JOURNAL OF TELECOMMUNICATIONS AND INFORMATION TECHNOLOGY

1/2017

Compensation of Fading Channels using Partial Combining Equalizer in MC-CDMA Systems
M. Zidane, S. Safi, and M. Sabri — Paper — 5

Unsupervised Phoneme Segmentation Based on Main Energy Change for Arabic Speech
N. Lachachi — Paper — 12

Optimization of Spectrum Sensing Parameters in Cognitive Radio Using Adaptive Genetic Algorithm
S. Chatterjee et al. — Paper — 21

Design Exploration of AES Accelerators on FPGAs and GPUs
V. Conti and S. Vitabile — Paper — 28

A Queue Monitoring System in OpenFlow Software Defined Networks
S. Rowshanrad, S. Namvarasl, and M. Keshtgari — Paper — 39

Sensor Hop-based Energy Efficient Networking Approach for Routing in Underwater Acoustic Communication
S. Kohli and P. P. Bhattacharya — Paper — 44

L-SCANN: Logarithmic Subcentroid and Nearest Neighbor
P. Mishra, C. Gandhi, and B. Singh — Paper — 50

Energy Efficient Scheduling Methods for Computational Grids and Clouds
A. Jakóbik et al. — Paper — 56

(*Contents Continued on Back Cover*)

Editorial Board

Editor-in Chief: ..... ***Paweł Szczepański***

Associate Editors: ..... ***Krzysztof Borzycki***
Marek Jaworski

Managing Editor: ..... ***Robert Magdziak***

Technical Editor: ..... ***Ewa Kapuściarek***

Editorial Advisory Board

Chairman: ..... ***Andrzej Jajszczyk***
Marek Amanowicz
Hovik Baghdasaryan
Wojciech Burakowski
Andrzej Dąbrowski
Andrzej Hildebrandt
Witold Hołubowicz
Andrzej Jakubowski
Marian Kowalewski
Andrzej Kowalski
Józef Lubacz
Tadeusz Łuba
Krzysztof Malinowski
Marian Marciniak
Józef Modelski
Ewa Orłowska
Tomasz Osuch
Andrzej Pach
Zdzisław Papir
Michał Pióro
Janusz Stokłosa
Andrzej P. Wierzbicki
Tadeusz Więckowski
Adam Wolisz
Józef Woźniak
Tadeusz A. Wysocki
Jan Zabrodzki
Andrzej Zieliński

ISSN 1509-4553 on-line: ISSN 1899-8852

© Copyright by National Institute of Telecommunications
Warsaw 2017

Circulation: 300 copies

Sowa – Druk na życzenie, www.sowadruk.pl, tel. 22 431-81-40

JOURNAL OF TELECOMMUNICATIONS AND INFORMATION TECHNOLOGY

Preface

Intelligent and data intensive computing remain the key paradigms in today's large-, medium- and ultra-scale communication networks. This volume of *Journal of Telecommunications and Information Technology* encompass twelve research papers reporting the recent developments and implementations in the modern networking.

The papers are divided into three groups. The first group contains three papers presenting the interesting examples of the usage of new methods (metaheuristics, monitoring systems) in the optimization of the network performances for specific applications.

Zidane *et al*. in *Compensation of Fading Channels Using Partial Combining Equalizer in MC-CDMA Systems* developed a new equalizer for Multi-Carrier Code Division Multiple Access (MC-CDMA) systems. They used the Broadband Radio Access Networks (BRAN A) channel model for channel identification and propagation in an indoor environment. The authors present the detailed theoretical analysis and numerical simulations in the noisy environment and for different Signal to Noise Ratio (SNR) for illustration of the high efficiency of the developed partial combining equalizer and its significant on the performance of MC-CDMA systems.

In the second paper, *Unsupervised phoneme segmentation based main energy change for Arabic speech*, Lachachi present a new method for segmenting the Arabic languages speech at the phoneme level. In this case, the novel hardware-based methods for the implementation of the Fast Fourier Transformation were investigated. The real aim of the research was the identification of the big energy changes in frequency over time, which can be described as phoneme boundaries. The author applied a frequency range analysis and search for detecting such energy modulation. The segmentation of the acoustic signals provided with such frequency range analysis show that over 80% of the boundaries were successfully identified.

Incumbent user spectrum band can be used in cognitive radio for communication purposes without can perform a communication using the without the interference caused by the users. Such incumbent user spectrum band uses the spectrum sensing technology. Chatterjee *et al.* in *Optimization of Spectrum Sensing Parameters in Cognitive Radio Using Adaptive Genetic Algorithm*, developed evolutionary-based AI metaheuristic for optimization of the Quality of Services spectrum sensing parameters, such as bit error rate, throughput, power consumption, interference, spectral efficiency, etc.

The second group of the papers addresses the problems of energy consumption and security aspects in new generation networks.

In the first paper in this part, Conti and Vitabile in *Design Exploration of AES Accelerators on FPGAs and GPUs* present the comparative analysis of different implementations of Rijndael AES cryptographic algorithm. Two implementations were presented: a novel FPGA using Celoxica RC1000 board, equipped with Agility's Handel-C compiler and a parallel OpenCL library based program running on GPU. Such implementation is a challenging task, and the compared FPGA and GPU used as accelerators for the AES cipher are very popular devices for parallel computation.

The analysis and optimization of the network traffic are among the most important methods of conservation of the energy in large and small areas networks. Rowshanrad *et al*. in *A Queue Monitoring System in OpenFlow Software Defined Networks*, used the Software Defined Networks for monitoring the performance of telecommunication networks. The proposed technology allows to achieve the 99% accuracy in monitoring of delay and available bandwidth of a queue on a link or path of the network, and in consequence, reduce the energy consumption in such networks.

A popular method of saving the energy in sensor networks is the improvement of the efficiency of the routing protocols. It can be very difficult in the specific real-world scenarios. Kohli and Bhattacharya in *Sensor Hop-based Energy Efficient Networking Approach for Routing in Underwater Acoustic Communication* used the Sensor Hop-based Energy Efficient Network Approach (SHEENA) for the construction of the energy-efficient, scalable and fault-tolerant Underwater Sensor Network (USN). They verified the QoS features of the network in shallow and deep-water environments and compared it with the USN with traditional multi-hop LEACH protocol. They achieved the 25–30% reduction rate of the energy consumption.

The optimal routing in mobile ad hoc networks remains a challenging problem. The greedy protocol uses the distance or direction metrics for detecting the relay node as a data transfer point from source to destination. Mishra *et al*. in *An Improved Greedy Forwarding Scheme in MANETs* propose the new position-based routing protocol in MANETs. They combine the distance-based and deviation-based node selection schemes into a joint weighted forwarding method. They show in the experimental section that their protocol is significantly better in the minimization of the average hop-count, end-to-end delay and routing overhead than distance and direction based routings considered separately.

Jakóbik *et al*. in *Energy Efficient Scheduling Methods for Computational Grids and Clouds* survey the recent models and technologies for measurement of the energy consumption in computational clouds. They provide an overview of resource allocation methods, tasks scheduling and load balancing methods. The ECT matrix is the basic scheduling model. The experiments defined for the Amazon Cloud instances presented at the very end of the paper show the impact of the optimal task scheduling on the energy consumed in the cloud environment.

Significant reduction of the consuming energy in large-scale HPC systems may be related to the implementation of the light-weighted authorization and users' authentication protocols, and cryptographic methods. Rayin in *Introduction to Big Data Management Based on Agent Oriented Cyber Security*, developed a new model of Big Data management based on the agent oriented cyber security model in public spaces. The use of the multi-agent system for such management process is very promising research area, especially in the context of the cyber security.

The game theory can be another methodology supporting the task and data management in large-scale networks. Jakóbik and Wilczyński in *Using Polymatrix Extensive Stackelberg Games in Security-Aware Resource Allocation and Task Scheduling in Computational Clouds* defined the Stackelberg game model for supporting the administrative decisions on tasks processing in Computational Clouds. Such game-based allows to optimize the computing capacities of the Virtual Machines for the processing of the generated schedules. The security aspects in that approach were addressed in mapping the security requirements specified for the tasks to virtual machine. The declared trust levels of the machines were interpreted as the mapping criterion. The number of virtual resources and their characteristics may be changed during the game. The optimal strategies for the game, and thus the proper security levels, were calculated automatically.

The last two papers show interesting examples of real-world applications of intelligent networking algorithms and models.

Earth observations are important area in modeling and simulations of such realistic scenarios. Sensing and radiosonde monitoring of the Earth atmosphere generate the large volume of data for the analysis. However, some data can be damaged during this monitoring process. Szuster in *Data fixing algorithm in radiosonde monitoring process* developed a new algorithm, which allows to repair such damaged data. The achieved efficiency improvement rate is significant and reaches up to 70%.

In the last paper of this issue, *My City Dashboard: Real-time Data Processing Platform for Smart Cities*, Usurelu and Pop address the problem of real-time processing of sensing data collected by sensors within a smart city. They visualized that data by using the dashboard model for citizen empowerment. This method is extremely important for the effective usage of the smart city application by average educated citizens without a strong background in computer science and ICT technologies.

I truly believe that this *JTIT* volume will serve as a reference for students, researchers, and industry practitioners currently working or interested in joining interdisciplinary works in the areas of intelligent modern networking using emergent large-scale distributed computing paradigms. It will also allow newcomers to grasp key concepts and potential solutions in advanced topics of theory, models, technologies, system architectures and implementation of applications in various types of sensor, wide and small area networks.

I am grateful to all the authors for their interesting papers, their time, efforts and their research results, which makes this volume a valuable source of the latest research advances and technology development on the next generation intelligent networks. I also would like to express my sincere thanks to the reviewers, who have helped me to ensure the quality of this volume. I gratefully acknowledge their time and valuable remarks and comments. My special thanks go to the journal editors for their patience, valuable editorial assistance and excellent cooperation in the preparation of this volume.

Joanna Kołodziej
Guest Editor

Paper

Compensation of Fading Channels Using Partial Combining Equalizer in MC-CDMA Systems

Mohammed Zidane[1], Said Safi[2], and Mohamed Sabri[1]

[1] *Department of Physics, Faculty of Sciences and Technology, Sultan Moulay Slimane University, Morocco*
[2] *Department of Mathematics and Informatics, Polydisciplinary Faculty, Sultan Moulay Slimane University, Morocco*

Abstract—In this paper the performance of a partial combining equalizer for Multi-Carrier Code Division Multiple Access (MC-CDMA) systems is analytically and numerically evaluated. In the part of channel identification, authors propose a blind algorithm based on Higher Order Cumulants (HOC) for identifying the parameters representing the indoor scenario of Broadband Radio Access Networks (BRAN A) channel model normalized for MC-CDMA systems. Theoretical analysis and numerical simulation results, in noisy environment and for different Signal to Noise Ratio (SNR), are presented to illustrate the performance of the proposed algorithm in the one hand, and the other hand the impact of partial combining equalizer on the performance of MC-CDMA systems.

Keywords—bit error rate, blind channel identification, higher order cumulants, MC-CDMA systems, partial combining equalizer.

1. Introduction

Fourth generation (4G) technology allows user to efficiently share common resources. However, the exponential growth of multimedia users request fast data rates and reliable transmission. The 4G wireless systems utilizing available limited bandwidth in a spectrally efficient manner. To attain these aims, there are two principle contending technologies, i.e. Orthogonal Frequency Division Multiplexing (OFDM) and Code Division Multiple Access (CDMA). Therefore, OFDM-CDMA and MC-CDMA gain a lot of attention for wireless mobile communication [1]–[2]. The principles of MC-CDMA [3] is that a single data symbol is transmitted on multiple narrow band subcarriers. Indeed, in MC-CDMA systems, spreading codes are applied in the frequency domain and transmitted over independent subcarriers. However, multicarrier systems are very sensitive to synchronization errors such as carrier frequency offset and phase noise. Synchronization errors cause loss of orthogonality among subcarriers and considerably degrade the performance especially when large number of subcarriers presents [4]–[5]. The MC-CDMA modulator is an effective technique for combating multipath fading over highly dispersive wireless channels. The problem encountered in digital communication is the synchronization between the transmitter and the receiver; there are many obstacles in the channels. Reflections from these obstacles degrade the transmitted signal before it reaches the receiver. Hence, channel equalization is required to reduce the Bit Error Rate (BER) of the receiver as small as possible. In fact, the goal of the equalization techniques is to reduce the effect of the fading and the interference while not enhancing the effect of the noise on the decision of what data symbol was transmitted. Therefore, the problem of channel identification appears.

In this paper, authors propose an algorithm for blind channel identification, using higher order cumulants. There are several motivations behind this interest [6]. Firstly, higher order cumulants are blind to all kinds of Gaussian noise that is, the additive noise Gaussian will vanish in the higher order cumulants domain. Secondly, cumulants are useful in identifying non-minimum phase channels when the input is non-Gaussian and is contaminated by Gaussian noise.

The problem of the blind identification of the Broadband Radio Access Network (BRAN) channels, normalized by the European Telecommunications Standards Institute (ETSI) [7]–[8], and downlink MC-CDMA equalization using higher order cumulants was proposed by several authors [9]–[12].

In this contribution, authors present a partial combining equalizer for downlink MC-CDMA systems equalization, for that is in the one hand, the problem of the blind identification of (BRAN A) channel using the proposed algorithm is considered, and in the other hand the presented equalizer after the channel identification to correct the channel's distortion is used. The numerical simulation results, in noisy environment, are presented to illustrate the accuracy of the proposed algorithm, and the performance of equalization.

2. Channel Identification using Higher Order Cumulants

2.1. Channel Model

The term channel refers to the transmitting space (medium) between the transmitter and the receiver antennas as shown in Fig. 1.

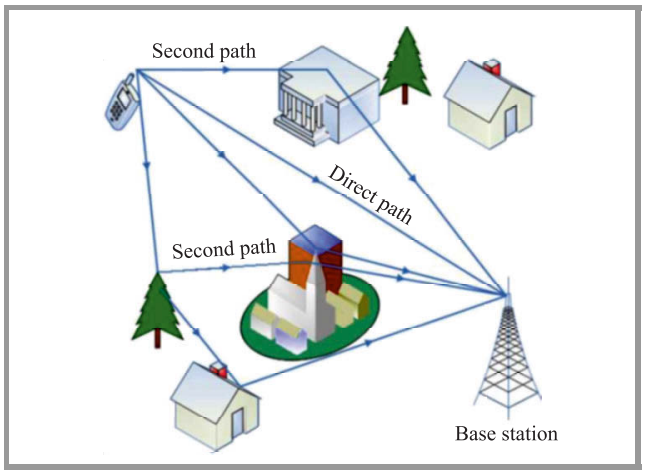


Fig. 1. Channel model.

The characteristics of wireless signal changes as it travels from the transmitter antenna to the receiver antenna. These characteristics depend upon the distance between the two antennas, the path(s) taken by the signal, and the environment (buildings and other objects) around the path. The profile of received signal can be obtained from that of the transmitted signal if we have a model of the medium between the two. This model of the medium is called channel model (Fig. 1).

In this paper it is assumed that the channel is time invariant and its impulse response is characterized by P paths of magnitudes β_p and phases θ_p. The impulse response is given by:

$$h(\tau) = \sum_{p=0}^{P-1} \beta_p e^{\theta_p} \delta(\tau - \tau_p). \tag{1}$$

In general, the received signal can be obtained by convolving the transmitted signal with the impulse response of the channel:

$$r(t) = h(t) * x(t) + n(t), \tag{2}$$

$$\begin{aligned} r(t) &= \int_{-\infty}^{+\infty} \sum_{p=0}^{P-1} \beta_p e^{\theta_p} \delta(\tau - \tau_p) x(t-\tau) d\tau + n(t) = \\ &= \sum_{p=0}^{P-1} \beta_p e^{\theta_p} x(t - \tau_p) + n(t), \end{aligned} \tag{3}$$

where $x(t)$ is the input sequence, $h(t)$ is the impulse response coefficients, τ_p is the time delay of the p-th path, and $n(t)$ is the additive noise sequence.

2.2. Moments and Cumulants

In this Subsection, a mathematical definitions of the estimated moments and cumulants needed to identifying the impulse response parameters of finite impulse response (FIR) systems are presented.

Let us consider a random non-Gaussian variable $y(k)$. The sample estimates are given by:

$$\widehat{M}_{m,y}(\tau_1, \ldots, \tau_{m-1}) = \frac{1}{N} \sum_{i=1}^{N} y(i) y(i+\tau_1) \ldots y(i+\tau_{m-1}). \tag{4}$$

As the cumulants are expressed in function of moments, the estimates cumulants of order two, three and four are defined respectively by:

$$\widehat{C}_{2,y}(\tau_1) = \widehat{M}_{2,y}(\tau_1) = \frac{1}{N} \sum_{i=1}^{N} y(i) y(i+\tau_1), \tag{5}$$

$$\begin{aligned} \widehat{C}_{3,y}(\tau_1, \tau_2) &= \widehat{M}_{3,y}(\tau_1, \tau_2) = \\ &= \frac{1}{N} \sum_{i=1}^{N} y(i) y(i+\tau_1) y(i+\tau_2), \end{aligned} \tag{6}$$

$$\begin{aligned} \widehat{C}_{4,y}(\tau_1, \tau_2, \tau_3) &= \widehat{M}_{4,y}(\tau_1, \tau_2, \tau_3) - \\ &- \widehat{M}_{2,y}(\tau_1) \widehat{M}_{2,y}(\tau_3 - \tau_2) - \\ &- \widehat{M}_{2,y}(\tau_2) \widehat{M}_{2,y}(\tau_3 - \tau_1) - \\ &- \widehat{M}_{2,y}(\tau_3) \widehat{M}_{2,y}(\tau_2 - \tau_1). \end{aligned} \tag{7}$$

2.3. Basic Relationships

In this section, the main general relationships between cumulants of the output signal and impulse response coefficients are described. The starting point for all algorithms based on higher order cumulants is Brillinger and Rosenblatt relation shows that the m-th order cumulants of $y(k)$ can be expressed as a function of impulse response coefficients $h(i)$ as follows [13]:

$$\begin{aligned} &C_{m,y}(\tau_1, \tau_2, \ldots, \tau_{m-1}) = \\ &= \xi_{m,x} \sum_{i=0}^{q} h(i) h(i+\tau_1) \ldots h(i+\tau_{m-1}), \end{aligned} \tag{8}$$

where $\xi_{m,x}$ represents the m-th order cumulants of the excitation signal $x(k)$ at origin.

Peyre *et al.* presents the relationship between different m-th and n-th order cumulants of the output signal, $y(k)$, and the coefficients $h(i)$, where $n > m$ and $(n,m) \in N^* - \{1\}$, are linked by the following relationship [14]:

$$\begin{aligned} &\sum_{j=0}^{q} h(j) C_{n,y}(j+\tau_1, j+\tau_2, \ldots, j+\tau_{m-1}, \tau_m, \ldots, \tau_{n-1}) = \\ &= \mu \sum_{i=0}^{q} h(i) \Big[\prod_{k=m}^{n-1} h(i+\tau_k) \Big] C_{m,y}(i+\tau_1, i+\tau_2, \ldots, i+\tau_{m-1}), \end{aligned} \tag{9}$$

where $\mu = \frac{\xi_{n,x}}{\xi_{m,x}}$.

In order to simplify the construction of the proposed algorithm we assume that:

- the input sequence, $x(k)$, is independent and identically distributed (i.i.d.) zero mean, and non-Gaussian;
- the system is causal and bounded, i.e. $h(i) = 0$ for $i < 0$ and $i > q$, where $h(0) = 1$,
- the system order q is known,

- the measurement noise sequence $n(k)$ is assumed zero mean, i.i.d., Gaussian and independent of $x(k)$ with unknown variance.

The problem statement is to identify the parameters of the system $h(i)_{(i=1,\ldots,q)}$ using the cumulants of the measured output signal $y(k)$.

3. Proposed Algorithm

By substituting $n=4$ and $m=2$ into Eq. (9) the following equation can be obtained:

$$\sum_{j=0}^{q} h(j)C_{4,y}(j+\tau_1,\tau_2,\tau_3) = \\ = \mu \sum_{i=0}^{q} h(i)\Big[\prod_{k=2}^{3} h(i+\tau_k)\Big]C_{2,y}(i+\tau_1), \qquad (10)$$

$$\sum_{j=0}^{q} h(j)C_{4,y}(j+\tau_1,\tau_2,\tau_3) = \\ = \mu \sum_{i=0}^{q} h(i)h(i+\tau_2)h(i+\tau_3)C_{2,y}(i+\tau_1), \qquad (11)$$

where $\mu = \frac{\xi_{4,x}}{\xi_{2,x}}$.

The autocorrelation function of the (FIR) systems vanishes for all values of $|\tau| > q$, equivalently:

$$C_{2,y}(\tau) = \begin{cases} \neq 0, & |\tau| \leq q; \\ 0 & \text{otherwise.} \end{cases}$$

If we suppose that $\tau_1 = q$ the Eq. (11) becomes:

$$\sum_{j=0}^{q} h(j)C_{4,y}(j+q,\tau_2,\tau_3) = \mu h(0)h(\tau_2)h(\tau_3)C_{2,y}(q), \quad (12)$$

and for $\tau_3 = 0$ the Eq. (12) becomes:

$$\sum_{j=0}^{q} h(j)C_{4,y}(j+q,\tau_2,0) = \mu h^2(0)h(\tau_2)C_{2,y}(q). \qquad (13)$$

The considered system is causal and bounded, thus, the interval of the τ_2 is $\tau_2 = 0,\ldots,q$.
Else if we suppose that $\tau_2 = 0$, and using the cumulants properties $C_{m,y}(\tau_1,\tau_2,\ldots,\tau_{m-1}) = 0$, if one of the variables $\tau_k > q$, where $k = 1,\ldots,m-1$, the Eq. (13) becomes:

$$C_{4,y}(q,0,0) = \mu h^3(0)C_{2,y}(q). \qquad (14)$$

Thus, we are based on Eq. (14) for eliminating $C_{2y}(q)$ in Eq. (13), we obtain the equation constituted of only the fourth order cumulants, this equation describe the proposed algorithm:

$$\sum_{j=0}^{q} h(j)C_{4,y}(j+q,\tau_2,0) = h(\tau_2)C_{4,y}(q,0,0). \qquad (15)$$

The system of Eq. (15) can be written in matrix form as:

$$\begin{bmatrix} C_{4,y}(q+1,0,0) & \ldots & C_{4,y}(2q,0,0) \\ C_{4,y}(q+1,1)-\alpha & \ldots & C_{4,y}(2q,1,0) \\ . & . & . \\ . & . & . \\ . & . & . \\ C_{4,y}(q+1,q,0) & \ldots & C_{4,y}(2q,q,0)-\alpha \end{bmatrix} \times \\ \times \begin{bmatrix} h(1) \\ . \\ . \\ . \\ h(i) \\ . \\ . \\ . \\ h(q) \end{bmatrix} = \begin{bmatrix} 0 \\ -C_{4,y}(q,1,0) \\ . \\ . \\ . \\ . \\ . \\ -C_{4,y}(q,q,0) \end{bmatrix}, \qquad (16)$$

where $\alpha = C_{4,y}(q,0,0)$. Or, in more compact form, the Eq. (16) can be written as follows:

$$Mh_e = d, \qquad (17)$$

where M is the matrix of size $(q+1)\times(q)$ elements, h_e is a column vector constituted by the unknown impulse response parameters $h(i)_{i=1,\ldots,q}$ and d is a column vector of size $(q+1)$ as indicated in the Eq. (16). The least squares solution of the system of Eq. (17), permits blindly identification of the parameters $h(i)$ and without any information of the input selective channel. Thus, the solution will be written under the following form:

$$\widehat{h}_e = (M^T M)^{-1} M^T d. \qquad (18)$$

4. MC-CDMA Model

The multicarrier code division multiple access (MC-CDMA) system is based on the combination of code division multiple access (CDMA) and orthogonal frequency division multiplexing (OFDM), which is potentially robust to channel frequency selectivity. However, the complex symbol a_i of each user i is, firstly, multiplied by each chip $c_{i,k}$ of spreading code, and then applied to the modulator

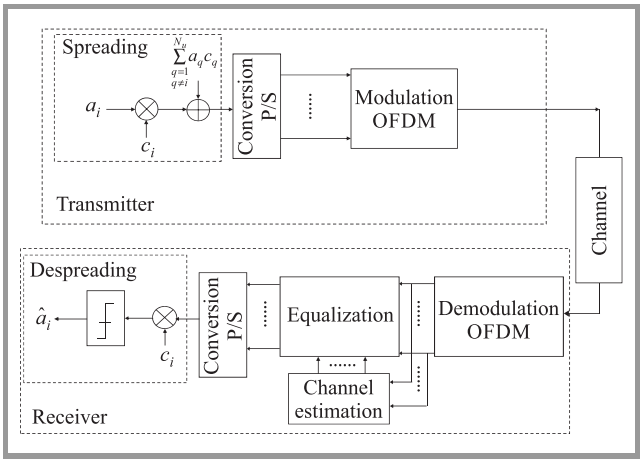


Fig. 2. MC-CDMA modulator principle.

of multicarriers. Each subcarrier transmits an element of information multiply by a code chip of that subcarrier. Figure 2 explains the principle of the MC-CDMA systems.

4.1. MC-CDMA Transmitter

The MC-CDMA signal is given by:

$$x(t) = \frac{a_i}{\sqrt{N_p}} \sum_{k=0}^{N_p-1} c_{i,k} e^{2j\pi f_k t}, \tag{19}$$

where $f_k = f_0 + \frac{k}{T_c}$, N_p is the number of subcarriers, and we consider $L_c = N_p$.

4.2. MC-CDMA Receiver

The downlink received MC-CDMA signal at the input receiver is given by the following equation:

$$\begin{aligned} r(t) = \frac{1}{\sqrt{N_p}} \sum_{p=0}^{P-1} \sum_{k=0}^{N_p-1} \sum_{i=0}^{N_u-1} \times \\ \times \Re\left\{ \beta_p e^{j\theta_p} a_i c_{i,k} e^{2j\pi(f_0+k/T_c)(t-\tau_p)} \right\} + n(t) \end{aligned} \tag{20}$$

The Eq. (20) can be written as follows:

$$r = HCa + n, \tag{21}$$

where r denotes a vector composed of the values received on each subcarrier:

$$r = [r_0, \ldots, r_{N_p-1}]^T. \tag{22}$$

The matrix H represents the matrix of complex coefficients of channel with size $N_p \times N_p$:

$$H = \begin{bmatrix} h_0 & 0 & \ldots & 0 \\ 0 & h_1 & \ldots & 0 \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \ldots & h_{N_p-1} \end{bmatrix}. \tag{23}$$

The matrix C represent the spreading codes:

$$C = \begin{bmatrix} c_{0,0} & c_{0,1} & \ldots & c_{0,N_u-1} \\ c_{1,0} & c_{1,1} & \ldots & c_{1,N_u-1} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ c_{L_c-1,0} & c_{L_c-1,1} & \ldots & c_{L_c-1,N_u-1} \end{bmatrix}, \tag{24}$$

where

$$\begin{aligned} c_i &= [c_{0,i}, c_{1,i}, \ldots, c_{L_c-1,i}]^T, \\ a &= [a_0, \ldots, a_{N_u-1}]^T, \\ n &= [n_0, \ldots, n_{N_p-1}]^T. \end{aligned} \tag{25}$$

At the reception, we demodulate the signal according the N_p subcarriers, and then we multiply the received sequence by the code of the user. Figure 3 explains the single user detection principle. Using the above matrix notation, it is possible to express G – the diagonal matrix composed of the coefficients g_k equalization:

$$G = \begin{bmatrix} g_0 & 0 & \ldots & 0 \\ 0 & g_1 & \ldots & 0 \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \ldots & g_{N_p-1} \end{bmatrix}. \tag{26}$$



Fig. 3. Principe of the single user-detection.

After the equalization and the spreading operation, the estimation $\widehat{a_i}$ of the emitted user symbol a_i, of the i-th user can be written by the following equations:

$$\begin{aligned} \widehat{a_i} &= c_i^T G r \\ &= c_i^T G(HCa + n) \\ &= c_i^T GHCa + c_i^T Gn. \end{aligned} \tag{27}$$

$$\begin{aligned} \widehat{a_i} &= \sum_{q=0}^{N_u-1} \sum_{k=0}^{N_p-1} c_{i,k}(g_k h_k c_{q,k} a_q + g_k n_k) = \\ &= \underbrace{\sum_{k=0}^{N_p-1} c_{i,k}^2 g_k h_k a_i}_{\mathbf{I}\ \ (i=q)} + \underbrace{\sum_{q=0}^{N_u-1} \sum_{k=0}^{N_p-1} c_{i,k} c_{q,k} g_k h_k a_q}_{\mathbf{II}\ \ (i\neq q)} + \\ &+ \underbrace{\sum_{k=0}^{N_p-1} c_{i,k} g_k n_k}_{\mathbf{III}}, \end{aligned} \tag{28}$$

where the term I, II and III of Eq. (28) are, respectively, the signal of the considered user, a signals of the others users (multiple access interferences) and the noise pondered by the equalization coeffcient and by spreading code of the chip.
If we suppose that the spreading code are orthogonal, i.e.

$$\sum_{k=0}^{N_p-1} c_{i,k} c_{q,k} = 0 \qquad \forall i \neq q, \tag{29}$$

Eq. (28) will become:

$$\widehat{a_i} = \underbrace{\sum_{k=0}^{N_p-1} c_{i,k}^2 g_k h_k a_i}_{\mathbf{I}\ \ (i=q)} + \underbrace{\sum_{k=0}^{N_p-1} c_{i,k} g_k n_k}_{\mathbf{III}}. \tag{30}$$

5. Partial Combining Equalizer

In [15] a partial combining (PC) technique was introduced, with coefficient g_k function of a PC parameter, $-1 \leq \beta \leq 1$, as given by:

$$g_k = \frac{h_k^*}{|h_k|^{1+\beta}}. \tag{31}$$

The estimated received symbol, $\widehat{a}_i$ of symbol a_i of the user i is described by:

$$\widehat{a}_i = \sum_{k=0}^{N_p-1} c_{i,k}^2 \frac{|h_k|^2}{|h_k|^{1+\beta}} a_i + \sum_{k=0}^{N_p-1} c_{i,k} \frac{h_k^*}{|h_k|^{1+\beta}} n_k \,. \tag{32}$$

5.1. Particular Case: $\beta = 1$ – Zero Forcing Equalizer

The gain factor of the zero forcing (ZF) equalizer, is given by the equation:

$$g_k = \frac{1}{h_k} \,. \tag{33}$$

The estimated received symbol, $\widehat{a}_i$ of symbol a_i of the user i is described by:

$$\widehat{a}_i = \sum_{k=0}^{N_p-1} c_{i,k}^2 a_i + \sum_{k=0}^{N_p-1} c_{i,k} \frac{1}{h_k} n_k \,. \tag{34}$$

6. Simulation Results

In this section the numerical results for blind identification and equalization in MC-CDMA systems are presented. For that we consider the BRAN A model representing the propagation in an indoor case. The Eq. (35) describe the impulse response of BRAN A channel:

$$h(\tau) = \sum_{i=0}^{N_T} A_i \delta(\tau - \tau_i) \,. \tag{35}$$

In the Table 1 the measured values corresponding the BRAN A radio channel impulse response are summarized.

Table 1
Delay and magnitudes of 18 targets of BRAN A channel

Delay τ_i [ns]	Mag. A_i [dB]	Delay τ_i [ns]	Mag. A_i [dB]
0	0	90	−7.8
10	−0.9	110	−4.7
20	−1.7	140	−7.3
30	−2.6	170	−9.9
40	−3.5	200	−12.5
50	−4.3	240	−13.7
60	−5.2	290	−18
70	−6.1	340	−22.4
80	−6.9	390	−26.7

Although, the BRAN A channel is constituted by $N_T = 18$ parameters and seeing that the latest parameters are very small, for that we have taking the following procedure:

- the BRAN A channel impulse response is decomposed into three subchannels:

$$h(i) = \sum_{j=1}^{3} h_j(i) \,; \tag{36}$$

- the parameters of each subchannel are estimated independently, using the proposed algorithm;
- all subchannel parameters are added, to construct the full BRAN A channel impulse response.

The simulation is performed with Matlab software and for different SNR.

6.1. Identification of BRAN A Channel using the Proposed Algorithm

In this subsection the BRAN A channel model is considered. Figure 4 show the impulse response estimation for this channel using the proposed algorithm for different SNR and an data length $N = 5400$.

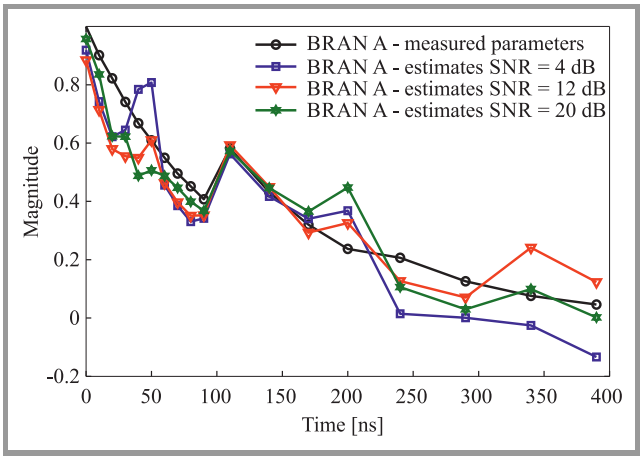


Fig. 4. Estimated of the BRAN A channel impulse response, for different SNR and a data length $N = 5400$.

This figure shows clearly the influence of Gaussian noise on parameter estimation of the BRAN A impulse response. This influence is clear principally for the last five values, where the estimated parameters do not follow those measured. But, before the last fifth values, the 13th first estimated values are closed to those measured are observed. This due that the additive Gaussian noise vanished in the higher order cumulants domain.

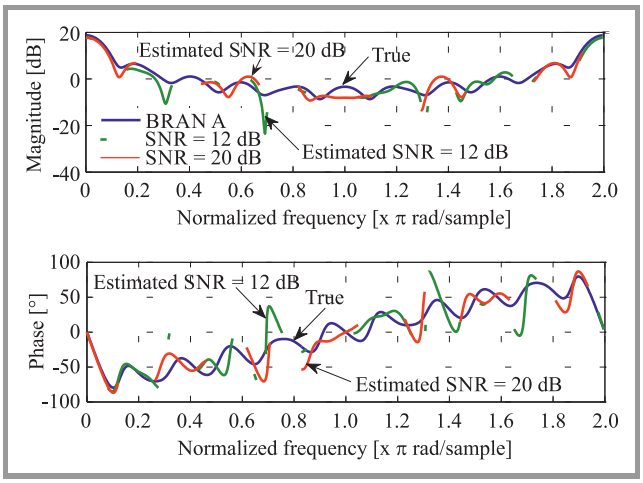


Fig. 5. Estimated of the BRAN A channel impulse response using all target, for different SNR and a data length $N = 5400$.

6.2. Magnitude and Phase Estimation of BRAN A Channel using the Proposed Algorithm

In Fig. 5 we represent the estimation magnitude and phase of the impulse response of the BRAN A channel using the proposed algorithm, for an SNR varying between 12 dB and 20 dB, the data length is 5400.
The estimated magnitude and phase have the same form. In the fact, one can see a low influence of the noise on the estimation of the magnitude and phase principally when the noise is SNR $>$ 12 dB, and we have not more difference between the estimated and the true ones.

6.3. Compensation of Fading Channels using Partial Combining Equalizer

In order to evaluate the performance of the MC-CDMA system, using the presented equalizer. These performances are evaluated by calculation of the BER, for different values of β, using the measured and estimated, using proposed algorithm of the BRAN A channel impulse response.

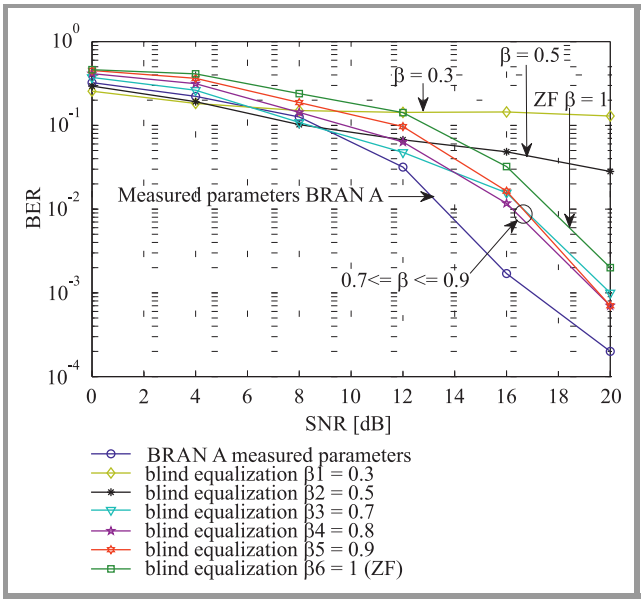


Fig. 6. BER of the estimated and measured BRAN A channel, for different SNR, using the presented equalizers.

Figure 6 shows the simulation results of BER estimation, for different SNR, using presented equalizer of the BRAN A channel impulse response when $0.7 \leq \beta \leq 0.9$ the partial combining equalizer is more precise and gives good results than those obtained by ZF equalizer. Also, the ZF equalizer is best compared to the results obtained using partial combining equalizer for $\beta \leq 0.6$.

7. Conclusion

In this contribution, a partial combining equalizer has been analytically and numerically investigated in MC-CDMA systems. In the part of the channel identification, the BRAN A model was used representing the propagation in an indoor case normalized for MC-CDMA systems. To estimate the coefficients of this equalizer, the authors have proposed an algorithm based on fourth order cumulants. The proposed algorithm shows their efficiency in the impulse response channel identification with very good precision. In the part of the equalization for the MC-CDMA systems using the presented equalizer, it has been demonstrated that the partial combining equalizer is very adequate for correcting the channel distortion for $0.7 \leq \beta \leq 0.9$.

