instructions mentioned in the following verses represent the first research, whereas instructions in the columns refer to the research 2. The expected minimal values should appear along a diagonal line thanks to the special construction of the chart. The gray fields stand for the smallest values where the analysis was correct. Meanwhile the incorrect results of analysis were marked with black background and white text. Fields marked with black border in particular columns refer to instructions, which were recognized as those of more possibility than tested instructions.

5 Conclusion

The research results presented in the paper indicate that there are unambiguous differences in the spectrum of conducted signal emitted by the microprocessor, depending on an executing instruction. Therefore, it can be concluded that there is a partly possibility to determine the currently executing instruction by the microprocessor based on the conducted disturbances emitted through the power supply lines. However, that situation may be dangerous for programs saved in memory of microprocessors, which should be protected from being copied.

In the further research the authors intend firstly to take into account 2-cycle instructions. Next, to increase the precise of receiving results, create a possibility

to determine instructions executing in the sequence, and yet consider the instructions parameters in tests.

If the further research allows increasing the precise of foreseeing currently executing instruction, then there is a serious risk for programs saved in the memory of microprocessors. In the conducted research it was possible to foreseen the currently executing instruction with the 79% of probability (19 of 24 instructions recognized correctly). Determination of the currently executing instruction based on the disturbances in the power supply lines, may allow for unauthorized reconstruction of program without directly intervention in the microprocessor memory.

References

1. Maćkowski, M., Skoroniak, K.: Electromagnetic emission measurement of microprocessor units. In: Kwiecień, A., Gaj, P., Stera, P. (eds.) 16th Conference on Computer Networks, CN 2009, Wisła, Poland. CCIS, vol. 39. Springer, Heidelberg (2009)
2. Bendhia, S., Labussiere-Dorgan, C., Sicard, E., Tao, J.: Modeling the electromagnetic emission of a microcontroller using a single model. IEEE transactions on Electromagnetic compatibility (2008)
3. Chen, C.K., Liao, S.S., Sicard, E., Yang, C.F., Yuan, S.Y.: EMI prediction under different program behavior. In: IEEE EMC Symposium, Honolulu, USA (2007)

Data Integration in Computer Distributed Systems

Błażej Kwiecień

Silesian University of Technology, Institute of Informatics,
ul. Akademicka 2A, 44-100 Gliwice, Poland
blazej.kwiecien@polsl.pl
http://www.polsl.pl

Abstract. In this article the author analyze a problem of data integration in a computer distributed systems. Exchange of information between different levels in integrated pyramid of enterprise process is fundamental with regard to efficient enterprise work. Communication and data exchange between levels are not always the same cause of necessity of different network protocols usage, communication medium, system response time, etc.

Keywords: PLC, ERP, MES, SCADA, data integration.

1 Introduction

The development of computer technologies is supporting creation of new solutions or optimizations already existing. In contemporary systems of the computer science, of which architecture is often created on the basis of distributed systems, applying computer networks is essential – Fig. 1. On account of the specificity of the work of these systems, the exchange of data between its nodes many times must be determined in time. Character of the network work and the communications bandwidth medium are forcing into applying appropriate communications protocols.

The desire of increasing data transfers and reliability became stimulus of going more and more popular standard of the network – Ethernet, which in spite of is not deterministic in time, more often find it applying in computer industrial installations [1]. It is possible to justify it by fact that both the software and the hardware connected with using the Ethernet network widely are spread and applying industrial technologies in many branches [1,2]. Using Ethernet is not limited only to steering, regulation and monitoring the industry of computer systems. It concerns also systems of managing the production – MES [3] (Manufacturing Execution System) and managing whole production plant – ERP [4] (Enterprise Resources Planning). For MES systems the information exchange in the real time is essential cause of the supervision in progress of a production process that thanks it is possible quickly to react to the state and parameters of the industrial computer object [3]. Next the ERP system supported by MES is integrating the information about events coming from the industrial computer

A. Kwiecień, P. Gaj, and P. Stera (Eds.): CN 2010, CCIS 79, pp. 183–188, 2010.

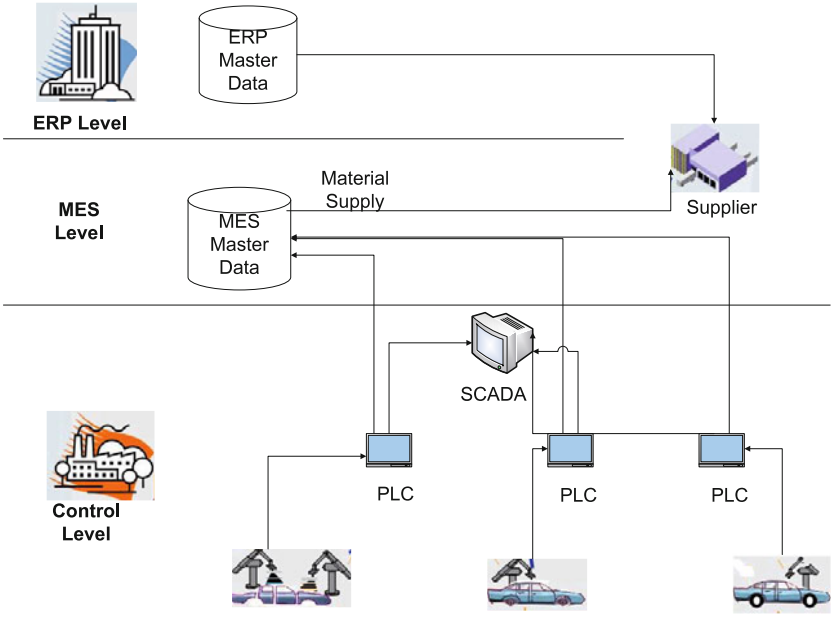


Fig. 1. Model of computer distributed systems

object with the information about business events, it delivers in the real time information about the current state and parameters of production and enables accurate analysis of the profitability and accurate costs of the production [5]. The typical, hierarchical model of management and controlling an enterprise is shown below – Fig. 2.

Level 1 takes systems of steering and the control represented by local like of also steering by distributed territorially real-time systems, carrying the service of algorithms out on the level of individual technologies resulting in a production process of the enterprise.

Level 2 takes systems of the HMI (Human Machine Interface) class, which is a platform between systems of steering and the maintenance crew staff serving them. Applied solutions on the level 2 enable an access to the information of devices describing the work put on the level 1. Additionally they allow making the parameterization of these devices and manual steering.

The level 3 assures information exchange between systems of steering (level 1) systems of the visualization (level 2), and ERP (Enterprise Resource Planning) class systems (level 4). ERP systems support production processes. Because of this systems organization of the production became easier as well as supporting the action starting from the stage of accepting the order, by the preparation, protecting and keeping up with the production, until the stage of the delivery of the final product to the recipient [6].

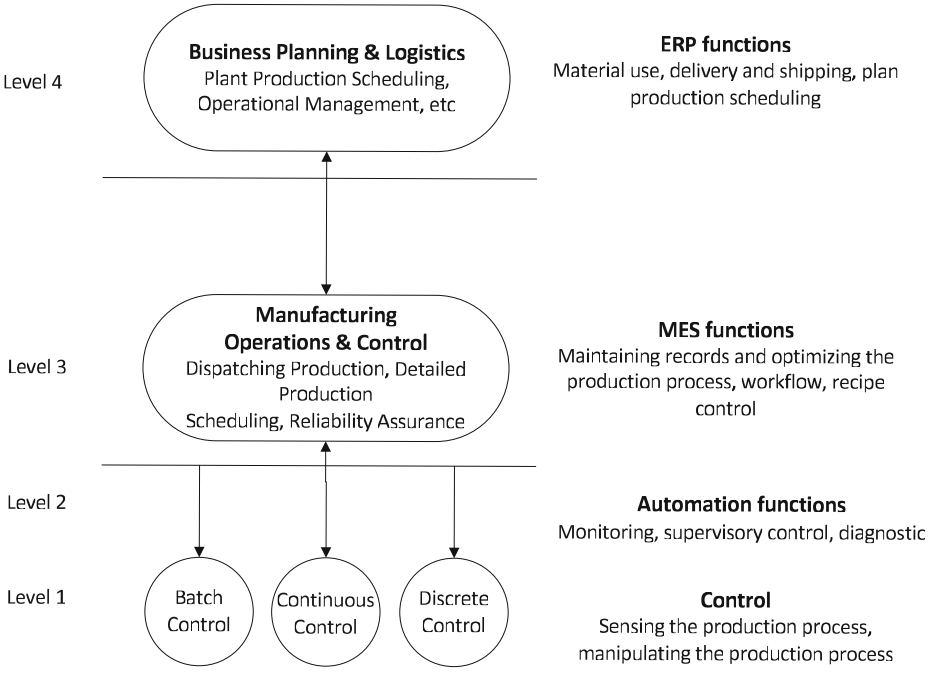


Fig. 2. Hierarchical model of industrial computer object [5]

All four levels of the presented model can harmonize with themselves only by dint of to the mutual information exchange. Integration of the different levels of the industrial computer object on the level of the computer network is crucial and essential for its correct functioning.

2 Systems Integration

In accordance to the Sect. 1, effective integration of production systems with business is possible, management layer must at first appear in the production layer. In other words, apart from devices and the SCADA application one should create the right infrastructure and the computer system, responsible for the realization and the recording of the production. MES systems are accomplishing the double role. Firstly, they provide indications essential for the staff. Those indications are current dates concerning the production – production orders, warehouse states, available machines and their productivity – those dates enable an efficient planning and what is more realization and the optimization of the production. Secondly, they enable a full integration of company management systems. This integration is most often two-way. The production systems systematically deliver and convert information about goods orders made by customers, specifications and required parameters of this orders. The ERP systems is being sent an information

about production in progress, productivity of this production, real costs, quality and all deviations from the plan which can influence on realization of the dates – of the ones and of different orders. All information is collected in the relational database which enables to prepare analyses of product, production orders, production part or to reconstruct of genealogy of products.

Integration of systems will be effective, elastic and it had the low cost of the exploitation only when it will be marked with certain special properties. Above all it should enable the automated, electronic exchange of production data with many ERP systems and with transaction systems. The big openness has been essential since in many factories we are dealing with the heterogeneous computer environment. The solution should also has mechanisms of the buffering and diagnostics, guaranteeing reaching the information into the right place and in case of the riot of the mistakes – to the buffering of the information and fast detecting causes of problems. Also applying standards have an important role in the system of data exchanging, the same as ISA/S95. The fact of applying the standards is not only organizing the way of the exchange of data, but it also helps to establish a clear function division.

3 Data Exchanging

In the company based on production collecting and management of the information is a crucial element in decision-making process. It leads to increasing the production effectiveness and the quality of computer systems improvement. What is more it is a great aid for functioning the company base on production on all its levels – from machines and production lines, through engineering departments, to all the way of administrative departments. In lowest levels there are applied superior systems of steering and the data acquisition – SCADA, industrial arrangements of steering CNC (Computer Numerical Control), PLC (Programmable Logic Controller), IPC (Industrial PC), sensors, executive elements and different devices of industrial automation. They constitute a specific computer system that is active in a real time, of which controlling machines and production lines is a basic task [7]. In highest levels of the industrial enterprise there applied systems, of which effective leading a production process on the basis of accurate and current production data coming from systems of steering and the data acquisition – MES systems, and systems managing supplies of materials, human resources, finances, enterprise resource planning – ERP systems.

An essential element of the modern MES system is a possibility of a simple integration with systems of industrial automation (PLC, SCADA) and with databases in which production data is stored. MES systems join steering of the production departments, keeping the movement departments, qualities departments and different sources of data into a uniform computer system, using provided standard components software so as: OPC Client, OPC Server and ActiveX. These modern computer technologies in the considerable way are reducing the total cost of MES integration with systems of automation and databases as well as they are securing IT investments in the future. However designing the

communications system isn't a simple task and it requires providing many technical parameters of the computer network, communications protocols, interfaces linking computer systems, as well as considering time requirements, the safety of the data transmission and limitations resulting from the bandwidth of the network and technical requirements [7]. The data acquisition in industrial plant show below – Fig. 3.

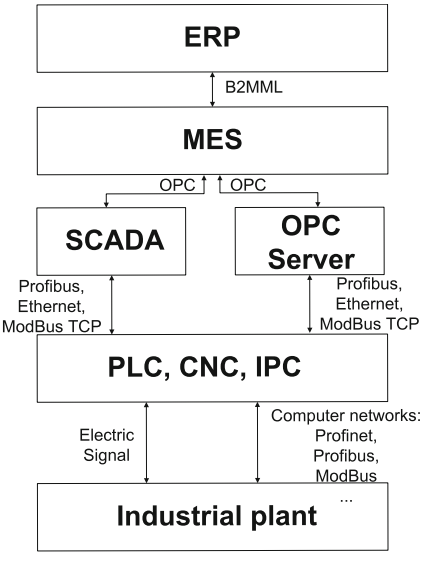


Fig. 3. Data acquisition

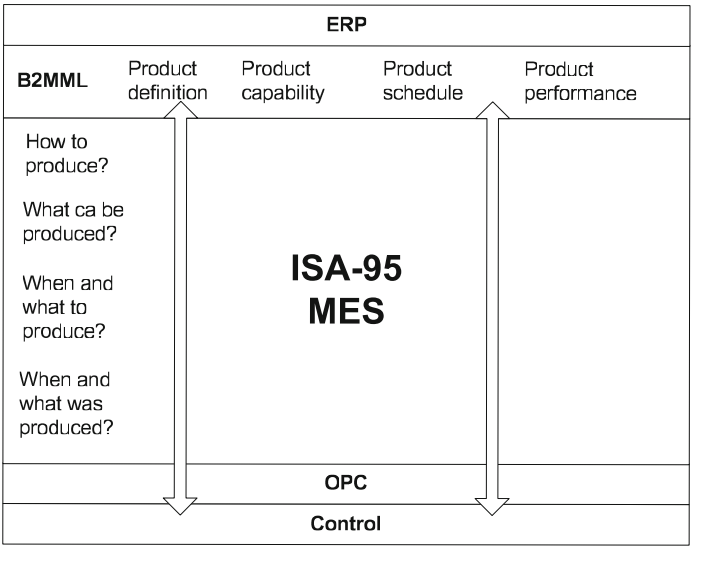


Fig. 4. Exchange of data between layers [8]

Between the ERP system (higher level) and the control layer (lower level), the MES system in accordance with ISA will be crucial for an exchange of data in both directions to help answer the questions: how to produce, what can be produced, when and what to produce, when and what was produced. In the ISA model, a B2MML layer provides connectivity between the ERP system and the MES. At the bottom, OPC (OLE for Process Control) connects data from the control system to the MES – Fig. 4 [8].

4 Conclusions

To sum up, the process of designing communications systems for the automation of production are applied computer systems, applications, interfaces, systems of control, computer networks, databases and functional requirements of final users. It should be put extra attention to analyze requirements concerning the amount and the time of the data transfer between applications. Industrial communications systems like OPC are very well fulfilling their objective in production area where they require transmission of data in the short time e.g. 1 s. These high time requirements are not having to be fulfilled in computer applications so as MES or the ERP. It is possible to imagine the situation, in which generating the report for the manager cockpit will be a second longer, because of load the computer network, and this delay will be acceptable, and not unnoticeable for the end user.

References

1. Kwiecień, A.: Analiza przepływu informacji w komputerowych sieciach przemysłowych. Wydawnictwo pracowni komputerowej J. Skalmierskiego, Gliwice (2000)
2. Cupek, R.: Protokół TCP/IP w systemach wizualizacji procesów przemysłowych. In: Studia Informatica, Gliwice, vol. 22(3) (2001)
3. Kletti, J.: MES – Manufacturing Execution System. Springer, Heidelberg (2007)
4. Lech, P.: Zintegrowane systemy zarządzania ERP/ERP II. Wykorzystanie w biznesie, wdrażanie; Wydawnictwo Difin, Warszawa (2003)
5. Zaborowski, M.: Sterowanie nadążne zasobami przedsiębiorstwa. In: Wydawnictwo pracowni komputerowej J. Skalmierskiego, Gliwice (2008)
6. Cupek, R.: Akwizycja danych z sieci PROFINET CBA do systemów klasy MES. In: Sieci komputerowe. Aplikacje i zastosowania, WKŁ, Warsaw, June 2007, vol. 2, pp. 389–399 (2007)
7. Skura, K., Smalec, Z.: Integracja systemów informatycznych w automatyzacji procesów produkcyjnych. PAR. Pomiary, Automatyka, Robotyka 8(7/8), 6–11 (2005)
8. Siemens Energy & Automation, Inc.: Why integrate MES and ERP? Because you can't afford not to! Process Automation Systems (February 2006)

Load Balancing for Heterogeneous Web Servers

Adam Piórkowski[1], Aleksander Kempny[2],
Adrian Hajduk[1], and Jacek Strzelczyk[1]

[1] Department of Geoinfomatics and Applied Computer Science,
AGH University of Science and Technology, Cracow, Poland
{adam.piorkowski,jacek.strzelczyk}@agh.edu.pl
http://www.agh.edu.pl
[2] Adult Congenital and Valvular Heart Disease Center
University of Muenster, Muenster, Germany
aleksander.kempny@ukmuenster.de
http://www.ukmuenster.de

Abstract. A load balancing issue for heterogeneous web servers is described in this article. The review of algorithms and solutions is shown. The selected Internet service for on-line echocardiography training is presented. The independence of simultaneous requests for this server is proved. Results of experimental tests are presented.

Keywords: load balancing, scalability, web server, minimum response time, throughput, on-line simulator.

1 Introduction

Modern web servers can handle millions of queries, although the performance of a single node is limited. Performance can be continuously increased, if the services are designed so that they can be scaled. The concept of scalability is closely related to load balancing. This technique has been used since the beginning of the first distributed systems, including rich client architecture. Most of the complex web systems use load balancing to improve performance, availability and security [1,2,3,4].

2 Load Balancing in Cluster of Web Servers

Clustering of web servers is a method of constructing scalable Internet services. The basic idea behind the construction of such a service is to set the relay server in a transparent connection to the cluster servers (Fig. 1). This server is called a dispatcher.

There are many implementations of load balancing. Some of them (like OpenSSI for Linux or MS Windows Network Load Balancing) use additional software that have to be installed on each cluster node. This software monitors the load of these nodes, but is dedicated for selected operating systems or web server software. It makes a construction of heterogeneous web servers clusters

A. Kwiecień, P. Gaj, and P. Stera (Eds.): CN 2010, CCIS 79, pp. 189–198, 2010.

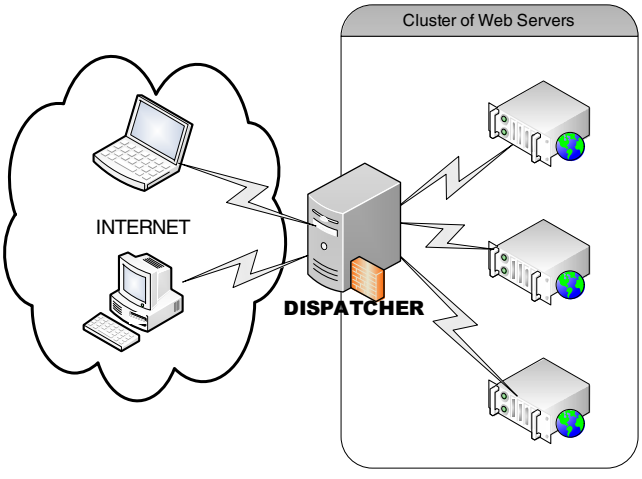


Fig. 1. Cluster of web servers schema

impossible. This article focuses only on some implementations that allow to create heterogeneous web server clusters. This means that it is desirable to create services based on the nodes that use different operating systems and different environments, but process the same requests. Technique which realizes load balancing fulfilling these assumptions is proxy load balancing.

2.1 The Algorithms of Load Balancing

Efficient load balancing requires an appropriate algorithm. There are several basic and common algorithms being discussed further in this paper:

- Round Robin [1],
- Weighted Round Robin,
- Least Loaded [1,2],
- Least Connection [1],
- Weighted Least-Connection,
- Locality-Based Least-Connection,
- Destination Hashing [5],
- Source Hashing [6,5],
- Fair [7],
- Never Queue,
- Shortest Queue First [8],
- Request Counting [9],
- Weighted Traffic Counting [9],
- Pending Request Counting [9].

The Round-Robin algorithm is a well-known algorithm and it is easy to implement. Request Counting algorithm distributes the requests among the various

nodes to ensure that each node gets its configured share of the number of requests [9]. Weighted Traffic Counting algorithm works in a similar way, but the maximum number of requests per server is determined on the network traffic, in bytes. Pending Request Counting algorithm's idea is to keep track of how many requests each worker is assigned at the time. New requests are assigned to the server with the lowest number of active requests. The fair algorithm is based on the standard round-robin algorithm and it tracks busy back end servers and balances the load to non-busy servers [7]. There are many other algorithms, some of them require special knowledge to predict the best scheduling [2].

2.2 Solutions

There are a few implementations of load balancing proxies that enable to create a heterogeneous cluster of web servers. The most interesting from the perspective of authors of scientific portals (non-commercial) are open software solutions. In this article we discussed six of them:

- Apache Server with Mod Proxy Balancer [9] – this is a standard solution used in the most popular web server, it implements three load balancing algorithms : Request Counting, Weighted Traffic Counting and Pending Request Counting,
- Pound [10] – a simple solution with Round-Robin algorithm, distributed under the GPL ,
- NGiNX [7] – this software implements Round Robin and Fair load balancing algorithms, NGiNX is licensed under 2-clause BSD-like license,
- Inlab Balance [6] – is an open source load balancer, under the GPL licensing terms, that implements two algorithms – Round-Robin and Client Hashing,
- HAProxy [5] – this solution implements standard Round Robin algorithm and others – Source Hashing and Destination Hashing,
- Lighttpd [8] – this is one of the famous and efficient web server, but also proxy balancer, that implements four algorithms: Static (fail-over), Round Robin, Cache Array Routing Protocol (similar to Source Hashing) and Shortest Queue First.

3 Features of the Web Servers

Performance is one of the most important aspects of scalable web services. To determine the performance the following factors should be considered:

- average response time (t_{avg}),
- minimum response time (t_{min}),
- throughput (th).

The average response time is a factor which value varies and depends on users load. Its value increases with the number of users.

The minimum response time is the minimum time in which a web request is completed. It depends on the server performance (hardware, software) and the request type. It can be constant for the same conditions. It should be measured at minimal load of a server.

The throughput is a very authoritative factor that describes the performance of a server. It tells how many requests can be processed at the unit of time at the saturation. However, the system that reached the maximum throughput cannot guarantee the acceptable response time.

3.1 The Efficiency of Request Processing

The requests that are processed by a server can be of two types:

- independent request,
- related request.

The independent requests are requests, that do not affect one another. They share resources (for example CPU), which are shared fairly between them. At saturation of a web server with one processor the relationship of minimum response time and throughput for this case can be given by the efficiency factor (1):

$$E_1 = t_{\min} * th \ . \tag{1}$$

The efficiency factor E_1 for series of identical independent request should have the value close to 1.0. For multiprocessor servers the value of E_1 factor should be close to the number of processors. In this case the efficiency factor is given by formula (2), where N – number of processors, there is no vectorization.

$$E = t_{\min} * th/N \ . \tag{2}$$

Another type of queries are the related. Mechanisms of optimization like caching or spooling can make processing shorter for a group of requests. This is for example the case of queries with pool connections to databases [11,12]. In this case the value of efficiency factor is above the number of processors. Some queries can generate a large overhead (for example allocating and deallocating big data tables, frequent context switching) – requests affect each other and the efficiency factor is below the number of processors.

3.2 CT2TEE – An Example of a Web Server

CT2TEE is a novel Internet-based simulator for transesophageal echocardiography (TEE) training [13]. It loads CT data into the memory and processes projections [14] and USG simulations for each request individually. The process of creating a projection is described on Fig. 2.

The output of CT2TEE application can be an image, that is a single frame (of JPG or GIF format, JPG quality: 80%) or an animation (of GIF format). The current version of CT2TEE generates the same projection with different noise pattern, but there will be motion implemented in the future. The GIF

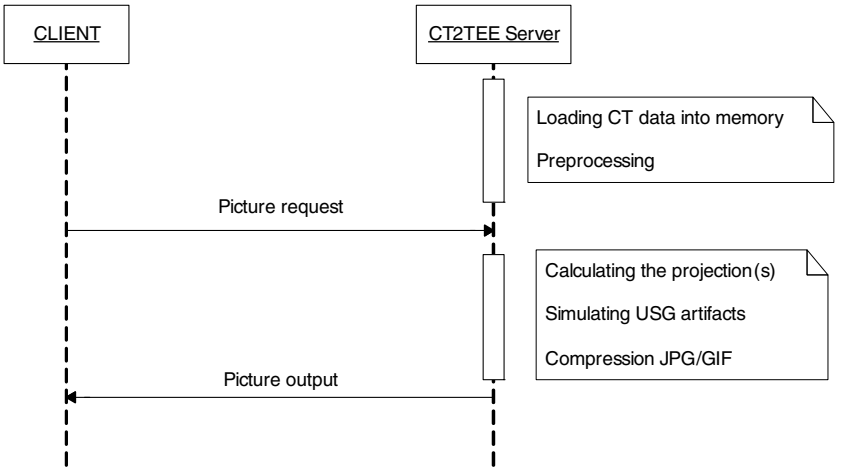


Fig. 2. Diagram for a request processing by CT2TEE server

format generates bigger files than JPG. The one of the most interesting features of CT2TEE application is a fact, that the efficiency factor (2) in this case on the current Internet server of CT2TEE (2 processors) is very close to value 1 (0.99). It is caused by the character of requests – they are calculations that share CPU only. Therefore the CT2TEE application is a good example to test load balancing on a cluster of servers.

4 Tests

The tests have been carried out to assess performance.

4.1 Hardware and Software Environment

The following hardware has been used for the tests:

- for web servers/proxy server/test clients: IBM Blade HS 21, CPU: 2.0 GHz, Intel Xeon (8 cores), RAM 16 GB,
- network: Ethernet 1 Gb, switch.

The following software was used:

- operating systems: Linux Fedora Core 12, Windows Server 2008,
- component environments: Mono 2.4.2.3, .NET 2.0,
- web servers: Mono XSP 2.4.2, IIS 7.0,
- load balancers: Apache Mod Proxy 2.2, NGiNX 0.7.65, Pound 2.5-1, Inlab Balance 3.52, HAProxy 1.4.1 and Lighttpd 1.4.26,
- load testers: JMeter, Apache Bench.

The results given by JMeter and Apache Bench were very similar, so we decided to use JMeter in all cases. The tests were divided into two parts:

- determining individual parameters of servers,
- determining performance of load balancing.

4.2 The Efficiency of Servers

Initially the tests for the main parameters of cluster servers have been carried out. The results (minimum times of requests $t_{\min}$ [ms], throughputs th [req/s] and efficiency factors E) are presented in Table 1.

Table 1. Minimum times of requests $t_{\min}$ [ms], throughputs th [req/s] and efficiency E for cluster servers with CT2TEE application

OS	server	G0			G1			G4		
		$t_{\min}$	th	E	$t_{\min}$	th	E	$t_{\min}$	th	E
Linux XSP	s1	193	33.1	0.799	203	38.5	0.977	760	10.0	0.950
	s2	188	34.1	0.801	200	37.6	0.940	762	10.4	0.991
	s3	190	34.8	0.827	201	36.8	0.925	767	10.3	0.988
	s4	191	33.4	0.797	197	36.8	0.906	776	10.1	0.980
WinSvr IIS	s1	140	52.5	0.919	137	47.4	0.812	517	14.9	0.963
	s2	141	49.8	0.878	137	48.2	0.825	519	15.2	0.986
	s3	138	53.2	0.918	136	46.3	0.787	519	14.9	0.967
	s4	139	51.0	0.886	137	50.3	0.861	518	14.8	0.958

4.3 The Performance of Load Balancing

Experiments for the six solutions (Apache Mod Proxy, NGiNX, Pound, Inlab, HAProxy and Lighttpd) were done. In the case of Apache Mod Proxy we tested all three algorithms: Request Counting (RC), Weighted Traffic Counting (WTC) and Pending Request Counting (PRC). In the case of Inlab we tested only the Round Robin algorithm. In the case of HAProxy we tested three algorithms: Round Robin (RR), Source Hashing (SRC) and Destination Hashing (URI). In the case of Lighttpd we tested four algorithms: Cache Array Routing Protocol (CARP), Round Robin (RR), Static (failover balancing, STAT) and Shortest Queue First (SQF). There were two kinds of heterogeneous environments:

- 3 servers running Linux+XSP and 1 server running Windows+IIS,
- 1 server running Linux+XSP and 3 servers running Windows+IIS.

We selected output to be a JPG (G0, small files of average size 20 KB) and GIF (G1, bigger files of average size 80 KB for 1 frame and 300 KB for 4 frames – G4). The G4 output was processed much longer than others.

The results are presented in Table 2 and on the plots (Fig. 3, 4). To compare these results with the maximum performance of a system an additional column (MAX) is placed. It contains sums of all server throughputs for the tested cases.

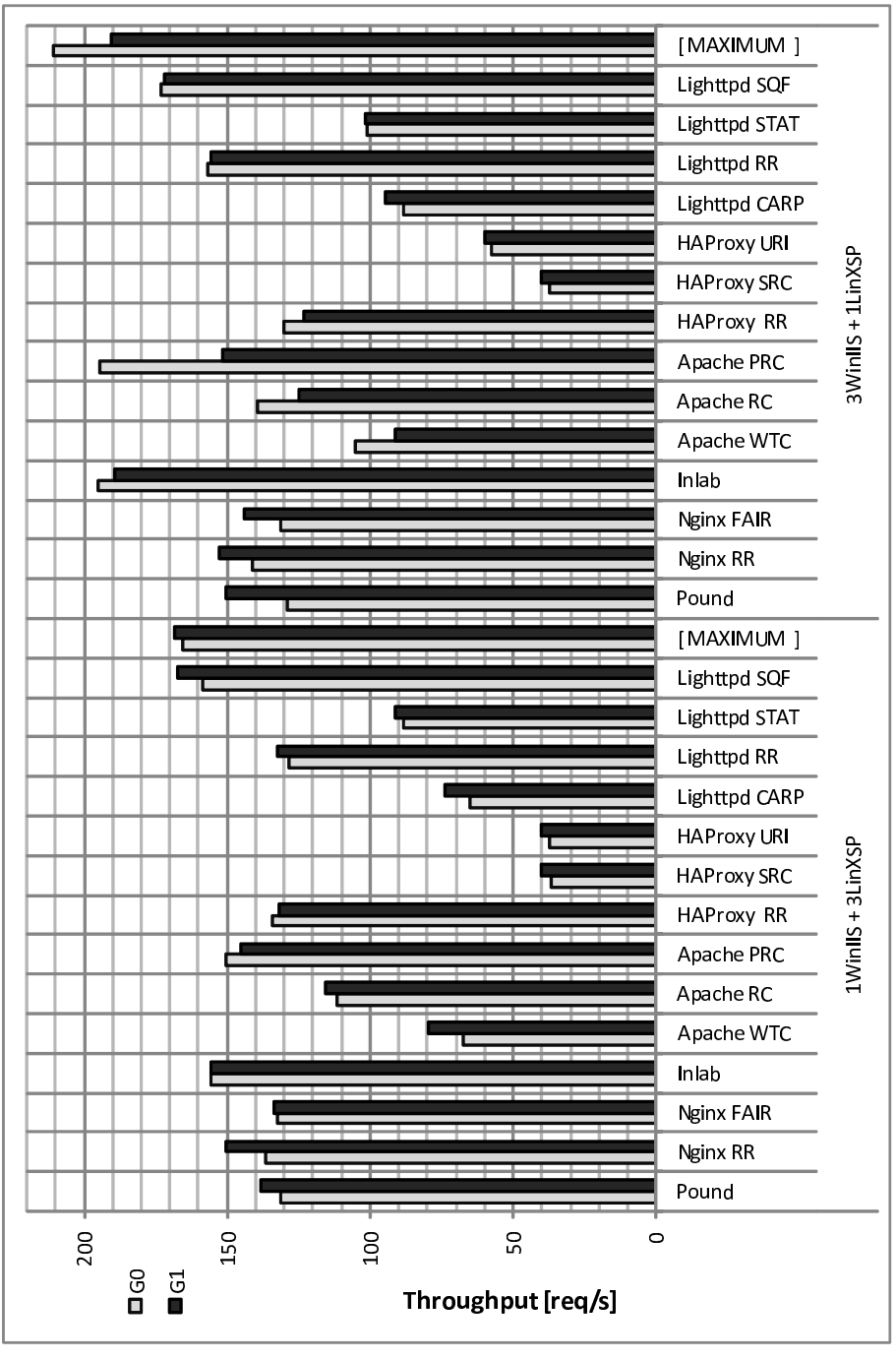


Fig. 3. The results of load balancing tests for G0 (JPG) and G1 (GIF, 1 frame) output

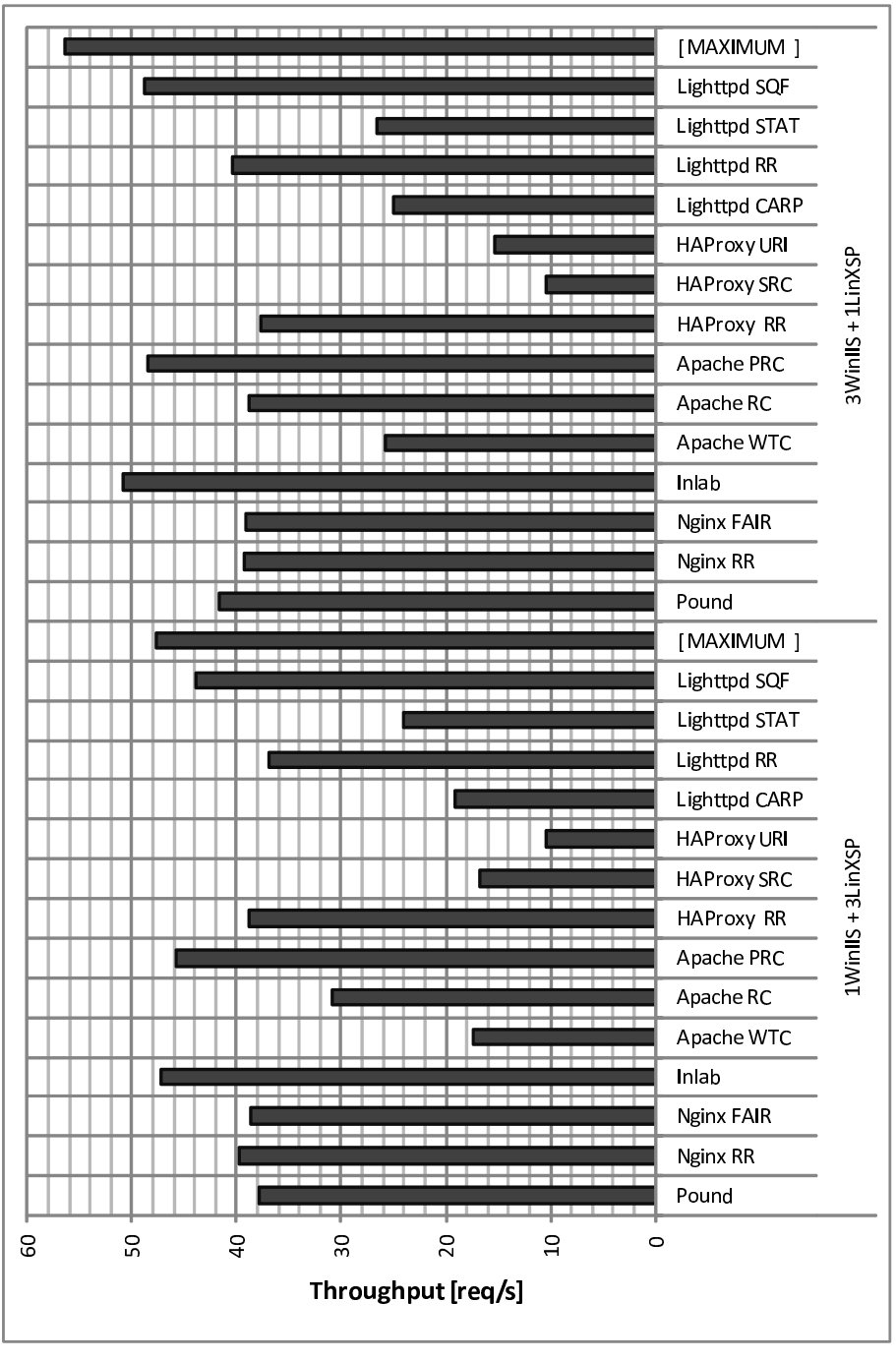


Fig. 4. The results of load balancing tests for G4 (an animated GIF, 4 frames) output

Table 2. The performance of load balancing – throughputs [req/s]

Load Balancer	1 WinIIS + 3 LinXSP			3 WinIIS + 1 LinXSP		
	G0	G1	G4	G0	G1	G4
Pound	131.50	138.40	37.80	129.10	150.20	41.60
NGiNX RR	136.30	150.40	39.70	141.00	152.60	39.20
NGiNX FAIR	132.20	133.40	38.60	131.50	144.30	39.10
Inlab	155.40	155.40	47.10	195.40	189.30	50.80
Apache WTC	67.10	79.80	17.30	105.30	91.30	25.70
Apache RC	111.40	115.60	30.90	139.60	124.60	38.80
Apache PRC	150.20	145.30	45.70	194.30	151.40	48.40
HAProxy RR	133.90	131.70	38.70	130.00	122.90	37.60
HAProxy SRC	36.70	39.90	16.70	36.90	40.10	10.40
HAProxy URI	37.40	40.10	10.40	57.40	59.50	15.30
Lighttpd CARP	65.10	74.00	19.10	88.50	94.60	24.90
Lighttpd RR	128.20	132.50	36.80	156.90	155.60	40.30
Lighttpd STAT	88.50	91.00	24.00	101.30	101.50	26.60
Lighttpd SQF	158.70	167.10	43.80	172.90	171.80	48.80
MAX	165.30	168.50	47.60	211.10	190.30	56.40

5 Summary

The tests have proved that the use of proxy load balancers effectively increases system throughput. Some of the servers provide several algorithms, the choice of one of them is crucial for performance. For tested solutions using the CT2TEE application server the best results are reached by Inlab, Lighttpd with Shortest Queue First algorithm and Apache Mod Proxy with Pending Request Counting algorithm. Slightly smaller througputs were achieved for the other solutions that use Round Robin algorithm – NGiNX (Round Robin and Fair algorithms), Pound, HAProxy (with Round Robin algorithm) and Lighttpd (with Round Robin algorithm). The worst results were produced by proxy balancers with Source Hashing and Destination Hashing algorithms. Apache Mod Proxy with Weighted Traffic Counting algorithm is over two times slower than the best results, but this algorithm is better in case of variable size of outputs. As we proved the choice of solution and algorithm is very important to reach the maximum performance of web server clusters.

Acknowledgment. This work was financed by the AGH – University of Science and Technology, Faculty of Geology, Geophysics and Environmental Protection as a part of statutory project number 11.11.140.561.

References

1. Teo, Y.M., Ayani, R.: Comparison of load balancing strategies on cluster-based web servers. Simulation 77(6), 185–195 (2001)
2. Guo, J., Bhuyan, L.N.: Load Balancing in a Cluster-Based Web Server for Multimedia Applications. IEEE Transactions On Parallel And Distributed Systems 17(11) (2006)

3. Ungureanu, V., Melamed, B., Katehakis, M.: Effective load balancing for cluster-based servers employing job preemption. Performance Evaluation 65(8), 606–622 (2008)
4. Wrzuszczak, J.: Auction mechanism in management of processing nodes in a computer cluster. Contemporary Aspects of Computer Networks 2, 259–265 (2008)
5. HAProxy – The Reliable, High Performance TCP/HTTP Load Balancer, http://haproxy.1wt.eu/
6. Inlab Balance, http://www.inlab.de/balance.pdf
7. NGiNX – HTTP and reverse proxy server, http://nginx.org/en/
8. Lighttpd – fly light, http://www.lighttpd.net/
9. Mod Proxy Balancer – Apache HTTP Server, http://httpd.apache.org/docs/2.1/mod/mod_proxy_balancer.html
10. Pound – Reverse-Proxy and Load-Balancer, http://www.apsis.ch/pound/
11. Bogardi-Meszoly, A., Szitas, Z., Levendovszky, T., Charaf, H.: Investigating Factors Influencing the Response Time in ASP.NET Web Applications. In: Bozanis, P., Houstis, E.N. (eds.) PCI 2005. LNCS, vol. 3746, pp. 223–233. Springer, Heidelberg (2005)
12. Gabiga, A., Piórkowski, A., Danek, T.: Efficiency analysis of servlet technology in selected database operations. In: Studia Informatica, vol. 30(84,2B) (2009)
13. Kempny, A., Piórkowski, A.: CT2TEE – a Novel, Internet-Based Simulator of Transoesophageal Echocardiography in Congenital Heart Disease. In: Kardiol Pol. 2010, vol. 68 (2010)
14. Piorkowski, A., Jajesnica, L., Szostek, K.: Creating 3D Web-Based Viewing Services for DICOM Images. In: Kwiecień, A., Gaj, P., Stera, P. (eds.) 16th Conference on Computer Networks, CN 2009,, Wisła, Poland. CCIS, vol. 39. Springer, Heidelberg (2009)

A Choice of Optimal Packet Dropping Function for Active Queue Management

Dariusz Rafał Augustyn[1], Adam Domański[1], and Joanna Domańska[2]

[1] Institute of Informatics, Silesian Technical University
Akademicka 16, 44–100 Gliwice, Poland
{draugustyn,adamd}@polsl.pl
[2] Institute of Theoretical and Applied Informatics, Polish Academy of Sciences
Baltycka 5, 44–100 Gliwice, Poland
joanna@iitis.gliwice.pl

Abstract. Algorithms of queue management in IP routers determine which packet should be deleted when necessary. The article investigates the influence of packet rejection probability function on the performance, i.e. response time for in case NLRED queues. For standard NLRED the linear packet dropping function is replaced by a quadratic function. Solution presented in this article based on orthogonal polynomials. Depending on a particular criterion of optimization a proper non-linear function may be chosen, but using averaged criterion leads up to choose the linear one as an optimal one.

Keywords: active queue menagement, nonlinear RED.

1 Introduction

Algorithms of queue management at IP routers determine which packet should be deleted when necessary [1,2]. The Active Queue Management, recommended now by IETF, enhances the efficiency of transfers and cooperates with TCP congestion window mechanism in adapting the flows intensity to the congestion at a network [3].

This paper describes another approach to packed dropping function used in Active Queue Management.

Sections 2 and 3 give basic and improve notions on active queue management. Section 4 gives simulation models of the considered NLRED. Section 5 discusses numerical results, some conclusions are given in Sect. 6.

2 Active Queue Management

In *passive* queue management, packets coming to a buffer are rejected only if there is no space in the buffer to store them, hence the senders have no earlier warning on the danger of growing congestion. In this case all packets coming during saturation of the buffer are lost. The existing schemes may differ on the choice of packet to be deleted (end of the tail, head of the tail, random) [4].

A. Kwiecień, P. Gaj, and P. Stera (Eds.): CN 2010, CCIS 79, pp. 199–206, 2010.

During a saturation period all connections are affected and all react in the same way, hence they become synchronized. To enhance the throughput and fairness of the link sharing, also to eliminate the synchronization, the Internet Engineering Task Force (IETF) recommends *active* algorithms of buffer management. They incorporate mechanisms of preventive packet dropping when there is still place to store some packets, to advertise that the queue is growing and the danger of congestion is ahead. The probability of packet rejection is growing together with the level of congestion. The packets are dropped randomly, hence only chosen users are notified and the global synchronization of connections is avoided. A detailed discussion of the active queue management goals may be found in [3].

The RED (Random Early Detection) algorithm was proposed by IETF to enhance the transmission via IP routers. It was primarily described by Sally Floyd and Van Jacobson in [5]. Its idea is based on a drop function giving probability that a packet is rejected. The argument avg of this function is a weighted moving average queue length, acting as a low-pass filter and calculated at the arrival of each packet as

$$avg = (1 - w)avg' + wq \tag{1}$$

where avg' is the previous value of avg, q is the current queue length and w is a weight determining the importance of the instantaneous queue length, typically $w \ll 1$. If w is too small, the reaction on arising congestion is too slow, if w is too large, the algorithm is too sensitive on ephemeral changes of the queue (noise). Articles [5,6] recommend $w = 0.001$ or $w = 0.002$, and [7] shows the efficiency of $w = 0.05$ and $w = 0.07$. Article [8] analyses the influence of w on queuing time fluctuations, obviously the larger w, the higher fluctuations. In RED drop function there are two thresholds Min_{th} and Max_{th}. If $avg < Min_{\text{th}}$ all packets are admitted, if $Min_{\text{th}} < avg < Max_{\text{th}}$ then dropping probability p is growing linearly from 0 to $p_{\max}$:

$$p = p_{\max}\frac{avg - Min_{\text{th}}}{Max_{\text{th}} - Min_{\text{th}}} \tag{2}$$

and if $avg > Max_{\text{th}}$ then all packets are dropped. The value of $p_{\max}$ has also a strong influence on the RED performance: if it is too large, the overall throughput is unnecessarily choked and if it's too small the danger of synchronization arises; [6] recommends $p_{\max} = 0.1$. The problem of the choice of parameters is still discussed, see e.g. [9,10]. The mean avg may be also determined in other way, see [11] for discussion. Despite of evident highlights, RED has also such drawbacks as low throughput, unfair bandwidth sharing, introduction of variable latency, deterioration of network stability. Therefore numerous propositions of basic algorithms improvements appear, their comparison may be found e.g. in [12].

For NLRED the linear packet dropping function is replaced by a quadratic function [13]. For this solution dropping probability p is growing non-linearly:

$$p = \begin{cases} 0 & \text{for } x < Min_{\text{th}} \\ \left(\frac{avg - Min_{\text{th}}}{Max_{\text{th}} - Min_{\text{th}}}\right)^2 max_{\text{p}} & \text{for } Min_{\text{th}} \le x \le Max_{\text{th}} \\ 1 & \text{for } x > Max_{\text{th}} \end{cases} . \tag{3}$$

Figure 1 shows dropping functions for RED and NLRED.

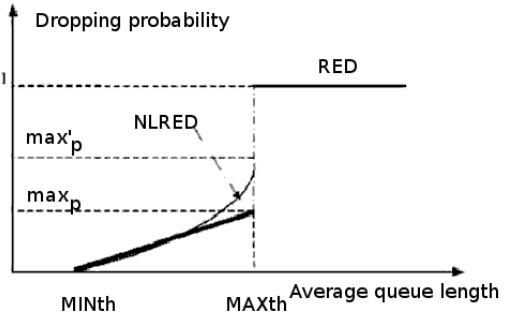


Fig. 1. Dropping functions for RED and NLRED

3 The Theoretical Background

This work bases on methods known in calculus of variations. The some function f(x) with domain $[0, l]$ can be approximated by as a finite linear combination of basis functions:

$$f(x) = \sum_{i=1}^{N} a_j \Phi_j(x) \ , \tag{4}$$

where a_j are undetermined parameters and Φ_j can be a series of orthogonal polynomials

$$\Phi_j = x^{j-1}(l - x) \ . \tag{5}$$

Optimal values of a_j can be numerically obtained by finding minimum of some functional J implicitly defined on f. In real application only a few elements (here $N = 2$) are required to obtain accurate approximation of f. Using approach described above (Equations (4) and (5)) the p-function i.e. the probability of packet dropping function is defined as follows [14]:

$$p(x, a_1, a_2, p_{\max}) = \begin{cases} 0 & \text{for } x < Min_{\text{th}} \\ \varphi_0(x) + a_1\varphi_1(x) + a_2\varphi_2(x) & \text{for } Min_{\text{th}} \le x \le Max_{\text{th}} \\ 1 & \text{for } x > Max_{\text{th}} \end{cases} \tag{6}$$

where basis functions are defined:

$$\varphi_0(x) = p_{max}\frac{x - Min_{\text{th}}}{Max_{\text{th}} - Min_{\text{th}}} \ , \tag{7}$$

$$\varphi_1(x) = (x - Min_{\text{th}})(Max_{\text{th}} - x) \ , \tag{8}$$

$$\varphi_2(x) = (x - Min_{\text{th}})^2(Max_{\text{th}} - x) \tag{9}$$

and x is an queue size.

The problem of choosing the optimal function $p(x)$ can be defined as finding such $p(x)$ where a functional J is minimal. The functional J can be defined using such parameters an average queue length or average waiting time or number of all dropped packets or number of packets dropped by RED, eventually some expression involving any mentioned parameters. Applied definitions of J are considered in Sect. 5. Because $p(x)$ is a polynomial determined by values of $(a_1, a_2, p_{\max})$, finding optimal $p(x)$ is equivalent to obtain the minimum of 3-variate function $J(a_1, a_2, p_{\max})$. Obtaining the domain of J i.e. the region in 3-dimensional space $A_1 \times A_2 \times P_{\max}$ where $p(x)$ has sense $(p(x) >= 0)$ is not trivial. The extremums of $p(x)$ can be easily found by solving the following equation:

$$\frac{dp(x)}{dx} = 0 \ . \tag{10}$$

(10) is equivalent to a square equation:

$$\begin{aligned} &a_1(Max_{\text{th}} - Min_{\text{th}})(Max_{\text{th}} + Min_{\text{th}} - 2x) + \\ &+a_1(Max_{\text{th}} - Min_{\text{th}})(x - Min_{\text{th}})(2Max_{\text{th}} + Min_{\text{th}} - 3x) + p_{\max} = 0 \ . \end{aligned} \tag{11}$$

Using (11) we can find two roots x_1, x_2. Complicated expressions defining x_1 and x_2 are not presented here. Using x_1 and x_2 we can define conditions for p-function values in points x_1 and x_2:

$$0 <= p(x_1) <= 1 \wedge 0 <= p(x_2) <= 1 \ . \tag{12}$$

To illustrate a complicated shape of domain $(a_1, a_2, p_{\max})$ satisfying conditions (12) we show square inequalities form (12) in 2-dimensional space (a_1, a_2) for given values $Mix_{\text{th}} = 100, Max_{\text{th}} = 200$ and $p_{\max} = 0.8$ on Fig. 2. The dark area shows pairs of (a_1, a_2) satisfying conditions. The domain of i.e. 3-dimensional

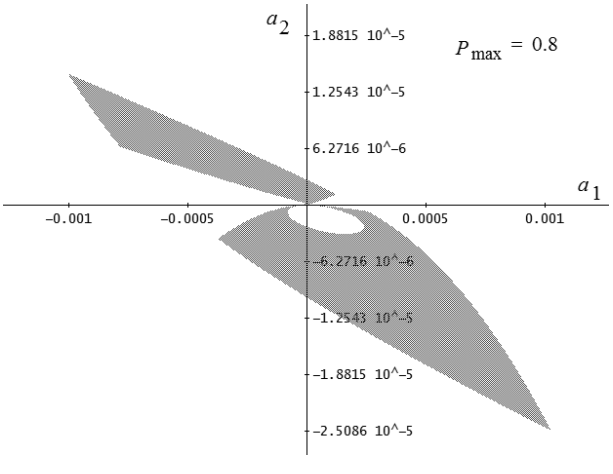


Fig. 2. 2-dimensional region of meaningful values of (a_1, a_2) pairs for $p_{\max} = 0.8, Min_{\text{th}} = 100, Max_{\text{th}} = 200$

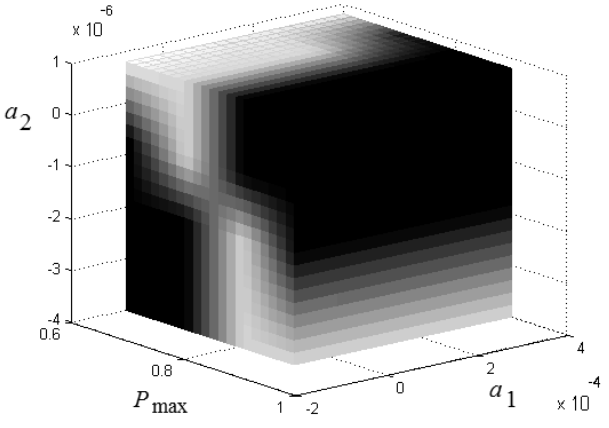


Fig. 3. Domain of functional J in space $A_1 \times A_2 \times P_{\max}$ for $Min_{\text{th}} = 100, Max_{\text{th}} = 200$

subspace $(a_1, a_2, p_{\max})$ for $Min_{\text{th}} = 100, Max_{\text{th}} = 200$ is presented on Fig. 3. The simple condition $0.6 <= p_{\max} <= 0.9$ was added, too. The white region of $A_1 \times A_2 \times P_{\max}$ corresponds to the assumed domain of J.

The domain of J was obtained using matlab.

4 Simulation Models

The simulation evaluations were carried out with the use of OMNeT++ simulation framework of discrete events. The OMNeT++ is the modular, component-based simulator, with a graphical environment, mainly designed for simulation of communication networks, queuing networks and performance evaluation. The framework is very popular in research and for academic purposes [15], [16].

Searching for optimal shape of probability dropping function requires a large number of performed simulations. A single simulation takes a long time. For these reasons NLRED is studied only in open-loop scenario and the simulation model developed for the purpose of research is very simple and contains only one source and only one queue. For the same reasons we assume only Poisson traffic. Simulation for self-similar sources takes much longer time than the simulation for the Poisson sources. However, the obtained results allowed us to reduce the search area of the best non-linear probability dropping function parameters. Hence, in future work we plan to repeat experiments for the self-similar traffic and for different traffic intensity.

The simulation parameters were as follows: input traffic intensity (for geometric traffic) was chosen as $\alpha = 0.5$, parameter μ of geometric distribution of service times (probability of the end of service within a current time-slot) was $\mu = 0.5$. The NLRED parameters had the following values: buffer size 250 packets, threshold values $Min_{\text{th}} = 100$ and $Max_{\text{th}} = 200$, $p_{\max}$ changes from 0.6 to 0.9.

5 Experimental Results

Results of simulations will be described using a set of sample points $(a_1, a_2, p_{\max})$ shown on Fig. 4. The domain of J from Fig. 3 includes the set of points form Fig. 4.

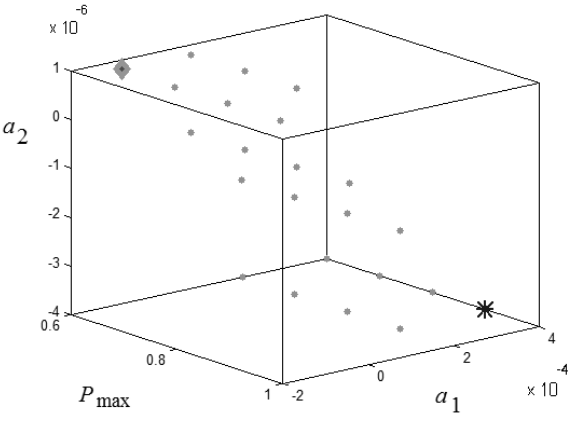


Fig. 4. 23 sample points from the domain of J functional

Set of p-function for all sample points form set of $(a_1, a_2, p_{\max})$ form Fig. 4 was presented on Fig. 5.

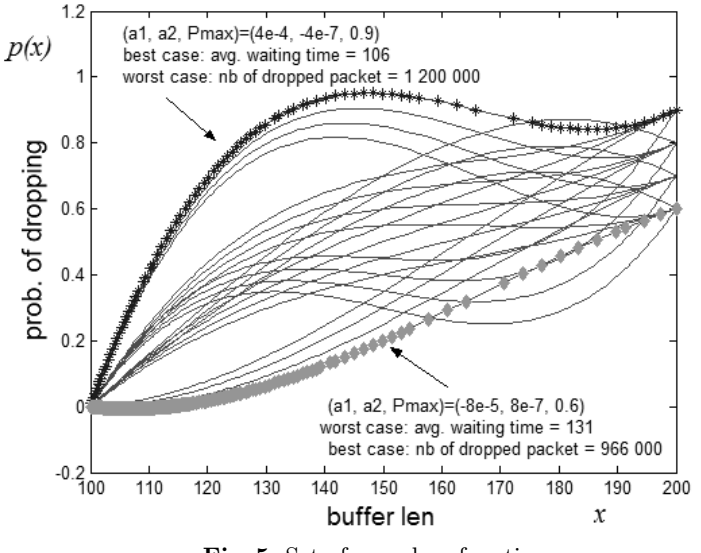


Fig. 5. Set of sample p-function

Assuming average waiting time as functional J we obtained the best case and worst one as it is shown on Fig. 5. The best case with the minimal value of average waiting time is denoted using asterisks. The worst case with the maximal value of average waiting time is denoted using diamond symbols. Using another criterion i.e. number of all dropped packets as value of J we obtained almost reverse results. They are shown on Fig. 5 too. The best case with the minimal value of number of all dropped packets is denoted using diamonds and the worst one (with maximum) is denoted by asterisks.

To take into account results using all kind of parameters:

- average queue length,
- average waiting time,
- number of all dropped packets,
- number of packets dropped by RED

we assumed some compound criterion.

Values of each parameter were normalized to interval $[0, 1]$. We assumed a general optimization criterion as a mean of all normalized parameters (with equal weight for each parameter). For such generalized functional J we found the optimal values of $(a_1, a_2, p_{\max})$. The optimal p-function proves to be a linear one. This result is shown on Fig. 6.

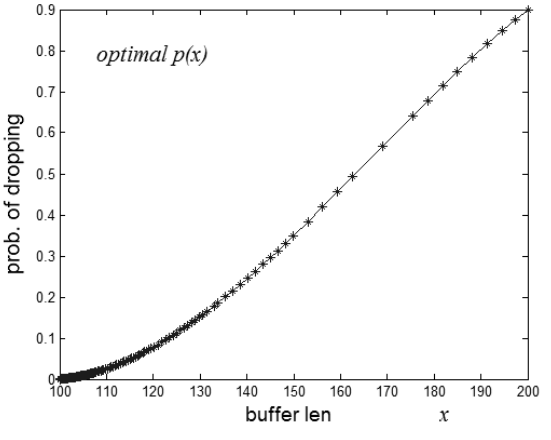


Fig. 6. Optimal almost linear function of probability of dropping function for assumed compound criterion

6 Conclusions

In this article we presented the problem of packet loss probability function and its influence the behavior of the AQM queue. We also described the problem of choosing the optimal shape of dropping packet function for RED alghorithm. Our solution we based on orthogonal polynomials. During the tests we analyzed the

following parameters of the transmission with AQM: the length of the queue, the number of rejected packets and waiting times in queues. Simulation studies have shown that finding the optimal dropping packet function, for certain network conditions, is possible. Depending on a particular criterion of optimization (e.g. average waiting time or number of all dropped packets) a proper form of non-linear function may be chosen, but using averaged criterion leads up to choose the linear one as an optimal one.

References

1. Domańska, J., Domański, A., Czachórski, T.: Implementation of modified AQM mechanisms in IP routers. Journal of Communications Software and Systems 4(1) (March 2008)
2. Domańska, J., Domański, A.: Active Queue Management in Linux based routers. IWSI (2008)
3. Braden, B., Clark, D., Crowcroft, J., Davie, B., Deering, S., Estrin, D., Floyd, S., Jacobson, V., Minshall, G., Partridge, C., Peterson, L., Ramakrishnan, K., Shenker, S., Wroclawski, J., Zhang, L.: Recommendations on queue management and congestion avoidance in the internet. RFC 2309, IETF (1998)
4. Domańska, J., Domański, A., Czachórski, T.: The Drop-From-Front Strategy in AQM. In: Koucheryavy, Y., Harju, J., Sayenko, A. (eds.) NEW2AN 2007. LNCS, vol. 4712, pp. 61–72. Springer, Heidelberg (2007)
5. Floyd, S., Jacobson, V.: Random early detection gateways for congestion avoidance. IEEE/ACM Transactions on Networking 1(4), 397–413 (1993)
6. Floyd, S.: Discussions of setting parameters (1997), http://www.icir.org/floyd/REDparameters.txt
7. Zheng, B., Atiquzzaman, M.: A framework to determine the optimal weight parameter of red in next generation internet routers. The University of Dayton, Department of Electrical and Computer Engineering, Tech. Rep. (2000)
8. May, M., Bonald, T., Bolot, J.: Analytic evaluation of red performance. In: IEEE Infocom 2000, Tel-Aviv, Izrael (2000)
9. Chang Feng, W., Kandlur, D., Saha, D.: Adaptive packet marking for maintaining end to end throughput in a differentiated service internet. IEEE/ACM Transactions on Networking 7(5), 685–697 (1999)
10. May, M., Diot, C., Lyles, B., Bolot, J.: Influence of active queue management parameters on aggregate traffic performance. Technical report, Research Report, Institut de Recherche en Informatique et en Automatique (2000)
11. Zheng, B., Atiquzzaman, M.: Low pass filter/over drop avoidance (lpf/oda): An algorithm to improve the response time of red gateways. Int. Journal of Communication Systems 15(10), 899–906 (2002)
12. Hassan, M., Jain, R.: High Performance TCP/IP Networking. Pearson Education Inc. (2004)
13. Zhou, K., Yeung, K.L., Li, V.: Nonlinear RED: Asimple yet efficient active queue management scheme. Computer Networks 50, 3784–3794 (2006)
14. Augustyn, D.R., Domański, A., Domańska, A.: Active Queue Management with non linear packet dropping function. In: Performance Modelling and Evaluation of Heterogenous Networks HET-NETs, pp. 133–143 (2010)
15. OMNET++ homepage, http://www.omnetpp.org/
16. Domanska, J., Grochla, K., Nowak, S.: Symulator zdarzeń dyskretnych OMNeT++. Wyd. Wyzsza Szkola Biznesu w Dabrowie Górniczej, Dabrowa Górnicza (2009)

A Contribution to the Fair Scheduling for the TCP and UDP Streams

Adam Domański[1], Joanna Domańska[2],
Sławomir Nowak[2], and Tadeusz Czachórski[2]

[1] Institute of Informatics, Silesian Technical University
Akademicka 16, 44–100 Gliwice, Poland
adamd@polsl.pl
[2] Institute of Theoretical and Applied Informatics, Polish Academy of Sciences
Baltycka 5, 44–100 Gliwice, Poland
joanna@iitis.gliwice.pl

Abstract. Nowadays, a lot of Internet applications is using UDP protocol for data transport. Congestion control mechanisms built into TCP protocol in conjunction with the Active Queue Management mechanisms, during normal operation of the Internet network, favor UDP streams. The article investigates the influence of active queue menagement and scheduling algorithm on fairness of TCP and UDP data streams.

Keywords: fairness queueing, active queue menagement, conquestion control.

1 Introduction

The algorithms of queue management in IP routers determine which packet should be deleted when necessary. The active queue management, recommended now by IETF, enhances the efficiency of transfers and cooperate with TCP congestion window mechanism in adapting the flows intensity to the congestion in the network [1,2]. Nowadays, a lot of Internet applications using UDP protocol to transport the data. For UDP traffic the examined parameters of the queue with AQM are significantly worse. The congestion control mechanisms built into TCP protocol in conjunction with the Active Queue Management mechanisms, during normal operation of the Internet network, favor UDP streams [3,4,5,6]. In this article, authors try to describe the problem of scheduling packets in the node allowing fair treatment both types of data streams (TCP and UDP).

Most AQM algorithms (RED, REM, BLUE, PI) do not differentiate between types of packages [7,8,9]. Some of them ensure the equitable distribution of resources among active streams (WRED, SFBLUE, CHOKe) [10,11,12]. However, their usefulness for the solution of the problem described above seems to be (by authors of this article) questionable. Algorithms based on stochastic or weighted fair queuing do not consider, as shown later in this article, the problem of TCP self-discrimination.

More interesting is the CHOKe algorithm, where single FIFO buffer in output router port is assumed [13,14]. The algorithm randomly penalizes unresponsive,

A. Kwiecień, P. Gaj, and P. Stera (Eds.): CN 2010, CCIS 79, pp. 207–216, 2010.

excessive flows. CHOKe calculates the average occupancy of the buffer using an exponential moving average and sets two threshold, *min* and *max* (like RED algorithm does). If the moving average excess *max* threshold every arriving packed is deleted. When average is less the *max* but greater then *min* arriving packets are compared to randomly selected packets. If they represent the same flow, both are dropped. When average is less the *min* every acets are queued into buffer.

When implemented, CHOKe outperforms RED and Drop Tail in UDP fairness (simulation evaluation, eg. [15], analytical results, eg. [16]) however, the problem of identifying the package in the buffer seems to be computationally complex. On the other hand some articles show that CHOKe aims to approximate fair queuing at a minimal implementation overhead [14]. The potential drawback is, that the algorithm may not correctly work for traffic associated with the exchange of P2P data or DDoS attacks (a large number of small transmissions).

Control of UDP packets is currently perceived as a problem. One solution is to change the rules of the UDP and implementation over UDP transport protocol conguestion control mechanisms [17,18].

For these reasons, we proposed the simple solution, based only on two queues, the first for the TCP stream, the other for UDP.

In this article we present simulation results. The simulation evaluations were carried out with the use of OMNeT++ (in version 4.0) simulation framework extended with the INET package. We add some improvement to the INET implementation. The extensions include: PRIO and SFQ scheduling (with special modifications), new sets of parameters and some new statistics, distribution and traffic scenarios.

Section 2 gives basic notions on the transport layer congestion control and active queue management. Section 3 shortly presents simulation model. Section 4 discusses numerical results. Some conclusions are given in section 5.

2 The Transport Layer Conguestion Control and Active Queue Menagement

The Internet applications can generate streams of network packets with different traffic profiles. Generally we can distinguish two types of traffic: stream traffic and elastic traffic [19]. Applications VoIP or VoD generate the stream network traffic. They often use the UDP protocol to transport data. Research shows that UDP packets traffic has recently been significant and the number of applications that use UDP growing. In contrast to the traffic stream, source adjusting the speed of sending data according to load the network, generate the elastic traffic. An example of such sources are applications that use the TCP protocol. There are many implementations of TCP [20]. The most important modification, in relation to the original version [6] was the introduction of a mechanism to prevent overloading the network (congestion avoidance) [21]. Currently there are two types of TCP congestion management in the network. The first type is based on packet loss during transmission. The most popular protocols in this family are TCP newRENO and TCP Sack [22]. Basically, for such protocols sources

reduce the transmission speed as a consequence packet loss. For the second type of protocol (TCP VEGAS) packet generation rate depends on the time delays in transmission. In Figure 1 one may see a significant reduction of transmission speed for packet loss for TCP Reno mechanism. For TCP Vegas (Fig. 2), trasmission speed is set at a middle level that causes no loss in the queues.

In *passive* queue management, packets coming to a buffer are rejected only if there is no space in the buffer to store them and the senders have no earlier warning on the danger of growing congestion. In this case all packets coming during saturation of the buffer are lost. To enhance the throughput and fairness of the link sharing, also to eliminate the synchronisation, the Internet Engineering Task Force (IETF) recommends *active* algorithms of buffer management [24,25]. They incorporate mechanisms of preventive packet dropping when there is still place to store some packets, to advertise that the queue is growing and the danger of

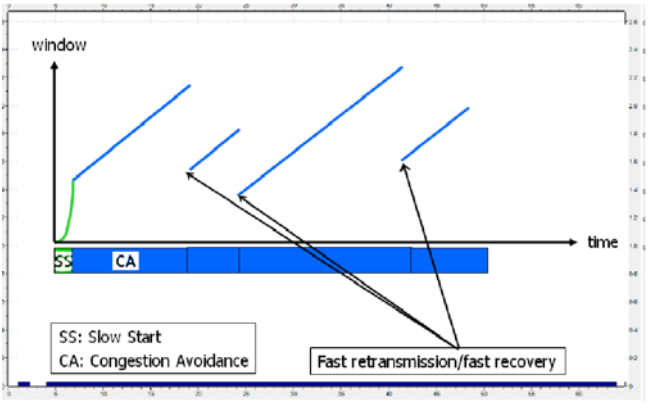


Fig. 1. Resizing the window transmission for TCP Reno protocol [23]

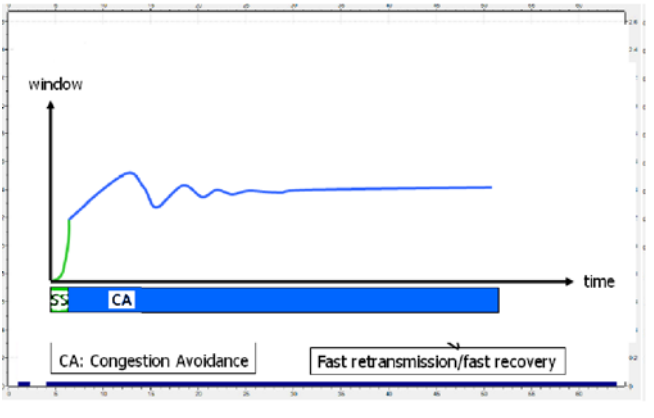


Fig. 2. Resizing the window transmission for TCP Vegas protocol [23]

congestion is ahead. The probability of packet rejection is growing together with the level of congestion [27,28]. The packets are dropped randomly, hence only chosen users are notified and the global synchronisation of connections is avoided [26]. The rejection of the package from the queue lowers the transmission speed. Unfortunately, this situation occurs only for the TCP transmissions. Discussed above mechanism must lead to unequal treatment of TCP and UDP streams during the competition for a common link.

3 Simulation Models

The simulation evaluations were carried out with the use of OMNeT++ (in version 4.0) simulation framework extended with the INET package. The OMNeT++ is the modular, component-based simulator, with an Eclipse-based IDE and a graphical environment, mainly designed for simulation of communication networks, queuing networks and performance evaluation. The framework is very popular in research and for academic purposes [29,30]. The INET Framework is the communication networks simulation extension for the OMNeT++ simulation environment and contains models for several Internet protocols: UDP, TCP, SCTP, IP, IPv6, Ethernet, PPP, IEEE 802.11, MPLS, OSPF, etc. [31].

The simulated network was based on the example provided with the INET package to evaluate the behavior of the queues in a simple network with different traffic scenarios. The INET built in queue algorithms are drop tail queue and RED tail drop algorithm. The network's topology is presented on Fig. 3.

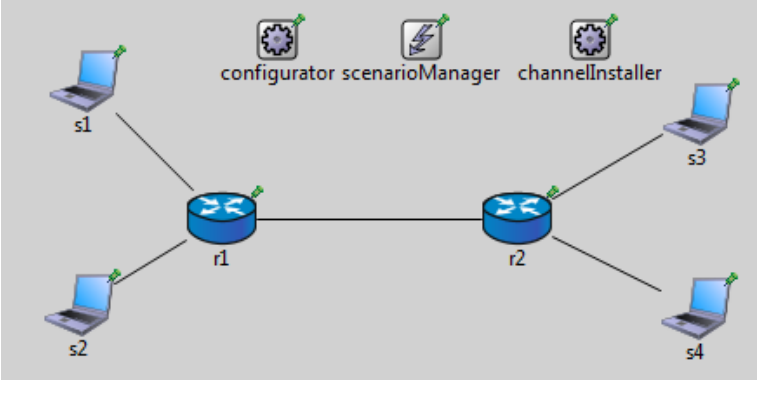


Fig. 3. The simulation network's topology

The link between r1 and r2 routers was the bootleneck of the network (its throughput had values of 100 kbps, 1 Mbps, 10 Mbps, 100 Mbps). The rest of links was 100 Mbps. The evaluated queue was the output queue for the r1 router. We proposed the following modification of the router. Buffer in the router consisted

of two queues, one for the TCP stream and one for UDP stream. The simulations were perfomed for different algorithms of the packet scheduling (PRIO, RR – round robin , weighted RR) and different algorithms of queue menagement (RED, FIFO). The simulations were perfomed for TCP connections only and for TCP operating together with UDP. The connection between hosts s1 and s3 was a TCP connection (TCP Reno). The UDP connection between hosts s2 and s4 corresponded to a simple video frames transmission. The general queue parameters were: DropQueue size = 25, RED (both cases): wq = 0.02, minth = 15, maxth = 25, maxp = 0.03 [32,33,34]. The simulated time was 200 [s].

4 Numerical Results

In this section we present chosen results achieved by simulation.

4.1 One RED Queue for TCP and UDP Streams

During the first experiment the both data streams (TCP and UDP) are placed in a one RED queue.

Distribution of the moving average queue length is presented in Fig. 4. It is visible that the queue was completely unstable. TCP parameters (RTT:0.24 (see Table 1), great number of unacknowledged segments, long data transmission time) indicate starvation TCP over UDP.

Additionally observing TCP traffic parameters (average: 0.24 RTT, large number of unacknowledged segments, long data transmission time) we can indicate that UDP stream fully appropriated link.

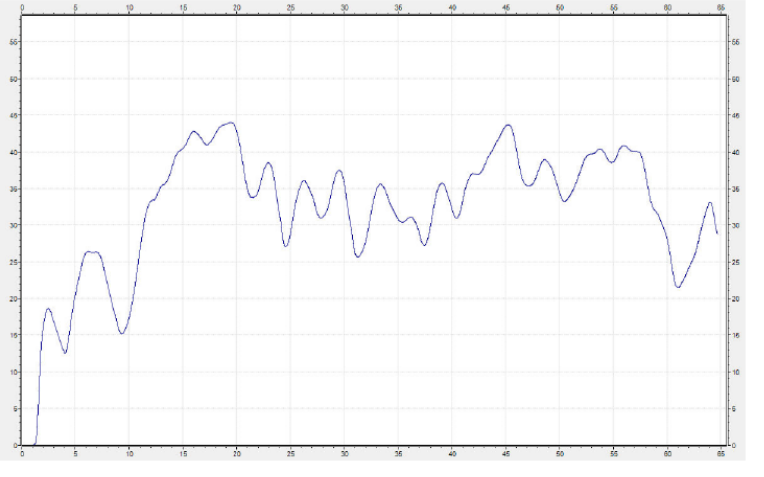


Fig. 4. Moving average queue length as a function of time – RED queue

Table 1. Round-trip delay time

Cases	QUEUE1		QUEUE2	
	type	Mean size (std dev)	type	Mean size (std dev)
One queue	3DropTail	36.7 (12.3)	-	-
One queue	RED (TailDrop)	32 (7.9)	-	-
Priority TCP	RED (TailDrop)	11 (3.2)	FIFO	960 (570)
Priority TCP	FIFO	69 (30)	FIFO	986 (555)
SFQ (4UDP/1TCP)	FIFO	111 (38)	FIFO	0.8 (0.68)
SFQ (1UDP/1TCP)	FIFO	90.9 (34)	FIFO	960 (563)
SFQ (1 UDP/1TCP)	RED (TailDrop)	6.9 (2.0)	FIFO	959 (562)
Priority UDP	RED (TailDrop)	7.7 (2.3)	FIFO	0.80 (0.68)

Cases	TCP 1		TCP 2		TCP 3	
	RTT	Smoothed RTT	RTT	Smoothed RTT	RTT	Smoothed RTT type
One queue	0.23	0.24	0.24	0.25	0.22	0.2
One queue	0.24	0.22	0.24	0.23	0.23	0.24
Priority TCP	0.12	0.09	0.12	0.11	0.11	0.12
Priority TCP	0.47	0.22	0.27	0.17	0.40	0.34
SFQ (4UDP/1TCP)	1.3	0.7	2.4	1.7	2.4	1.7
SFQ (1UDP/1TCP)	0.95	0.29	0.94	0.69	0.47	0.59
SFQ (1 UDP/1TCP)	0.14	0.13	0.13	0.12	0.16	0.15
Priority UDP	0.29	0.27	0.26	0.25	0.28	0.28

4.2 Two Queues for TCP and UDP Streams with PRIO Scheduling

In the next phase of the experiment, we tried to increase the chances of the TCP stream. In the router r1 we created a double queue with PRIO scheduling. The first queue for the TCP segments and second for the UDP datagrams. UDP packets were sent when the first queue is empty. The queue for the TCP has implemented the RED mechanism. The UDP queue was FIFO. The results of the experiment are shown in Fig. 5.

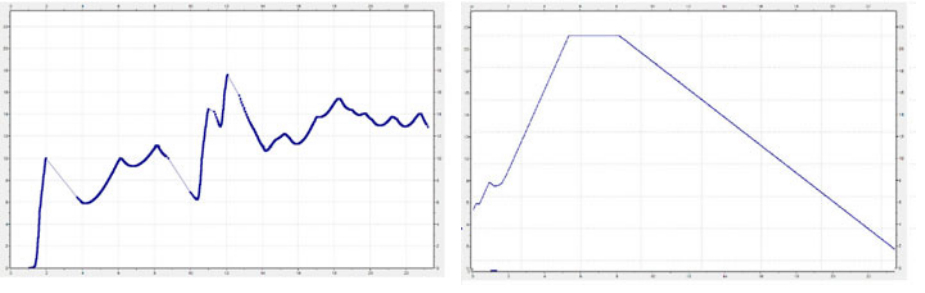


Fig. 5. Real queue length as a function of time – TCP RED queue (left), UDP FIFO queue (right) PRIO scheduling

This case is very comfortable for TCP. RTT depends on the particular transmissions ranges from 0.09 to 0.12. However, looking at the UDP queue, we see

that the UDP broadcast is being held in queue until the completion of the TCP transmission. Therefore, in the next simulation we modified the algorithm PRIO. UDP packets were sent when the first queue was empty or when the RED alghoritm dropped packet from the first queue. The obtained results were practically the same. The reason of this situation was too small number of packets dropped by the RED algorithm.

Figure 6 shows a totally opposite situation. The UDP traffic was redirected to the first queue. Contrary to expectations, this situation has proved to be the best. UDP broadcast video transmission was sent without problems. TCP transmission speed is very satisfactory. The obtained results were caused by very specific distribution of packets in a stream of UDP.

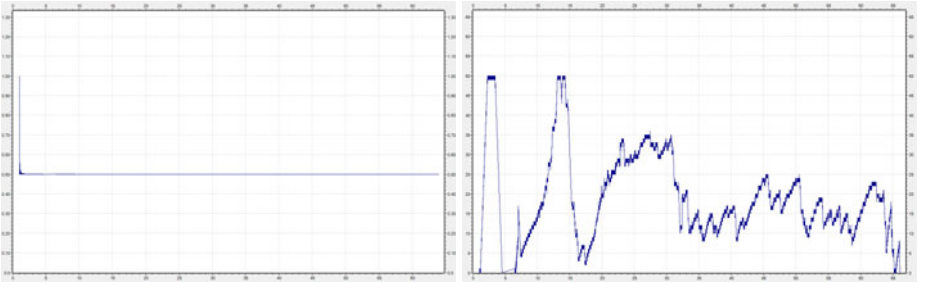


Fig. 6. Real queue length as a function of time – UDP RED queue (left), TCP RED queue (right) PRIO with modification scheduling

4.3 Two Queues for TCP and UDP Streams with RR – Round Robin Scheduling

It is clear that the PRIO algorithm is suitable only in very specific network conditions. Therefore, in subsequent simulations, we used SFQ (Stochastic Fairness Queueuing) algorithm – one packet from the first queue, one packet from the second. This case also is not favorable, because the UDP need more bandwidth for trouble-free transmission (about three-quarters), and this case gave him only half (Fig. 7). This simulation showed that we can exactly predict what the band was needed for smooth motion video broadcast.

For this reason, the last two simulations used the weighted Round Robin schedule. Dequeueuing mechanism fetched one packet from the TCP queue and four from the UDP queue. Results for both the FIFO queues are shown in Fig. 8. Situation for the RED queue for the TCP flow is shown in Fig. 9.

Both cases were very good for the UDP stream. The RED mechanism for TCP queue guarantees the preservation of the suitable value of RTT (RTT average of 0.25 to 0.29 [s]).

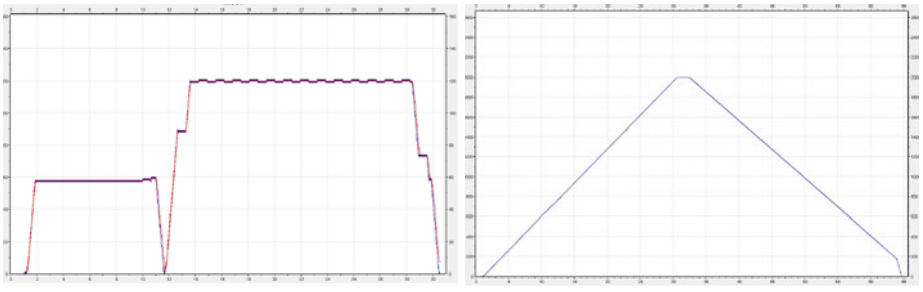


Fig. 7. Real queue length as a function of time – TCP FIFO queue (left), UDP FIFO queue (right) RR scheduling

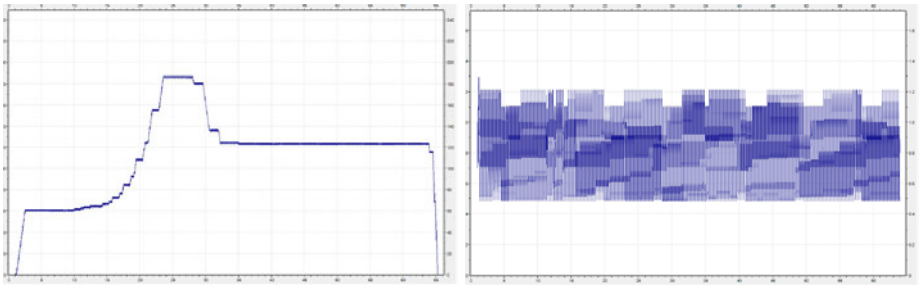


Fig. 8. Real queue length as a function of time – TCP FIFO queue (left), UDP FIFO queue (right) WRR scheduling 4*1 for UDP

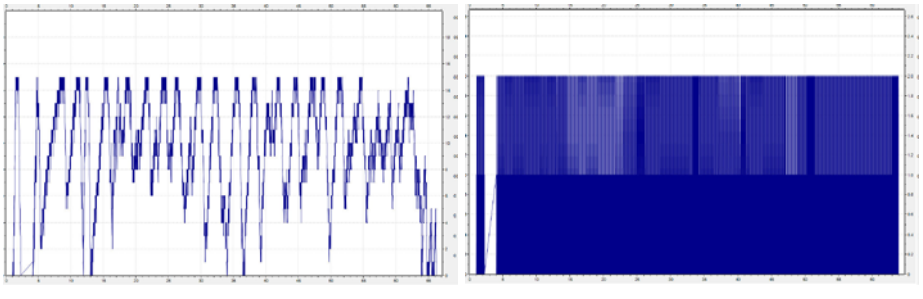


Fig. 9. Real queue length as a function of time – TCP RED queue (left), UDP FIFO queue (right) WRR scheduling 4*1 for UDP

5 Conclusions

In this article we investigate the problem of unfair distribution of bandwidth between the TCP and UDP streams. This problem is growing, where transmission takes place through the nodes, with AQM mechanisms. During the tests

we tried to answer the question whether it is possible to create such conditions in the queue for both types of data streams to be treated fairly. Our research was carried out in the environment of new Omnet++ (in version 4.0) simulation framework extended with the INET package.

During the tests we analyzed the following parameters of the transmission with AQM: the length of the queue, the number of rejected packets, transmission time and the RTT parameter. Classical RED algorithms are suitable only for TCP protocols with congestion algorithms. For UDP traffic the examined parameters of the transmission with AQM are significantly worse. Moreover, UDP transfers may in this case practically eliminate the TCP transmissions. Performed experiments have shown also that it is possible, using well-known queuing mechanisms, to ensure fair treatment for both TCP and UDP streams. The best solution would be here an automatic selection of the parameters of queuing algorithms, depending on the statistical data collected during normal operation of the router. Our research has shown that this is possible but very difficult.

References

1. Hashem, E.: Analysis of random drop for gateway congestion control, http://www.worldcatlibraries.org/oclc/61689324
2. Feng, W.-C., Kandlur, D.D., Saha, D., Shin, K.G.: A Self-Configuring RED Gateway, http://citeseer.ist.psu.edu/470052.html
3. Nowak, S., Domańska, J., Domański, A.: Performance modeling of selected AQM mechanisms in TCP/IP network. In: IWSI (2009)
4. Domańska, J., Domański, A., Czachórski, T.: Implementation of modified AQM mechanisms in IP routers. Journal of Communications Software and Systems 4(1) (March 2008)
5. Domańska, J., Domański, A.: Active Queue Management in Linux based routers. In: IWSI (2008)
6. Postel, J., et al. (eds.): Transmission Control Protocol. Information Sciences Institute, University of Southern California, Internet RFC 0793 (September 1981), http://www.faqs.org/rfcs/rfc793.html
7. Kapadia, A., Feng, W., Campbell, R.H.: GREEN: A TCP Equation Based Approach to Active Queue Management, http://www.cs.dartmouth.edu/~akapadia/papers/UIUCDCS-R-2004-2408.pdf
8. Athuraliya, S., Li, V.H., Low, S.H., Yin, Q.: REM: Active Queue Management, http://netlab.caltech.edu/FAST/papers/cbef.pdf
9. Feng, W., Kandlur, D., Saha, D., Shin, K.: Blue: A New Class of Active Queue Management Algorithms, http://citeseer.ist.psu.edu/feng99blue.html
10. Srisankar, Kunniyur, S., Srikant, R.: An Adaptive Virtual Queue (AVQ) Algorithm for Active Queue Management, http://comm.csl.uiuc.edu/srikant/Papers/avq.pdf
11. Lin, D., Morris, R.: Dynamics of Random Early Detection, https://pdos.csail.mit.edu/~rtm/papers/fred.pdf
12. Alemu, T., Jean-Marie, A.: Dynamic Configuration of RED Parameters, http://citeseer.ist.psu.edu/728472.html
13. Brachman, A.: Wplyw mechanizmow kontroli ruchu na jakosc uslug w sieciach bezprzewodowych. Phd Thessis, Gliwice (2008)

14. Pan, R., Prabhakar, B., Psounis, K.: CHOKe. A stateless active queue management scheme for approximating fair bandwidth allocation. IEEE Infocom (2000)
15. Ningning, H., Liu, R.: Jichuan: Evaluation of Queue Management Algorithms. Course Project Report for 15-744 Computer Networks
16. Jiantao, W., Ao, T., Steven, L.H.: Maximum and Asymptotic UDP Throughput under CHOKe. In: ACM SIGMETRICS Performance Evaluation Review (2003)
17. Floyd, S., Handley, M., Kohler, E.: Problem Statement for the Datagram Congestion Control Protocol (DCCP). Internet RFC 4336 (March 2006)
18. Floyd, S., Handley, M., Padhye, J., Widmer, J.: EquationBased Congestion Control for Unicast Applications. In: SIGCOMM (May 2000)
19. Burakowski, W., Tarasiuk, H., Krawiec, P.: Analiza algorytmow i mechanizmow sterowania ruchem na poziomie pakietow w sieci IP QoS. Polska (2008)
20. Hassan, M., Jain, R.: High Performance TCP/IP Networking. Pearson Education Inc., London (2004)
21. Stevens, W.R.: TCP Slow Start, Congestion Avoidance, Fast Retransmit and Fast Recovery Algorithm. In: Internet RFC 2001 (January 1997)
22. Lee, H., Lee, S.H., Choi, Y.: The Influence of the Large Bandwidth-Delay Product on TCP Reno, NewReno and SACK, Japan (2001)
23. Low, S.: Equilibrium and Dynamics of TCP/AQM (March 2002), http://netlab.caltech.edu
24. Floyd, S., Jacobson, V.: Random Early Detection gateways for Congestion Avoidance, http://www.cs.ucsd.edu/classes/wi01/cse222/papers/floyd-red-ton93.pdf
25. Random Early Detection (RED): Algorithm, Modeling and Parameters Configuration, http://photon.poly.edu/~jefftao/JTao_RED_report.pdf
26. Floyd, S., Gummadi, R., Shenker, S.: Adaptive RED: An Algorithm for Increasing the Robustness of REDs Active Queue Management, http://citeseer.ist.psu.edu/448749.html
27. Domańska, J., Domański, A., Czachórski, T.: The Drop-From-Front Strategy in AQM. In: Koucheryavy, Y., Harju, J., Sayenko, A. (eds.) NEW2AN 2007. LNCS, vol. 4712, pp. 61–72. Springer, Heidelberg (2007)
28. Yang, X., Chen, H., Lang, S.: Estimation Method of Maximum Discard Probability in RED Parameters, http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1712588
29. OMNET++ homepage, http://www.omnetpp.org/
30. Domanska, J., Grochla, K., Nowak, S.: Symulator zdarzeń dyskretnych OMNeT++. Wyd. In: Wyzsza Szkola Biznesu w Dabrowie Górniczej, Dabrowa Górnicza (2009)
31. INET homepage, http://inet.omnetpp.org/
32. Floyd, S.: Discussions of setting parameters (1997), http://www.icir.org/floyd/REDparameters.txt
33. Zheng, B., Atiquzzaman, M.: A framework to determine the optimal weight parameter of red in next generation internet routers. The University of Dayton, Department of Electrical and Computer Engineering, Tech. Rep. (2000)
34. May, M., Bonald, T., Bolot, J.: Analytic evaluation of red performance. In: IEEE Infocom 2000, Tel-Aviv, Izrael (2000)

Mobile Console for a Server of MBean Components

Krzysztof Dobosz

Institute of Informatics, Silesian University of Technology, Gliwice
krzysztof.dobosz@polsl.pl

Abstract. Author presents an idea of remote management with applications using mobile devices. Proposed architecture consists of: applications run in the Java Virtual Machine environment and use JMX technology for representing resources, mobile clients run on Java ME platform, and a proxy server. The access to JMX mechanisms requires an implementation of RMI protocol. Unfortunately the Java ME Platform does not define proper API. That is why we need a proxy server representing JMX services for non-RMI mobile clients. The role of the proxy server is two-directional translation between text descriptions of MBeans and remote method invocations. Advantages of proposed solution: easy extensibility and platform independence.

Keywords: JMX, MBean, mobile console, Java ME.

1 Introduction

The Java Management Extensions (JMX) technology provides the tools for building distributed and modular solutions for managing and monitoring devices, applications, services, and the Java virtual machine [1,2,3]. That is defining software components called managed beans (MBeans) to the representation of every resource. The JMX API is a standard application interface introduced in the Java Standard Edition (Java SE) Platform [4]. The JMX technology was developed by the Java Community Process (JCP) as Java Specification Request (JSR) 3 [5] and 160 [6]. The JMX offers three ways of controlling remote resources [7]:

- monitoring and changing objects attributes,
- calling objects methods,
- receiving notifications.

These mechanisms implicate following typical uses of the JMX technology:

- consulting and changing application configuration,
- accumulating statistics about application behavior and making them available,
- notifying of state changes and erroneous conditions.

The JMX technology simplifies a design of remotely managed systems. Very important is a fact, that the mechanism of the server of MBean components is

A. Kwiecień, P. Gaj, and P. Stera (Eds.): CN 2010, CCIS 79, pp. 217–225, 2010.

already an integral part of the implementation of the JVM virtual machine [4,8]. In spite of the growing JMX popularity, still are lacking of application for mobile phones equipped with the implementation of Java Micro Edition Platform, which with functionality would equal client applications for personal computers. An original approach for the remote monitoring MBeans made according to the JMX technology from the level of the mobile device, was a purpose of the research.

2 Existing Solutions

The JMX technology is widely applied in different projects written in Java. Some of them:

- JRemoteRun [9] – the MBean installed on the JMX server. It enables to start any program.
- Eclipse-jmx [10] – plug-in for the Eclipse environment. This is a client of a JMX server, which manages with applications running on any JMX Server.
- JMXView [11] – HTML adapter. It can be installed on the Server and use web browser In the role of a JMX client.
- JMeasurement [12] – tool measuring the efficiency of chosen parts of code. Results of monitoring can be downloaded from the server by using MBeans.
- JAFS (Just Another FTP Server) [13] – the FTP server, which can be managed with a set of MBeans.
- Global Discovery Service for JMX Architecture [14] – the project introducing of agent oriented design for establishing connections and resources discovery.
- MobiMon [15] – so far the only solution for Java ME Platform. That mobile application connects to a JMX server at using HTTP protocol. The role of a mediating element plays a web component made in the Java Servlets technology.

3 A New Approach

In this section the new approach to the JMX is described. This uses a proxy server as a middle layer between a mobile client and a JMX server. Apart from three-layered architecture, application interface (API) for mobile devices is also offered.

3.1 Architecture

A popular mobile phone has the processor clock with clock 400 MHz and 1 MB of the RAM memory. These are little sources in relation to personal computers. Therefore virtual machine of the Java meant for mobile phones had to be simplified. Defining the Java ME Platform a set of classes and the functions was limited. In the case of personal computers, data transmitted between a client application and the JMX server is serialized. The transmission uses RMI protocol, which is a base protocol for the JMX. The Java ME Platform does not specify these three elements: data serialization, RMI, JMX. Theoretically is possible to

prepare own implementations of indicated elements, but in the reality they will be too complex for weak mobile devices. The idea is to introduce a proxy server mediating in the communication between the mobile client and the JMX server (Fig. 1). Mobile application called MConsole, connects to the proxy server using TCP/IP protocol and text-based commands. The proxy server connecting to the JMX server uses standard JMX mechanisms.

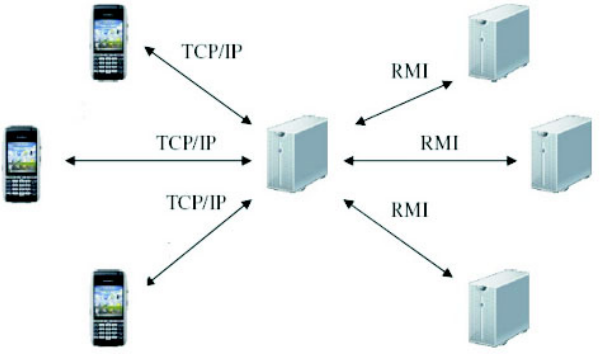


Fig. 1. The architecture consists of mobile clients, JMX servers and the proxy server in the center

3.2 Functional and Non-functional Requirements

Preparing MConsole application in the analysis stage, should be defined functional and non-functional requirements. Among functional requirements it is possible to point:

- establishing and closing connection to the proxy server,
- establishing and closing connection, and managing the any JMX server without any additional configuration of the proxy server and a JMX server,
- storing the list of JMX servers introduced by the user,
- downloading and looking through the list of MBeans located on a JMX server,
- acquiring information about chosen MBean,
- reading and changing objects attributes,
- remote operations executing,
- notifications handling.

In every project we can describe also criteria referring to technical aspects. They are described by non-functional requirements:

- optimization of data transferred between the MConsole and the proxy server,
- safe data transferring,
- easy to use user interface.

3.3 MConsole API

A part of the project realizing communication to the proxy server is made in a form of separated library with clearly defined specification. Main package `mconsole.jmx` includes:

- `NotificationListener` – interface implemented for notification handling,
- `ArraysUtil` – several static methods for arrays processing,
- `ServerConfiguration` – contains configuration of server connection (address, port, login, password).

Apart from that, the application interface consists of two sub-packages dedicated to communication: `mconsole.jmx.io` and `mconsole.jmx.dto`. The package `mconsole.jmx.io` includes API responsible for the connection to the proxy server:

- `Serializable` – interface implemented in classes, which instances are transmitted to the proxy server,
- `Serializable` – interface implemented in classes, which instances are transmitted to the proxy server,
- `ProxyConnection` – represents connection to the proxy server, that includes method for manipulation with MBeans,
- `ProxyConnector` – establishes connection to the proxy server,
- `ResponseTypes` – contains types of reply of the proxy server after processing a request,
- `SecureSerializer` – operates with serialized objects,
- `Serializer` – cooperates with objects of the `Serializable` type, corresponds for writing and reading them out to the value from the data stream,
- `ProxyConnectionException` – represents a mistake of the connection with the proxy server,
- `ProxyTimeoutException` – represents an exception of the time out for the proxy server response,
- `RemoteException` – represents exceptions, which are thrown on the JMX Server side.

The package `mconsole.jmx.dto` includes classes, which serialized instances are sent through the Internet. This part of MConsole API includes:

- `Value` – the interface implemented by classes of attributes, which are sent to the proxy server,
- `Attribute` – MBean attribute,
- `AttributeInfo` – metadata of a MBean attribute,
- `BeanInfo` – information about MBean,
- `BeansTreeNode` – node of a tree structure with information about MBeans,
- `BigDecimal` – immutable, arbitrary-precision signed decimal numbers,
- `BigInteger` – immutable arbitrary-precision integers,
- `CompositeData` – contains an array of objects,
- `ConstructorInfo` – metadata of a MBean constructor,
- `Descriptor` – contains pairs key-value describing elements of MBean (attribute, operation or notification),

- `FeatureInfo` – base class for classes of metadata: `AttributeInfo`, `ConstructorInfo`, `NotificationInfo`, `OperationInfo`,
- `Notification` – represents notification generated by the JMX server after event,
- `NotificationInfo` – metadata of a notification,
- `NotSupportedValue` – represent the value not supported by the JMX server,
- `ObjectName` – MBean name,
- `OperationInfo` – metadata of an operation,
- `ParameterInfo` – metadata of operation parameters,
- `SimpleValue` – class enclosing objects from `java.lang package`,
- `TabularData` – includes array of objects of `CompositeData` type.

4 The Role of the Proxy Server

The main task of the proxy server is a two-directional translation between text descriptions of MBeans sending via TCP/IP and remote method invocations. Apart from that the proxy server also forwards JMX server's notifications to the mobile client.

4.1 Data Mapping

An engine of the proxy server uses a standard `SocketServer` class in the implementation. It is responsible for starting the server and for the service of requests for clients. A service part of the proxy server waits for clients connections and creates a new service thread for every one. In the thread the exchange of keys encoding between the client and the server is realized. Next, the clue of communication is performed – the server gets operation identifier and the list of its arguments, looks for them in `MBeansProxy` class, calls it, and gets a result back. The connection is still opened after the first response, and the proxy server still waits for next requests of the same client. The mobile client initiates of testing attributes and methods invocations. Generating notifications is the only action initiated by the JMX server. Notifications are the result of events occurred on the JMX server. They usually generated independently of the state of the mobile client. Their idea is very similar to triggers in databases. In the case of notification, the JMX server sends information about an event to the proxy server. The proxy server forwards it to the mobile client via opened TCP connection (this is not possible at use HTTP protocol). Waiting to be sent, it is being checked, whether in the given moment he isn't undergoing other communication with the same client. That is realized with a mechanism of threads synchronization implemented in JVM virtual machine.

4.2 Data Serialization

The communication between mobile application and the proxy server uses the `ProxyConenction` class. Every request sent to the proxy server consists of function name and list of parameters. The server after receiving a request, uses a

mechanisms of reflection for searching functions in the `MBeansProxy` class. Found functions are being invocated with received parameters. The result of the function is sent back to the mobile client. Such a communication model enables the simple development of the server by new functions. Adding the new method to the `MBeansProxy` class doesn't require introduction of changes to the protocol of the communication, and no additional configuration. At once after giving the new function on the JMX server the mobile client can use it. All data transmitted between the mobile application and the proxy server is sent in the form of serialized objects. A `Serializer` class is responsible for a serialization of these objects. This class allows to transform following types to the byte stream:

- all primitive types (`byte`, `short`, `int`, ...),
- classes wrap a values of primitive types in objects (Byte, Short, Integer, ...),
- `Vector`, `Stack`, `String`, `Date`, `Hashtable`,
- classes implementing `Serializable` interface,
- one-dimensional arrays of above types.

Definitions of remaining classes for data transmission, which can be serialized, are placed in the `mconsole.jmx.dto` package. All of them implements `Serializable` interface. This interface consists of two methods: write(`DataOutputStream`) and read(`DataInputStream`). If sent (or received) is an object of the class, which does not implement of `Serializable` interface, then writing (or reading) to the stream is realized in the `Serializer` class. In other case an identifier of a class implementing `Serializable` interface is write into the stream. Next `write()` method is called. Method `write()` is responsible for a storing of values of object attributes to the stream. The `Serializer` object also reads data from the stream. If object implementing `Seralizable` is received, the `Serializer` object reads its class identifier from the stream, makes an instance of this class invocating a non-argument constructor, and finally calls method `read()`. This method reads values of object attributes up one by one. The MConsole API also includes `SecureSerializer` class, which uses the `Serializer` class and additionally realizes encryption of transmitted data.

4.3 Creating the List of MBeans for the Mobile Client

Collecting a list of MBeans from the server, their names and all attributes with values are being taken. Sending of the complete information with every value isn't necessary. The proxy server for data from the JMX server, uses a structure of objects of the `BeansTreeNode` class. This structure keeps information of MBeans in a form of tree, where every node represents one pair: key/value. The node also consists of references to other nodes. The full sequential list of data is made by recursively searching from top to bottom of the tree. The result is a list being created according to the following schema:

```
MBean1:key1=value1,...,keyN=valueN.MBean2:key1=value1,...
```

This way we can optimize a set of data transmitted through the net. In the case of the JBoss server, we have over 500 MBeans [16]. The full list of data takes 46 409 bytes. Using proposed method it can be reduced to 32 969 bytes.

5 Security

Managed Beans can represent very important resources, so transmitted data between JMX server and proxy server, and between proxy server and mobile application should be encrypted for increasing the safety.

5.1 Data Encryption between the Proxy Server and JMX Server

In the case of connection between the proxy server and the JMX server can be used mechanisms supported by JMX technology, which means the SSL protocol. The SSL protocol requires holding the certificate of the server in a store of certificates. If the server allows clients to public access, then a trusted supplier should deliver a certificate. However in some cases a self-signed certificate is sufficient. Java Development Kit delivers a proper tool: `keytool`. Sample use:

```
keytool -genkey -alias jmxServer -keyalg RSA
-keystore jmx.keystore
```

where: `jmxServer` – certificate name, `jmx.keystore` – certificates store

The certificate of the server should be exported to the file:

```
keytool -export -alias jmxServer -keystore jmx.keystore
-file jmxServer.cer
```

where: `jmxServer.cer` – file name.

Next a store of certificates for a client (in our case: the proxy server) should be created, and certificate of the server can be imported:

```
keytool -genkey -alias jmxClient -keyalg RSA
-keystore jmx.truststore
keytool -import -file jmxServer.cer
-keystore jmx.truststore
```

The method of an encryption is dependent on the used application server. For example in order to include this function for the JBoss server, following lines should be added to the script starting the server:

```
set JAVA_OPTS=%JAVA_OPTS%
-Dcom.sun.management.jmxremote.ssl=true
set JAVA_OPTS=%JAVA_OPTS%
-Djavax.net.ssl.keyStore=../conf/jmx.keystore
set JAVA_OPTS=%JAVA_OPTS%
-Djavax.net.ssl.keyStorePassword=changeit
```

Starting the client application with proper parameters is the last step. That will allow gaining access to the certificate of the server:

```
client -J -Djavax.net.ssl.trustStore=jmx.truststore
       -J -Djavax.net.ssl.trustStorePassword=trustword
```

5.2 Data Encryption between the Proxy Server and JMX Server

In this case also SSL protocol can be used. However all steps from previous section cannot be repeated. One of steps is importing certificate of the server into the store. Most producers of mobile devices, do not implement such functionality in their products. They implement read-only store contains several standard certificates, and programmers has no opportunity to add next. The SSL protocol can be replaced by external library for data encryption, i.e. Bouncy Castle Crypto Package [17]. Establishing of a connection to the server is similarly as in the case of the SSL protocol. Further steps of the algorithm establishing the encoded connection:

1. The client is beginning the connection to the server.
2. On the initial stage, data is encrypted by asynchronous algorithm. The server creates a pair of keys (public and private), and next sends the public key to the client.
3. The client generates the key used for synchronous encoding, then data is encoded with the public key obtained from the server and it is sent to the server.
4. The server using its private key is decoding the key received from the client.
5. From this moment all data is encoded by a synchronous algorithm.