References

[1] Z. Wang and G. B. Giannakis, "Wireless multicarrier communications: where Fourier meets Shannon", *IEEE Sig. Process. Mag.*, vol. 17, no. 3, pp. 29–48, 2000.

[2] S. Hara and R. Prasad, "Overview of multicarrier CDMA", *IEEE Commun. Mag.*, vol. 35, pp. 126–133, 1997.

[3] N. Yee, J.-P. M. G. Linnartz, and G. Fettweis, "Multi-Carrier-CDMA in indoor wireless networks", in *Proc. 4th Int. Symp. on Personal, Indoor and Mob. Radio Commun. PIMRC'93*, Yokohama, Japan, 1993, pp. 109–113.

[4] M. Frikel, B. Targui, F. Hamon, and M. M'Saad, "Adaptive equalization using controlled equal gain combining for uplink/downlink MC-CDMA systems", *Int. J. of Signal Process.*, vol. 4, no. 3, pp. 230–237, 2008.

[5] M. Frikel, S. Safi, B. Targui, and M. M'Saad, "Channel Identification using chaos for an uplink/downlink multicarrier code division multiple access system", *J. Telecommun. and Inform. Technol.*, no. 1, pp. 48–54, 2010.

[6] M. Boulouird and M. M'Rabet Hassani, "Blind identification of MA models using cumulants", *Int. J. of Electrical, Electronic Sci. & Engin.*, vol. 1, no. 6, pp. 619–623, 2007.

[7] ETSI, "Broadband Radio Access Networks (BRAN); High Performance Radio Logical Area Network (HIPERLAN) Type 2; Requirements and architectures for wireless broadband access", Jan. 1999.

[8] ETSI, "Broadband Radio Access Networks (BRAN); HIPERLAN Type 2; Physical Layer", Dec. 2001.

[9] M. Zidane, S. Safi, M. Sabri, A. Boumezzough, and M. Frikel, "Broadband radio access network channel identification and downlink MC-CDMA equalization", *Int. J. of Energy, Inform. and Commun.*, vol. 5, no. 2, pp. 13–34, 2014.

[10] M. Zidane, S. Safi, M. Sabri, and A. Boumezzough, "Blind Identification Channel Using Higher Order Cumulants with Application to Equalization for MC-CDMA System", *Int. J. of Elec., Robot., Electron. and Commun. Engin.*, vol. 8, no. 2, pp. 369–375, 2014.

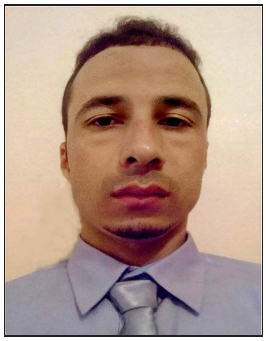
[11] S. Safi, M. Frikel, A. Zeroual, and M. M'Saad, "Higher order cumulants for identification and equalization of multicarrier spreading spectrum systems", *J. of Telecommun. and Inform. Technol.*, no. 1, pp. 74–84, 2011.

[12] M. Zidane, S. Safi, M. Sabri, A. Boumezzough, and M. Frikel, "Adaptive algorithms versus higher order cumulants for identification and equalization of MC-CDMA", *J. of Telecom. & Inform. Technol.*, no. 3, pp. 53–62, 2014.

[13] D. Brillinger and M. Rosenblatt, "Computation and interpretation of *k*-th order spectra", in *Spectral Analysis of Time Signals*, B. Harris, Ed. New York: Wiley, 1967, pp. 189–232.

[14] J. L. Peyre, D. Dembélé, and G. Favier, "Identification of MA models using higher order statistics: A general formulation", in *Proc. 2nd ATHOS Worksh. on System Identif. and Higher Order Statist.*, Sophia-Antipolis, France, 1993, pp. 20–21.

[15] A. Conti, "MC-CDMA bit error probability and outage minimization through partial combining", *IEEE Commun. Let.*, vol. 9, no. 12, pp. 1055–1057, 2005.



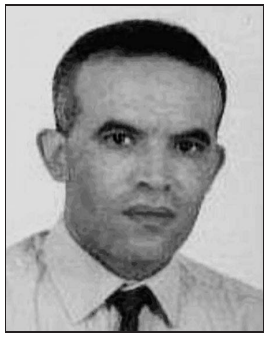
Mohammed Zidane received the M.Sc. in Electronic Engineering from Sultan Moulay Slimane University, Beni Mellal, Morocco, and M.Sc. in Optoelectronics and Laser Instrumentation from Faculty of Science and Techniques Hassan first University, Settat, Morocco. He has Ph.D. in Digital Telecommunications and Signal Processing from Sultan Moulay Slimane University, Beni Mellal, Morocco. His research interest includes digital communications and signal processing, linear and non-linear Broadband Radio Access Network (BRAN) channels identification, higher order statistics, blind identification and equalization in MC-CDMA systems, subjects on which he has published 8 journal papers and 4 conference papers.

E-mail: zidane.ilco@gmail.com
Department of Physics
Faculty of Sciences and Techniques
Sultan Moulay Slimane University
PO box 523, Beni Mellal, Morocco



Said Safi received the B.Sc. degree in Physics from Cadi Ayyad University, Marrakech, Morocco, in 1995, M.Sc. and Ph.D. degrees from Chouaib Doukkali University and Cadi Ayyad University, Morocco, in 1997 and 2002, respectively. He has been a professor of information theory and telecommunication systems at the National School for Applied Sciences, Tangier Morocco, from 2003 to 2005. Since 2006, he is a professor of applied mathematics and programming at the Faculty of Science and Technic, Beni Mellal Morocco. In 2008 he received the Ph.D. degree in telecommunication and informatics from the Cadi Ayyad University. His general interests span the areas of communications and signal processing, estimation, time-series analysis, and system identification-subjects on which he has published 10 journal papers and more than 40 conference papers. Current research topics focus on transmitter and receiver diversity techniques for single- and multi-user fading communication channels, and wide-band wireless communication systems.

E-mail: safi.said@gmail.com
Department of Mathematic and Informatic
Polydisciplinary Faculty
Sultan Moulay Slimane University
PO box 592, Beni Mellal, Morocco



Mohamed Sabri received the Ph.D. degree in Signal Processing and Telecommunications, from Rennes I University, France. His current research interests are communication networks evolution and human face detection and recognition. He is currently working as a Professor in Department of Physics, Faculty of Sciences and Techniques, University of Sultan Moulay Slimane, Beni Mellal, Morocco.

E-mail: sipt03@yahoo.fr
Department of Physics
Faculty of Sciences and Techniques
Sultan Moulay Slimane University
PO box 523 Beni Mellal, Morocco

Paper

Unsupervised Phoneme Segmentation Based on Main Energy Change for Arabic Speech

Noureddine Lachachi

Faculty of Exact and Applied Sciences, University of Oran 1 Ahmed Ben Bella, Oran, Algeria

Abstract—In this paper, a new method for segmenting speech at the phoneme level is presented. For this purpose, author uses the short-time Fourier transform of the speech signal. The goal is to identify the locations of main energy changes in frequency over time, which can be described as phoneme boundaries. A frequency range analysis and search for energy changes in individual area is applied to obtain further precision to identify speech segments that carry out vowel and consonant segment confined in small number of narrow spectral areas. This method merely utilizes the power spectrum of the signal for segmentation. There is no need for any adaptation of the parameters or training for different speakers in advance. In addition, no transcript information, neither any prior linguistic knowledge about the phonemes is needed, or voiced/unvoiced decision making is required. Segmentation results with proposed method have been compared with a manual segmentation, and compared with three same kinds of segmentation methods. These results show that 81% of the boundaries are successfully identified. This research aims to improve the acoustic parameters for all the processing systems of the Arab speech.

Keywords—band frequencies, energy changes, formant analysis, phoneme segmentation.

1. Introduction

Phonetic segmentation is the action of dividing the speech signal into its basic language functional units: the phonemes. The accurate segmentation and labeling of speech into phoneme units is useful for diverse purposes, as for example the initialization of speech recognizers, the creation of databases for concatenated text-to-speech systems, the evaluation of the performance of speech recognition tasks, and the health related assessment of speech. In this last point, there are special topics in cognitive communication information that require the segmentation of speech signal into phoneme sized units in the processing of continuous speech. There are many types of applications, where the precise knowledge of phoneme is not important, just the type of the given sound, like vowel, nasal, voiced/unvoiced fricative, stop, etc. In these applications, the linguistic content is not important, just the acoustic characteristics are needed. This kind of segmentation is necessary, when the desired behavior depends on speech timing, like rhythm or the place of voiced sounds. Moreover, such segmentation technique is useful for the visualization of the acoustical parameters of speech in an audio-visual pronunciation training system [1]–[3].

In these issues, automatic alignment tools have been developed (e.g. EasyAlign [4], SPPAS [5]). They offer a consistent and reproducible alignment at reduced cost. The task they perform is known as "linguistically constrained segmentation" or "forced alignment". In these systems, only the time boundaries of the phonemes have to be determined. For this purpose, acoustic modeling based on Hidden Markov Models (HMMs), relying on speech segmentation techniques, has been shown to achieve the best results [6].

As described for example in [7], freely spoken language consists of sequences of various phonemes. Such phonemes can be classified into both voiced and unvoiced sounds. Depending on the manner how these sounds are produced, two different cases can be distinguished. First, voiced sounds such as normal vowels are characterized by a set of several characteristic frequencies that are called formants of the respective phoneme. Second, unvoiced phonemes also show characteristic formants. However, due to the fact that these sounds do not dominantly come from an associated vibration of the vocal folds (rather turbulent and irregular air flows are involved in the corresponding sound production), these phonemes are characterized by broader frequency ranges [8].

Analysis and presentation of the speech signal in the frequency domain are of a great importance in studying the nature of speech signal and its acoustic properties. The prominent part of speech signal spectrum belongs to formants that correspond to the vocal tract resonant frequencies. These are usually referred to as $F1$ indicating the first formant, $F2$ indicating the second formant, $F3$ indicating the third formant, etc. The quality of some of the most important systems for speech recognition and speech identification as well as those for formant based speech synthesis are dependent on how accurate the formant frequencies are determined. The formant defines the range of frequencies that is used for detecting the delimitations of the phonemes in a speech signals. Hence it conduct to the task of segmentation. There are many research works on automatic speech segmentation to classify speech into phonetic classes, but in Arabic language, the segmentation has not been well studied. Therefore, this paper proposes an ef-

fective segmentation, suitable for Arabic automatic speech recognition and related applications.

The purpose of this document is to identify segments of phonemes on a frequency range limited to a narrow spectral areas. Presented study is more relevant on the spectral distribution of voice signals where six areas are used.

Using formant analysis of Arabic language, we attempt to detect vowels and consonants that are spoken. Here a standard approach for detect the phonemes in continuous speech is described based on three frequency formants: $F1$, $F2$ and $F3$ to define the range of area frequency. We have investigated the correlations between formants in each phoneme and developed an algorithm to segment speech based on the overlap different vowels in $F1-F2$ and $F2-F3$ planes.

The results, have been compared with a manual segmentation in order to calculate the accuracy that shows the performance, and have been compared with three same kinds of segmentation methods.

2. Supervised and Unsupervised Speech Segmentation

Automatic speech segmentation is the partitioning of a continuous speech signal into discrete, non-overlapping units. Generally, automatic speech segmentation methods are divided in two types.

2.1. Supervised Speech Segmentation

This methods require training on speech material and a priori knowledge [9], [10]. The segmentation algorithm relies on the linguistic knowledge associated with the input speech signal, such as its phonetic transcription or the knowledge of its phoneme sequence as well as by the number of phonemes present. This means that the representation of the utterance in terms of discrete units is known, and pretrained acoustic models of these units are needed for the forced alignment. Thus, the system is only required to locate optimally the boundary locations that best coincide with the phoneme sequence given. The task of the segmentation algorithm is then to locate optimally the phonemes boundaries [11].

2.2. Unsupervised Speech Segmentation

These methods do not require training data to segment speech signal [12], it uses a set of rules derived from the decoding of human knowledge issued of the nature of the floor to make the operation of segmentation. Indeed, the segmentation algorithms are designed without any prior linguistic knowledge about the phoneme sequence of the input speech signal. The system blindly determines the best estimate of the number of phonemes along with their boundary locations, based on the acoustic cues extracted from the speech signal.

Acoustic (rate of) change (see [13] for early work on unsupervised automatic speech segmentation and below for more recent work) is an example of prior human knowledge that is used to solve the speech segmentation task. The task for an unsupervised segmentation algorithm is based in two point. The number of segments in the speech signal needs to be determined and the position of the boundaries determined on the basic characteristics of the acoustic signal. The unsupervised methods yield a desirable and more flexible framework for the automatic segmentation of speech and their algorithms are generally simpler than used in supervised methods [14].

2.3. Unsupervised Speech Segmentation Application

Some applications of the unsupervised speech segmentation include:

- Speaker verification systems. To achieve a phoneme level segmentation (without orthographic information) of a user selectable password in a text-dependent speaker verification systems.
- Speech recognition systems. To obtain phoneme level segmentation (level modeling phoneme) in a low-to-medium size vocabulary speech recognition systems, with user-defined vocabulary (such as, in voice dialing applications).
- Language identification systems. To find a phoneme level segmentation for multilingual un-transcribed corpus applied to automatic language identification.
- Speech corpus segmentation and labeling. To obtain a great level of phoneme segmentation of a speech corpus. This can be used as seed values to aid the subsequent manual process of phonetic transcription.

3. Modern Standard Arabic

The Arabic language has a standard pronunciation, which basically is the one used to recite the Quran. The same pronunciation is used in newscasts, discourses and formal actuations of all types [15]. Spoken in the Middle East and North Africa, Arabic has different dialects where some letters are pronounced in different manner [16], [17]. However, the literary Arabic also called Modern Standard Arabic (MSA) or Al-fus-ha. One of the differences between the spoken and written Arabic is the presence of diacritics marks (spoken segments that not present in the written form). The complexity of this language is due to the unusual morphology: words are formed using a "root and pattern" scheme, where the root is composed of 3 consonants, leading to several possibilities using one root.

3.1. Arabic Phonology

Phonetically, MSA has 34 basic phonemes of which six are vowels (short vowels **/i/**, **/u/**, and **/a/** and long vowels

/i:/, **/u:/**, and **/a:/**), and 28 are consonants. The Arabic alphabet only consists of letters for long vowels and consonants. Other pronunciation phenomena, including short vowels (*harakat*), nunation (*tanwin*) and consonant doubling (*shadda*), are not typically written. However, they can be explicitly indicated using diacritics. Vowel diacritics represent the three short vowels: **a**, **i**, and **u** (*fatha*, *kasra* and *damma*) or the absence of a vowel (*sukun*).
Additionally, pharyngeal and emphatic phonemes comprise two distinctive classes of phonemes in Arabic. These two classes are found only in Semitic. These phonemes can be grouped according to the articulation of the lips and tongue during speech [18].

3.2. Arabic Syllables

The syllable types allowed in MSA are **CV**, **CVC**, and **CCV**, where **V** indicates a (long or short) vowel and **C** indicates a consonant. Arabic sentences must start with a consonant [19], and all Arabic syllables must contain at least one vowel. In addition, while such vowels cannot occur in word initial position, they can occur between two consonants or in word-final position. This is in contrast with other major languages, i.e. English, Japanese. In Japanese language, vowel can occur at any position of a word and most of the Japanese words end with vowel like pronunciation. Arabic syllables can be classified as short or long. The **CV** syllable type is a short syllable while all others are long. Syllables can also be classified as open or closed. An open syllable ends with a vowel, while a closed syllable ends with a consonant. For Arabic, a vowel always forms a syllable nucleus, and there are as many syllables in a word, as there are vowels in it [20].

3.3. Formant Analysis in Arabic Speech

It has been noted that generally most of the energy of vowel lies below 2 kHz and in case of voiced consonants lies below 3 kHz as shown in Fig. 1 [21]. Vowels are lower-frequency components of speech and create the sound volume of speech.
Vowels are among the essential components of any spoken language. The analyze and the study of vowels in Arabic is very important designing reliable and robust speech processing systems due to the fact that almost 60 to 70% of Arabic speech is vowels [22].

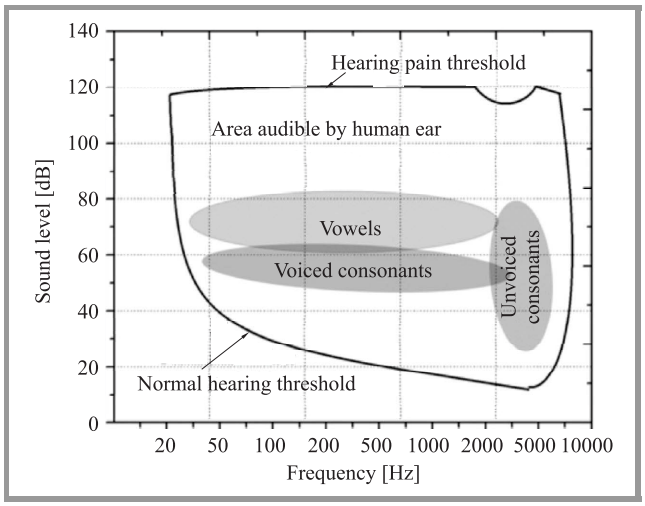


Fig. 1. Normal hearing frequency distribution of human speech.

Table 1
The relationship between the vocal tract characteristic and the two formants $F1$, $F2$

Vocal tract characteristic	$F1$	$F2$
Length of the pharyngeal oral tract	Inversely proportional	Inversely proportional
Oral constriction in the front half of the vocal tract	Inversely proportional	No effect
Pharyngeal constriction	Proportional	No effect
Back tongue constriction	No effect	Inversely proportional
Front tongue constriction	No effect	Proportional
Lip rounding	Inversely proportional	Inversely proportional

Table 1 give $F1$ and $F2$ give indication about the constrictions of the vocal tract in generating vowels [23].

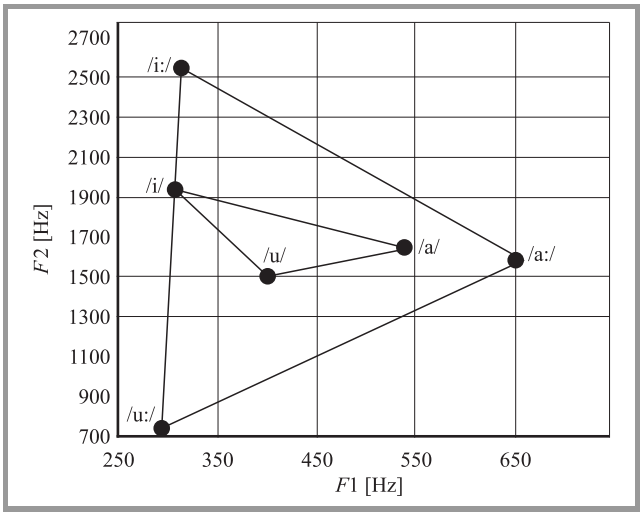


Fig. 2. The formant triangle of MSA vowel [15].

Based on the study [24], a formant-based analysis for the six vowels of MSA language was carried out and the values of the first three-formant frequencies were captured. Their results were compared to some previously published ones conducted on MSA and other Arabic dialects. The comparison was performed from geometric perspective using the Euclidean distance. The comparison results were found to be consistent with the visual inspection of the vowel triangles as shown in Fig. 2.

In Fig. 2, one can see that the vowels **/i:/** and **/i/** have low frequencies in $F1$ and high frequencies in $F2$. Moreover, the frequencies $F1$ and $F2$ are both low for the vowels **/u:/** and **/u/**. In the case of **/a:/** and **/a/**, both have $F1$ with high frequency and $F2$ with an average frequency. Therefore, when these vowels are plotted $F1$ to $F2$, they form two triangles.

The results of the analysis of the first three formants are summarized in Table 2 where speakers uttered perfectly Arabic phonemes without any influence by their local dialects [24].

Table 2
Results of $F1$, $F2$ and $F3$

Vowels	$F1$ [Hz]	$F2$ [Hz]	$F3$ [Hz]
/a:/	651.5	1588.1	3058.3
/i:/	314.1	2549.8	3278.9
/u:/	295.4	744.3	2560.2
/a/	535.0	1635.0	5890.6
/i/	307.5	1942.1	2702.7
/u/	407.9	1520.3	2777.7

4. Methodology

This section outlines in detail the settings of band frequencies, algorithm and computation conducted with references to the research presented in this paper. The proposed strategy based spectral analysis extracts phonemes from the raw speech waveforms. It requires no learning and it is language independent applied for Arabic speech.

4.1. Band Frequencies Definition

It is well known that an acoustic speech signal contains information beyond its linguistic content. This paralinguistic information includes clues to a speaker's accent and identity, which are exploited by automatic accent identification (AID) and speaker identification (SID) systems. The relationship between AID and SID is asymmetric, since accent information is relevant to SID but speaker information is a distraction in the context of AID.

For instance, the speaker identification study in [25], performed on the clean TIMIT corpus using mono Gaussian modeling, showed that the frequency regions below 600 Hz and above 3000 Hz provided better SID than the middle frequency regions. However, no similar study has been conducted for AID. In [26], the contrasting importance of different frequency bands for AID and SID are investigated, using contemporary GMM-based systems. These bands are defined in center frequency shown in Table 3 [26].

According to [26], it is useful to divide the spectrum into four areas: A (0 to 0.77 kHz), B (0.34 to 3.44 kHz), C (2.23 to 5.25 kHz) and D (3.40 to 11.02 kHz). The results suggest that speaker information dominates in areas A and D. The first area A, corresponding to primary vocal tract resonance information, and the second area D, corresponding to high-frequency sounds. These results are consistent with [25]. In contrast, area B is most useful for AID, indicating that the vocal tract resonance information in this region is linguistic biased, rather than speaker information. Area C contains both types of information, although speaker information appears dominant.

Table 3
The center frequency for 31 Mel-spaced band-pass filters [26]

Filter number	Center frequency [Hz]	Filter number	Center frequency [Hz]
1	129	17	2239
2	258	18	2497
3	344	19	2799
4	473	20	3100
5	559	21	3445
6	645	22	3832
7	775	23	4263
8	861	24	4737
9	990	25	5254
10	1076	26	5857
11	1205	27	6503
12	1335	28	7235
13	1464	29	8253
14	1636	30	8957
15	1808	31	9948
16	2024		

Based on the assumption that the majority of phonemes used in Spanish language are used in Arabic language, we consider the study given in [27].

In [27], it is shown that for Portuguese language (or Spanish language) there are 48 different phonemes used for the SID grouped into 11 classes. These classes are: silence, voiced fricative and unvoiced fricative, voiced plosive and unvoiced plosive, nasal consonants, nasal vowels, front vowels, median vowels, back vowels and liquid consonants.

For each class, a given set of representative parameters is largely used for phoneme classification. The parameters used for each class are as follows [28].

Silence – only the total energy of the analysis window is used (threshold –35.8 dB). The boundary between the silence and other classes is set up at the frame where the total energy becomes greater than the threshold.

Vowels (median, front, back, nasal) – four parameters are used: total energy of the analysis window, first ($F1$) and second ($F2$) formant values and energy profile. The transition between vowels and the other classes is determined

by using the total energy of the analysis windows (transition is set where the energy is below –28 dB). Energy profile, $F1$ and $F2$ values are used to separate vowels in diphthongs. $F1$ is used to separate median vowels from back and front vowels, and the boundary is set up at the frame where $F1$ is below 673 Hz. Energy profile and $F2$ value are used to separate front vowels from back vowels. The transition is determined at the frame where $F2$ is below 1845 Hz and the energy profile is below 2106 Hz. Energy profile represents the frequency band carrying a given percentage of the total energy and is calculated from the Discrete Fourier Transform (DFT) of the windowed speech signal.

Fricative (voiced and unvoiced) – two parameters are used: Zero Crossing Rate [29] (thresholds 0.35 for voiced fricatives and 0.62 for unvoiced fricatives) and gravity spectral center (threshold 2500 Hz). The gravity spectral center represents the frequencies where 50% of the total energy of the windowed signal is concentrated. The transition from fricatives to other classes is determined at the frame where the parameters values are below the thresholds.

Plosive (voiced and unvoiced) – three parameters are employed: energy in the frequency bands $[0-F3]$ and $[F3-f_s/2]$ [30] and the first order derivative of $F2$, where $F2$ and $F3$ represent the second and third formant frequencies and f_s is the sampling frequency. As the derivative of $F2$ exhibits a peak at the transition from plosive to other classes where the peak position represents the boundary. Energy is combined with the derivative permit to avoid spurious peaks. The energy in the frequency band $[0-F3]$ for voiced plosive is above of –5 dB and in the frequency band $[F3-f_s/2]$ is above of –2 dB. For unvoiced plosive the energy is above of 5 dB and 10 dB for the bands $[0-F3]$ and $[F3-f_s/2]$ respectively.

Nasal consonants – two parameters are used: $F1$ value (threshold 280 Hz) and the ratio between the spectral energy in the frequency bands $[0-353]$ Hz and $[358-5373]$ Hz (threshold 0.87). When the $F1$ value is greater than 280 Hz and the spectral energy ratio is below 0.87, a transition has occurred from nasal consonant to another class.

Liquids – two parameters are employed: spectral energy band $[0-2600]$ Hz (threshold above 6.5 dB) and its first order derivative. Transition from liquid to another class tends to exhibit a peak in the first derivative of the spectral energy. The peak determines the transition and at this frame, the spectral energy threshold must be below 6.5 dB.

Based the study in [26], [27] and the one of formant frequencies defined in Arabic speech developed in Section 3, one can see that if we divide all frequencies centers [26] indicated in Table 3 into six zones (Table 4), we get closer to the syntheses given above [28].

To investigate the effect of different frequencies areas, segmentation experiments were conducted using frequency band limited speech data comprising the outputs of adjacent filters regions. For example in **LF2** area, we considered $k=4$ overlapping sub-bands, where the N-th sub-band comprises the outputs of filters N to $+3$ ($N=1\ldots4$).

Table 4
Definition of the six region band frequencies

LF – low frequency			
Band LF1	Center frequency [Hz]	Band LF2	Center frequency [Hz]
1	129	1	559
2	258	2	645
3	344	3	775
4	473	4	861
5	559	5	960
		6	1071
MF – medium frequency			
Band MF1	Center frequency [Hz]	Band MF2	Center frequency [Hz]
1	1076	1	1808
2	1205	2	2024
3	1335	3	2239
4	1464	4	2457
5	1636	5	2799
6	1808	6	3100
HF – high frequency			
Band HF1	Center frequency [Hz]	Band HF2	Center frequency [Hz]
1	3100	1	5254
2	3445	2	5854
3	3832	3	6503
4	4263	4	7235
5	4737	5	8000
6	5254		

4.2. Energy Computation over a Frequency Band

For the human ear perceiving speech along a nonlinear scale in the frequency domain [31], one approach is to use a uniformly space-warped frequency scale, such as the Mel scale.

The relation between Mel-scale frequency and frequency (Hz) is described by the following equation:

$$Mel = 2595\log(1+f/700), \tag{1}$$

where Mel is the Mel-frequency scale and f is in Hz. The filter bank is then designed according to the Mel scale. For example, we take 4 frequency bands in LF2 area (see Table 4) that are approximated by simulating 4 triangular band-pass filters, (i,k) $(1\leq i\leq 4, 11\leq k\leq 21)$. Over a frequency range of $559-1076$ Hz, we consider that the speech signal is sampled at 16 kHz windowed over 10 ms

(each window of 10 ms has 160 point), and the spacing as well as the bandwidth are determined by a constant Mel frequency interval by Eq. 1. Considering a given time-domain noisy speech signal, $x_{time}(m,n)$, representing the magnitude of the n-th point of the m-th frame, we first find the spectrum, $x_{freq}(m,k)$, of this signal by 160-point DFT:

$$x_{freq}(m,k) = \sum_{n=0}^{N-1} x_{time}(m,n) W_N^{kn}, \tag{2}$$

$$0 \le k \le N-1,\ 0 \le m \le M-1,$$

$$W_N = e^{\frac{-j2\pi}{N}}, \tag{3}$$

where $x_{freq}(m,k)$ is the magnitude of the k-th point of the spectrum of the m-th frame, N is 160 and M is the number of frames of the speech signal for analysis. Then, we multiply the spectrum $x_{freq}(m,k)$ by the weighting factors $f(i,k)$ on the Mel-scale frequency bank and sum the products for all k to get the energy $x(m,i)$ of each frequency band i of the m-th frame:

$$x(m,i) = \sum_{k=0}^{N-1} \left| x_{freq}(m,k) \right| f(i,k), \tag{4}$$

$$0 \le m \le M,\ 1 \le i \le 20,$$

where i is the filter band index, k is the spectrum index, m is the frame number, and M is the number of frames for analysis.

In order to remove some undesired impulse noise in Eq. 4, we further smooth it by using a three-point median filter to get $x_s(m,i)$:

$$x_s(m,i) = \frac{x(m-1,i) + x(m,i) + x(m+1,i)}{3}. \tag{5}$$

Finally, the smoothed energy, $x_s(m,i)$, is normalized by removing the frequency energy of background noise, $Noise_{freq}$, to get the energy of almost pure speech signal, $X(m,i)$. The smoothed and normalized frequency energies of a clean speech signal, $X(m,i)$ is described by Eq. 6. The energy of background noise is estimated by averaging the frequency energy of the first five frames of the recording:

$$X(m,i) = x_s(m,i) - Noise_{freq} = x_s(m,i) - \frac{\sum_{n=0}^{4} x_s(n,i)}{5}. \tag{6}$$

With the smoothed and normalized energy of the i-th band of the m-th frame, $X(m,i)$, we can calculate the total energy of the almost pure speech signal at the i-th band as $E(i)$:

$$E(i) = \sum_{m=0}^{M-1} \left| X(m,i) \right|. \tag{7}$$

The goal is to select some useful bands area having the maximum word signal information. It is obvious that $E(i)$ in Eq. 7 is a good indicator since the band with higher $E(i)$ contains more pure speech information.

Based on this computation for each band area cited in Table 4, the Fig. 3 shows the six energies computed of the six areas frequencies that specify each vocal segment of a speech Arabic signal for 2 s.

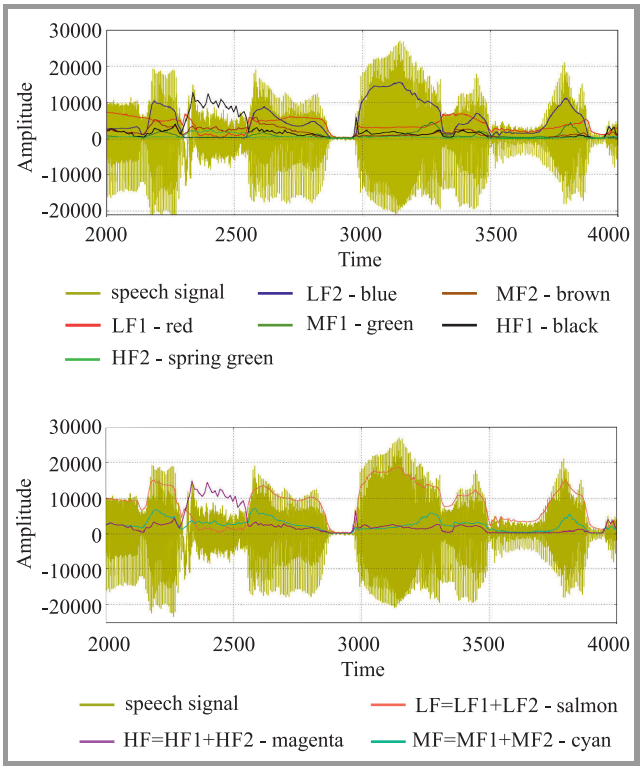


Fig. 3. Energies of six region bands in an Arabic speech signal frame. (See color pictures online at www.nit.eu/publications/journal-jtit)

4.3. Segmentation Algorithm

In each frame of an Arabic speech signal, the segmentation is based on three steps.

1. All closure and fricative phonemes for all point in segment where HF1 energy signal is greater than the sum of the energies signals LF1 and LF2 are selected (Fig. 4).
2. The vocalic segment for all point in segment where the sum of the energies signals LF1 and LF2 is greater than the mean of the energy signal HF2 is selected (Fig. 5).
3. The vowels and other voiced consonant in vocalic segment for all segment are selected (Fig. 6) where we are:
   - crossing between energy signal LF1 and energy signal LF2,
   - crossing between energy signal LF1 and the sum of the energies signals MF1 and MF2,
   - crossing between energy signal LF2 and the sum of the energies signals MF1 and MF2.

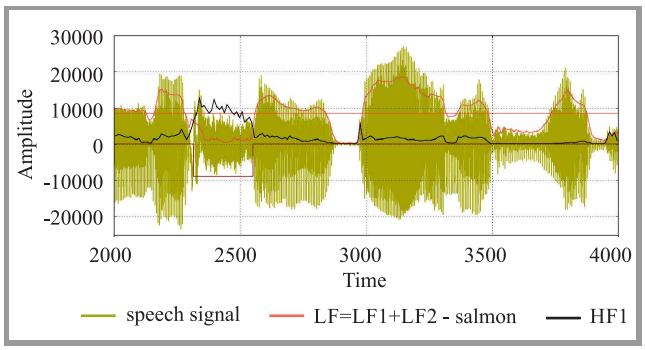


Fig. 4. Selection of a closure phoneme (first step).

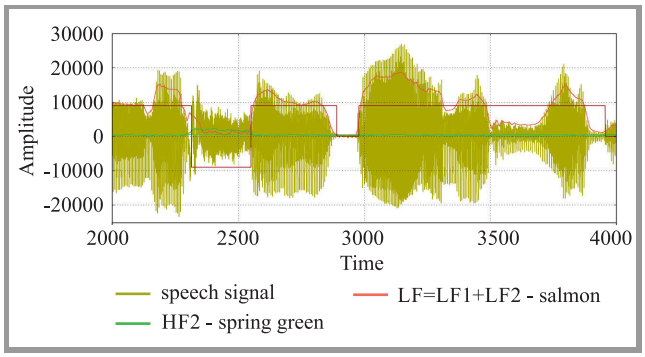


Fig. 5. Selection of vocalic segment (second step).

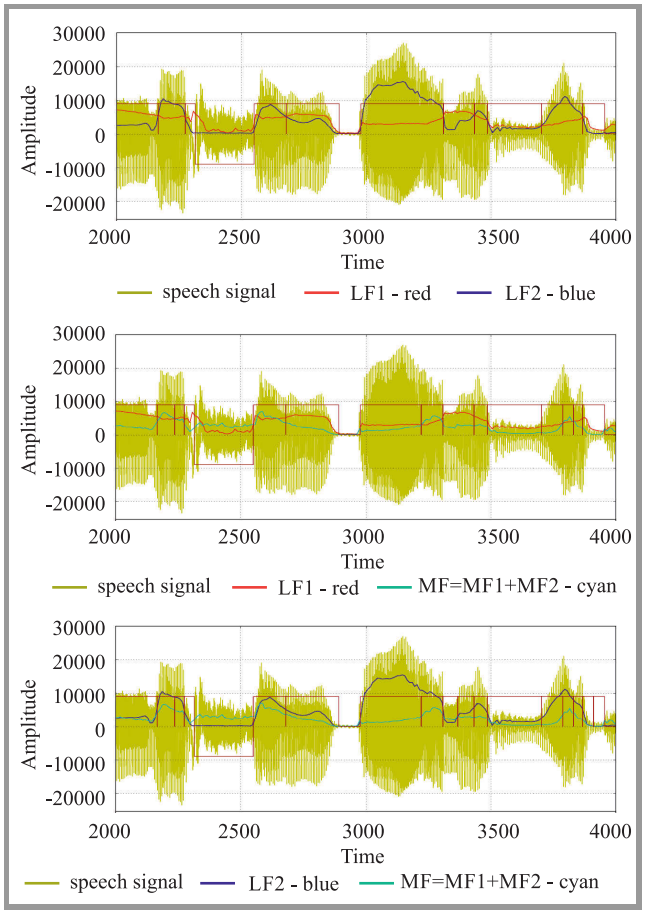


Fig. 6. Selection of vowels and voiced consonant (third step).

5. Experimentation and Evaluation

5.1. Data Set

The speech was recorded at a sampling rate of 44.1 kHz by means of a 16-bit mono analog-to-digital converter (ADC) per sample and they were down sampled to 16 kHz. Data are recorded with the help of a unidirectional microphone using Audacity recording tool in a normal room with minimum external noise. Ten subjects (10 male) in the 22–35 age range were participated in the recording process. All subjects were monodialectal speakers of MSA. They were free of any speech or hearing disorders by self-report based on a screening interview and as later judged by the investigator during the recording session. Each subject recorded twenty verses in Quranic recitation according to the *tajweed rules*. Then, all the files recorded of the data set are segmented into fixed size of 30 s. Additionally, a silence period is added to the beginning and end of each sample file. The input speech data are pre emphasized with coefficient of 0.97 using a first order digital filter and then window by a Hamming window. The resulting windowed frames of 20 ms are used for the phoneme boundary detection in our experiment. For comparison of boundaries detection does with proposed algorithm, the task of transcription of phonemes for our entire data set is done by an expert phonetician.

5.2. Performance Measure

In order to evaluate the proposed algorithm, the metrics required for speech segmentation performance evaluation are used whose definitions are as follow:

- H_R (hit rate): represents the rate of correctly detected boundaries ($\frac{N_H}{N_R}$). It utilizes the number of correctly detected boundaries (N_H) and the total number of boundaries (N_R);
- F_A (false alarm rate): represents the rate of erroneously detected boundaries $\frac{(N_T - N_H)}{N_T}$, which utilizes the total number of detected boundaries N_T and the number of correctly detected boundaries N_H;
- OS (over segmentation rate): shows how much more (or less) is total number of algorithm detections, compared to the total number of reference boundaries taken from the manual transcription $\frac{(N_T - N_R)}{N_R}$;
- PCR (precision rate) $=1 - F_A$: describes the likelihood of how often algorithm identifies a correct boundary whenever a boundary is detected.