The above method is combining advantages of asynchronous (better safety, greater resistance to attempts to break a code) and synchronous (short time) encoding. For asynchronous encrypting can be used the RSA algorithm with 1024-bit key, and for synchronous encrypting: DES algorithm with 64-bit key.

6 Implementation

The architecture described in the section was implemented in the Java language [18]. The mobile client consists of two parts: implementation of MConsole API and interactive graphical user interface. This reference implementation of the application interface has a form of a library, and can be an add-ins for other applications. The size of the first version was about 1 MB. It was so large because of including Bouncy Castle Crypto Package for encryption. Additionally this encoding library uses some classes from Java SE Platform, which are not the part of Java ME Platform. They are i.e. `java.math.BigInteger` or `java.util.StringTokenizer`. These classes are also implemented in Bouncy Castle Crypto Package for mobile devices. The size of the whole mobile application was too large, so it was made smaller. Using the lexicographical method of size optimization [19] and removing unused classes, the size of the implementation of the mobile client was reduced to 300 KB.

7 Conclusions

In the section is presented the new approach to the mobile access to the JMX server. Advantages of proposed solution: easy extensibility – software for mobile device

should have a form of a library implementing proposed API, platform independence – proxy server accepts text commands. Certainly a better idea is a two-tier architecture. However, it would require of enforcing the RMI protocol as the standard for the Java ME Platform. Another idea is introduction of direct access to MBeans by using TCP protocol. Such a solutions are still developing, but they are not common, because require an improvement of the standard JMX server.

References

1. Perry, J.S.: Java Management Extensions. O'Reilly, Sebastopol (2002)
2. Starosta, B.: JMX. Zarządzanie aplikacjami w języku Java. PJWSTK, Warszawa (2006)
3. Sullins, B., Whipple, M.: JMX in Action. Manning Publication Co. (2002)
4. Java Management Extensions (JMX) Technology (2009), http://java.sun.com/javase/technologies/core/mntr-mgmt/javamanagement
5. JSR 3: Java Management Extensions (JMXTM) Specification (2009), http://jcp.org/en/jsr/detail?id=3
6. JSR 160: JavaTM Management Extensions (JMX) Remote API (2009), http://jcp.org/en/jsr/detail?id=160
7. Kreger, H., Harold, W., Williamson, L.: Java and JMX: Building Manageable Systems. Addison-Wesley, Reading (2002)
8. JMX Programming. Mike Jasnowski Publisher: Wiley, John and Sons, Incorporated (2002)
9. JRemoteRun (2009), http://jremoterun.sourceforge.net/
10. Eclipse-jmx (2009), http://code.google.com/p/eclipse-jmx/
11. JMXView (2009), http://jmxview.sourceforge.net/
12. JMeasurement, http://sourceforge.net/projects/jmeasurement2/
13. JAFS (2009), http://www.sbbi.net/site/jafs/index.html
14. Midura, J., Balos, K., Zielinski, K.: Global Discovery Service for JMX Architecture. In: Bubak, M., van Albada, G.D., Sloot, P.M.A., Dongarra, J. (eds.) ICCS 2004. LNCS, vol. 3038, p. 114. Springer, Heidelberg (2004)
15. MobiMon (2009), http://mobimon.sourceforge.net
16. Fleury, M., Lindfors, J.: JMX: Managing J2EE with Java Management Extensions. Sams Publishing, USA (2002)
17. Bouncy Castle Crypto Package (2009), http://www.bouncycastle.org/
18. Klima, K.: Mobile console remotely monitoring mbeans running on the JVM platforms. M.Sc. thesis. Silesian University of Technology, Faculty of Automatic Control, Electronics and Computer Science, Institute of Informatics (2009)
19. Dobosz, K., Wesołowski, T.: Applying tools for byte-code obfuscation in the process of the memory optimization. Studia Informatica 30(3A), 47–57 (2009)

Using Cloud Storage in Production Monitoring Systems

Arkadiusz Jestratjew and Andrzej Kwiecień

Silesian University of Technology, Institute of Informatics,
Akademicka 16, 44-101 Gliwice, Poland
{arkadiusz.jestratjew,andrzej.kwiecien}@polsl.pl

Abstract. *Cloud Computing* has received much rumour during the last year. Although the idea of lending processing power and data storage is, in essence, known since the very beginning of the computer industry, a new distributed computing platform coming from Microsoft is expected to have much impact on the enterprise software development and usage.

However, production monitoring systems are traditionally built with on-site data stores and specialized hardware and software solutions that interconnects data sources (process controllers) and data stores. We propose an alternative approach. Using modern PLC solutions, one is able to avoid on-site data store in favor of *Cloud Storage* available over the Internet, lowering capital and maintenance expenses.

Keywords: availability, cloud computing, database, PLC, REST.

1 Introduction

In a global economy and strong competition world, enterprises are encouraged to provide its customers with low cost and high quality goods. In order to achieve that goal, production processes are continuously monitored for any deviations of process parameters. Monitoring systems help in detecting problems early before they arise and provide opportunity for applying proper preventive or correction actions.

Classic production process monitoring system (Fig. 1) consists of the following main components: a *data source* (i.e. process controller), an on-site *data store* interconnected by *intermediary communication infrastructure* built with specialized hardware and software solutions.

2 Production Monitoring System Components

2.1 Data Source

Data source is a programmable device performing data acquisition in process monitoring system. A data source must execute program with real-time constraints to assure that no process data is lost and typically provide some kind of data buffering features, improving reliability of the whole monitoring system.

A. Kwiecień, P. Gaj, and P. Stera (Eds.): CN 2010, CCIS 79, pp. 226–235, 2010.

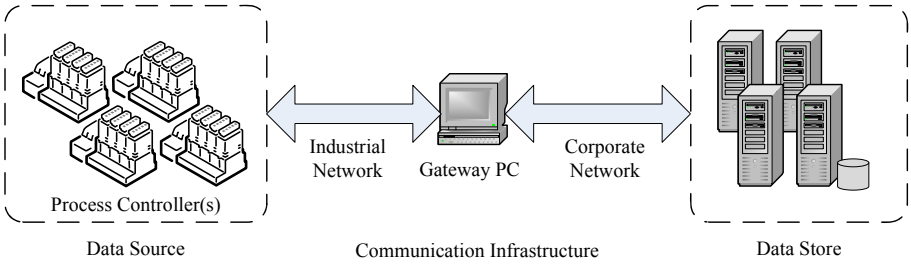


Fig. 1. Architecture of an industrial process monitoring system

Process controllers, for example *Programmable Logic Controllers* (PLC, e.g. [1]), *Distributed Control Systems* (DCS, e.g. [2]) or *Programmable Automation Controllers* (PAC, e.g. [3]), are natural data sources for monitoring systems, as all process data flows by them. It is also possible that separate devices perform acquisition of process data and actual process control. In such case, the device that performs data acquisition function is considered a data source.

2.2 Communication Infrastructure

Communication infrastructure in typical monitoring system consists of the following three subcomponents.

1. Industrial network with dedicated communication protocol and possibly dedicated physical layer hardware.
2. Gateway device equipped with necessary hardware and software that executes protocol stacks and perform data conversion.
3. Corporate network that provides access to the data store, typically using a fast Ethernet connection.

Many modern process controllers are equipped with Ethernet interface, simplifying development of industrial monitoring systems, as dedicated physical layer hardware is no longer necessary. In such environment, the gateway is used mainly as application layer protocol converter, and could be avoided if a data source was able to handle data store communication protocol, as described in paper [4].

The data store may be considered as a black box accessible by either dedicated binary protocols (e.g. TDS [5], Oracle SQL*Net, etc.) or HTTP-based protocols (SOAP [6] or REST [7,8]). However, it is required that implementing custom protocols in application software is supported by the firmware of the process controller. Examples of such devices are Saia-Burgess PCD series programmable logic controllers [9].

2.3 Data Store

Data store is a part of the back-end corporate IT infrastructure, located in data processing centers. Nowadays, multiple products and technologies for data processing centers are commercially available, including database server clustering,

connection brokers, load balancers, application server farms and hardware redundancy, to name just a few of them. These technologies together with physical security, controlled environmental conditions and valid operation procedures allow building back-end systems that fulfill corporate requirements of availability, scalability and performance [10,11].

In context of industrial monitoring systems, it shall be considered that building a data store with required availability and processing power is more a cost-to-benefit problem than engineering problem. Enterprise that is building its own data processing center faces high costs of ownership, both costs of initial hardware and software deployment and costs of operation and maintenance. For small-to-medium sized companies such costs can be a barrier that makes implementation of production monitoring systems economically unfeasible. Lowering such barriers is possible by using hosting services provided by third parties.

3 Software as a Service

The *Software as a Service* (SaaS) term is used to describe software delivery model, where users pay fee for actually using the application rather than for rights (license) to use it [12]. It is often assumed that the application is hosted in the vendor's data center rather than the user's own one, turning capital expenses of building the data center into operational expenses of using it. As applications are accessed remotely over the Internet, this model is also referred to as *Cloud Computing*.

Not all software vendors have to maintain its own data centers to deliver software as a service. Some, especially smaller size software vendors, may choose to use commercially available hosting services to deliver the software product at lower costs because of economy of scale effect. An example of such hosting services is Windows Azure, "a platform for running Windows applications and storing their data in the cloud" [13], i.e. on "a large number of machines, all located in Microsoft data centers and accessible via the Internet" [13]. These services are available with defined Service Level Agreements (SLAs) [14] at relatively low costs that makes it attractive even for small sized companies.

The SaaS model gain popularity in some enterprise software markets, like enterprise resource planning (ERP), customer relationship management (CRM), project management to name just a few. However, it is rather uncommon in software used on so called *factory floor level*, e.g. production control, monitoring and visualization software.

4 Windows Azure Platform

Microsoft Windows Azure platform is a collective name for a group of services that are available to software developers. The following short description of the platform is based on [13] and [15]. The platform currently consists of the following services.

1. Microsoft Windows Azure Compute.
2. Microsoft Windows Azure Storage.
3. Microsoft SQL Azure.
4. Microsoft Windows Azure platform AppFabric Access Control.
5. Microsoft Windows Azure platform AppFabric Service Bus.

4.1 Windows Azure Compute Service

Windows Azure Compute is a hosting service that provides Windows-based environment for running applications. It is designed to be highly scalable to handle a large number of simultaneous users. That is achieved by *scaling out* the application, i.e. running multiple instances of the application. Each instance is executed in its own virtual machine, created and managed automatically. Number of instances of the application can be changed as needed. Applications for the Compute service can be created either in native or managed (.NET Framework) code, using a variety of development tools and technologies available for Windows.

Two different instance types (roles) are available for developers: *Web role* instances and *Worker role* instances. Web role instance runs Internet Information Services 7 (IIS) that accepts incoming HTTP or HTTPS requests and passes them to the user's application, built with ASP.NET, Windows Communications Foundation or other technology supported by IIS. If multiple Web role instances are configured for the application, the platform balance requests across all instances. That means the application must be stateless – any client-specific state must be persisted using either Azure Storage or SQL Azure or included in the request. RESTful, stateless web services are best fitted there.

Worker role instances lack the IIS. Such instances are designed primarily to perform processing tasks, using data persisted by Azure Storage or SQL Azure services. There are no built-in limits, however. It is even possible that the application executed by the Worker role instance runs its own HTTP server and handles requests.

4.2 Windows Azure Storage Service

Windows Azure Storage is a data persistence service that provides several types of data containers: *blobs*, *tables* and *queues*. A blob (binary large object) contains application-specific binary data and can be huge – up to terabyte each. Blobs can also be used as storage for *Azure Drive*. Azure Drive implements NTFS file system atop the blob and allow the application to store files within it.

Azure Storage tables store data as a group of *entities* that contains named *properties*. Despite similar name, storage tables are not relational tables and do not enforce fixed schema. Instead, each entity can have its own set of properties with various names and data types. However, each entity must include the following properties: PartitionKey, RowKey (both are string values up to 1 KB in size) and Timestamp (a DateTime value). The developer controls values of PartitionKey and RowKey. The Storage service manages the value of Timestamp

to record the time an entity was last modified to provide optimistic concurrency [16]. An entity can be up to 1 MB in size.

PartitionKey value is used to scale-out data by distribute data across storage nodes. Entities having equal PartitionKey values are expected to be stored within the same node. RowKey value uniquely identifies the entity within a given partition. Concatenated values of PartitionKey and RowKey uniquely identify the entity within the table.

Azure Storage queues are designed for communication purposes rather that permanent data storage. A queue stores unlimited number of messages that can be up to 8 KB in size. Messages are appended to the end of the queue and retrieved from the front, however first in first out behavior is not guaranteed [17]. A primary function of queues is to provide a way for Web role instances to communicate asynchronously with Worker role instances and vice-versa.

Windows Azure Storage services are available with RESTful approach over HTTP or HTTPS protocol. These services can be used either by applications hosted within Azure Compute services or desktop applications accessing data over the Internet (Fig. 2). All request sent to Azure Storage services must be signed with HMAC-SHA256 algorithm with shared secret key [18].

The platform is responsible for data replication (all data is replicated three times) and distribution to achieve fault tolerance and keep high performance even for huge amounts of data.

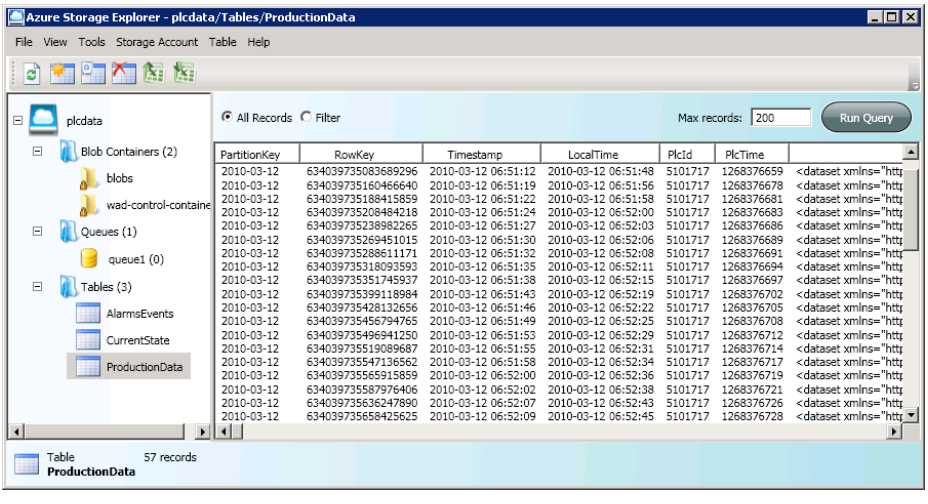


Fig. 2. Azure Storage account accessed by Windows application

4.3 SQL Azure Service

Microsoft SQL Azure is a relational database management system compatible with Microsoft SQL Server. SQL Azure databases can only be accessed with binary TDS [5] protocol, just as regular SQL Server databases, using any of

existing SQL Server client libraries including ADO.NET and ODBC. Standard programming tools, like Management Studio, Reporting Services, etc. can also be used. A single SQL Azure database is limited to 10 GB in size, however multiple databases can be used if necessary.

4.4 Azure Platform AppFabric Services

AppFabric services are provided to ease interconnections between applications running either on user's desktop or in the cloud. It shall be noted that despite similar name, AppFabric services are not related to the Azure Fabric. The Azure Fabric is an infrastructure component that manages virtual machine instances used by Azure Compute service.

AppFabric Access Control service provides a flexible mechanism to create, manage and transfer information about *application identity* using REST approach. Currently handling user identity information is not supported, however it will be in future. Client application, after successful authentication over encrypted HTTPS channel, receives a signed *token* that contains identity information about this client, expressed as set of *claims*, i.e. human readable name/value pairs. These claims are created by Access Control service with use a set of configurable rules.

AppFabric Service Bus allows for easily and safely exposing proprietary services running inside private networks to client software running outside the private network. This client software might be running either on a cloud platform or inside another organization. Service Bus acts as a well-known endpoint that both client and server software can bind to, allowing to communicate with proprietary services without the need for exposing them directly to the Internet, having fixed IP address, opening ports on firewalls etc. The service binds to the AppFabric Service Bus and periodically pulls for client requests.

5 Accessing Cloud Storage by Process Controllers

Assuming the process controller is able to issue HTTP requests, one can build production monitoring system using Cloud Storage instead of on-site data store. A simple example of such system is shown on Fig. 3. In this section, the Windows Azure platform is considered as an example of the hosting services and Saia-Burgess PCD Series programmable logic controllers as an example of production monitoring system data source.

Proposed system contains only a few components: data sources, dedicated local area network and a router with NAT and/or firewall protecting data sources from external interference.

Instead of dedicated router, one can use a dedicated port on the main Internet router used by the organization. Production data stored in the cloud can be accessed by any client application and used to create reports. Despite its simplicity, proposed system architecture has two important drawbacks.

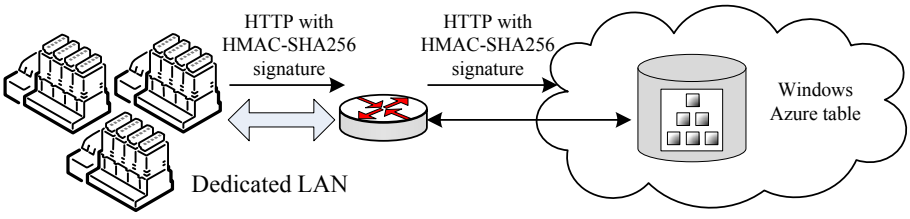


Fig. 3. Direct PLC access to Azure Storage service

1. Data source PLC must sign each request issued to the Azure Storage service with HMAC-SHA256 algorithm. That is computation intensive operation that increases the total processing time, thus affecting real-time properties of the PLC. Moreover, it is also required to distribute the secret key to every process controller used as data source, increasing maintenance efforts.
2. All requests are done with unencrypted HTTP protocol, as implementing SSL/TLS encryption layer within application software of the PLC seems to be unfeasible, mainly because of high encryption processing costs. As such, these requests are vulnerable to man-in-the middle attack leading to information disclosure.

Using SSL/TLS tunneling features available in some routers, the HTTP requests issued by data sources can be converted into encrypted HTTPS requests on the perimeter of the dedicated LAN, removing the possibility of information disclosure (Fig. 4).

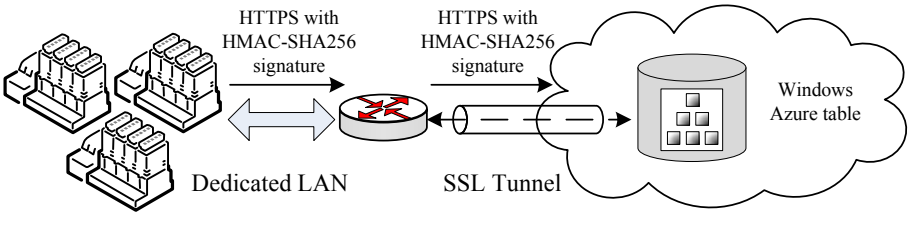


Fig. 4. Accessing Azure Storage with SSL tunneling

Because both sides of the SSL connection can be configured to authenticate each other using certificates, signing of HTTP requests can be avoided. However, as all requests to Azure Storage must be signed, additional proxy layer is needed in form of the Azure Web role instance (Fig. 5). The data sources issue HTTP requests that are tunneled securely within encrypted SSL connection. The Web role instance receives these requests, signs them and forwards to the Azure Storage service. While this solution increases operating costs, as computational time of the Web role instance is charged, it significantly simplifies implementation of the data source PLC application software and reduces hardware costs as less powerful PLCs can be used.

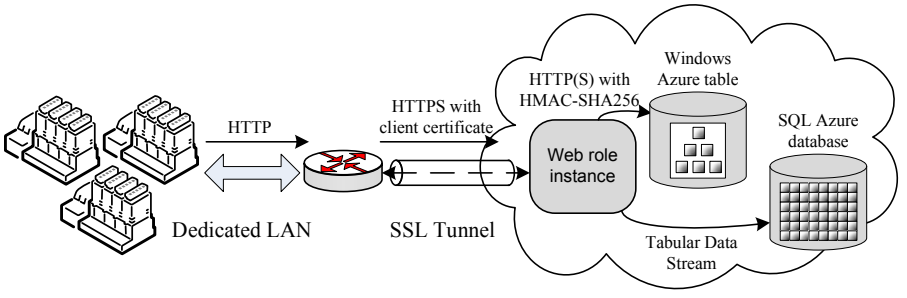


Fig. 5. Accessing Azure Storage through Azure Compute Web role instance

6 Performance Considerations

To validate presented concepts a monitoring system that follows Fig. 5 was developed and deployed. Saia-Burgess PCD3.M5540 programmable logic controller was used as the data source. A very simple implementation of HTTP client

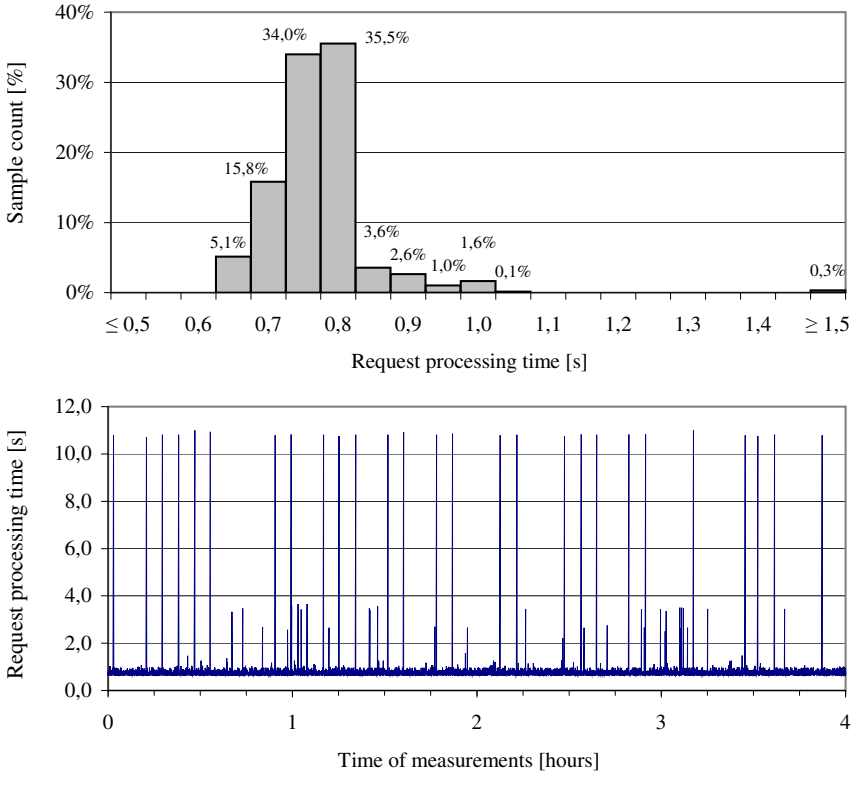


Fig. 6. Measured Azure Storage performance characteristics

software for that PLC discussed in [4] was further developed into production-quality HTTP 1.0 [19] client code. That implementation allows the PLC to access relational database servers exposing its functionality using HTTP-based RESTful [8] web services with JSON [20] data encoding. Such web service was developed with Microsoft ASP.NET Data Services Framework [21]. Microsoft Azure SDK [22] was used to handle data transfers to the Azure Storage. The service was tested and then deployed into Azure Compute Web role instance. HTTP-to-HTTPS tunneling was implemented with Stunnel 4.31 [23] running on Windows 2003 Server Standard Edition Service Pack 2.

During the experiment, HTTP requests were issued one after another. Each HTTP request was about 4 KB in size and response was about 4.5 KB in size. Broadband (over 10 Mbps) Internet connection was used. Results of measurements are shown on Fig. 6. Over 20 000 individual requests were processed. Azure Storage access times as seen by the data source are mostly less than a second. However, there can be observed regular delays in service response times that increase request processing time up to about 11 seconds.

7 Conclusions

Implementing production monitoring systems that use Cloud Storage services rather than on-site data storage centers is technically feasible with some of the currently available programmable logic controllers. That creates an interesting business alternative for some of small-to-medium sized companies, as fully featured monitoring systems can be built without capital costs of dedicated data storage infrastructure.

Unfortunately, PLC vendors rarely support creating custom communication protocols within application software for Ethernet-based networks. Moreover, implementing custom HTTP client encounters difficulties in both code complexity and execution time because programming languages that are available are rather inadequate for string processing that is necessary for HTTP-based communication. Above problems shall be resolved by PLC vendors, by implementing in the firmware fully featured HTTP 1.1 client software.

References

1. Saia-Burgess Controls Ltd.: Hardware Manual for the PCD3 Series. Document 26/789 Edition E8, Murten (2007)
2. ABB Automation Technologies: Industrial^{IT} System 800xA System Architecture Overview (2005)
3. GE Fanuc Automation: PACSystems^{TM}CPU Reference Manual. Document GFK-2222K (2007)
4. Jestratjew, A.: Improving Availability of Industrial Monitoring Systems through Direct Database Access. In: Kwiecień, A., Gaj, P., Stera, P. (eds.) CN 2009. CCIS, vol. 39, pp. 344–351. Springer, Heidelberg (2009)
5. Microsoft Corp.: Tabular Data Stream Protocol Specification. MSDN Library

6. SOAP Version 1.2 (Second Edition). W3C Recommendation (2007)
7. Fielding, R.T.: Architectural Styles and the Design of Network-based Software Architectures. PhD thesis, University of California, Irvine (2000)
8. Fielding, R.T., Taylor, R.N.: Principled design of the modern Web architecture. ACM Trans. on Internet Technology 2(2), 115–150 (2002)
9. Saia-Burgess Controls Ltd.: SAIA PCD Controllers with Ethernet-TCP/IP Manual. Document 26/776 Edition E3, Murten (2003)
10. Hirt, A.: Pro SQL Server 2005 High Availability. Apress (2007)
11. Microsoft Corp.: Microsoft High Availability Overview White Paper (2008)
12. Bennett, K., Layzell, P., Budgen, D., Brereton, P., Macaulay, L., Munro, M.: Service-based software: the future for flexible software. In: Proc. of 7th Asia-Pacific Software Engineering Conference APSEC, pp. 214–221 (2000)
13. Chappell, D.: Introducing the Windows Azure platform. Chappell & Associates (2009), http://go.microsoft.com/fwlink/?LinkId=158011
14. Microsoft Corp.: Windows Azure platform Service Level Agreements, http://www.microsoft.com/windowsazure/sla/
15. Chappell, D.: Introducing the Windows Azure. Chappell & Associates (2009)
16. Microsoft Corp.: Understanding the Table Service Data Model. MSDN Library
17. Microsoft Corp.: Queue Service API. MSDN Library
18. Microsoft Corp.: Windows Azure Storage Services REST API Reference, Authentication Schemes. MSDN Library
19. RFC1945 Hypertext Transfer Protocol – HTTP/1.0. Internet Engineering Task Force, The Internet Society (1996)
20. RFC4627 The application/json Media Type for JavaScript Object Notation (JSON). Internet Engineering Task Force, The Internet Society (2006)
21. Microsoft Corp.: ADO.NET Data Services Framework. MSDN Library, http://msdn.microsoft.com
22. Microsoft Corp.: Windows Azure SDK. MSDN Library
23. Stunnel – multiplatform SSL tunneling proxy, http://stunnel.mirt.net/

Speech Processing Application Based on Phonetics and Phonology of the Polish Language

Piotr Kłosowski

Silesian University of Technology, Akademicka Str. 16,
44-100 Gliwice, Poland
Piotr.Klosowski@polsl.pl

Abstract. The article presents methods of improving speech processing based on phonetics and phonology of Polish language. The new presented method for speech recognition was based on detection of distinctive acoustic parameters of phonemes in Polish language. Distinctivity has been assumed as the most important selection of parameters, which have represented objects from recognized classes. Speech recognition is widely used in telecommunications applications.

Keywords: speech, speech recognition, speech processing.

1 Introduction

Division of Telecommunication, a part of the Institute of Electronics Silesian University of Technology, since many years specializes in advanced fields of telecommunication engineering. One of them is speech communication applications. Main research areas on this field are: speech synthesis, speech recognition and speaker verification and identification systems. Typical speech communication applications are presented on Fig. 1 [1].

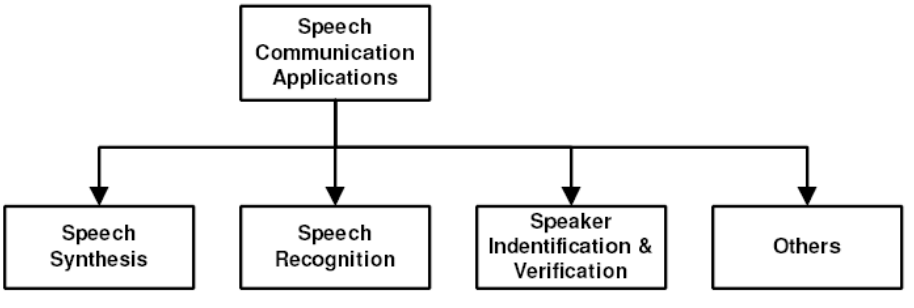


Fig. 1. Speech communication applications

2 Results of Research on Speech Synthesis

At present, the speech synthesis is widely used in many applications, the first of all in telecommunications [2,3]. In Institute of Electronics ware developed

A. Kwiecień, P. Gaj, and P. Stera (Eds.): CN 2010, CCIS 79, pp. 236–244, 2010.

two generation of speech synthesizer for Polish based on TTS (Text to Speech) technology. Structure of TTS synthesis system is presented on Fig. 2.

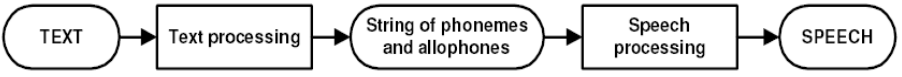


Fig. 2. Structure of text-to-speech synthesis system

The full TTS system converts an arbitrary ASCII text to speech. The first task of the system is to extract the phonetic components of the required message realized in text processing unit shown in Fig. 3.

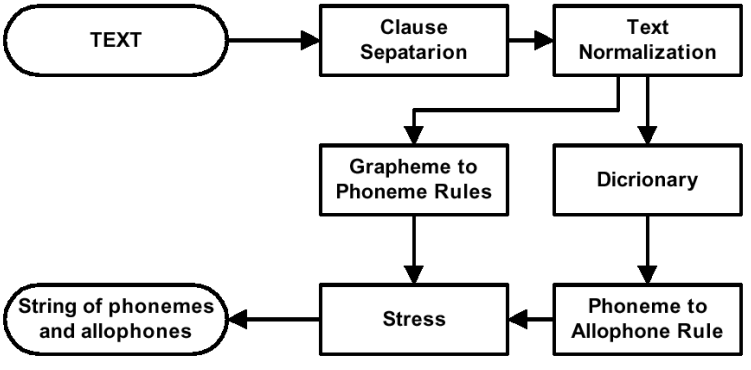


Fig. 3. Structure of text processing unit

The output of this stage is a string of symbols representing sound-units (phonemes or allophones), boundaries between words, phrases and sentences along with a set of prosody markers (indicating the speed of speech, the intonation etc.). The second part of the process is to match the sequence of symbols up with items stored in the phonetic inventory, link them together and send to a voice output device. This task is realized in speech processing unit shown on Fig. 4.

A combination of linguistic analysis must be done in the first stage which involves: converting abbreviations and special symbols (decimal points, plus, minus, etc.) to spoken form. On the Fig. 5 are shown two generations of speech synthesis systems for Polish developed by Institute of Electronics, Silesian University of Technology.

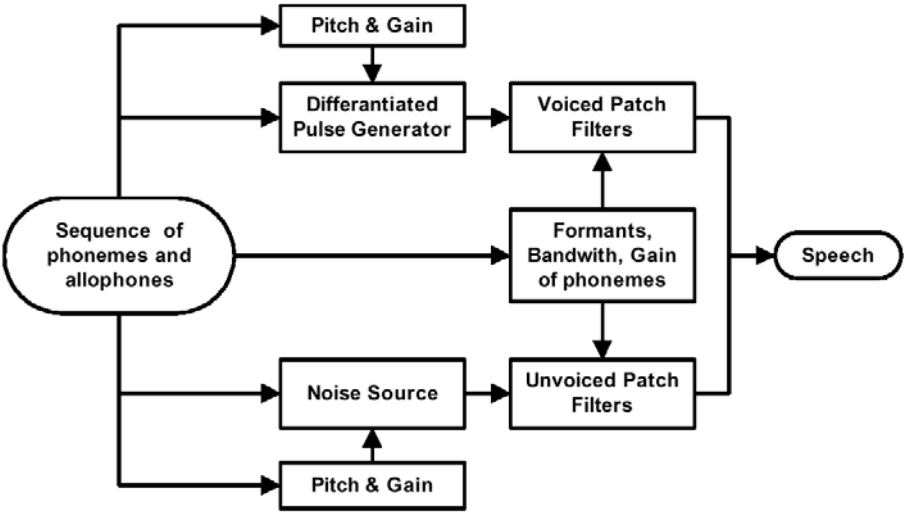


Fig. 4. Structure of speech processing unit

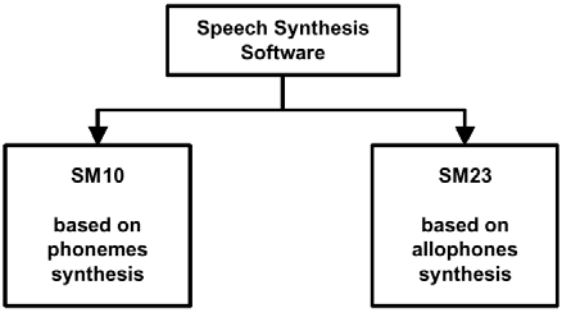


Fig. 5. Two generations of speech synthesis system for Polish developed by Institute of Electronics Silesian University of Technology

Developed speech synthesis systems are presented on Fig. 5:

- SM10 text-to-speech system for Polish, was the first speech synthesizer developed in Institute of Electronics, Silesian University of Technology, to simulate the human vocal tract, dedicated for blind persons. SM10 allows proper word pronunciation and word stress by means of full phoneme transcription. Speech synthesis was made on the phoneme level.
- SM23 is the next generation of speech synthesis system based on allophonic level. Allophonic speech synthesis quality is better than quality of SM10 phoneme speech synthesizer. SM23 software provides natural-sounding, highly intelligible text-to-speech synthesis.

3 Results of Research on Speech Recognition

Speech recognition is a conversion from an acoustic waveform to a written equivalent of the message information. The nature of speech recognition problem is heavily dependent upon the constraints placed on speaker, speaking situation and message context. Detailed structure of speech recognition process is shown on Fig. 6.

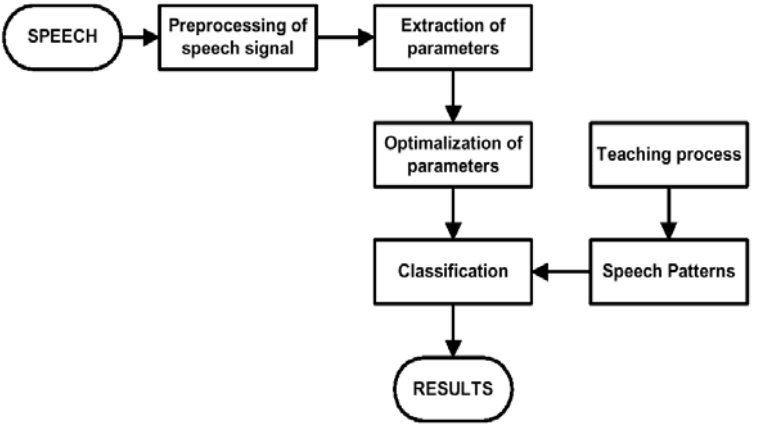


Fig. 6. Detailed structure of speech recognition process

Speech recognition process is realized in two steps presented on Fig. 7. In the first step speech signal is processed by phonemes recognition system. The result of this process is sequence of phonemes or allophones. This sequence is processed by phonemes to text conversion unit with elements of speech understanding system. Final result is this process is text.

Fig. 7. Two steps of speech recognition process

The second major of research in speech communication applications is speech recognition and particularly improving speech recognition process of Polish language using linguistic knowledge (phonetics and phonology) [4]. This idea is presented on Fig. 8. Improving speech recognition process is realized by using acoustic, phonetic, syntactic and semantic knowledge of Polish language.

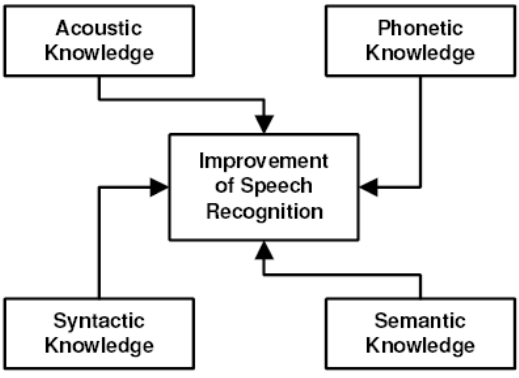


Fig. 8. Methods of improve speech recognition process

Result of this research was creation of multilayer speech recognition system. Each layer realizes one step of speech recognition process. There are: acoustic layer, articulation layer, phonetic layer, syntactic layer, semantic layer and application layer. Model of multilayer speech recognition system is shown on Fig. 9.

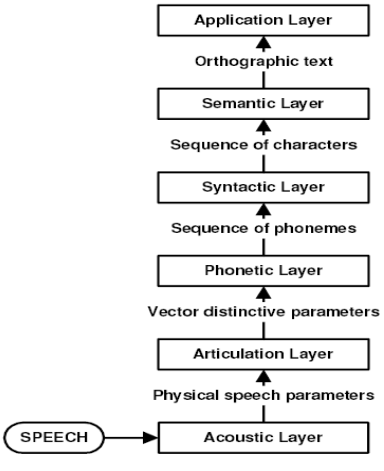


Fig. 9. Multilayer speech recognition system with elements of speech understanding

The first acoustic layer provides physical parameters of speech. Second articulation layer provides vectors of distinctive parameters of speech. Phonetic layer on the basics of these vectors generates sequence of speech phonemes. Syntactic layer using dictionary of pronunciation rules provide orthographical notation of speech. Semantic layer establishes of meaning orthographical sequents of characters and provides sentences in Polish language. Task of application layer depends on destination of speech recognition system. The new used method for

speech recognition was based on detection of distinctive acoustic parameters of phonemes in Polish language. Distinctivity has been assumed as a most important selection of parameters which have represented objects from recognized classes of phonemes.

4 Improving Recognition Process of Polish Phonemes

Phonemes are sound units determined meaning of words. Effective phonemes recognition, sound units of each language allow to effective recognizing continuous speech. Improving phonemes recognition process is possible ussing phonetics and phonology of Polish language [5,6]. The new method of speech recognition was based on detection of distinctive acoustic parameters of phonemes in Polish language. Each phoneme is specific by vector of distinctive parameters of speech signal. The first distinctive parameter means class of phoneme. Second means place of phoneme articulation. Third parameter means method of phoneme

k	Polish phoneme	Probability of *k*-th phoneme articulation	Number of distinctive parameters required to recognize *k*-th phoneme
1	b	0.013	3
2	p	0.027	3
3	d	0.019	3
4	t	0.038	3
5	ɡ́	0.001	3
6	ḱ	0.006	3
7	ɡ	0.013	3
8	k	0.023	3
9	v	0.030	3
10	f	0.013	3
11	z	0.015	3
12	s	0.026	3
13	ż	0.010	3
14	š	0.017	3
15	ź	0.002	3
16	ś	0.013	3
17	χ	0.009	2
18	ʒ	0.007	3
19	c	0.013	3
20	ǯ	0.0005	3
21	č	0.010	3
22	ʃ́	0.002	3
23	ć	0.011	3
24	i	0.034	3
25	y	0.035	3
26	e	0.088	3
27	a	0.080	3
28	u	0.029	3
29	o	0.078	3
30	m	0.030	2
31	n	0.034	2
32	ń	0.022	2
33	ŋ	0.00	2
34	r	0.007	1
35	l	0.018	1
36	j	0.039	2
37	ɰ	0.019	2

Fig. 10. Set of distinctive parameters of Polish phonemes with articulation probability and number of distinctive parameters required to recognize each phoneme

articulation. Average number of distinctive parameters required to recognize one phoneme equals 2.71, and was estimate using formula (1):

$$Ns = \sum_{k=1}^{M} p_k \cdot N_k = \sum_{k=1}^{37} p_k \cdot N_k = 2.71 \tag{1}$$

where: Ns is average number of distinctive parameters of speech, M number of phonemes, p_k probability of k-th phoneme articulation, N_k number of distinctive parameters required to recognize k-th phoneme.

Figure 10 presents set of distinctive parameters of Polish phonemes with articulation probability. Set of Polish phonemes is presented on Fig. 11.

Nr	Phoneme	Example word	Nr	Phoneme	Example word
1	i	wici	21	ś	**si**ano
2	y	syty	22	ź	**zi**arno
3	e	serce	23	χ	**h**igiena
4	a	baba	24	p	**p**raca
5	o	oko	25	b	**b**aba
6	u	buk	26	t	**t**rawa
7	i̯	jajo	27	d	**d**udek
8	u̯	łysy	28	k	**k**ot
9	r	rok	29	g	mogą
10	l	lato	30	ḱ	**ki**no
11	m	mama	31	ǵ	ma**gi**czny
12	n	noc	32	c	**c**acko
13	ń	koń	33	ʒ	na**dz**y
14	ŋ	ręka	34	č	**cz**arny
15	f	fala	35	š	dro**żdż**e
16	v	wada	36	ć	**ci**asto
17	s	sok	37	ʒ́	**dzi**edzic
18	z	koza			
19	š	szyszka			
20	ž	każdy			

Fig. 11. Set of Polish phonemes

5 Examples of Language Processing in Polish

We can distinguish two very important tasks in Polish language processing:

- letter-to-phoneme and phoneme-to-sound conversions in speech synthesis and,
- sound-to-phoneme and phoneme-to-letter conversion in speech recognition process.

The letter-to-phoneme conversion changes ASCII text sequences to phoneme sequences. The phoneme-to-letter conversion performs reverse operations. It is based on implementation and employment of rule-based system and the dictionary for exceptions. This is very crucial fragment of the code within the entire speech processing software. Pronunciation of Polish language words is not very complicated. Even though the letter-to-phoneme conversion has more than 90 pronunciation rules, which requires an exception dictionary. Each phoneme is actually represented by a structure that contains a phonemic symbol and phonemic attributes that include duration, stress, and other proprietary tags that control phoneme synthesis. This scheme is used for handling allophonic variations of a phoneme. The term phoneme refers either to this structure or to the particular phone specified by the phonemic symbol in this structure. Figure 12 present examples of this process.

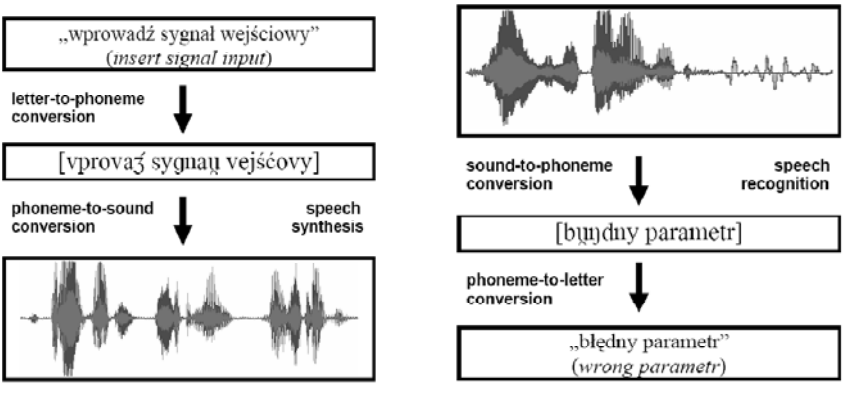


Fig. 12. Examples of letter-to-speech conversion and speech recognition

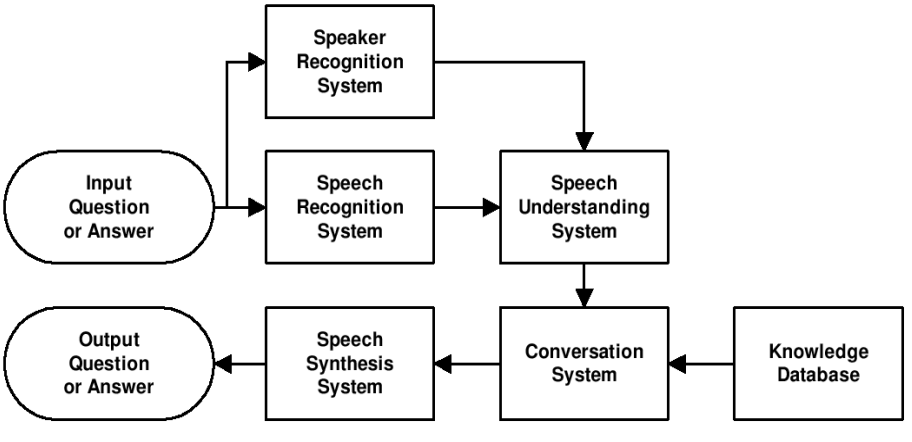


Fig. 13. Structure of dialog system with speaker reconciliation feature

6 Summary

The research on speech recognition is continued. At present efforts concentrate in creation efficient speech recognition system based on multilayer speech recognition model using distinctive parameters of speech. The second major of effort is creation speaker verification and identification system and implement some speaker identification algorithms in speech recognition system. Future goal of research is construction of full speech dialog system with elements speech understanding based on artificial intelligence technology. Structure of this system is presented on Fig. 13 [7].

References

1. Rabiner, L.R.: Applications of Voice Processing to Telecommunications. Proc. of the IEEE 82(2), 197–228 (1994)
2. Kłosowski, P.: Speech Communication Applications. In: The First Seminar – University of Central Florida and The Silesian University of Technology, Niedzica Castle, Poland (2002)
3. Kłosowski, P.: Speech Communication Applications. In: International Conference Programmable Devices and Systems PDS 2003 IFAC Workshop, Ostrava, pp. 332–337 (2003)
4. Ostaszewska, D., Tambor, J.: Podstawowe wiadomości z fonetyki i fonologii współczesnego języka polskiego. Uniwersity of Silesia nr 488, Katowice (1993) (in Polish)
5. Kłosowski, P., Izydorczyk, J.: Base Acoustic Properties of Polish Speech. In: International Conference Programmable Devices and Systems PDS2001 IFAC Workshop, Gliwice (2001)
6. Kłosowski, P.: Improving of speech recognition process for polish language. In: 5th International Conference On Technical Informatics, Timisoara, Romania, Transactions on Automatic Control and Computer Science, October 18-19, vol. 47(61), pp. 111–115 (2002)
7. Kłosowski, P., Pułka, A.: Polish Semantic Speech Recognition Expert System Supporting Electronic Design System. In: Proceedings of The International Conference on Human System Interaction: HIS 2008, Krakow, Poland (2008)

Distant Analysis of the GENEPI-ENTB Databank – System Overview

Paweł Foszner[1], Aleksandra Gruca[1], and Joanna Polańska[2]

[1] Institute of Informatics, Silesian University of Technology, Gliwice, Poland
{Pawel.Foszner,Aleksandra.Gruca}@polsl.pl
[2] Institute of Automatic Control, Silesian University of Technology, Gliwice, Poland
Joanna.Polanska@polsl.pl

Abstract. This paper presents the Internet application, which allows to perform distant statistical analysis of the data form the GENEPI-ENTB database. The database includes tissues from irradiated patients with different types of cancer linked out to a detailed description of treatment and outcome. The main purpose of the system presented in the paper is to provide to the users an access to the GENEPI-ENTB data and allow to perform statistical analysis of the data. The authors describe the system architecture, the analysis that can be done by tools available in the system and how exactly the system works. The paper also includes short description of the GENEPI-ENTB database, characterization of the project, and plans of the future development of the system.

Keywords: statistcal analysis, late effects of radiation tratment, distant analysis, R project.

1 Introduction

The degree of reaction of a human body to ionizing radiation strongly depends on the individual radiosensitivity. A standard treatment may result in a range of normal and tumour tissue reactions, thus ability of predicting the consequences of treating an individual patient with particular dose might significantly improve the results of a therapy and quality of a patient life after treatment. However, our knowledge of mechanism underlying inter-individual differences is still insufficient, thus in many clinical situations, radical doses are limited by the minority of patients whose normal tissues are particularly sensitive [1]. In Figure 1 a schematic representation of the frequency distribution of normal tissue responses amongst patients which would result from identical radiotherapeutic treatment is presented [2]. In a group of patients treated to the same radical dose, a wide range of reactions are observed and variation in sensitivity to gradation is known to be influenced by genetic factors. If we would know the genetic profile of the non- or over-responders (NR, OR), we could maximize radiation doses to cancer cells while minimizing damage to surrounding healthy tissue.

The GENEPI-ENTB project was established to create a database including normal and tumor tissue bank of a patients receiving radiotherapy which

A. Kwiecień, P. Gaj, and P. Stera (Eds.): CN 2010, CCIS 79, pp. 245–252, 2010.

could serve as an infrastructure for molecular research on tissues of irradiated patients [3]. The GENEPI-ENTB databank was established with EURATOM (European Atomic Energy Community) support in FP5 (fifth framework programme of the European Community for research) and was further developed and maintained as a part of the GENEPI-ENTB2 project in FP6. The dataset includes tissues from irradiated patients with head and neck, breast, rectal or prostate cancer linked to a detailed treatment (dose distributions with dose gradients, total dose, dose per fraction) and outcome data.

In this paper we present a system which allows to obtain distant access to the data included in GENEPI-ENTB database and to perform statistical analysis of the collected data. Currently, the database includes data for over 7 000 patients from 13 different medical centers, which makes GENEPI-ENTB database the largest infrastructure in this field worldwide and enables the selection of statistically relevant groups for analysis. To be able to perform efficient analysis of such amount of data, the system must provide to the user tools for selection different groups of patients and various features describing them. As the system is designed mainly for biologists and medics, the analysis offered by the system are descriptive statistics and simple statistical tests for independence. Additionally, due to the ethical issues, the system must strictly satisfy the security requirements, thus the access to the GENEPI-ENTB database cannot be freely available.