The overall quality of proposed algorithm is described by computing F_{mesure} from precision rate and hit rate whose expression is $F_{mesure} = \frac{(2 \times PCR \times H_R)}{(PCR + H_R)}$. Another global measure, referred to as the R_{mesure}, decreases as the distance to the target grows, i.e. similarly as the F_{mesure} does, but

is critical towards over-segmentation [32]. It calculated by $R_{mesure} = 1 - \frac{(|r1| + |r2|)}{2}$ with $r1 = \sqrt{(1-H_R)^2 + OS^2}$ and $r2 = \frac{(H_R - OS - 1)}{\sqrt{2}}$.

5.3. Performance Evaluation

By observing the Figs. 7 and 8, the segmentation appears in concordance with the spectrum. Compared to the manual transcription showed in the two figures, segmentation gives more information characteristic definition of the speaker and the phonemes have better boundaries.

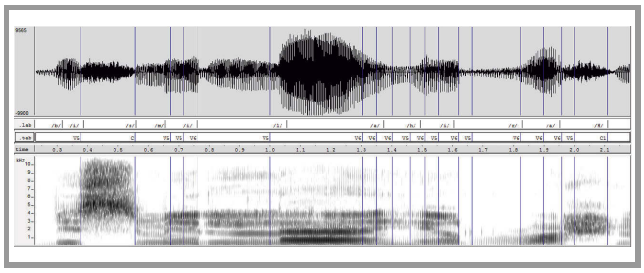


Fig. 7. Segmentation of Basmala (Surat Fatiha – Holy Coran).

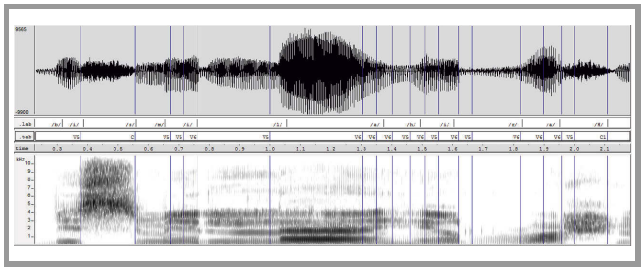


Fig. 8. Segmentation Verset 02 of Fatiha (Holy Coran).

The above calculations (Subsection 4.3) were performed for the analysis of the results obtained through the application of aforementioned algorithm.
The methodology was repeated on 83 different files of Quranic Arabic speech obtained from trained speakers.

Table 5
Segmentation performance

Files	N_H	N_R	N_T	$F_{measure}$ [%]
Speaker 01	87	109	105	81.31
Speaker 02	78	118	108	69.03
Speaker 03	76	105	98	74.88
Speaker 04	84	113	98	79.62
Speaker 05	83	109	101	79.05
Speaker 06	103	123	112	87.66
Speaker 07	105	127	118	85.71
Speaker 08	96	117	103	87.29
Speaker 09	107	124	115	89.54
Speaker 10	85	111	102	79.81
Mean measure	90	115	106	81.39

Speech signal was divided into different frames. For each frame, the trends of the signal to find the number of consecutive boundaries specifying phonemes were checked. As a result, each vowel or consonant detected, starting boundary, ending boundary of each phoneme is transcript.
To illustrate this, the results generated algorithmically from 10 different files of various speakers are presented in Table 5. The table shows the total number of different limits and measuring performance during the application of the proposed methodology.

5.4. Comparison Test

The proposed method was compared with the three same kinds of segmentation methods using mean $F_{measure}$ shown in Table 6. The first method [33] uses average level crossing rate (ALCR) and root-mean-square (RMS) energy to detect the phonetic boundary between obstruent initial consonant and preceding/following vowel. The second method [34] uses frequency synchrony and average signal level as input to a two-layered support vector machine based (SVM) system to detect phoneme boundaries. The third method [35] uses unsupervised phoneme boundary detection based on band-energy tracing technique.

Table 6
Comparison of segmentation performance

Method	PCR [%]	H_R [%]	$F_{measure}$ [%]
First method [33]	79.82	78.83	79.32
Second method [34]	81.12	78.91	79.99
Third method [34]	82.33	75.07	78.53
Proposed method	85.11	78.01	81.39

6. Conclusion

This work proves that it is possible to extract the information of phonemes from the energy of the acoustic signal. Following the formant technique, a study is done on Modern standard Arabic vowels. It shows that it has six basic vowels included in the constricting of vocal tract that has permit to the segmentation to be deployed in proposed system. The system shows that the formants are very effective for detecting phonemes correctly.
The experimentation shows that with this method, we can detect a mean of 81% of all boundaries manually transcribed of a speech raw file, and give better result than other methods developed in the literature.

References

[1] K. Vicsi and D. Sztahó, "Recognition of emotions on the basis of different levels of speech segments", *J. of Adv. Comput. Intell. and Intelligent Inform.*, vol. 16, no. 2, pp. 335–340, 2012.

[2] K. Vicsi, D. Sztahó, and G. Kiss, "Examination of the sensitivity of acoustic-phonetic parameters of speech to depression", in *Proc. 3rd IEEE Int. Conf. on Cognitive Infocommun. CogInfoCom 2012*, Kosice, Slovakia, 2012, pp. 511–515 (doi: 10.1109/CogInfoCom.2012.6422035).

[3] K. Vicsi, V. Imre, and G. Kiss, "Improving the classification of healthy and pathological continuous speech", in *Proc. 15th Int. Conf. Text, Speech and Dialogue TSD 2012*, Brno, Czech Republic, 2012, pp. 581–588.

[4] J. P. Goldman, "EasyAlign: An automatic phonetic alignment tool under Praat", in *Proc. 12th Ann. Conf. of the Int. Speech Commun. Assoc. Interspeech 2011*, Florence, Italy, 2011.

[5] B. Bigi and D. Hirst, "Speech phonetization alignment and syllabication (SPPAS): A tool for the automatic analysis of speech prosody", in *Proc. 6th Int. Conf. Speech Prosody*, Shanghai, China, 2012.

[6] S. Brognaux and T. Drugman, "HMM-based speech segmentation: Improvements of fully automatic approaches", *IEEE/ACM Trans. on Audio, Speech, and Lang. Process.*, vol. 24, no. 1, pp. 5–15, 2016.

[7] G. Gosztolya and L. Toth, "Detection of phoneme boundaries using spiking neurons", in *Proc. 9th Intell. Conf. on Artif. Intell. and Soft Comput. ICAISC 2008*, Zakopane, Poland, 2008, pp. 782–793.

[8] E. C. Zsiga, *The Sounds of Language: An Introduction to Phonetics and Phonology*. Chichester, UK: Wiley, 2012.

[9] M. Malcangi, "Soft computing approach to segmentation of speech in phonetic units", *Int. J. of Computers and Commun.*, vol. 3, no. 3, pp. 41–48, 2009.

[10] G. Kiss, D. Sztahó, and K. Vicsi, "Language independent automatic speech segmentation into phoneme-like units on the base of acoustic distinctive features", in *Proc. 4th IEEE Int. Conf. on Cognitive Infocommun. CogInfoCom 2013*, Budapest, Hungary, 2013, pp. 579–582.

[11] A. Stolcke *et al*., "Highly accurate phonetic segmentation using boundary correction models and system fusion", in *Proc. of IEEE Int. Conf. on Acoust., Speech and Signal Process. ICASSP 2014*, Florence, Italy, 2014, pp. 5552–5556 (doi: 10.1109/ICASSP.2014.6854665).

[12] O. Scharenborg, V. Wan, and M. Ernestus, "Unsupervised speech segmentation: An analysis of the hypothesized phone boundaries", *J. of the Acoust. Soc. of America*, vol. 127, no. 2, pp. 1084–1095, 2010 (doi: 10.1121/1.3277194).

[13] M. Sharma and R. Mammone, "Blind speech segmentation: Automatic segmentation of speech without linguistic knowledge", in *Proc. of Int. Conf. on Spoken Lang. Process. ICSLP 96*, Philadelpia, USA, 1996, pp. 1237–1240.

[14] S. Dusan and L. Rabiner, "On the relation between maximum spectral transition positions and phone boundaries", in *Proc. 9th Int. Conf. on Spoken Lang. Process. INTERSPEECH 2006 – ICSLP*, Pittsburgh, PA, USA, 2006, pp. 645–648.

[15] Y. A. Alotaibi and S. A. Selouani, "Evaluating the MSA West Point Speech Corpus", *Int. J. of Comp. Process. of Lang.*, vol. 22, no. 4, pp. 285–304, 2009.

[16] O. A. A. Ali, M. M. Moselhy, and A. Bzeih, "A comparative study of Arabic speech recognition", in *Proc. 16th IEEE Mediterranean in Electrotech. Conf. MELECON 2012*, Hammamet, Tunisia, 2012.

[17] F. Biadsy, J. Hirschberg, and N. Habash, "Spoken Arabic dialect identification using phonotactic modeling", in *Proc. of Worksh. on Computat. Approaches to Semitic Lang.*, Athens, Greece, pp. 53–61, 2009.

[18] N. Hajj and M. Awad, "Weighted entropy cortical algorithms for isolated Arabic speech recognition", in *Proc. Int. Joint Conf. on Neural Netw. IJCNN 2013*, Dallas, TX, USA, 2013 (doi: 10.1109/IJCNN.2013.6706753).

[19] J. F. Bonnot, "Experimentale de Certains aspects de la germination et de l'emphase en Arabe", *Travaux de l'Institut Phonétique de Strasbourg*, vol. 11, pp. 109–118, 1979 (in French).

[20] M. Alkhouli, "Alaswaat Alaghawaiyah", Daar Alfalah, Jordan, 1990 (in Arabic).

[21] A. Biswas, P. K. Sahu, A. Bhowmick, and M. Chandra, "Admissible wavelet packet sub-band-based harmonic energy features for Hindi phoneme recognition", *J. of IET Sig. Process.*, vol. 9, no. 6, pp. 511–519, 2015.

[22] A. Nabil and M. Hesham, "Formant distortion after codecs for Arabic", in *Proceeding of the 4th Int. Symp. on Commun. Control and Sig. Process. ISCCSP 2010*, Limassol, Cyprus, 2010, pp. 1–5.

[23] J. R. Deller Jr., J. H. L. Hansen, and J. G. Proakis, *Discrete-Time Processing of Speech Signals*. Wiley, 2000.

[24] Y. M. Seddiq and Y. A. Alotaibi, "Formant based analysis of vowels in Modern Standard Arabic – Preliminary results", in *Proc. 11th Int. Conf. on Inform. Sci., Sig. Process. and their Appl. ISSPA 2012*, Montreal, QC, Canada, 2012, pp. 689–694.

[25] L. Besacier, J. Bonastre, and C. Fredouille, "Localization and selection of speaker-specific information with statistical modeling", *Speech Commun.*, vol. 31, pp. 89–106, 2000.

[26] S. Safavi, A. Hanani, M. Russell, P. Jancovic, and M. J. Carey, "Contrasting the effects of different frequency bands on speaker and accent identification", in *Proc. of IEEE Sig. Process. Lett.*, vol. 19, no. 12, pp. 829–832, 2012.

[27] A. M. Selmini and F. Violaro, "Acoustic-phonetic features for refining the explicit speech segmentation", in *Proc. 8th Ann. Conf. Int. Speech Commun. Assoc. INTERSPEECH 2007*, Antwerp, Belgium, 2007, pp. 1314–1317.

[28] A. M. Selmini and F. Violaro, "Improving the Explicit automatic speech segmentation provided by HMMs", in *Proc. of the Int. Worksh. on Telecommun. IWT 2007*, Santa Rita do Sapucaí, Brazil, 2007, pp. 220–226.

[29] M. A. Ben Messaoud, A. Bouzid, and N. Ellouze, "Automatic segmentation of the clean speech signal", *Int. J. of Elec., Comp., Energe., Electron. & Commun. Engin.*, vol. 9, no. 1, pp. 114–117, 2015.

[30] A. Juneja, "Speech recognition based on phonetic features and acoustic landmarks", PhD Thesis, University of Maryland, College Park, USA, 2004.

[31] S. B. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences", *Proc. IEEE Trans. Acoust. Speech & Sig. Process.*, vol. 28, no. 4, pp. 357–366, 1980.

[32] O. J. Rasanen, U. K. Laine, and T. Altosaar, "An improved speech segmentation quality measure: the R-value", in *Proc. 10th Ann. Conf. of the Int. Speech Commun. Assoc. INTERSPEECH 2009*, Brighton, UK, 2009, pp. 1851–1854.

[33] S. Potisuk, "A novel method for blind segmentation of Thai continuous speech", in *Proc. of IEEE Sig. Process. & Signal Process. Edu. Worksh. SP/SPE 2015*, Snowbird, UT, USA, 2015, pp. 415–420.

[34] S. King and M. Hasegawa-Johnson, "Accurate speech segmentation by mimicking human auditory processing", in *Proc. of IEEE Int. Conf. on Acoust., Speech & Sig. Process. ICASSP 2013*, Vancouver, BC, Canada, 2013, pp. 8096–8100.

[35] D.-T. Hoang and H.-C. Wang, "A phone segmentation method and its evaluation on mandarin speech corpus", in *Proc. of 8th Int. Symp. on Chinese Spoken Lang. Process. ISCSLP 2012*, Hong Kong, China, 2012, pp. 373–377.



Noureddine Lachachi received his Ph.D. in Computer Science at the University of Oran 1 Ahmed ben Bella. He is a Professor at the Department of computer science, University of Oran 1 Ahmed ben Bella. His research interests include issues related to speech processing specialized in Arabic and Maghreb dialects identification systems. He is an author of some research studies published at national and international journals, conference proceedings.

E-mail: Lach_nour@yahoo.fr
Department of Computer Science
Faculty of Exact and Applied Sciences
University of Oran 1 Ahmed Ben Bella
Oran, Algeria

Paper

Optimization of Spectrum Sensing Parameters in Cognitive Radio Using Adaptive Genetic Algorithm

Subhajit Chatterjee[1], Swaham Dutta[2], Partha Pratim Bhattacharya[3], and Jibendu Sekhar Roy[4]

[1] *University of Engineering and Management, Kolkata, West Bengal, India*
[2] *Reliance Jio Infocomm Ltd., Big Data Analysis, Mumbai, Maharashtra, India*
[3] *Mody University of Science and Technology, Lakshmangarh, Sikar, Rajasthan, India*
[4] *School of Electronics Engineering, KIIT University, Bhubaneswar, Odisha, India*

Abstract—Quality of service parameters of cognitive radio, like, bandwidth, throughput and spectral efficiency are optimized using adaptive and demand based genetic algorithm. Simulation results show that the proposed method gives better real life solution to the cognitive radio network than other known approach.

Keywords—adaptive genetic algorithm, bandwidth, cognitive radio, spectrum sensing, throughput.

1. Introduction

Launching of new wireless services has become difficult due to the shortage of available radio spectrum and the technology cognitive radio (CR) is capable for providing an intelligent solution for better spectrum utilization. The occupancy and non-occupancy of the channels can be sensed by CR transceiver and instantly get into the non-occupied channel avoiding preoccupied ones, which minimizes the interference with other users [1].

Spectrum access is the main perception on which CR is based on. The contributors to the CR are mainly the license holders known as the primary users (PU) allow permission for accessing the spectrum to the non-licensed users, i.e. secondary users (SU) as long as interference to PU activity is minimal and limited [2].

Spectrum sensing is the most crucial activity in CR because the SU senses a spectrum to check the presence or absence of a PU signal and dependent upon the QoS parameters or sensing parameters like bandwidth, signal-to-noise ratio (SNR), bit error probability, spectral efficiency, throughput. In this work, sensing parameters, i.e. bandwidth, throughput, and spectral efficiency, are considered and studied. These parameters are optimized on time varying characteristic of spectrum hole under deliberation. Implementation of cognitive radio system lies on two primary steps, spectrum sensing and estimation [3], [4]. SU performs the spectrum sensing which involves sensing to detect the presence of any PU signals to avoid interference and identify accessing opportunity by SU (secondary access). In this work, a genetic algorithm (GA) is used as dependable tool applicable to the radio environment because adaptability is perfect and spectrum efficiency can reach up to 98.50% resulting decrement of sensing time [1], [5], [6].

2. Related Work

Quang *et al.* [6] proposed a throughput-aware routing algorithm for enhancing throughput and decreasing end-to-end delay in industrial cognitive radio sensor networks. The limitation of proposed algorithm is the fact that it requires extra-equipped cluster-heads.

Le *et al.* [7] proposed a bandwidth-aware localized routing algorithm that is suitable for applying to large networks since it is capable of reducing the high computational complexity in such networks.

Kaur *et al.* [4] and Kaur *et al.* [8] proposed algorithms based on the prime principles of GA for optimization of the radio transmission parameters. But, as spectrum hole has the time-varying characteristic in cognitive radio network, the activity of the primary user is one of the concerned factors in time-varying characteristic, i.e. the heterogeneous nature of the spectrum hole is observed by CR user [9].

A number of measurement campaigns relating to spectrum occupancy were conducted worldwide [10], [11]. It is found in one of the campaign that a mean occupancy is as low as 17.4% in the frequency band 30 to 3000 MHz.

3. Demand Based Adaptive Genetic Algorithm

To improve the performance an adaptive GA is proposed where operators and parameters are adaptive to the changing conditions of the spectrum and are executed by controlling the operators and parameters in such a manner that they will alter the values if the population is not producing individuals fit enough. In this work roulette wheel selection is employed with Monte Carlo adaptation by incorporating the time and geographical variance of the radio spectrum.

The simulation of the CR engine is performed by GA to determine the optimal set of sensing parameters. A system is guided by fitness objective towards an optimal state. To achieve this, one multiobjective fitness function is used with weighted sum approach with a purpose each objective can have a representation by a level to symbolize its weight. The algorithm runs with different sets of sensing parameters, denoted as application requested value. The application requested value are time varied for each parameter.

The structure of GA chromosome is composed of three parameters or genes: bandwidth (BW), signal-to-noise ratio (SNR) and bit error probability. Integration of these parameters forms a string (chromosome). Table 1 gives the summarized values of the order of the genes, the ranges of operation and the binary bits required to encode the corresponding integer values.

Table 1
Summarized values of the chromosome structure

Gene set	1	2	3
	Bandwidth [MHz]	Throughput [bits/channel]	Spectral efficiency [bits/s/Hz]
Range	450 to 3000 3000	59795 to 373717	0.000064 to 0.00032
Step size	10	4900	0.0000020
Decimal value range	2550	313922	0.000256
Number of bits required	8	6	7

The three set of parameters specified as genes in chromosome structure need 21 bits for its construction. The composition of this bit string is important because the mutation operation performs at bit level. It is assumed that the parameters be x_1, x_2 and x_3 corresponding to the bandwidth, spectral efficiency and throughput, respectively. Fitness functions for each parameter are generated by [4]:

$$f_i = \left[\frac{w_i |x_i - x_i^d|}{x_i^d} \right], \quad \text{if} \quad |x_i - x_i^d| < |x_i^d|, \tag{1}$$

where x_d^i is the required QoS parameter. w_i is the weight subject to

$$\sum_{i=1}^{3} w_i = 1, \tag{2}$$

where $i = 1, 2, 3$.

Overall fitness function f_{total} is the cumulative sum of individual fitness functions of the parameters given as:

$$f_{total} = \sum_{1}^{3} f_i. \tag{3}$$

Ideally each w_i should be equal 25%, which signifies each gene will have the same weight. But in practical scenario, the weighing factor w can vary according to QoS specifications. The probability of selection for each individual chromosome is given by:

$$p_i = \frac{f_i}{\sum_{i=1}^{n} f_i}, \tag{4}$$

where p_i is the probability of selection of individual chromosome, f_i is the fitness of the individual gene i and n is the number of chromosome in the population.

The throughput can be expressed as [12]

$$Throughput = K\left[\log_2 N_t + \log_2 M\right], \tag{5}$$

where K is the number of primary users, N_t is the Line of Sight (LOS) of transmitting antennas and M is the cardinality of the modulation scheme (in power of 2).
Table 2 shows the working functions of the parameters.

Table 2
Functions of the parameters

Parameters	Equation (function)	Remarks	Constant values
Throughput	Throughput given by Eq. 5	K = number of primary users; N_t = LOS of transmitting antennas; M = cardinality of the modulation scheme	8 users per 200 kHz
Spectral efficiency (SE)	$SE = \frac{mR}{B}$	m = modulation index; R = symbol rate; B = frequency	$m = 0.7$

4. Derivation of TFM and GFM with Chromosome

First step is the creation of initial population and each element of the initial population matrix is represented in binary form. In order to encode the real values of each parameter, corresponding decimal values are used to map them to each binary set of numbers. These decimal values are the reference values on which the total fitness measure (TFM) and gene fitness measure (GFM) are calculated. Apart from this, application requested gene value (taken as user input), gene weight (derived with respect to the gene configuration which is taken as user input) and fitness point of genes (taken as user input) are three prime aspects on which the above parameters are also dependent.

In this work each gene is assigned with decimal value. These values are derived from number of bits assigned to

the individual gene and the step size. The individual genes are frequency (bandwidth) marked as Gene 1, throughput (Gene 2) and spectral efficiency (Gene 3). Each gene has an operating range divided into few decimal values and theirs assigned value is dependent on the step size of individual gene. To derive TFM and GFM with chromosome, the decimal values of genes, configuration of gene, and total number of bits in a chromosome are designed. Gene weight is calculated and the demand and need based adaptability of the GA is taken into consideration, i.e. the application requested value of each gene. The application requested value of bandwidth can be of any value, which corresponds to a frequency and this frequency has corresponding application requested values of other genes in the chromosome structure, i.e. SNR, spectral efficiency, and throughput. This application requested values represent the respective values of the mentioned genes in the chromosome.

The TFM can be expressed as:

$$\text{TFM} = 100 - \sum(\text{gene fitness measure of all genes}), \quad (6)$$

$$\text{TFM} = 100 - \sum(\text{GFM_G1} + \text{GFM_G2} + \text{GFM_G3}), \quad (7)$$

where GFM_G1, GFM_G2, and GFM_G3 are the individual gene fitness measures. In this way, the TFM and GFM of a chromosome can be derived (Table 3).

Table 3
Derivation of the TFM and GFM of a chromosome

Input parameters	Value		
Initial population size	100		
Maximum number of generations to be iterated	100		
Crossover rate	80%		
Mutation rate	2%		
Number of bits assigned to each of the genes of the chromosome	8	6	7
Application requested gene values	226	121	10
Fitness points of gene	110	50	6

The gene fitness measure is inversely proportional to the total fitness measure of the chromosome.

5. Simulation

The simulation was made for three application requested values covering almost the entire range. The result is shown for three sets where the plot of total fitness measure, gene fitness measure, variation of each parameters, unoptimized and optimized regions of each parameters are shown.

Case 1 is based on application requested decimal values: bandwidth = 16, throughput = 4, spectral efficiency = 4, initial population = 100, and maximum generations = 100. The Figs. 1 and 2 show TFM and GFM of the adaptive GA.

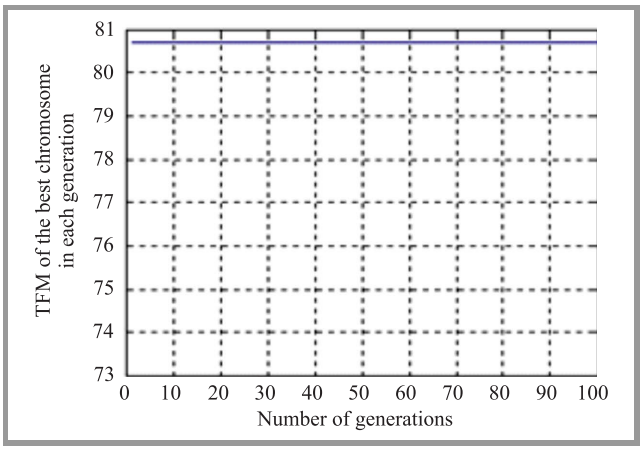


Fig. 1. Variation of TFM of the best chromosome with number of generation.

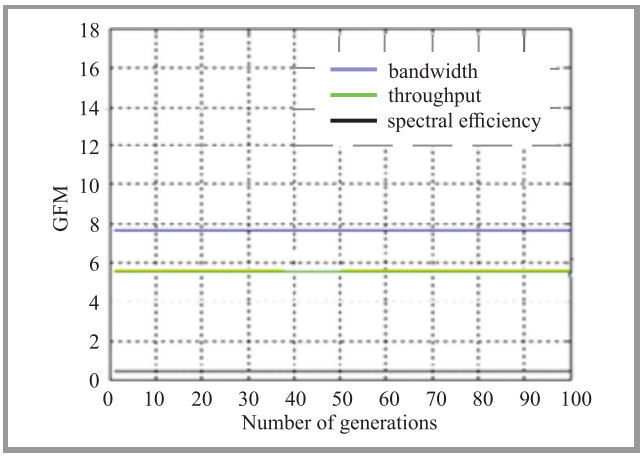


Fig. 2. Variation of GFM of Gene 1, Gene 2, and Gene 3 with number of generations. (See color pictures online at www.nit.eu/publications/journal-jtit)

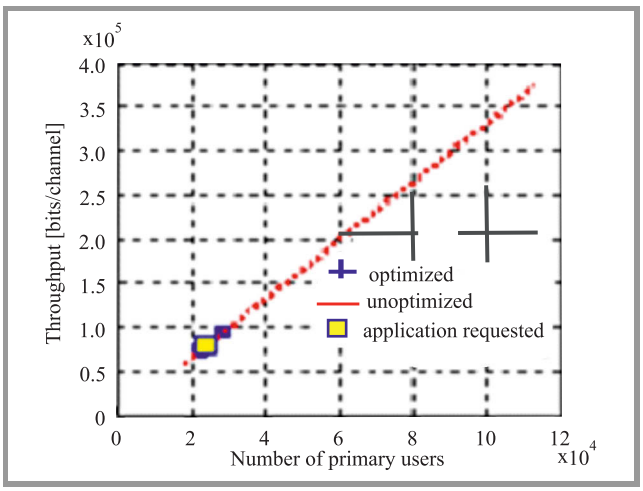


Fig. 3. Variation of throughput with number of primary users.

Figures 3 and 4 show the GA optimized plots of throughput and spectral efficiency for the application requested value considered. After the GA execution throughput is 79725.29 bits/channel for application requested decimal

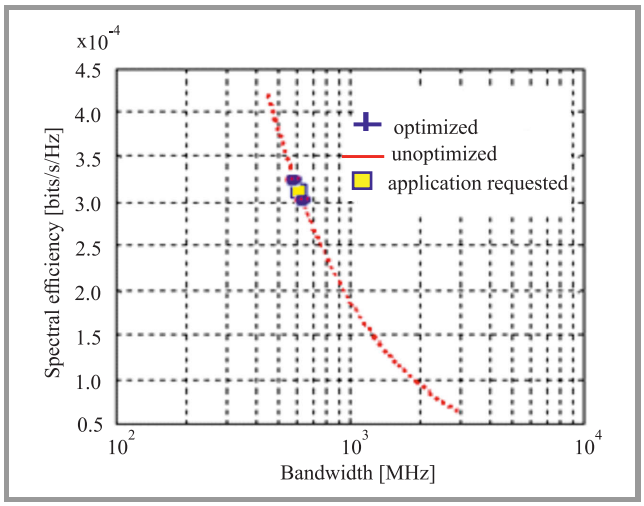


Fig. 4. Variation of spectral efficiency with bandwidth.

value 4. The exact value of the corresponding number of primary user is 24000, which is very close to the actual value 79726.28 bits/channel. Having application requested

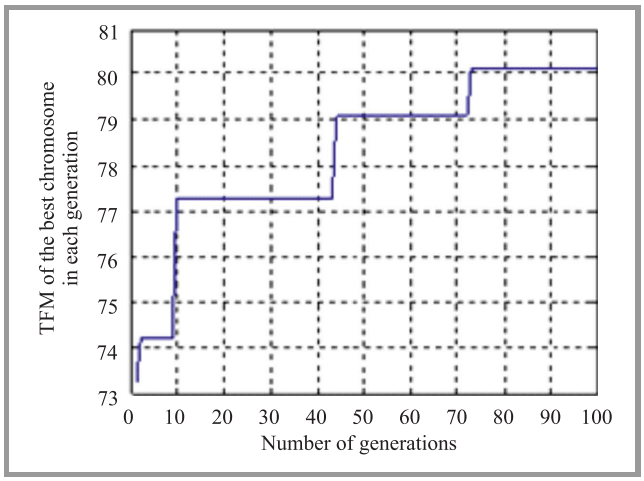


Fig. 5. Variation of TFM of the best chromosome with number of generations.

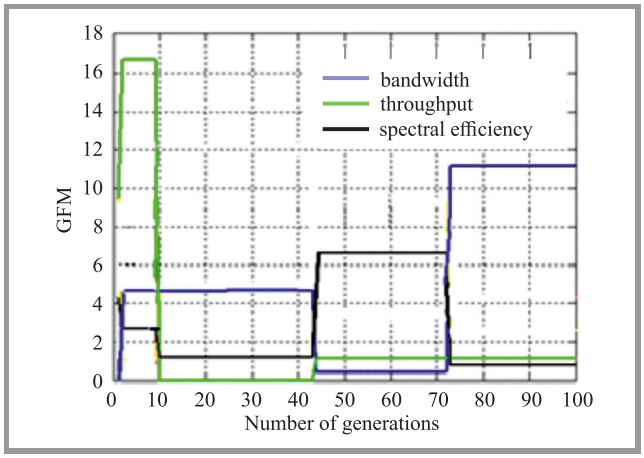


Fig. 6. Variation of GFM of Gene 1, Gene 2, and Gene 3 with number of generations.

decimal value 4 it can be rounded up 79726 bits/channel resulting spectral efficiency 0.000312 for application requested decimal value 4. The exact value of the corresponding bandwidth is 607.638 MHz, and can be rounded up to 608 MHz, which corresponding to application requested value of bandwidth 16.

In case 2 the used application requested decimal values are: bandwidth = 46, throughput = 12, spectral efficiency = 56, initial population = 100, maximum generations = 100.

Figures 5 and 6 show the TFM and GFM of the adaptive GA.

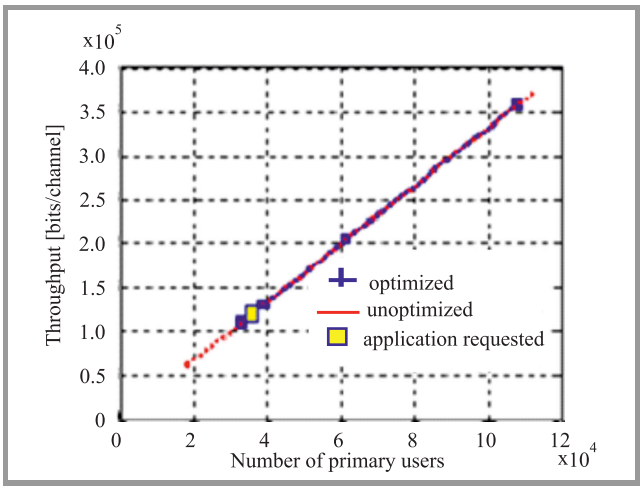


Fig. 7. Variation of throughput with number of primary users.

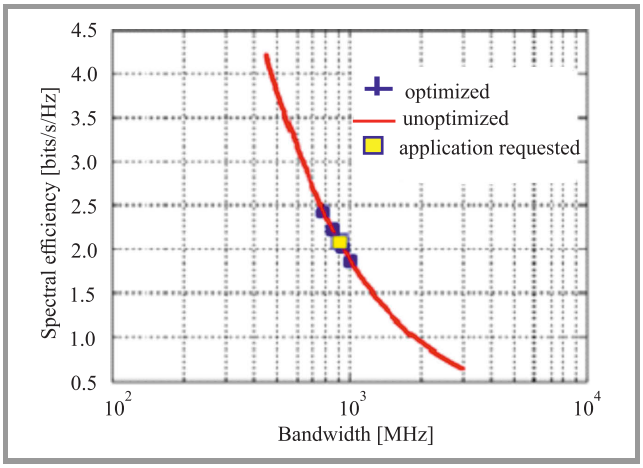


Fig. 8. Variation of spectral efficiency with bandwidth.

Figures 7 and 8 show the GA optimized plots of throughput and spectral efficiency for the application requested value considered. After the execution of GA. Throughput is 119589.41 bits/channel and spectral efficiency is 0.000208 for application requested decimal value 56. The exact value of the corresponding bandwidth is 911.45 MHz and very close to 910 MHz corresponding to application requested value of bandwidth 46. The deviation is around 1 MHz.

Case 3 uses the application requested decimal values: bandwidth = 72, throughput = 19, spectral efficiency = 79, initial population = 100, maximum generations = 100.
Figures 9 and 10 show the TFM and GFM of the adaptive GA for this simulation set.

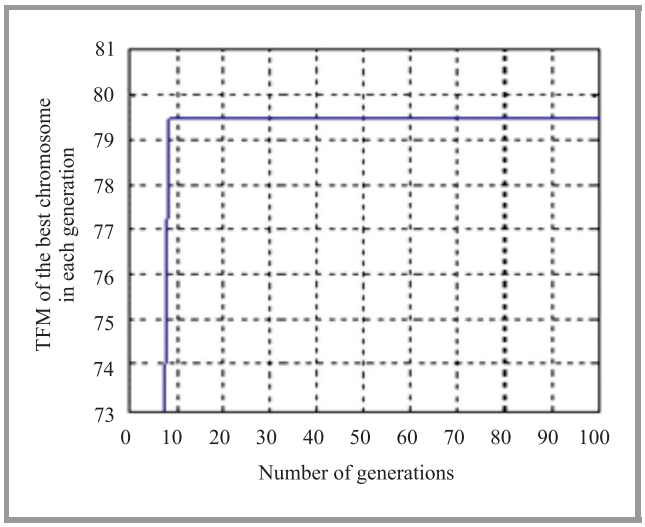


Fig. 9. Variation of TFM of the best chromosome with number of generations.

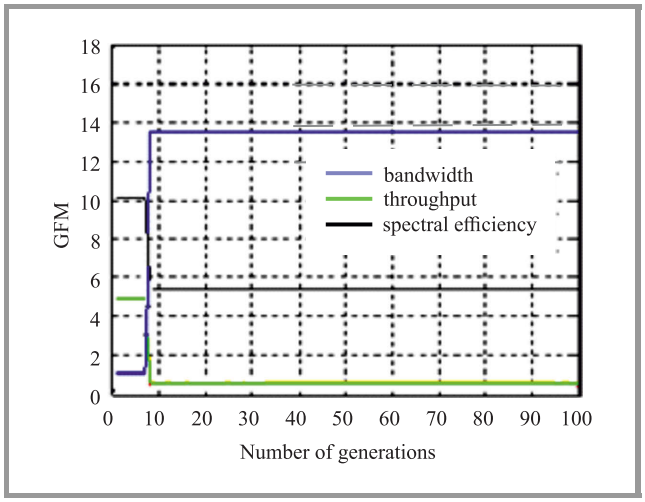


Fig. 10. Variation of GFM of Gene 1, Gene 2, and Gene 3 with number of generations.

Figures 11 and 12 show the GA optimized plots of throughput and spectral efficiency for the application requested value considered. After GA is executed, the spectral efficiency 0.0001637 was received for application requested decimal value 79 and the exact value of the corresponding bandwidth is 1170.26 MHz instead of the actual value of the bandwidth 1170 MHz, which corresponding to application requested value of bandwidth 72. The deviation is of 0.26 MHz.
In addition the throughput 154469.656 bits/channel was achieved whereas the exact value is 154469.66 bits/channel, which can be rounded to 154470 bits/channel and the

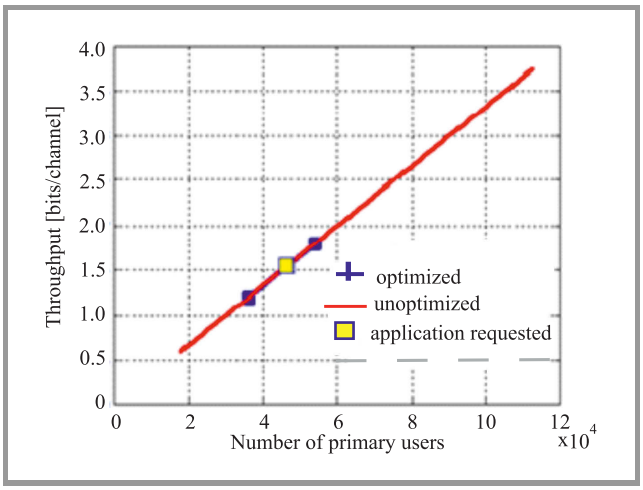


Fig. 11. Variation of throughput with number of primary users.

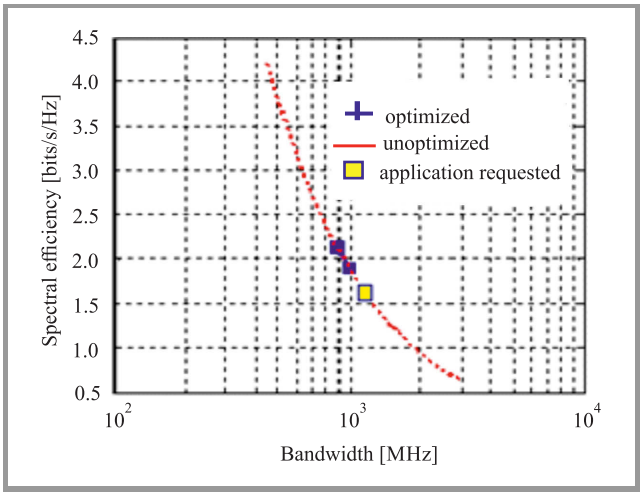


Fig. 12. Variation of spectral efficiency with bandwidth.

exact value of the corresponding number of primary user is 46500.
Comparing Figs. 1 to 12 for all the different sets of application requested decimal value and the discussion shows that maximum percentage deviation from the actual value is around 0.1. The deviations noticed are due to the randomness and can be minimized/eliminated by several executions techniques. The aim of the work is to propose a demand based GA by considering different regions of the spectrum and optimizing the sensing parameters considering the time varying nature of the spectrum and comparison theoretically calculated values with the values after execution of the GA.

6. Conclusion

The simulation results shows that the proposed method gives better real life solution to the cognitive system as time varying nature of spectrum hole is considered and having capability of adaption with the varying nature of spectrum holes.

References

[1] J. Elhachmi and Z. Guennoun, "Cognitive radio spectrum allocation using genetic algorithm", *EURASIP J. on Wirel. Commun. and Netw.*, vol. 2016, pp. 133–143, 2016 (doi: 10.1186/s13638-016-0620-6).

[2] S. Chatterjee, J. S. Roy, and P. P. Bhattacharya, "Spectrum sensing techniques for cognitive radio – a survey", *Int. J. of Appl. Engin. Res.*, vol. 10, no. 7, 2015, pp. 16665–16684.

[3] R. Deka, S. Chakraborty, and J. S. Roy, "Optimization of spectrum sensing in cognitive radio using genetic algorithm", *Facta Universit., Ser: Elec. Energ.*, vol. 25, no. 3, pp. 235–243, 2012 (doi: 10.2298/FUUEE1203235D).

[4] M. J. Kaur, M. Uddin, and H. K Verma, "Optimization of QOS parameters in cognitive radio using adaptive genetic algorithm" *Int. J. of Next-Gener. Netw.*, vol. 4, no. 2, pp. 1–15, 2012.

[5] J. Ramesh and A. Raman, "Optimization of sensing parameters using PSO, GA for cognitive radio", *Recent Trends in Sensor Res. & Technol.*, vol. 2, no. 3, pp. 23–30, 2015.

[6] P. T. A. Quang, S. R. Kim, and D. S. Kim, "A throughput-aware routing for distributed industrial cognitive radio sensor networks", in *Proc. 9th IEEE Int. Worksh. on Factory Commun. Syst. WFCS 2012*, Detmold, Germany, 2012, pp. 87–90.

[7] T. T. Le, and D.-S. Kim, "An efficient throughput improvement through bandwidth awareness in cognitive radio networks", *J. of Commun. & Netw.*, vol. 16, no. 2, pp. 146–154, 2014 (doi: 10.1109/JCN.2014.000025).

[8] S. Kaur and I. K. Aulakh, "Optimization of cognitive radio sensing techniques using genetic algorithm", *Int. J. of Innov. Res. in Comp. & Commun. Engin.*, vol. 3, no. 5, pp. 4131–4139, 2015 (doi: 10.15680/ijircce.2015.0305104).

[9] I. F. Akyildiz, W. Y. Lee, and K. R. Chowdhury, "Spectrum management in cognitive radio ad hoc networks", *IEEE Network*, vol. 23, no. 4, pp. 6–12, 2009.

[10] A. Marţian, C. Vlădeanu, I. Marcu, and I. Marghescu, "Evaluation of spectrum occupancy in an urban environment in a cognitive radio context", *Int. J. on Adv. in Telecommun.*, vol. 3, no. 3–4, pp. 172–181, 2010.

[11] M. Lopez-Benitez *et al.*, "Spectral occupation measurements and blind standard recognition sensor for cognitive radio networks", in *Proc. 4th Int. Conf. on Cognitive Radio Oriented Wireless Networks & Commun. CrownCom 2009*, Hannover, Germany, 2009, pp. 1–9.

[12] X. Meng, S. Wu, L. Kuang, D. Huang, and J. Lu, "Multi-user detection for spatial modulation via structured approximate message passing", *IEEE Commun. Lett.*, vol. 20, no. 8, pp. 1527–1530, 2016 (doi: 10.1109/LCOMM.2016.2577627).



Subhajit Chatterjee is an Assistant Professor in the Department of Electronics & Communication Engineering, University of Engineering & Management, Kolkata, West Bengal, India. His research interests include signal processing, wireless network and cognitive radio. Currently he is a registered Ph.D. scholar at KIIT University, Bhubaneswar, Odisha, India.

E-mail: chsubhajitch@gmail.com
Department of Electronics
and Communication Engineering
University of Engineering & Management
Kolkata, West Bengal, India



Swaham Dutta is working in the capacity of Assistant Manager in Big Data Analysis division, Relience Jio Infocomm Ltd., Mumbai, and Maharashtra, India. He has completed the B.Tech. degree from West Bengal University of Technology, Kolkata, India. His research interests are wireless sensor network, data processing and cognitive radio.

E-mail: swahamd@gmail.com
Reliance Jio Infocomm Ltd.
Big Data Analysis
Mumbai, Maharashtra, India



Partha Pratim Bhattacharya is working as Professor in Department of Electronics and Communication Engineering in the College of Engineering and Technology, Mody University of Science and Technology (formerly, Mody Institute of Technology and Science), Rajasthan, India. His present research interest includes mobile cellular communication, wireless sensor network and cognitive radio. He is a member of The Institution of Electronics and Telecommunication Engineers, India and The Institution of Engineers, India. He is working as reviewer in many reputed journals like IEEE Journal on Selected Areas in Communications, IET Communications, Springer's IEIB, Elesevier's Computer Communication, Elsevier's Journal of Network and Computer Applications, Adhoc and Sensor Wireless Networks, Annals of Telecommunications – Annales des Télécommunications, Elsevier's Physical Communication, Indian Journal of Science and Technology, etc.

E-mail: hereispartha@gmail.com
Department of Electronics and Communication
Engineering
College of Engineering and Technology
Mody University of Science and Technology
Lakshmangarh – 332311, Rajasthan, India



Jibendu Sekhar Roy is a Professor in the School of Electronics Engineering, KIIT University, Bhubaneswar, Odisha, India. From 1998 to 2009, he was lecturer, reader and professor in the ECE department of Birla Institute of Technology, Mesra, Ranchi, India. He has received Ph.D. degree from the Department of Electronics & Telecommunication Engineering, Jadavpur University, Calcutta, India in 1991. From 1991 to 1993, he was a post-doctoral research associate of CNRS, Govt. of France in IRCOM, University of Limoges, France. From 1994 to 1998, he was a research associate of CSIR in ETCE department, Jadavpur University, Calcutta. His research interest includes microwave and millimeter wave antennas for wireless communication, optimization of thinned array antenna, smart antennas, channel assignment & MIMO-OFDM algorithms, cognitive radio.

E-mail: drjsroy@rediffmail.com
School of Electronics Engineering
KIIT University
Bhubaneswar, Odisha
India

Paper

Design Exploration of AES Accelerators on FPGAs and GPUs

Vincenzo Conti[1] and Salvatore Vitabile[2]

[1] *Faculty of Engineering and Architecture, University of Enna Kore, Enna, Italy*
[2] *Department of Biopathology and Medical Biotechnologies, University of Palermo, Palermo, Italy*

Abstract—The embedded systems are increasingly becoming a key technological component of all kinds of complex technical systems and an exhaustive analysis of the state of the art of all current performance with respect to architectures, design methodologies, test and applications could be very interesting. The Advanced Encryption Standard (AES), based on the well-known algorithm Rijndael, is designed to be easily implemented in hardware and software platforms. General purpose computing on graphics processing unit (GPGPU) is an alternative to reconfigurable accelerators based on FPGA devices. This paper presents a direct comparison between FPGA and GPU used as accelerators for the AES cipher. The results achieved on both platforms and their analysis has been compared to several others in order to establish which device is best at playing the role of hardware accelerator by each solution showing interesting considerations in terms of throughput, speedup factor, and resource usage. This analysis suggests that, while hardware design on FPGA remains the natural choice for consumer-product design, GPUs are nowadays the preferable choice for PC based accelerators, especially when the processing routines are highly parallelizable.

Keywords—AES, accelerators, FPGA prototyping, GPGPU, OpenCL.

1. Introduction

In the last decade the complexity of the architecture of graphical processing units has grown exponentially, pushing them outside the world of the dedicated processors to embrace the general-purpose applications field.

Recently Graphic Processing Unit (GPU) manufacturers have focused their attention not only on typical graphical processing tasks, equipping their products with characteristics explicitly aimed to the general purpose computing (IEEE-754 compliance floating point units is just an example). Nevertheless, the massive parallel design, which is a key feature for a GPU architecture, is an attractive property in many number crunching applications.

With the introduction of the Nvidia Fermi architecture [1] the interest in GPGPU has grown, because of its ambitious goal: for the first time, a GPU architecture was expressly designed to allow general-purpose computations. Even before the Fermi architecture, with the introduction of technologies such as CUDA, Stream and OpenCL the word GPGPU has assumed a new meaning. Before these frameworks, the only way to access the GPU processing power for general computing was to use shaders, by resorting to a cumbersome process in which data to process was encoded in textures pixels with many piratical limitations. However, many of this proof of concept showed the true potential of GPU devices. Subsequently GPU devices were used as accelerators for many scientific applications, ranging from image processing to Basic Linear Algebra Subprograms (BLAS), with successful results.

At the same time, FPGA devices have been traditionally used for various and different purposes, thanks to the very high degree of customization available to the designer. With more details this technology has been used to implement video processing [2] and [3], biometric recognition systems [4] and [5], mathematical and/or biological coprocessors [6] and [7], security access management [8], [9] and [10], and so on.

The difference in terms of overall costs, development time and background knowledge required to target both platforms justifies the interest by the scientific community in a full comparison. To make this comparison effective, an algorithm that can be easily implemented in both hardware and software platforms is needed. Rijndael algorithm is a good candidate for this purpose as it was designed keeping an eye on both platforms.

In this paper, two implementations of the AES encryption cipher in counter (CTR) mode are presented: a novel FPGA design for the Celoxica RC1000 board, developed with Agility's Handel-C compiler, and parallel OpenCL software which runs on GPU. GPGPU is an alternative to reconfigurable accelerators based on FPGA devices. The FPGA implementation consists of four AES cores, each of which performs a single AES encryption in 0.48 μs with 70 MHz clock, delivering a throughput of about 1036 Mb/s. The OpenCL software is a simple port of an ANSI C implementation of the Rijndael algorithm. The two solutions exhibit good performance compared to a general-purpose CPU implementation, thus are both suitable to be used as accelerators. In addition, the architectural constraints, power consumption, speedup factors, overall costs of the two projects and their analysis has been compared to several others in order to establish which device is best at playing the role of hardware accelerator by each solution showing interesting considerations in terms of throughput, speedup factor, and resource usage. This analysis suggests that, while hardware design on FPGA remains the natural choice for consumer-product design, GPUs are nowadays the preferable choice for PC based accelera-

tors, especially when the processing routines are highly parallelizable.

The paper is structured as follows. Section 2 presents a review of other works available in literature in which the two technologies are compared. In Section 3 the Rijndael algorithm is briefly described, together with the implemented CTR mode of operation. Sections 4 and 5 illustrate the FPGA and OpenCL proposed implementations respectively. Section 6 describes the testing environment while in Section 7 the results are extensively analyzed and commented. Section 8 presents an overview of similar works with a comment on the performance achieved by the proposed solutions. Finally, Section 9 contains the conclusions of this work.

2. Related Works

There is a variety of publications in literature that compare FPGA and GPGPU implementations and the results may vary depending on the platforms used.

In 2005 Cope *et al*. [11] pointed out the limitations of GPU based solutions compared to FPGA devices due to the low memory bandwidth. In the same year, Mali *et al*. [12] showed an implementation of AES on FPGA, using the same platform used for this paper (the Celoxica RC1000 board). The proposed solution is c.a. 5.7 times faster running at a lower clock frequency.

Lately, in 2007 another interesting comparison was made by Baker *et al*. [13]. In their work, they implemented a matched filter on both FPGA and GPU devices, obtaining similar throughput. Moreover, when comparing throughput against costs, they show how GPU solutions are the cheapest.

Costs involved in targeting FPGAs and GPUs have been analyzed by Shuai Che *et al*. in [14] comparing the two solutions in three different tasks: Gaussian elimination, Needleman-Wunsch and DES.

However, the answer to the question "Have GPUs made FPGAs redundant as accelerator devices?" is still open. Contrasting results were shown depending on many factors, including the algorithm implemented, the targeted devices and the programming frameworks.

For example, in [15] the performance of common image processing algorithm implemented in FPGA and GPU are compared. The FPGA implementation outperforms the GPU, especially in those algorithms were a careful memory access policy is necessary to synchronize the GPU threads.

Different results are shown in [16] where an implementation of common SPICE routines is presented giving similar results in both hardware and software approaches. Even if FPGA can outperform small factor devices, when compared to most powerful GPU they suffer for the limited resources on board and the poor scalability.

Depending on the application, the results may be even more different. In [17] sparse matrix vector multiplication implemented on GPU outperforms the FPGA counterpart, although the authors point out that their FPGA solution is highly penalized by a very poor memory bandwidth.

Finally, in [18] a SEAL encryption implementation is presented in both FPGA and GPU. Both platforms achieve the same overall performance. In this paper, the implementation of an encryption algorithm is also discussed, but it is worth to note that AES is slightly more complex than SEAL and so it better exploit the differences between the two processing platforms.

3. The AES Standard

AES is the standard currently recommended by NIST for symmetric block cipher encryption. The actual standard publication [19], issued in November 2001, includes a detailed description of the Rijndael algorithm, which was chosen among others like MARS, RC6, Serpent and Twfish, because of its high degree of cryptographic security and its simplicity. The Rijndael selection process was carried through openly and with the full support of the scientific community. This has gained to AES the interest of many operators in the cryptographic security field and made the transition to the new standard very quick.

3.1. The Rijndael Algorithm

Rijndael is a symmetric block cipher algorithm, which runs a certain number of rounds on every input block. In Fig. 1 the algorithm structure is shown. Its design, which is totally different with respect to the previous standard DES, is very far from the traditional Feistel cipher structure. The Rijndael cipher applies Galois's Finite Field arithmetic to match the confusion and diffusion requirements and it is composed of two distinct procedures for encryption and decryption. The input blocks size is of 128 bits while the key can be 128, 192 or 256 bit wide, depending on the security degree required. The key size is also related to the number or rounds of the encryption/decryption procedures as shown in Table 1.

Table 1
Rijndael key size and rounds number

Key size [bit]	Rounds number
128	10
192	12
256	14

3.2. The AES Round Structure

Rijndael iterates the same sequence of operators, named *round*, on every input block. The plaintext is split in chunks of 16 bytes and each of these is treated as a 4×4 matrix called the *state* vector.

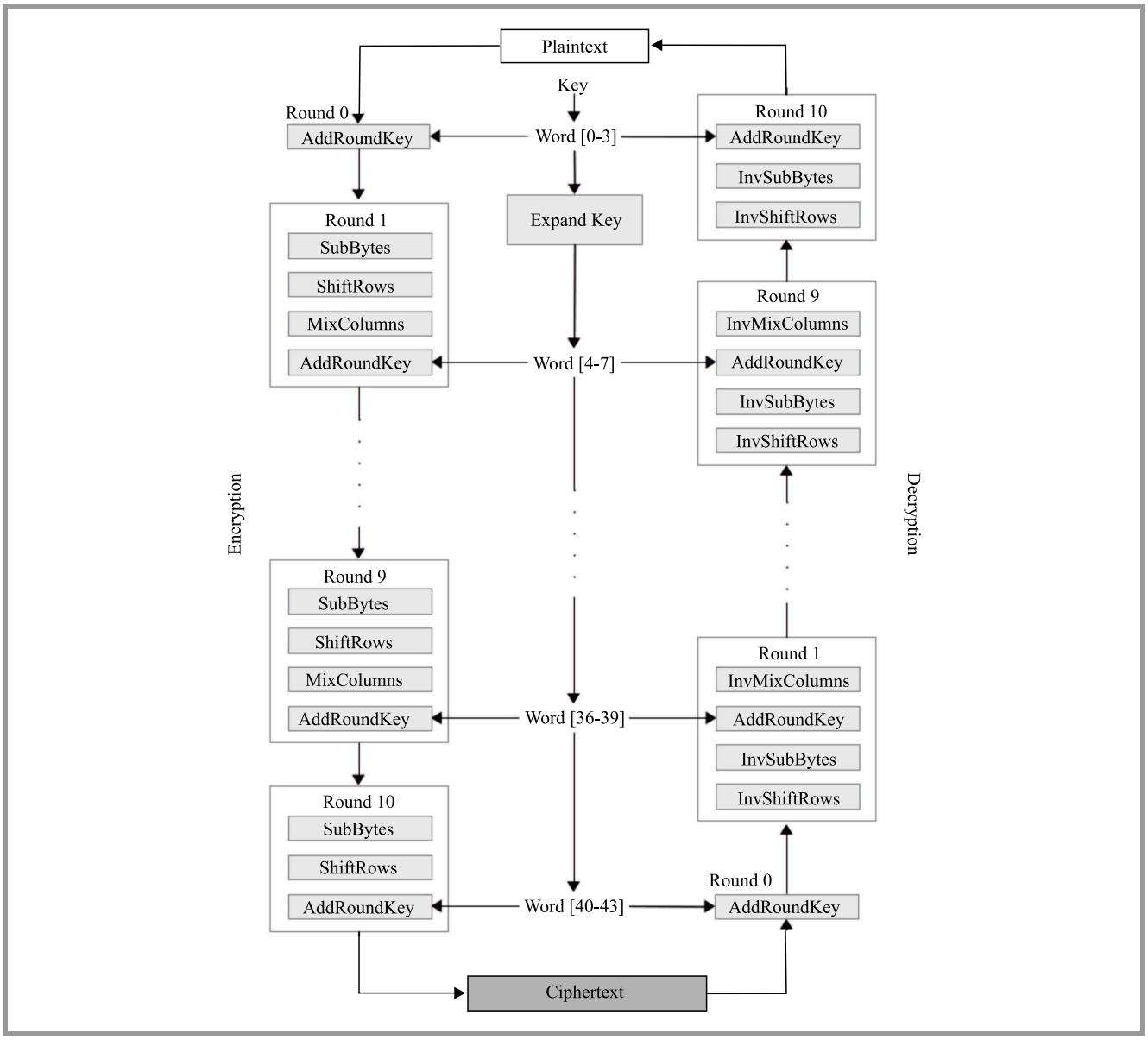


Fig. 1. The Rijndael algorithm.

The four operators *SubBytes*, *ShiftRows*, *MixColumns* and *AddRoundKey* are used in every round but the first and the last, which are defined differently.

The SubBytes function uses a substitution box, named *Sbox*, to map every byte in the state vector on a proper 8 bit value. The mapping output is obtained with the following affine transformation applied to the multiplicative inverse $x_7x_6\ldots x_0$ in the $GF(2^8)$ of the input byte:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} \oplus \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}.$$

The 0x00 value, whose multiplicative inverse is not defined in $GF(2^8)$, is simply mapped to the 0x63 byte. The Sbox is usually stored in memory and accessed like a look-up table to speed up the substitution function.

The ShiftRows function consists of a circular left shift of 1, 2 and 3 positions for the rows 2, 3 and 4 respectively of the state vector. The first row remains unchanged.

The MixColumn function consists of a linear transformation which is applied to the elements of each column:

$$\begin{bmatrix} \hat{s}_{c0} \\ \hat{s}_{c1} \\ \hat{s}_{c2} \\ \hat{s}_{c3} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \times \begin{bmatrix} s_{c0} \\ s_{c1} \\ s_{c2} \\ s_{c3} \end{bmatrix}.$$

The c subscript is the column index. The multiplication and the add operators used in the matrix product are those defined in $GF(2^8)$.

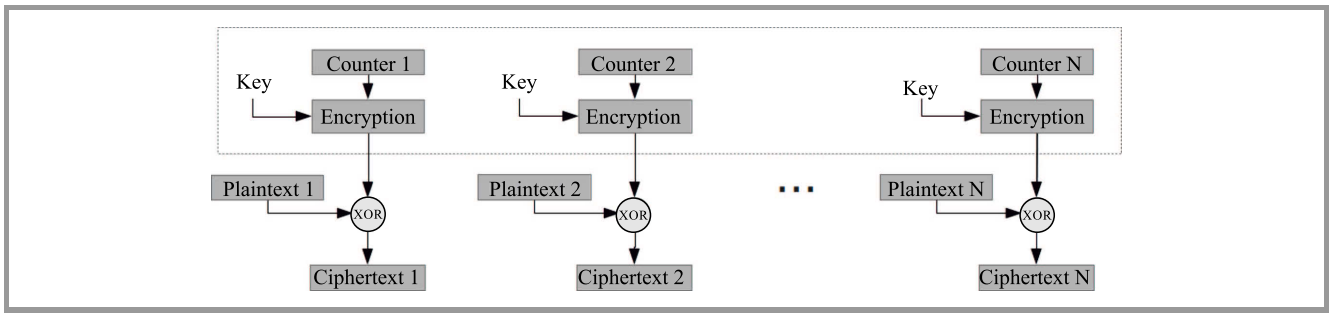


Fig. 2. CTR mode encryption.

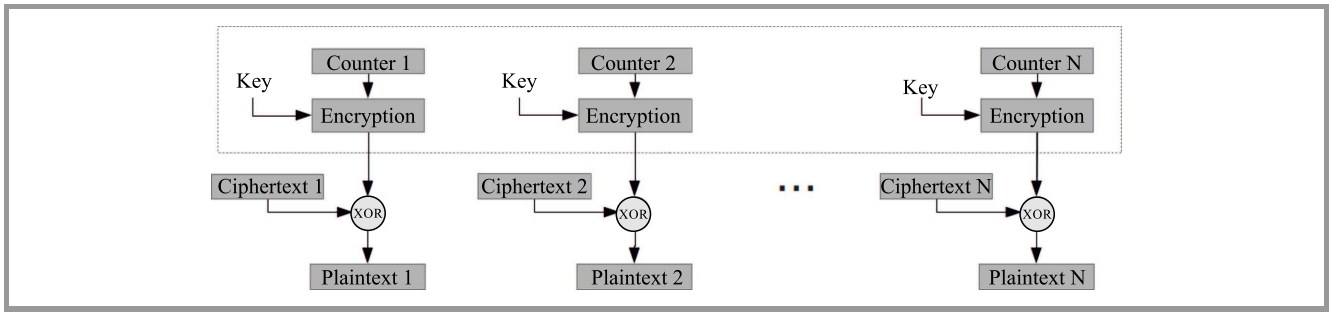


Fig. 3. CTR mode decryption.

The AddRoundKey function is the only operator, which involves the secret key. A distinct 128 bit subkey for each round is extracted from the key and is XOR-ed with the state vector. The key scheduling procedure is also described in [19].

3.2.1. Counter Mode

The design presented in this work uses the counter (CTR) mode of operation [20] because it allows the parallel execution of the cipher on each block while ensuring a strong degree of resistance to cryptanalysis.
Another interesting feature of the CTR mode consists in the use of the same encryption procedure for both encryption and decryption. This comes very useful for the AES cipher, which would normally require two distinct implementation for the encryption/decryption routines.
Looking at the Fig. 2 it is easy to note that the data being codified by the cipher is a special value, named *counter*, which is XOR-ed with each block, and is different for every block (e.g. incremented by 1 for each block encryption).
The same operation has to be performed in decryption: the reversibility of the cipher actually resides on the use of the XOR operator (see Fig. 3).
The seed value for the counter can also be kept secret to increase the overall degree of security of the AES cipher with respect to brute-force attack.

4. AES Processor Design

The proposed design is an implementation of the 8 bit oriented version of AES. Each round operation takes a single clock cycle, except the SubBytes and ShiftRows operation that were mixed together. Some of the suggestions shown in [21] where used to reduce area occupation and maximum delay path without compromising the throughput. This lead to a total of 33 clock cycles required to perform a single AES encryption. A summary of the characteristics of this design is shown in Table 2, while the overall architecture is shown in Fig. 4.

Table 2
Proposed AES processor summary

Core operating frequency	70 MHz
Memory operating frequency	33 MHz
Average throughput	1036 Mb/s
Occupation	18048 slices
Maximum delay path	13.92 ns

In next subsection a detailed description of the proposed architecture is discussed. Parallel and pipelined processing has been used to achieve high throughput performance.

4.1. Overall Architecture

A first level of parallelization is easily achieved by instantiating multiple AES cores on the chip. The memory interface of the Celoxica RC1000 board allows parallel access of the 4 memory banks. So in the proposed design four independent AES blocks are capable of running with full parallelism, achieving an overall performance of 4 times the single AES core, scoring a little more than 1 Gb/s.

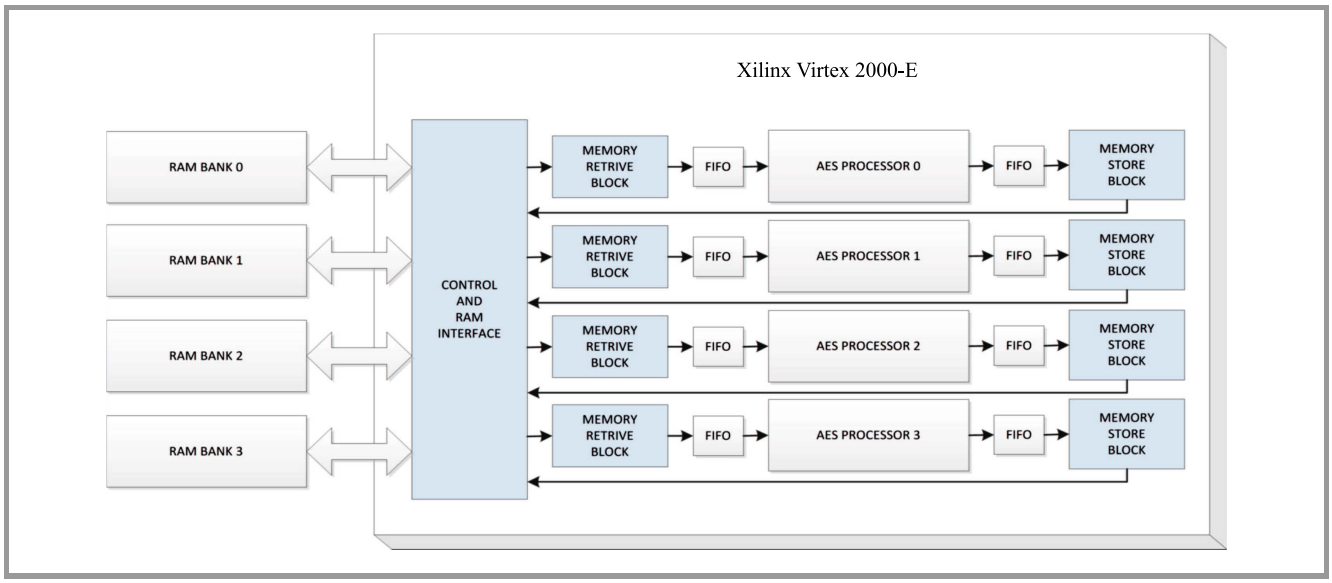


Fig. 4. Overall architecture implementing AES processor using Xilinx Virtex 2000-E FPGA.

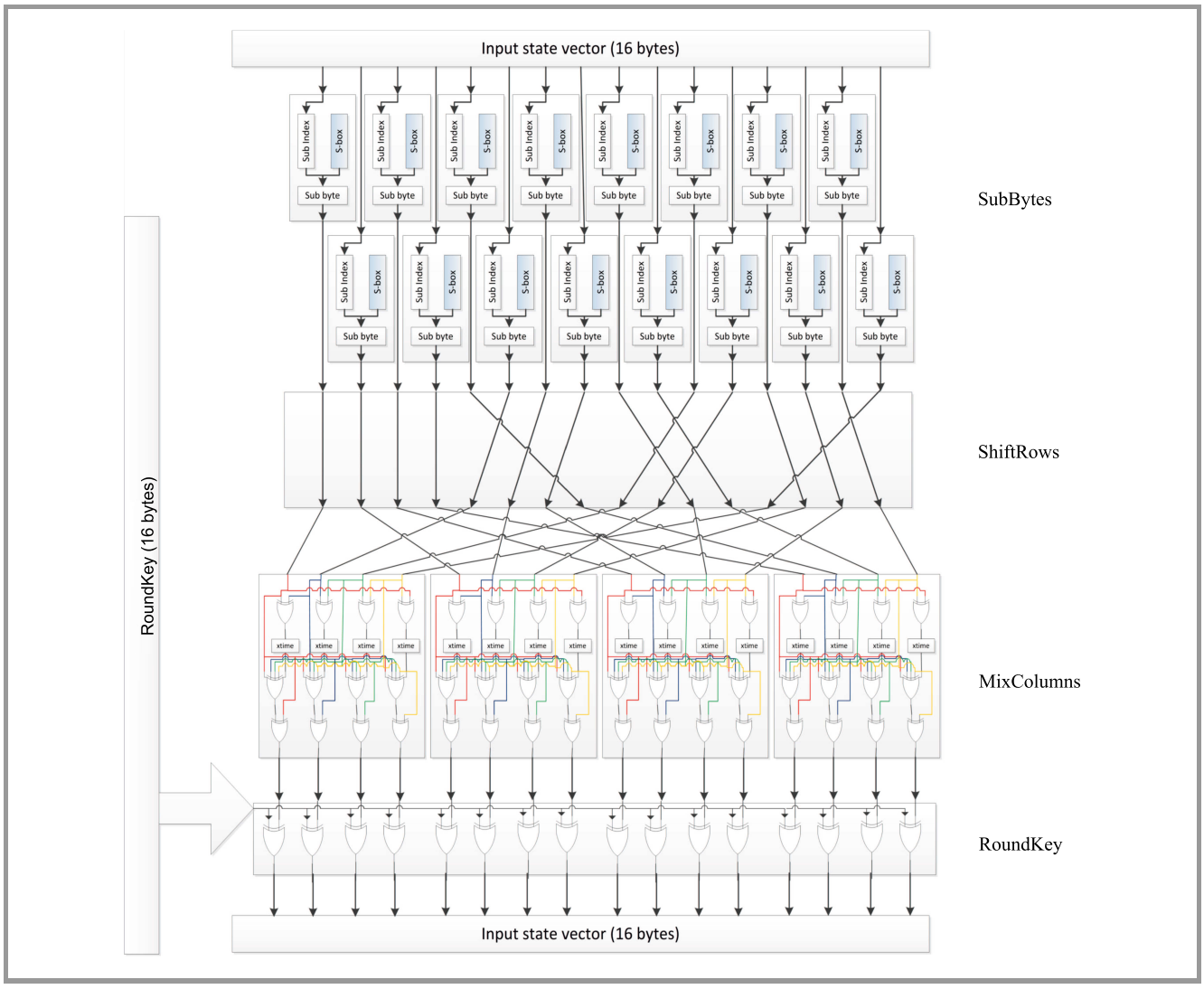


Fig. 5. AES round architecture.

4.2. AES Core Architecture

One AES core contains the circuitry required to perform AES 128 bit encryption. Table 3 shows the performance of the proposed AES core. Figure 5 shows the architecture of a single round circuit. The full round operation takes 3 clock cycles. To allow full parallelism to the SubBytes operation, 16 S-boxes have been instantiated in the ROM memory. Allocating registers array in Handel-C is very resource consuming compared to the usage of ROM bits, but obviously, the same ROM cannot be accessed simultaneously by multiple circuits. This led to the choice of allocating multiple S-boxes. Even though this choice sacrifices more area, the high advantage in the overall performance is a good compromise. Each AES core is implemented in 3360 slices (c.a. 17.5% of the total available on chip).

Table 3
Proposed AES block summary

Total latency	0.48 μs
Operating frequency	70 MHz
Average throughput	259 Mb/s
Occupation	3360 slices
Maximum delay path	13.92 ns

4.3. Pipelined Design

Unfortunately, the Celoxica RC1000 has very high latency memory, which cannot be accessed at frequencies higher than circa 33 MHz [22], [23]. The proposed AES circuit has a maximum delay path of 13.92 ns, so it can theoretically reach up to 71 MHz. To reduce the penalization introduced by the very poor memory interface, a double domain clock design was used. One domain clock, running at 33 MHz, contains the circuitry for data fetching and write back, while the other, running at 70 MHz, contains the 4 AES cores. The communication between the two clock domains is ensured by eight 128 bit channels, each of which equipped with a FIFO queue. The data-fetching block and the write back block are running in a parallel fashion. The synchronization between these two blocks is guaranteed by 4 semaphores. With this solution, memory fetches can happen while encrypting previously fetched blocks, increasing the overall performance (see Fig. 6). The total time required to encrypt 8 MB is nearly the same required to simply access the data to the on board RAM.

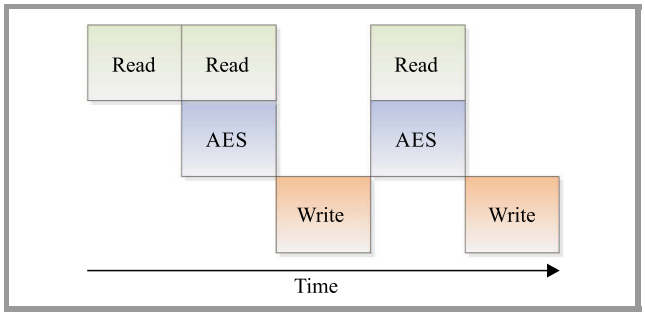


Fig. 6. Pipelined execution.

5. OpenCL Implementation

The GPU version has been implemented using OpenCL rather than similar but proprietary technologies for its portability. The results obtained by running the same implementation on different platforms (the Nvidia GT520 and GT555M and the Intel Core i7 processor) are reported in the following subsections.

5.1. The Threading Model

AES in CTR mode is perfectly suitable for parallel applications. As previously discussed in Subsection 3.2.1, in CTR mode every block encryption is independent, and thus there is no need to implement ad-hoc thread synchronization polices. Therefore, the adopted threading model can be simply summarized as follows:

- a grid is defined with only one work-group, and a single kernel running the Rijndael algorithm;
- in the work-group, each 128 bit data block is mapped into a single work-item. Thus, the number of work-items will be equal to the number of 128 bit data block in our stream;
- parallel execution of the work-items. The counter to use in CTR mode is calculated from the thread ID, as the threads are mapped 1:1 to the data blocks.

5.2. Targeting the GPU on a Consumer Grade PC

Special care need to be taken when working with consumer grade computers, as most probably the GPU used as accelerator will be the only one available to the system, and so it will be shared by several concurrent tasks, such as: desktop environment running in background updating the screen content, any application using 3D capabilities, accelerated video playback, etc. Therefore, it is important to understand that a single OpenCL program cannot lock the GPU for an undefined time. On some platforms, this may be a strict requirement. In the Microsoft Windows environment, for example, the video driver is automatically reset if the GPU doesn't respond to the OS commands within a predefined timeout (usually just a couple of seconds). A bad designed OpenCL program could never terminate correctly. The solution used in this work is simple but effective: the input data is divided into chunks that the GPU can process without hogging the system. The size of the chunk is a critical point of choice: a small size will cause an underutilization of the GPU processing power, while a large size can cause system hogging. In conducted experiments, the input chunk was set to 8 MB, as this size showed the best compromise between the utilization of the device and the

overall performance. Moreover, 8 MB is exactly the same size used for the FPGA implementation, as it is the total amount of on board RAM memory. Using the same size increase the accuracy of our measurements as the overhead introduced to divide the data in multiple chunks is the same regardless of the processing platform being tested.

5.3. Practical Aspects

Another great advantage offered by OpenCL is the compatibility of the C99 specification [24]. With a few adjustments, our C code developed for the CPU was ported successfully to the GPU. In particular, only some decoration was added to the function prototypes to correctly address the various memory spaces available in OpenCL. A sensible increase in performance over the standard CPU implementation required less than a person-day work. OpenCL programs are compiled on the fly at run time, so the compatibility with different platforms is guaranteed by the underlying software layer. This may contrast with the possibility to optimize the code for a particular device or architecture. In this case, multiple versions of the same OpenCL software can be developed and then selected at run time depending on the running platform. As an example, consider how an OpenCL program accesses the global memory. Since the memory hierarchy may vary from architecture to architecture, different ways of implementing global memory access were examined. In particular, to ensure the maximum performance the data alignment of the write back operation matched the alignment of the running device.

6. Testing Environment

Each implementation was initially tested using the AES standard test vectors recommended in [19]. A software library named *FastAESlib* was then developed to create a common interface for accessing each processing platform (FPGAs, GPUs and CPUs) addressed in this work. It can perform several tasks, as summarized below:

- enumerate at run-time the processing platforms available in the system (FPGAs, GPUs and CPUs),
- offload the processing task to any of the available processing platforms,
- setup platform specific parameters (e.g. the working frequency of the FPGA),
- report the progress of the current task,
- measure the overall execution time (using the OS high resolution timers),
- measure the processing execution time reported from the devices (on board timers for FPGAs and OpenCL event timers for GPUs).

This library was then used to develop three software applications designed to test the various platforms on different scenarios. The first one is an image encipher/decipher, which processes uncompressed image data. Figures 7 and 8 show the user interface of this application. The user can set all the processing parameters exposed by the FastAESlib library and obtain on screen the performance counters measurements (both execution time and throughput). As shown in Fig. 8, after the encryption phase the image data is completely scrambled, without exposing neither the chromatic information nor the original image structure. This visually proves how powerful is the CTR mode compared to other standard modes of operation. As a proof of concept, another software application named *FileCrypter* was developed to test implementations with large files. This application can encipher/decipher a file with a single password.