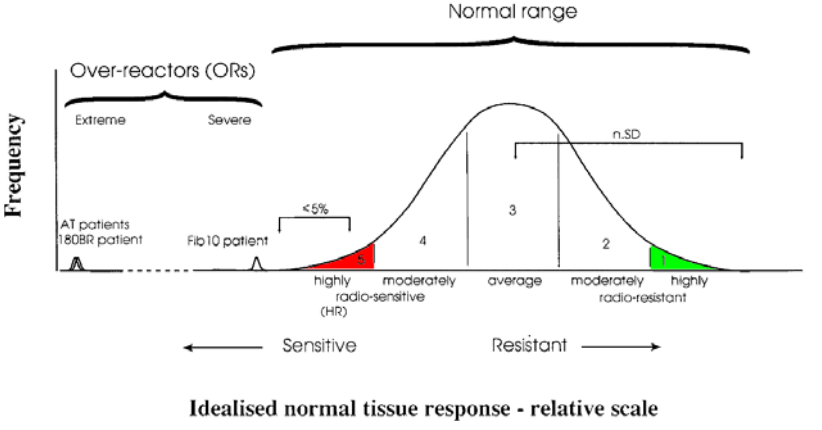


Fig. 1. This figure originates from paper [2] and describe the variation of responses amongst patients received the same therapeutic treatment

2 Statistical Analysis

In the current implementation, data is subjected to only simple statistical analysis. Exemplary, for specified search criteria, information is displayed in percentage regarding to sex of people that meet them. This method can be used also for

the other enumerated types. For continuous data, for example data on radiation dose, it can be read from the chart what percentage of people received a dose from a given range. Such an analysis provides only a general overview of the data (statistical information, eg. number of females, average age, etc.), but we want to know the dependencies between different features. For example, information about whether being overweight or smoking have an influence on the type of cancer or dependencies between type of cancer and gender. Such an analysis is the most interesting for us and that is the main purpose of this web site.

So what dependencies we are looking for? Everybody knows that smoking causes lung cancer, and if such an information is detected by the system it is not valuable. We should skip the obvious dependencies and take care of what can generate real profits. Site should then focus, for example, on data relating to individual patient radiotherapy. It is worth finding the relationship between type of radiotherapy, adopted medicines and type of cancer or size of side effects. Additionally, we can also attempt to make a comparison of different medical centers in terms of effectiveness.

The first and simplest statistical test that can be used for such analysis, is the chi-square test of independence [4]. This test is typically used to determine whether distribution of categorical variables differ significantly from the expected one. To apply this test, we need to create a contingency table consisting of rows and columns including frequencies of observed features. In this statistics we always have two features. First feature is described in columns and the second in rows, and on the cross-section we can see dependencies between features. Table 1 is an example of such a contingency table.

Table 1. General form of a contingency table

	feature 1 value 1	feature 1 value 2	...	feature 1 value k
feature 2 value 1	n_{11}	n_{12}	...	n_{1k}
feature 2 value 2	n_{21}	n_{22}	...	n_{2k}
...	...	...	...	...
feature 2 value r	n_{r1}	n_{r2}	...	n_{rk}

On the cross-section of the table there is a number of both features within each category. We denote these features by the letters X and Y. We can assume the following null hypothesis: the attributes are independent, what is denoted as H_0:

$$H_0 : P(X = x_i, Y = y_i) = P(X = x_i) * P(Y = y_i) \ , \tag{1}$$

and the alternative hypothesis: there is a dependency between attributes, denoted as H_1:

$$H_1 : P(X = x_i, Y = y_i) \neq P(X = x_i) * P(Y = y_i) \tag{2}$$

Finally, the chi-square statistic is computed as follows:

$$\chi^2 = \sum_{i=1}^{k} \sum_{j=1}^{r} \frac{(n_{ji} - \hat{n}_{ji})^2}{\hat{n}_{ji}} \tag{3}$$

where:

$\hat{n}_{ji} = \frac{\sum_{i=1}^{k} n_{ji} * \sum_{j=1}^{r} n_{ji}}{n}$ – theoretical number of elements in the sample,
n_{ij} – number of elements in the sample,
k – number of columns in the contingency table,
r – number of rows in the contingency table,
n – number of all elements.

3 System Architecture

The system was designed as the Internet application, for multiple users with different roles. Until now, there is a normal user role and the administrator role. The difference between them is essential. Normal user may be anyone who submits a registration form, and then must be approved by a system administrator. In the current implementation, anyone can fill in the form, so there is a risk that the administrator will have many forms to review. One idea is to reduce the access to registration form only to the users with e-mails from GENEPI Consortium. For example, users from Silesian University of Technology have an email addresses in a domain `polsl.pl`. Administrator will be able to manage the available domains or add users manually. Another more restrictive idea is to give possibility of adding new users only to administrator. Thus summing up the issue of users roles: normal users can only view data and use the functions available to them. The administrator has the rights of a normal user, and also can manage users, data, and the site as a whole.

The site aims to provide data from the database GENEPI and enable statistical analysis of these data. Data are provided as CSV files downloaded from the project homepage. Currently, the database is stored in about 108 separated files and each of them include different information. The files are divided into two groups. The first group of files describes different types of cancer. This group is divided into files which include information about different types of visits, side effects, disease status, etc. In the second group, files contains general information about patients, radiotherapy, etc. The number of files is very large, so extracting useful information from them is a non-trivial task. If there is a new version of GENEPI-ENTB database available, the administrator can download the new data from GENEPI site, and upload them on a server using an appropriate form. However, after updating the database there is still possibility to return to an older version. Administrator decides which version should be currently used and has the ability to change version of database. If he removes the current version, the system selects the most recent version of database. However,

if the system contains only one version of data and administrator will remove it, system will not contain any data, and none of the functionality associated with it will be accessible.

Storing the database in CSV files is the weakest part of the system. Access time is inefficient and thus the calculations are performed slowly. A significant improvement would be to place these data in a relational database which is planned in the future.

The application is based on standard open source tools and services. In addition, statistical analysis is performed using scripts written in R [5]. The Figure 2 presents the structure of the system architecture.

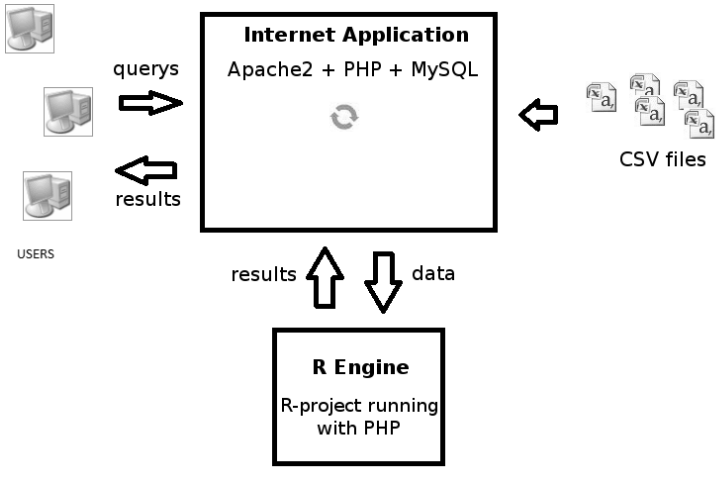


Fig. 2. The application structure

The whole logic of application, including login, registration, selection of data for analysis, user management, data management, etc. is written in PHP. All calculations and statistical analysis are performed using R engine. To store informations about registered users, its roles and versions of GENEPI-ENTB database, the MySQL database is used.

Taken into account that the data are valuable and collecting them cost a lot of effort, a very important aspect is safety (eg. protection from unauthorized access). As mentioned earlier, system functionality is available only for logged in users. But this is obviously not a sufficient security. All communication between client – server is also sent by encrypted connection. In addition, the CSV database files are available only for the server. The site is also protected against search engines and robots.

The system is still in development phase and can be found at site address `http://bioinfo.polsl.pl/genepi` [6]. Double login is require in current version of the site. The first is common, and allowing access to the site. Second login requires an individual password and login.

4 Exemplary Analysis of GENEPI-ENTB Data

After logging to the system, the user can select the features that are going to be analyzed. The user chooses whether it is data only from a single or multiple centers. He can choose patients witch specific types of cancer, sex, age, weight, etc. In the current implementation, after determination of criteria, the data are subjected to simple statistical analysis. In the future versions, the user will be able to choose more advanced tools for analysis.

The system analyze only the most important information, leaving comments, not relevant dates and so on. For more complex statistics, system prepares a file with the input data for the R, and then run appropriate R script (for example, chi-square test). The results of the script is read from the text file. Less complex statistics (count, average, etc.) are implemented in PHP scripts. Results are presented using pie charts, bar charts, and using text information depending on the selected data and the type of results. Supplied data are often incomplete, and therefore next to each result a chart is presented showing what percentage of the data was complete. The descriptive statistics are generated only for records including complete information.

On the Figure 3 we present the information about number of patients with breast cancer, with visits marked as "Follow Up" and "tumor status". As you can see the incompleteness of the data is a major problem. Approximately half of the records are left blank. Further analysis considers only the complete records.

Except simple descriptive statistics, the user can also perform statistical tests on selected features. For example, we can examine the relationship between the type of cancer and a gender. Table 2 presents an exemplary contingency table

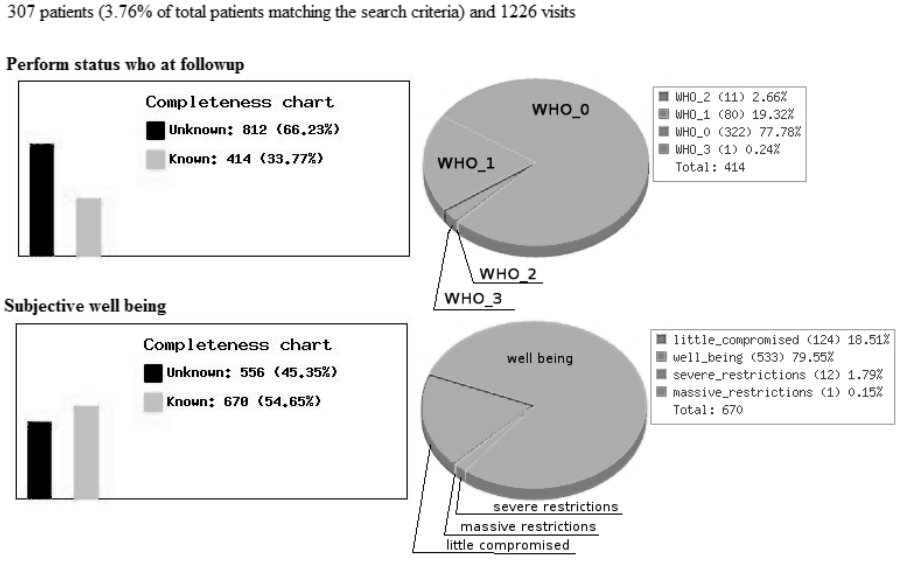


Fig. 3. The example results from the application

Table 2. Contingency table describing dependencies between gender and cancer type

	Female	Male
Breast	3867	3
Head & Neck	998	1436
Lung	75	251
Prostate	1	353
Rectal	86	133

including frequencies of different types of cancer depending on gender. The data presented in the table originate from the GENEPI-ENTB database.

The data were collected on February 2, 2010. As we examine the values in the table we noticed a strange case – there is a woman with a prostate cancer. Most probably it is related to problems with database inserts.

Based on data from Table 2, and using the Formula (3), we can compute a value of the χ^2 statistics assuming the following null hypothesis: there is no dependency between gender and a type of cancer.

The final value of the test is: $\chi 2 = 3869.7$. Assuming a significance level of the test equal to 0.01, and $(r-1)(k-1) = 4$ degrees of freedom, we can obtain from tables a critical value of the statistic which is 13.227. Based on the fact that $3869.7 \gg 13.227$, we can reject the hypothesis H_0 in favor of the alternative hypothesis H_1, which supports the conclusion that a probability of developing certain type of cancer depends strongly on a patient gender.

5 Conclusions

In this paper technical details of the Internet application for distant analysis of data from the GENEPI-ENTB database were presented. The main purpose of the system is to provide an access to GENEPI-ENTB resources by allowing to perform statistical analysis of the collected data. Presently, the system is available only for members of GENEPI Consortium due to the fact, that information stored in the database is unique and valuable research material. Additionally, the database is still at development stage and the information included in it is incomplete. However, for any research institution, there is possibility to apply for the access to the database.

The presented system is still extensively developed. Future works will focus on extending the functionality of the system by implementing more sophisticated tools for statistical analysis of the data. The authors also plan to introduce to the system methods of data analysis based on data mining techniques designed for incomplete data. Other works will include development tools for importing the information from CSV files to the relational database which will replace solution based on CSV files. However this part of the system development will be considered later due to the fact that the GENEPI-ENTB database is still in development phase and no one can assure that its structure is finally established.

The fact that the system provides interface for analysis of the gathered data is an important benefit. Until now, the research centers (hospitals, laboratories) could only analyze its own experimental data and the only possibility to obtain results submitted to the project by the other members was to import the whole database in a form of CSV files and write own scripts, and procedures to process these files. Another problem in data extraction from CSV files is related to the fact that there is no documentation describing the GENPI-ENTB database structure available, thus one must carefully analyze dozens of CSV files in order to understand the database structure, which is time consuming. The possibility of having access to the system via Internet is very convenient both for the system users and administrator. Users do not need to install any software on their computers to use the system – they can simply access the data and perform statistical analysis via Internet browser. From the administration point of view, if there would be any changes to the database structure or any additional functionality would be added to the application, a new version of the system can be provided to its users very easily.

Acknowledgements. This paper was partially supported by the European FP6 grant, GENEPI ENTB2, Genetic pathways for the Prediction of the effect of Irradiation – European normal an tumor tissue bank and data base.

References

1. Burent, N.G., Wurm, R., Nyman, J., Peacock, J.H.: Normal tissue radiosensitivity – how important is it? Clin. Oncol. 8(1), 23–34 (1996)
2. Burent, N.G., et al.: Describing patients' normal tissue reactions: concerning the possibility of individualising radiotherapy dose prescriptions based on potential predictive assays of normal tissue radiosensitivity. Int. J. Cancer. 79(6), 606–613 (1998)
3. Hőlscher, T., Verfaillie, C., Baumann, M.: Genepi-ENTB: The european normal and tumour tissue bank and database Radiotherapy and Oncology 78, S67 (2006)
4. Rice, J.A.: Mathematical Statistics and Data Analysis, 2nd edn. Duxbury Press, Belmont (1995)
5. R Development Core Team: A language and environment for statistical computing, reference index version 2.10.1. R Foundation for Statistical Computing, Vienna, Austria (2009), http://www.R-project.org
6. http://bioinfo.polsl.pl/genepi/

Monitoring Malware Activity on the LAN Network

Mirosław Skrzewski

Institute of Informatics, Silesian University of Technology
Akademicka 16, 44-100 Gliwice, Poland
miroslaw.skrzewski@polsl.pl

Abstract. Many security related organizations periodically publish current network and systems security information, with the lists of top malware programs. These lists raises the question how these threats spreads out, if the worms (the only threat with own communication abilities) are low or missing on these lists. The paper discuss the research on malware network activity, aimed to deliver the answer to the question, what is the main infection channel of modern malware, done with the usage of virtual honeypot systems on dedicated, unprotected network. Systems setup, network and systems monitoring solutions, results of over three months of network traffic and malware monitoring are presented, along with the proposed answer to our research question.

Keywords: network threats monitoring, malware detection, virtual honeypot systems.

1 Introduction

Many security related organizations [1,2,3] periodically release papers with information about the current state of network security, citing numbers of tops malware programs visible in some period of time. Dominant positions on these lists take programs called trojans, backdoors, bots, spyware and so on, and all statistics shows continuous growth in numbers of infected systems, installed malware programs and companies financial losses. But these types of programs are passive in nature and need to be delivered on place to make their malevolent activity. The only types of malware with own communication abilities are worms, but they are rather low or missing on these lists.

This raises the questions: how this infections spreads out, how invasive are this threats, how can they get to user systems? What are the main infection channels of modern malware? Published papers contain less and less details about the operation of captured examples of malware, it is difficult to find information how the infection of the system takes place.

To find the answers to these questions and to gather information about the threats landscape one can find on a small LAN network with direct Internet connection, an research environment with a set of selected types of honeypot systems were set on separated sub-network, exposed directly on internet traffic

A. Kwiecień, P. Gaj, and P. Stera (Eds.): CN 2010, CCIS 79, pp. 253–262, 2010.

(without any packet filtration). The systems operate 24/7 for over three month and were periodically monitored for signs of infections. At the end the systems were checked in detail for presence of any malware. The paper present details of the network and systems setup, monitoring tools used and results of system and network traffic monitoring. An initial analysis of gathered data concludes the paper.

2 Setup of the Research Environment

Research was conducted on separated class C sub-network connected with external world via access router through dedicated VPN channel without any packet filtration. LAN network traffic monitoring was based on the packet *argus* (*audit record generator and utilization system*), installed on dedicated system configured as a layer two bridge (without assigned IP address), placed between research network and access router. Access router works also as DHCP server, assigning IP addresses to systems on research network and as a DNS server. IP address space was divided in two parts: small upper group of addresses assigned to installed systems and remaining space of unused addresses, for network activity monitoring.

On the network some physical and virtual trap systems were installed and operate, waiting for malware infections. The trap (*honeypot*) systems were equipped with dedicated monitoring tools, to gather information about the infection and malware network operation. For threats investigation we experiments with two types of honeypots: full operating system with real system services as high interaction honeypot and modified operating system with partly simulated, selected services as middle interaction honeypot.

2.1 Virtual Platform Configuration

For installation of trap systems the Xen virtualization platform were selected. On systems with quad core processors and 4 or 8 MB RAM there were installed Xen 3.1 hypervisor and CentOS 5.4 operating systems as dom0 and configured as platforms for systems virtualization. Network operation of Xen hypervisor is based on a set of virtual bridges, defined for each physical system network interface [4]. For first physical network interface (*eth0*) a *xenbr0* is defined (Fig. 1), real network card is attached to bridge as *peth0* interface, dom0 virtual network interface is renamed to *eth0*, and for each virtual system (dom0 and domU) a pairs of virtual interfaces *vifi.0* and *tapi* connecting system with bridge were defined. On dom0 system all *xenbr0* interfaces are visible and accessible for traffic monitoring.

For network activity monitoring two tools were used. The network auditing packet (*argus*) was installed and configured for monitoring all communication on *peth0* interface, giving external global view of virtual systems communication and protocol analyzer Wireshark for on-demand detailed recording and analysis of traffic on selected *xenbr0* interface.

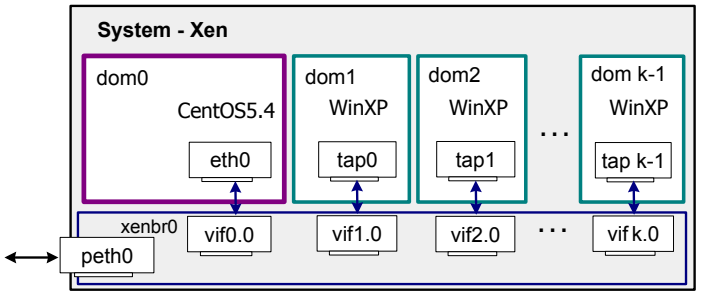


Fig. 1. Xen platform network communication interface model

2.2 Virtual Systems Configuration

A group of virtual Windows XP SP2 systems (domU) were installed on Xen platforms as high and middle interaction honeypot systems in the default, post install Windows configuration, with automatic actualization switched off. The trap systems were supplemented with selected monitoring tools, aimed on checking different area of system activity. On all honeypots file system integrity checking and system network activity monitoring tools were installed, as the main sources of information about possible malware infection.

The rationale behind this types of monitoring was based on information [5,6] that current threats are often released in many short series, are polymorphic in nature, so file system modification and/or network activity changes are rather only sure signs of system infection, even in the case of unknown zero-day threats. File system integrity checking is based on computing checksums of system files and storing them in dedicated database. For virtual Windows XP systems an *afick* (*another file integrity checker*) package [7] with perl compiler were used.

To keep the virtual systems slim and to not install too much tools on them, the file system snapshots were taken off-line, from another dedicated XP systems with honeypot virtual disk connected as second drive. The snapshot checksums databases were also left on this dedicated system. For network activity monitoring on each domU system an local network activity monitoring packet *TDILog* [8] were installed, recording activity of all open ports, open network connections and system processes responsible for network communication over system network (*tapi*) interface.

2.3 Honeypots Configuration

The systems for threats investigations were configured as one of two types of honeypots: high-interaction or middle-interaction honeypot, both based of Windows XP SP2 system installation. General idea of honeypot operation is to make available some resources or illusion of resources as a trap for malware program and monitor program behavior in its attempts of resource usage. The difference between honeypot types concerns the extent of resource reality and the method of malware operation monitoring.

High-interaction means that the real operating systems resources are available for malware operation (one allows for full system infection) and to monitor malware behavior one needs to monitor and analyze low level system operations relating to this resources. Honeynet.org [9] project packets *sebek* and *CaptureBat* are examples of such tools. If one is interested mainly at high level information – there was (or no) malware operation on the system, one can monitor integrity of file system and analyze, whether the changes in the file system are related to malware operation.

Middle-interaction means that resources or system services are simulated in details, and the simulator may record the malware code and monitor any operation of malware program in simulated environment at required level of details. One of such environments, packet *MultiPot* [10] from `iDefense.com` was used in our research. The packet attempt to simulate windows system after some selected worms attacks (simulate related backdoors on characteristics ports), and allows also for simulation of some vulnerabilities in services related to 445 TCP port operation, normally used by windows for server message blocks (SMB) protocol operations.

To start *MultiPot* operation on this port there was necessary to stop windows operation on it, via some registry modifications. Another packet installed on these honeypots, a behavioral analysis tool of applications for the Win32 operating system *CaptureBat* was capable of recording low level kernel operation for selected applications and was expected to show details of operation of malware program.

3 Threats Monitoring Course and Results

Threats research started from setting up and monitoring high-interaction honeypot systems. Virtual Windows XP systems with *sebek* were installed on Xen platform on research network and operate continuously few weeks, without any success (no visible malware infection). Operation of virtual systems on research network won't go smooth; there were a lot of virtual systems shutdowns and restarts. Then on other virtual XP system a middle-interaction honeypot packet (*MultiPot*) was installed along with *CaptureBat* tools and start to records same communication attempts to served ports, mainly to port 445. Both types of honeypots operate on research network for over three month. The *MultiPot* packet recorded a lot of code samples, systems recorded a lot of network statistics, but general results was quite odd – there was no one successful attempt to infect Windows XP SP2 system in default configuration, left alone operating continuously on the unprotected network.

3.1 Network Traffic in Research Network

Honeypot systems alone should generate minimal traffic on the network – there were no users, no working application, no user imposed system to system communication, so our expectation about volume of recorded network information

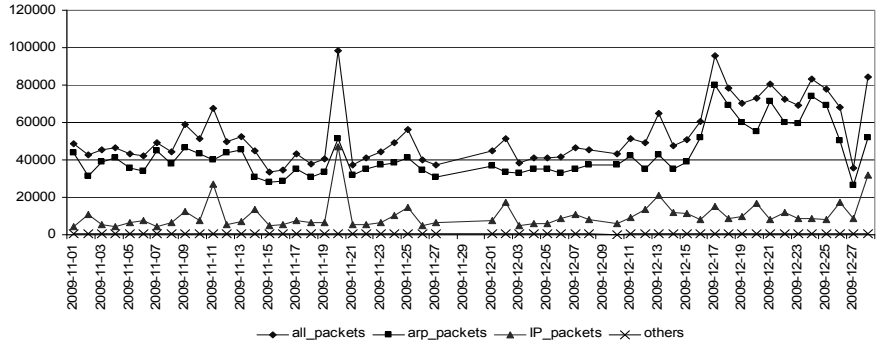


Fig. 2. Numbers of packet recorded daily on research network

was quite low – rare external connections to honeypot systems with malware infection attempts. To catch all information about these events, network traffic was monitored in many places – locally on honeypot systems, on virtualization platforms and on input and output of access router.

Realty overcomes the expectations, Xen monitoring system *argus* generated daily from 6 to 20 MB of logs, recording 40 000–100 000 packets a day (Fig. 2). Most of this traffic was due to arp packets (over 80%). Analysis of packets MAC addresses shows, that they are generated by the network gateway, trying to find systems assigned IP addresses from research network address space, in reaction to external packets directed to these addresses. Because most of the addresses from network address space remain unassigned, these arp request remain without response and gateway resend them after next external request.

Figure 3 show this in details. The mean number of scan per IP address is quite stable, and remains between 1 000 and 1 500, but there are addresses with much higher value, up to 4 000 in this case. Why some single IP is scanned much more then other remain unclear. Assigned IP addresses do not generate much

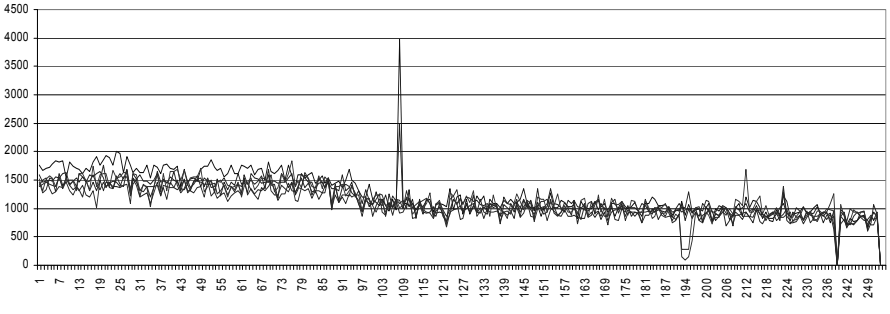


Fig. 3. Numbers of arp scans recorded per single class C IP address (horizontal axis) between 1–6 October

arp traffic, virtual systems addresses were recorded les then 10 requests a day, physical system addresses (gateway, Xen platforms) are often missing in records as there were no one arp request recorded for thise addresses.

3.2 Honeypot Operation Records and Captures

High interaction honeypots do not recorded interesting information at all. They were operating continuously about four weeks, there were from time to time unexpected shutdowns and restarts of Windows systems, but there were no signs of malware infections. Packet *sebek* was totally silent, and *CaptureBat* recorded only internal signs of TDILog operation (frequent writes to logs due to events on network interface) and system interface reaction to user attempts to check virtual machine state. It looks like there was no malware activity to record on the research network at all. Then on one of operating virtual Windows XP systems the *MultiPot* packet (middle interaction honeypot) was added and in a few hours start recording some suspicous transmissions on the network and catching some packets and code samples (Fig. 4).

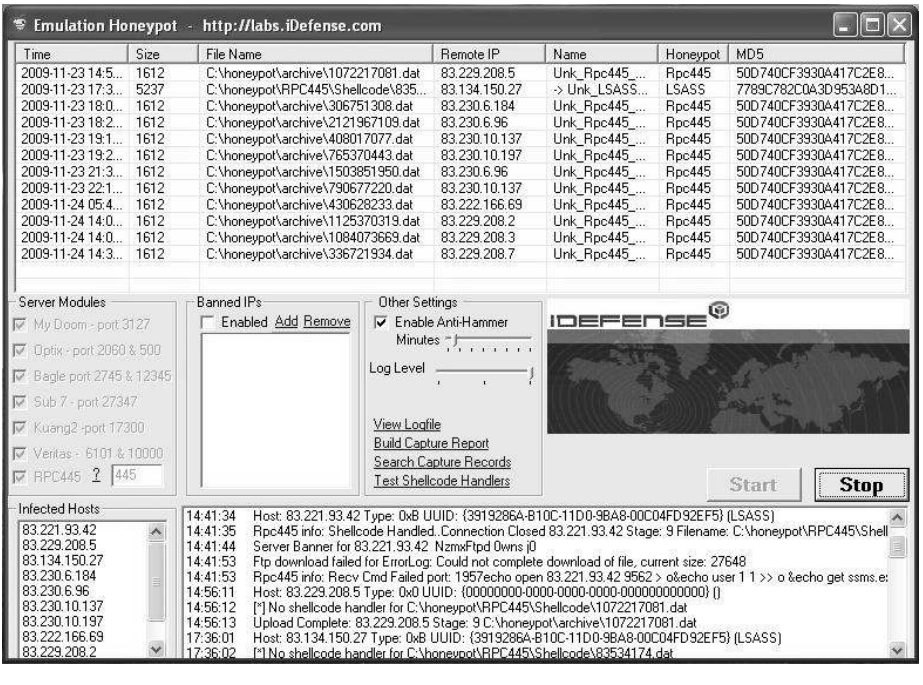


Fig. 4. Emulation honeypot (*multipot*) window with captured samples information

The tool simulates selected vulnerabilities of Windows system and has prepared handlers for some types of shellcodes. Captured code are classified as junk, received commands, received files, shellcodes, malware samples and stored

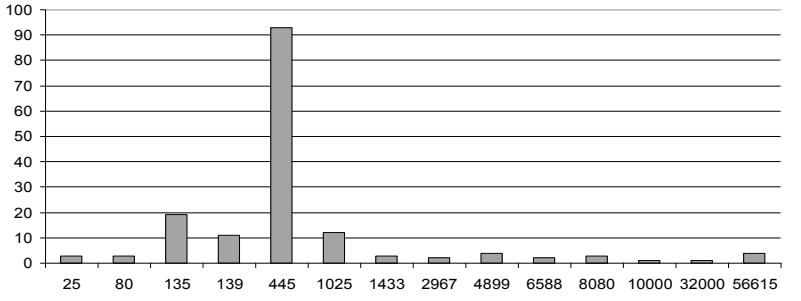


Fig. 5. Daily numbers of connection attempts to various honeypot ports

in subdirectories of main directory honeypot. Current operation are presented in *MultiPot* window, and are recorded in daily rotated log. Samples has various sizes, smaller then 1 KB were stored as junk; shellcodes, commands and recv_files were from 2 to 10 KB, and malware samples sizes range from 80 to 300 KB. Nearly all samples were captured by modules simulating services on port 445. It was the most scanned port (Fig. 5) daily, but other also scanned ports has no prepared shellcode handlers.

Each log record contains time of the event, short event description and for acquired code sender IP address, path to directory for given sample type and assigned random sample name. Figure 6 presents log excerpts describing the events related to acquisition of different code samples. The first line states the event of storing short (137 B) sample in junk folder. Second excerpt describes malware attempt to infect the system, there was no shellcode handler for this attack, so only shellcode sample was recorded and stored in archive folder. The last excerpt shows full shellcode handler operation. External system sends shellcode to trap

```
10:54:08  Moving to Junk Folder: C:\honeypot\RPC445\Shellcode\843862643.dat Size: 137 Stage: 1
          Host: 79.163.17.131

20:40:14  Host: 83.167.124.27 Type: 0x0 UUID: {00000000-0000-0000-0000-000000000000} ()
20:40:15  [*] No shellcode handler for C:\honeypot\RPC445\Shellcode\1087311903.dat
20:40:15  Upload Complete: 83.167.124.27 Stage: 9 C:\honeypot\archive\1087311903.dat

22:29:24  Host: 83.97.229.82 Type: 0xB UUID: {3919286A-B10C-11D0-9BA8-00C04FD92EF5} (LSASS)
22:29:25  Rpc445 info: Shellcode Handled.. Connection Closed 83.97.229.82 Stage: 9 Filename:
            C:\honeypot\RPC445\Shellcode\1251912246.dat
22:29:31  Server Banner for 83.97.229.82  NzmxFtpd 0wns j0
22:29:35  [*] FTP Downloaded File: C:\honeypot\RPC445\1015390059.dat
22:29:35  Rpc445 info: Command: port: 1957 echo open 83.97.229.82 16093 > o &echo user 1 1 >> o
            &echo get ssms.exe >> o &echo quit >> o &ftp -n -s:o &del /F /Q o &ssms.exe <CR>
            <LF> <CR> <LF>ssms.exe <CR> <LF> <CR> <LF>
22:29:35  Upload Complete: 83.97.229.82 Stage: 9 C:\honeypot\archive\1015390059.dat
```

Fig. 6. *MultiPot* log excerpts with samples acquisition processing

system, the code was stored and successful processing of shellcode was simulated. Then attacked system opens back ftp connection to sender system and the main malware code was transferred on attacked platform, attempted to execute and stored as a second code sample in this event.

Operation of *MultiPot* system was quite successful, there were recorded 15–25 malware samples per month, and two to three times more shellcode samples per virtual system. To identify malware samples, for each file there were computed MD5 hashes and searched via Google site for information. Some of the samples were recorded in many places on Internet, and have assigned some descriptive names, like Trojan.Win32.Kreeper.aub or Backdoor.Win32.Rbot.rgk, the other were only recorded on malware hashes lists in one or two sites.

After some time spent on Google searches it comes out, that there exists many confusing descriptions and names for the same hash value, many sites or companies use different classification for the same files, many different hashes has assigned the same generic malware name, like Trojan.Agent/Gen-Rbot[Worm] and part of the samples – about 20–25% is unknown on Internet at all. The search for shellcode information on internet was totaly unsuccessful. There are recorded by the tool many copies of basic two – three sample hashes, and their values are also totally unknown on internet.

4 Conclusion

The operation of virtual Windows XP SP2 systems as high and middle interaction honeypots on direct to Internet connected network provide us with some answers to our question about malware network operation, but also rise some others. From the tests with *MultiPot* system it is clear, that there are a lot of chances to get system infected during network operation, especially without any packet filtration, and that there are plenty known and unknown worm programs active on the network, trying to infect vulnerable systems, so network remains one of the important malware spreading channel.

From the operation of high interaction honeypot it is quite unclear, why rather old, unpatched Windows XP SP2 systems in default configuration remains uninfected after few month of continuous operation on the network. To make sure, that tested trap systems were not infected during network tests, after removing from the systems captured samples, at the end of the tests, virtual systems were scanned with antivirus internet scanners with current signatures databases. The scanners – Panda Security Active Scan 2.0, McAfee Online and Eset Online Scanner confirms, that there are no active malware infection of scanned systems.

Active Scan found some copies of recorded samples in one of *MultiPot* directory and classified most of them as W32/Gaobot.PCK.worm and one copy of Bck/Sdbot.JED.worm, and on another system in `Temporary Internet Files \Content.IE5` directory five suspicious files with the names `22[1].exe`, `27[1].exe` and so on, classified as Win32/AutoRun.IRCBot.DI worm and variant of Win32/Injector.AOA trojan. The same files were identified as malware by Online Scanner. McAfee scanner found also as unknown version of virus the *sebek* packets, installed on the system.

How to explain the fact, that Windows XP SP2 proves resistant to malware attacks on unprotected network? There is no easy explanation. Chances are, that used systems are national language (polish) distribution, slightly different from standard us english version, so carefully crafted shellcode may not work correctly in this environment. Maybe modern malware is also not so back compatible with older "legacy" versions of Windows system and is targeted on newer system versions (Vista, Windows 7), but this is rather weak explanation.

One of our experiments may a bit support this thesis. One of high interaction honeypots was also installed on Windows XP SP3 system and run on the same Xen platform with two SP2 systems. Xen hypervisor assign domU systems a number ID, starting from 1 for first virtual system. This ID's are incremented after each restart of virtual system. After checking the status of virtual systems on Xen platform, with Xen uptime of 30 days a SP3 system has ID of 104, and two SP2 systems has ID of 6 and 26. The only explanation of this numbers is that SP3 system suffers 3 to 4 times more frequent automatic restarts then SP2 systems, probably due to more often unsuccessful malware attacks.

But maybe the explanation is different and quite simple – these honeypots are Windows virtual systems, and malware programs after few first steps of execution may discover virtual environment and cease own operation. Captured samples of code are often somehow protected against simple strings checking, but not always. One of the samples, with MD5 hash 93218CAC0574F19DD8D6C10D3FE8C1B8, size 335 872 bytes, contains such unicode strings:

```
...
regmon.exe
procmon.exe
filemon.exe
ollydbg.exe
[Regmon] - [Filemon] - [Procmon] - [OllyDebug] Detected!
VM Additions S3 Trio32/64
S3 Trio32/64
VirtualBox Graphics Adapter
VMware SVGA II Detected!
...
```

which suggests that some tests of its execution environment are in place and there are attempts of detection of active process monitors, debuggers and virtual machine environment. Maybe this is the real explanation of the long time of unsuccessful operation of high interaction honeypots. If this will turn out to be the truth, it can be the strong vote for system desktop virtualization as one of the possible forms of some malware protection.

References

1. Annual Report PandaLabs (2009), http://www.pandasecurity.com/img/enc/Annual_Report_Pandalabs_2009.pdf
2. The Top Cyber Security Risks, http://www.sans.org/top-cyber-security-risks/

3. Sunbelt Software Announces Top 10 Malware Threats for November, http://www.sunbeltsoftware.com/Press/Releases/?id=323
4. XenNetworking – Xen Wiki, http://wiki.xensource.com/xenwiki/XenNetworking
5. Q1 2007 Malware Outbreak Trends: Server-Side Polymorphic Malware Explodes Across Email, http://www.commtouch.com/files/document%20library
6. Q3 2009 Internet Threats Trend Report, http://www.commtouch.com/download/1548
7. AFICK (Another File Integrity Checker), http://afick.sourceforge.net/
8. Skrzewski, M.: Wykrywanie działania niepożądanego oprogramowania. In: Nowe technologie sieci komputerowych, XIV konferencja Sieci Komputerowe, Zakopane 2007, WKiŁ, Warszawa (2007)
9. The Honeynet Project, http://www.honeynet.org/project
10. http://labs.idefense.com/software/malcode.php#more_multipot

Using Mobile Devices for User Authentication

Jacek Lach

Silesian Technical University, Akademicka 16, 44-100 Gliwice, Poland
jacek.lach@polsl.pl
http://www.polsl.pl

Abstract. One of the most common methods of user authentication utilizes passwords. This technique is widely spread and well known. Its main drawback is that it stresses human memory to allow for successful authentication and still remains in the domain which is called weak authentication. This paper presents a solution to this problem by introducing a form of two factor authentication by utilization of smartphones. The idea of such system is presented as well as a sample implementation, called MOTH (an acronym for MObile auTHentication), which allows for user authentication. The discussion of the system's security and its limitations is also presented.

Keywords: security, user authentication, mobile devices.

1 Introduction

Authentication is the binding of an identity to a subject [1]. In simple terms authentication is proving to the system that the user is who they claim to be. Most common form of authentication is through utilization of passwords. Passwords are example of mechanism which is based on what people know. This is one of three fundamental factors that are commonly used for authentication. The three factors are [2]:

- something known – password, PIN,
- something possessed – token, smart card,
- something inherent – fingerprint, signature.

Passwords are both well known (it is most common way of authentication) and widely accepted (in contrast to some biometrical techniques where physical contact is needed to acquire authentication pattern). Classical scheme for using passwords in UNIX system was presented by Morris and Thompson in [3]. Passwords are also called weak authentication. Several weaknesses as well as improvements were described [4,5,6]. For common users one of most important drawback is need for remembering passwords, that to be secure should be both long and dedicated to single service which makes them rather hard to memorize. To overcome this shortcoming solutions coming from two other groups came to play. Both groups are nowadays well established and play important role in world

A. Kwiecień, P. Gaj, and P. Stera (Eds.): CN 2010, CCIS 79, pp. 263–268, 2010.

business. When solution uses techniques from two groups it is known as two factor authentication and is considered as strong authentication. Systems which attract much attention at present combine passwords (or PINs) with "something you have" or "something you are". Example of the first case is solution promoted by ie. RSA Corporation which bind password with electronic token which must be used for successful authentication. Another example would be systems which reduce additional costs by using something many users already have: cellular phones. Systems which use SMS to provide user with additional authentication data are currently very popular. The latter case, when password is used along with biometric pattern, is robust enough when it comes to algorithms, the problem still remains in low accessibility to scanning devices or low quality of samples (ie. using low quality telephone network to acquire sample). Current two factor authentication solutions do not relieve user from remembering passwords as they are seldom a combination of something possessed and inherent. When strong authentication is not the case, often techniques from the group of "something you have" are used.

Mobile devices are nowadays very popular, with its rising processing power they become more and more important in everyday life. From that fact arise security considerations as those devices frequently store valuable and sensitive data. As security consciousness rise those devices become fairly well protected against physical theft making them good candidates for additional components of security architecture. This potential has been already noticed and several solutions were proposed. Oprea et al. in [7] propose using mobile device as a security token that would allow untrusted workstation temporary access to trusted service. Mobile devices such as smartphones were also proposed as a security token by Wu et al. in [8] to access remote service via security proxy which uses SMS for delivering authentication information. Ready to use implementations of electronic wallets for storing passwords on mobile devices running most popular mobile operating systems are also available.

In this paper a system called MOTH is proposed for utilization of a mobile device as a security token. This token communicates with the system by a Bluetooth link. When the user is authenticated by the system, the mobile device presents authentication data. A private key that is used for signing authentication data in the mobile device is encrypted so that theft of the device would not compromise the sensitive data. Next section explains the details of the solution with some technical notes on implementation issues. Further some security aspects of the proposed solution are discussed as well as limitations that arise while using proposed system.

2 System Overview

In proposed system a mobile device with Bluetooth interface is used as a security token. Mobile device is 'something you have' that is used for the authentication. System architecture is presented in Fig. 1. At the beginning workstation as well as the mobile device have to be properly configured for the authentication. Workstation has to be provided with the public keys of all the users

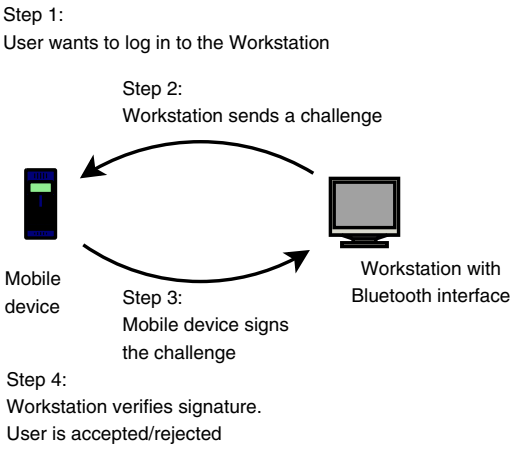


Fig. 1. An architecture of the proposed solution

which should have access to the system. It should also have a properly configured Bluetooth interface for the communication with users' mobile devices. For granting access to the system it should also run dedicated authentication service which in the implemented solution is provided by MOTH system. Mobile device should also be equipped with the dedicated software as well as the user's private key. When the user wants to access the system he begins standard authentication and ensures that his mobile device runs authentication software. If the user authenticates to the system for the first time it may be necessary to properly configure the Bluetooth link, after proper pairing of devices this step can be later automated depending on the configuration of the device. After user enters his login name further authentication is accomplished with user's mobile device. Workstation optionally can verify Bluetooth address of mobile device. In the next step workstation connects to the mobile authentication service and sends a random challenge which is signed by a mobile device. The signature is verified by the workstation and after the verification the user is allowed to access the system. The solution is an implementation of a protocol presented in FIPS 196 [9]. The document states that protecting private key with password is optional but in proposed solution this is obligatory making the solution more secure as two factors are needed for authentication: hardware device with installed private key (*something possessed*) and password for accessing the private key (*something known*). After accessing the service user ends his session in a traditional way and no further communication with mobile device is needed.

3 Implementation

The prototype of the proposed architecture was implemented as MOTH system. MOTH provides both parts of the authentication service that is needed. Since the implementation is under development it supports now only Linux system on the workstation side and mobile devices which are able to run Java midlets.

Authentication service on the workstation part is implemented as Linux Pluggable Authentication Module (PAM). It is up to the administrator to configure the system for using PAM for system wide authentication, but since this is a standard configuration of most novel Linux distributions, the integration of MOTH module should be straightforward. It is recommended to configure the system to use only MOTH service for authentication and not to fall back to standard authentication with passwords as this would be an obvious way to circumvent whole solution. When MOTH module is initialized it receives user's login name and searches for the authentication record which stores user's public key and the Bluetooth address of his mobile device. If the user is known to MOTH it then initializes searching for MOTH service on the mobile device. This i done by searching for UUID which leads to identifying port on which Bluetooth connection is set. After service is located the connection is made using Bluetooth socket for further communication which begins with sending random challenge. The response from mobile device is a signature of the challenge. The result of signature verification is returned as result from the authentication module.

MOTH mobile service is run by the user which should provide correct password to access his private key. After running the midlet user can edit his private key or run authentication service using previously selected key. When the challenge is received, it is signed using Digital Signature Algorithm and the signature is returned to the workstation.

MOTH service has been implemented and tested on Linux system running kernel 2.6 with *bluez* – standard Bluetooth protocol stack. For cryptographic functions *OpenSSL* library was used. Java midlet was created with *Netbeans IDE* with *Sun's Wireless Toolkit*. During the implementation the Bluetooth support was provided by *BlueCove* library. Implementation of Digital Signature Algorithm came from Bouncy Castle cryptography architecture provider. This provider delivers classes for *java.lang* package which made utilization of code obfuscator necessary. Code obfuscator additionally allowed for considerable reduction of midlet size which with provider library became significant.

4 MOTH Security and Limitations

The main advantage of the proposed solution is that it brings two factor authentication to the system in an easy and almost costless way. The ease of deployment is ensured by using standard Plugabble Authentication Modules for Linux. Low cost of deployment is achieved by choosing Java midlets for mobile part of the service which can be run on devices already used by users. In MOTH system two factors must be used for the user to authenticate properly. First factor is possessing physical device on which system is run and second factor is knowing the password to access user's private key. The usage of the latter factor is enforced by MOTH: private key can not be stored until password is configured.

Besides solution's obvious advantages one must also be aware of its limitations and security issues that can arise during its usage. Though the usage of digital signatures to solve the problem of user authentication is well described

it still leaves some open issues to be resolved. The main issue is the problem of binding user to his key. In digital world most common solution to this problem is using digital certificates issued by trusted third party. In MOTH system, digital certificates are not used. It is up to the administrator of the system to ensure that in MOTH configuration database correct public keys for given users are configured. With correct user to key binding ensured by system administrator a proper authentication can be performed by verification of signature issued by user's software.

For the MOTH system to increase system security there must be a system configuration which enforces users to use MOTH. If the system is configured in a way that it falls back to standard passwords when MOTH authentication fails (possible with PAM module stacking) the whole solution is no more secure than standard configuration with passwords. Another important fact that has to be taken into consideration is the fact that mobile device is not a smart card and data stored inside can be read once device is stolen. There is no hardware security involved here but enforcing private key encryption minimizes this risk factor.

Proposed solution is especially dedicated for solving issues that arise during local login process such as situations when someone is being watched while entering the password or when one would like to circumvent the usage of a key logger. Limiting usage of MOTH system to local logging-in is also one of its limitations as it currently does not allow to authenticate the user to begin a remote session. Another limitation is platforms on which MOTH system can be used, currently only Linux systems. In current version of MOTH system there is no device authentication. Neither is the workstation authenticated nor is the mobile device. Mobile device's hardware address can be stored in MOTH configuration but its usage for mobile device authentication is limited as it is possible to sniff this information off of the air and device with faked address can be configured. Nevertheless, user authentication can be successfully carried out, though there exist potential vulnerability to spoofing.

5 Conclusion and Future Work

This paper presents authentication scheme which uses popular mobile device in place of a typical hardware token. Limited implementation of subset of the system specifications was also prepared and described. Implemented solution is not yet final and several features are to be applied. Among most important ones one can enumerate:

- Signature algorithm negotiation. Currently the default algorithm is Digital Signature Algorithm and there is no way to change it besides source code modification. In the future RSA and ECC algorithms could be implemented. The negotiation that would precede exchange of data should establish common algorithm based on algorithms implemented both on the workstation and on the mobile device as well as the availability of keys suitable for given algorithms.

- Multiple keys support. Currently only one key is used for signing the challenge. If the device should be used for another set of keys then there is no way to associate different keys with different users, only one key can be defined.
- Session timeout. In current version if the midlet is left running it stores the password needed to access private key. If the device is stolen with running midlet it can be used to authenticate as the user.
- Multiple operating system support. Though mobile part of the MOTH system is quite portable as it employs widely available Java virtual machine, the part which is run on the workstation is currently limited only to the Linux operating system. MOTH system should also be integrated with MS Windows^{TM}platform.

Though the implementation is not finished and will be further worked on it can already be used in environments where enumerated shortcomings are not an issue.

References

1. Bishop, M.: Computer Security. Art and Science. Addison-Wesley, Reading (2005)
2. Menezes, A.J., Oorschot van, P.C., Vanstone, S.A.: Handbook of applied cryptography. CRC Press, Boca Raton (1997)
3. Morris, R., Thompson, K.: Password Security: A case history. Communications of the ACM 22(11), 594–597 (1979)
4. Klein, D.V.: Foiling the Cracker: A Survey of, and Improvements to, Password Security. In: Proceedings of the second USENIX Workshop on Security, pp. 5–14 (1990)
5. Manber, U.: A simple scheme to make passwords based on one-way functions much harder to crack. Computers and Security 15(2), 171–176 (1996)
6. Yan, J.: A Note on Proactive Password Checking. In: Proceedings of 2001 ACM New Security Paradigms Workshop, New Mexico, USA (September 2001)
7. Oprea, A., Balfanz, D., Durfee, G., Smetters, D.K.: Securing a remote terminal application with a mobile trusted device. In: Computer Security Applications Conference, pp. 438–447 (2004)
8. Wu, M., Garfinkel, S., Miller, R.: Secure Web Authentication with Mobile Phones. In: DIMACS Workshop on Usable Privacy and Security Software (2004)
9. NIST. Entity Authentication Using Public Key Cryptography. FIPS PUB 196, http://csrc.nist.gov/publications/fips/fips196/fips196.pdf

SPOT: Optimization Tool for Network Adaptable Security

Bogdan Ksiezopolski[1], Pawel Szalachowski[1], and Zbigniew Kotulski[2,3]

[1] Institute of Computer Science, Maria Curie-Sklodowska University,
pl. M. Curie-Sklodowskiej 5, 20-031 Lublin, Poland
[2] Institute of Fundamental Technological Research of PAS,
Pawiskiego 5B, 02-106 Warsaw, Poland
[3] Institute of Telecommunications of WUT,
Nowowiejska 15/19, 00-665 Warsaw, Poland

Abstract. Recently we have observed the growth of the intelligent application especially with its mobile character, called e-anything. The implementation of these applications provides guarantee of security requirements of the cryptographic protocols which are used in the application. Traditionally the protocols have been configured with the strongest possible security mechanisms. Unfortunately, when the application is used by means of the mobile devices, the strongest protection can lead to the denial of services for them. The solution of this problem is introducing the quality of protection models which will scale the protection level depending on the actual threat level. In this article we would like to introduce the application which manages the protection level of the processes in the mobile environment. The Security Protocol Optimizing Tool (SPOT) optimizes the cryptographic protocol and defines the protocol version appropriate to the actual threat level. In this article the architecture of the SPOT is presented with a detailed description of the included modules.

Keywords: security tools, security metrics, quality of protection, adaptable security, security models, security architecture, cryptographic protocols.

1 Introduction

Nowadays, advanced teleinformation technologies provide us with a wide range of possibilities for the development of industry institutions and public services. Emphasis is put on the development of well-available, mobile information services called e-anything like e-government, e-money and e-banking. Implementation of these services would be connected with the choice of proper level of security information sent between parties of protocols [1,2]. One of the important problems is establishing an appropriate level of security information, represented by security services in a given protocol. Traditionally, the problem was solved

A. Kwiecień, P. Gaj, and P. Stera (Eds.): CN 2010, CCIS 79, pp. 269–279, 2010.

by providing the strongest possible security. However, the use of strong mechanisms may deteriorate the performance of a device with limited resources and pave the way for new threats, such as, resource exhaustion [3,4]. In the end, it decreases system efficiency, availability and introduces redundancy. Another effect of overestimated security mechanisms is increasing the system complexity, which later influences implementation of a given project and imposing restrictions that decrease their functionality. The adequate solution in such cases is the introduction of adaptable (or scalable) security model for the protocols, which can change the security level depending on particular conditions that take place at a certain moment and in given external conditions.

In the literature the security adaptable models are introduced as the Quality of Protection (QoP) models [5,6,7,8,9]. QoP models allow calculation for different versions of the protocol which protects the transmitted data on different security levels. One can find only a couple of articles that documented QoP because this security topic is one of the latest approaches. S.Lindskog and E.Jonsson tried to extend security layers in a few Quality of Service (QoS) architectures [6]. Unfortunately, the descriptions of the methods are limited to the confidentiality of the data and based on different configurations of the cryptographic modules. C.S. Ong et al. in [8] present the QoP mechanisms, which define security levels depending on security parameters. These parameters are: a key length, the block length and contents of an encrypted block of data. P.Schneck and K.Schwan [7] proposed an adaptable protocol concentrating on the authentication. By means of this protocol, one can change the version of the authentication protocol which finally changes the parameters of the asymmetric and symmetric ciphers. Y.Sun and A.Kumar [9] created QoP models based on the vulnerability analysis which is represented by the attack trees. The leaves of the trees are described by means of the special metrics of security. These metrics are used for describing individual characteristics of the attack. Unfortunately, the majority of the QoP models can be recognized only for the three main security services: confidentiality, integrity and authentication. In article [5] B.Ksiezopolski and Z.Kotulski introduced mechanisms for adaptable security which can be used for all the security services. In Appendix A we briefly present the model which is introduced in [5].

The Quality of Protection models are complex and the protocol optimization takes a lot of time. The analysis of any type of the security protocol is difficult when the experts do not use visualization tools. In literature, we can find many programs which helped the experts analyze the protocols. As the example we can point to the AVISPA tool [10,11] or ProVerif [12] application which helps automated validation of security protocols.

In this article we present the Security Protocol Optimization Tool, (SPOT), application whose main function is the management of the security level of the exchanged data in the local network. This tool used the adaptable model [5] which introduces the Quality of Protection for all security services guaranteed in the cryptographic protocol. In this article the SPOT architecture and all included modules are presented.

2 SPOT: Security Protocol Optimization Tool

The main objective of the Security Protocol Optimization Tool (SPOT) is the management of the security level of the transmitted data in the organization. The SPOT was based on the adaptable model which was presented in the articles [5,3]. The main aim of the adaptable model is the calculation of the versions of the given protocol which realized its functionality on different levels of protection. The management system can switch between calculated versions of the protocol.

2.1 The Architecture of the SPOT

The SPOT architecture is presented in Fig. 1. The architecture is based on the four main modules: the presentation module, core module, optimization module and visualization module. The SPOT can work in two modes the *automatic mode* and the *user mode*. In Fig. 1 the automatic mode (white arrows) is separated from the user mode (black arrows) by the dashed line. In the automatic mode the SPOT is controlled by the configuration files where the details about cryptographic protocol and required quality of protection are defined. This mode is fully automatic and the results are generated without any interaction with the expert. The SPOT in this mode is the soft real-time system [13] so the SPOT response time is important but not critical for the system. In the user mode the SPOT can be configured by means of the graphic interface. This mode is not automatic and every operation must be defined manually by the expert. In the next sections we will present the SPOT in detail.

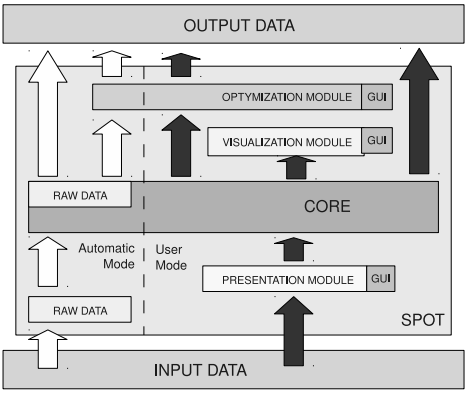


Fig. 1. The architecture of the SPOT

Presentation Module. The input information required for the adaptable model of security [5] is complex. In the SPOT application, one can use the presentation module which gives the possibility of defining the model by means of user friendly graphic interface. This module is available only in the user mode

because in the automatic mode there are no interactions with the user. In the automatic mode the input data to the core module have to be manually prepared by means of any text editor or by the external module responsible for it. The specific configuration can be generated by the presentation module and after that can be saved to the file and used in the automatic mode. All input and output data used in the SPOT application are formatted into the XML standard. Methodology of preparing the SPOT configuration according to the adaptable model consists of two stages: the configuration stage and the usage stage (Fig. 2).

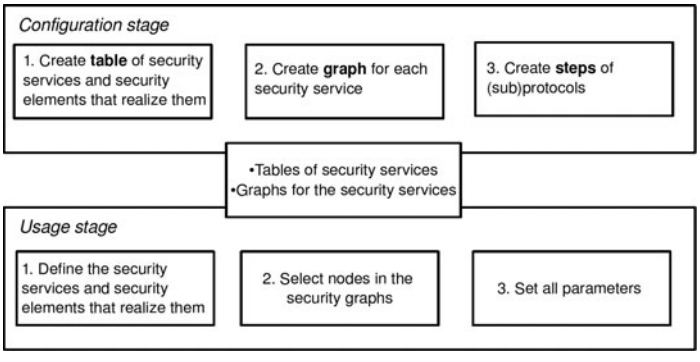


Fig. 2. The architecture of the presentation layer

Stage I: Configuration Stage. Configuration stage consists of three steps which are based on preparing the table of security mechanisms, graphs of security services and steps of the (sub)protocol. The theoretical description of these elements is described in the articles [5,3]. In the first step we create the table of the security mechanisms which can be used during the protection required in the protocol security services. Moreover, we add the impact of the particular security mechanisms for global protection of the service. The graphic interface which is used for creation of the table is shown in Fig. 3.

In the next step the user has to create the graphs of security services. These graphs define the possible versions of the protocol in detail. The graphs must be created for all required security services in the protocol. Additionally, the user must configure all nodes of the graph by defining its parameters. In the presentation module this configuration is created by means of the graphic interface which is presented in Fig. 4.

The last step in the configuration stage refers to defining the cryptographic protocol for which the user wants to apply the adaptable security. To achieve this goal the user has to split the protocol into the subprotocols and furthermore, in the subprotocols we can distinguish single actions. After the configuration stage the user can define the particular version of the given cryptographic protocol. The user chooses the security services which are required in the particular step

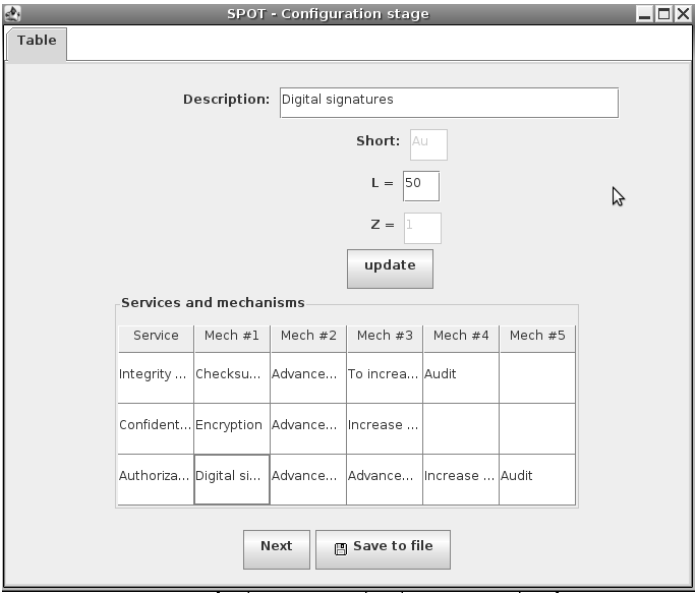


Fig. 3. The graphic interface for the creation of the table of security mechanisms

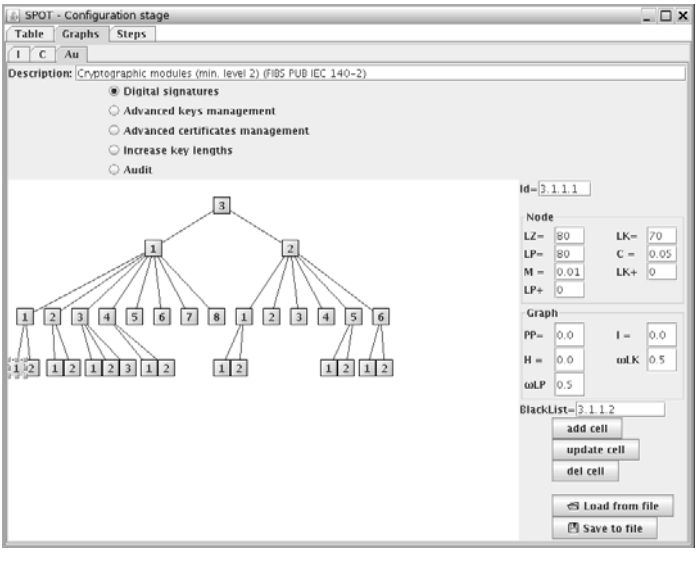


Fig. 4. The graphic interface for the creation of the security graphs

of the protocol and after that chooses the required security mechanisms which guarantee the chosen security services. In the presentation module these actions are carried out by means of graphic user interface (Fig. 5).

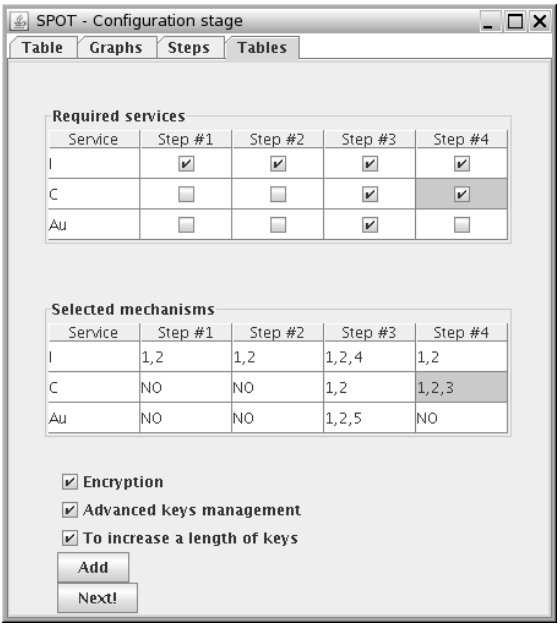


Fig. 5. Preparing a version of the protocol: defining security services and security mechanisms

All operations described in this section prepare the SPOT application for specific conditions used by the given protocol. These configurations will be used in the next stage.

Stage II: Usage Stage. In the first step in the usage stage the user defines the security services and the security mechanisms which accomplish them. In the second step the user determines the rest of the parameters needed in the adaptable model. These parameters are connected with the calculation of probability of incident occurrence and impact of successful attack. The most important factors the graph path represents is the specific realization of the security mechanisms. The graphic interface which is used for defining the graph path is presented in the Fig. 6.

Core Module. The most important module in the SPOT architecture is the Core module. This module is responsible for the calculation of the protection level, probability of incident occurrence, impact of successful attack and the global security level. The CORE module accepts the XML files as the format

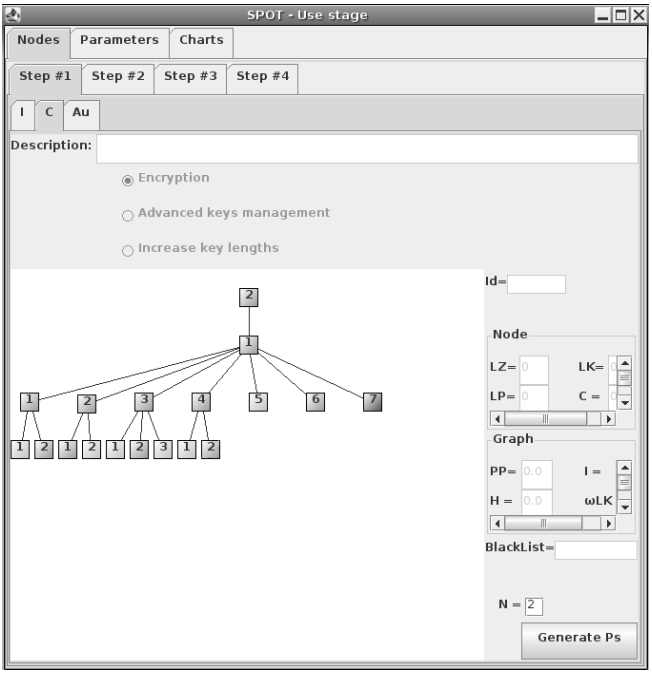


Fig. 6. Preparing a version of the protocol: defining graph path

of the input data. All calculated values in the layer are saved into the XML files. In both user and automatic modes the output data can be transmitted to the optimization module or if the output data do not need the optimization, it can be directly transported as the final output data. These output data include the instructions how to configure the system which accomplishes the cryptographic protocol. In the user mode the output data from the core module can be visualized in the visualization mode.

Visualization Module. The visualization module is responsible for the presentation of the output data from the CORE module in user's friendly ways. The user can compare different versions of the cryptographic protocol in separate tables.

Owing to that kind of presentation users can compare the results from particular factors of the model and they could be: protection level, probability of incident occurrence, impact of successful attack and global security level. This module is available only in the user's mode. The graphic interface is presented in Fig. 7.

For each computation of global security level, the SPOT automatically creates the appropriate charts and imposes them on the previous charts. It helps see how changing the parameters affects the results. We can easily compare the versions of the protocol.

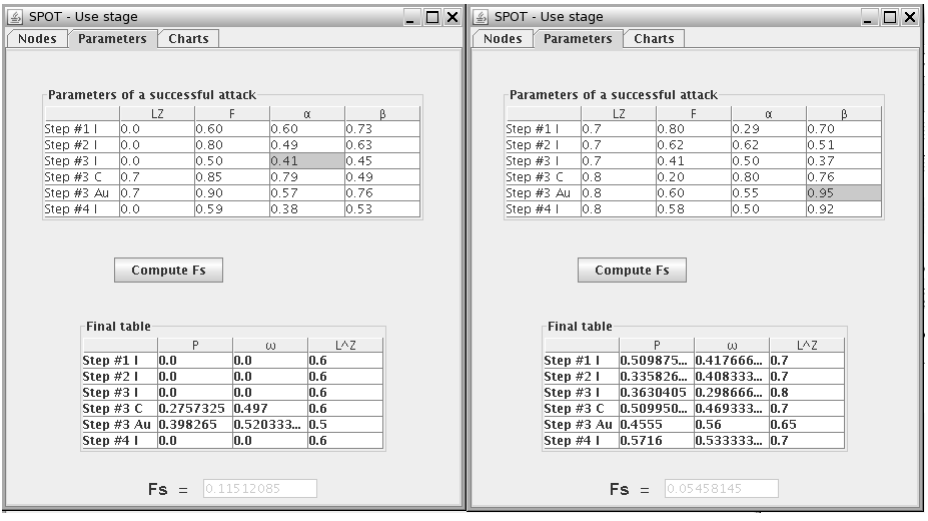


Fig. 7. Comparison of two versions of the protocol

Optimization Module. The output data from the CORE module can be optimized. The Core module makes the calculation according to the input data transmitted from the presentation module (user mode) or directly as raw data in the XML format (automatic mode).

The expert user which configures the versions of the given protocol can not take into consideration all possible versions of the protocol. The optimization module can calculate all possible versions of the protocol and find these which satisfy the defined conditions. In the user mode the optimization module is controlled by the graphic interface (Fig. 8). In the automatic mode the condition about the optimization is defined during preparation of the SPOT configuration file.

2.2 Qualitative Estimation

In the SPOT, the results are presented as quantitative estimation of security parameters and, finally, the global security level. Additionally, the application introduces a qualitative interpretation of the results. That kind of estimation is done for the most important parameters and, of course, the final result.

In the SPOT, we have 5 levels of valuation: very low, low, medium, high and very high. It is important that existing correlation between the quantitative and the qualitative results are not only of theoretical character but also of real one. Theoretically, the ranges of parameters are well defined in the model but these ranges are not permanent and they are different for a specific version of the analyzed protocol. In some situations, the protocol can not be proceeded in a version for which the probability of an incident occurrence is close to 0. A practical character of the qualitative estimation of the security level is obtained

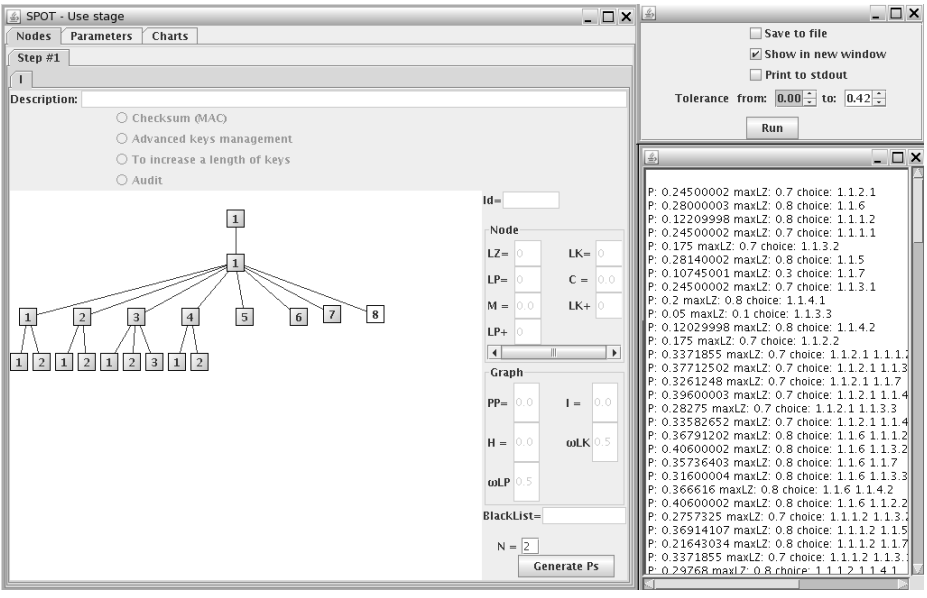


Fig. 8. Generating all available choices for this graph in this step

because the SPOT calculates the possible minimal and maximal values of the parameters for a version of the analyzed protocol. The ranges of parameters for the qualitative evaluation are calculated by the formulas presented below:

$$\begin{aligned} VeryLow &= \langle Q_{\min}, Q_{\min} + X\rangle \\ Low &= \langle Q_{\min} + X, Q_{\min} + 2X\rangle \\ Medium &= \langle Q_{\min} + 2X, Q_{\min} + 3X\rangle \\ High &= \langle Q_{\min} + 3X, Q_{\min} + 4X\rangle \\ VeryHigh &= \langle Q_{\min} + 4X, Q_{\min} + 5X\rangle \\ X &= \frac{Q_{\max} - Q_{\min}}{5} \end{aligned} \tag{1}$$

where:

$Q_{\max}$ – The maximum value for the parameter calculated for the specific version of the protocol,

$Q_{\min}$ – The minimum value for the parameter calculated for the specific version of the protocol.

2.3 Goals Obtained by the SPOT

1. Introducing the tool which will prepare the configuration of cryptographic protocol according to the specific requirements. The protocol configuration

can be prepared in an automatic way when no user interaction is needed. This goal gives the possibility to introduce the quality of protection in the soft real time systems.
2. Visualize the adaptable model of security [5]. The tool allows users to select interactively the data to be displayed in a friendly way. The adaptable model is complex so creating and analyzing protocols without a friendly tool is difficult and time-consuming.
3. Parallel comparing (with all features given by the model) versions of the protocol.
4. The ability to analyze the performance data in charts. It is very helpful to see how elements of the protocol behave in the current configuration of the protocol.
5. Assistance in planing, reviewing, maintaining and understanding logical structure of the protocol. We can study protocols and easily make changes in their configurations.
6. JAVA was used to implement the SPOT and we distribute this tool in one package, so it makes SPOT platform-independent and very portable.

2.4 Features of the SPOT

1. Getting the results is very fast and simple.
2. The SPOT provides visualization for the results. For each computation of global security level (F_s), the SPOT automatically creates appropriate charts and imposes them on the previous ones. It can help see how changing the parameters affects the results. We can easily compare the versions of the protocol.
3. After creating the logic of protocol by expert, he can write this to the file. It is very useful because configuration can be loaded from the file. A File's format is XML and this is a well known international standard. Owing to the usage of XML international standard, the output data can be easily used by other security application in a given architecture.

3 Conclusions

In this paper we have presented the Security Protocol Optimization Tool for the network adaptable security. The SPOT can determine the version of cryptographic protocols according to the specific requirements of the electronic processes. The application is based on the adaptable model [5] which introduces the quality of protection to the electronic process. One of the important features of the tool is its automatic mode which gives possibility to manage the protection level of the process in soft real time without the user interaction. Additionally, the SPOT visualizes the adaptable model [5] which is very helpful during detailed protocol analysis. The results of these analyses can be saved into the XML standard and after this the system can be controlled according to the determined data. By means of the SPOT application one can define different versions of the protocol which realize the process on different protection levels.

References

1. Merabti, M., Shi, Q., Oppliger, R.: Advanced security techniques for network protection. Computer Communications 23, 151–158 (2000)
2. Lambrinoudakis, C., Gritzalis, S., Dridi, F., Pernul, G.: Security requirements for e-government services: a methodological approach for developing a common PKI-based security policy. Computers & Security 26, 1873–1883 (2003)
3. Ksiezopolski, B., Kotulski, Z., Szalachowski, P.: Adaptive approach to network security. In: Kwiecień, A., Gaj, P., Stera, P. (eds.) 16th Conference of Computer Networks, CN 2009, Wisła, Poland. CCIS, vol. 39, pp. 233–241. Springer, Heidelberg (2009)
4. Szalachowski, P., Ksiezopolski, B., Kotulski, Z.: CMAC, CCM and GCM/GMAC: advanced modes of operation of symmetric block ciphers in the Wireless Sensor Networks. In: Elsevier: Information Processing Letters (accepted paper) (2010)
5. Ksiezopolski, B., Kotulski, Z.: Adaptable security mechanism for the dynamic environments. Computers & Security 26, 246–255 (2007)
6. Lindskog, S.: Modeling and Tuning Security from a Quality of Service Perspective. PhD dissertation, Department of Computer Science and Engineering, Chalmers University of Technology, Goteborg, Sweden (2005)
7. Schneck, P., Schwan, K.: Authenticast: An Adaptive Protocol for High-Performance, Secure Network Applications, Technical Report GIT-CC-97-22 (1997)
8. Ong, C.S., Nahrstedt, K., Yuan, W.: Quality of protection for mobile applications. In: IEEE International Conference on Multimedia & Expo., pp. 137–140 (2003)
9. Sun, Y., Kumar, A.: Quality od Protection(QoP): A quantitative methodology to grade security services. In: 28th Confrence on Distributed Computing Systems Workshop, pp. 394–399 (2008)
10. Blanco, V., Gonzalez, P., Cabaleiro, J.C., Heras, D.B., Pena, T.F., Pombo, J.J., Rivera, F.F.: AVISPA: visualizing the performance prediction of parallel iterative solvers. Future Generation Computer Systems 19, 721–733 (2003)
11. Vigano, L.: Automated Security Protocol Analysis With the AVISPA Tool. Electronic Notes in Theoretical Computer Science 115, 61–86 (2006)
12. Blanchet, B., Chaudhuri, A.: Automated Formal Analysis of a Protocol for Secure File Sharing on Untrusted Storage. In: Proceedings of the 29th IEEE Symposium on Security and Privacy, pp. 417–431 (2008)
13. Stankovic, J.A.: Real-Time Computing, University of Massachusetts (1992)

On Authentication Method Impact upon Data Sampling Delay in Wireless Sensor Networks

Pawel Szalachowski[1], Bogdan Ksiezopolski[1], and Zbigniew Kotulski[2,3]

[1] Institute of Computer Science, Maria Curie-Sklodowska University,
pl. M. Curie-Sklodowskiej 5, 20-031 Lublin, Poland
[2] Institute of Fundamental Technological Research of PAS,
Pawinskiego 5B, 02-106 Warsaw, Poland
[3] Institute of Telecommunications of WUT
Nowowiejska 15/19, 00-665 Warsaw, Poland

Abstract. Traffic in Wireless Sensor Network (WSN) consists of short packets sent by nodes that are usually identical in respect of software applied and their hardware architecture. In such a communication environment it is important to guarantee authentication of the nodes. The most popular way to achieve this basic security service is using Message Authentication Code (MAC). The sensor node's harbware is very limited so the cryptography used must be very efficient. In the article we focus on the influence of the authentication method's performance on delays in data sampling by the sensor nodes. We present efficiency results for MACs generation in the node. We compare the results for approved, standardized and commonly-used schemes: CMAC, GMAC and HMAC based on MD5 and SHA-1. Additionally, we compare the obtained results with the performance of PKC-based authentication method using the ECDSA.

Keywords: wireless sensor network, cryptographic protocols, block cipher modes, data sampling, sensor's security, utilization efficiency.

1 Introduction

The main task of WSN is to collect informations from surroundings and relay them to base station where are analyzed. All communication is realized by radio frequency and it causes many vulnerabilities. Attack on WSN is much simpler than wired network. Some classes of attacks are easy to perform and very malicious. These classes are generally related to possibility of eavesdropping, unauthorized modifying and injecting data into network. Such attacks are: Sinkhole attack, Sybil attack [1], Wormholes, Flood attacks and other presented in [2]. We must prevent against these attacks but we are limited by next disadvantage of the nodes: hardware limitations. Another aspect which is assumed by WSN paradigm is autonomy of network. To achieve that, network must be reliable, with self-organizing [3] and self-healing properties. Traffic in network must be always authenticated and sometimes confidential.

A. Kwiecień, P. Gaj, and P. Stera (Eds.): CN 2010, CCIS 79, pp. 280–289, 2010.

All these security advantages without cryptography are impossible to obtain (in standard WSN environment). Cryptographic primitives and protocols involve providing security services to system. The goal of these services is to protect network against various types of attacks. As most important security services we can specify:

- Authentication,
- Confidentiality,
- Data Integrity.

Authentication addresses a problem of identity. Parts of communication often need to know real identity of each other. Confidentiality provides privacy, in insecure channel, for authenticated participant of given protocol. Data Integrity prevents to unauthorized manipulation of data.

Some type of applications which are used for example in the seismic analysis, industrial vibration monitoring or HVAC control require high-performance data acquisition. In many of these applications the data authentication is crucial. The guaranteeing of this service is realized by means of cryptography. Unfortunately, the cryptographic operations take the time and the data must be transmitted to other node before the next data will be collected. When one used not efficient authentication methods then the data sampling delays can took place.

In this paper we take into consideration Authentication and Data Integrity. Security mechanisms, which realize these services are approved, standardized and commonly-used. We consider these schemes in terms of efficiency on WSN node.

This paper is organized as follows: In Section 2 we present a motivation of need of Authentication and realization of it in WSN. In Section 3 we describe the characterization of data sampling process. Section 4 presents approaches to Message Authentication Code (MAC) and ECDSA scheme which we decided to evaluate. Next, in Sect. 5 we present results for chosen authentication methods, details of measurements and impact of these methods upon data sampling delay. And the last section is for comments and conclusions.

2 Authentication in WSN

The authentication is a basis for robust and reliable network. A node must rely on other nodes when it sends, receives or passes on packets. It is especially important with case of routing and forming messages because these informations are crucial for many sensors in WSN. Authentication can be realized by variety of methods but in case of WSN it is required that these methods must be very fast and memory efficient.

There are many authentication approaches dedicated for WSN. In [4] Security Protocols for Sensor Networks (SPINS) is presented. It provides authentication, confidentiality, data freshness and authenticated broadcast by two subprotocols. Next example is TinySec [5]. These schemes uses MAC for authentication and don't need Public-Key Cryptography (PKC). But in this class of solutions, key

management scheme [6] is essential. Due to shared keys, each party of secure communication needs at least one common key. In the case of a large and often changing network, it can cause big memory overhead.

PKC would be very helpful approach in such dynamic environments. But PKC realization on sensor node, without special hardware modules are not efficient. Standard mechanisms like RSA are too slow and inappropriate. Researchers try to achieve the same properties by more efficient methods and noteworthy technique is Elliptic Curve Cryptography (ECC) [7]. In [8,9] comparison, on sensor node, of RSA [10] and ECC is presented and ECC solution is shown as promising one. So in this paper we compare ECDSA (based on ECC) as efficient representative of the digital signature scheme with popular MAC codes.

3 Data Sampling

The sensor collect the data in different frequencies. The WSN applications which require high-performance data sampling need special data acquisition card. In the Table 1 we present the data sampling frequencies of the selected cards produced by the Xbow [11]. We choose the card produced by Xbow because in the article we are checking the performance of the authentication methods by means of sensor produced by the same corporation.

Table 1. The characteristic of data acquisition card

Data acquisition card	Max. Frequency [Hz]	Min. period [ms]
MTS420/400	50	20
MDA320CA	200	5

The maximum collection frequency of the MDA320CA card is 200 Hz and it means that every 5 ms the data is collected from the sensor. Another presented in the Table 1 card is not as much precise as MDA320CA and collected the data every 20 ms.

Another issue should be considerate in case of the delaying of the collected data. After the acquisition of the data, the sensor node have to relay it to other sensor node or to base station. The data transmission is one of the most time consuming operation. In the Table 2 we present the maximum data rate of the radio of the motes produced by Xbow [11].

Table 2. Radio transmission time of the analyzed motes

Mote	Max. Data Rate [kbits/sec]	Min. transmission time of			
		128 b	256 b	384 b	512 b
IRIS/MICAz/MICA2/Imote2	250	0.5 ms	1 ms	1.5 ms	2 ms
MICA2DOT	38.4	3.3 ms	6.7 ms	10 ms	13.3 ms

The sensors which have maximum data rate equal to 250 kbits/sec are able to transmit 256 bits of data every 1 ms. If we compare it with the data sampling frequency (max. every 5 ms) we can say that every measured data can be transmitted without the delays. In this analysis we omit the time which is required for the cryptographic operation realized security requirements. In this article, we would analyze the data sampling Malays when the data authentication will be guaranteed.

The MICA2DOT sensor have maximum data rate equal to 38.4 kbits/sec. The time which is needed for transmitting of 256 bits of data is 6.7 ms. We can say that if this sensor will used the MDA320CA board then the collected data will have delays because this board have maximum collection time equal to 5 ms. In this situation acquisition card could not work with maximum frequency.

4 Message Authentication Methods

4.1 Message Authentication Codes

Message Authentication Code is a short tag (checksum) used to authenticate a message. It is produced by MAC scheme and it provide both, authenticity and data integrity of a message. MAC links message with its originator. Except message, MAC algorithm needs key, which is shared between verified peers. MACs are one of the most popular security mechanisms, and are used in many security protocols in WSN [4,5]. We can define four types of MACs [12], which we briefly describe below.

1. Unconditionally secure MAC based on encryption with a one-time pad or one-time secret key. Stinson and Simmons in [13] present this scheme. But in WSN applications it seems to be useless because of redundancy, memory restriction and pads management problem.
2. Keyed-Hash Message Authentication Code (HMAC) is popular approach which uses cryptographic hash function in combination with a secret key to produce tag.
3. MAC based on stream ciphers is next class of algorithms. These proposals are presented in [14,15,16] and they are promising ones for low powered systems. The reason is efficiency of some stream ciphers. These schemes often use Linear Feedback Shift Register (LFSR) for MAC generation.
4. The last type is MAC based on block ciphers. It takes advantage of encryption procedure of block cipher to generate checksum. It is commonly-used and efficient way to realize MAC. We describe below and evaluate in paper representative (CMAC) of this method.

There are many other interesting constructions, e.g. UMAC [17] proposed by Black. It uses new universal hash function with block cipher or cryptographic hash function (selection is made by the user). In this paper we consider solutions from 2 to 4. points as most popular. In the next subsections we describe CMAC, GMAC and HMAC schemes.

CMAC. Cipher-based MAC (CMAC) is mode for authentication. It is described in [18]. Especially this mode can be used when we want to use approved block cipher instead of approved hash function to generate message authentication code (MAC). This approach is very useful in WSN because the sensor node does not need hash function if block cipher is available. It can save memory significantly. CMAC mode generates tag up to 128 bits.

GMAC. Next solution called GMAC is correctly Galois/Counter Mode (GCM) [19]. GCM mode let us to authenticate data or authenticate and encrypt data simultaneously. Usage this mode without encryption is GMAC. GCM/GMAC is specified by NIST and it is used in many standards (e.g. SSH, IETF IPsec). This mode requires 128-bit block cipher and CMAC generates tag up to 128 bits.

GMAC has a few interesting features, which make it noteworthy in WSN applications.

- The length of data to authenticate is no required in advance.
- This mode can perform authentication or authenticate-encryption on message.
- Both hardware and software high-throughput implementations of these mode are possible. Software optimization is realized by lookup table which is precomputed from the key and is stored in memory.
- The authentication of the protected data can be verified independently from the recovery of the confidential data from its encrypted form.

HMAC. HMAC is described in [20], it is also standardized by NIST. As mentioned above, HMAC uses cryptographic hash function with secret key to achieve authentication code. The cryptographic strength of the HMAC depends upon the strength of cryptographic hash function and the length of the secret key that is in used. Size of the secret key should be at least like size of the tag generated by hash function.

Decision about usage of hash function is crucial. We decided to compare HMAC based on popular and widespread cryptographic hash functions. First choice is MD5 [21] and second is SHA1 [22]. These function are supersede by their more secure substitutes but still are used in many critical applications and in our opinion it is good compromise between security and efficiency.

4.2 ECC and ECDSA

ECC [7] is efficient way to provide PKC. It is based on the algebraic structure of Elliptic Curves over finite fields. In cryptography we used to use $F(2^m)$ and $F(p)$ fields. p is the large prime number. Key operation for ECC is defined over these finite fields. It is point multiplication. For detailed description of ECC we refer to literature [7].

Standard solution like RSA [10] is resoluteness inappropriate [9] for sensor node. ECC provides the same functionality but in the efficient way. The Elliptic Curve Digital Signature Algorithm (ECDSA) is a variant of the digital signature algorithm. It is based on ECC so it is more efficient than standard equivalent.

In terms of very constrained hardware we still must accelerate the process of the message authentication. There is many ways to optimize ECC operations. Performances compared in this paper based on TinyECC [23] results. This implementation use Berret Reduction, Sliding Window, Shamir's Trick [7] and others tricks to improve ECC/ECDSA efficiency.

5 Schemes Performance

In this section we show efficiency results for MACs schemes which we decided to compare. We choose: four variants of GMAC mode, CMAC mode, HMAC based on SHA1 and HMAC based on MD5. All these methods are well-known, standardized and commonly used. We compare the results of execution of authors implementation of the algorithms. We do not consider security strength of the mechanisms, we evaluate only performance and usefulness in sensor network environments.

5.1 Implementations and Measurements

As approved block cipher we use Advanced Encryption Standard (AES) [24] with 128-bit block size and 128-bit key. For the implementation of the AES, SHA1 and MD5 algorithms we used the libraries from the DAS Labor [25]. The ECDSA algorithm was implemented in TinyECC [23]. Other algorithms used in the paper was implemented by the authors. All of the codes, except for the AES, SHA1 and MD5 codes are written in C. For compilation we use the AVR-GCC compiler (using `-Os` flag for size optimization). AES, SHA1 and MD5 are the most important part of the cryptosystem and were written in Assembler, to improve speed of execution. As a platform for testing we have chosen XBow IRIS Mote [11]. It is equipped with ATmega1281 Microcontroller with 8 MHz clock speed, 8 KB of RAM and 128 KB of Flash. ATmega1281 has 32×8-bit general purpose registers with a single clock cycle access time. It allows the execution of single-cycle Arithmetic Logic Unit (ALU) operations. On this platform we have run modified LiteOS [26] to measure the efficiency of codes. We run the program with the absolute priority and we count the CPU cycles.

ECDSA implementation is imported from TinyECC [23] and it is written in nesC. Crucial fragments of code, for platform speedup, are written in Assembler. This solution assumes use of the secp160r1 parameter, which is elliptic curve domain parameter over $F(p)$ and it is defined in [27]. We also use all available optimizations to accelerate ECDSA executions.

5.2 Results

Performance results for the MACs and ECDSA schemes are shown in Table 3. Efficiency of GCM/GMAC can be improved by lookup tables that are loaded into the code. In the tables GCM and GMAC denote the modes without acceleration. Additionally, we decided to use the lookup tables of the sizes: 256 B, 4 KB and

Table 3. Performance of authentication methods

Mode	Code size	Init	Message size (in bits)			
			128	256	384	512
CMAC	2240 B	0.7 ms	0.4 ms	0.7 ms	1.0 ms	1.4 ms
GMAC	5706 B	2.6 ms	1.7 ms	2.5 ms	3.2 ms	4.0 ms
GMAC-256 B	6220 B	3.1 ms	1.2 ms	1.7 ms	2.3 ms	2.8 ms
GMAC-4 KB	10271 B	7.2 ms	0.7 ms	1.0 ms	1.3 ms	1.6 ms
GMAC-8 KB	14108 B	25.1 ms	0.5 ms	0.7 ms	0.9 ms	1.2 ms
HMAC-SHA1	5252 B	0.0 ms	4.7 ms	4.7 ms	4.8 ms	4.8 ms
HMAC-MD5	6348 B	0.0 ms	3.6 ms	3.6 ms	3.7 ms	3.7 ms
ECDSA sign	19308 B	3493.4 ms	2001.6 ms	2001.6 ms	2001.6 ms	2001.6 ms
ECDSA verify	19308 B	3493 ms	2436.5 ms	2436.5 ms	2436.5 ms	2436.5 ms

8 KB. GCM-XB and GMAC-XB denote the modes with acceleration using the X bytes lookup table.

Size column describes how much memory a code needs. Depending on the case, it includes all auxiliary tables. Initialization phase and other essential functions are required for all additional codes.

The *Init* results show how much time is required for all preparatory operations needed for running a proper mode. It includes keys length checking, buffers erasing, data validation, keys and IVs setup, counters generation and acceleration tables generation (for the GMAC modes). It is expressed in the milliseconds of CPU time, and it includes all required operations to start MAC generation or ECDSA execution. Also all remaining results are expressed in milliseconds.

We tested presented schemes on messages with fixed sizes. These sizes are: 128 bits, 256 bits, 384 bits, 512 bits, and in our opinion messages with these lengths are characteristic of WSN packet traffic.

Only GMAC-8KB is faster than CMAC, but it needs many more memory and time for initialization. CMAC and GMAC-4KB have similar results, but in terms of size CMAC is the best. GMAC without acceleration is relatively slow solution. MACs based on cryptographic hash functions are inefficient as well. CPU usage overhead in case of short messages is significant. It is caused by block size of SHA1 and MD5 which is 512 bits and using these HMAC is not scalable for short packets. CMAC mode is very interesting, because of memory and CPU time consumption. In this comparison ECDSA is very slow, and we must remember that this is one of the fastest PKC solutions. In case of short message authentication is very inefficient approach.

5.3 Data Sampling Delays

In this article we focus on the authentication methods and its influence on data sampling delays in Wireless Sensor Networks. In our analysis we take into consideration only the most time consuming operation and there are cryptographic operations and transmission of the data. The PKC solution is very not efficient and cause large delays. This method can not be used with high-performance

data acquisition card. In case of using the data acquisition card with frequency equal to 50 Hz (the data sampling is in every 20 ms) all analyzed authentication method can be used for two types of analyzed sensor radio. In these situation no delays would take place.

When we analyzed the high-performance data acquisition card with frequency equal to 200 Hz (the data sampling is in every 5 ms) then we have to choose the authentication method carefully. The data delays for the sensor radio equal to 250 kbits/sec are presented in the Table 4 and for sensor radio equal to 38.4 kbits/sec in the Table 5. If we consider the sensor radio with data rate equal to 250 kbits/sec then we can choose CMAC and GMAC with lookup table for every analyzed message size. We can use without delay the HMAC-MD5 method for 128 b and 256 b message. The GMAC without lookup table acceleration can be use without delays for 128 b, 256 b and 384 b. The HMAC-SHA1 method will provide the delays for any message size. If we consider the sensor radio with data rate equal to 38.4 kbits/sec then we can collect the data without the delays only for 128 b message. In this case we can use CMAC and any variation of GMAC. The HMAC methods can not be used without data sampling delays. In this case the main problem refers to the radio efficiency.

Table 4. The data sampling delays for data acquisition card with frequency equal to 200 Hz and maximum radio rate 250 kbits/sec

Mode	Message size (in bits)			
	128	256	384	512
CMAC	no delay	no delay	no delay	no delay
GMAC	no delay	no delay	no delay	1.0 ms
GMAC-256B	no delay	no delay	no delay	no delay
GMAC-4KB	no delay	no delay	no delay	no delay
GMAC-8KB	no delay	no delay	no delay	no delay
HMAC-SHA1	0.2 ms	0.7 ms	1.3 ms	1.8 ms
HMAC-MD5	no delay	no delay	0.2 ms	0.7 ms

Table 5. The data sampling delays for data acquisition card with frequency equal to 200 Hz and maximum radio rate 38.4 kbits/sec

Mode	Message size (in bits)			
	128	256	384	512
CMAC	no delay	2.4 ms	6 ms	9.7 ms
GMAC	no delay	4.2 ms	8.2 ms	12.3 ms
GMAC-256B	no delay	3.4 ms	7.3 ms	11.1 ms
GMAC-4KB	no delay	2.7 ms	6.3 ms	9.9 ms
GMAC-8KB	no delay	2 ms	5.9 ms	9.5 ms
HMAC-SHA1	3 ms	6.4 ms	9.8 ms	13.1 ms
HMAC-MD5	1.9 ms	5.3 ms	8.7 ms	12 ms

6 Conclusions

Choice of the authentication scheme depends on many factors. But in general usage we recommend CMAC mode. It is efficient, and it include encryption procedure, that can be additionally used to standard encryption. GMAC-8KB is also interesting on condition that memory for acceleration table is not problem. Advantages of this solution are speed of the MAC generation and easily with Authenticate-Encryption realization.

Based on presented results we can estimate performance of the other schemes. For example CMAC has very similar construction to CBC-MAC or OMAC [28] and results should be very similar too. Notice that even usage of very efficient PKC like ECDSA for short message authentication is inefficient. Of course in some situations (very dynamic network) it is great solution, but in our terms and in comparison with MACs, ECDSA is unacceptable.

The data sampling delays can be real problem in the sensors with high-performance data acquisition card. When the WSN applications would require the data sampling with the frequency equal to 250 Hz and the radio rate equal to 250 kbits/sec then we recommended CMAC method or GMAC with lookup tables acceleration. In this scenario the HMAC method is not efficient and should not be used.

In this article we study the sensor with limited radio rate, as well. That kind of sensor configuration is not efficient and should not be used in the WSN Application when the high-performance data sampling is required.

References

1. Douceur, J., Donath, J.S.: The sybil attack, pp. 251–260 (2002)
2. Karlof, C., Wagner, D.: Secure routing in wireless sensor networks: Attacks and countermeasures. In: First IEEE International Workshop on Sensor Network Protocols and Applications, pp. 113–127 (2002)
3. Sohrabi, K., Gao, J., Ailawadhi, V., Pottie, G.J.: Protocols for self-organization of a wireless sensor network. IEEE Personal Communications 7, 16–27 (2000)
4. Perrig, A., Szewczyk, R., Tygar, J.D., Wen, V., Culler, D.E.: Spins: security protocols for sensor networks. Wirel. Netw. 8(5), 521–534 (2002)
5. Karlof, C., Sastry, N., Wagner, D.: Tinysec: a link layer security architecture for wireless sensor networks. In: SenSys 2004: Proceedings of the 2nd international conference on Embedded networked sensor systems, pp. 162–175. ACM, New York (2004)
6. Xiao, Y., Rayi, V.K., Sun, B., Du, X., Hu, F., Galloway, M.: A survey of key management schemes in wireless sensor networks. Comput. Commun. 30(11-12), 2314–2341 (2007)
7. Hankerson, D., Menezes, A.J., Vanstone, S.: Guide to Elliptic Curve Cryptography. Springer, New York (2003)
8. Wander, A.S., Gura, N., Eberle, H., Gupta, V., Shantz, S.C.: Energy analysis of public-key cryptography for wireless sensor networks. In: PERCOM 2005: Proceedings of the Third IEEE International Conference on Pervasive Computing and Communications, Washington, DC, USA, pp. 324–328. IEEE Computer Society, Los Alamitos (2005)

9. Gura, N., Patel, A., Wander, A., Eberle, H., Shantz, S.C.: Comparing elliptic curve cryptography and RSA on 8-bit cPUs. In: Joye, M., Quisquater, J.-J. (eds.) CHES 2004. LNCS, vol. 3156, pp. 119–132. Springer, Heidelberg (2004)
10. Rivest, R.L., Shamir, A., Adleman, L.: A method for obtaining digital signatures and public-key cryptosystems. ACM Commun. 21(2), 120–126 (1978)
11. Iris datasheet, http://www.xbow.com/
12. Laboratories, R.: Pkcs 6: Extended-certificate syntax standard (1993)
13. Stinson, D.R.: Cryptography: Theory and Practice. CRC Press, Inc., Boca Raton (1995)
14. Lai, X., Rueppel, R.A., Woollven, J.: A fast cryptographic checksum algorithm based on stream ciphers. In: ASIACRYPT 1992: Proceedings of the Workshop on the Theory and Application of Cryptographic Techniques, London, UK, pp. 339–348. Springer, Heidelberg (1993)
15. Lim, S.Y., Pu, C.C., Lim, H.T., Lee, H.J.: Dragon-mac: Securing wireless sensor networks with authenticated encryption
16. Zoltak, B.: Tail-mac: A message authentication scheme for stream ciphers (2004)
17. Black, J., Halevi, S., Krawczyk, H., Krovetz, T., Rogaway, P.: Umac: Fast and secure message authentication (1999)
18. NIST – Current modes, http://csrc.nist.gov/groups/ST/toolkit/BCM/current_modes.html
19. McGrew, D.A., Viega, J.: The galois/counter mode of operation. gcm (2004)
20. Krawczyk, H., Bellare, M., Canetti, R.: Hmac: keyed-hashing for message authentication. RFC 2104, 1–12 (1997)
21. Rivest, R.: The md5 message-digest algorithm (1992)
22. Eastlake, 3rd, D., Jones, P.: Us secure hash algorithm 1, sha1 (2001)
23. Liu, A., Ning, P.: Tinyecc: A configurable library for elliptic curve cryptography in wireless sensor networks. In: IPSN 2008: Proceedings of the 7th international conference on Information processing in sensor networks, Washington, DC, USA, pp. 245–256. IEEE Computer Society Press, Los Alamitos (2008)
24. Daemen, J., Rijmen, V.: The Design of Rijndael. Springer, New York (2002)
25. Das labor page: https://das-labor.org/svn/microcontroller-2/crypto-lib/
26. Cao, Q., Abdelzaher, T., Stankovic, J., He, T.: The liteos operating system: Towards unix-like abstractions for wireless sensor networks. In: IPSN 2008: Proceedings of the 7th international conference on Information processing in sensor networks, Washington, DC, USA, pp. 233–244. IEEE Computer Society Press, Los Alamitos (2008)
27. Research, C.: Sec 2: Recommended elliptic curve domain parameters. Standards for efficient cryptography version 1.0 (2000)
28. Iwata, T., Kurosawa, K.: OMAC: One-key CBC MAC. In: Johansson, T. (ed.) FSE 2003. LNCS, vol. 2887, pp. 137–161. Springer, Heidelberg (2003)

Company's Data Security – Case Study

Piotr Stera

Silesian University of Technology, Institute of Computer Science,
Akademicka 16, 44-100 Gliwice, Poland
piotr.stera@polsl.pl
http://www.polsl.pl

Abstract. This paper describes a computer network and data security problems in an existing company. Two main issues were pointed out: data loss protection and uncontrolled data copying. Security system was designed and implemented. The system consists of many dedicated programs. This system protect from data loss and detected unauthorized file copying from company's server by a dishonest employee.

Keywords: LAN security, file server, abuse detect system, firewall, data security.

1 Introduction

This paper describes a situation in an existing company. The company makes projects and produces devices for other factories. There are about 30 employees in the company. They can be divided into several groups based on their tasks. Company's staff use computers to generate and change a lot of data (projects, documents, etc.). Before introducing security policies described below, employees saved their data on local hard disks installed on their workstations. This data was accessible for all staff via computer network through a share disk mechanism existing in Microsoft Windows system (without passwords).

The company experienced following problems:

- damaged hard disk on workstation caused some project data were lost,
- a competition company tried corrupting one of employees to copy data from company's computers and deliver them to the competition company.

Because of these facts company decided to change data organization and access.

2 Solutions

Described here solutions are not one homogeneous idea. They were added step by step. Following elements were added to existing ones according to changing security policy in company. The additional requirement was always not to stop work process in company. Because of it introduced solutions are not optimal and some of them are non-typical.

A. Kwiecień, P. Gaj, and P. Stera (Eds.): CN 2010, CCIS 79, pp. 290–296, 2010.

2.1 Data Loss Protection

File Server. Company wanted to control all their data and decided to install a central file server. All data was moved from workstations to this computer. The file server runs Linux operating system, because of cost, elastic solutions and ability to make changes in the system. Files for MS Windows workstations are accessible by Samba service installed on the server – this is a network interface for SMB Protocol using by MS Windows systems. There are many versions of MS Windows that are installed on workstations and it is not possible to create a MS Windows domain – files are accessible with a share disk mechanism. Server is protected against power disturb with UPS system.

Mirror Server. Damage to the server can stop work flow in the company for a long time. A second server, called mirror server, was installed to protect against such situations: data created on a hard disk of the first server is being copied to the second server. In emergency situation the mirror server can replace the first one and normal operation can be continued. The mirror server was installed in a separate room, far from the first server. This increases survival chances of at least one of the servers in case of fire or other physical danger. The mirror server is protected against power failure by additional UPS system.

Requirement to install mirror server was defined when the first server was in work process and it was impossible to do many changes to its configuration. It was considered DRBD mirror system (Distributed Replicated Block Device) [1], but it required too many changes in server configuration (kernel version, disk partitions etc.).

The second solution is a script, which regularly copies changed files to mirror server via SSH. It was implemented and used for some time, but it was too slow for a lot of large data files. Finally, SSH was changed to NFS (Network File System), which works much faster, but without data encryption. In this case encryption is not needed, because servers have direct connection via separated network. More information about it is in [2].

Network Trash. Sometimes, an employee removes a file from the server and after some time he wants to recover it. *Samba* does not provide such features. Network trash was constructed using mirror server: when employee is removing file from the first server, information about this operation is being sent to mirror server. Then, on the mirror server, the file is moved to a special folder – network trash. If the removed file is necessary, administrator can recover it form this folder. This feature was added to mirror service script. After fixed time unnecessary files are removed from trash.

Backup. There is a backup system except the mirror server. Administrator regularly copies data to external hard disk and once year to DVDs.

Mail. Company's network is connected to Internet via telephone line. This connection does not ensure constant IP address. Company has an email address

well known for its business partners. Its mailbox is outside company's network, on system administrated by external provider. Only several persons have access to the company's mailbox. Before changes, they read and delivered mails to proper receiver (employee). Such procedure was very time-consuming, but all emails was archived in one place. After analysis of the situation and considering the mailbox provider's offer following changes were introduced. Present mail system is shown in Fig. 1.