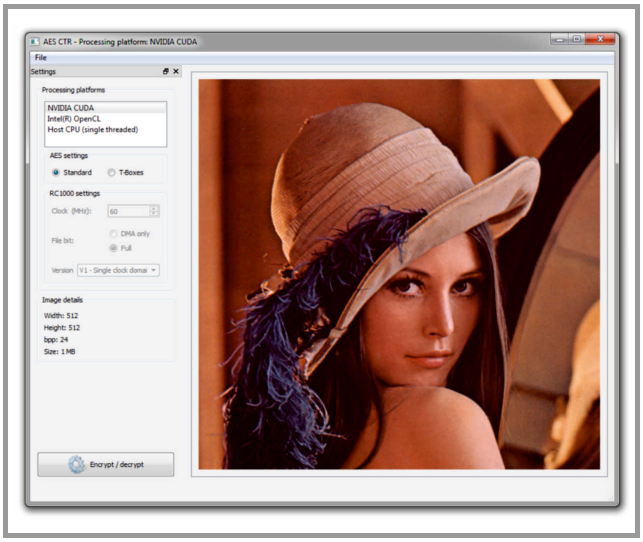


Fig. 7. Screenshot of the software ImageCrypt. (See color pictures online at www.nit.eu/publications/ journal-jtit)

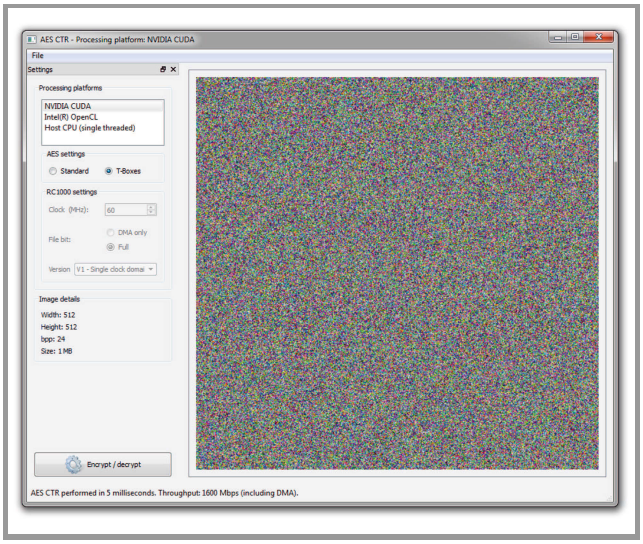


Fig. 8. Screenshot of the software ImageCrypt after encryption.

Lastly, a scripted application was developed to benchmark the various implementations discussed in this work.

This software utility performs AES encryption/decryption a specified number of times (in our tests 20 times) and calculates the average execution time and throughput. Moreover, when using the FPGA based processing platform, it can repeat the testing sequence at different clock frequencies, verifying the correctness of the result at each iteration. The results obtained using this tool is discussed in Section 6.

7. Experimental Results

The presented implementations show interesting results compared against a standard CPU. Table 4 shows the overall performance of the target systems including memory transfers time. When considering only the data rate, the fastest solution appears to be the OpenCL based implementation. However, it is important to consider the differences in the following three areas:

- the memory bandwidth can have a significant impact on the overall performance,
- the throughput should be normalized considering the different working frequencies,
- the various devices have a very different power consumption levels.

Table 4
Overall performance of the target platforms

Platform	Clock [MHz]	Rate [Mb/s]	Rate/clock ratio
FPGA	70	198	2.828
Nvidia GT 520	1620	520	0.321
Nvidia GT 555M	1180	1280	1.084
Intel Pentium 4	2000	42	0.0210
Intel Core i7	2500	81	0.0324

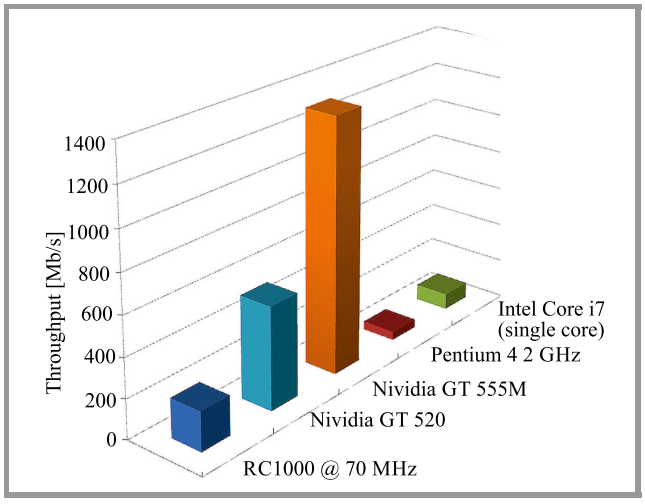


Fig. 9. Overall performance of the target platforms.

In Table 4 and Fig. 9 the overall performance measurements, but normalized with respect to the clock frequency, are shown. It is clear that the FPGA based solution can achieve better performance at lower clock rates, but it is worth to note that the GPU based solution exhibit a similar throughput/clock ratio, while the values reached by general purpose CPUs are two orders of magnitude lower. Interesting results are obtained when filtering out the time consumed by memory transfers (from the central memory to the on board memory). Table 5 and Fig. 10 show the processing throughput. This shows how the memory latency negatively affects the throughput of the RC1000 board, while the GPU based solutions are only lightly affected by the DMA operation. This is a logical consequence of the different technologies used by the two devices. Table 6 highlights the main differences.

Table 5
Performance of the target platforms without DMA time

Platform	Clock [MHz]	Rate [Mb/s]	Rate/clock ratio
FPGA	70	1036	14.8
Nvidia GT 520	1620	548	0.338
Nvidia GT 555M	1180	1440	1.22

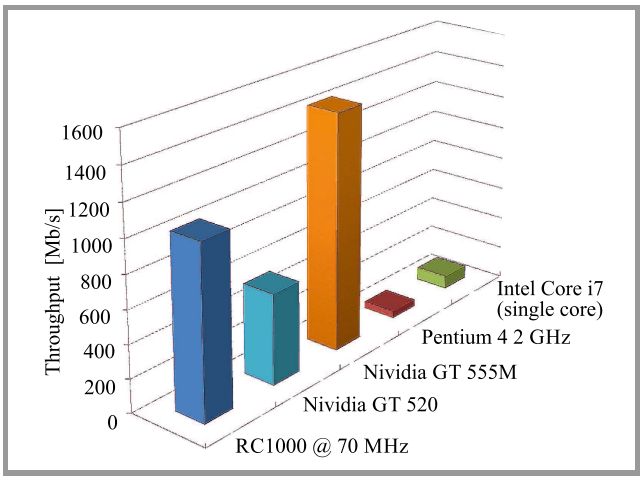


Fig. 10. Overall performance of the target platforms without DMA time.

Table 6
RAM memory comparison

Property	FPGA	GT 520	GT 555
Latency [ns]	25	10	10
Bus width [bit]	32	64	192
Clock [MHz]	33	900	900
Technology	SRAM	DDR3	SDDR3

The normalized throughput/clock ratio without DMA time shows how powerful the FPGA implementation is (see Table 5). When artificially scaling the three platforms'

clocks to the same frequency, the FPGA is 12 times faster of the fastest GPU based solution. Other interesting considerations can be made about the power consumption. While FPGAs are low power devices, GPUs are generally power-demanding processors. However, compared to a general purpose CPU, both the FPGA and the GPU platforms are the most energy efficient. An important aspect worth to note is the development cost. Regardless of the hardware cost, where a substantial difference exists between FPGAs and GPUs, another major disequilibrium can be found in the Time To Market (TTM) parameter. Even if TTM is low for FPGAs, designing hardware is generally a more time consuming task when compared to software development. Lastly, another key advantage of GPGPU technologies is the portability of the code. The same code can be executed on different OpenCL compliant devices without adjustments exploiting their potentials. FPGA designs need careful handling when ported from one device to another, making the porting operation hard and the previously developed code less reusable.

8. Discussion and Comparison

This section is devoted to the analysis of several other AES implementations on both GPU and FPGA devices. The direct experience of the implementation described in the previous sections is the starting point of our analysis, but first comes a little digression on the parameters that will be considered as terms of comparison. A comparison based on throughput vs. clock rate would give no useful results when comparing such different architectures. A targeted approach is needed to analyze each one's peculiarities before a direct comparison can be evaluated. FPGAs throughput will be analyzed against resources usage while GPUs' total number of stream processors will be considered as the main trade-off factor. When comparing the performance of such different devices it is important to investigate the different approaches available to the designers. For instance,

Table 7
Comparison of discussed FPGA implementations

Paper (characteristic)	Slices	Clock [MHz]	Throughput [Gb/s]
Rodriguez *et al.* [26] (pipelined)	5677	34.2	4.21
Mali *et al.* [12]	–	74	0.18
Kotturi *et al.* [27] (pipelined)	5408	232.6	29.77
Sivakumar *et al.* [28] (AES-CTR)	6766 CLB	194	2.257
Singh e*t al.* [29] (pipelined)	6352	347.6	44.2
Hoang *et al.* [25]	895	–	1.03
The proposed system (AES-CTR)	3360	70	0.25

Table 8
Comparison of discussed GPU implementations

Paper (characteristic)	GPU	Clock [MHz]	Throughput [Gb/s]
Manavski *et al.* [30] (CUDA)	128	575	8.2
Wang *et al.* [31] (OpenCL)	240	1476	1.05
Wang *et al.* [31] (CUDA)	240	1476	1.2
Keisuke *et al.* [32] (CUDA)	240	1476	32.5
The proposed system (OpenCL)	144	1180	1.25

AES can be implemented with or without look-up tables (T-boxes). Moreover when targeting hardware, pipelining is a natural choice against task parallelism, which is the foundation of the GPGPU computing model. In what follows, different FPGAs designs for AES are analyzed first. Next, parallel implementation of AES on GPU is examined. Table 7 shows a summary of the results of several AES implementations on FPGAs. In [12], Mali *et al.* presented a AES processor design in Handel-C on the same FPGA device used in this paper. The clock rate is slightly different, but the maximum throughput achieved from the solution proposed in this paper is higher. This may be due to the Handel-C compiler, which is very sensitive to the instruction order, and the control flow structures used. As another example of the impact of the Handel-C designing process on the result, consider that the AES processor design proposed in this paper requires 48 clock cycles to complete one 128-bit block encryption. Hoang *et al.* [25] proposed a VHDL design that completes 128-bit block encryption in 13 clock cycles requiring a lower number of slices, and therefore can potentially achieve higher throughput at the same clock speed. As previously mentioned, another important point is the processor design. The highest throughputs reported in Table 7 are relative to fully pipelined implementation of AES ([26], [27] and [29]). In this case, it is interesting to notice that, while the slices usage is slightly varying, the throughput/clock ratio is almost the same for each of these implementations. This observation leads to the conclusion that the performance of an optimal AES processor design for FPGA scales almost linearly with the clock rate given a fixed slices usage. Table 8 shows the results achieved by several AES parallel implementations running on GPU. The OpenCL implementation proposed in this paper was made out of an ANSI C implementation of the AES encryption routine. Therefore, no particular code optimization technique was adopted. Several test runs on the same GPU device showed heavy performance variations with different number of executing threads. In general, particular care must be taken in order to achieve optimal performance on GPU. As an example,

[31] and [32] report dramatically different throughputs on the same GPU, but is worth to mention that Wang *et al.* are relative to the XTS mode of operation of AES, which implies that some additional operations are executed within the Rijndael encryption procedure. Another critical parameter is, of course, algorithm design. Manavski [30] achieves a significantly higher throughput on a graphics processor equipped with less cores and a lower clock speed that the present work. As a side note, when directly compared, the solutions based on CUDA achieve a slightly better throughput than OpenCL. From this analysis results that GPUs and FPGAs achievements are comparable in terms of throughput. However, several differences are noticeable in the way these results are achieved on both devices. One of the factors that make the difference in achieving high throughput for FPGAs is the presence of high-bandwidth I/O capabilities, since clock speed is relatively small compared to ordinary graphics processor units. On the other hand, host-device I/O bandwidth is usually a limiting constraint for performance achievement on GPU, but this is usually compensated by the possibility of limiting data transfer for devices equipped with extended on board memory and by the high clock speed at the expense of increased power consumption levels. In the context of hardware accelerator design, where both FPGAs and GPUs are currently widely used, I/O capabilities are maybe the best point to evaluate the choice of one over the other achievement when the main concern is high performance.

9. Conclusion

This paper presents a direct comparison between FPGA and GPU used as accelerators for the AES cipher. The analysis of the results achieved on both has been compared to several others in order to establish which device is best at playing the role of hardware accelerator. In addition, the possibility of making a direct comparison between such different architectures have been investigated. This analysis suggests that, while hardware design on FPGA remains the natural choice for consumer-product design, GPUs are nowadays the preferable choice for PC based accelerators, especially when the processing routines are highly parallelizable. In fact, FPGA devices are still capable of delivering very high performance at low power consumption, but the possibility of programming GPUs with procedural paradigms, using the OpenCL or CUDA technologies, helped in making GPGPU an alternative to the use of FPGAs in the context of high performance computing, compensating for high power consuming levels.

References

[1] Nvidia "Fermi Compute Architecture Whitepaper", 2010.

[2] D. Lu, Q. Chen, K. Weng, and A. Zhang, "High speed video signal acquisition and processing system based on FPGA technique", in *Proc. Int. IEEE Conf. on Neural Netw. & Sig. Process.*, Nanjing, China, 2003, pp. 1233–1236.

[3] K. F. K. Wong, V. Yap, and T. P. Chiong, "Hardware accelerator implementation on FPGA for video processing", in *Proc. IEEE Conf. on Open Syst. ICOS 2013*, Kuching, Malaysia, 2013, pp. 47–51 (doi: 101109/ICOS.2013.6735046).

[4] V. Conti, C. Militello, F. Sorbello, and S. Vitabile, "Biometric sensors rapid prototyping on FPGA", *The Knowl. Engin. Rev.*, vol. 30, no. 2, pp. 201–219, 2015.

[5] C. Militello, V. Conti, S. Vitabile, and F. Sorbello, "An embedded iris recognizer for portable and mobile devices", *Comp. Syst. Science and Engin.*, vol. 25, no. 2, pp. 119–131, 2010.

[6] N. Neveset *et al.*, "BioBlaze: Multi-core SIMD ASIP for DNA sequence alignment", in *Proc. Int. IEEE Conf. on Appli.-Spec. Syst., Architec. & Process. ASAP 2013*, Ashburn, VA, USA, 2013, pp. 241–244.

[7] E. Fusella, A. Cilardo, and A. Mazzeo, "Scheduling-aware interconnect synthesis for FPGA-based Multi-Processor Systems-on-Chip", in *Proc. 25th Int. IEEE Conf. on Field Programmable Logic & Appl. FPL 2015*, London, UK, 2015, pp. 1–2.

[8] S. Vitabile, V. Conti, C. Militello, and F. Sorbello, "An extended jade-s based framework for developing secure multi-agent systems", *Comp. Stand. and Interfaces*, vol. 31, no. 5, pp. 913–930, 2009.

[9] Y. Wang and Y. Ha, "FPGA based Rekeying for cryptographic key management in Storage Area Network", in *Proc. 23rd Int. IEEE Conf. on Field Programmable Logic and Appl. FPL 2013*, Porto, Portugal, 2013, pp. 1–6.

[10] C. Militello, V. Conti, S. Vitabile, and F. Sorbello, "Embedded access points for trusted data and resources access in HPC systems", *The J. of Supercomput.*, vol. 55, no. 1, pp. 4–27, 2011.

[11] B. Cope, P. Cheung, W. Luk, and S. Witt, "Have GPUs made FPGAs redundant in the field of video processing?", in *Proc. Int. IEEE Conf. on Field-Programmable Technol. FPT 2005*, Singapore, 2005, pp. 111–118.

[12] A. B. M. Mali and F. Novak, "Hardware implementation of AES algorithm", *J. of Elec. Engin.*, vol. 56, no. 9–10, pp. 265–269, 2005.

[13] Z. Baker, M. Gokhale, and J. Tripp, "Matched filter computation on FPGA, cell and GPU", in *Proc. 15th IEEE Symp. on Field-Programmable Custom Comput. Machin. FCCM 2007*, Napa, CA, USA, 2007, pp. 207–218.

[14] S. Che, J. Li, J. Sheaffer, K. Skadron, and J. Lach, "Accelerating compute-intensive applications with GPUs and FPGAs", in *Proc. Symp. on Appl. Specific Process. SASP 2008*, Anaheim, CA, USA, 2008, pp. 101–107.

[15] S. Asano, T. Maruyama, and Y. Yamaguchi, "Performance comparison of FPGA, GPU and CPU in image processing", in *Proc. 19th Int. IEEE Conf. on Field Programmable Logic and Appl. FPL 2009*, Prague, Czech Republic, 2009, pp. 126–131.

[16] N. Kapre and A. DeHon, "Performance comparison of single-precision spice model-evaluation on FPGA, GPU, cell, and multi-core processors", in *Proc. 19th Int. IEEE Conf. on Field Programmable Logic and Appl. FPL 2009*, Prague, Czech Republic, 2009, pp. 65–72.

[17] Y. Zhang, Y. Shalabi, R. Jain, K. Nagar, and J. Bakos, "FPGA vs. GPU for sparse matrix vector multiply", *Proc. Int. IEEE Conf. on Field-Programmable Technol. FPT 2009*, Sydney, Australia, 2009, pp. 255–262.

[18] K. Theoharoulis, C. Antoniadis, N. Bellas, and C. Antonopoulos, "Implementation and performance analysis of seal encryption on FPGA, GPU and multi-core processors", in *Proc. 19th Ann. Int. IEEE Symp. on Field-Programmable Custom Comput. Machines FCCM 2011*, Salt Lake City, UT, USA, 2011, pp. 65–68.

[19] "Federal Information Processing Standards Publication 197", Tech. rep., National Institute of Standards and Technology, 2001 [Online]. Available: http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

[20] M. Dworkin, "Recommendation for Block Cipher Modes of Operation", Tech. rep., National Institute of Standards and Technology, 2001 [Online]. Available: http://csrc.nist.gov/publications/nistpubs/800-38a/sp800-38a.pdf

[21] Celoxica – Mentor Graphics, Handel-C advanced optimization [Online]. Available: https://www.mentor.com/

[22] Celoxica – Mentor Graphics, RC1000 Hardware Reference Manual [Online]. Available: https://www.mentor.com/

[23] Cypress. CY7C1049 512K x 8 Static RAM datasheet [Online]. Available: http://www.cypress.com/file/42811/download

[24] "The OpenCL Specification v 1.2", Tech. rep., Khronos OpenCL Working Group, 2011 [Online]. Available: http://www.khronos.org/registry/cl/specs/opencl-1.2.pdf

[25] T. Hoang and V. L. Nguyen, "An efficient FPGA implementation of the advanced encryption standard algorithm", in *Proc. Int. IEEE Conf. on Comput. & Commun. Technol., Res., Innov., and Vision for the Future RIVF 2012*, Ho Chi Minh, Vietnam, 2012, pp. 1–4.

[26] N. S. E. Rodriguez-Henriquez and A. Diaz-Pkrez, "4.2 Gbit/s single-chip FPGA implementation of AES algorithm", *Electron. Lett.*, vol. 39, no. 15, pp. 1115–1116, 2003.

[27] D. Kotturi, S.-M. Yoo, and J. Blizzard, "AES crypto chip utilizing high-speed parallel pipelined architecture", in *Proc. Int. IEEE Symp. on Circ. and Syst. ISCAS 2005*, Kobe, Japan, 2005, vol. 5, pp. 4653–4656.

[28] C. Sivakumar and A. Velmurugan, "High speed VLSI design CCMP AES cipher for WLAN (IEEE 802.11i)", *Int. Conf. on Sig. Process., Commun.*, Chennai, India, 2007, pp. 398–403.

[29] R. M. Gurmail Singh, "FPGA based high speed and area efficient AES encryption for data security", *Int. J. of Res. and Innov. in Comp. Engin.*, vol. 1, no. 2, pp. 53–56, 2011.

[30] S. Manavski, "CUDA compatible GPU as an efficient hardware accelerator for AES cryptography", in *Proc. International IEEE Conference on Signal Processing and Communications ICSPC 2007*, Dubai, United Arab Emirates, 2007, pp. 65–68.

[31] X. Wang, X. Li, M. Zou, and J. Zhou, "AES finalists implementation for GPU and multi-core CPU based on OpenCL", in *Proc. Int. IEEE Conf. on Anti-Counterfeiting, Secur. & Identif. ASID 2011*, Xiamen, China, 2011, pp. 38–42.

[32] T. K. Keisuke Iwai, N. Nishikawa, "Acceleration of AES encryption on CUDA GPU", *Int. J. of Netw. & Comput.*, vol. 2, no. 1, pp. 131–145, 2011.



Vincenzo Conti is an Assistant Professor with the Faculty of Engineering and Architecture at the Kore University of Enna, Italy. He received his Laurea cum Laude degree and his Ph.D. in Computer Science from the University of Palermo in 2000 and 2005, respectively. His research interests include biometric recognition systems, programmable architectures, bio-inspired processing system, and user ownership in multi-agent systems. He has joined the Editorial Board of the International Journal on Security and Communication Networks (Hindawi), of the International Journal of Biosensors and Bioelectronics (MedCrave) and of the Complex and Intensive Systems Journal (Springer), has chaired and participated at several national and international conferences, and he has co-authored over 60 scientific publications, journals and conferences. Moreover, he has participated to several research projects funded by industries and research institutes in his research areas. Currently, he collaborates with the Department of Industrial and Digital Innovation (DIID) and the Department of Biopathology and Medical Biotechnologies (DBMED) of the University of Palermo, and with Italian National Council of Researches (CNR – Cefalù, Palermo).

E-mail: vincenzo.conti@unikore.it
Faculty of Engineering and Architecture
University of Enna KORE
Via delle Olimpiadi - Cittadella Universitaria
94100 Enna, Italy



Salvatore Vitabile received the Laurea degree in Electronic Engineering and the Ph.D. degree in Computer Science from the University of Palermo in 1994 and 1999, respectively. He is currently an Associate Professor with the Department of Biopathology and Medical Biotechnologies, University of Palermo, Italy. In 2007, he was a visiting professor in the Department of Radiology, Ohio State University, Columbus, USA. Dr. Vitabile is the Editor-in-Chief of the International Journal of Adaptive and Innovative Systems (Inderscience Publishers). He has also joined the Editorial Board of the Journal of Ambient Intelligence and Humanized Computing (Springer), the International Journal of Information Technology, Communications and Convergence (Inderscience Publishers), and the International Journal of Space-Based and Situated Computing (Inderscience Publishers). He has co-authored more than 150 scientific papers in referred journals and conferences. He has chaired, organized, and served as member of the organizing committee of several international conferences and workshops. Dr. Vitabile is currently member of the Board of Directors of SIREN (Italian Society of Neural Networks). His research interests include specialized architecture design and prototyping, biometric authentication systems, driver assistance systems, and medical data processing and analysis.

E-mail: salvatore.vitabile@unipa.it
Department of Biopathology and Medical Biotechnologies
University of Palermo
Via del Vespro 129
90127 Palermo, Italy

Paper

A Queue Monitoring System in OpenFlow Software Defined Networks

Shiva Rowshanrad, Sahar Namvarasl, and Manijeh Keshtgari

Computer Engineering and IT Department, Shiraz University of Technology, Shiraz, Iran

Abstract—Real-time traffic characteristic is different and it is very sensitive to delay. To meet traffic specifications in real time, monitoring systems are used as an important part of networking. Many monitoring systems are deployed to have an update view of the network QoS parameters and performance. Most of these systems are implemented to measure QoS parameters in links. Here, in this paper, a system for monitoring queues in each link by means of Software Defined Networks is proposed. The monitoring system is implemented by extending Floodlight controller, which uses OpenFlow as southbound protocol. The controller has a centralized view of the network. By the help of OpenFlow it also can provide flow level statistics. Using these advantages, the proposed system can monitor delay and available bandwidth of a queue on a link or path. Despite of monitoring systems in traditional networks, the proposed monitoring system makes a low overhead in network thanks to OpenFlow protocol messages. It is also integrated into the network controller, which enables QoS and traffic engineering applications to use the system's reports for automatic traffic management and QoS setup. The experimental results show a 99% accuracy of the proposed system for monitoring of both bandwidth and delay.

Keywords—Floodlight, OpenFlow, queue monitoring, Software Defined Networks.

1. Introduction

In today's networks, there are many kinds of traffic, such as video streaming, video conferencing, VoIP, FTP, etc. Each of these traffics has different QoS requirements. VoIP and video conferencing need less than a 150 ms delay and less than 1% packet loss [1]. Therefore, these data types need different traffic engineering (TE) to transmit efficiently. Queuing disciplines are common examples of TEs. However, in order to have a complete traffic management a queuing monitoring system is required, which can provide a real-time report about QoS parameters of each queue on a requested path or link.

Many monitoring systems have been proposed in traditional networks, but usually they consume too many resources such as bandwidth and computational power, they may need additional hardware and are not very accurate nor flexible. Also some of these methods may not work properly on heavy load [2], [3].

Recently many have taking advantage of Software Defined Networks (SDNs) and OpenFlow protocol to create monitoring systems. In SDN, control plane is separated from data plane and placed into a centralized server named controller. The controller communicates with network forwarders using an open interface such as OpenFlow [4]. Each forwarder keeps track of counters related to packets and flows. The controller can be aware of these counters by polling forwarders using OpenFlow statistic messages. The counters are per table, per flow, per port, per queue, including received packet/byte, transmitted packet/bytes, their duration, number of received/transmitted drops, etc. [5].

In this paper a system for monitoring delay and available bandwidth of queues, is proposed. It provides TE for QoS management.

The remainder of this paper is organized as follows. In Section 2, related works in SDNs and traditional networks are described. Section 3 introduces different parts of the proposed monitoring system. In Section 4, the experimental tests and their results are presented. Finally, Section 5 concludes the paper.

2. Related Works

One of the earliest tools for network monitoring is Simple Network Management Protocol (SNMP). It uses port counters across every switch to estimate links load. Although SNMP is widely used in many monitoring devices in traditional networks, it has some drawbacks. First, it may result high CPU overhead. Moreover, SNMP is unable to collect flow-level statistics and measuring metrics such as loss and delay. It also requires additional infrastructure [6], [7].

The sFlow [8] and NetFlow [9] are two flow-based monitoring systems, which use packet sampling. They both use agents at switches and routers to sample packets and collect statistics. sFlow agents can push the information to a centralized collector, while NetFlow agents would be polled by the collector.

Recently many monitoring/measurement systems for SDNs are proposed. IBM had leveraged sFlow sampling and implemented OpenSample over Floodlight controller [10]. Using packet sampling from flows, which have sequence numbers (e.g. TCP), by centralized collector, OpenSample can have a fast detection of elephant flows and link utiliza-

Table 1
Monitoring systems comparison

Monitoring system	Network	Parameters	Method	Implementation	Strengths and weaknesses
SNMP	Traditional	Bandwidth/link utilization	Poll port counters which gather information about packets	SNMP manager and agents	• single parameter, • high CPU utilization, • needs additional infrastructure
sFlow			Packet sampling agents push information to a centralized collector	sFlow agents and collector	• single parameter, • can be used in SDN but not appropriate, • flow-based measurement, • needs additional infrastructure
NetFlow			Polling packet sampling agents	NetFlow agents and collector	• single parameter, • can be used in SDN but not appropriate, • high CPU usage, • Flow-based measurement, • needs additional infrastructure
OpenSample	SDN	Flow/link utilization	Pocket sampling agents push information to a centralized collector	Floodlight controller	• single parameter, • without end device modification, • high accuracy, • low latency
PayLess		Bandwidth/link utilization	Polling OpenFlow (OF) statistics from switches	Flodlight controller	• single parameter, • trade-off between accuracy and overhead in different polling intervals
FlowSense		Per flow link utilization	Push based	Not described	• single parameter, • high accuracy, • low overhead
Phemius *et al.* [2]		Link latency	Polling OF messages, use of probe packets	Floodlight controller	• single parameter, • high accuracy, • low overhead
OpenTM		Bandwidth/link utilization	Polling based (5 different methods)	NOX controller	• single parameter, • high accuracy, • high overhead on edge switches
OpenNetMon		Per flow packet loss, delay and throughput	Polling OF messages, use of probe packets	POX controller	• trade-off between accuracy and overhead

tion. The throughput of OpenSample can be up to 150% over sFlow in some cases.

In [11] a monitoring system named PayLess is implemented over Floodlight [12]. PayLess collects statistics from switches by polling them. The applications on top of the controller can request the desired QoS metrics, which can be extracted from these statistics, using RESTful API. As a use case, PayLess was evaluated for link utilization information. The results show that PayLess can collect more accurate statistics than FlowSence [13], which estimates link utilization by analyzing control messages sent from switches to the controller instead of polling switches. In addition, PayLess messaging overhead is 50% of overhead in an equivalent method using polling statistics.

In [2] a link latency monitor over Floodlight controller is proposed. The method for measuring latency is based on sending an Ethernet frame, with an unknown Ethernet-type value, over the link from the controller and measuring the time until the packet comes back. The difference of this time from half of the Round Trip Time (RTT) of statistic messages between edge switches of the link and controller would be the latency value of that link. The overhead of this method is 81% less than ping utility while its accuracy is 99.25% comparing to ping.

In [14] OpenTM is presented. OpenTM is a traffic matrix estimator, implemented using NOX controller [15]. It presents the traffic load between each pair of switches in an OpenFlow network by polling statistics. Due to packet

loss, the statistics of different switches in path would have different results. For this reason, five design methods were compared to each other: polling the last switch, polling switches uniformly at random, round robin querying, none uniform random polling, polling the least loaded switch. The evaluation results show that polling the last switch gives the most accurate values. However, it makes a high load on edge switches.

OpenNetMon [3] is another SDN monitoring system. It is implemented over POX controller and monitors per-flow metrics such as throughput, delay and packet loss. The flow throughput is measured using statistics related to amount of bytes sent in a flow and its duration. The required statistics are achieved by polling the last switch, while in case of packet loss measurement. The statistics are polled from the first and last switch of the path. The packet loss value is calculated from subtract of switches packet counters. An equivalent method as in [2] is used for delay monitoring by means of probe packets. The results of evaluation show that the proposed method for monitoring throughput is quite accurate, while methods for packet loss and delay measurements are not. These inaccuracies caused by lack of synchronization between measurement setups and software fluctuations. Table 1 shows the comparison between mentioned monitoring systems.

3. Proposed Monitoring System

All the systems, which have been proposed in SDNs were for monitoring links. Here, in this paper a monitoring system, which can monitor queues in each link or path in terms of delay and available bandwidth is proposed. The authors extended Floodlight controller, which uses OpenFlow protocol, for implementing this system.

The key parameter of QoS is bandwidth. If the bandwidth were insufficient for transmitting traffics, loss and delay would be also unfavorable. For measuring the used bandwidth of each queue on each link, there is a need to poll the "transmitted bytes" counter of the queue from the switch at the head of the link. It is required to keep track of the last polling time and counter to calculate the used bandwidth of queue. Then the bandwidth would be calculated as:

$$BWq = \frac{TB_n - TB_{n-1}}{t_n - t_{n-1}}, \tag{1}$$

where BWq stands for used bandwidth of the desired queue, TB is the transmitted bytes and t is the polling time in seconds. The free capacity can be calculated by subtracting BWq from the base bandwidth of the queue.

Delay is another important QoS parameter, which also can have a huge effect on packet loss. Different applications have different delay requirement, so it is important to determine, which queue can best satisfy which application at the moment.

The delay of each link can be calculated by sending a packet with arbitrary Ethernet-type value from the controller to the switch located at the head of the link. Then the switch sends this packet to the next switch at the other side of the link. As there is no entry in the flow table, matching this Ethernet-type value, the second switch sends the packet back to the controller. So the controller knows the times of sending and receiving the packet, hence it knows the duration.

For measuring delay of a queue, the packet should be queued. It means beside the output action, an enqueue action must be set for the packet. In this case, the time between sending and receiving the packet consists of queuing delay and propagation delay. Multi-arbitrary Ethernet-type values can be used to determine between the probe packets related to each queue.

For having the exact delay of queue on a link between two switches, it is necessary to subtract the measured time from half of RTT's between the switches and the controller. This RTT can be measured using Statistics_Request and Statistics_Reply messages. Equation (2) shows the delay of a queue on a specific link:

$$\text{Delay}_q = T_c - \frac{\text{RTT}_{s1}}{2} - \frac{\text{RTT}_{s2}}{2}, \tag{2}$$

where T_c is the duration of sending and receiving the probe packet by controller.

4. Experimental Results and Performance Evaluation

The monitoring module was implemented by extending Floodlight controller. OpenFlow was used as the southbound protocol and OpenVswitch [16] was installed on Ubuntu 14.04 for creating OpenFlow switches. A linear topology with three switches and four hosts running on Intel Core i5/i7 CPUs PC with Windows 8/Ubuntu 14.04 (as shown in Fig. 1) was used.

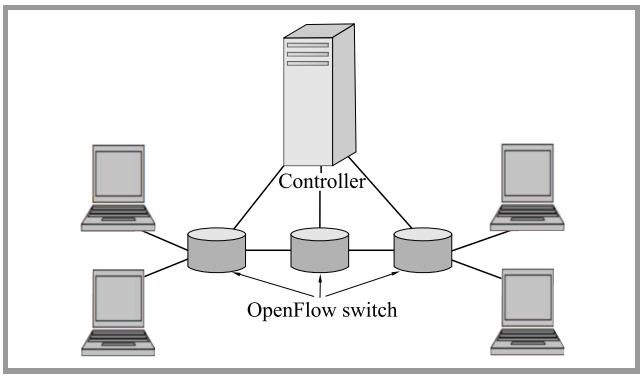


Fig. 1. Performance evaluation topology.

For testing the proposed monitoring system, two queues with maximum bandwidth of 30 Mb/s were created on

each link, and then special TCP and UDP flows such as video streaming, video conferencing, etc., were allocated to each, by means of floodlight RESTful API. We also tried to generate different TCP/UDP flows using traffic generators in every host, flowing to the opposite one. The flows were increased in time to fulfill the queues' bandwidth in about 15 s. The switches polling interval of proposed system is set to 1 s. The bandwidth was also checked in every second by JPERF tool. Some special flows were monitored by Wireshark to be analyzed for delay. As the proposed system can measure the latency of queues in links between switches, Wireshark application also set to monitor the packets on edge switches to have accurate measurements.

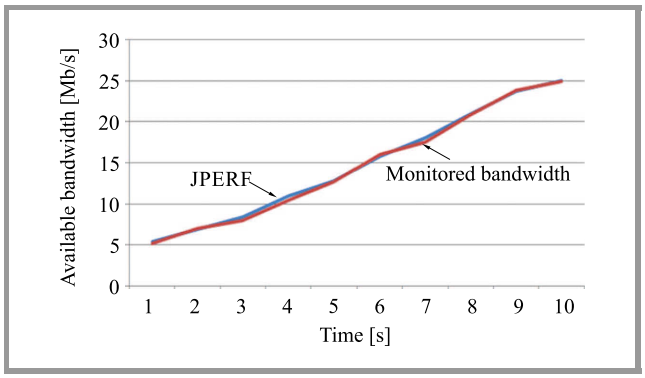


Fig. 2. Comparing monitored bandwidth by JPERF and proposed monitoring method. (See color pictures online at www.nit.eu/publications/journal-jtit)

Figure 2 shows the monitored bandwidth in first 10 s of the test. According to analysis in Table 2, the proposed method results are as the same as JPERF results because the 95% confidence interval of system averages includes zero. From the averages, we can say that the proposed system accuracy compared to JPERF is 99%. One of the advantages of proposed system over JPERF is that the system can monitor bandwidth even if the queue is full. JPERF may face difficulties in high-load networks from when there is a little bandwidth left, and crashed in last seconds of test.

Table 2
Monitored bandwidth analysis

JPERF average bandwidth [Mb/s]	Monitored average bandwidth [Mb/s]	95% confidence interval of averages difference
7.01	7.07	(–0.03, 0.33)

For delay measurement performance evaluation, the delays of both queues were monitored with monitoring module. Also the average delay of some special flow was calculated from the monitored results of Wireshark.
The q0 is the lowest priority queue while the q1 has higher priority. Figure 3 shows the monitored delay in

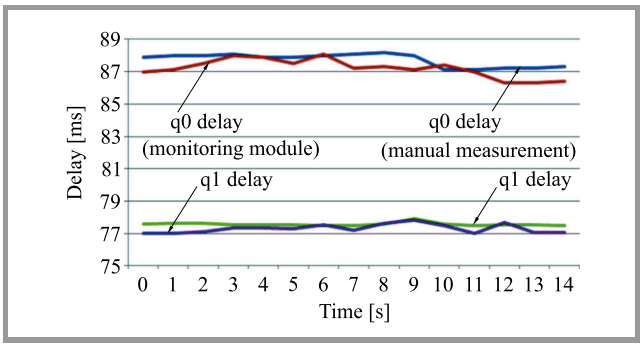


Fig. 3. Comparing monitored delay of queues.

Table 3
Delay analysis of monitored queues

Delay in 15 s	Average of monitoring module measurement in 15 s	Average of manual measurement in 15 s	95% confidence interval of averages difference
q0	87.21	87.73	(0.28, 0.78)
q1	77.30	77.57	(0.13, 0.41)

each second for these queues. Although the calculated delay values, monitored delay values and their averages are close to each other, the confidence intervals in Table 3 do not include zero. This means that the systems do not have the exactly the same results. This difference can be due to the difference between delays of each two different flows in the queue. The delay values of flows in a queue differ in a short range because of their difference in packet size, processing time, etc. Therefore, the monitored delay can be considered as an estimated delay of desired queue in desired path with a high accuracy about 99%.

5. Conclusion

In this paper, a monitoring system for measuring queue QoS parameters such as available bandwidth and delay is proposed. The proposed monitoring system is the first system for monitoring queue parameters for SDNs. It is implemented over Floodlight controller and uses OpenFlow statistic messages and probe packets to measure the mentioned parameters. Use of OpenFlow protocol messages gives the advantage of monitoring network with low network overhead. Integrating the system as a software module makes it independent of using network devices' resources. It also is a cheaper system comparing to traditional monitoring systems, as it doesn't need extra infrastructure. The performance evaluation of the system shows an accuracy of 99% for measuring both available bandwidth and delay of a queue in a desired path or link. The next step is to use this monitoring system for networks, which uses queuing as their TE method, for optimizing network's performance automatically.

References

[1] T. Szigeti and C. Hattingh, *Quality of Service Design Overview*. San Jose, CA: Cisco Press, 2004, pp. 15–16.