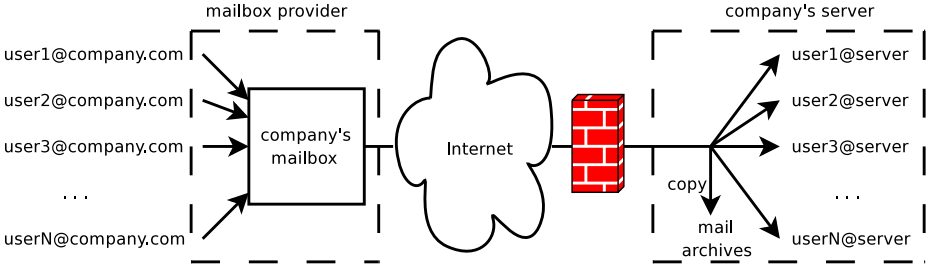


Fig. 1. Mail system

Mailbox provider's system holds one company mailbox, like before. Employees have aliases to the mailbox. On the file server mail server *Sendmail* [3]) and mailboxes for all employees are installed. Company's file server gets emails from company mailbox and puts them into employees' mailboxes on file server according to field *To:* in emails (this field contains employees' aliases [4]). This task is realized by a script for *procmail* (mail processor). Employees regularly read emails from their mailboxes on file server.

Sending emails is realized in a similar manner: employees send theirs mails to file server, which delivers them via company's mailbox. File server makes a copy of every email and puts it into archives. This task is realized by *Synonym* script for *Sendmail* [5,6]. This way all company's emails are archived in one place.

Warnings. There is no person in company, who continually monitor file servers. This is the primary reason why a specialized set of scripts has been written. They monitor file servers and warn the administrator in situations, which can disturb proper work of the file server (e.g. decreasing free space on hard disk below warning limit). The first channel of communication with the computer administrator is through audio – server plays recorded warnings via speakers. In this way, if necessary, servers remind the administrator to do a backup.

2.2 Protection against Uncontrolled Data Coping

Following channels of uncontrolled data copying were identified:

- theft physical medium with data,
- data copying via Internet,
- data copying by employees.

Physical Data Medium Protection. Servers and backup copies of data are located in protected places. Independent of that, data on disks are encrypted – this protect data before being read without digital key and password. To realize this task *DM-Crypt* and *LUKS* (Linux Unified Key Setup) were used [7,8]. The choice was natural – they are included in installed Linux distribution and integrated with the system, other programs (e.g. *Truecrypt* [9]) needs to be compiled and installation of additional libraries or Linux kernel is also required.

Data Coping via Internet. The structure of the company's network is shown in Fig. 2.

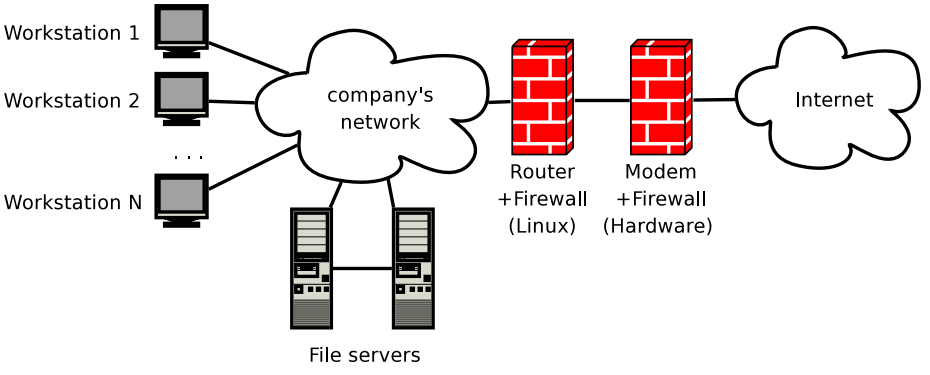


Fig. 2. Structure of the company's network

There are two firewalls (basic access control system at the network layer [10]) between Internet and company's network. There are two devices:

- modem including plain firewall realizes connection to Internet provider; the firewall is not sophisticated enough to realize all needed functions;
- router is a computer based on Linux, which realizes router and firewall tasks (it uses *iptables* system).

Firewalls' configuration blocks all communication between servers and Internet. It is only one exception: server is able to connect to company's mailbox on the external system. Server has its own firewall. To transmit data from or to the server, computer has to be connected to company's network and be registered on the server.

Data Access Inside the Company. The base security policy are defines access rules to data on the file server. Employees were divided into several groups based on their tasks. For each group access rights on the file server system were defined. It is impossible to block access to all, because employees need access to file server resources and data to do their job. They often need access to previous projects and elements designed by other groups to adapt own projects to rest elements of whole end firm product. Finally, employee has access to a lot of data on the file server.

Protection Against Dishonest Employee. Firewalls and server's access rules do not protect from data copying by employees. Dishonest or unreliable employee is able to copy data from file server and deliver it to unauthorized persons [11]. The second undesirable situation is, when unknown program, without user's knowledge try copying data outside workstation. Protection against described situations is far from simple – employees need access to the Internet and ports on their workstations to do their job. To decrease danger there were designed and implemented some mechanisms.

The first protection consists of a program written by this paper's author and installed on the router. It controls how much data is being sent outside company's network from every workstation. If a given computer is sending more data than a specified limit, then the firewall on router blocks suspicious connection. Limits can be specified as a sum for all connections from workstation or individually for every network connection. Limits set on the router prevent copying of large quantities of data to an outside entity. (potentially undesirable data copying), in a way guaranteeing undesturbed work of the employees. (e.g. finding needed information on the Internet). Configuration of these limits is based on logged statistic. Algorithm of this program regards the bursty nature of packet data: it allows for short transfer over limit (on credit). If in next measured period workstation is not sending data then credit is decremented otherwise credit is incremented. When credit is reaching credit limit then connection is blocked. This mechanism is not a absolute protection, but it is able to prevent from copying a lot of company's data to the Internet suddenly, before anyone is able to react.

Additionally, *Squidguard* – a program for control of WWW connections based on the *Sqiud* [12,13] was installed. It enables blocking of selected WWW addresses. It is a very weak protection. It was company's request to install such mechanism. It can be difficult for inadvanced educated employees in network function (part of described company's staff).

The next protection mechanism is logging of all read operations from the file server. Based on log files, it is possibile to detect employees' abuse. This paper's author has written a set of scripts realizing this task. Plain logging does not always correspond to real user actions – many employees use user-friendly interfaces on their workstations. Such interfaces read many needless files, e.g. some of them read file headers to show properly icon on user graphical desktop. Such mechanism generates a lot of data in log files. To eliminate this situation, the system reviews logs and removes notices, when only a fragment of the file was read. If user works on a project then often each file is being read multiple times. – in employee abuse detect process each file is taken into account only ones. Some files are not important for this process, e.g. temporary files, and system omits them. Employee abuse detect system creates database with information about reading files form file server and according to designed algorithm. It calculate for every user a final parameter (p). This value is compared to defined limit and if it is greater than the system informs administrator about it. This is only warning for the administrator – he has to analyze the situation and decide what

to do. The limit depends on programs used by an employee. It is fixed based on statistics, individually for every user. In the described company it was assumed that protection of newer files is more important than the older ones and the parameter is calculated in the following way:

$$p = \sum_{i=1}^{N} p_i \ , \tag{1}$$

$$p_i = \begin{cases} 1 & \text{for} \quad t < t_1 \\ 1 - (t - t_1)/(t_2 - t_1) & \text{for} \quad t_1 < t < t_2 \\ 0 & \text{for} \quad t > t_2 \end{cases} \tag{2}$$

where N – number of read files, t – file's age, t_1 and t_2 – fixed time periods ($t_1 < t_2$).

Described employee abuse detect system is not ideal, but it can be a very useful tool. Nevertheless it requires proper configuration.

3 Conclusions

It is not easy to design a security system. It is necessary to take into account many dependencies. Some requirements can be contradictory to each other. The most difficult problem is a protection against the dishonest employee, because employees need data access for their work. This paper was described security system in existing firm including original author programs. The system is complicated, uses many programs and is not ideal, but can be effective. It detected a dangerous incident: one of employees copied data from the server probably to give them to transfer it to the competition. Based on information from log files police made a search and found copies of company's data – the implemented system prevented the company from a great loss.

References

1. DRBD – Software Development for High Availability Clusters, `http://www.drbd.org`
2. Stera, P.: Klaster niezawodnościowy dla serwera plików w systemie Linux – przykłady rozwiązań. In: Sieci komputerowe, Wydawnictwa Komunikacji i Łączności (to appear)
3. Sendmail, `http://www.sendmail.org`
4. Resnick, P. (ed.): Internet Message Format. RFC 2822 (April 2001), `http://www.faqs.org/rfcs/rfc2822.html`
5. Synonym, `http://dv8.ro/Synonym/synonym.html`
6. Copying all messages with Sendmail, `http://www.brandonhutchinson.com/Copying_all_messages_with_Sendmail.html`
7. Saout, C.: dm-crypt – a device-mapper crypto target, `http://www.saout.de/misc/dm-crypt`
8. Linux Unified Key Setup, `http://code.google.com/p/cryptsetup`
9. Truecrypt – free open-source on-the-fly encryption, `http://www.truecrypt.org`

10. Kruegel, C., Valeur, F., Vigna, G.: Intrusion Detection and Correlation – Challenges and Solutions. Springer, Boston (2005)
11. Data Loss Prevention Best Practices. Managing Sensitive Data in the Enterprise. A Messaging Media Publication (2007)
12. Squid: Optimising Web Delivery, `http://www.squid-cache.org`
13. SquidGuard, `http://www.squidguard.org`

Pessimistic Useful Efficiency of EPL Network Cycle

Piotr Gaj

Silesian University of Technology, Institute of Informatics,
ul. Akademicka 16, 44-100 Gliwice, Poland
piotr.gaj@polsl.pl
http://www.polsl.pl

Abstract. Nowadays, there are many industrial Ethernet protocols which could act as a fieldbus functionality. There is a necessity to select an independent comparative factor in order to define the scope and the usage domain of the given protocol. In this article the author considers the construction of transactions in EPL (Ethernet PowerLink) protocol and qualification of its features to describe time characteristic and to compare EPL with other protocols of real-time Ethernet type. Author presents advantages and common traits of some of the existing solutions.

Keywords: industrial, real-time, Ethernet, EPL, PowerLink, network cycle, efficiency, transaction, exchange, time analysis.

1 Introduction

Distributed computer systems are used on a control level of a factory automation. The devices and the whole data flow on this level have to be subject to the real time processing due to physical process conditions. The typical communication solutions on this level are a kind of industrial computer networks named fieldbuses [1]. The main task of the fieldbus type networks is to deliver IO data between control devices. The whole traffic on the local network segment has to be performed in the real time. Each transaction has to be done within a strictly defined range of time.

The Ethernet as well as a fieldbus can be used as a network on a control level with full conformity to the time limitation requirements. However, the Ethernet, beside its own protocol defined in physical and link layer [2], has to have a particular protocol designed for time deterministic control of access to the medium, located mostly in the application layer of the protocol stack [3]. The Ethernet provides many advantages coming from its popularity and ubiquitous and well known technology. In the scope of basic fieldbus functionality, the Ethernet could became a universal standardized platform of industrial data exchange.

Unfortunately, many incompatible protocols have been created by commercial and non-profit organizations. There are protocols based on the standardized Ethernet technology as well as on specialized ASICs [4,5]. The EPL protocol is one of at least 22 industrial Ethernet protocols. The selection of the right

A. Kwiecień, P. Gaj, and P. Stera (Eds.): CN 2010, CCIS 79, pp. 297–305, 2010.

protocol according to system requirements as well as a reliable comparison of accessible solutions can be a problem. Sometimes it is claimed that high speed (baudrate) of physical medium determines high efficiency and throughput. It is especially pronounced in case of the Ethernet solutions. Moreover, the interesting question arises whether there are any significant differences between existing real-time Ethernet protocols, from the time characteristic of the transmission point of view. In order to compare such protocols based on fieldbus functionalities an independent factor should be used. One of such criteria can be a useful efficiency [6] defined as a quotient of the transmission time of the useful data T_{U} and the transmission time of the total data used by the network T_{N} during the considered operation (formula (1)).

$$\eta = \frac{T_{\mathrm{U}}}{T_{\mathrm{N}}} \ , \ \text{(non-dimensional, as factor 0–1)}, \tag{1}$$

Such quotient describes the protocol's ability to pass the application data via a given transport mechanism. Calculation of a useful efficiency can be a good base for comparison of the given types of network transactions of such protocols. Useful data come from the node's application and can be simply considered as a set of application variables to be sent through the network, i.e. through the source and the destination protocol stack and the physical infrastructure.

The EPL is quite an interesting solution because of achieving a short period of network cycle based on a standard hardware and software solution according to legacy IEEE 802.3u (Fast Ethernet). The minimum value of a network's cycle period is 200 μs with jitter of 1 μs. The EPL is classified as a high performance solution of the IAONA real-time class 4 [7]. Just like each type of the real-time Ethernet solutions it works on a closed segment with the application control layer above the legacy Ethernet layers.

The protocol is compliant with norms: IEC 61784-2, 3, IEC 61158, IEC15745-1, 4, EN 50325-4 as well as it is a standard solution for the CANOpen protocol over Ethernet (EN50325-4). The EPL is developed and supported by the EPSG organization (Ethernet Powerlink Standardization Group) according to the Open Source Technology idea (BSD license in 2008), which means patent-free without any license fees.

Generally, in the EPL two types of transactions can be distinguished. The useful data can be transmitted between nodes utilizing isochronous cyclic way and asynchronous acyclic way.

Beside the Ethernet MAC addressing the unique EPL nodes addressing is used. The internal addressing is indispensable for sending requests during execution of message cyclic exchange with predefined communication relationships of every node's application as well as for asynchronous dialogue. The cyclic sequence is pre-planned but the acyclic one is dynamically scheduled.

Let us name the sequence as 'exchange scenario'. All replies during this process are based on the Ethernet multicast and all requests on the Ethernet unicast. All useful data serviced isochronously is transmitted via an abstract unit named PDO (Process Data Object).

Subscribers are divided into two categories:

- MN – Managing Node – there is only one node in a single segment. The node is designed to execute the exchange scenario over the whole segment and is able to but not obliged to exchanging useful data.
- CN – Controlled Nodes – there are many nodes in a single segment (all run-time nodes except MN). CN nodes are dedicated to exchanging useful data. CN subscribers are passive and do not send any frames without an MN request.

The EPL uses the SCNM (Slot Communication Network Management) model of medium access. It is a specific sort of the TDMA (Time Division Multiple Access) model where special synchronization frames are used, and following that the duration of network cycle is precisely synchronized. Within the time slices of each network cycle the Master-Slave model of a single node transaction is used. However, the node response is based on the Producer-Consumer 'push' relationship. So generally, in the EPL the PDC deterministic model of network is used with time distribution via synchronized time slicing distributed by the NM.

2 Transactions in EPL

All exchanges of the useful data happen according to an exchange scenario placed in MN. It is a roundtable conversation based on the request & response transactions.

All transactions are executed within a network cycle which is a repetitive time window. A network cycle consists of one or many elementary sub-windows designed to group transactions of a given type. Each elementary window consists of three parts. The window for cyclic RT transactions, the window for acyclic transactions, and the special idle window dedicated to cycle synchronization – i.e. for assuring the constant time duration of a given elementary window. It is presented in Fig. 1.

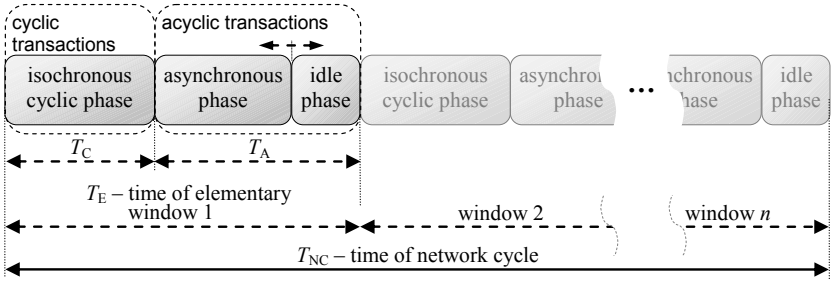


Fig. 1. The schema of the EPL network cycle

Because of the fact that the T_C in the given cycle is constant and T_A is also constant, the duration of each sub-window T_E is constant. Additionally, the number of elementary windows n depends on the scenario construction, and in

the given conditions it is also constant, so the duration of the whole network cycle T_{NC} is constant as well.

The PDO distribution within cyclic phase of the given elementary window can be performed with a constant cycle (continuous) and/or a multiplexed cycle. The difference lies in transactions scheduling of all cyclic variables. Thanks to that, various periods of the useful data exchange can be established in the given network cycle.

In order to illustrate the time characteristic of a network there is a need to select a given aspect of network operation. It is impossible to calculate the value of efficiency or throughput for the whole network activity. However, the minimal and maximal value can be calculated for a given transaction or a set of transactions.

The fundamental transaction in industrial networks is the cyclic one. This type of transaction is crucial from the vertical exchanges point of view. In the EPL case the cyclic transaction is executed within the cyclic phase, according to publish-subscribe relationship similar to the producer-consumer model based on pool request & pool response type of dialog. The MN device generates a request (PReq frame) to the given CN following the scenario order, and the CN device responses via a broadcast (PRsp frame) to all nodes interested in its data. The cyclic dialog is proceeded by the broadcast of the special frame (SOC – Start of Cycle) which synchronizes the start point of cyclic phase within all nodes. The acyclic phase works almost in the same way except direct requesting. Only one transaction of ASnd frame can be performed, due to dynamically filled queues in NM. The schema of the EPL transactions is presented in Fig. 2.

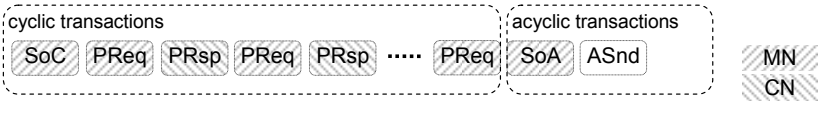


Fig. 2. Schema of EPL transactions

In order to calculate the useful efficiency, the transmission time of data should be defined first and followed by the construction of the frames. To perform a single exchange of useful data between only two CNs, four frames are necessary from the network point of view: SoC, PReq, PRsp, and SoA. The frames are presented in Fig. 3. All sizes are in bytes (octets).

The EPL is supported by hubs and/or switches and node's network interfaces. The frames processing in such a physical infrastructure causes delays, but their values depend on a type and construction of the middleware devices. However, because of short processing time within a single hub device (for instance about 70 ns) or within other ones, and because of the fact that delays concern both useful and control data, the values can be neglected from the useful efficiency point of view defined as (1). The same refers to the propagation time. Nevertheless, in order to perform a precise calculation of given data passing from the

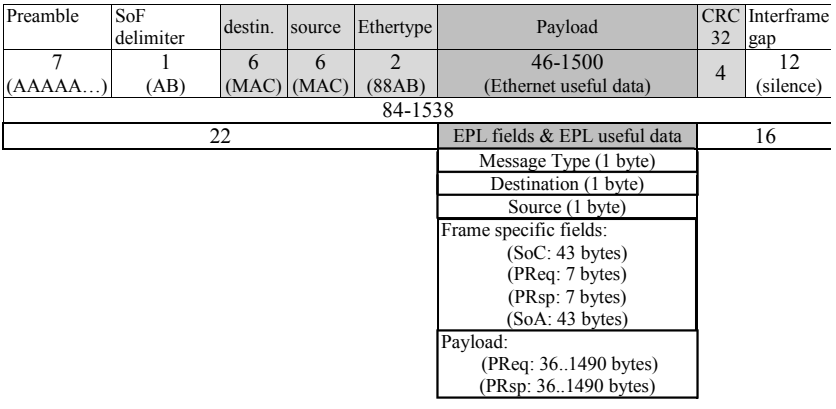


Fig. 3. Schema of EPL frames

application of one node to the application of another node, one should consider the time related to useful data acquisition from the application working within CPU to the application layer of the EPL stack working within a coprocessor of every node involved. In this article, all data transfer is considered only from the network point of view, with assumption that useful data is still available within the network coprocessor's memory.

3 Efficiency

Useful efficiency defined as (1) is independent from network speed and refers to a given type of transaction or some other abstract organization unit of data passing via network. Thus, it can be analyzed as an efficiency of the network transaction of the given PDO, node, an elementary window or even the whole network cycle. The most interesting aspect is the definition of efficiency value for a pessimistic network usage when the amount of useful data is small. In the industry, on the fieldbus level, the size of most of the application variables is either one bit for a discrete signal or up to 16 bits for measurements. Thus, the total data size relevant to the given node is approximately up to several dozen bytes.

Let us assume the size of the useful data u in the range $1 \geq u \geq 128$ bytes. It is a typical range of the data size in automation systems to be exchanged in a cyclic way. Additionally, there is an assumption of Ethernet transmission speed: $V = 100\,\mathrm{Mbit/s}$. The Ethernet/EPL frame sizes presented in Fig. 3 rely on the principle that the minimum EPL frame payload size is 36 bytes.

Let us calculate the useful efficiency of a cyclic transaction of a single node. You can notice that this transaction allows for transmission of data from one node to the remaining nodes in almost the same time, with jitter time depending only on the propagation and the processing within devices. The schema of the transaction is presented in Fig. 4.

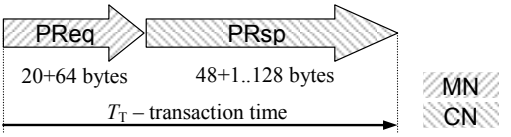


Fig. 4. Schema of the cyclic transaction of a single node

Based on (1) the useful efficiency can be calculated as:

$$\eta = \frac{T_U}{T_T} = \frac{T_U}{T_{PReq} + T_{PRsp}} \tag{2}$$

$$\eta = \begin{cases} \frac{\frac{8n}{V}}{8\left(\frac{84}{V} + \frac{(48+n)}{V}\right)} = \frac{n}{132+n} & \text{for } n \geq 36 \\ \frac{n}{168} & \text{for } n < 20 \end{cases} \tag{3}$$

However, a more interesting case is calculation of useful efficiency with taking into consideration all necessary frames to perform the above-mentioned transaction. The schema is presented in Fig. 5. The assumption is that no ASnd frame and no PRsp frame from MN occurs.

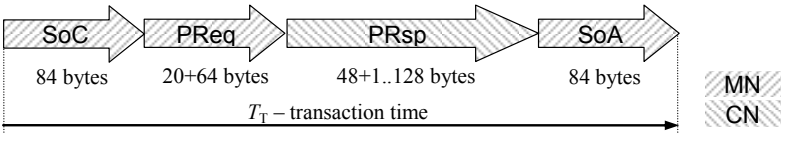


Fig. 5. Schema of the cyclic transaction of a single node with a minimum set of the necessary frames from the network cycle

Based on (1) the useful efficiency is as follows:

$$\eta = \frac{T_U}{T_T} = \frac{T_U}{T_{SOC} + T_{PReq} + T_{PRsp} + T_{SOA}} \tag{4}$$

$$\eta = \begin{cases} \frac{\frac{8n}{V}}{8\left(\frac{84}{V} + \frac{84}{V} + \frac{(20+28+n)}{V} + \frac{84}{V}\right)} & \text{for } n \geq 36 \\ \frac{n}{336} & \text{for } n < 36 \end{cases} \tag{5}$$

Calculating useful efficiency for a single PDO, a node or even a time window is possible but the results are strictly theoretical and do not refer to the real behavior of the network. This is caused by the fact that the transaction with any of the mentioned elements never occurs alone. Thus, in order to perform any actions on the network, the cycle has to be established and the considered transaction within the cycle has to be executed. So, the pessimistic condition is to execute a single cyclic exchange of short useful data within the cycle of the minimum suited size. The other extreme condition is to transmit the maximum amount of useful data within the same network cycle. It gives a real factor of

protocol efficiency as well as of real throughput. Taking into consideration that most of the real-time Ethernet protocols with functionality of the cyclic data distribution rely on the network cycles, it is good to compare the existing solutions. Of course, the additional transactions involving execution of transactions related to the specific communication relationships are not taken into consideration in this approach. It is appropriate to describe only the producer-consumer push model. Nevertheless, this simplification should be sufficient to describe the time characteristic irrespective of the exchange models used together with the specific relationships.

There are additional assumptions for calculations for the whole cycle:

- the minimum EPL cycle period is $T_{\mathrm{NC}} = 200\,\mu\mathrm{s}$,
- only one node is cyclically requested, no ASnd, no PRsp from MN.

Finally, let us calculate the cyclic transaction within the EPL continuous cycle. The transaction is presented in Fig. 6.

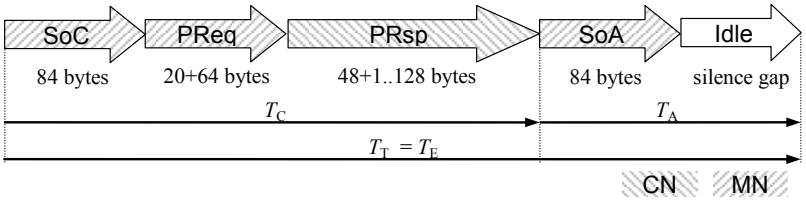


Fig. 6. Schema of the cyclic transaction of a single node within a single elementary cycle

The time of all network exchanges T_{X} consists of:

$$T_{\mathrm{X}} = T_{\mathrm{SOC}} + T_{\mathrm{PReq}} + T_{\mathrm{PRsp}} + T_{\mathrm{SOA}} \tag{6}$$

The transmission time of useful data can be calculated as:

$$T_{\mathrm{U}} = \frac{8u}{V}\,[\mathrm{s}] \tag{7}$$

Based on (1), the useful efficiency can be calculated according to the formula (8):

$$\eta = \frac{T_{\mathrm{U}}}{T_{\mathrm{NC}}} \tag{8}$$

Including equation (7) into (8) and the above-mentioned assumptions:

$$\eta = \frac{\frac{8n}{V}}{T_{\mathrm{NC}}} = \frac{8n}{T_{\mathrm{NC}}V} \tag{9}$$

The equation is valid only when:

$$T_{\mathrm{NC}} \geq T_{\mathrm{X}} \tag{10}$$

In the other case the transaction cannot occur because of not sufficient time of the network cycle to contain T_X. To calculate this case the T_{NC} should be increased by the multiplication of the cycle time granularity which is 100 μs or 125 μs. For one node this inequality is always true because the maximum value of T_X is 137 μs.

The calculation results are presented in Fig. 7. The values are compared with the values of Profinet IO and EGD protocols calculated in the same way [8].

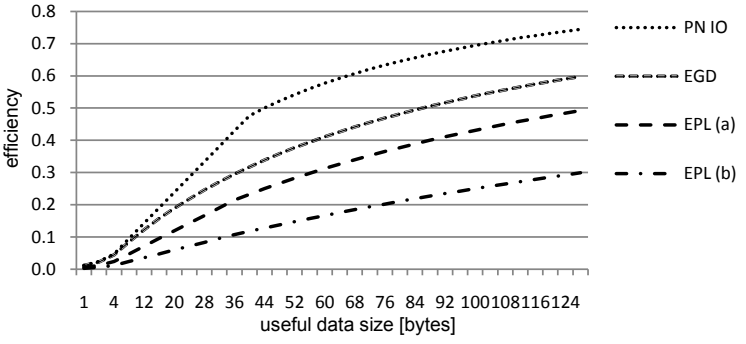


Fig. 7. Comparison of useful efficiency of EPL cyclic transaction without taking into consideration the network cycle: case (a) from Fig. 4, case (b) from Fig. 5

The calculated useful efficiency with various periods of the network cycle taken into account is presented in Fig. 8.

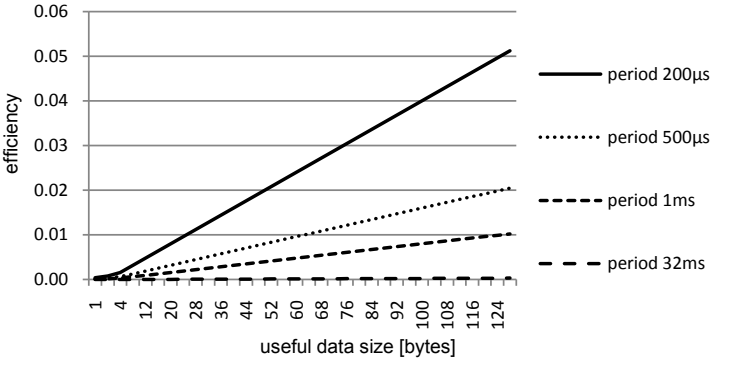


Fig. 8. Comparison of useful efficiency of EPL cyclic transaction with the network cycle taken into account

4 Conclusions

It can be observed that there is a big difference between useful efficiency calculated for an internal element of the cycle and the whole cycle (Figs. 7, 8). Nevertheless, the efficiency of internal transmission units is strictly theoretical and useless.

It is necessary to notice that for the given protocol class based on network cycles, useful efficiency calculated for the given activity within the whole network cycle does not depend on the protocol construction. Efficiency is calculated as a quotient of the transmission time of the useful data and the cycle time, which is constant. Both the first and second value, when the transmission speed is constant, are independent from the frame construction and the execution of transaction. So there are no differences between protocols of this type from this point of view. The differences lie in the internal construction of the communication relationships, however, their separate analysis is pointless, as it has been mentioned above.

There is a necessity to analyze such protocols with utilization of the calculation of the useful efficiency per cycle and while considering utilization of cycle space. Such analysis is possible if service of more than one node is considered.

The values obtained in this article result from the network cycle and describe the pessimistic values of useful efficiency within the cycle. The interesting point is calculation of optimistic values of useful efficiency and useful throughput. Such values can be calculated for a cycle fully utilized by passing the given data via the given mechanism selected from the available ones.

References

1. Mahalik, N.: Fieldbus Technology. Springer, Heidelberg (2003)
2. Krysiak, K.: Sieci komputerowe. Helion, Gliwice (2003)
3. Gaj, P., Ober, J.: Problemy z wykorzystaniem sieci Ethernet w aplikacjach przemysłowych. In: Studia Informatica, Gliwice, Poland, vol. 24 (2003)
4. Larsson, L.H.: Fourteen industrial Ethernet solutions under the spotlight. The Industrial Ethernet Book web page
5. Popp, M., Weber, K.: The rapid way to Profinet. PNO (2004)
6. Gaj, P., Ober, J.: Parametry czasowe sieciowych transakcji cyklicznych wykorzystujących kontrolę wymian w sieci Ethernet. In: Proceedings of the SCR 2003 Conference, Gliwice, Poland (2003)
7. EPSG Draft Standard 301, Ethernet POWERLINK Communication Profile Specification Version 1.1.0. EPSG (2008)
8. Gaj, P., Kwiecień, B.: Useful efficiency in cyclic transactions of Profinet IO. In: Studia Informatica, Gliwice, Poland (2010)

Spanning Tree Protocol in Wireless Industrial Communication System

Wojciech Domagała

Silesian University of Technology, Institute of Informatics,
Akademicka 16, 41-100 Gliwice, Poland
wojciech.domagala@polsl.pl

Abstract. The paper forms a basic classification of wireless networks basing on the subsequent functionalities provided by different topologies in comparison to the wired network. Then the focus is placed on the idea of a wireless mesh network as a structure freely mobile and versatile providing many extremely interesting and innovative opportunities in industrial, real-time communication systems. A set of following measurements verifies usability of one of the most popular mesh-based topology control protocols – Spanning Tree Protocol – in terms of different kinds of time-deterministic systems or high reliability systems. In the end final conclusion and an attempt to point the most intriguing issues and directions of research and development is made.

Keywords: industrial network, mesh network, self-healing, redundancy.

1 Introduction

Despite increasing interest in wireless communication technologies their application in industrial distributed control and supervisory systems is still rather uncommon. The field of wireless sensor networks seems to be leading in increasing the wireless solutions' share among industrial communications systems, however, it doesn't contribute to the the general approval of the wireless communication as an attention-worthy technology. Recent years have brought some interesting solutions in the field of sensor networks or broadband based on IEEE 802.11 standard (mainly due to it's flexibility and effortless integration with wired Ethernet) for mobile controlled objects but they were usually established on a client-access point model. Although this model allowes multiple access points and mobile client roaming, it usually still involves static wired network with access points as network's wireless entry points.

Automation and control engineers and designers together with widely understood industry seem to expect more additional functionality and flexibility from wireless communications. There are also strict requirements in terms of security which need a lot of new solutions forced by the open character of the transmitting medium and reliability which has been already extremely enhanced in wired networks. Producers reacted to these demands and their products implemented

A. Kwiecień, P. Gaj, and P. Stera (Eds.): CN 2010, CCIS 79, pp. 306–314, 2010.

features already known from managed wired networks. Among these innovations is Spanning Tree Protocol (STP) which enables communication engineers to build ad-hoc, self-healing mesh networks. Extended list of features provided by STP looks promising but does it meet all industry's demands? This paper contains a general view on the subject of STP paying special attention to its usability in industrial communication applications with specific reuirements.

2 Concepts for Wireless Control Systems

2.1 General Classification

Wireless systems are enabling engineers to build a whole new class of communication networks providing many new functionalities. Most general and simplified classification could be drawn as below, dividing wireless systems into three subclasses.

Wire-Replacing Interconnections. Undoubtedly simplest and first to be applied example of a wireless technology. This one (presented in Fig. 1) substitutes wire with a wireless link interconnecting existing network structures and can be described as a point to point connection. Sensible solution where wiring can cause difficulties but not presenting any real innovations in terms of functionality.

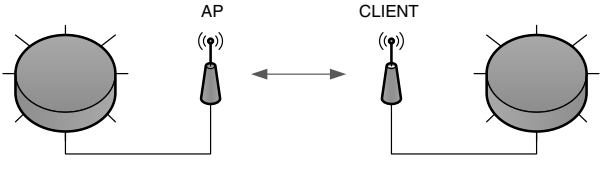


Fig. 1. Static Network Structures Interconnection

Flexible Network Extension. This solution (Fig. 2) provides features which were rare while using wired technologies. An idea of connecting many clients to an access point to form point to multi-point structure creates an opportunity to build communications systems with relatively freely mobile clients constantly connected to the network.

Basic structure, with a singe access point, is already a step forward, however, many access points across the network provide further functionality and rise clients' mobility to a higher level with an opportunity of roaming between access points.

Wireless Distribution System. The most advanced and utilizing the idea of wireless system to the highest degree class of wireless systems. It virtually abandons (or, more precisely – allows abandoning) the idea of dividing network nodes to clients and access points. The mesh network (basis for Wireless Distribution Systems) can consist of access points only but unlike in structures

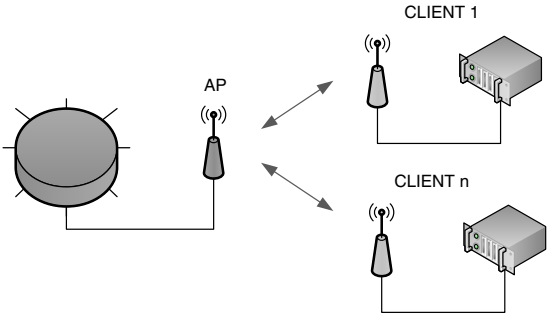


Fig. 2. Static or Mobile Clients Access

mentioned earlier in the paper, access points can connect to each other forming the whole network structure (Fig. 3). Wireless medium allows to dynamically change connections while the network is operating so the actual topology can adapt to the current situation. Nodes' mobility is also further increased in this case, any node can move anywhere, as long as in the range of any other node, maintaining connection to the network.

This flexibility, however, unfortunately also generates problems. Freely interconnecting nodes may form loops which are generally disallowed in a properly functioning network. Such an architecture is a very interesting field of experiment and development and many solutions to different issues concerning mesh networks have already come to existence hence the rest of the paper is focused on this particular class of systems.

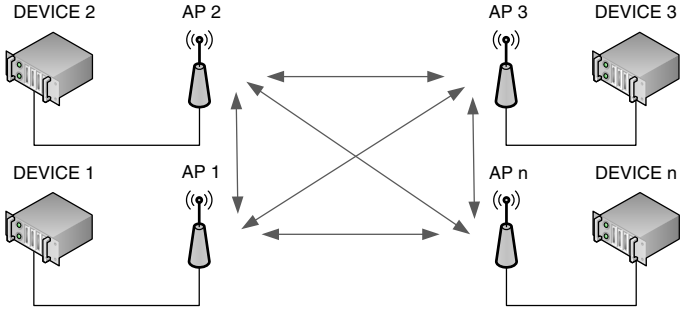


Fig. 3. Nodes Forming a Mesh

2.2 Spanning Tree Protocol

Spanning Tree Protocol used in wireless systems is an idea derived directly from wired networks. Its basic task is to build a mesh-based topology providing redundancy and at the same time preventing loops in the network (ensuring a single active path between two nodes). There are some differences between wired and

wireless version of STP however the principal working rule is the same and time parameters suited for tuning the spanning tree algorythm closely resemble their equivalents from wired networks. There are many parameters but usually only a set of three is available for editing, the rest is set to default values recommended by standard creator (IEEE) and changing them is depreciated (or even impossible in many cases, when the end user cannot even check their values) [1]. Parameters provided for tuning are:

- Bridge Hello Time,
- Bridge Forward Delay,
- Bridge Maximum Age.

They are all usually represented in seconds (taking on integral values) though some industrial equipment manufacturers provide shorter time base [2].

Hello Time determines cyclic hello message broadcast to other network nodes, Forward Delay determines the time of listening and learning before traffic forwarding begins and Maximum Age determines the time of storinf protocol information in the node.

The redundancy and self-healing features look promising for anyone dealing with computer systems of increased reliability demand. Following sections of this paper consider using STP in different kinds of such systems.

3 Equipment and Measurements

3.1 Network Model

To put the Spanning Tree Protocol under a test a small network model was built in the laboratory to simulate a simple, generalized industrial communication system (decentralized control system [3]). The network consisted of four wireless access points working in an IEEE 802.11 standard, with ability to create Wireless Distribution System utilizing STP. Two access points served as entry points for devices communicating in the network, in this case – PLCs[1]. The simplest communication routine was used – one PLC was a data source, called Producer, and the other was a data sink, called Consumer. The Producer repeatedly transmitted a pack of data at a constant time rate of 20 milliseconds. Laboratory conditions have enabled every device to connect to any other therefore access points were configured to connect to a limited list of nodes, forcing more than a single hop between the Producer and the Consumer (Fig. 4). This provided a simple structure permitting to break the working connection and observe the consequences and the self-healing process. Observations were made without any additional equipment. Consumer device with a sweep time of roughly 0.2 milliseconds was running an application counting incoming data packets and recording the maximum time span observed between subsequent packets.

[1] Programmable Logic Controller.

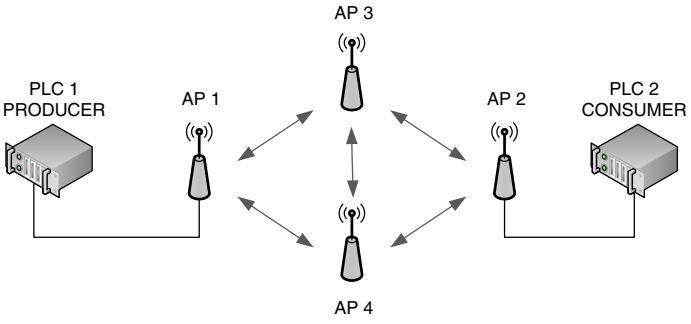


Fig. 4. Measuring equipment

3.2 Measurements

Measuring research has been carried out in seven measurement series. Each series illustrated the way in which the modelled network was working for different values of Hello time (t_{H}), Forward Delay (t_{FD}) and Maximum Age (t_{MA}). Each series consisted of two stages:

- measuring in stabile conditions,
- reaction measuring in case of network breakdown.

Measurements in stabile conditions were aimed to determine the influence of additional load caused by using Spanning Tree Protocol (with parameters' values appropriate for every series) to the model network [4]. The values of maximum delay time (t_{D}) and percentage of lost packets (L) during the measurement[2] series were recorded by the monitoring software.

Reaction measurement in case of network breakdown was based on the already mentioned specific confiuration of inter-access point connections forcing hops through the nodes not directly connected to communicating devices (in this case – PLCs). The breakdown was initiated by turning off the node which was forwarding traffic through and then waiting for reestablishing data transfer between devices. The measurement provided values of this reestablishment time (t_{R}) being actually another measurement of maximum delay time, however, in different conditions.

Each measurement series has been carried out several times for complete picture, checking the repetitiveness of results and their credibility. Following Table 1 has been constructed by averaging repeated measurements belonging to a particular series – the wireless system was not shielded in any way so measurements could have been randomly affected and altered by interference from the surrounding environment – this led to averaging results and not including any

[2] Every measurement series lasted the same amount of time and the constant time value of synchronic data production allowed to evaluate the theoretical number of data packets send which, together with measured value of received packets, resulted in evaluation of percentage of lost packets.

additional statistical analysis, as measurements were anticipated only to check the general rule and tendency, not precisely describe STP with absolute numbers (this would require a lot more preparations, isolation of measured network and many more measurements for different devices' parameters and network load characteristics).

Table 1. Measurements summary (average values)

SN	t_{H}[s]	t_{FD}[s]	t_{MA}[s]	t_{D}[s]	L[%]	t_{R}[s]
1	1	3	8	0.28	0.3	7.9
2	0.125	3	8	0.24	0.7	8.0
3	0.125	1.5	8	0.36	0.3	5.0
4	0.125	1.5	4	0.24	0.2	5.1
5	0.125	1.5	2	0.49	0.7	4.9
6	0.125	1	1.5	0.84	6.0	4.0
7	0.125	0.5	2	2.27	32.5	–

One glance is enough to conclude several points from the results gathered in the Table 1:

- the parameter which influences the operation of network utilizing STP most (both in the positive and negative manner) is Forward Delay time,
- the scope of packet loss values suggests that limits of t_{FD} values, where the quality of network services is acceptable have been reached,
- the meaning of remaining two parameters seem to be irrelevant in comparison to t_{FD},
- the minimum value of communication reestablishing time is two orders of magnitude bigger than average data production time (20 milliseconds) which, in case of breakdown, can lead to significant data loss or critical control errors.

Let's examine a little closer the most influential parameter. Attached charts display the relation between Forward Delay time and percent packet loss (Fig. 5) increase of which is considered negative – meaning cost, and network self-healing time and transmission reestablishment (Fig. 6) decrease of which is considered positive (the reaction of the network to the damage is improving) – meaning gain or profit. These charts clearly demonstrate that changing (decreasing) the t_{FD} parameter down to the certain threshold is followed by linear increase of the gain, however, above that threshold (in this case $t_{\mathrm{FD}} = 1\,\mathrm{s}$) the increase of gain is accompanied by the increase of loss which is, in contrary to gain change characteristic, close to exponential. The influence of STP protocol on the network quickly becomes inacceptable. Even for the small model network the minimum self-healing time achievable while network quality is still fine (arbitraty assumption of packet losses under 1%) is 5 seconds.

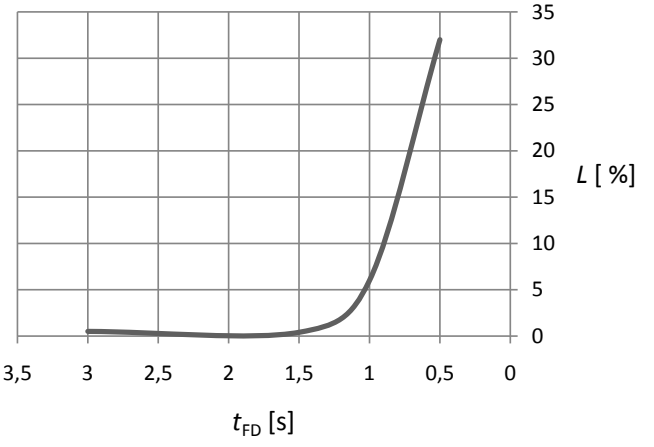


Fig. 5. Percentage of lost packets $L[\%]$ for different $t_{\mathrm{FD}}[\mathrm{s}]$ values

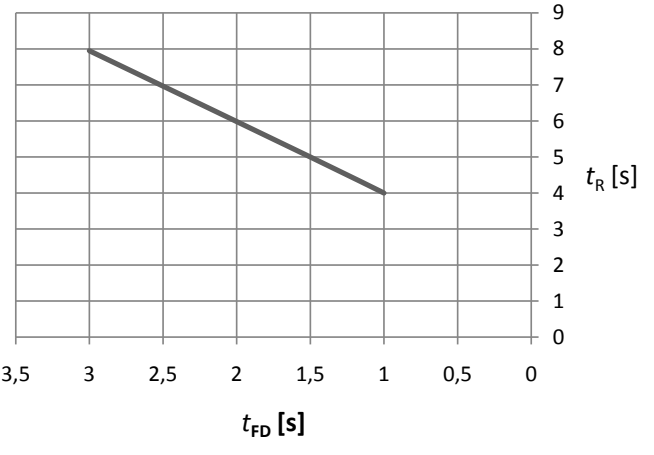


Fig. 6. Communication resume time $t_{\mathrm{R}}[\mathrm{s}]$ for different $t_{\mathrm{FD}}[\mathrm{s}]$ values

3.3 Additional Observations and Conclusions

Apart from insights described in a previous section other symptoms influencing the general conclusions have also been observed:

- during the breakdown the broken node resuming also causes a several second delay, when the protocol agorithm builds a new spanning tree,
- during stabile operation maximum delay is not always directly affecting the packet loss.

Measurements from series no. 6 can be used as an example of the latter observation. For the measurement series of 30 seconds a delay of 0.49 second makes roughly 1.6% of the whole measuring time. Having in mind a constant time rate of data production, this value should cause a packet loss of similar value, however the measured value is distinctly lower. This is a basis for saying that proper traffic buffering in network nodes could be used to reduce the negative effects of strict parameters adjustment.

From examining the measurements and observations as a whole we can draw a conclusion, that with particular transmission parameters and network throughput configuration it can be difficult to reach communication resuming delay time low enough to be acceptable in most of control processes. These included measurements are naturally not to be taken as absolute indicators of STP protocol usability – after all they were carried out for a single specific model, with many parameters set to fixed values and not considered at all (e.g. amount of transmitted data, inter-node maximum throughput). However, results demonstrate what to expect of network parametrizing of a certain structure (topology) and requirements (traffic characteristics and target parameters).

Despite the fact, that the connection resuming time is not fully satisfactory it is still low enough to significantly outclass human capabilities of reaction to breakdown and the time of resolving it by maintenance service. From this reason such system can be used in applications gathering not critical[3] data providing a compromise between a self-healing solution and relatively low amount of lost data.

4 Summary and Future Work

Conclusions drawn from measurements are enough to see the general problem of proper network parametrization for industrial communication system although they don't discredit STP as an interesting solution, especially as new standards for mesh based networks are constantly under development, both for specialized and universal devices. However to fully understand limitations and capabilities of these methods more precise and thorough measurements have to be carried out, considering also other parameters describing computer networks and demands put on them.

Next steps following conclusions of the paper above will most probably involve general analysis of cable Ethernet solutions (to find out the most innovative methods in the field from which the IEEE 802.11 often derives) and further examining of wireless solutions in both industrial and of-the-shelf versions (where the latter are usually much cheaper and tend to implement new standards and technologies sooner).

References

1. Cisco Systems, Configuring STP and IEEE 802.11 MST(Document: 78-14099-04)
2. Westermo Teleindustri AB, RM-240 Reference Guide (6193-5201)

[3] Where the break of real-time communication does not influence the direct risk for people's health, machines or the technological process.

3. Kwiecień, A.: Analiza przepływu informacji w komputerowych sieciach przemysłowych. Wydawnictwo Pracowni Komputerowej Jacka Skalmierskiego (2003)
4. Stój, J.: Wpływ redundancji na zależności czasowe w rozproszonych informatycznych systemach czasu rzeczywistego. Politechnika Śląska. PhD dissertation (2009)
5. Breyer, R., Rileyi, S.: Switched, Fast and Gigabit Ethernet. Macmillan Technical Publishing, Basingstoke (1999)
6. Willig, A., Matheus, K., Wolisz, A.: Wireless Technology in Industrial Networks. IEEE, Los Alamitos (2005)
7. Roshnan, P., Leary, J.: 802.11 Wireless LAN Fundamentals. Cisco Press (2003)
8. Grzywak, A., Kwiecień, A., Gaj, P., et al.: Laboratorium Sieci Komputerowych. Wydawnictwo Politechniki Śląskiej (1999)
9. IEEE Std 802.11-2007, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. IEEE, Los Alamitos (2007)

OPC Data Acquisition Server for CPDev Engineering Environment

Dariusz Rzońca, Jan Sadolewski, and Bartosz Trybus

Rzeszow University of Technology, ul. W. Pola 2, 35-959 Rzeszów, Poland
drzonca@prz-rzeszow.pl, js@prz-rzeszow.pl, btrybus@prz-rzeszow.pl

Abstract. OPC Server has been created for the CPDev engineering environment, which provides classified process data for OPC client applications. Hierarchical Coloured Petri nets are used at design stage to model communications of the server with CPDev target controllers. Implementation involves an universal interface for acquisition data via different communication protocols like Modbus or .NET Remoting.

Keywords: CPDev, OPC DA, HTCPN, Petri nets.

1 Introduction

OPC (*OLE for Process Control*) is a group of standards specifying the communication in industrial automations, based on OLE (*Object Linking and Embedding*), COM (*Component Object Model*), and DCOM (*Distributed Component Object Model*) technologies. Due to its complexity, it has been mainly used in huge systems, however nowadays, it becomes more common, also in smaller applications. It allows connecting control devices with engineering environment, as well as local or remote data sharing between different applications. It is maintained by the OPC Foundation, which constantly develops and publishes new revisions of the specifications (http://www.opcfoundation.org/).

The paper discusses design and implementation of an OPC Server for the CPDev (*Control Program Developer*) engineering environment created at Rzeszow University of Technology. The design is based on Hierarchical Coloured Petri Net model. CPDev is a multi-target solution, so the implementation of the OPC Server requires an universal interface for acquiring real-time data via different communication protocols.

2 OPC Standard

Communication in the OPC standard is based on the client-server model [1]. Typically, an OPC server interfaces engineering applications with automation appliances. It can be integrated with an industrial device or work independently as an autonomous module. Sharing data involves custom, automation or SOAP (*Simple Object Access Protocol*) interface (last two are optional).

The OPC standard includes few specifications, distinguished by data and access types. Most common are:

A. Kwiecień, P. Gaj, and P. Stera (Eds.): CN 2010, CCIS 79, pp. 315–321, 2010.

- OPC DA (*Data Access*) provides real-time data sharing. Clients are able to read and write actual data from the server, alternatively the server can send set of data to the client when the value is changed.
- OPC HDA (*Historical Data Access*) allows storing historical data and transferring values from a specified period. As a result, statistical data (like maximal or mean values) can be obtained.
- OPC A&E (*Alarms and Events*) was developed to inform the operator about situations which demand immediate reaction or particular attention. OPC A&E servers detect alarm conditions in connected devices.

The OPC DA specification [2] is the most common. The OPC DA server data structure is usually hierarchical, similar to a directory tree in file systems. *Items* are organized in such a way that shows logical or physical connections between them. Each item provides several *properties*. Essential ones are *value*, *quality* and *time stamp*. Value contains the actual data of a plant device. It can be provided in many formats, like integers, strings or complex types. The OPC client is able to set a new value, which triggers the modification of the real output. Changing the physical value monitored by an input sensor involves actualization in the OPC server. Last modification date is provided as the time stamp, thus client is informed at which point of time the value was set. Quality is related to the validity of the data.

Although data can be read and set directly, as described, it is not typical. Preferred way of communications involves using *groups*. A group is the logical set of items chosen by the client. This way items can be conveniently grouped together, according to the client request, regardless of server hierarchy. Server can also monitor data values in a group, and notify the client when it changes. This way of data sharing is more efficient than periodic data pooling by a client, especially if data are changed sporadically.

3 CPDev Environment

The Control Program Developer (CPDev) [3] is an engineering environment which provides tools and solutions for creating, compiling and running control applications. A user starts a project within the CPDev Integrated Development Environment (IDE) for writing program algorithms in Structured Text (ST), a language defined in IEC 61131-3 standard [4]. One can also choose IL (Instruction List) [5] or FBD diagram. Figure 1 shows the main window of the IDE with START-STOP program written in ST.