[2] K. Phemius and M. Bouet," Monitoring latency with OpenFlow", in *Proc. 9th Int. Conf. on Netw. and Service Manag. CNSM 2013*, Zürich, Switzerland, 2013, pp. 122–125 (doi:10.1109/CNSM.2013.6727820).

[3] N. L. M. van Adrichem, C. Doerr, and F. A. Kuipers, "OpenNetMon: Network monitoring in OpenFlow software-defined networks", in *Proc. IEEE/IFIP Network Operations and Management Symp. NOMS 2014*, Kraków, Poland, 2014, pp. 1–8 (doi:10.1109/NOMS.2014.6838228).

[4] S. Rowshanrad, S. Namvarasl, A. Abdi, M. Hajizadeh, and M. Keshtgary, "A survey on SDN, the future of networking", *J. of Adv. Comp. Sci. & Technol.*, vol. 3, no. 2, pp. 232-248, 2014 (doi:10.14419/jacst.v3i2.3754).

[5] OpenFlow Switch Consortium, OpenFlow Switch Specification Version 1.0.0., 2009 [Online]. Available: http://archive.openflow.org/documents/openflow-spec-v1.0.0.pdf

[6] W. Stallings, *SNMP, SNMPv2, SNMPv3, and RMON 1 and 2*. Boston: Addison-Wesley Longman Publ. Co., 1998.

[7] J. Case, M. Fedor, M. Schoffstall, and C. Davin, "A simple network management protocol (SNMP)", RFC 1098, Network Information Center, SRI International, Menlo Park, CA, USA, 1989 [Online]. Available: http://archive.openflow.org/documents/openflow-spec-v1.0.0.pdf

[8] P. Phaal, "sFlow Specification Version 5", July 2004 [Online]. Available: http://sflow.org/sflow_version_5.txt (accessed: 25 May 2015).

[9] B. Claise, "Cisco systems NetFlow services export version 9", 2004 [Online]. Available: http://tools.ietf.org/html/rfc3954.html (accessed: 25 May 2015).

[10] J. Suh, T. Kwon, C. Dixon, W. Felter, and J. Carter, "OpenSample: A low-latency, sampling-based measurement platform for SDN", in *Proc. IEEE 34th Int. Conf. Distrib. Comput. Syst. ICDCS 2014*, Madrid, Spain, 2014, pp. 228–237 (doi:10.1109/ICDCS.2014.31).

[11] S. R. Chowdhury, M. F. Bari, R. Ahmed, and R. Boutaba, "PayLess: A low cost network monitoring framework for software defined networks", in *Proc. IEEE/IFIP Netw. Operat. and Manag. Symp. NOMS 2014*, Kraków, Poland, 2014 (doi:10.1109/NOMS.2014.6838227).

[12] Floodlight OpenFlow Controller [Online]. Available: http://www.projectfloodlight.org/floodlight/ (accessed: 25 May 2015).

[13] C. Yu, C. Lumezanu, Y. Zhang, V. Singh, G. Jiang, and H. V. Madhyastha, "FlowSense: Monitoring network utilization with zero measurement cost" in Proc. 14th Int. Conf. on Passive and Active Measurement PAM'13, Hong Kong, China, 2013, pp. 31-41 (doi:10.1007/978-3-642-36516-4_4).

[14] A. Tootoonchian, M. Ghobadi, and Y. Ganjali, "OpenTM: Traffic matrix estimator for OpenFlow networks", in Proc. 11th Int. Conf. on Passive and Active Measurement PAM'10, Zurich, Switzerland, 2010, pp. 201–210 (doi:10.1007/978-3-642-12334-4_21).

[15] NOX [Online]. Available: http://www.noxrepo.org/nox/about-nox/ (accessed: 25 May 2015).

[16] OpenVSwitch [Online]. Available: http://openvswitch.org (accessed: 25 May 2015).



Shiva Rowshanrad received her M.Sc. degree in Information Technology Engineering (Computer networks field) from Shiraz University of Technology (SUTECH), Shiraz, Iran. Her main research interest is Software Defined Networking. Her other research interests are Named Data Networking, Wireless Sensor Networks and multimedia.
E-mail: shiva.rrad@gmail.com
Computer Engineering and IT Department
Shiraz University of Technology
Shiraz, Iran



Sahar Namvarasl received her B.Sc. in Information Technology from Shiraz University of technology (SUTECH), Shiraz, Iran. She is currently working toward M.Sc. degree at SUTECH, majoring in computer networks. Her research interests are in the area of Software Defined Networks, virtualization and cloud computing.
E-mail: sahar.namvarasl@gmail.com
Computer Engineering and IT Department
Shiraz University of Technology
Shiraz, Iran



Manijeh Keshtgari is a faculty member of Department of Computer Engineering and IT, Shiraz University of Technology, Shiraz, Iran. She received her M.Sc. degree in Electrical and Computer Engineering from Colorado State University, CSU, Fort Collins, USA in 1993 and her Ph.D. degree in Computer Engineering from Sharif University of Technology in 2005. Her research interests include MANET, Wireless Sensor Networks and GSM security issues.
E-mail: keshtgari@sutech.ac.ir
Computer Engineering and IT Department
Shiraz University of Technology
Shiraz, Iran

Paper

Sensor Hop-based Energy Efficient Networking Approach for Routing in Underwater Acoustic Communication

Sheena Kohli and Partha Pratim Bhattacharya

Mody University of Science and Technology, Lakshmangarh, Rajasthan, India

Abstract—Underwater Wireless Sensor Networks are deployed to explore the world under the water, measure different parameters and communicate the data to the surface, in the widespread applications. The main operating technology of these networks is the acoustic communication. The communication among the sensors and finally to the surface station requires a routing protocol. The sensors being battery limited and unfeasible to be replaced under the water requires an energy efficient routing protocol. Clustering imparted in routing is an energy saving technique in sensor networks. The routing may involve single or multi hop communication in the sensor networks. The paper gives a comparative study of the benchmark protocol multi-hop LEACH with the proposed Sensor Hop-based Energy Efficient Networking Approach (SHEENA) for the shallow as well as deep water in three dimensional Underwater Wireless Sensor Networks. The network energy model for the Underwater Wireless Sensor Networks is based among the different acoustic channel characteristics. The proposed approach is found to give better response.

Keywords—attenuation, clustering, multi-hop routing, signal to noise ratio, transmission loss.

1. Introduction

A category of Wireless Sensor Networks (WSN), known as Underwater Wireless Sensor Network (UWSN), comprises of the sensors or the nodes which are wirelessly connected to each other, deployed under the sea or ocean or any water body. Underwater wireless sensor nodes are tiny devices, equipped with sensing units, capable of detecting data from the external environment and communicating this data to the surface sink or the Base Station (BS). Each sensor node transmits and receives data packets. Underwater acoustic communication is the technique of sending and receiving message below water [1].

Figure 1 shows the basic view of UWSN environment. The deployment of nodes may be two-dimensional (2D) or three-dimensional (3D) in UWSN. The 2D UWSN involves the nodes to be anchored to the bottom of the ocean. The arrangement of nodes is in the form of clusters or groups. Each cluster has a cluster head, which acts as a gateway or relay for transmitting the collected data to the surface station after processing it. In 3D UWSN, the nodes are placed at the different depth levels of the water. The nodes either may be hanged from the surface buoy floating on the top of the water surface or may be deployed with the help of anchor drawn sensor devices placed at the bottom [2].

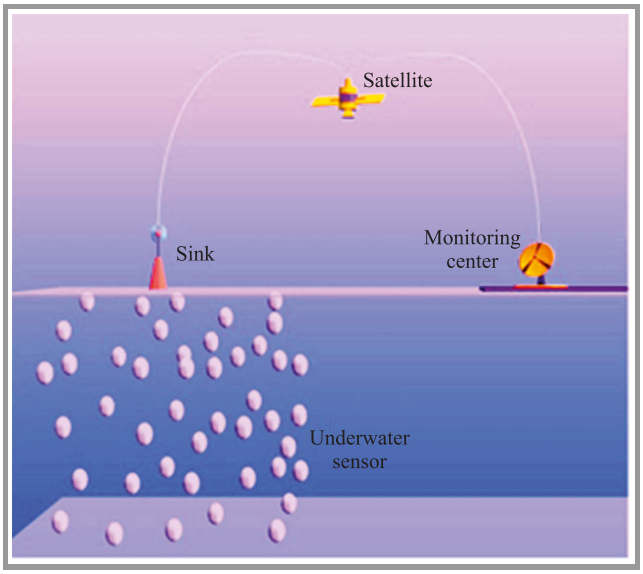


Fig. 1. Underwater Wireless Sensor Network concept.

UWSNs are being widely utilized in different areas of marine research including environmental monitoring, disaster prevention, micro-habitat monitoring, oil and gas exploration, sensing of chemical contamination and biological phenomena, distributed tactical surveillance, seismic studies, etc. [1], [3]. The topic is still in the beginning stage compared to its terrestrial counterpart due to the involvement of high cost and physical challenges.

To understand the basics of UWSNs, we can utilize many design principles and tools used in terrestrial sensor networks. But they are characteristically different in some fundamental points. Most importantly, radio is unsuitable for underwater sensors due to their limited propagation ability [4]. This is when acoustic signals are being utilized for underwater communication, which again poses many challenges like path loss, noise, multi-path, Doppler spread, and high and variable propagation delay [5]. Hence, the requirement for specially designed routing protocols for UWSNs becomes inevitable. Thus, intense research is be-

ing undertaken for designing efficient protocols considering the unique characteristics of underwater communication networks.

2. Routing in Underwater Wireless Sensor Networks

WSNs are formed by miniature devices interacting over radio wireless links without using a determined networked infrastructure. Because of restricted transmission range, communication between any two devices requires associating intermediate forwarding network nodes [6].

Routing is a process of determining a path between source and destination upon request of data transmission. Designing an optimum routing protocol is the basic issue involved with any network. The sensor networks generally depend on gateway nodes to handle huge amounts of data over extended ranges. The field of underwater sensor networking and routing protocols are in the incipient stage of research. Earth comprising of majority of water, gives a lot of opportunity to explore this field. Sensor networks being limited in battery power, allows finding method to support the development of energy efficient protocols in wireless sensor network [7].

Grouping sensor nodes into clusters has been widely adopted by researchers to assure the scalability and achieve high-energy efficiency to prolong network lifetime in WSN environments. The hierarchical cluster-based organization of the sensor nodes allow data fusion and aggregation, thus leading to significant energy savings. Clustering involves hierarchically organizing the network topology. Sensor nodes in cluster architecture are grouped into clusters in which a cluster head is elected and group of source sensor nodes are directly attached to the cluster head. The cluster head usually performs the special tasks like (fusion and aggregation) and several common sensor nodes as members [8]. Figure 2 shows the cluster arrangement in WSNs.

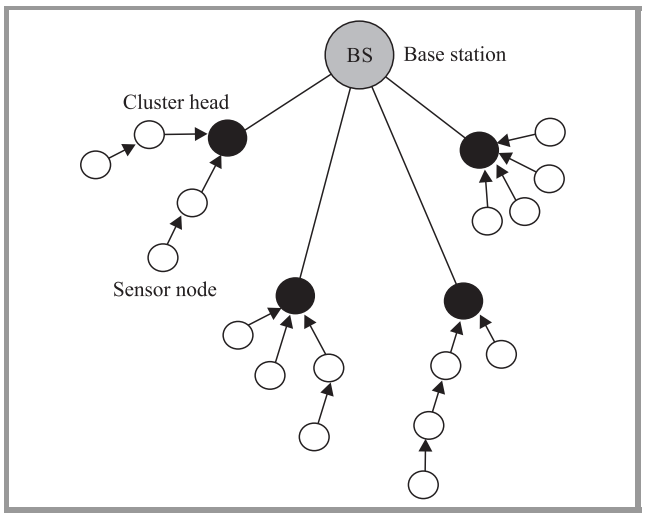


Fig. 2. Clustering in WSNs.

Generally, a clustered network employs single hop routing in each cluster. The one-hop clustering can reduce the energy consumption of communication by forwarding source nodes data to the cluster head via one hop. However, when communication distance increases, single-hop communication consumes more energy and becomes less energy efficient method. For a large network, where internodes' distance is important, multi-hop communication is a more energy efficient approach [9]. Therefore, a new approach called Sensor Hop-based Energy Efficient Networking Approach (SHEENA) for 3D UWSNs is proposed and compared with the widely used protocol called multi-hop Low-Energy Adaptive Clustering Hierarchy (LEACH).

2.1. LEACH Protocol

LEACH [10] is the first self-adaptive and self-organized protocol of hierarchical routings, which proposed data fusion. It is of milestone significance in clustering routing protocols. LEACH protocol uses round as unit. Each round is made up of setup stage and steady-state stage. For reducing unnecessary energy costs, the later must be much longer than the former one.

At the stage of cluster forming, a node randomly picks a number between 0 to 1, compares this number to the threshold values $t(n)$, if the number is less than $t(n)$, then it becomes cluster head in this round, else it becomes the common node. Threshold $t(n)$ is determined by the following equation:

$$t(n) = \frac{P}{1 - P\left(r \bmod \frac{1}{P}\right)} \quad \text{if} \quad n \in G \quad \text{else} \quad 0, \qquad (1)$$

where P is the percentage of the cluster head nodes in all nodes, r is the number of rounds and G is the collection of the nodes that have not yet been head nodes in the first $\frac{1}{P}$ rounds.

When clusters have been formed, the nodes start to transmit the captured data. Cluster heads receive data sent from the other nodes and forward it to the sink after being fused. This is a frame data transmission. In order to reduce unnecessary energy cost, steady stage is composed of multiple frames and the steady stage is much longer than the setup stage. Here we perform multi-hop LEACH [11] for acoustic channel. Multi-hop LEACH protocol is almost the same as LEACH protocol, only makes communication mode from single-hop to multi-hop between cluster heads and sink.

2.2. SHEENA

UWSNs are composed of a large number of pre-powered battery operated sensors deployed in the target environment. To achieve energy-efficient, scalable and fault-tolerant system structure, SHEENA is proposed as strategy to reduce the power consumption. In this model, network is presented with a predefined number of nodes. These nodes are divided into their respective roles. The node roles possible in presented model are as sensor nodes, cluster heads

and super heads. Nodes are randomly deployed under the water. The sensor nodes sense and send the data to their respective cluster heads. The cluster heads forward the collected data to the super heads, which are assumed as energy rich devices, capable of doing data aggregation and processing in an efficient manner. The super head is a powerful node in the underwater wireless sensor network and it can reach a wide range of communication area. The super head serves as the gateway for external communication. If the super head has been invaded then the whole network will be taken over, so it is assumed that the super head is well protected and can always be trusted. Cluster head is selected based on energy. Node having maximum energy among all other sensors is elected as a cluster head. The same applies for the super heads too.
Both the approaches have been explained in Figs. 3 and 4.

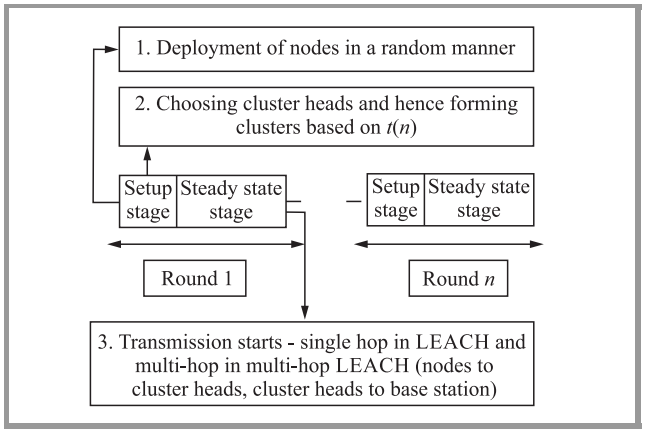


Fig. 3. Routing in LEACH protocol.

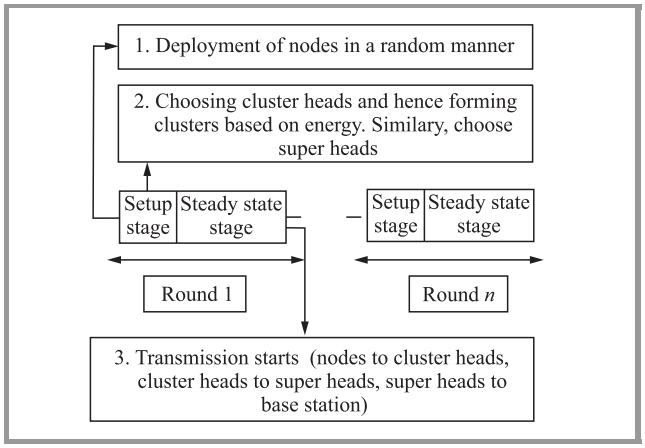


Fig. 4. Routing in SHEENA.

These approaches have been applied here to 3D UWSN. It is considered that sensor nodes are deployed at different depths in a 3D UWSN. A generic model for the same has each sensor node assigned with a triple of coordinates (x, y, z). The function (u, v) defines the distance between two nodes in a 3D Euclidean space as:

$$\delta(u,v) = \sqrt{(u_x - v_x)^2 + (u_y - v_y)^2 + (u_z - v_z)^2}\,. \qquad (2)$$

3. Energy Model for UWSNs

To transmit data from one node to another node over a distance d, the energy dissipation in underwater channel is given by [12]:

$$E(d) = E_t(d) + E_r(d)\,, \qquad (3)$$

$$E_t(d) = l\,(E_{elec} + E_{amp}) + P_t \cdot \frac{l}{h \cdot B(d)}\,, \qquad (4)$$

$$E_r(d) = l\,(E_{elec} + E_{DA}) + P_r \cdot \frac{l}{h \cdot B(d)}\,. \qquad (5)$$

Here, P_t and P_r are the transmission and reception power levels for transmission energy E_t and reception energy E_r of the network respectively, l is packet size, $B(d)$ is the bandwidth available, E_{elec} is the energy consumed by the electronics to process one bit of message, E_{amp} is the energy consumed by amplifier, E_{DA} is the energy for data aggregation. The variable h is the bandwidth efficiency of modulation (in b/s/Hz), given by:

$$h = \log_2(l + SNR)\,. \qquad (6)$$

In UWSNs, signal to noise ratio (SNR) of a transmitted signal by a node is expressed in the terms of source level (SL), transmission loss (TL), ambient noise or noise level (NL) and directivity index (DI). SNR (in dB) is expressed as [13]:

$$SNR = SL - TL - NL + DI\,. \qquad (7)$$

The SL (in dB re μPa) depends upon transmission power intensity I_t and transmission power (P_t), expressed as:

$$SL = 10 \log\left(\frac{I_t}{0.067 \cdot 10^{-18}}\right). \qquad (8)$$

Given the Transmission Power (P_t), Transmission Power Intensity (I_t) of an underwater signal at 1 m from the source can be obtained for the shallow water (in W/m^2) through the following expression:

$$I_t = \left(\frac{P_t}{2\pi \cdot 1m \cdot d}\right). \qquad (9)$$

where d is depth in meters.

Equation (9) will be varied by replacing 2π to 4π for deep-water scenarios as referred in [14].
Transmission loss (TL) is the abatement in sound intensity through the path from transmitting node to receiving node in the network [15]. It is dependent on the transmission range and attenuation. The transmission loss (in dB) is expressed as:

$$TL = SS + \alpha \cdot 10^{-3}, \qquad (10)$$

where SS is spherical spreading factor $SS = 20 \log r$, α is attenuation factor (in dB), calculated from Thorp formula as given in Eq. (11), and r is transmission range (in meters).

Attenuation occurs due to the transformation of acoustic energy into heat. Energy absorbed by the water is proportional to the frequency of the signal. The Thorp model proposed in [16] involves the simplest equation for attenua-

tion, taking into account the effect of the frequency utilized. The Thorp equation is formulated as:

$$\alpha = 0.11\frac{f^2}{1+f^2} + 44\frac{f^2}{4100+f^2} + 2.75\cdot 10^{-4} f^2 + 0.003\,, \quad (11)$$

where f is frequency in kHz.

The Directivity Index (DL) is set to zero (because we assume omnidirectional hydrophones). The Noise Level (NL), i.e. the ambient noise of underwater wireless sensor networks is expressed in terms of summation of turbulence noise, shipping noise, wave noise and thermal noise, summing up into [17]:

$$N(f) = N_t(f) + N_s(f) + N_w(f) + N_{th}(f)\,. \quad (12)$$

In the Eq. (12) the turbulence noise may be expressed as

$$10\log N_t(f) = 17 - 30\log(f)\,. \quad (13)$$

The shipping noise is calculated by:

$$10\log N_s(f) = 40 + 20(s-0.5) + 26\log(f)\,, \quad (14)$$

where s is the shipping factor, which ranges from 0 to 1 for low to high activities, respectively.

The wave noise is given by

$$10\log N_s(f) = 50 + 7.5\sqrt{w} + 20\log(f) - 40\log(f+0.4)\,, \quad (15)$$

where w is the wind speed.

The thermal noise is represented by

$$10\log N_{th}(f) = -15 + 20\log(f)\,. \quad (16)$$

In all equations for noise components f is the frequency in kHz.

4. Simulation and Analysis

During simulation in Matlab [18] the network of 100 nodes using random topology in $200 \times 200 \times 200$ m environment have been deployed. The base station is placed at (200, 200, 200).

We applied multi-hop LEACH and Sensor Hop-based Energy Efficient Networking Approach to the 3D Underwater Wireless Sensor Network. The scenario for the proposed approach is shown in Fig. 5, in which all the deployed nodes are connected to their respective cluster heads represented by the blue lines. Cluster heads are connected with each other as depicted by the green lines. Further, the cluster heads can be connected with super head as shown by red lines in the picture.

As the energy parameter depletes after some duration of time, i.e. after some number of rounds, some of the sensor nodes will have energy level much below threshold and they can be regarded as dead nodes. The aim of the proposed approach is to delay the dying of nodes by saving the energy of the network.

The simulation parameters included in the implementation are given in Table 1. Some of the values have been referred from [12].

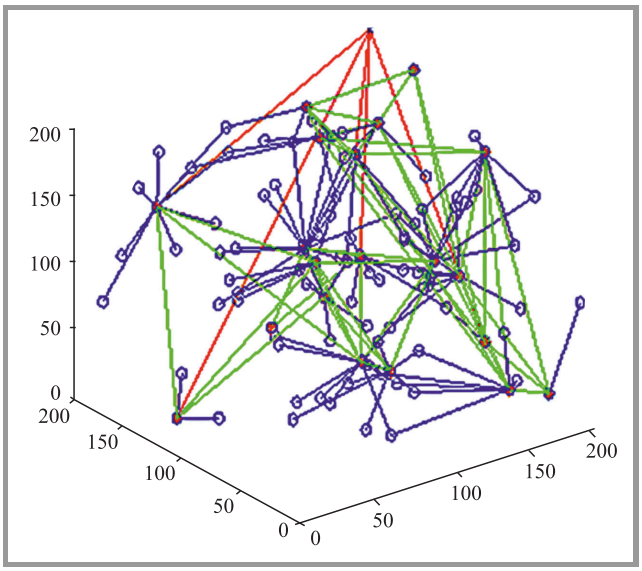


Fig. 5. Interconnection of nodes. (See color pictures online at www.nit.eu/ publications/journal-jtit)

Table 1
The simulation parameters

Parameter	Variable	Value
Network sink		$200 \times 200 \times 200$ m
Size		200, 200, 200
Number of nodes		100
Data packet size		240 bytes
Initial energy of every node	E_0	5 J
Amplifier energy	E_{amp}	0.0013 pJ/bit/m^4
Electronics energy	E_{elec}	50 nJ/bit
Energy for data aggregation	E_{DA}	5 nJ/bit
Number of simulation rounds	$r_{\max}$	6000
Bandwidth	$B(d)$	4 kHz
Frequency	f	10 kHz
Distance	d	20 m
Range	r	50 m
Transmission power	P_t	70 mW
Reception power	P_r	16 mW
Shipping factor	s	0.5
Wind speed	w	6 m/s

4.1. Energy Consumption for Shallow and Deep Water

The main parameter of analysis to be considered is the energy consumption in the network. The energy model described in Section 3 is followed to calculate consumption of energy for the network in case of both traditional multi-hop LEACH and the proposed scheme. Table 2 and Fig. 6 show the results obtained on implementation de-

picting the variation in energy consumption in the network for LEACH and SHEENA at shallow water.

Table 2
Energy consumed by the UWSN in shallow water

No.	Depth [m]	Energy consumption for LEACH [J]	Energy consumption for SHEENA [J]
1	20	32.57	18.89
2	40	37.3	17.83
3	60	32.6	16.51
4	80	31.44	16.22
5	100	29.22	18.77

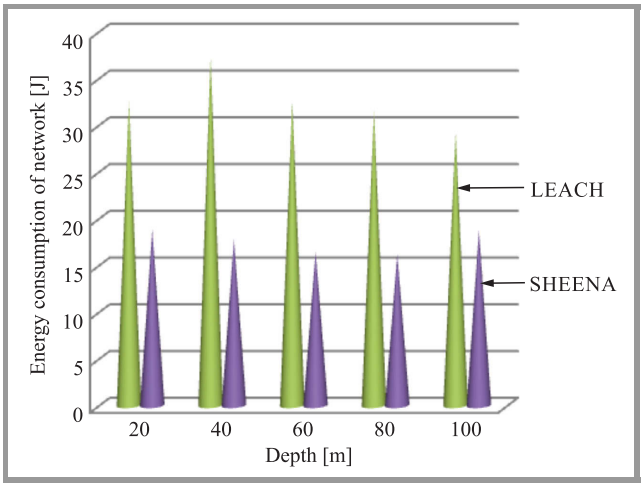


Fig. 6. Energy consumption vs. depth (shallow water).

The energy consumed by LEACH network is larger in contrast to hop-based clustering scheme, proving the proposed approach to be an energy saving one.

Table 3
Energy consumed by the UWSN in deep water

No.	Depth [m]	Energy consumption for LEACH [J]	Energy consumption for SHEENA [J]
1	500	29.4	25.02
2	2000	27.466	15.28
3	4000	26.62	13.922
4	6000	26.52	17.14
5	8000	25.05	18.851

Next, deep water is considered. Table 3 and Fig. 7 show the variation in energy consumption for LEACH and SHEENA at deep water. The proposed approach is energy efficient as compared to multi-hop LEACH even in deep water.

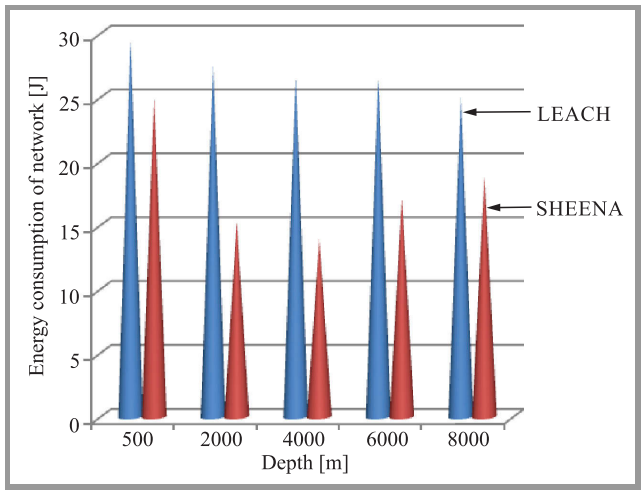


Fig. 7. Energy consumption vs. depth (deep water).

4.2. Dying of Nodes

The lifetime of a network depends upon the time when the first node of the network dies and when whole network becomes dead due to lack of energy in all the nodes.

Table 4
Dying of nodes

Depth	First node dead		Last node dead	
	Hop-based clustering scheme	LEACH multi-hop	Hop-based clustering scheme	LEACH multi-hop
Shallow water	13	9	150	100
Deep water	10	6	100	86

Table 4 shows the round number when the first and the last node becomes dead in both LEACH and SHEENA for the shallow and deep water. Figures 8 and 9 illustrate the same respectively.

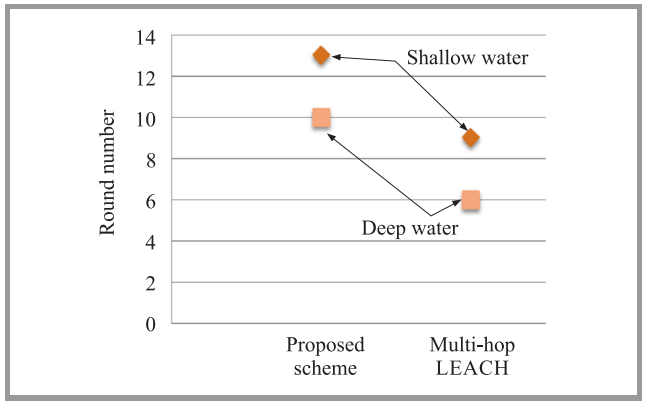


Fig. 8. Round number vs. first node dead in two approaches.

The results show that the nodes start dying later in the proposed SHEENA for both shallow and deep water,

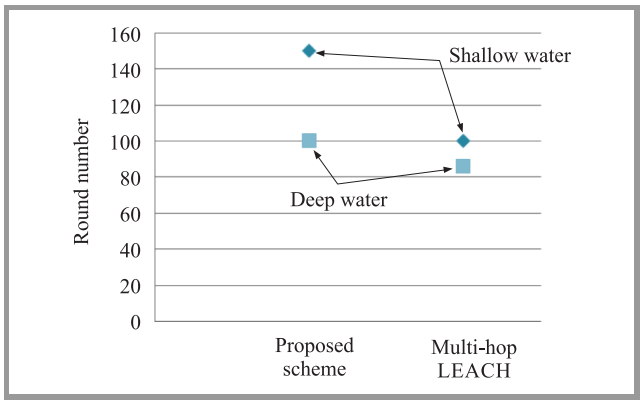


Fig. 9. Round number vs. last node dead in two approaches.

letting to increase the network lifetime in contrast to multi-hop LEACH.

5. Conclusion

The analysis of the research conducted shows that the proposed Sensor Hop-based Energy Efficient Networking Approach (SHEENA) gives better lifetime and consumes lesser energy in both shallow and deep water environments when compared with the traditional multi-hop LEACH protocol. The energy consumed in the network having Hop based clustering scheme is less than that of LEACH network. The dying of nodes is slower leading to increase the lifetime of the network in the proposed technique.

References

[1] I. F. Akyildiz, D. Pompili, and T. Melodia, "Underwater acoustic sensor networks: Research challenges", *Ad Hoc Networks*, vol. 3, no. 3, pp. 257–279, 2005.

[2] S. Kohli and P. P. Bhattacharya, "Characterization of acoustic channel for underwater wireless sensor networks", in *Proc. Ann. IEEE India Conf. INDICON 2015*, New Dehli, India, 2015 (doi: 10.1109/INDICON.2015.7443243).

[3] M. Ayaz and A. Azween, "Underwater wireless sensor networks: routing issues and future challenges", in *Proc. 7th Int. Conf. on Adv. in Mob. Comput. & Multim. MoMM 2009*, Kuala Lumpur, Malaysia, 2009, pp. 370–375 (doi: 10.1145/1821748.1821819).

[4] J. Heidemann *et al.*, "Research challenges and applications for underwater sensor networking", in *Proc. IEEE Wirel. Commun. & Netw. Conf. WCNC 2006*, Las Vegas, NV, USA, 2006, vol. 1, pp. 228-235 (doi: 10.1109/WCNC.2006.1683469).

[5] I. F. Akyildiz, D. Pompili, and T. Melodia, "Challenges for efficient communication in underwater acoustic sensor networks", *ACM Sigbed Rev.*, vol. 1, no. 2, pp. 3–8, 2004 (doi. 10.1145/1121776.1121779).

[6] H. Frey, S. Rührup, and I. Stojmenović, "Routing in wireless sensor networks", in *Guide to Wireless Sensor Networks*, S. C. Misra, I. Woungang, and S. Misra, Eds. London: Springer, 2009, pp. 81–111.

[7] C. Schurger and M. B. Srivastava, "Energy efficient routing in wireless sensor networks", in *Proc. Military Commun. Conf. MILCOM 2001. Communications for Network-Centric Operations: Creating the Information Force*, McLean, VA, USA, 2001, vol. 1 (doi: 10.1109/MILICOM.2001.985819).

[8] B. Mamalis *et al.*, "Clustering in wireless sensor networks", in *RFID and Sensor Networks: Architectures, Protocols, Security and Integrations*, Y. Zhang, L. T. Yang, J. Chen, Eds. CRC Press, 2009, pp. 324–353.

[9] M. Arioua *et al.*, "Multi-hop cluster based routing approach for wireless sensor networks", *Procedia Computer Science*, vol. 83, pp. 584–591, 2016 (doi: 10.1016/j.procs.2016.04.277).

[10] I. F. Akyildiz *et al.*, "Wireless sensor networks: A survey", *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.

[11] F. Xiangning and S. Yulin, "Improvement on LEACH protocol of wireless sensor network", in *Proc. Int. Conf. on Sensor Technol. & Appl. SensorComm 2007*, Valencia, Spain, 2007, pp. 260–264 (doi: 10.1109/SENSORCOMM.2007.60).

[12] T. V. Padmavathy, V. Gayathri, V. Indumathi, and G. Karthika, "Network lifetime extension based on network coding technique in underwater acoustic sensor networks", *Int. J. of Distrib. and Parallel Syst.*, vol. 3, no. 3, pp. 85–100, 2012 (doi: 10.5121/ijdps.2012.3309).

[13] M. Felamban, B. Shihada, and K. Jamshaid, "Optimal node placement in underwater wireless sensor networks", in *Proc. 27th IEEE Int. Conf. on Adv. Inform. Netw. & Appl. AINA 2013*, Barcelona, Spain, 2013, pp. 492–499 (doi: 101109/AINA.2013.40).

[14] M. C. Domingo and R. Prior, "Energy analysis of routing protocols for underwater wireless sensor networks", *Computer Commun.*, vol. 31, no. 6, pp. 1227–1238, 2008 (doi: 10.1016/j.comcom.2007.11.005).

[15] J. A. L. Sirvent, "Realistic acoustic prediction models to efficiently design higher layer protocols in underwater wireless sensor networks", Ph.D. Thesis, Universidad Miguel Hernández De Elche, 2012.

[16] W. H. Thorp, "Analytic description of the low-frequency attenuation coefficient", *The J. of the Acoust. Soc. of America*, vol. 42, no. 1, p. 270, 1967 (doi: 10.1121/1.1910566).

[17] J. Llor, E. Torres, P. Garrido, and M. P. Malumbres, "Analyzing the behavior of acoustic link models in underwater wireless sensor networks", in *Proc. 4th ACM Worksh. on Perform. Monit. & Measur. of Heterogen. Wirel. and Wired Netw. PM2HW2N '09*, Tenerife, Canary Islands, Spain, 2009, pp. 9–16 (doi: 10.1145/1641913.1641915).

[18] D. J. Higham and N. J. Higham, *MATLAB Guide*. Siam, 2005.



Sheena Kohli is currently working as Assistant Professor with Department of Computer Science and Engineering in College of Engineering and Technology at Mody University of Science and Technology, Lakshmangarh, Rajasthan, India. She has received her B.Tech. degree in Information Technology from Rajasthan Technical University in 2010. She completed her M.Tech. in Information Technology from Banasthali University, Rajasthan, India, in 2012. Her research interests include wireless sensor networks and underwater acoustic sensor networks.

E-mail: sheena7kohli@gmail.com
Department of Electronics and Communication Engineering
College of Engineering and Technology
Mody University of Science and Technology
Lakshmangarh – 332311, Rajasthan, India

Partha Pratim Bhattacharya – for biography, see this issue, p. 26.

Paper

An Improved Greedy Forwarding Scheme in MANETs

Priya Mishra[1], Charu Gandhi[1], and Buddha Singh[2]

[1] *Department of Computer Science and Information Technology, JIIT, Noida, U.P., India*
[2] *School of Computer and Systems Science, JNU, Delhi, India*

Abstract—Position-based routing protocols are widely accepted efficient solution for routing in MANETs. The main feature of position-based routing protocols is to use greedy forwarding methods to route data. The greedy forwarding methods select a node, either having maximum progress towards destination (distance-based principle) or minimum deviation with line between source and destination (direction-based strategy). The first method minimizes the hopcount in a path and on the other hand, second method minimizes the spatial distance between source and destination. The distance-based routing has a great impact on the selection of reliable node and the direction based routing plays a major role to increase the stability of route towards destination. Therefore, in this paper authors propose a weighted forwarding method, which combines both the selection, schemes to select an optimal next forwarding node in a range. The simulation results show that the proposed scheme performs better than existing position-based routing protocols.

Keywords—MANET, distance-based routing, direction-based routing, greedy forwarding.

1. Introduction

A Mobile Ad Hoc Network (MANET) is an infrastructureless network with nodes. They change their topology dynamically and work as a host as well as routers at the same time in the network. Therefore, they perform an important role to route data. If source and destination are in transmission range of each other, they can transmit data directly. However, if source and destination are out of transmission range of each other, they have to be dependent on other intermediate nodes to forward message to the destination. Since, the mobile nodes move in any direction, which causes frequent linkages formation and breakage. In such scenarios, the traditional routing protocols [1], [2] are not an efficient choice for routing in MANETs.

To overcome these issues, position-based routing protocols are accepted a better solution for routing in MANETs. The main feature is to use location information of neighbors and destination to route data and further use greedy forwarding mechanisms to forward a message to the neighbor closest to the destination. Greedy forwarding strategies use distance or direction of nodes as a parameter for the selection of next forwarding node to route data. The first scheme selects a neighbor with the largest distance towards the destination to minimize the hop count. On the other hand, the second method favors a neighbor with the lowest angle deviation toward the destination to minimize the spatial distance between nodes.

The distance-based routing has a great impact on the selection of reliable node and the direction-based routing plays a major role to increase the stability of route towards destination. This shows that these schemes give a suboptimal solution for the selection of next forwarding node. Therefore, this paper focuses to propose a combined forwarding scheme, which considers both the methods while selecting a next forwarding node to achieve better performance in terms of routing overhead, hop count and end-to-end delay over distance-based and direction-based forwarding schemes. Further, to combine these routing schemes, a weighted factor (denoted by α) is introduced, which helps to apply both the schemes in a flexible manner.