CPDev compiler is used to produce an executable code of the program. The code, called VMASM can be run by CPDev virtual machine (VM). The concept of the virtual machine has been introduced with the following assumptions in mind:

- to provide a common execution layer for programs written in different IEC languages,
- to adapt miscellaneous hardware of control devices to CPDev easily.

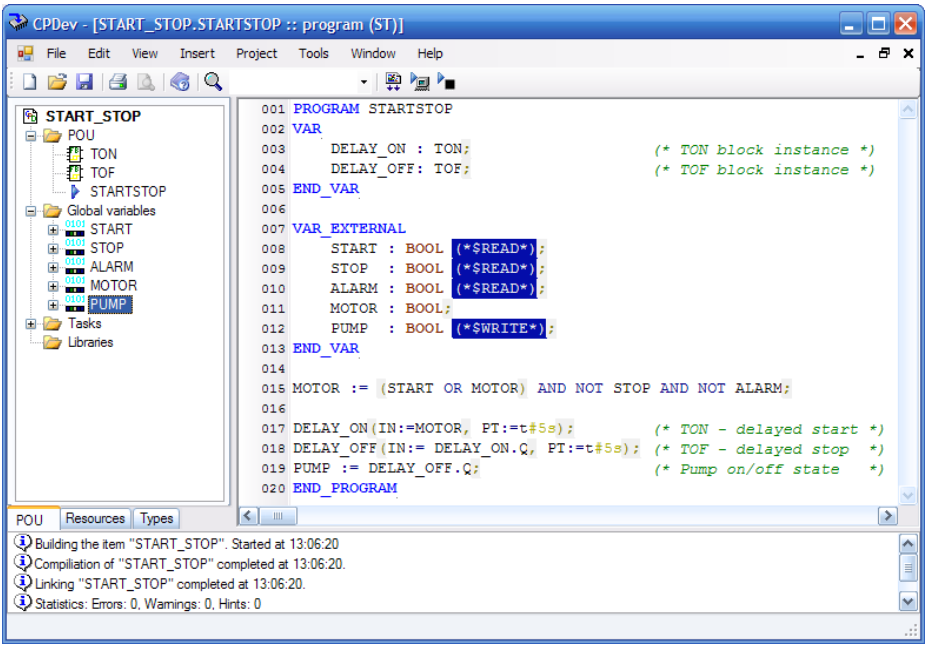


Fig. 1. CPDev IDE with Start-Stop project

The virtual machine should be implemented on a particular target platform to execute the universal code. So far, the virtual machine (written in C) has been implemented for the SMC controller [6], ARM-based boards, Windows Embedded machines and Windows soft-controller CPCtrl [7]. Special version of the machine is used by CPSim simulator. There is also a prototype controller using FPGA technology.

4 HTCP-net Model

HTCP-net [8] model of the OPC server has been created at the design stage. It is shown in Fig. 2. *In* and *Out* places are ports, connecting this subnet with the main model of the CPDev virtual machine via substitution transition. The CPDev VM model has been described in details in [9], and will not be discussed here.

Upper part of the net represents synchronization of the buffers with data stored in the virtual machine memory. Such synchronization is performed once in every cycle of the virtual machine. It has been assumed that such synchronization takes `sync_time` milliseconds, so when transition *Buffers synchronization* fires, tokens will be delayed accordingly. Part of the net located below models the communication with OPC clients. Only three clients has been shown here for simplification, however more of them can be considered if necessary. Clients

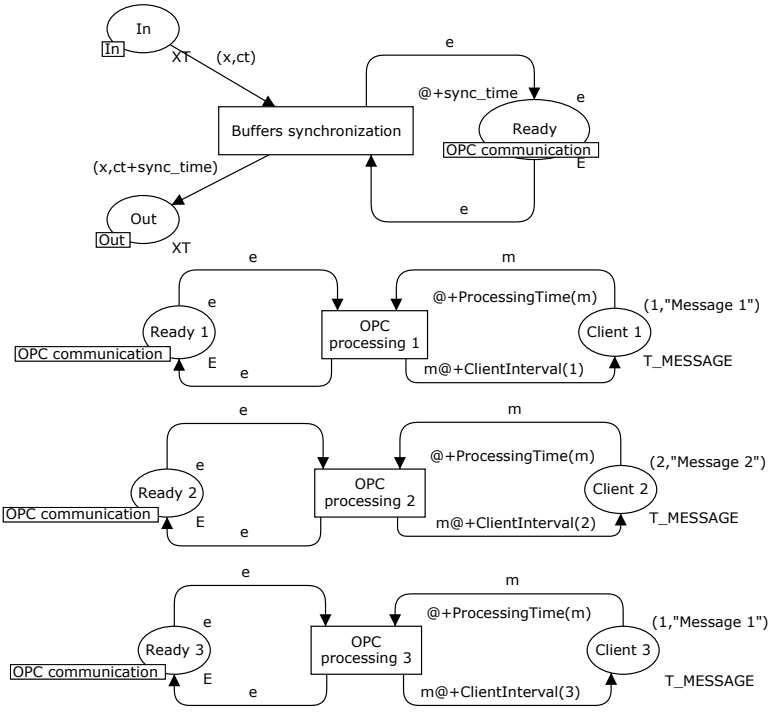


Fig. 2. HTCP-net of the OPC server

periodically send messages, processed by the *OPC processing* transitions. Places *Ready* set up the *OPC_communication* fusion place. A token here represents readiness for communication, or synchronization of the buffers. Every of the *OPC processing* transitions grabs the token during processing of the received message, and delays it according to the `Processing Time` function. This function models various processing time for different messages, and is defined as follows:

```
fun ProcessingTime(no: INT, s:STRING) = case no of
1 => 5
|2 => 10
|_ => 0;
```

Inscription at the arc from *OPC processing* to *Client* delays token according to the `ClientInterval` function. As a result, appropriate delay between consecutive messages from the same client can be defined.

5 Data Acquisition Interface

Although the CPDev target platforms rely on the same virtual machine, they may differ much in the way the program variables can be retrieved or modified

on-line. For example, in case of SMC controller, this can be achieved via Modbus protocol [10], while Windows-based soft-controller is managed from outside via interprocess communications. For a list of possible platforms and their communication interfaces see Table 1.

Table 1. CPDev data sources

Target platform	Communication interface
SMC controller	Modbus RTU
CPCtrl soft-controller	Interprocess communication via .NET Remoting
Windows Embedded	Interprocess communication via Distributed COM (DCOM)
FPGA-based controller	Intel-Hex protocol

The main idea of the CPDev OPC server was to make it platform-independent and provide services for different targets. The targets with data acquisition and modification capabilities will be called *data sources* here. To bring them together an universal interface has been introduced. By implementing this interface, a data source can be attached to the OPC server (Fig. 3).

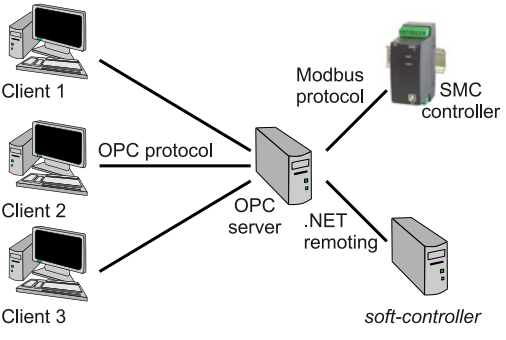


Fig. 3. CPDev OPC server attached to different target platforms

The data source interface consists of several methods and properties. Some of them are shown in Table 2.

Basic functions used by the OPC server are *GetData* and *SetData*, which handle values of a program variables at execution time. Other functions (*GetCycle*, *GetSystemClock*, *GetProgramInfo*) provide general information about the program state or manage the virtual machine (*Run*, *Stop*, *LoadProgram*). The function *Configure* is used mostly for setting communication parameters (serial port, speed, etc.).

Data source handling procedures are compiled into the form of .NET assembly. Such library can be then registered and used by the OPC server (Fig. 4).

Table 2. Functions of the data source interface

Function	Description
GetData	Returns a program variable
SetData	Sets a variable to a new value
GetSystemClock	Returns a value of the target's system clock
GetCycle	Returns program cycle counter
GetProgramInfo	Returns program information (name, CRC, etc.)
Run, Stop	Starts or stops program execution
LoadProgram	Loads a specified program to the target device
Configure	Performs a configuration of the data source

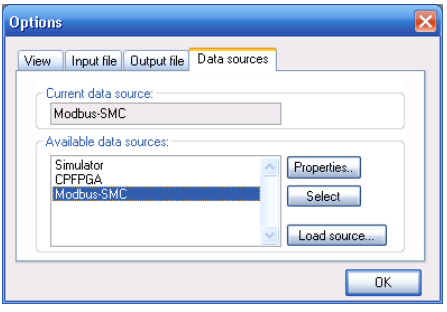


Fig. 4. Data source management in OPC server configuration window

6 Program Variable Lookup

The CPDev virtual machine does not operate on variable names, but on their addresses. The data source interface functions expect then variable addresses as parameters, while OPC functionality requires variable names. A bridge between the two is a *data map* produced by the CPDev compiler (beside the executable code). The data map is an XML document, which stores all information about global and internal variables, function blocks, arrays and structures, tasks etc. A portion of such document for the START-STOP project is shown below.

```
<DATA_MAP>
 <OBJ ObjType="VAR" LName="START"
  Addr="0000" Size="1" Type="BOOL" />
 <OBJ ObjType="VAR" LName="STOP"
  Addr="0001" Size="1" Type="BOOL" />
 <OBJ ObjType="VAR" LName="ALARM"
  Addr="0002" Size="1" Type="BOOL" />
 <OBJ ObjType="VAR" LName="MOTOR"
  Addr="0003" Size="1" Type="BOOL" />
 <OBJ ObjType="VAR" LName="PUMP"
  Addr="0004" Size="1" Type="BOOL" />
</DATA_MAP>
```

The document describes global variables (START, STOP, ALARM, PUMP) with their addresses, types (boolean) and sizes. For a more advanced project a hierarchy of objects and variables is created with function blocks, arrays,

structures etc. The XML data map is read by the OPC server, which builds a tree of items to be accessed by an OPC Client.

Beside regular program variables, the server exposes some additional read-only items like program name, cycle count, system clock etc. In case of an error in communication with a data source, the OPC quality field of a variable value changes from 'good' to 'bad'. OPC Data Access specification in version 2 is currently used by CPDev OPC server to access the variables.

7 Conclusions

The new OPC Server for CPDev engineering environment has been described in the paper. It has been created from a formal Petri-net model constructed at design stage. The main functionality of the server is to provide a common data access from multiple CPDev targets. Variable hierarchy in built automatically from the program data map generated by the CPDev compiler. Although the target controllers run the same virtual machine, they use different protocols from distributed interprocess communication (soft-controller) to Modbus (SMC controller). The universal interface integrates those data sources and allows OPC clients to read or write real-time program variables from multiple targets.

References

1. Bożek, A., Rzońca, D.: OPC/DA communications in a microcontroller board prototype. Pomiary Automatyka Robotyka 4/2007, 12–14 (2007) (in Polish)
2. OPC Data Access Custom Interface Specification, Version 3.00. OPC Foundation (2003)
3. Rzońca, D., Sadolewski, J., Stec, A., Świder, Z., Trybus, B., Trybus, L.: Mini–DCS System Programming in IEC 61131-3 Structured Text. Journal of Automation, Mobile Robotics & Intelligent Systems 2(3), 48–54 (2008)
4. IEC 61131-3 standard: Programmable Controllers Part 3, Programming Languages. IEC (2003)
5. Szmyd, E., Sadolewski, J.: Polish compiler and controllers programmable with IL language of IEC 61131-3 standard. Pomiary Automatyka Robotyka, 2/2009, CD, 615–622 (2009) (in Polish)
6. Rzońca, D., Trybus, B.: Application of coloured Petri net for design of SMC controller communication subsystem. Studia Informatica 29(4B(81)), 15–26 (2008)
7. Krok, K., Rzońca, D., Sadolewski, J.: Application of I/O boards in the CPDev engineering environment. Pomiary Automatyka Robotyka 2/2009, 9–11 (2009) (in Polish)
8. Jensen, K.: Coloured Petri Nets. In: Basic Concepts, Analysis Methods and Practical Use, Springer, Heidelberg (1997)
9. Rzońca, D., Trybus, B.: Hierarchical Petri Net for the CPDev Virtual Machine with Communications. In: Kwiecień, A., Gaj, P., Stera, P. (eds.) 16th Conference on Computer Networks, CN 2009, Wisła, Poland. CCIS, vol. 39, pp. 264–271. Springer, Heidelberg (2009)
10. Modicon MODBUS Protocol Reference Guide. MODICON, Inc., IAS, Massachusetts (1996), http://www.modbus.org/docs/PI_MBUS_300.pdf

Analysis of Parallel Iterative Solving Methods for Markovian Models of Call-Centers

Jarosław Bylina and Beata Bylina

Institute of Mathematics
Marie Curie-Sklodowska University
Pl. M. Curie-Skłodowskiej 1, 20-031 Lublin, Poland
{jmbylina,beatas}@hektor.umcs.lublin.pl

Abstract. The authors consider some numerical and parallel aspects of their iterative parallel algorithm for solving Markov models. The algorithm is a combination of block Jacobi and Gauss-Seidel iterative methods in a parallel version. In the research the numerical accuracy is investigated and some parallel characteristics (performance time, speedup, efficiency, scalability of the algorithm) are examined. The authors analyze the different properties of this algorithm with increasing number of processors and various problem sizes. Research is conducted for large sparse linear equation systems resulting from Markov chains – the matrices are obtained from a Markovian model of a call-center.

Keywords: Markov chains, block iterative methods, parallel algorithms, Jacobi method, Gauss-Seidel method, call-centers.

1 Introduction

The problem of a call center functioning is a very important issue from the point of view of various disciplines: economy, teleinformatics, sociology, psychology. The subject of the teleinformatics is – among many others – to forecast the performance of call centers with the use of simulation [1,2,3,4,5] and analytical modelling [6,7,8].

Both manners – simulation and analytical modelling – have their own advantages and flaws.

The simulation is an easy subject to modifications and can be simply adapted to every system. Moreover, the arbitrary accuracy can be achieved by simulation – given sufficient time. However, simulation can take a lot of time, especially when precise results are needed.

On the other hand some analytical methods can be used. They consists in preparing a mathematical model of the investigated system and then solving some equations describing such a model. From their solutions various characteristics of the system can be obtained.

Among analytical methods we can name: discrete approximation, diffusion approximation, mean value analysis, stochastic Petri nets and many others. But as the most intuitive and natural way of modelling call centers we chose Markovian

A. Kwiecień, P. Gaj, and P. Stera (Eds.): CN 2010, CCIS 79, pp. 322–331, 2010.

queuing models solved with the use of continuous time Markov chains (CTMCs). Such analytical models can give better accuracy in shorter time than simulation mentioned above.

The simplest queuing model describing a very simple call center is $M/M/s$ [7,8] It describes a system with clients of only one class, with s identical serving agents, and with exponential arriving time and exponential serving time.

Some more complicated Markovian models for call centers were considered in [9,10,11].

The authors in [5] presented a queuing model with clients of two classes and agents of three classes (the case with skill based routing – also written SBR – which means that incoming calls are assigned to the most suitable agent and some agents are proficient in serving both clients' classes).

In the present paper authors try to find steady-state probabilities of various events in a call center using parallel programming. In order to computate the steady-state probabilities we must solve an equation $\mathbf{Q}^T\mathbf{x} = \mathbf{0}$. This is reducible to the classical problem of solving a (homogeneous) sparse system of linear equations of the form $\mathbf{Ax} = \mathbf{b}$ of the size equal to the number of states in the CTMC.

A limitation of the Markovian modeling approach is the fact that the CTMC models tend to grow extremely large due to the state space explosion problem. This is caused by the fact that a system is usually composed of a number of concurrent subsystems, and that the size of the state space of the overall system is generally exponential in the number of subsystems. As a consequence, much research is focused on the development of techniques, that is methods and data structures, which minimize the computational space and time requirements for analyzing large and complex systems.

One of such techniques is parallelization. Problems of parallel computations for such systems and finding its steady-state probabilities in parallel is brought up in [12,13,14,15].

The parallel Jacobi method was discussed in [15] and a parallel method for the CTMC steady-state solution is presented there. The Gauss-Seidel method is used for parallel solving of Markov chains in [16,17].

In our previous work [18] a combination of two classical iterative methods for solving linear equation systems, namely the block Jacobi method and the Gauss-Seidel method is presented. These methods were chosen because the presented algorithm is intended for computer clusters and Jacobi method is inherently parallel (Gauss-Seidel method has not got such a property and its parallelization requires a lot of communication), but Gauss-Seidel method usually converges much faster than Jacobi method [19].

The algorithm investigate here is a kind of a 'divide and conquer' algorithm. It consists in dividing the whole problem into some smaller ones – independent as much as it is possible, what makes things easier. Properties of such a combined method (block Jacobi and Gauss-Seidel) are experimentally examined in this paper for a call-center model. Here we investigate some numerical and parallel aspects of the proposed method.

We expect that such a combination of Jacobi method and Gauss-Seidel method will be more accurate than Jacobi method alone (and less accurate than Gauss-Seidel method alone) for the same iteration number (and essentially the same number of floating points operations). This statement will be tested experimentally for exemplary very sparse matrices describing Markovian models of call-centers. We present time execution, relative speedup and efficiency of the algorithm – as the traditional characteristics of parallel algorithms.

This paper is organized as follows. Section 2 explain how we construct our CTMC model where the rate of clients' arrival depends on time. In Sect. 3 we discuss some mathematical aspects of the parallel method of block Jacobi. In Sect. 4 we give some details of the parallel algorithm and its implementation. Section 5 describes conducted experiments. We conclude in Sect. 6 with a summary and future research directions.

2 CTMC Model

Consider a scheme of a call center presented in Fig. 1. The system consists of two queues and three agents' classes. Let us denote:

- q_{T}, q_{B} – the maximum number of clients in the queues Q_{T} and Q_{B}, respectively;
- s_{T}, s_{B}, s_{TB} – the number of agents in groups S_{T}, S_{B} and S_{TB}, respectively;
- λ_{T}, λ_{B} – the intensity of incoming traffic (i.e. the mean value of arrival interval) of clients of class T and B, respectively;
- μ – the intensity of outgoing traffic (i.e. the mean value of service time).

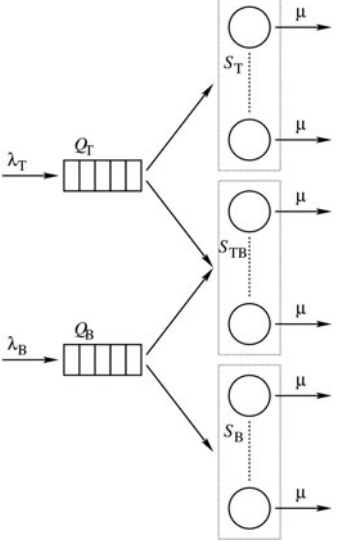


Fig. 1. A queuing model for the investigated call center

We will define the state of the system in the following way:

$$X = (n_{\mathrm{T}}, n_{\mathrm{B}}, n_{\mathrm{TB}})$$

where n_{T} denotes the quantity of clients served by agents from the group S_{T} together with the number of clients in the queue Q_{T}. Similarly, n_{B} denotes the quantity of clients served by agents from the group S_{B} together with the number of clients in the queue Q_{B}. Finally, n_{TB} denotes the quantity of clients served by agents from the group S_{TB}.

The modelled call center is one with skill based routing, which is realized as follows. As long as there are free agents in the group S_{T}, clients from the queue Q_{T} are directed to them – as is the case of S_{B} and Q_{B}, respectively. When there are clients in Q_{T} and there are no free agents in S_{T}, the clients are directed to agents from S_{TB}; similarly for Q_{B} and S_{B}.

The constraints for the elements of the system state $(n_{\mathrm{T}}, n_{\mathrm{B}}, n_{\mathrm{TB}})$ are:

$$\begin{cases} n_{\mathrm{TB}} \in [0, s_{\mathrm{TB}} - 1] \\ n_{\mathrm{T}} \in [0, s_{\mathrm{T}}] \\ n_{\mathrm{B}} \in [0, s_{\mathrm{B}}] \end{cases} \quad \text{or} \quad \begin{cases} n_{\mathrm{TB}} = s_{\mathrm{TB}} \\ n_{\mathrm{T}} \in [0, s_{\mathrm{T}} + q_{\mathrm{T}}] \\ n_{\mathrm{B}} \in [0, s_{\mathrm{B}} + q_{\mathrm{B}}] \end{cases} .$$

Every possible transitions of the system from any given state $(n_{\mathrm{T}}, n_{\mathrm{B}}, n_{\mathrm{TB}})$ are presented in Fig. 2.

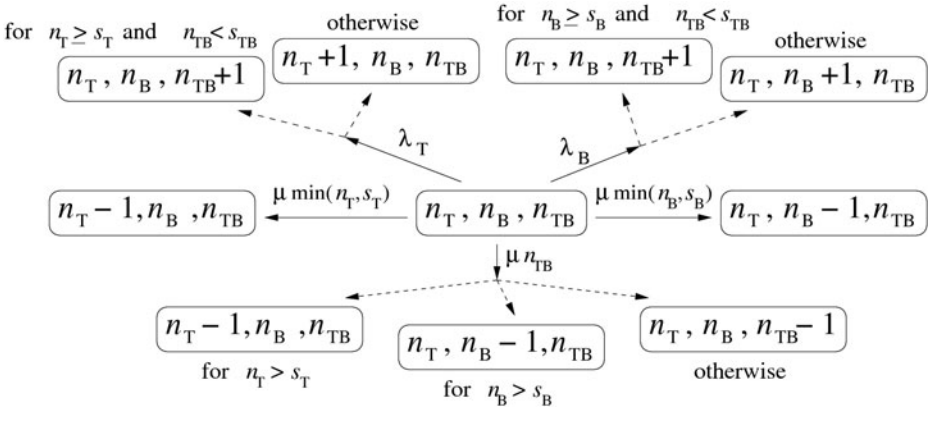


Fig. 2. Possible transitions among states in the presented model

To find the probabilities $\boldsymbol{\pi}$ of states – and then some characteristics of the model – we were to generate a transition rate matrix $\mathbf{Q}$ for our CTMC and then solve a linear equation system

$$\mathbf{Q}^T \mathbf{x} = \mathbf{0} \ . \tag{1}$$

3 A Block Method of Jacobi/Gauss-Seidel

In this section we present an algorithm, which takes advantage of the division of the matrix $\mathbf{Q}$ between computational nodes described in Sect. 4.1. The algorithm proposed here is a combination of Jacobi and Gauss-Seidel iterative methods.

To start from a usual Jacobi method, there is

$$\mathbf{Q}^T = \mathbf{D} - (\mathbf{L} + \mathbf{U}) \ . \tag{2}$$

Let

$$\mathbf{H} = \mathbf{D}^{-1}(\mathbf{L} + \mathbf{U}) \ . \tag{3}$$

Thus, we can write

$$\mathbf{x}^{(k+1)} = \mathbf{H}\mathbf{x}^{(k)} \ . \tag{4}$$

In a block iterative method the linear system (4) is divided into some subsystems. In this section the matrix $\mathbf{H}$ is divided into p blocks (p is the number of computational nodes), each block $\mathbf{H}_i$ ($i = 1, \ldots, p$) of the size of n columns and n/p rows (last one, $\mathbf{H}_p$ can be shorter – it does not influences general considerations). Such a division corresponds to the division of the matrix $\mathbf{Q}^T$ proposed in Sect. 4.1, because to obtain elements of the matrix $\mathbf{H}_i$ one needs only elements of the matrix $\mathbf{Q}_i^T$.

Similarly, the vector $\mathbf{x}$ (and some auxiliary vectors in the implementation) is divided into p subvectors, each of the size n/p.

Now, the (4) can be written:

$$\begin{bmatrix} \mathbf{x}_1^{(k+1)} \\ \mathbf{x}_2^{(k+1)} \\ \vdots \\ \mathbf{x}_p^{(k+1)} \end{bmatrix} = \begin{bmatrix} \mathbf{H}_1 \\ \mathbf{H}_2 \\ \vdots \\ \mathbf{H}_p \end{bmatrix} \mathbf{x}^{(k)} \tag{5}$$

and, in another manner, for $i = 1, \ldots, p$:

$$\mathbf{x}_i^{(k+1)} = \mathbf{H}_i \mathbf{x}^{(k)} \ . \tag{6}$$

(6) is a formula for the block Jacobi method. All the equations in (6) are solved independently, so it is very suitable to parallelize for p processors – each of them solves one equation and then they exchange the resulting vectors $\mathbf{x}_i^{(k+1)}$ to build its new version $\mathbf{x}^{(k+1)}$.

However, the Jacobi method is rather slowly convergent, so in the presented algorithm, every computational node can employ the formula of the Gauss-Seidel algorithm instead of the Jacobi algorithm – using the newly obtained elements of the vector $\mathbf{x}_i^{(k+1)}$ (although only those which are stored in the same computational nodes) and hoping for the better convergence.

In other words, the algorithm can be described as a block Jacobi iterative method with solving inner blocks with the Gauss-Seidel iterative method.

In borderline cases the presented algorithm reduces to the pure Jacobi algorithm (for $p = n$) and to the pure Gauss-Seidel algorithm (for $p = 1$).

4 Parallel Algorithm

Before studying parallel algorithms it is important to consider the data distribution that will be used. An appropriate data distribution will determine the load balancing of parallel algorithms and the communicating operations involved.

4.1 Data Distribution

In the algorithm described later (in Sect. 3) the matrix $\mathbf{Q}^T$ is divided among cluster nodes horizontally – that is the matrix is divided into p rectangular submatrices $\mathbf{Q}_i^T$, each stored in the ith cluster node.

This matrix partitioning is called unidimensional row oriented. The matrix is divided into row blocks of about an equal number of non-zero elements. It is interesting to study the different methods for mapping these data on the processors. The choice of an appropriate mapping strategy will allow a good load balancing of parallel algorithms. In the present work, we use a direct mapping (one of the paritions is assigned to each processor) of data partitioning among the nodes. Each submatrix is a sparse matrix and takes part in the computations on its node (where it is stored). For distribution we use the algorithm proposed and studied by the authors at work [20].

4.2 Implementation Details

We propose an algorithm for homogeneous cluster environments. This algorithm is based on a message-passing paradigm (utilizing the language C and the standard MPI) and consists of one module for each of nodes. The algorithm presented below is just a skeleton, and the detailed implementation, such as data preparation, parameters passing, and so forth, might be different according to requirements of various applications.

The algorithm is composed of several steps. First, the starting information is acquired. Next, the matrix $\mathbf{Q}$ is generated in parts so that every node keeps only needed states. Next, in loop, in every node for its block we make a step of the Gauss-Seidel method.

The number of tasks is constant and there is exactly one processor for one task. The mapping of the tasks to processors is static, what means that it is done at the beginning of the program and do not change during the program execution. The amount of computation carried out in every task is almost the same (because the amount of data dealt with by the particular tasks is very similar).

More on implementation details can be found in [18].

5 Tests, Results and Comparison

5.1 Model Parameters

Because we were interested in investigating the algorithm, not the model itself, we fixed some parameters which are almost insgnificant to the accuracy and the performance of the algorithm, namely $\lambda_T = 0.02$, $\lambda_B = 0.03$ and $\mu = 0.002$. Other parameters are shown in Table 1, together with the matrices' poperties.

Table 1. Model parameters and matrices' sizes (N: size of the matrix; nz number of non-zeros; $d = nz/N$)

$q_{\mathrm{T}} = q_{\mathrm{B}}$	s_{T}	s_{B}	s_{TB}	N	nz	d
10	8	12	5	1 022	4 474	4.38
100	80	120	50	530 051	2 599 300	4.90
200	160	240	100	4 039 301	19 995 400	4.95
300	240	360	150	13 407 751	66 588 300	4.97

All the values are based on data from a real call center originated from [7].

5.2 Properties of Matrices

For a numerical solution of the model we generated the transition rate matrix $\mathbf{Q}$ on the basis of the transition graph (Fig. 2) numbering possible states. The matrix could be very big, so we employ some parallel, distributed algorithm for its generation (as we employed in [20]). Table 1 shows the parameters of the model and sizes of the matrices. We can see from the table, that all the tested matrices are very sparse (less the 5 elements a row).

5.3 Test for Convergence of Iterative Methods

The residual vector $\mathbf{r}$ of a system of linear equations $\mathbf{Ax} = \mathbf{b}$ is defined by $\mathbf{r} = \mathbf{b} - \mathbf{Ax}$. For an iterative method, the initial value for the residual vector, $\mathbf{r}^{(0)}$, can be computed by $\mathbf{r}^{(0)} = \mathbf{b} - \mathbf{Ax}^{(0)}$, using some initial approximation $\mathbf{x}^{(0)}$ of the solution vector. During successive approximations, the goal is to obtain $\mathbf{r} = \mathbf{0}$ (or rather as close to $\mathbf{0}$ as it is possible and necessary), which gives the desired solution $\mathbf{x}$ for the linear equation system. We would say that an iterative algorithm has converged after k iterations, if the magnitude of the residual vector $\mathbf{r}^{(k)}$ becomes desirably small. We chose for the convergence test to compare in every iteration the 1st norm of the residual vector with an assumed accuracy. The 1st norm is given by

$$||\mathbf{r}^{(k)}||_1 = \sum_{i=1}^{n} |r_i^{(k)}| \ .$$

In our tests we measured $||\mathbf{r}^{(k)}||_1$ for 500 iterations. Table 2 shows the 1st norm $||\mathbf{r}^{(k)}||_1$ for tested algorithms and matrices. We can see that the accuracy is significantly better for our algorithm (J+GS) than for pure Jacobi (J), and that for large matrices the accuracy is little dependent on the number of employed processors. Of course, the accuracy gets worse if we utilize more processors, because then the algorithm becomes "less Gauss-Seidel, more Jacobi".

5.4 Analysis of the Parallel Algorithm

The parameters of the computers used for tests have two processors Intel Quad-core 3.2 GHz each and 32 GB RAM connected with 1 Gb Ehternet.

Table 2. The 1st norm of the residual vector in 500th iteration for various matrices and both tested algorithms (J+GS: block Jacobi with inner Gauss; J: pure Jacobi)

nodes	procs. per node	1 022 (J+GS) [$\times 10^{-10}$]	1 022 (J) [$\times 10^{-3}$]	530 051 (J+GS) [$\times 10^{-4}$]	530 051 (J) [$\times 10^{-3}$]	4 039 301 (J+GS) [$\times 10^{-3}$]	4 039 301 (J) [$\times 10^{-3}$]	13 407 751 (J+GS) [$\times 10^{-3}$]	13 407 751 (J) [$\times 10^{-3}$]
1	1	7.05	1.23	1.98	1.31	1.57	5.77	3.02	6.41
1	2	9.66	1.23	1.98	1.31	1.57	5.77	3.02	6.41
1	3	13.5	1.23	1.98	1.31	1.57	5.77	3.02	6.41
1	4	16.4	1.23	1.98	1.31	1.57	5.77	3.02	6.41
1	5	23.3	1.23	1.98	1.31	1.57	5.77	3.02	6.41
1	6	28.0	1.23	1.98	1.31	1.57	5.77	3.02	6.41
1	7	28.0	1.23	1.98	1.31	1.57	5.77	3.02	6.41
1	8	31.1	1.23	1.98	1.31	1.57	5.77	3.02	6.41
2	8	134	1.23	1.98	1.31	1.57	5.77	3.02	6.41

The following characteristics were obtained (among others) and analyzed:

- the time of the execution on p processors (T_p);
- the speed-up ($S_p = T_1/T_p$);
- the efficiency ($E_p = S_p/p$).

All the characteristics are presented in Figs. 3, 4, 5.

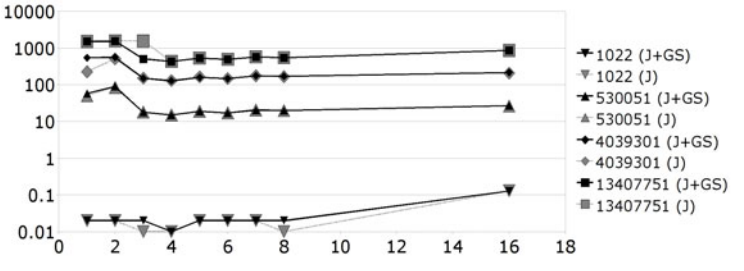


Fig. 3. The performace time (seconds, logarithmic scale) of the tested algorithms for various matrices sizes as a function of number of processors

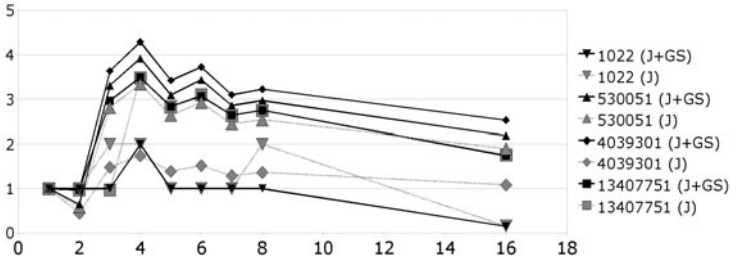


Fig. 4. The speed-up of the tested algorithms for various matrices sizes as a function of number of processors

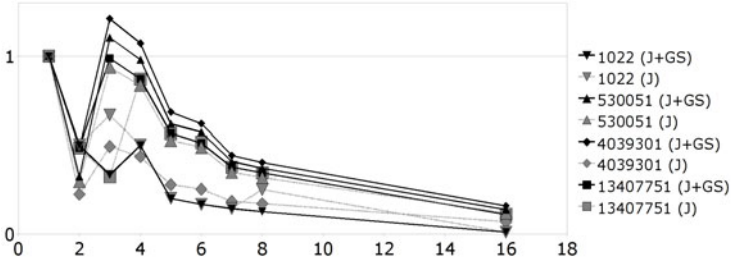


Fig. 5. The efficiency of the tested algorithms for various matrices sizes as a function of number of processors

6 Conclusion

As we can see, the accuracy of the algorithm is quite good (compared to the Gauss-Seidel method what is the case of J+GS on one processor), but the speed-up and efficency is not completely satisfying and the scalability of the algorithm is questionable. However, we are going to improve the algorithm and test it in another computer clusters consisting of the processors Itanium2 connected with InfiniBand 10 GB/s because the commuication (as always in parallel algorithms) is the most troublesome part of the algotithm. To improve it, the algorithm also could be refactored to be an asynchronous one.

References

1. Perrone, L.F., Wieland, F.P., Liu, J., Lawson, B.G., Nicol, D.M., Fujimoto, R.M. (eds.): Proceedings of the 2006 Winter Simulation Conference of Variance Reduction in the Simulation of Call Centers (2006)
2. Pichitlamken, J., Deslauriers, A., L'Ecuyer, P., Avramidis, A.N.: Modelling and simulation of a telephone call center. In: Proceedings of the 2003 Winter Simulation Conference, pp. 1805–1812 (2003)
3. Skaraczyński, T., Zoła, A.: Komponentowy system modelowania kolejek w zastosowaniu do telefonicznych centrów obsługi. In: Współczesne aspekty sieci komputerowych, Wydawnictwa Komunikacji i Łączności, Warszawa, vol. 1, pp. 89–96 (2008) (in Polish)
4. Skaraczyński, T., Zoła, A.: Optimizing employment in telephone call centers. Theor. and Appl. Inform. 20(1), 39–47 (2008)
5. Skaraczyński, T., Zoła, A., Bylina, J., Bylina, B.: Markovian method of modeling call-centers. In: Proceedings of CSIT 2008, Computer Science and Information Technologies, Lviv, pp. 51–54 (2008)
6. Bhulai, S., Koole, G.: A queueing model for call blending in call center. IEEE Trans. on Aut. Contr. 48, 1434–1438 (2003)
7. Gans, N., Koole, G., Mandelbaum, A.: Telephone Call Centers: tutorial. Review and Research Prospects, Manufact. and Service Oper. Manag. 5, 79–141 (2003)
8. Koole, G., Amndelbaum, A.: Queueing Models of Call Centers. An Introduction. Ann. of Oper. Res. 113, 41–55 (2002)

9. Deslauriers, A., L'Ecuyer, P., Pichitlamken, J., Ingolfsson, A., Avramidis, A.N.: Markov chain models of a telephone call center with call blending. Computers & OR 34(6), 1616–1645 (2007)
10. Ingolfsson, A.: Modeling the M(t)/M/s(t) Queue with an Exhaustive Discipline (submitted)
11. Whitt, W.: Engineering Solution of a Basic Call-Center Model. Manag. Sci. 51(2), 221–235 (2005)
12. Buchholz, P., Fischer, M., Kemper, P.: Distrbuted steady state analysis using Kronecker algebra. In: Proceedings of the Third International Conference on the Numerical Solution of Markov Chains (NSNC 1999), Zaragoza, Spain, pp. 76–95 (September 1999)
13. Dingle, N.J., Harrison, P.G., Knottenbelt, W.J.: Uniformization and hypergraph partitioning for the distributed computation of response time densities in very large Markov models. Journal of Parallel and Distributed Computing 64, 908–920 (2004)
14. Knottenbelt, W.: Distributed disk-based solution techniques for large Markov models. In: Proceedings of the Third International Conference on the Numerical Solution of Markov Chains (NSNC 1999), Zaragoza, Spain (September 1999)
15. Mehmood, R., Crowcroft, J.: Parallel Iterative Solution Method for Large Sparse Linear Equation Systems. Technical Report UCAM-CL-TR-650, Computer Laboratory, University of Cambridge, UK (October 2005)
16. Kwiatkowska, M., Parker, D., Zhang, Y., Mehmood, R.: Dual-processor parallelisation of symbolic probabilistic model checking. In: Proceedings of 12th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS 2004), pp. 123–130 (2004)
17. Zhang, Y., Parker, D., Kwiatkowska, M.: A wavefront parallelisation of CTMC solution using MTBDDs. In: Proceedings of International Conference on Dependable Systems and Networks (DSN 2005), pp. 732–742. IEEE Computer Society Press, Los Alamitos (2005)
18. Bylina, J., Bylina, B.: Merging Jacobi and Gauss-Seidel Methods for Solving Markov Chains on Computer Clusters. In: Proceedings of the International Multiconference on Computer Science and Information Technology, vol. 3, pp. 263–268 (2008)
19. Bahi, J.M., Contessot-Vivier, S., Coutier, R.: Parallel iterative algorithms. In: From sequential to Grid Computing, Chapman & Hall/CRC (2007)
20. Bylina, J., Bylina, B.: Development of a distributed algorithm of matrix generation for Markovian models of congestion control and its performance analysis on a computer cluster. In: Contemporary Aspects of Computer Networks, vol. 2, pp. 251–258, Wydawnictwa Komunikacji i Łączności, Warszawa (2008)

New Simplified HEED Algorithm for Wireless Sensor Networks

Rafał Pawlak, Bartosz Wojciechowski, and Maciej Nikodem

Institute of Computers, Control and Robotics, Wroclaw University of Technology
ul. Wybżerze Wyspiańskiego 27, 50-370 Wrocław

Abstract. We present a new simplified version of Hybrid Energy Efficient Distributed clustering algorithm. Our proposal reduces the communication overhead of the algorithm and achieves slightly better properties when compared to its predecessor. This is verified through extensive simulation. We also give a precise estimation of the intercluster communication range and the number of nodes that need to be deployed in the area in order to ensure that all nodes can send messages to the base station. This estimation is independent of the clustering algorithm used.

Keywords: wireless sensor network, HEED, clustering.

1 Introduction and Previous Work

Wireless Sensor Networks (WSNs) are extensively used whenever there is a need to monitor areas that cannot be accessed easily and to detect and alert hazardous conditions, such as fires, pollution or enemy approaching. Real life applications take advantage of the fact that WSN consist of hundreds or thousands of small devices communicating over the radio channel and battery powered. Nodes are equipped with sensors and use dedicated hardware and software to ensure their operation for many months without any attention. When deployed nodes of the WSN perform simple sensing tasks and send measurements to the base station (BS) that is a gateway to the outside world (e.g. Internet or a PC computer). Since area covered by WSN usually exceeds communication range of a single sensor, therefore sending messages to the BS requires that nodes cooperate and retransmit messages from each other.

Small dimensions of a single node, battery power supply and wireless communication widens practical application of WSN. However, restricted power source introduces variety of limitations and constrains that do not exist in computer, WiFi or other traditional networks. Problems and challenges such as routing, security, dependability and confidentiality, that are relatively simply solved in traditional networks, require new, clever, and effective solutions when addressed in WSN. The task is even more challenging since in order to preserve power, nodes use simple microcontrollers that on one hand incorporate various on-board functional modules (such as analog-to-digital converters, UART and SPI interfaces, general purpose pin outs and timers) but at the same time limit the efficiency,

A. Kwiecień, P. Gaj, and P. Stera (Eds.): CN 2010, CCIS 79, pp. 332–341, 2010.

slow down the operation and reduce computational capabilities. Low-power radio transceivers, that save energy on radio communication, yield short communication ranges thus introducing the need for dedicated routing protocols. Therefore, WSNs require algorithms and protocols that limit communication overhead, help to conserve battery and extend network lifetime.

A sound solution that enables to achieve this goal is to group neighboring nodes into clusters that reflect normal operation of the wireless sensor network and enable to aggregate data from neighboring nodes [1]. Each cluster consists of exactly one cluster head (CH) and zero or more regular nodes (RN). Regular nodes are responsible for measuring environment parameters and sending results to their CH while cluster heads are also responsible for collecting, aggregating and forwarding messages [2,1,3,4,5]. Low Energy Adaptive Clustering Hierarchy (LEACH) is one of the early proposals for effective organization of WSN [3]. It assumes nodes divide into clusters and take turns in being cluster head thus leading to even power consumption. Moreover it uses time division multiple access (TDMA) and direct sequence spread spectrum (DSSS) to minimise collisions in communication channel and reduce communication overhead further. This solution is sound but nowadays WSN transceivers don't use TDMA so it needs to be implemented manually, with additional costs. Moreover LEACH assumes that nodes can communicate over the whole network which limits its practical application since typical communication range varies between 70 and 100 meters.

TEEN and APTEEN [4] focus on time-critical applications, like event detection systems, so that nodes do not send all measurements to the base station but only those that fall into (or exceed) predefined range. This enables to put the transceiver into sleep mode for some periods of time and save battery power.

Hybrid Energy Efficient Distributed clustering algorithm (HEED) was proposed to address WSN that exceed communication ranges of a single sensor [5]. HEED addresses data sensing, aggregation and retransmission issues and aims for ensuring reduced power consumption and extended network lifetime. It also allows to use different communication cost measures like number of nodes in a cluster or average energy required to communicate with nodes in a cluster. Paper [5] also introduced the notion of intracluster and intercluster communication ranges and their relationship that ensures connectivity of the WSN. Later works by Lin et al. [6] and Huang et al. [7] gave another condition for the connectivity of clustered network and achieved smaller number of clusters in relatively few rounds, especially in sparse networks.

Without going into details, the original HEED algorithm [5] operates in three phases. In the initialization phase each node listens for broadcast messages from its neighbours and itself sends a message containing its communication cost. Also every node computes initial probability of becoming a cluster head based on the remaining energy. The probability is always bigger than some $p_{\min}$ to ensure that the algorithm terminates in a predefined number of rounds.

The second phase, called *Main Processing* executes in N_{iter} iterations. Each node terminates this phase when either its probability of becoming a cluster head CH_{prob} reaches 1 or becomes a regular node and joins some cluster. If a

regular node has more then one cluster head in its communication range it joins the cluster with the least communication cost to the cluster head. If there are no cluster heads nearby, the node decides to become a cluster head itself. After each iteration the probability CH_{prob} is doubled and once the value CH_{prob} is 1, the node becomes a final cluster head.

In the finalization phase, a node that is a final cluster head (is_final_CH set) or that has no final cluster heads in its neighbourhood, sends a broadcast message with its ID and cost, and becomes a final cluster head. If the node is not a final cluster head and has at least one cluster head in communication range, it joins the cluster with the least cost.

The original HEED allows to use different cost measures in order to determine cluster heads in a way that ensures the best properties of the network. Possibility to use variety of cost measures requires that all neighbouring nodes calculate them during the initialization phase. This requires that nodes determine their neighbours, exchange some data with each other and spend energy on additional communication that is required. On one hand initialization phase enables to determine different cost measures but is also expensive in terms of time and power. Moreover, in many situations initialization can be seen as surplus and can be omitted. As we present in this paper a simplified version of the algorithm can achieve the same properties while maintaining lower communication and thus lower power consumption.

2 Simplified HEED

This paper presents simplified version of the HEED algorithm – sHEED. Our proposal takes advantage of relative communication costs, determined during execution of the main processing phase, thus eliminating the need for the initialization phase. The advantage is that the communication cost can be estimated as a byproduct of regular communication, since most of WSN radios calculate received signal strength indicator (RSSI) for each message received. RSSI indicator can be used to estimate the communication cost.

Apart form simplifying the clustering algorithm and its verification we also give precise estimation of the intercluster communication range and the number of nodes required to be deployed in a spot area to ensure connectivity.

2.1 Network Model and Clustering Problem

In clustered network the RNs are responsible for measuring environment parameters and sending results to their CHs. Radio communication range of a RN is therefore relatively short since it is only required that every RN reaches its CH. For simplicity it is usually assumed that communication range of a RN, called intracluster communication range R_{c}, is constant. Regular nodes within R_{c} range from the CH constitute a cluster.

Cluster heads are normal nodes but they perform additional tasks being responsible for collecting, aggregating and forwarding data received from their

RNs and other CHs, to the BS. Since CHs communicate with other CHs, therefore intercluster communication range – R_t, has to be larger then R_c. Moreover, dimensions of a WSN usually exceed R_t thus communication to the BS has to be performed in a multi-hop way. Uneven power consumption between CHs and RNs is compensated by repeating the clustering and selecting CHs based on node's residual energy. Additional parameters are sometimes used to make CHs power consumption uniform [5]. Ensuring even power consumption targets extension of network lifetime defined as delivering measurements data from the area of the WSN. Repeated clustering combines simplicity and randomness of clustering algorithms to improve even power consumption. Despite the fact that clustering seems to be an effective method of WSN management there are several aspects that have to be addressed when designing the algorithm:

- Distributed algorithm – all nodes in the WSN share the same communication channel which may be a bottleneck of any centralized algorithms. Therefore, clustering algorithm has to enable neighboring nodes to select CHs and RNs on their own, reducing the communication.
- Network connectivity – when clustering algorithm finishes the whole WSN has to be connected i.e. at least one path from any node to the BS must exist. This can be ensured in various ways but trade-off between complexity, memory and communication overhead has to be always taken into account.
- Distribution of RNs in clusters – since CHs transmit messages from their RNs thus number of RNs within the cluster is essential. Ensuring equal number of RNs in each cluster aims at uniform power consumption by all CHs.
- Low communication overhead – there is no clustering without communication, however the number of communication has to be kept as small as possible.
- Casual usage – clustering is a process that prepares the WSN for normal operation and can be seen as surplus. Therefore, clustering should be used only when it is really needed.
- CH selection – since CHs are the most heavily loaded therefore clustering algorithm should prefer CHs that have higher residual energy.

All of above properties show that good clustering protocol has to ensure that a WSN covers the whole spot area and cost of clustering is unimportant when compared to the cost of the normal operation of the network.

2.2 The sHEED Algorithm

The simplified algorithm does not use the initialization and finalization phase as compared to the original HEED. However, the main processing is extended with measurement of RSSI for every message received and additional *join cluster* messages send at the end of clustering process. It is worth to mention that measuring RSSI is a standard procedure performed by all WSN radio transceivers and *join cluster* messages can be omitted in applications, where CHs don't need to know their RNs. The pseudocode of the proposed algorithm is given in algorithm 1. The proposed algorithm starts with every node computing initial

Algorithm 1. *SimplifiedHEED*

1: $CH_{\text{prob}} \leftarrow \max(C_{\text{prob}} \times \frac{E_{\text{residual}}}{E_{\text{max}}}, p_{\text{min}})$
2: **repeat**
3: **if** $(S_{\text{CH}} \leftarrow \{v : v \text{ is a cluster head}\} \neq \phi)$ **then**
4: $my\_cluster\_head \leftarrow least\_cost(S_{\text{CH}})$
5: **else if** $Random(0,1) \leq CH_{\text{prob}}$ **then**
6: Cluster_head_msg(NodeID, cost)
7: $CH_{\text{previous}} \leftarrow 1$
8: **end if**
9: $CH_{\text{previous}} \leftarrow CH_{\text{prob}}$
10: $CH_{\text{prob}} \leftarrow \min(CH_{\text{prob}} \times 2, 1)$
11: **until** $CH_{\text{previous}} = 1$
12: **if** $my\_cluster\_head \neq NodeID$ **then**
13: $join\_cluster(my\_cluster\_head\_ID, NodeID)$
14: **end if**

probability of becoming a cluster head: CH_{prob}. This probability depends on the residual energy of each node, thus effectively promoting nodes with higher energy reserve to become cluster heads. Similarly as in HEED, this probability can't be lower than some predefined value p_{min} to ensure the algorithm works in at most $k = \lceil \log_2 \frac{1}{p_{\text{min}}} \rceil$ iterations. At the begining of each iteration every node listens for messages from neighbouring cluster heads. If it hears at least one message, it joins the cluster head with the least cost. If there are no cluster heads in comunication range, the sensor becomes a cluster head with probability CH_{prob} and sends a broadcast message to neighbours with its node ID and cost. After each iteration the probability of becoming a cluster head is doubled. The algorithm ends for a given node, when the probability of becoming a cluster head was 1 in the previous iteration. This means that every node either becomes a cluster head, or joins some cluster.

The proposed algorithm satisfies all properties mentioned above. Precisely, sHEED is a distributed algorithm executed by each node of the WSN within R_{c} communication range of each node. It uses no general information about the whole WSN and selects the CHs based on nodes' own parameters (residual energy and communication cost estimated with RSSI) and local communication thus, ensuring low communication overhead and preferring CHs with higher residual energy. Simplified HEED also enables to ensure network connectivity. Notice, that all RNs can communicate with their CHs (otherwise they would become CH themselves). Network connectivity is thus ensured if there is at least one communication path between every CH and the BS. Using estimations presented in Sect. 3, putting $R_{\text{c}} = R_{\text{t}}$ one can determine the number of nodes that have to be deployed to ensure network connectivity with probability almost equal 1. Obviously this will give the lower band for number of nodes that may affect the network structure and may lead to networks that contain more CHs than RNs. Nevertheless, the sHEED algorithm gives the opportunity to ensure network connectivity.

3 Intracluster and Intercluster Communication Ranges

Ensuring network connectivity is one of the most important aspects of clustering algorithms and requires that number of nodes in the network, intercluster and intracluster communication ranges are determined. To be precise we need to define physical and logical clusters. *Physical clusters* are usually defined so that all nodes residing in the same cluster can communicate directly [5]. Physical cluster can be approximated by a square area of diagonal R_c, where R_c denotes intracluster communication range.