The rest of this paper is organized as follows. Section 2 presents some geographical routing schemes proposed in the literature for mobile ad hoc networks. Section 3 outlines and discusses the key features of proposed protocol. The comparisons of simulation results are presented in Section 4. Conclusion and future researches directions are discussed in Section 5.

2. Related Work

In literature a variety of position-based schemes has been proposed as an efficient and scalable solution for routing in MANETs. The position-based routings use greedy schemes based on local forwarding decisions to construct a path dynamically from source to destination. These forwarding schemes are categorized as distance-based and direction-based routings.

The MFR is proposed by [3] as a distance-based greedy routing algorithm. MFR helps to reduce the path length by selecting the next forwarding node largest progress (closer) towards the destination. This method is loop-free and finds a short path but it does not guarantee to find a path from source to the destination. Another issue with MFR routing is high packet drop rate. An improved version of MFR scheme to achieve guarantee delivery and eliminate looping problem has been proposed named as F-MFR protocol [4]. The other latest improvements of MFR are discussed in [5] and [6].

The first direction based routing has been proposed in [7] and is called as Compass routing (DIR), which selects the neighbor having the minimum deviation from Line of Sight (LOS). This feature of protocol results to find a most direct path to route data. The protocol successfully progress around a boundary, which can help in a higher rate of

path completion but the protocol is not loop free and suffers from congestion produced by frequent beaconing messages. Another variant of Compass routing, named Random Compass [4] has been proposed to select the next hop randomly between the two nodes on the closest angle to the destination.

Q-DIR [8] – directional routing with restricted flooding protocol is proposed to restrict the broadcast area to all nodes in the same quadrant as the source and destination. To overcome the limitation of basic greedy forwarding schemes combined Greedy-Compass [9], [10] has also proposed to improve the performance of forwarding schemes to combine the basic greedy forwarding schemes, in which selects one of the two nodes, which is at the minimum distance from the destination.

Another variant of greedy routing protocol is GEDIR [11], proposed by Liao *et al*. to eliminate the loops during routing and makes it loop free. In literature, hybrid schemes are also proposed, which consider both greedy routings protocol while selecting a next forwarding node. Angular Routing Protocol (ARP) [12] is a position-based routing protocol proposed to forward data from source and destination. The protocol starts data forwarding by using greedy forwarding, if greedy forwarding fails, the protocol switches to angle-based forwarding to avoid voids in sparse networks.

A hybrid-weighted forwarding scheme named (HGR) is proposed by Chen *et al*. [13], which combines distance and direction metrics in a flexible manner. The protocol helps to tradeoff between energy usages and end-to-end delay during the routing procedure. Further, the authors proposed a dynamic variant of HGR (DHGR) mechanisms based on the basic HGR scheme. These schemes aim to define the balance between end-to-end delay and energy consumption. The protocol reduces the energy consumption during finding the path.

An Improved Progress Position Based Beacon Less Routing algorithm (I-PBBLR) [14] considers the progress metric with the direction to select the next forwarding node for improved and efficient routing between source and destination. This routing protocol guarantees loop free forwarding closer to the destination.

A few other hybrid-based combined forwarding schemes are also proposed in [15], [16].

3. Proposed Work

The used model utilizes the mobile ad hoc network as a set of nodes deployed in a two-dimensional area, where each node has unique position. The model uses the location information of nodes and makes some assumptions:

- the node can obtain the location information through the support of GPS devices;
- the source is aware of its own location and location information of destination;
- the intermediate nodes are aware of their own positions. When source S wants to send data to destination D, S utilizes the known location information of destination;
- each node has same transmission range and moving with same speed.

3.1. Description of Proposed Protocol

Generally, the position-based routing protocols select the next hop either applying the distance-based (MFR), or as direction-based strategy (Compass). The distance-based strategy tries to select a neighbor closer to the destination to minimize the hopcount and the direction-based selects a neighbor, which makes lowest angle from LOS toward the destination to minimize the spatial distance. The distance based routing has a great impact on the selection of reliable node and the direction based routing plays a major role to increase the stability of route towards destination. The selection of next forwarding node based on basic greedy principles is given in Fig. 1.

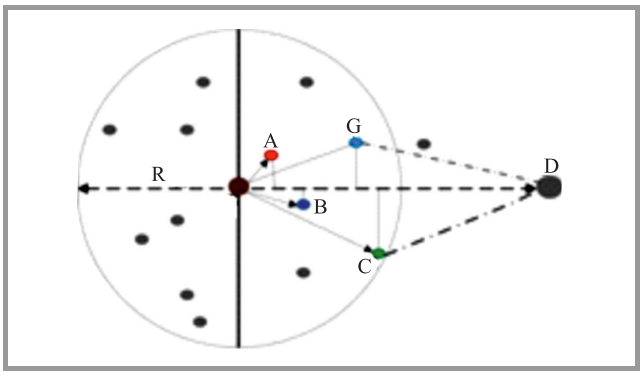


Fig. 1. Greedy forwarding methods: S and D are source and destination nodes. The S has different choices to find a next forwarding node; A = nearest with forwarding progress; C = most forwarding progress within Radius (MFR); B = Compass routing; G = greedy.

These features motivate to combine both distance and direction metrics while choosing next hop to forward message from source to destination. In this work, a joint forwarding scheme is proposed, which is blessed with both the metrics. To combine these metrics, a weighted scheme is defined and calculated. The node having the highest score is selected as the next forwarding node within the transmission range of the source node towards the destination. The formula to calculate the weights is given below:

$$w = (1-\alpha)\left(1-\frac{x}{\mathrm{R}}\right)+\alpha\left(1-\frac{\theta}{90^{\circ}}\right), \qquad (1)$$

where w is the weighted score of a node to become a next forwarding node, α is the adjustment factor to combine these metrics, x is the distance from node i to destination D, R is the transmission range, and $\frac{x}{\mathrm{R}}$ is the closeness of next candidate hop. Angle θ is the deviation of

node i from the straight line between source S to destination D. In Fig. 2, x and θ denote the projected progress and the deviation angle of node i respectively. When source S wants to send data to destination D, it utilizes the location information and tries to find next forwarding node to forward the data to destination. As an intermediate node receives a data packet, it calculates the distances and deviations of nodes by using its angle formula. The distance between the source S and destination D is denoted by h, l and x are the distances between node i and destination D and node S and node i respectively. These calculations are given in Eqs. (2)–(4). The deviation angle θ is calculated in Eq. (5).

$$h = \sqrt{(X_S - X_D)^2 + (Y_S - Y_D)^2}, \tag{2}$$

$$x = \sqrt{(X_S - X_i)^2 + (Y_S - Y_i)^2}, \tag{3}$$

$$l = \sqrt{(X_i - X_D)^2 + (Y_i - Y_D)^2}, \tag{4}$$

$$\theta = \frac{h}{2x\cos(x^2 + h^2 - l^2)}. \tag{5}$$

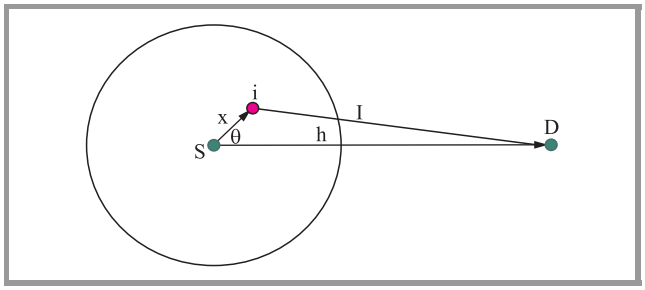


Fig. 2. Calculation of distance and direction of node.

To enhance the capability of traditional greedy forwarding, presented approach considers nodes in any potential regions defined below to perform forwarding decision. In general, whenever a source S attempts to forward a message, it will first search the potential region and pick the node in that region that is the closest to the destination. There might be several nodes within these defined regions, therefore, to select the best forwarding node, protocol calculates the progress and deviation of each node within the region. These values are further used to determine the value of weight score (w) for selecting a next hop to forward data. The node with highest weighted score among all the nodes will be chosen as next hop. Secondly, by adjusting α, distance- and direction-based forwarding schemes could be balanced. There are two scenarios:

1. If $\alpha = 0$, protocol behaves like a pure distance-based forwarding when nodes are reside in region A.
2. Distance-based routing fails, if there is no node in region A, in this situation the value of α lies at $0 \leq \alpha \leq 1$. The node switches to hybrid mode and the distance and angular deviation of each node are calculated. These values are combined by using the weighted factor.

In simulation, we set the weights according to the distance and direction of node. If value $\frac{x}{R} < \frac{\theta}{90^\circ}$, the value of α goes to upper side that is 1 means more weightage to direction metric than distance metric. Else, hybrid forwarding approach with more priority to distance based forwarding.

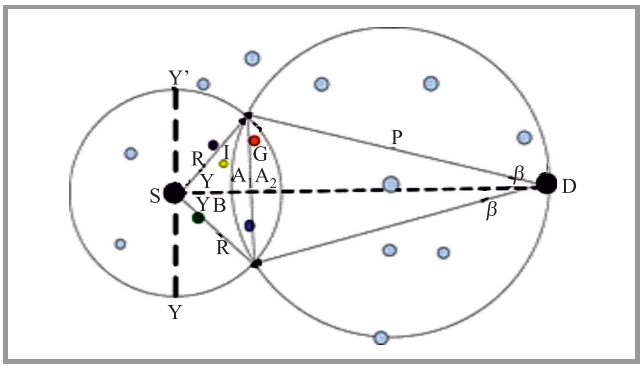


Fig. 3. Next node selection method.

In Fig. 3, the selection criterion for next node is presented. In this example, forwarding area is divided into sub regions border region (A) and reminder region (B). These areas are potential regions for distance and combined routings respectively. To define the border area, we first draw tangent from the point D (center of destination) to the circle with radius R around the source S. Draw a circle having radius P and they are shown by blue shaded area in Figs. 2 and 3. This area (A1+A2) [17] can also be called as area of interaction of the two circles and calculated by Eqs. (6)–(7).

$$\mathrm{A1} = \mathrm{R}^2\gamma - \frac{\mathrm{R}^2 \sin 2\gamma}{2}, \tag{6}$$

$$\mathrm{A2} = \mathrm{P}^2\gamma - \frac{\mathrm{P}^2 \sin 2\gamma}{2}. \tag{7}$$

The distance-based principle chooses a next node in area A, because area A lies closest to borderline of the sender's transmission range. Greedy routing is the appropriate method to selects a next-hop node for the given region, it has been analytically proved in [18]. Further, the protocol defines the potential regions (reminder region) for combined forwarding scheme. This reminder region is represented by B and Eq. (8).

$$\mathrm{B} = \mathrm{area}(\mathrm{SD} - \mathrm{A1} + \mathrm{A2}) \tag{8}$$

where

$$\mathrm{B} = \frac{\pi r^2}{2} - \mathrm{R}^2\gamma - \frac{\mathrm{R}^2 \sin 2\gamma}{2} + \mathrm{P}^2\beta - \frac{\mathrm{P}^2 \sin 2\beta}{2}.$$

In this region, both the progress (distance) and deviation of nodes are compared by applying the weighted methods and adjustment criterion given in scenario 2.

4. Simulation and Performance Analysis

To evaluate the performance of the proposed protocol, implementation is carried out in Matlab 7.0. The simulation results are compared with distance-based routing (MFR), and direction-based routing (Compass). The simulation setup is given with parameters is shown in Table 1.

Table 1
Simulation parameters

Simulation parameter	Value
Topology size	1000 · 1000 m
Number of nodes	20–100
Speeds	5–25 m/s
Mobility model	Random way-point
Simulation time	200 s
Channel rate	2 Mb/s
MAC layer protocol	IEEE 802.11b
Radio propagation model	Two ray-ground
Transmission range	200 m
Traffic type	CBR

To analyze the performance of proposed protocol is compared by using below discussed metrics. Node density and mobility are the important factors that affect the performance of routing protocols.

Routing overhead. Routing overhead is an important measure for the scalability of routing protocols. It is a metric to determine the efficiency of the routing protocol and calculated as number/size of routing control packets sent by the protocol.

Hop count. The number of nodes encounter to the path from source to destination. The path length is directly related to number of hops in the path.

End-to-end delay. The ratio of the packets that successfully reach destination to the original sent ones.

4.1. The Impact of Varying Number of Nodes

This section presents the simulation results of proposed protocol at varying number of nodes from 20 to 100 while the speed is fixed at 10 m/s.

The simulation results in Fig. 4 show that the routing overhead increases linearly for all the protocols on increasing the nodes in network. The proposed scheme produces less routing overhead in comparison to basic distance-based greedy forwarding schemes and higher routing overhead than direction-based routing. The reason behind these results is that the distance-based routing select the node closer to the destination but it may fail, if it has no neighbors closer to the destination. Then they do re-routing to find the path and on contrast, direction based routing algorithms has lowest routing overhead because it chose the node, which is less deviated from LOS without considering the progress. The proposed protocol has lesser routing overhead than distance-based routing and higher than direction-based routing protocol.

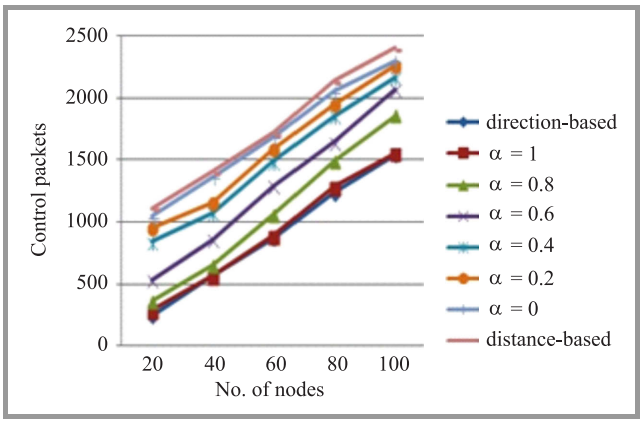


Fig. 4. Routing overhead at speed 10 m/s. (See color pictures online at www.nit.eu/publications/ journal-jtit)

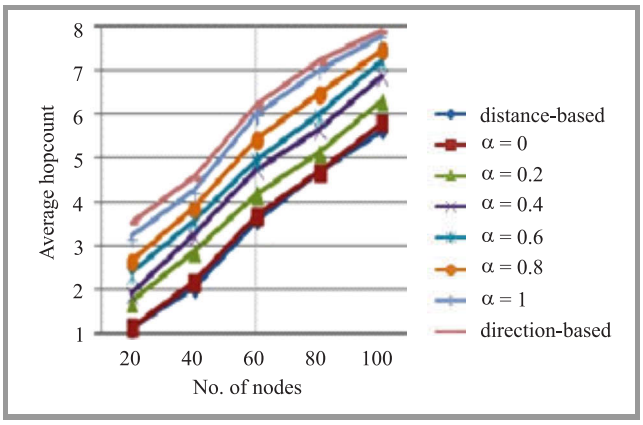


Fig. 5. Average hopcount at speed 10 m/s.

Figure 5 illustrates the number of hops vs. the varying number of nodes and the hopcount of all the protocols increases with the number of nodes. The results show that the hopcount of the proposed protocol is lower than direction-aware routing and higher than distance-aware routing protocol.

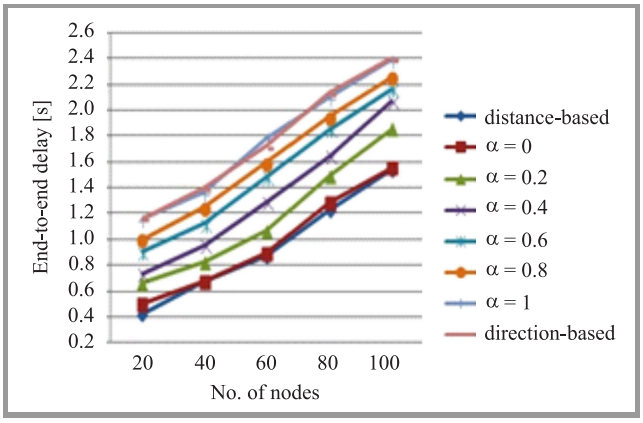


Fig. 6. End-to end delay at speed 10 m/s.

In Fig. 6 the simulation results of end-to-end delay vs. speeds are presented. The result reveals that the delays of all the schemes are inversely related to speed values. The angular scheme shows the highest delays in comparison to other protocols. The reason behind is that it only focuses the directions (deviation from the LOS) while ignoring progress towards destination and selects a path with larger numbers of hops and, thus, longer delays. On the other hand, distance-based routing has the smallest delay and proposed protocol has larger delay than distance-based routing and smaller delays in comparison with direction-based routing protocols. The delay of all the protocols increases on increasing the number of nodes.

The performance of proposed protocol also depends on weight factor α. If α is larger, proposed protocol behaves more like distance-based routing and on the other hand if value is small, protocol behaves like direction-based routing. The weight factor used in proposed protocol also affects the performance of proposed protocol.

4.2. The Impact of Node Speeds

In this simulation the speed is varied from 5 to 25 m/s while the number of nodes are fixed at 40.

The routing overheads of all the protocols and comparison study is given in Fig. 7. The results show that the routing overhead of proposed protocol depends on α. When α is high, protocol produces less overhead as direction-based. As the value of α decrease, the protocol produces more routing overhead and behaves as a distance-based.

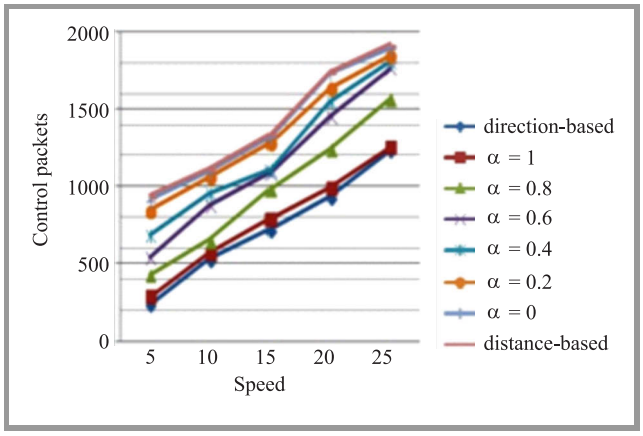


Fig. 7. Routing overhead with 40 nodes.

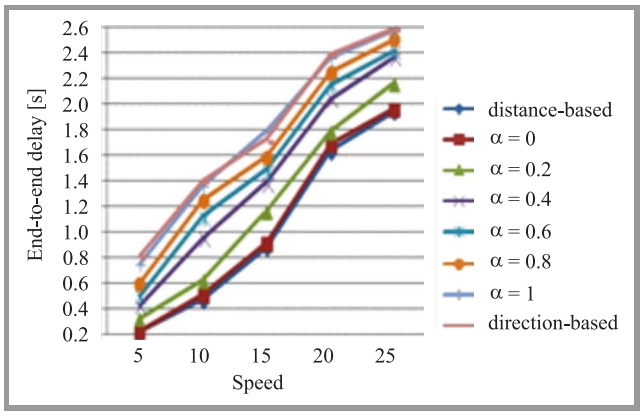


Fig. 8. End-to-end delay for 40 nodes.

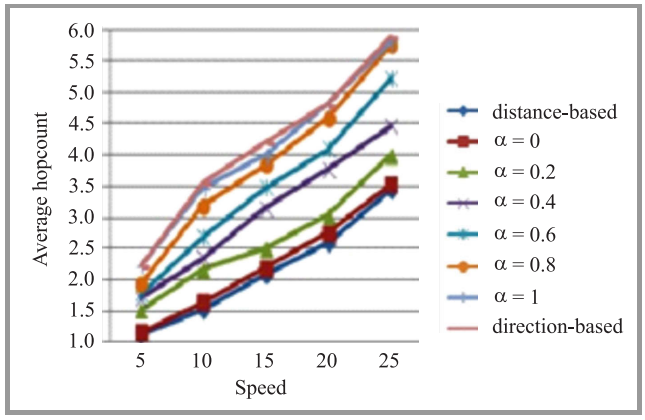


Fig. 9. Average hopcount for 40 nodes network.

In Fig. 8 the simulation results of end-to-end delay vs. speeds are given and result reveals that the delays of all the schemes decrease as higher speeds. The direction-based scheme has the highest delays because it only focuses the directions (deviation from the LOS) while ignoring progress towards destination and selects a path with larger numbers of hops and, thus, longer delays. On the other hand, distance-based routing has the smallest delay in comparison with the proposed protocol. In simulation, the end-to-end delay of proposed protocol will depend on α. If α is closer to 1 then end-to-end delay of proposed protocol is more like distance-based routing. Figure 9 illustrates the number of hops vs. the varying node's speeds and the results show that the hopcount of the proposed protocol is lower than direction-based routing and higher than distance-based routing protocol.

5. Conclusion

The simulation results show the proposed protocol outperforms than distance-based and direction-based routing significantly in the terms of average hopcount, end-to-end delay and routing overhead. The protocol increases the quality of route in terms of both stability and reliability over conventional distance and direction-based algorithms if they are used separately.

References

[1] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc on-demand distance vector (AODV) routing", RFC 3561, IETF, 2003 [Online]. Available: http://www.ietf.org/rfc/rfc3561.txt

[2] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks", in *Mobile Computing*, T. Imielinski and H. F. Korth, Eds. *The Kluwer International Series in Engineering and Computer Science*, vol. 353, pp. 153–181. Springer, 1996.

[3] H. Takagi and L. Klienrock, "Optimal transmission ranges for randomly distributed packet radio terminals", *IEEE Trans. on Commun.*, vol. Com-32, no. 3, pp. 246–257, 1984 (doi: 10.1109/TCOM.1984.1096061).

[4] I. Stojmenovic and X. Lin, "Loop-free hybrid single-path/flooding routing algorithms with guaranteed delivery for wireless networks", *IEEE Trans. on Parallel and Distrib. Syst.*, vol. 12, no. 10, pp. 1023–1032, 2001.

[5] R. S. Raw and D. K. Lobiyal, " B-MFR routing protocol for vehicular ad hoc networks", in *Proc. Int. Conf. on Netw. and Inform. Technol. ICNIT 2010*, Manila, Philippines, 2010, pp. 420–423.

[6] V. Raji and N. Mohan Kumar, "Void aware position based opportunistic routing for QoS in mobile ad hoc networks", *Circuits and Syst.*, vol. 7, pp. 1504–1521, 2016 (doi: 10.4236/cs.2016.78132).

[7] E. Kranakis, H. Singh, and J. Urrutia, "Compass routing on geometric networks", in *Proc. 11th Canadian Conf. on Comput. Geometry CCCG 1999*, Vancouver, Canada, 1999, pp. 51–54.

[8] L. A. Latiff, N. Fisal, S. A. Arifin, and A. A. Ahmed, "Directional routing protocol in wireless mobile ad hoc network", in *Trends in Telecommunications Technologies*, C. J. Bouras. Ed. InTech, 2010 (doi: 10.5772/8486).

[9] P. Bose and P. Morin, "Online routing in triangulations", in *Proc. 10th Ann. Int. Symp. Algorithms & Computation ISAAC 1999*, Chennai, India, 2010, pp. 113–122.

[10] P. R. Morin, "On line routing in geometric graphs", Ph.D. Thesis, School of Computer Science, Carleton University, 2001.

[11] W.-H. Liao, J.-P. Sheu, and Y.-C. Tseng, "GRID: A fully location-aware routing protocol for mobile ad hoc networks", *Telecommun. Syst.*, vol. 18, no. 1, pp. 37–60, 2001 (doi: 10.1023/A:1016735301732).

[12] V. Giruka and M. Singhal, "Angular routing protocol for mobile ad-hoc networks", in *Proc. 25th IEEE Int. Conf. on Distrib. Comput. Syst. Worksh. ICDCSW'05*, Columbus, OH, USA, 2005, pp. 551–557 (doi: 10.1109/ICDCSW.2005.42).

[13] M. Chen, V. Leung, S. Mao, Y. Xiao, and I. Chlamtac, "Hybrid geographic routing for flexible energy-delay tradeoff", *IEEE Trans. on Veh. Technol.*, vol. 58, no. 9, pp. 4976–4988, 2009.

[14] Y. Cao and S. Xie, "A position based beaconless routing algorithm for mobile ad hoc networks", in *Proc. Int. Conf. on Commun., Circ. & Syst. ICCCAS 2005*, Hong Kong, China, 2005, vol. 1, pp. 303–307.

[15] K. Z. Ghafoor *et al*., "Fuzzy logic-assisted geographical routing over vehicular ad hoc networks", *Int. J. of Innov. Comput., Inform. & Control*, vol. 8, no. 7, pp. 5095–5120, 2012.

[16] P. Mishra, S. K. Raina, and B. Singh, "Effective fuzzy-based location-aware routing with adjusting transmission range in MANET", *Int. J. of Syst., Control & Commun.*, vol. 7, no. 4, pp. 360–379, 2016 (doi: 10.1504/IJSCC.2016.079432).

[17] R. S. Raw, Vikas Toor, and N. Singh, "Estimation and analysis of path duration in vehicular ad hoc networks using position-based routing protocol", *Special Issue of Int. J. of Comp. Appl. on Int. Conf. on Issues and Challenges in Networking, Intelligence and Computing Technol. ICNICT 2012*, no. 3, pp. 39–39, 2012.

[18] J. RejinaParvin and C. Vasanthanayaki, "Particle swarm optimization-based clustering by preventing residual nodes in wireless sensor networks", *IEEE Sensors J.*, vol. 15, no. 8, pp. 4264–4274, 2015 (doi: 10.1109/JSEN.2015.2416208).



Priya Mishra received her Master of Computer Application (MCA) degree from Rani Durgawati Vishwavidyalaya, Jabalpur, India. She received her M.Tech. degree in Computer Science and Engineering from Uttar Pradesh Technical University, India. She has total 13 years teaching experience and working as a Guest Faculty in Gautam Buddha University, Gr. Noida in ICT department. She is Ph.D. candidate in the field of mobile ad hoc network in JIIT University Noida, India. Her research interests are mobile ad hoc network, mobile computing, and soft computing.
E-mail: amipriya@gmail.com
Department of Computer Science
and Information Technology
JAYPEE Institute of Information Technology, University
A-10 Sector 62
Noida, India



Charu Gandhi received her B.Sc. degree in Computer Science and Engineering, from Kurukshetra University, Kurukshetra, India. She received M.Tech. from Banasthali Vidyapith, Rajasthan and Ph.D. degree in Computer Science from Kurukshetra University, Kurukshetra. She has total 12 years experience in teaching and research. She is working as an Associate Professor in computer science department in JIIT University Noida, India. Her expert areas are mobile ad hoc networks and wireless sensor networks.
E-mail: charu.kumar.jiit@jiit.ac.in
Department of Computer Science
and Information Technology
JAYPEE Institute of Information Technology, University
sector 128
Noida, India



Buddha Singh received his B.Sc. degree in Information Technology from Madhav Institute of Technology and Science, Gwalior, India. He received his M.Tech. and Ph.D. degree in Computer Science and Technology from Jawaharlal Nehru University, New Delhi, India. He is working as an Assistant Professor in school of Computer and System Sciences in Jawaharlal Nehru University, New Delhi, India. His research areas of interest are mobile ad hoc networks, wireless sensor networks, cognitive radio big data analytics, complex networks, mobile computing.
E-mail: b.singh.jnu@gmail.com
Department of Computer Science
and Information Technology
School of Computer and System Sciences
in Jawaharlal Nehru University
New Delhi, India

Paper

Energy Efficient Scheduling Methods for Computational Grids and Clouds

Agnieszka Jakóbik[1], Daniel Grzonka[1], Joanna Kołodziej[1], Adriana E. Chis[2], and Horacio González-Vélez[2]

[1] *Tadeusz Kościuszko Cracow University of Technology, Cracow, Poland*
[2] *Cloud Competency Centre, National College of Ireland, Dublin, Ireland*

Abstract—This paper presents an overview of techniques developed to improve energy efficiency of grid and cloud computing. Power consumption models and energy usage profiles are presented together with energy efficiency measuring methods. Modeling of computing dynamics is discussed from the viewpoint of system identification theory, indicating basic experiment design problems and challenges. Novel approaches to cluster and network-wide energy usage optimization are surveyed, including multi-level power and software control systems, energy-aware task scheduling, resource allocation algorithms and frameworks for backbone networks management. Software-development techniques and tools are also presented as a new promising way to reduce power consumption at the computing node level. Finally, energy-aware control mechanisms are presented. In addition, this paper introduces the example of batch scheduler based on ETC matrix approach.

Keywords—batch scheduling, cloud computing, energy efficient, grids, power consumption, resource allocation.

1. Introduction

In the context of a continuous increase in the demand for computing resources, the resource allocation solutions should aim not only to allocate computing resources so that they offer satisfactory service level agreements (SLAs) but also to consume the energy in an efficient way. Therefore, efficient energy-aware scheduling and resource allocation techniques are very important.

The reduction of energy consumption is one of the major challenges arising with development of grid and cloud computing infrastructures. To meet the ever-increasing demand for computing power, recent research efforts have been taking holistic views to energy-aware design of hardware, middleware and data processing applications. Indeed, advances in hardware layer development require immediate improvements in the design of system control software. For this to be possible new power management capabilities of hardware layer, need to be exposed in the form of flexible Application Program Interfaces (APIs). Consequently, novel APIs for clouds and cluster management allow for system-wide regulation as far as energy consumption. They are capable of collecting and processing detailed performance measurements, and taking real-time coordinated actions across the infrastructure.

The paper is organized as follows. Section 2 presents techniques for power consumption measurement. Section 3 is the overview of resource allocation, tasks scheduling and load balancing methods for grid and clouds considering energy expenditure control. In Section 4 the example of Expected Time to Compute (ECT) matrix scheduling process for chosen Amazon Cloud instances and its impact on the energy consumed by this environment is described. Section 5 presents a short summary of the methods presented in the paper.

2. Power Consumption Measurement and Control

In this section we present approaches for measuring, estimating, and modeling the power consumption of computing resources. The power consumption is given by the aggregated power consumed by CPU, disk, memory, network and cooling system [1], [2].

Fan *et al*. [3] investigate the power provisioning for a datacenter, and find that the actual peak power is less than 60% of the total power budget. The research shows that the CPU and the memory are the main contributors to the peak power, followed by the disk. The authors propose a model for estimating the power usage of a server based on a linear relationship between the power consumption and CPU utilization, namely they take into account the power of busy and idle servers. The evaluation shows that the model approximates the total power usage. However, for each category of servers a calibration is needed to obtain the power usage model. In addition, two techniques are presented for saving power: Dynamic Voltage/Frequency Scaling (DVFS) and improving the efficiency of non-peak power as the idle power is never lower than 50% of the peak power.

It is worth mentioning that often the power consumed by an idle machine is high, over 50% and up to 70% of the peak power consumed [3]. Therefore, to reduce the power consumption a number of approaches relying on switching idle nodes off [4] or to sleep [5] are used.

Nathuji and Schwan introduce VirtualPower [6], a system for online power management for virtualized data centers. This is a novel approach which enables virtual machines

(VMs) to have access to "soft" power states and VM specific management policies with the aim of reducing the power consumption.

Kusik *et al*. [7] propose a dynamic resource-provisioning framework for virtualized computing environments. Their approach is formulated as a sequential optimization, which employs limited lookahead control to decide the number and characteristics of the allocated resources. The goal of the research is to maximize the revenue corresponding to the provided resources by reducing the power consumption and minimizing the number of SLA violations.

Dhiman *et al*. [8] propose a system for dynamic power prediction in virtualized environments. The authors highlight that the power consumption is different for each VM and depends on the type of workload and the different characteristics of each VM and physical machine. Based on this insight they propose a solution to predict the active power usage (i.e. the power used due to the execution of a workload) at both physical machine and VMs. The prediction uses a Gaussian mixture model based predictor to estimate the power consumption based on the architectural metrics of the physical machine and its VMs. The implementation and evaluation of the proposed solution shows that the average prediction error for the power consumption is lower than 10%.

3. Energy Efficient Task Scheduling and Load Balancing

The problem of efficient task scheduling and balancing of loads over computational nodes remains challenging in the massive, extremely dynamic, elastic, diverse and heterogeneous computational environments such as computational clouds. The main issue is to distribute workloads and perform the tasks on appropriate resources in order to optimize selected objectives.

Task scheduling and workloads balancing are strongly connected with resource allocation problem. This issue becomes even more complex when energy utilization, beyond the most common optimization criteria, is treated as additional scheduling objective.

This Section highlights the most recent research in the energy efficient task scheduling and load balancing in cloud-based environments. In addition, energy-aware resource allocation approaches are also discussed.

3.1. Energy-aware Resource Allocation Heuristics Models

Resource allocation is the key issue in every distributed virtual environment. Especially energy-aware optimization is very important. There are several approaches successfully dealing with this problem. A conceptual taxonomy on energy efficient resource allocation techniques for cloud computing systems is presented in [9]. The authors define the following instances of the problem.

Resource allocation adaptation policy. An energy-aware resource allocator is reacting and adapting to changing or uncertain cloud environment. Three categories – predictive, reactive, and hybrid – are considered.

- **Predictive resource allocation adaptation policy**. Knowledge-driven machine learning techniques are used. The aim is to dynamically anticipate and capture the relationship between users QoS targets, assumed energy efficiency objective function, and given hardware resources. The knowledge about system behavior must be recorded by the monitoring service, running continuously. Resource usage planning is done before task and jobs are performed. Several machine-learning techniques such as neural networks, genetic algorithms, or reinforcement learning [10] are used.
- **Reactive resource allocation adaptation policy**. These techniques are based on monitoring of the state of a system and detecting predefined corrective actions when the negative specified event occurs. They led to the increasing of the system energy cost. The efficiency of reactive allocation depends on the ability to detect fluctuations. This approach is computationally appealing because no extensive model of the system is necessary.
- **Hybrid resource allocation adaptation policy**. This model combines predictive with reactive allocation techniques. Predictive allocation resources is performed before the processing the work. When the system is operating the reactive allocation is switched on when the monitoring system detected abnormality.

Objective function based scheduling and resource allocation. This methodology assumes finding the mathematical expression (cost function) according to the system constraints that should be minimized by numerical methods. The value of the cost function corresponds to cost of the energy utilization.

Two main closely related characteristics of cloud system might be taken into consideration during "green" scheduler constructing:

- **power-aware methods**, aiming on reducing power dissipation, power consumption, and energy cost.
- **thermal-aware methods**, targeting on reducing the thermal effects, lowering the temperature in the location of the system hardware and increasing the energy and cost to cool down the system.

3.2. Task Scheduling and Load Balancing Problem Formulation

Task scheduling is one of the most crucial issue in cloud processing. Effective scheduling approach should guarantee users' requirements and efficient resources utilization. To ensure the last one, the balancing of task loads is used.

Load balancing helps to distribute large processing load among the computing nodes. This approach has a number of goals, i.e. [11]:

- proper resources utilization,
- fair allocation of computing resources,
- support for scalability and stability of the environment,
- avoiding network and computing bottlenecks,
- extend the life of hardware resources.

We can divide load-balancing approaches into two categories: static (divides the traffic equivalently between all nodes) and dynamic (divides the traffic depending on the current state of the environment). The dynamic balancing considers two approaches: centralized, where only one node manages and distributes the whole load, and distributed – each node independently builds its own local load vector and makes all decisions [11].
In the general case, the balancing of task loads is achieved through task scheduling. The goal of this issue is to distribute workloads and perform the tasks on appropriate machines that optimize selected objectives. The problem of task scheduling in computational clouds can be reduced to the mapping tasks on individual virtual machines. Schedule can be represented by the vectors of virtual machines or tasks labels. Two different encoding methods of schedules are defined [12]:

- direct representation:

  Definition 1: Let us denote by $\mathscr{S}$ the set of all permutations **with repetition** of the length n over the set of machine labels M_l. An element $s \in \mathscr{S}$ is termed a schedule and it is encoded by the vector:

  $$s = [i_1, \ldots, i_n]^T, \tag{1}$$

  where $i_j \in M_l$ denotes the number of machine on which the task labeled by j is executed.

- permutation-based representation:

  Definition 2: Let us denote by $\mathscr{S}_{(1)}$ the set of all permutations **without repetitions** of the length n over the set of task labels N_l. A permutation $u \in \mathscr{S}_{(1)}$ is called a permutation-based representation of a schedule in CG and can be defined by the vector:

  $$u = [u_1, \ldots, u_n]^T, \tag{2}$$

  where $u_i \in N_l$, $i = 1, \ldots, n$. The cardinality of $\mathscr{S}_{(1)}$ is $n!$.

Based on the scheduling terminology introduced in [13] and [14] researchers adopted model in the form: $A|B|C$, where A specifies the resource layer and architecture type, B specifies the processing characteristics and the constraints, and C specifies the scheduling criteria. Formally, the model can be defined as follows:

$$Rm\big[\{(batch/on\text{-}line), \ldots \tag{3}$$

$$(indep/dep/wf), (stat/dyn), (dist/centr)\}\big](obj), \tag{4}$$

where:

- Rm – tasks are send into parallel resources of various computing capabilities,
- $batch/on\text{-}line$ – the task processing mode is batch mode or on-line,
- *indep/dep/wf* – independency/dependency/workflow as the task interrelation,
- $stat/dyn$ – static/dynamic mode, when given number and characteristics of VMs remains/not remains the same during scheduling process,
- $dist/centr$ – references that the scheduling objectives are optimized for multi-cloud environment, where a central meta-scheduler interacts with local cloud schedulers in order to define the optimal schedules, or the centralized mode for single cloud scheduling,
- obj – denotes the set of the considered scheduling objective functions.

Definition of the main scheduling attributes is necessary for the specification of a particular scheduling problem in clouds.
Scheduling procedure can be realized in the following six steps [15]:

1. gathering the information on available resources,
2. assembling the details of pending tasks,
3. cumulating facts about data hosts where files for tasks completion are required,
4. preparing a batch of tasks or single task and compute a schedule for that batch/single mode on available machines and data hosts,
5. allocating tasks to resources,
6. monitoring the energy spent on the process when power-aware scheduling is incorporated or thermal effects, when thermal-aware scheduling was assumed.