Logic cluster – or *cluster* for short, is defined so that it covers a single CH and all RNs that can communicate directly with it. Logic cluster is thus a circle area of radius R_c centered at CH. Logic clusters are created during clustering process that determines CHs, RNs and ascribes every RN to a single CH. Clustering requires that nodes of the WSN are deployed in such a way that they can communicate with at least one neighbor using R_c communication range. To achieve this it is typically assumed that there is at least one node in each physical cluster. Based on this assumption we can estimate the number of nodes n that have to be deployed with uniform distribution on a WSN area. Since the area of WSN is a square of side L thus it consists of $N = 2L^2/R_\mathrm{c}^2$ physical clusters. The probability that each physical cluster contains at least one node can be estimated as

$$P = 1 - N\left(1 - \frac{1}{N}\right)^n \quad . \tag{1}$$

If one takes

$$n = 2N \log N = 4\frac{L^2}{R_\mathrm{c}^2} \log \frac{2L^2}{R_\mathrm{c}^2} \tag{2}$$

nodes then this probability equals

$$P = 1 - N\left(1 - \frac{1}{N}\right)^{2N \log N} \geq 1 - N\left(\frac{1}{e}\right)^{2 \log N} = 1 - \frac{1}{N} \quad . \tag{3}$$

Since $L >> R_\mathrm{c}$ then $N >> 1$ and the probability (3) is almost 1. It means that deploying n nodes ensures that there is at least one node in each physical cluster.

Clustering algorithms organize the WSN in logic clusters and it is very unlikely that the distance between two CHs is smaller then R_c. This brings the need to define so called intercluster communication range R_t that represents the distance over witch cluster heads can send their messages. To ensure network connectivity R_t has to be bigger then R_c but it cannot be too large as it affects power consumption heavily. Paper [5] states that R_t has to equal $6R_\mathrm{c}$ but this distance is unnecessarily too large.

Lemma 1. *The longest communication range between two neighboring cluster heads is smaller than* $R_\mathrm{t} = 6R_\mathrm{c}$.

Proof. It was proven in [5] that there must be at least one CH in every square area of side $2.7R_\mathrm{c}$. We may then analyze two cluster heads: CH_1 is located in the

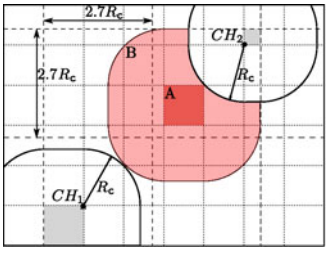


Fig. 1. Example showing that the longest communication range between two neighboring CHs is smaller then $6R_c$

lower left corner of such square area while CH_2 is located in the farthest corner of the neighboring area (see Fig. 1). These two CHs can be located in any point of their physical clusters (gray areas on the figure) but upper right and lower left corners are positions from which the distance between the nodes is the smallest. Otherwise, the maximal distance between nodes equals $6R_c$ approximately, from where the statement of [5] comes from.

However, even if nodes CH_1 and CH_2 are located as close as possible to each other, while still in their physical clusters, then at least one CH has to exist in area A or B. Since at least one node has to exist in A, then it is either a cluster head, or it has to be within R_c range from one. Hence, there is a CH in B and the longest communication range is smaller than $6R_c$. □

Work of Younis et al. [5] was later revisited by Lin and Tsai [6] who presented that $R_t = 2.7R_c$ is long enough to ensure connectivity of the clustered network. Unfortunately, such intercluster communication range does not ensure connectivity in every possible situation.

Lemma 2. *Intercluster communication range* $R_t = 2.7R_c$ *may be to short to ensure connectivity of clustered WSN.*

Proof. Figure 2 (left) presents a clustered WSN where each cluster contains exactly one node of the network and each is either a CH or belongs to some cluster. We assume intercluster communication range $R_t = 3R_c$ and show that such radius of communication does not ensure connectivity. In fact, connectivity is ensured within groups of cluster heads CH_1–CH_6 and CH_7–CH_9 but these groups are out of each other's communication range. Therefore, the whole network is not connected. □

Works by Younis et al. [5] and Lin et al. [6] give lower and upper bounds for the intercluster communication range. Unfortunately $R_t = 6R_c$ leads to increased, unnecessary power consumption, while $R_t = 2.7R_c$ may cause parts of the network to have no connection to the BS (Fig. 2 (left)).

In order to select intercluster communication range that ensures connectivity while keeping the power consumption low, we have to investigate the worst case scenario of nodes deployment. In such a scenario we want cluster heads to be as

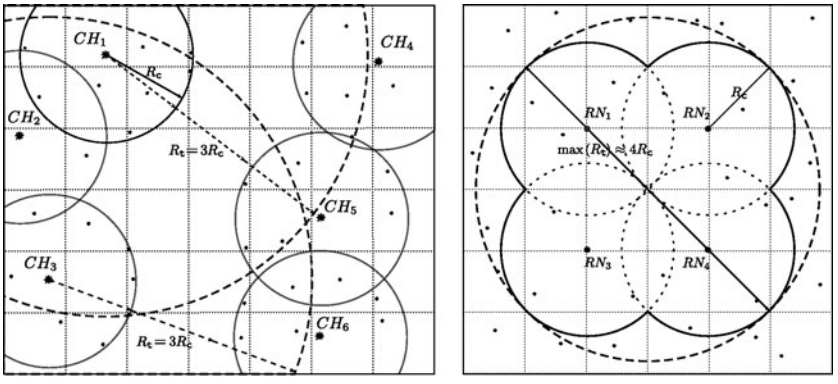


Fig. 2. Example showing that $R_t = 3R_c$ may be to short to ensure connectivity of the WSN (left) and that $R_t = 4Rc$ is long enough in the worst case scenario (right)

distant as possible. The worst case is when four regular nodes $(RN_1, \ldots, RN_4)$ belong to four different clusters and are located in the corners of their physical clusters (Fig. 2 (right)). Their cluster heads $(CH_1, \ldots, CH_4)$ are located at most R_c distance away. Therefore, in the worst case CHs are located on the clover-shaped line surrounding the four regular nodes. This line can be inscribed within a circle of diameter $4R_c$ which is the worst case intercluster communication range.

4 Evaluation of the sHEED Algorithm

We have analysed properties of both HEED and sHEED algorithms for a WSN consisting of 250 nodes deployed on the square area of side $L = 300\,\text{m}$. Taking into account maximal communication range of real-life sensor nodes we have assumed that $70 \leq \max(R_t) \leq 100$. For $3 \leq k \leq 6$ and $R_t = k \cdot R_c$, R_c falls between 15 m and 30 m. Using estimation (2) for this case, we get that between 104 and 265 nodes are required to ensure connectivity in such a network. All presented results were obtained with MATLAB simulator and are averaged over at least 15 trials, to take into account intrinsic randomness of node distribution and the algorithm itself. Trials for each set of values were repeated as long as there was a significant difference in all measured parameters. Difference was thought to be significant if the value of parameter in all trials differed from its value in last ten trials by more then 1%.

Figure 3 presents connectivity of the WSN for intracluster communication ranges $R_c = 15\,\text{m}$ and 25 m versus different values of cofactor k. For both values of R_c and $k = 4$ the network is connected and there is no difference between HEED and sHEED protocols. For smaller values of k some networks were not connected meaning that some CHs were not able to send messages to the BS. However, even for small values of $R_c = 15\,\text{m}$ and $k = 2.5$ more then 90% of randomly deployed networks were fully connected. It follows that in order to

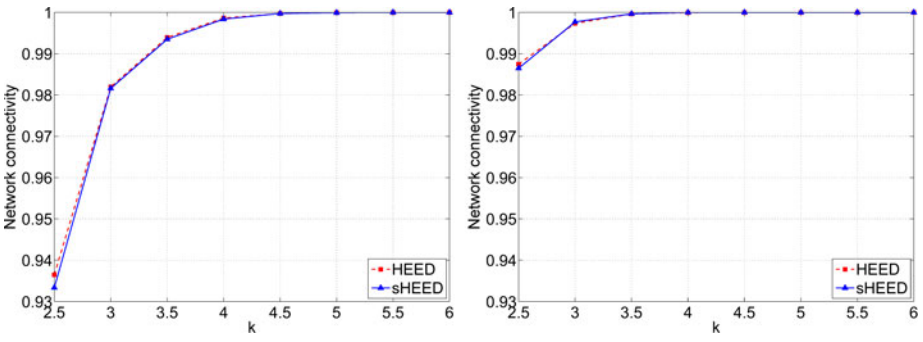


Fig. 3. Connectivity for R_c =15 m (left) and 25 m (right) vs. the cofactor k

ensure connectivity one can either select small R_c and large k or large R_c and small k. Simplified HEED algorithm does not affect the number of cluster heads in the network. The reason is that the number of CHs depends heavily on the intracluster communication range not the way CHs are selected. On the other hand, there is a difference in number of nodes in a cluster. Precisely, there is no difference in the mean value of nodes in a cluster (Fig. 4 left) but there is a difference in its variance (Fig. 4 right). Results of our simulations indicate that the number of nodes in clusters created with HEED algorithm varies more then when sHEED is used. It means that sHEED establishes clusters consisting of more or less the same number of nodes while in HEED there is a large number of small clusters and a few numerous clusters for which the number of nodes exceeds the mean value heavily. Equal distribution of nodes to clusters is an advantage of sHEED algorithm since it helps to ensure even power consumption throughout the whole WSN. Since variance of number of nodes in a cluster increases with R_c thus keeping R_c small helps to ensure even distribution of nodes.

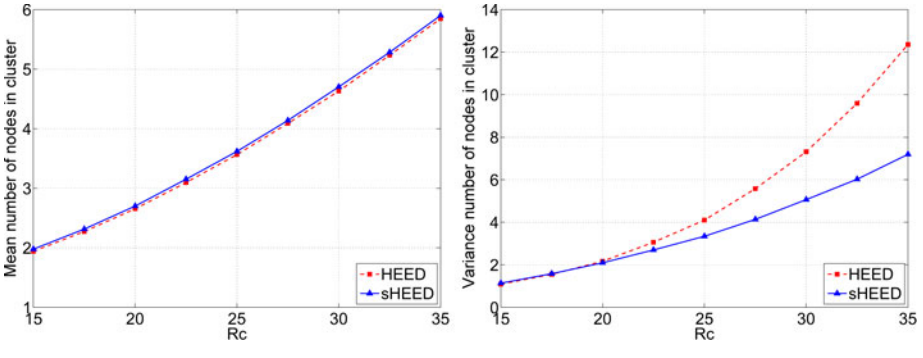


Fig. 4. Mean (left) and variance (right) of average number of nodes in clusters vs. R_c for $k = 4$

If clusters of equal size are preferred then it is better to use sHEED and HEED algorithm with small value of R_c and large k (e.g. $R_c = 20$, $k = 4$) rather than large R_c and small k (e.g. $R_c = 30$ and $k = 3$). Moreover, if communication costs are estimated using RSSI only then it is advisable to use sHEED since it has lower communications overhead, preserves power and establishes equal size clusters.

5 Conclusion

In this paper we propose a simplified version of HEED algorithm for WSN – sHEED. By means of theoretical analysis and extensive simulation we show that the properties of the algorithm for a wide range of reasonable parameters are as good as that of a much more complicated and communication demanding algorithm. We provide optimal parameters of radio communication range and number of nodes to be deployed for maximum network longevity and energy efficiency.

Acknowledgements. This work was partially supported by E.U. Regional Development Fund and by Polish Government, POIG.01.03.01-02-002/08-00.

References

1. Abbasi, A.A., Younis, M.: A survey on clustering algorithms for wireless sensor networks. Comput. Commun. 30(14-15), 2826–2841 (2007)
2. Lindsey, S., Raghavendra, C.S.: Pegasis: Power-efficient gathering in sensor information systems. In: Aerospace Conference Proceedings, vol. 3, pp. 1125–1130. IEEE, Los Alamitos (2002)
3. Heinzelman, W.R., Chandrakasan, A., Balakrishnan, H.: Energy-efficient communication protocol for wireless microsensor networks. In: HICSS 2000: Proceedings of the 33rd Hawaii International Conference on System Sciences, Washington, DC, USA, vol. 8, p. 8020. IEEE Computer Society, Los Alamitos (2000)
4. Manjeshwar, A., Agrawal, D.: Teen: a routing protocol for enhanced efficiency in wireless sensor networks. In: Proceedings 15th International of Parallel and Distributed Processing Symposium, pp. 2009–2015 (2001)
5. Younis, O., Fahmy, S.: HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. IEEE Transactions on Mobile Computing, 366–379 (2004)
6. Lin, C., Tsai, M.: A comment on HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. IEEE Transactions on Mobile Computing 5(10), 147–1296 (2006)
7. Huang, H., Wu, J.: A probabilistic clustering algorithm in wireless sensor networks. In: IEEE Vehicular Technology Conference, vol. 62, p. 1796 (2005)

Comparison of MPI Benchmarks for Different Ethernet Connection Bandwidths in a Computer Cluster

Jolanta Wrzuszczak-Noga and Leszek Borzemski

Institute of Informatics, Wroclaw University of Technology, Wroclaw, Poland
{jolanta.wrzuszczak,leszek.borzemski}@pwr.wroc.pl

Abstract. In this paper a contribution has been given about the investigations of the performance of the MPI communication in a computer cluster. The experiments had been executed with two different benchmarks that evaluated fundamental performance parameters of Ethernet communication network i.e. bandwidth, and latency in relation to different size of the messages, type of Ethernet network and forwarding schema.

Keywords: internetworking traffic benchmarks, PC cluster performance.

1 Introduction

This paper deals with MPI (Message Passing Interface) benchmarks applied in a blade-based hybrid computer cluster containing fourteen nodes, each consisting of two processors with four cores. Nowadays it is very important to guarantee a certain level of Quality of Service (QoS) or to predict the execution time of the given task in the distributed computational environment. Matching of that kind of requirements implies establishing the cluster with a sufficient computing power for performing calculations. To check the operation ability of a cluster system some tests (experiments) have to be performed based on suitable benchmarks. There are many kinds of benchmarks oriented on selected performance metrics i.e. latency, bandwidth, efficiency. They usually offer mostly very sophisticated options for shaping the schemas of the exciting traffic [1,2,3,4].

In this study the bandwidth variations will be tested and discussed depending on different network communication interfaces. Performance will be evaluated based on the experiment runs in the MPI environment on the real-life computing cluster [5,6,2,7].

Some MPI benchmarks suitable for clusters testing will be discussed in the next section. The third section presents the foundations of performed tests. In the fourth section a description of the experiment will be given. In the summary some concluding remarks will be presented.

A. Kwiecień, P. Gaj, and P. Stera (Eds.): CN 2010, CCIS 79, pp. 342–348, 2010.

2 MPI Benchmarks

MPI Benchmarks are programs, which are used to test the performance of machines or communication interfaces and their impact on the computing power of the related system. They are designed for defined features testing (i.e. latency, bandwidth, efficiency) and they are provided as open source tools or as commercial programs. Since years many tests were evaluated using MPI libraries i.e. Ethernet versus Myrinet or High Performance Computing for AIX using InfiniBand where MPI Communication was compared [8,9]. Topics of measuring MPI Send and Receive overhead application availability was discussed in [10] and MPI application in grid were also presented [11]. Measurement of cluster metrics is used to assure optimal system utilization.

In our work two different benchmarks will be considered: OMB-3.1.1 and Phloem [12,13,14,15,4].

OMB-3.1.1 was developed by Network-Based Computing Laboratory Department of Computer Science and Engineering, Ohio University and designed for testing the following performance parameters: latency, bandwidth, bi-directional bandwidth, multiple bandwidth/message rate, multi-pair latency and broadcast. These tests have been developed in the C language based on MPI environment [14].

The *latency test* is based on the ping-pong fashion, so the time of communication between sender and receiver is measured. At the beginning, the sender sends a command message of a certain data size. After the receiver gets the whole message, it starts to send the response message. This process is carried out for 10 000 iteration and than the average time is obtained. The test uses different data size (i.e. from 1 B to 1 048 576 B). The results are given in μs [14].

The *multi-pair bandwidth and message rate* test is based on multiple pairs of processes and evaluates aggregate unidirectional bandwidth. Every process sends a fixed number of messages back-to-back to the paired receiving process before waiting for a reply from the receiver. This test is executed through several iterations. This benchmark is designed for different size of messages and for different number of processes [14].

Phloem has been designed by Lawrence Livermore National Laboratory in Canada and includes subsystem functionality and performance tests. It was written in C language using MPI communication interface. This MPI benchmark evaluates the stability of various MPI aspects: performance, including interconnect messaging rate, latency, aggregate bandwidth and collective latencies under heavy network loads [13].

The *mpiBench Allreduce latency* Phloem test selects the maximum latency from all subpartition (processed) processes performing the MPI communication. This test for a given partition, iterates 1 000 times over the MPI_ALLREDUCE operation utilizing the MPI_COMM_WORLD communicator. The maximum Allreduce latency is the maximum over all MPI tasks of the individual MPI task Allreduce latency latencies. The results are provided in μs [13].

The purpose of the *SQMR (Sequoia Message Rate Benchmark)* test is to evaluate the maximal message rate of a single computer node. The first num_cores ranks are expected to reside on the 'core' compute node for which message rate

is being tested. After that, the next num_nbors ranks are neighbors for the first core rank, the next set of num_nbors ranks are neighbors for the second core rank, and so on. The communication sender receiver is based on different message size (from 1 B to 4 MB). This test iterates 4 096 times and gives the results in MB/s [13,16].

3 Test Setup

The idea of experiments was to test presented benchmarks for different networks, namely 1 Gbps and 10 Gbps Ethernets. The benchmarks (OSU-3.1.1 and Phloem) under consideration are designed to test different size of data. They iterate many times, so the influence of single interference could be excluded. During test two features are measured: the latency [μs] and the bandwidth [MB/s]. The first performance index was defined as latency for test 1 and 3 and the second performance index was defined as bandwidth for test 2 and 4.

The first test – latency from benchmarks OSU – was running for two processes for different message sizes (from 32 B to 1 048 576 B) and the second test – multi-pair bandwidth and message rate test was performed on 64 processes for message size from 1 B to 65 536 B.

The third test – allreduce latency from Phloem benchmark was executed on 64 processes for following message size: from 8 B to 2 048 B, and the fourth test the Sequoia Message Rate Benchmark was running on 64 processes for message size from 1 B to 32 768 B.

Each test had been performed for 1 Gbps and 10 Gbps Ethernet connections.

The structure of the cluster is shown in Fig. 1. The cluster features the IBM BladeCenter H technology with 14xHS21 blade servers and 2x Nortel 10 Gb blade switches and 2x Cisco 1 Gb blade switches. Testing installation consists of fourteen servers – working nodes (onyx01, onyx02,..., onyx14) and one master node (Opal) with the Centos 5.2 operating system. Each node has 8 GB RAM and two 3 GHz Intel Xeon x5365 CPUs [17].

Each cluster node has three network interfaces, two are connected to 10 Gbps and one to 1 Gbps network.

4 Experiment

The experiments have been performed in the Distributed Computer Systems Laboratory at the Institute of Informatics, Wroclaw University of Technology.

Two performance indexes: latency and bandwidth have been evaluated and compared for two different benchmarks to evaluate performance of the network message forwarding in our cluster environment.

Two performance indexes: latency and bandwidth have been evaluated and compared for two different benchmarks of performance of the network message forwarding in the cluster environment.

Before running benchmarks all non system processes had been killed, so the benchmarks were not disturbed from other processes and the network was not loaded from any unregistered processes.

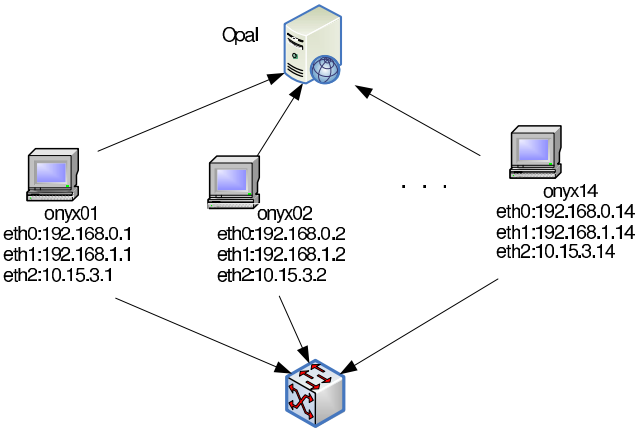


Fig. 1. Schema of the cluster

The first phase of the experiment has based on four succeeding tests with communication link of 10 Gbps. After this phase the network interface 10 Gbps had been put down (from the managing node Opal) and four tests as described in Sect. 3 had been performed.

The results of the related experiments are presented in Figs. 2–5. Figure 2 and Fig. 4 show the latency in [μs] versus size of the message whilst Fig. 3 and Fig. 5 demonstrate the bandwidth [MB/s] as a function of message volume [Bytes]. The line with squares depicts a performance index for the 10 Gbps network and the line with small rhombus represent the value of performance index for 1 Gbps respectively.

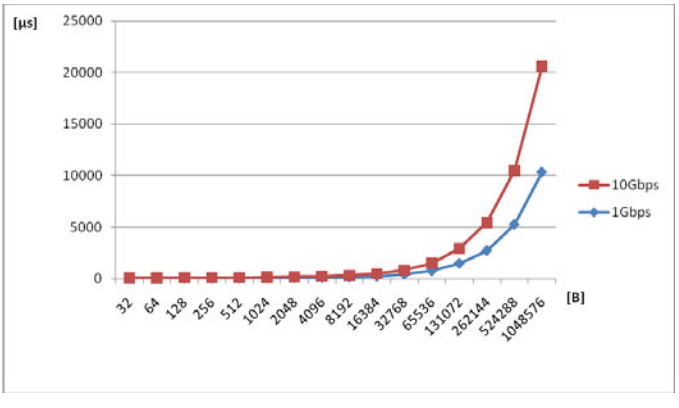


Fig. 2. Latency [μs] versus message size for OSU-3.1.1 latency test

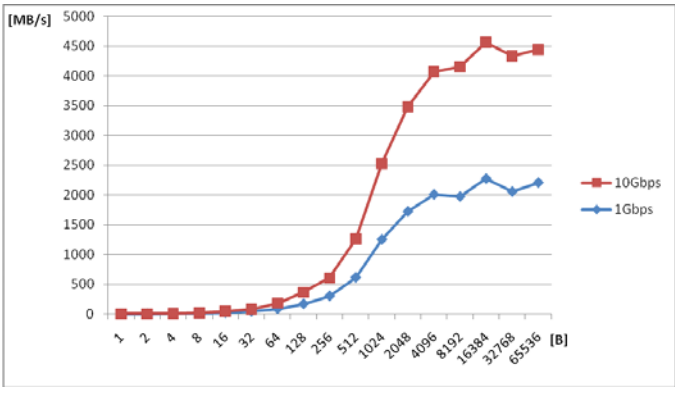


Fig. 3. Bandwidth [MB/s] versus message size for OSU-3.1.1 multi-pair bandwidth and message rate test

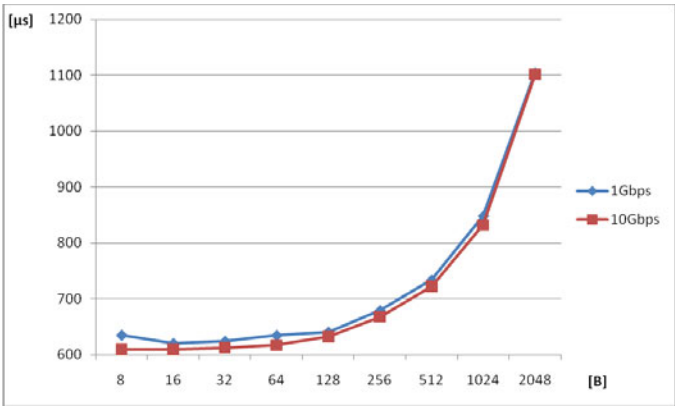


Fig. 4. Latency [μs] versus size of message for Phloem mpiBench Allreduce latency test

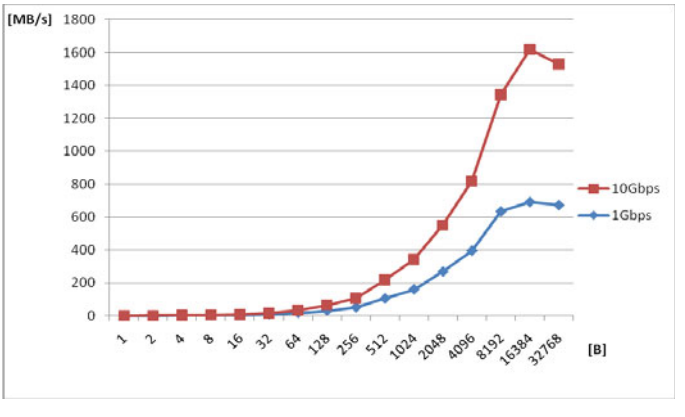


Fig. 5. Bandwidth [MB/s] versus size of message for Phloem SQMR test

5 Conclusions

Analyzing the behavior of latency for two studied benchmarks (OSU latency and Phloem mpi allreduce latency) it could be observed from Fig. 2 and Fig. 4 that the latency increases versus message size in both cases. For OSU benchmark the difference between 10 Gbps and 1 Gbps tests was bigger comparing to the Phloem latency benchmark. The latency for OSU benchmark is changing in a different way for both networks when the message size is more than 3 KB. In case of Phloem benchmark the latency functions for both networks have similar shape. The difference between latencies was the same (about 20 μs) and didnt depend on message size sent.

The behavior of the second performance index the bandwidth is shown in Fig. 3 and Fig. 5. Both graphs are increasing functions of bandwidth versus message size. It was noticed, that the difference between network bandwidths was speeding up for data size more than 512 Bytes. The biggest difference between networks for OSU benchmark is about 2 000 MB/s and for Phloem is about 1 000 MB/s.

To summarize, the use of a network which is ten times faster does not guarantee that the benchmark could reach 10 times better performance There are other parameters which may influence the result, for example number of frames sent, time of saving information on the disk drive, and many others.

Acknowledgments. This work was partially supported by the Polish Ministry of Science and Higher Education under Grant DPN/N181/COST/2009.

References

1. Bailey, D.H., Dagum, L., Barszcz, E., Simon, H.D.: NAS parallel benchmark results. In: Supercomputing 1992: Proceedings of the 1992 ACM/IEEE Conference on Supercomputing, pp. 386–393. IEEE Computer Society Press, Los Alamitos (1992)
2. Borzemski, L.: Load Balancing in Parallel and Distributed Processing of Tree-Based Multiple-Task Jobs. In: Proceedings Euromicro Workshop on Parallel and Distributed Processing, January 25-27, pp. 98–105 (1995)
3. Gropp, W., Lusk, E.: Reproducible Measurements of MPI Performance Characteristics. In: Margalef, T., Dongarra, J., Luque, E. (eds.) PVM/MPI 1999. LNCS, vol. 1697, pp. 11–18. Springer, Heidelberg (1999)
4. Underwood, K.D., Brightwell, R.: The impact of MPI queue Osage on message latency. In: Proceedings of the International Conference on Parallel Processing (ICPP), Montreal, Canada (August 2004)
5. Banikazemi, M., Govihdaraju, R.K., Blackmore, R., Panda, D.K.: MPI-LAPI: an efficient implementation of MPI for IBM RS/6000 SP systems. IEEE Transactions on Parallel and Distributed Systems 12(10), 1081–1093 (2001)
6. Barrett, B.W., Hemmert, K.S.: An application based MPI message throughput benchmark. In: Cluster Computing and Workshops, CLUSTER 2009, pp. 1–8. IEEE, Los Alamitos (2009)

7. Borzemski, L., Wrzuszczak, J.: Multidecision making in clusters as an illness recognition problem. In: Information Systems Architecture and Technology, ISAT 2007, pp. 241–249. Wyd. Politechniki Wrocławskiej, Wrocław (2007)
8. http://www.cs.rice.edu/CS/Architecture/docs/majumder-lcr04.pdf
9. http://www-03.ibm.com/systems/resources/systems_deepcomputing_pdf_sptiHPCAIXIBperformancewp101107.pdf
10. http://www.docstoc.com/docs/4379568/Measuring-MPI-Send-and-Receive-Overhead-and-Application-Availability-in-High-Performance-Network-Interfaces
11. http://www.public.iastate.edu/~coti/download.php?file=CHC09.pdf
12. Borzemski, L., Wrzuszczak, J., Kotowski, G.: Management of Web service delivering multimedia files based on the bid strategy. In: Information Systems Architecture and Technology ISAT 2008, pp. 13–23 (2008)
13. https://asc.llnl.gov/sequoia/benchmarks/PhloemMPIBenchmarks_summary_v1.0.pdf
14. http://mvapich.cse.ohio-state.edu/benchmarks/
15. Mamidala, A.R., Kumar, R., De, D., Panda, D.K.: MPI Collectives on Modern Multicore Clusters: Performance Optimizations and Communication Characteristics. In: 8th IEEE Cluster Computing and the Grid, CCGRID 2008, pp. 130–137 (2008)
16. Lee, I.: Characterizing communication patterns of NAS-MPI benchmark programs. In: SOUTHEASTCON 2009, pp. 158–163. IEEE, Los Alamitos (2009)
17. http://www.redbooks.ibm.com/Redbooks.nsf/RedbookAbstracts/sg247523.html?Open

Blind Signatures in Electronic Voting Systems

Marcin Kucharczyk

Silesian University of Technology, Institute of Electronics,
ul. Akademicka 16, 44-100 Gliwice, Poland
marcin.kuchraczyk@polsl.pl

Abstract. Electronic voting systems are created to facilitate the election, accelerate voting and counting votes and increase turnout for the election. The main problems of the electronic systems are related with the assurance of system security (cryptographic security and protection against frauds) and a guarantee that the voters are anonymous. The authorization of voters and the anonymity of them seem to be contradictory, but it is possible to create such a system where the both requirements are met. Some solutions, the most common e-voting systems, are presented in the paper. These models are adjusted to the demands and implemented in the polls system for quality of teaching evaluation where anonymity is very important. The blind signature protocol, model directed to user's anonymity, is a very good solution of the authorization and anonymity problem in the polls system with remote access only, as it was described in the paper.

Keywords: electronic voting, system authorization, user anonymity, blind signatures.

1 Introduction

In democracy, in order to recognize the will of majority, elections are carried out. Elections must be free and fair, i.e.:

- each citizen (entitled to vote) has a right to participate in elections;
- each voter casts the same amount of votes;
- in many cases a voter has a right to remain anonymous during voting.

The above mentioned rules refer not only to general voting on national (regional) level but also to minor groups which belong to certain determined environments. Such groups are formed by, among others, academic environments. Within a given University, election for the authorities of the University, faculties, institutes and departments are carried out. Also, the polls are performed. A particular type of the poll is evaluation of the quality of education, when the students estimate their lecturers.

Voting carried out in the academic institutions is subject to the same rules as general election. A list of persons authorized to vote is limited, depending on the type of election. Limits refer to the membership of voters in the academic

A. Kwiecień, P. Gaj, and P. Stera (Eds.): CN 2010, CCIS 79, pp. 349–358, 2010.

structures. After proving the authorization, a voter has a right to express himself anonymously, which may be a condition for casting a vote, according to the voter's will. Anonymity of election must assure that the voting is representative.

2 Electronic Voting

Definition of electronic voting covers a use of electronic devices on any stage of election – during voting or *only* in the process of votes counting.

Creation of the electronic system for votes counting does not encounter a protest from the public opinion. Significant acceleration of the counting process and presentation of the results is an unquestionable advantage of the computing machines. Assuming that the frequently used statement: *secure computer is a computer cut off from the network or turned off* is correct, introduction of the computer connected to the network into the election, decreases safety of the whole system. But there are still ballots, which may be collected, calculated and compared with the results presented by the computer, and hence any attempts to manipulate with the results can be detected. This argument will be irrelevant if the voting process is also conducted in the electronic way. On the other hand, a vision of the efficient elections (elimination human factor during counting of votes), decrease of costs (no polling stations, no paper ballots) and increase of the voter turnout (each person can vote from home) makes the electronic voting (e-voting) a subject of analysis and scientific researches [1,2,3,4].

A more significant undertaking related with voting is a poll to test the academic staff related to the evaluation of education quality. Detailed orders of the university or technical university define a scope of questions the respondents (students) answer to, and the regulations concerning access to the results of polls (availability for public or only for the academic authorities and persons to whom they concern). Irrespective of the scope of questions and availability of results, the requirements for the polls are similar to those of fair and free elections:

- Only students who attended the course led by a teacher to be estimated should participate in the poll.
- Number of votes for each student is limited. He may cast only one vote for a chosen professor and his lectures, but he can evaluate many professors and their lectures at the same time.
- Voting should be anonymous. The teachers who are subject of the questionnaires should have no access to information concerning voting students (the information shouldn't be available to anybody).

The last condition is seemingly contradictory to the previous ones, but it is possible to create such a system for the voter to remain anonymous in the process of electronic voting (even Internet voting) in spite of the necessity to confirm the authorization in the system.

3 Users Authorization

Authorization of user during a standard voting consists on identification with the use of a document, that confirms the identity in front of the member of electoral

committee in the polling station. The committee's task is to check if the person who presents a document is its holder, and on the basis of the data from the document to check in the base (printed list) whether he or she is authorized to vote in this polling station.

There is a double authorization in this case. The liability of the identity document and inspection of details which confirm its authenticity are subject to verification. The identity of the person who presents the document is also checked against the photograph on the document. In the European Union documents with electronic systems are being introduced. Apart from the alphanumeric data they contain biometrical data. Verification of identity when using such document may be realized without the presence of the human being. Electronic system of biometrical data verification are becoming more common. However, they are not, just like human being, infallible. These systems are characterized by false approval or false rejection determined factors.

Combining devices for verification of the document authenticity and identification on the basis of biometric data with the opinion of a member of the electoral committee, shall practically eliminate the possibility to cast a vote by an unauthorized person. In case of remote voting and authorization through the Internet, actually only one of the above authorization proceedings may be executed. It is possible to check the authorization data of the person who is trying to use the voting system: number of the identity document, data included in the document or special data ascribed to a user (the password). Verification, if a person who logins into a voting server is a holder of the data used is a separate issue. It is a wide field of activity for people who will be trying to falsify voting results. It may lead to a voting market, which is actually a sale of the rights for voting. Presence of the person authorized for voting at the moment of voting may be confirmed when all the voters are equipped with the readers of the biometrical data like fingerprint, eye print or voice verification [5,6].

In the first general election, where it was possible to vote through the Internet, election in Estonia in 2007 (in 2005 Internet voting was carried out during local elections), citizens gained a possibility to vote on-line, but they could also vote by traditional method, by casting a vote into the ballot box [7]. The users were authorized using asymmetrical cryptography. Standard X.509 was used, where a user has two types of key, private and public, which have a form of certificate [8]. Each citizen of Estonia has these certificates recorded in the integrated circuit located in the identity card. Use of the X.509 certificates, which are considered as safe [9], assures a higher level of system trust in authenticity of the voters than the simple use of a username and a password. It will make an access to the system by unauthorized persons more difficult and assure high level of safety to cast votes. Nevertheless, often it is sufficient to authorize with the username and password only. The polls system for evaluation of the quality of education within the University is a good example.

Let's assume that the problem with authorization of the system user is solved. The security level related with the access is considered as sufficient. Persons who intend to get an access to the system have rights to vote and can confirm the

rights. System acknowledges that the appropriate person performed authorization. How the anonymity of this person can be assured?

4 Voter's Anonymity

Voter's anonymity is not always a necessary element in the electronic voting system. For example in parliament voting, the whole proceeding is public. However, there are systems where the anonymity is as important as the confirmation of the voting authorization. Such are voting in general, parliament and president, election. Anonymity in this type of voting is guaranteed by appropriate provisions in acts and even by the Constitution. In the case of voting to evaluate the lecturers at the University by the questioned students, anonymity is very important too. If a student is not sure that he fills the questionnaire anonymous, the presented opinion on the chosen professor or lecturer may be unreliable or incomplete.

4.1 Anonymity in a System with Direct Authorization

Anonymous casting of vote means, that the person who is counting a vote does no have information which would identity the voter. This may be achieved even in the system with direct authorization when the votes are sent together with the signature of the voter. System user is not voting anonymously, but the creators of the voting system assure that the information about the voter's identity will not be used to identify a person for each vote in the system. This assurance is also a guarantee that no-one (not only persons who are counting the votes) will check what was the content of the vote of particular voter. In Figure 1 voting procedure is described. Vote, encoded with the public key of the polling station is signed by a system user, who already proved the identity while logged into the system. A vote in such a system is sent to the vote collecting system. The system is checking the correctness of the signature, using voter's public key and the signature is removed from the vote. The signature is moved to a trash bin and the encoded vote is sent to the polling station which having a private key may decode it and calculate for the election results.

If all the guarantees of the system creators are realized and voting is carried out in a way presented in the Fig. 1, the voter may be sure that he voted anonymously. Polling station (persons/devices which count the voters) is not able to assign a vote to a person. Still it is known, that there is information somewhere in the procedure which connects the content of the vote with the voting person.

General elections in Estonia are an example of the use of this model [3]. Electronic votes are kept in the system until voting in regular polling stations is finished and only then they are deprived of the signatures and decoded by the polling station computers. It enables a change or removal of the vote by the voter and verification if the vote remains in the system and is unchanged.

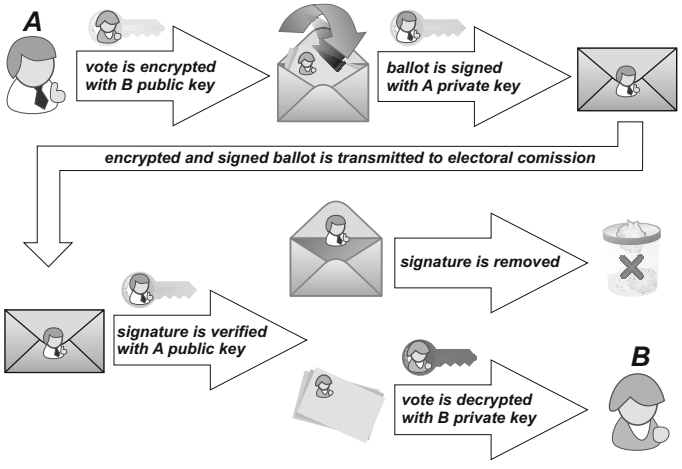


Fig. 1. Voter (A) trusts that his signature is removed before counting the vote by electoral committee (B)

An example of similar voting system is a procedure available for students using USOS system (http://usos.edu.pl/) on a few Universities in Poland. In USOS system, the procedure of user's authorization is simplified. In order to enter into the system, username and password are required. Once the questionnaire is completed it is placed in the data base without the information about the person who filled it in. It is impossible to re-fill the questionnaire that has the same properties: teacher and subject, because the system remembers that given student has already filled the questionnaire with those parameters.

Both examples, presented above, base on the trust of the system user that the voting procedure is realized in accordance with the system creators. This is not a one-side trust. The polling station that is counting votes must believe that the person voting at a distance not only has appropriate authorization data but uses them by himself. The inconvenience of above voting systems is lack of verification by a voter after the termination of voting, whether the vote was calculated and whether it is the same as the vote cast (just like in the traditional voting). Violation of anonymity by a slight change of the system software is very easy.

4.2 Authorization of Voter with Anonymous Identifiers

Voting through the Internet is the most advanced form of electronic voting. The voter is authorized by electronic devices and casts a vote remotely from any place, using Internet public network, the votes are counted by the computer system. In order to assure that the voter is anonymous, we should resign from the authorization (just like in anonymous polls available on some Internet portals)

or separate the procedure of identity verification from the authorization which is required to access to the system.

Computers are very useful tool during voting, especially when collecting and counting the results. The more voters participate, the significance of machines is greater, especially in respect of the time that passes from the conclusion of voting to publishing the results. At present, elections are electronic at the stage of votes counting, though there are countries where voting is realized using electronic devices in regular polling stations. Person, authorized for voting, comes to polling station, confirms their authorization by showing at the polling station the identity card and gets entitlement to cast a vote using electronic voting device (computer) which is made available to the voter. The voting act and the authorization of the voter are independent.

Nowadays, electronic voting devices are used in some countries, i.e. the United States, Brazil, India. The voting procedure besides the mathematic complexity related with the cryptography should be clear for voting persons, polling stations members and intermediaries who control the compliance with the voting procedures. The devices with closed source code are difficult to control and understand for anybody so the voting devices have been decommissioned after public concerns in the Netherlands (used before on a large scale) and in Ireland (the devices were purchased but they were not approved for use after the audit).

Similar idea of voting system is used in the poll system for measurement of education quality at the Silesian University of Technology created by the Author [10]. The priority of creating the system was the anonymity of voters, in order to assure the reliability of the questionnaires. Access to the system required identifier which is not related with the student data – it is the random string of letters and numbers called token. Following confirmation of the student's membership in a group of eligible voters (proving the student card), the student draws a token which authorizes the student to enter into the poll system. Number of the token, on the basis of the entry in the data base, limits the scope of voting, i.e. a list of available lecturers and the courses on questionnaires.

Because the token which gives authorization for voting does not depend on the personal data of the student who is using it, voting result is recorded together with the token. It is not necessary to remove voter's signature from the ballot (like in the Fig. 1). It does not violate the anonymity of the voter and enables editing of the questionnaire filled. When the voting is closed, the voter can check if the questionnaire is still in the system and if it was not changed.

A simple voting procedure enabled through the Internet in the mentioned poll system maintaining the requirement of system user authorization and the anonymity. Due to the importance of the voting results, general elections require better cryptographic security. Identifiers using public key infrastructure and X.509 certificates should be created. A voter who wants to take advantage of the possibility to vote through the Internet comes to the office or the polling station at least a day before the elections, draws an identifier from the ballot box and confirms with his signature the receipt of taking the identifier. The identifier

will enable voting for particular candidate in the election from any terminal. Signature on the list will disable the possibility to vote once again in the polling station. So far, there is no possibility to obtain another identifier in case it is lost. The mentioned polls system for students includes the same restrictions.

The above method has a significant disadvantage for voters: they still need to go to the polling station to obtain the identifier for authorization. The solution for the identifiers distribution inconveniences may be combining the two earlier presented authorization methods. To get the access entitlements user need to identify himself also in the electronic system. Using the remote connection the voter places electronic signature on the election list and receives anonymous token which give him rights to use a voting system. Such a combined access model was implemented in the system used at the Silesian University of Technology. The local LDAP authorization connected with the student's course services called SOTS (`http://sotsinfo.polsl.pl/`) is used for authorize the student. The obtained token authorizes user for fill in the questionnaires. It need to be remarked that, in the modified system, again it is easy to break anonymity of users by slight software change like in the system with direct authorization.

The use of additional identifier on one hand eliminates the problem of maintaining anonymity during the remote controlled voting and gives the possibility for voter's access control in the system, but at the same time opens the possibility for the vote trade. This form of electronic voting system has built-in option for creating the vote proofs, also after election time is over.

4.3 Blind Signatures in the E-Voting System

Two models of electronic voting systems using remote only access were presented. Both of the systems are fully anonymous but also the anonymity break is easy in both of them. The information connecting person with the vote (the first case) or with the identifier (the second case) is not recorded, but there is a search for such a system, where it would be impossible to break the anonymity. Use of the blind signatures protocol, designed and presented for the first time by Chaum [11], may be a solution of the problem. The protocol's idea is as follows:

1. A voting person prepares a message m and than encodes it using random value (blinding factor) r. It results in encoded message m'. Message m' should be created in such a way, as to make its deciphering to the original message m impossible (precisely: very difficult).
2. Message m' is delivered to the voting institution for the electronic signature. Person who requests signature is a subject to authorization and only after the successful one, the authorization center is signing the message, which results in signed message s'. Additionally, during the authorization it is possible to collect data which make it impossible to retrieve the signature.
3. The voter removes from the received message s' a random value r and the result is a message s, which is a digitally signed original message m.

4. The voter sends message m and its signed version s to the voting system, which checking its own signature can confirm the authenticity of the signature and the conformity of message m with the signed message s.

For mathematical example, using RSA algorithm, the procedure is as follows:

1. $m' = (m \cdot r^e) \bmod n$, where (e, n) is a public key of the authorization center;
2. $s' = (m')^d \bmod n$, where (d, n) is a private key of the authorization center;
3. $s = s' \cdot r^{-1} \bmod n$;
4. $s \equiv m^d \bmod n$, because in RSA algorithm: $r^{ed} \equiv r$, so $s' \cdot r^{-1} = (m')^d \cdot r^{-1} = (m \cdot r^e)^d \cdot r^{-1} = m^d \cdot r \cdot r^{-1} = m^d$, which is an original message signed by the authorization center.

The suggested use of blind signatures, most often means encoding and signing of the election votes [12]. Voter casts a vote anonymously using one system. The vote is locally processed (encoded and encrypted by a defined hash function) and in this form is send for a signature to other system, which is authorizing the voter. Obtaining signature is a condition for proper authorization of the voter, i.e. verification of the identity and the scope of rights for voting. At the same time e-voting system records, that the voter used his rights and blocks the possibility to vote more than once. After the removal of the random component of the message from the signed request the voter uses the voting system once again, where he sends his vote and the signed version. The presence of signature confirms the authorization of the voting person for participation in the elections.

The difference between student's polls and general elections is the right for multiple voting. Poll system allows the student to cast just one vote for a chosen professor and lecture, but at the same time the student casts votes concerning a dozen of professors and their lectures or classes. If the above method of using the blind signatures protocol will be implemented in the polls system then each vote should be blinded and signed before it is cast, so the person who is using the system would have to carry out the signing procedure many times. The solution is a use of blind signatures for the approval of the identifier required for authorization in the voting system [13]. System model operating on this basis is presented in Fig. 2.

The procedures of obtaining the identifier, signing it and authorization in the voting system are independent. Just like in previous model: creation of a vote, signing the vote and voting. As a result of this independence, each stage may be realized on different computer and at different time. It makes it more difficult to trace the user's identity on the basis of the voting time or the location.

The blind signature algorithm was implemented in the poll system for education quality evaluation at Silesian University of Technology. The access token hashed and encrypted with the blinding factor is signed using RSA algorithm. The offline application, with public available source code, is used for blinding the token and unblinding the signed one. The anonymity of the voter is assured by the random factor r and blinding algorithm used in the procedure. Nevertheless, the cost of the higher anonymity of the user is more complex, so less convenient procedure of voting.

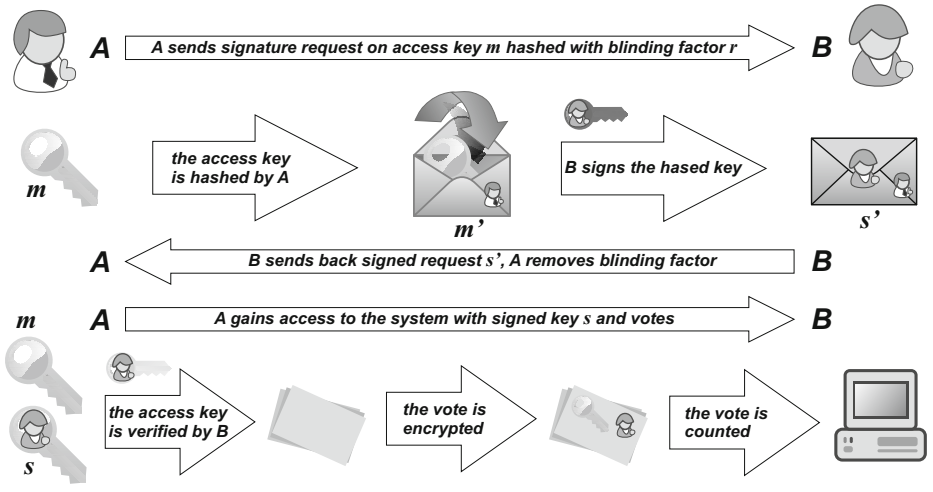


Fig. 2. The user's (A) secure access key to the electronic voting system is blindly signed by the electoral committee (B)

5 Conclusion

Moving the voting procedures to electronic devices shall decrease the costs of elections, accelerate voting and counting votes and facilitate participation in the election. The increased turnout shall lead to more reliable election results.

In the article a few models of electronic voting were presented, including systems which enable voting through the Internet. A short analysis of each of the systems shows that it is possible to assure the secrecy of voting, anonymity of the user, and at the same time verification of the user and provision of as representative results as possible. Anonymity connected with authorization is possible, but it is very easy to lose the separation between the vote and the voting person. Concentrating more on the authorization it is very easy to modify system in such a way as to have the voter's data stored together with the vote's content. By creating a system directed more to anonymity, the voter has the advantage of getting a tool necessary for the vote or vote rights trade (Table 1).

Table 1. A comparison of different models of voting systems

	remote access only	anonymity break	results manipulation	votes selling	verification of votes
traditional voting	no	hard	hard	hard	no
direct authorization	yes	easy	easy	medium	no
anonymous identifiers	no	hard	hard	easy	yes
blind signatures	yes	hard	hard	easy	yes

The fully anonymous voting system with remote access only over the Internet can be created using the blind signature protocol. The voters are authorized in the system so the access is limited only to entitled users. The users have a possibility of vote verification and can control system administrators in case of results manipulation. Such a model is a good solution in the polls system for teaching quality evaluation where the student anonymity is very important condition. The disadvantage of blind signature protocol is possibility of creating vote proofs and as a result vote market can be made. It is unacceptable in general election where the winners achieve power.

It is equally important to limit the rights for single voting and to maintain the secrecy of voting. All models presented in the paper meets these requirements. The problems from Table 1 concerning not intended use of the system. Opening the source code [14] and protecting code changes using TPM devices [4] can eliminate anonymity and result manipulation issues. Using of the blind signature protocol assures voter's anonymity without such efforts.

References

1. Ansari, N., Sakarindr, P., Haghani, E., Zhang, C., Jain, A.K., Shi, Y.Q.: Evaluating Electronic Voting Systems Equipped with Voter-Verified Paper Records. IEEE Security & Privacy 6(3), 30–39 (2008)
2. Chaum, D.: Secret-Ballot Receipts: True Voter-Verifiable Elections. IEEE Security & Privacy 2(1), 38–47 (2004)
3. Epstein, J.: Electronic Voting. Computer 40(8), 92–95 (2007)
4. Paul, N., Tanenbaum, A.S.: Trustworthy Voting: From Machine to System. Computer 42(5), 23–29 (2009)
5. Dustor, A., Bak, M.: Biometryczny system weryfikacji tozsamosci. In: Nowe technologie sieci komputerowych, Tom, vol. 2, pp. 393–400, WKiL, Gliwice (2006)
6. Jain, A.K., Ross, A., Pankanti, S.: Biometrics: A Tool for Information Security. IEEE Tran. on Information Forensics and Security 1(3), 311–329 (2006)
7. Estonian National Electoral Committee: Internet Voting in Estonia, http://www.vvk.ee/index.php?id=11178
8. Kohnfelder, L.M.: Towards a Practical Public-Key Cryptosystem. MIT, Cambridge (1978)
9. Toussaint, M.J.: A New Method for Analyzing the Security of Cryptographic Protocols. IEEE Jou. on Selected Areas in Comm. 11(5), 702–714 (1993)
10. Kucharczyk, M.: Internetowy system oceny zajec dydaktycznych. In: Wspolczesne aspekty sieci komputerowych, Tom, vol. 1, pp. 423–432. WKiL, Warszawa (2008)
11. Chaum, D.: Blind Signatures for Untraceable Payments. In: Crypto 1982, pp. 199–203. Plenum Press, New York (1983)
12. Ibrahim, S., Kamat, M., Salleh, M., Aziz, S.R.A.: Secure E-voting with Blind Signature. In: NCTT 2003 Proceedings, 4th National Conference on Telecommunication Technology, Malaysia (2003)
13. Cetinkaya, O., Doganaksoy, A.: Pseudo-Voter Identity (PVID) Scheme for e-Voting Protocols. In: The Second International Conference on Availability, Reliability and Security (ARES 2007), Austria, pp. 1190–1196 (2007)
14. Open Voting Consortium: Open Source Voting: Accurate, Accountable, http://www.openvotingconsortium.org/

Author Index