Due to the three level services offered by the cloud vendors, these procedures may be divided as far as the scale of optimized system is concerned. Therefore, single server, compute cluster, distributed virtualized infrastructure, data centre, and the whole cloud system [16] may be taken into consideration.

3.3. Scheduling Measures and Criteria

In the problem of task scheduling we have to find schedules that minimize chosen possible objectives. The most popular scheduling criteria, namely makespan, flowtime and maximal lateness, are defined as [17]:

- makespan – the most popular time-based objective. It indicates the finishing time of the last task from task pool. The makespan can be calculated by:

$$C_{\max} = \min_{S \in Schedules} \left\{ \max_{j \in Tasks} C_j \right\}, \tag{5}$$

  where C_j denotes the time when task j is finalized (in other words, it is the machine completion time), *Tasks* denotes the set of all tasks submitted to the cloud, and *Schedules* is the set of all possible schedules;

- flowtime – defines the sum of finalization times of all the tasks. It can be defined as:

$$F = \min_{S \in Schedules} \left\{ \sum_{j \in Tasks} C_j \right\}, \tag{6}$$

  where the variables are as above;

- maximal lateness – defines the maximum time elapsed between the finalization and assumed deadline of a task. The maximum lateness is calculated as:

$$Lat_{\max} = \max_{j \in Tasks} Lat_j, \tag{7}$$

  where Lat_j denotes the lateness for the task j and

$$Lat_j = C_j - d_j, \tag{8}$$

  where C_j denotes the time when task j is finalized, and d_j is the deadline for task j;

- total energy consumption – defines the cumulative energy consumed during task batch processing. It can be defined as:

$$E_{total} = \left\{ \sum_{i \in Machines} E_i \right\}, \tag{9}$$

  where *Tasks* denotes the set of all virtual machines, and E_i the cumulative energy utilized by the machine i for the completion of all tasks from the batch that are assigned to this machine.

When the scheduling is made according to the energy consumption one of the presented criterion – Eqs. (5)–(7) – is considered as a primary scheduling criterion. The total energy consumption by Eq. (9) is the second scheduling criterion.

3.4. ETC Matrix Model Based Energy-aware Independent Batch Scheduling

An example problem that is the subject of many studies in modern task scheduling methods is Independent Batch Scheduling (IBS) [12], [18], [19]. In this problem the tasks are gathered into batches and independently processed on assigned resources. According to the notation introduced in formula (3), the problem can be defined as:

$$Rm[\{batch, indep, (stat, dyn), centr\}](obj). \tag{10}$$

The problem of IBS can be considered under several criteria. The most popular are makespan and flowtime Eqs. (5) and (6), respectively. For estimating the execution times of tasks on machines can be the ETC matrix model adopted. The model is proposed in [20] and adapted for energy-aware independent batch scheduling in [17] and [19].
In the general case, the entries of the $ETC[j][i]$ parameters can be calculated as the ratio of the workload wl of task j and computing capacity cc of machine i:

$$ETC[j][i] = \frac{wl_j}{cc_i}. \tag{11}$$

According to [17] and [19], the average energy consumption can be considered as a complementary scheduling criterion along with the makespan – see Eq. (5) – as the primary objective. The makespan is expressed as the maximum completion time of the machines. Completion time also includes the time needed for reloading the machine i after finalizing the previously assigned tasks. The minimization of the total energy consumed in the process of tasks batch execution is considered as the second step of the suboptimal schedule selection.

3.5. Energy Efficient Task Scheduling Methods for Grids

Proposed model considers two main scheduling scenarios.

Max-Min Mode, in which each machine works at the maximal DVFS during the execution and computation of tasks, and enters into idle mode after the execution of all tasks assigned to this machine. In this scenario the completion time can be defined as:

$$completion_I[i] = ready_i + \sum_{j \in Tasks(i)} ETC[j][i], \tag{12}$$

where: $ready_i$ – the ready time of machine i and $ETC[j][i]$ – the expected completion times for task j on machine i. The makespan in this scenario is calculated as:

$$(C_{max})_I = \max_{i=1}^{m} completion_I[i]. \tag{13}$$

For Max-Min Mode the average energy consumed in the system is defined as:

$$E_I = \frac{1}{m} \cdot \sum_{i=1}^{m} \gamma \cdot (completion_I[i] \cdot f \times \\ \times [v_{s_{max}}(i)]^2 + f_{s_{min}}(i) \cdot [v_{s_{min}}(i)]^2 \cdot Idle_I[i]). \tag{14}$$

where: m – number of machines, $\gamma = A \cdot C$ (C is the total capacitance load, A is the number of switches per clock cycle), $completion_I[i]$ – completion time of the machine i, f – frequency of the machine i, $v_{s_{max}}(i)$ – machine voltage supply, $s_{min/max}$ – minimum/maximum DVFS level, $Idle_I[i]$ – the idle time for the machine i given by:

$$Idle_I[i] = (C_{max})_I - completion_I[i]\,. \tag{15}$$

Modular Power Supply Mode, in which each machine can work at different DVFS levels during the task executions and can then enter into idle mode. In this scenario the completion time, makespan, and idle time at the level s^i take specific forms given by:

$$completion_{II}[i] = ready_i + \sum_{j \in Tasks(i)} \frac{1}{f_{s_l}(i)} \cdot ETC[j][i]\,, \tag{16}$$

$$(C_{max})_{II} = \max_{i=1}^{m} completion_{II}[i]\,, \tag{17}$$

$$Idle_{II}[i] = (C_{max})_{II} - completion_{II}[i]\,. \tag{18}$$

Whereas, the average cumulative energy is defined as:

$$E_{II} = \frac{\sum_{i=1}^{m} E_i}{m}\,, \tag{19}$$

where:

$$E_i = \gamma \cdot f \cdot \sum_{\substack{j \in T(i) \\ l \in L_i}} ([(v_{s_l}(i))_j]^2 \cdot ETC[j][i]) +$$

$$+[v_{s_{max}}(i)]^2 \cdot ready_i + f_{s_{min}}(i) \cdot [v_{s_{min}}(i)]^2 \cdot Idle[i]\,, \tag{20}$$

where: $T(i)$ – a set of tasks assigned to machine i, L_i – set of DVFS levels specified for tasks assigned to machine i, and the remaining variables as in Eqs. (12)–(14).

The objective function was assumed as minimization of E_I and E_{II}.

The above-mentioned scenarios are based on Dynamic Voltage and Frequency Scaling (DVFS) technology. This method is based on decreasing power consumption of hardware by lowering the clock frequency and/or voltage of the CPU and attached peripherals under the assumption of known computational load. DVFS optimization is taking into account only CPUs. The peripherals, i.e. interfaces, memory, and disks, are being kept at the original operating frequency [21].

For the case of control all the resources of the physical machine is used less flexible technology – Dynamic Power Management (DPM). DPM methods consist of technologies to improve power conservation capabilities of computer system during runtime by shutting down the whole servers. A scheduler used in cooperation with DPM technique have to find a minimum set of computing resources for a given jobs. This approach is more efficient because the power consumption of a each server is proportional to its CPU utilization. When server is idle it still consumes around two-thirds of its peak-load consumption. This energy is spend on keeping memory, disks, and I/O resources running and ready for next task [21].

DVFS and DPM are the most popular technologies for power management of in distributed high-performance environments.

3.6. Energy Efficient Task Scheduling Methods for Clouds

There is a significant body of research on task scheduling approaches that target an efficient energy usage [2], [22]–[27]. Many of these approaches also employ switching the idle machines to sleep mode to save further on energy consumption [5], [28], [29].

Beloglazov *et al.* [5] introduce an architectural framework and principles for energy-efficient cloud computing. The authors define policies and scheduling algorithms for energy-efficient resource allocation ensuring that take into account both the quality of service provided and the power consumption. In that research the authors use the following power model

$$P(u) = k \cdot P_{max} + (1-k) \cdot P_{max} \cdot u\,, \tag{21}$$

where P_{max} is the maximum power consumed of a fully used server, k is the ratio of the power consumed by the idle servers, i.e. 70% in that paper, and u is the CPU utilization. The authors consider P_{max} as 250 W based on results offered by SPECpower benchmark[1]. The CPU utilization is workload dependent, hence changes in time. Consequently, the total energy consumed by a physical node E can be defined as:

$$C = \int_{t_0}^{t_1} P\big(u(t)\big)\mathrm{d}t\,. \tag{22}$$

The authors evaluate the proposed heuristic using modeling and simulation, and they show that using a heuristic based on minimizing the number of VMs to be migrated and considering the performance-related SLA requirements offers good energy savings.

Follow-up work by Beloglazov and Buyya [29] introduce an optimal online deterministic algorithms and heuristics for energy- and performance-efficient dynamic VM consolidation. In the context of dynamic VM consolidation, the authors defined the cost as:

$$C = \sum_{t=t_0}^{T} \left(C_p \sum_{i=0}^{n} a_{ti} + C_v \sum_{j=0}^{n} v_{tj} \right), \tag{23}$$

where t_0 is the initial time and T is the total time. a_{ti} shows whether the host i is active at time t, and v_{ti} shows whether the host j has a SLA violation at time t, the values of a_{ti} and $v_{ti} \in 0, 1$. The cost includes both the cost of power and the cost of any violation of the SLA – in this work that is when

[1]https://www.spec.org/power_ssj2008/

the service level performance, measured as maximum allowed CPU performance, cannot be met. The authors introduce novel adaptive strategies based on historical resource usage analysis for the energy efficient dynamic consolidation of VMs that minimize the total cost C. The authors propose a power-aware VM placement algorithm where all the VMs are queued in decreasing order of their CPU utilizations, and each VM will be allocated to the host that offers the minimum increase of the power usage due to the VM allocation. The evaluation of the proposed approach uses CloudSim [30], a research cloud simulator toolkit. The experiments are conducted against a simulated data center of 800 heterogeneous physical nodes. The evaluation shows that the proposed Local Regression (LR)-based algorithm combined with the Minimum Migration Time (MMT) VM selection policy provides better results for the minimization of energy and the SLA violations because of a lower number of SLA violations and VM migrations.

Mhedheb *et al*. propose ThaS [22] a load and thermal-aware VM scheduling approach with the aim to both minimize the energy consumption and ensure a good load-balancing. ThaS has been implemented on top of CloudSim [30], a research cloud simulator toolkit. The scheduler detects all the hosts that exceed either a particular temperature threshold or a CPU threshold. Next, the scheduler determines the VMs to be migrated and the target hosts. The target hosts are chosen based on temperature first, and the resources requirements second.

3.7. Meta-heuristic Energy Efficient Task Scheduling Methods

Modern energy-aware task scheduling methods are often based on a heuristic approach. These methods are usually classified into three main categories: calculus-based (greedy algorithms and ad-hoc methods), stochastic (guided and non-guided methods) and enumerative methods (dynamic programming and branch-and-bound algorithm). According to [31], the most important and efficient scheduling methods are ad-hoc, local search-based and population-based meta-heuristics methods.

Basing on proposed taxonomy, the following exemplary methods dedicated to the problem of energy aware task scheduling can be classified as meta-heuristics methods:

- **Hierarchic Genetic Strategy Based Scheduler (HGS-Sched)** is the model proposed in [17] and [19]. HGS-Sched model in the aforementioned papers was defined as meta-heuristic scheduler for solving the problem of IBS. This scheduling problem was defined by using the ETC matrix model with estimated time needed for the completion of the task j on the machine i;
- **PATC and PALS Energy-aware parallel task schedulers** [32] presented the Power Aware Task Clustering algorithm for parallel task scheduling and the Power Aware List-based Scheduling algorithm for parallel tasks.

4. Example of Batch Scheduling for Clouds Based on ETC Matrix Approach

The example of such scheduler implementation is presented in [33], [34]. It is based on additional scheduling criteria considering security of tasks computation. From among the many cloud computing security issues, [35] the mapping the task security demand into the proper VM offering the required trust level was considered. Here, for the clarity of presentation, the case considering two chosen Amazon instances will be presented. The makespan criterion, see Eq. (5), was used for scheduling. First VM (VM_1) is based on Amazon m4.16large instance with Intel Xenon E6-2686 v4 processor. Second VM (VM_2) is m4.large instance, equipped with Xenon E6-2676 v3 processor. Computing capacities of both are: $cc_1 = 2.7$ GHz $\times$ 18 cores $\times 16 = 777.6$ GFLOPS and $cc_2 = 2.4$ GHz $\times$ 12 cores $\times 16 = 460$ GFLOPS.

The batch consisting thee tasks was considered. The workload of tasks was: $wl_1 = 2000$, $wl_2 = 4000$, $wl_3 = 10000$.

The ECT matrix for such a batch is:

$$ECT = \begin{bmatrix} 2.57 & 5.14 & 12.86 \\ 4.36 & 8.68 & 21.70 \end{bmatrix}. \tag{24}$$

The possible schedules and makespans are presented in Table 1. One can see that the proper scheduling enables to save $30.38 - 8.68 = 21.7$ s. That is to shorten the makespan of tasks by over 71%.

Table 1
Possible schedules and their makespans

Schedule no.	VM_1 tasks	VM_2	Makespan [s]
1	1	2.3	28.38
2	2	1.3	26.04
3	3	1.2	13.03
4	1.2	3	21.70
5	1.3	2	8.68
6	2.3	1	30.38

Considering two time independent states of both VMs: busy (100% computational power used for tasks calculations) and idle (70% of maximal power used for system maintaining), we may calculate the energy necessary for this tasks.

Let the t_i^1 and t_i^2 be the time when VMs are idle, and t_{busy}^1 and t_{busy}^2 be the time when they are fully loaded. Let the P_i^1 and P_i^2 be the power necessary for VMs to keep idle state, and P_{busy}^1 and P_{busy}^2 be the power of VMs when they are calculating tasks. Then:

$$E_{total} = E(VM_1) + E(VM_2) =$$
$$= \int_0^{completiontime} Pow_{VM_1}(t)\mathrm{d}t + \int_0^{completiontime} Pow_{VM_1}(t)\mathrm{d}t =$$
$$= P_i^1 \cdot t_i^1 + P_i^2 \cdot t_i^2 + P_{busy}^1 \cdot t_{busy}^1 + P_{busy}^2 \cdot t_{busy}^2 . \quad (25)$$

Following [14], the VM power is estimated as the most simple linear function of virtual CPU power consumption. According to [17], the power necessary for both VMs to keep the idle state was assumed as the 70% percent of working VM. Assuming levels of VM energy:

$$\begin{aligned} P_i^1 &= 231 \text{ W}, \quad P_i^2 = 140 \text{ W}, \\ P_{busy}^1 &= 330 \text{ W}, \quad P_{busy}^2 = 200 \text{ W}, \end{aligned} \quad (26)$$

the energy consumed by each VM during processing assumed batch can be calculated, see Table 2.

Table 2
Energy and energy efficiency for possible schedules for the whole environment and particular VMs

Schedule no.	Energy	$E_{efficiency}$
1	12486.21	1.28
2	11732.10	1.36
3	6889.07	2.32
4	10412.99	1.53
5	454.37	32.21
6	14413.20	1.11

The last schedule saves $14413.2 - 454.37 = 13958.83$ W. That is over 96% comparing the worst case scheduling.
Considering different energy levels for both VMs:

$$P_i^1 = 70\% \cdot P_{busy}^1, \quad P_i^2 = 70\% \cdot P_{busy}^1, \quad (27)$$

$$P_{busy}^1 \in [100, 500] \text{ W } P_{busy}^2 \in [100, 500] \text{ W} \quad (28)$$

we may find the energy dynamics necessary for this batch processing, for best (no. 5) and worst (no. 6) schedule, see Fig. 1. It shows that even for the most simple energy model, the gain from proper scheduling is significant. The energy is saved for all power configurations. Moreover, the percentage savings are bigger when energy consumption of VMs are high.

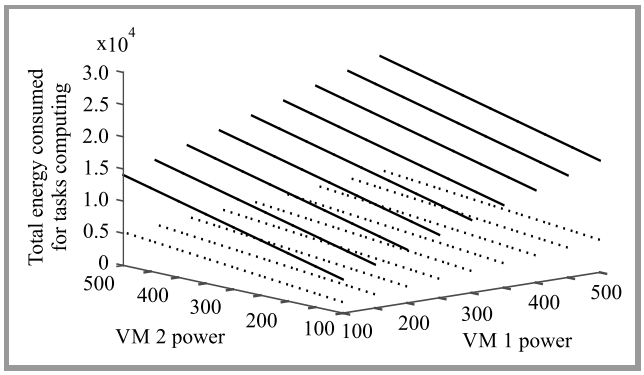


Fig. 1. Energy of batch processing for different VMs power levels for best and worst makespan schedules.

In the considered example, the power of VM was the linear function of computing capacity for particular configuration from Tables 1–2 problem of finding the schedule that minimizes the makespan is equal to the problem of finding the schedule that minimizes the total energy.
In general, the problem of finding the schedule that minimizes the makespan may be written in the form:

$$\arg \min_{s \in Schedules} \sum_{i=1,2,\ j=1,2,3} \frac{wl_j}{cc_i} \delta_{i,j}, \quad (29)$$

where $\delta_{i,j} = 0$ when the task number j is not scheduled for the machine i, $\delta_{i,j} = 1$ otherwise.
The problem of finding the schedule that minimizes the total energy may be written in the form:

$$\arg \min_{s \in Schedules} \sum_{j=1,2,3}^{\delta_{i,j}=1, i=1} P_{busy}^1 \frac{wl_j}{cc_i} \delta_{i,j} + \sum_{j=1,2,3}^{\delta_{i,j}=1, i=2} P_{busy}^2 \frac{wl_j}{cc_i} \delta_{i,j} + \sum_{j=1,2,3}^{\delta_{i,j}=0, i=1} P_i^1 \frac{wl_j}{cc_i} \delta_{i,j} + \sum_{j=1,2,3}^{\delta_{i,j}=0, i=2} P_i^2 \frac{wl_j}{cc_i} \delta_{i,j}. \quad (30)$$

One can see in this case the solution of finding the schedule that minimizes the makespan and the energy expenditure is the same. This is due to the fact that the power consumption is increasing as the computer capacity is growing, see Eq. (23).
For the schedule s and given tasks batch, the energy efficiency may be defined as the number of operations performed per energy unit (see Table 2):

$$E_{efficiency}(s) = \frac{\sum_{j=1,\ldots,n} wl_j}{E_{total}(s)}. \quad (31)$$

It reflects the quality of energy aware scheduling considering given energy usage by virtual environment.

5. Summary

In this paper we addressed the problem of energy efficient task scheduling and load balancing in cloud environments. We have reviewed and discussed the methods and approaches applied for the reduction of energy consumption. The analysis shows that the problem of energy-aware task scheduling and load balancing are still very challenging.
The described model considers the multi-objective optimization problem. It focuses not only on energy consumption, but also on taking into account the time-based objectives, which are crucial in the problem of energy consumption. As a result, it considers the problem of finding the right compromise between the makespan and energy efficiency.
Additionally, we presented simple numerical example illustrating the influence of proper scheduling into energy saving.

All presented models achieved effective results in this field and are worthy of additional attention.

References

[1] K. H. Kim, A. Beloglazov, and R. Buyya, "Power-aware provisioning of cloud resources for real-time services", in *Proc. 7th Int. Worksh. on Middleware for Grids, Clouds and e-Science MGC'09*, Urbana Champaign, IL, USA, 2009, pp. 1:1–1:6 (doi: 10.1145/1657120.1657121).

[2] A. Beloglazov and R. Buyya, "Adaptive threshold-based approach for energy-efficient consolidation of virtual machines in cloud data centers", in *Proc. 8th Int. Worksh. on Middleware for Grids, Clouds and e-Science MGC'10*, Bangalore, India, 2010, pp. 4:1–4:6 (doi: 10.1145/1890799.189080333).

[3] X. Fan, W.-D. Weber, and L. A. Barroso, "Power provisioning for a warehouse-sized computer", in *Proc. 34th Ann. Int. Symp. on Comp. Architec. ISCA'07*, San Dieco, CA, USA, 2007, pp. 13–23.

[4] E. Pinheiro, R. Bianchini, E. Carrera, and T. Heath, "Load balancing and unbalancing for power and performancee in cluster-based systems" in *Proc. of the Worksh. on Compilers and Operat. Syst. for Low Power COLP'01*, Barcelona, Spain, 2001, pp. 182–195.

[5] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing", *Future Gener. Comput. Syst.*, vol. 28, no. 5, pp. 755–768, 2012.

[6] R. Nathuji and K. Schwan, "Virtualpower: Coordinated power management in virtualized enterprise systems", in *Proc. 21st ACM SIGOPS Symposium on Operat. Syst. Principles SOSP'07*, Stevenson, WA, USA, 2007, pp. 265–278.

[7] D. Kusic, J. O. Kephart, J. E. Hanson, N. Kandasamy, and G. Jiang, "Power and performance management of virtualized computing environments via lookahead control", *Cluster Comput.*, vol. 12, no. 1, pp. 1–15, 2009.

[8] G. Dhiman, K. Mihic, and T. Rosing, "A system for online power prediction in virtualized environments using Gaussian mixture models", in *Proc. 47th Design Autom. Conf. DAC'10* Anaheim, CA, USA, 2010, pp. 807–812.

[9] A. Hameed *et al.*, "A survey and taxonomy on energy efficient resource allocation techniques for cloud computing systems", *Computing*, vol. 98, no. 7, pp. 751–774, 2016.

[10] S. Russell and P. Norvig, *Artificial Intelligence. A modern approach.* Englewood Cliffs: Prentice-Hall, 1995.

[11] R. Kaur and P. Luthra, "Load balancing in cloud computing", in *Proc. of Int. Conf. on Recent Trends in Inform., Telecom. and Comput. ITC '12*, Bangalore, India, 2012.

[12] D. Grzonka, J. Kołodziej, J. Tao, and S. U. Khan, "Artificial neural network support to monitoring of the evolutionary driven security aware scheduling in computational distributed environments", *Future Gener. Comput. Syst.*, vol. 51, no. C, pp. 72–86, 2015.

[13] P. Fibich, L. Matyska, and H. Rudová, "Model of grid scheduling problem", in *Proc. Worksh. on Explor. Plann. and Schedul. for Web Services, Grid and Autonomic Comput.*, Pittsburgh, PA, USA, 2005, pp. 17–24.

[14] D. Klusáček and H. Rudová, "Efficient grid scheduling through the incremental schedule-based approach", *Computat. Intell.*, vol. 27, no. 1, pp. 4–22, 2011.

[15] J. Kołodziej, S. U. Khan, L. Wang, A. Byrski, N. Min-Allah, and S. A. Madani, "Hierarchical genetic-based grid scheduling with energy optimization", *Cluster Comput.*, vol. 16, no. 3, pp. 591–609, 2013.

[16] C. Cai, L. Wang, S. U. Khan, and J. Tao, "Energy-aware high performance computing: A taxonomy study", in *Proc. 2011 IEEE 17th Int. Conf. on Parallel and Distrib. Syst. ICPADS 2011*, Tainan, Taiwan, 2011, pp. 953–958.

[17] J. Kołodziej, *Evolutionary Hierarchical Multi-Criteria Metaheuristics for Scheduling in Large-Scale Grid Systems*. *Studies in Computational Intelligence*, vol. 419. Springer, 2012.

[18] J. R. Anette, W. A. Banu, and Shriram, "A taxonomy and survey of scheduling algorithms in cloud: Based on task dependency", *Int. J. of Comp. Appl.*, vol. 82, no. 15, pp. 20–26, 2013.

[19] J. Kołodziej, S. U. Khan, and F. Xhafa, "Genetic algorithms for energy-aware scheduling in computational grids", in *Proc. Int. Conf. on P2P, Parallel, Grid, Cloud and Internet Comput. 3PGCIC 2011*, Barcelona, Spain, 2011, pp. 17–24.

[20] S. Ali, H. J. Siegel, M. Maheswaran, D. Hensgen, and S. Ali, "Task execution time modeling for heterogeneous computing systems", in *Proc. 9th Heterogen. Comput. Worksh. HCW 2000 (Cat. No. PR00556)*, Cancun, Mexico, 2000, pp. 185–199 (doi: 10.1109/HCW.2000.843743).

[21] D. Kliazovich, P. Bouvry, and S. U. Khan, "Dens: Data center energy-efficient network-aware scheduling", in *Int. Conf. on Green Comput. & Commun. GreenCom 2010 and IEEE/ACM Int. Conf. on Cyber, Physical & Social Comput. CPSCom 2010)*, Hangzhou, China, 2010, pp. 69–75 (doi: 10.1109/GreenCom-CPSCom.2010.31).

[22] Y. Mhedheb, F. Jrad, J. Tao, J. Zhao, J. Kołodziej, and A. Streit, "Load and thermal-aware VM scheduling on the cloud", in *Algorithms and Architectures for Parallel Processing: 13th International Conference, ICA3PP 2013, Vietri sul Mare, Italy, Dec. 18-20, 2013, Proceedings, Part II*, R. Aversa, J. Kołodziej, J. Zhang, F. Amato, and F. Giancarlo, Eds. Springer, 2013, pp. 101–114.

[23] C. Ghribi, M. Hadji, and D. Zeghlache, "Energy efficient VM scheduling for cloud data centers: Exact allocation and migration algorithms", in *Proc. 13th IEEE/ACM Int. Symp. on Cluster, Cloud, and Grid Comput. CCGrid 2013*, Delft, The Netherlands, 2013, pp. 671–678.

[24] C. Ghribi and D. Zeghlache, "Exact and heuristic graph-coloring for energy efficient advance cloud resource reservation", in *Proc. IEEE 7th Int. Conf. on Cloud Comput. CLOUD 2014*, Anchorage, AK, USA, 2014, pp. 112–119.

[25] A. Beloglazov, "Energy-efficient management of virtual machines in data centers for cloud computing", Ph.D. thesis, The University of Melbourne, 2013 [Online]. Available: http://beloglazov.info/thesis.pdf

[26] A. Beloglazov and R. Buyya, "OpenStack Neat: A framework for dynamic and energy-efficient consolidation of virtual machines in openstack clouds", *Concurr. Comput.: Pract. Exper.*, vol. 27, no. 5, pp. 1310–1333, 2015 (doi: 10.1002/CPC.3314).

[27] A. Iqbal, C. Pattinson, and A. L. Kor, "Managing energy efficiency in the cloud computing environment using SNMPV3: A quantitative analysis of processing and power usage", in *Proc. IEEE 14th Int. Conf. Dependable, Autonom. & Secure Comput., 14th Int. Conf. Pervasive Intell. & Comput., 2nd Int. Conf. Big Data Intell. & Comput., and Cyber Sci. and Technol. Congr. DASC/PiCom/DataCom/ CyberSciTech*, Auckland, New Zealand, 2016, pp. 239–244 (doi: 10.1109/DASC-PICom-DataCom-CyberSciTec.2016.60).

[28] H. Goudarzi, M. Ghasemazar, and M. Pedram, "SLA-based optimization of power and migration cost in cloud computing", in *Proc. 12th IEEE/ACM Int. Symp. on Cluster, Cloud and Grid Comput. CCGrid 2012*, Ottawa, Canada, 2012, pp. 172–179.

[29] A. Beloglazov and R. Buyya, "Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers", *Concurr. Comput.: Pract. Exper.*, vol. 24, no. 13, pp. 1397–1420, 2012 (doi: 10.1002/CPC.1867).

[30] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya, "CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms", *Softw. Pract. Exper.*, vol. 41, no. 1, pp. 23–50, 2011.

[31] J. Kołodziej, F. Xhafa, L. Barolli, and V. Kolici, "A taxonomy of data scheduling in data grids and data centers: Problems and intelligent resolution techniques", in *Proc. Int. Conf. on Emerging Intell. Data and Web Technol. EiIDWT 2011*, Tirana, Albania, 2011, pp. 63–71 (doi: 10.1109/EIDWT.2011.20).

[32] L. Wang, S. U. Khan, D. Chen, J. Kołodziej, R. Ranjan, C.-Z. Xu, and A. Zomaya, "Energy-aware parallel task scheduling in a cluster", *Future Gener. Comput. Syst.*, vol. 29, no. 7, pp. 1661–1670, 2013.

[33] A. Jakóbik, D. Grzonka, and F. Palmieri, "Non-deterministic security driven meta scheduler for distributed cloud organizations", *Simul. Modell. Practice and Theory*, Elsevier, Nov. 2016 (doi: 10.1016/j.simpat.2016.10.011).

[34] A. Jakóbik, D. Grzonka, J. Kołodziej, and H. Gonzalez-Velez, "Towards secure non-deterministic meta-scheduling for clouds", in *Proc. 30th Eur. Conf. on Modell. and Simul. ECMS 2016*, Regensburg, Germany, 2016, pp. 596–602 (doi: 10.7148/2016-0596).

[35] A. Jakóbik, "Big data security", in *Resource Management for Big Data Platforms: Algorithms, Modelling, and High-Performance Computing Techniques*, F. Pop, J. Kołodziej, and B. Di Martino, Eds. Springer, 2016, pp. 241–261.



Agnieszka Jakóbik (Krok) received her M.Sc. in the field of stochastic processes at the Jagiellonian University, Cracow, Poland and Ph.D. degree in the field of neural networks at Tadeusz Kosciuszko Cracow University of Technology, Poland, in 2003 and 2007, respectively. From 2009 she is an Assistant Professor at Faculty of Physics, Mathematics and Computer Science, Tadeusz Kosciuszko Cracow University of Technology. Her main scientific and didactic interests are focused mainly on cloud computing security, artificial neural networks, genetic algorithms, and additionally on cryptography.
E-mail: agneskrok@gmail.com
Faculty of Physics, Mathematics and Computer Science
Tadeusz Kościuszko Cracow University of Technology
Warszawska st 24
31-155 Cracow, Poland



Daniel Grzonka received his B.Sc. and M.Sc. degrees with distinctions in Computer Science at Cracow University of Technology, Poland, in 2012 and 2013, respectively. Currently, he is Research and Teaching Assistant at Cracow University of Technology and Ph.D. student at Jagiellonian University in cooperation with Polish Academy of Sciences. He is also a member of Polish Information Processing Society and IPC member of several international conferences. The main topics of his research are grid and cloud computing, multi-agent systems and high-performance computing.
E-mail: grzonka.daniel@gmail.com
Institute of Computer Science
Faculty of Physics, Mathematics and Computer Science
Tadeusz Kościuszko Cracow University of Technology
Warszawska st 24
31-155 Cracow, Poland



Joanna Kołodziej is an Associate Professor in Department of Computer Science of Cracow University of Technology. She is a vice Head of the Department for Sciences and Development. She serves also as the President of the Polish Chapter of IEEE Computational Intelligence Society. She published over 150 papers in the international journals and conference proceedings. She is also a Honorary Chair of the HIPMOS track of ECMS. The main topics of here research is artificial intelligence, grid and cloud computing, multiagent systems.
E-mail: jokolodziej@pk.edu.pl
Faculty of Physics, Mathematics and Computer Science
Tadeusz Kościuszko Cracow University of Technology
Warszawska st 24
31-155 Cracow, Poland



Adriana E. Chis received her Diploma-Engineer (Honors) title in Computer Science and Engineering from the Department of Computer and Software Engineering, Faculty of Automation and Computers, "Politehnica" University of Timisoara, Romania in 2007. She received a Ph.D. in Computer Science from University College Dublin in 2013. Currently, she is a Lecturer in the School of Computing at National College of Ireland.
E-mail: adriana.chis@ncirl.ie
Cloud Competency Centre
National College of Ireland
Dublin, Ireland



Horacio González-Vélez is currently an Associate Professor and Head of the Cloud Competency Centre at the National College of Ireland. He spent over a decade working in systems engineering and product marketing for innovation-driven companies such as Silicon Graphics and Sun Microsystems. He earned a Ph.D. in Informatics from the University of Edinburgh.
E-mail: horacio@ncirl.ie
Cloud Competency Centre
National College of Ireland
Dublin, Ireland

Paper

Introduction to Big Data Management Based on Agent Oriented Cyber Security

Jamal Raiyn

Computer Science Department, Al Qasmi Academic College, Baqa Al Gharbiah, Israel

Abstract—This paper deals with information security and safety issues in public open spaces. Public open spaces include high streets, street markets, shopping centers, community gardens, parks, and playgrounds, each of which plays a vital role in the social, cultural and economic life of a community. Those outdoor public places are mashed up with various ICT tools, such as video surveillance, smartphone apps, Internet of Things (IoT) technologies, and biometric big data (called Cyber Parks). Security and safety in public places may include video surveillance of movement and the securing of personalized information and location-based services. The article introduces technologies used in Cyber Parks to achieve information security in big data era.

Keywords—big data, cyber security, information security.

1. Introduction

The data volume used in Internet technologies is rising rapidly. This huge amount is known as big data [1] and is characterized by three aspects according to Madden [2]:

- the data are numerous,
- the data cannot be categorized into regular relational databases,
- the data are generated, captured, and processed very quickly.

Big data has generated significant interest in various fields, including the manufacturing of healthcare machines, banking transactions, social media, and satellite imaging. Big data challenges have been described by Michael and Miller [3], such as rapid data growth, transfer speeds, the diversity of data, and security issues. Big data is still in its infancy stage and has not been reviewed in general. Hence, this study comprehensively surveys and classifies its various attributes, i.e. volume, management, analysis, security, nature, definitions, and rapid growth rate. The development of new IT technologies has rapidly increased the volume of information, which cannot be processed using existing technologies and methods [4]–[6]. In computational sciences, big data presents critical problems that require serious attention [7]. In the IT industry as a whole, the rapid rise of big data has generated new challenges with respect to data management and analysis. According to Khan *et al*. [8], five common issues involve: volume, variety, velocity, value, and complexity. Madden [2] note additional issues such as the fast growth of volume, variety, value, management, security, and efficiency. In some fields, data have grown rapidly. However, the type of data that increases most rapidly is unstructured data. This type is characterized by "human information" such as high-definition videos, movies, photos, scientific simulations, financial transactions, phone records, genomic datasets, seismic images, geospatial maps, e-mails, tweets, website data, call-center conversations, mobile phone calls, documents, sensor data, telemetry information, medical records and images, climatology and weather records, log files, and text. According to Khan *et al*. [8], unstructured information may account for more than 70% to 80% of all data in organizations. Currently, 84% of IT managers process unstructured data, and this percentage is expected to drop by 44% in the near future [9]. Most unstructured data are not modeled, are random, and are difficult to analyze.

Big data technology aims to minimize hardware and processing costs and to verify the value of information before committing significant company resources. Properly managed big data are accessible, reliable, secure, and manageable. Hence, such applications can be applied in various complex scientific disciplines (either single or interdisciplinary), including atmospheric science, astronomy, medicine, biology, genomics, and biogeochemistry. Khan *et al*. [8] have proposed a new data life cycle that uses the technologies and terminologies of big data. This new approach to data management and handling required in e-science is reflected in the scientific data life cycle management (SDLM) model. With this model, existing practices are analyzed in different scientific communities. The generic life cycle of scientific data is composed of sequential stages, including experiment planning (for research projects), data collection and processing, discussion, feedback, and archiving. The proposed data life cycle consists of the following stages: collection, filtering and classification, data analysis, storing, sharing and publishing, data retrieval and discovery.

In processing big data, users face several challenges [10]. Applications requires a huge storage capacity, rapidly search engines, sharing and analysis capabilities, and in some areas data visualization. These and others challenges

need to overcome to maximize big data. Currently, various techniques and technologies are used, such as SAS, R, machine learning platforms and Matlab to handle extensive data analysis. However, the proposed schemes are limited in managing big data effectively and are still lacking. According to Khan *et al*. [8], others challenges to big data analysis include data inconsistency and incompleteness, scalability, timeliness, and security.
This paper introduces a new scheme for big data management based on agent oriented cyber security in public spaces.

2. IoT Big Data Generation

In the IoT various area devices with enormous of sensors networks are used in different fields, such as, security and privacy, social network, transportation, medical care, industry, traffic, and public department. IoT devices are grown up quickly and collect the most important part of big data. IoT is considered an important source of big data.

2.1. Security and Privacy Indoor

Video surveillance system is the most important issue in homeland security field because of its ability to track and to detect a particular person. To overcome the lack of the conventional video surveillance system that is based on human perception this paper introduces a novel cognitive video surveillance system (CVS) that is based on mobile agents. CVS offers important attributes such as suspect objects detection, smart camera cooperation for person tracking. According to many studies, an agent-based approach is appropriate for distributed systems, since mobile agents can transfer copies of themselves to other servers in the system.
Various numbers of papers in the literature have been proposed and focused on computer vision problems in the context of multi-camera surveillance systems. The main problems highlighted in these papers are object detection and tracking and site-wide, multi-target, multi-camera tracking. The importance of accurate detection and tracking is obvious, since the extracted tracking, information can be directly used for site activity/event detection. Furthermore, tracking data is needed as a first step toward controlling a set of security cameras to acquire high-quality images, and toward, for example, building biometric signatures of the tracked targets automatically. The security camera is controlled to track and capture one target at a time, with the next target chosen as the nearest one to the current target. These heuristics-based algorithms provide a simple way of computing. Here the scenario is considered that the smart camera captures two similar objects (e.g. twins), then each object selects different path. The tracking process will become confused. Furthermore, the smart camera is limited to cover certain zone in public place (indoor).
The suggested solutions to improve the conventional video surveillance system are extended in various ways. A part of the approaches was to use an active camera to track a person automatically, thus the security camera moves in a synchronized motion along with the projected movement of the targeted person. These approaches are capable of locating and tracking small number of people. Another common approach was to position the camera at strategic surveillance locations. This is not possible in some situations due to the number of cameras that would be necessary for full coverage, and in such cases, this approach is not feasible due to limited resources. A third approach is to identify and track numerous targeted people at the same time involves image processing and installation of video cameras at any designated location. Such image processing increases server load.
The limitation of human perception system in conventional video surveillance system increases the demand to develop cognitive surveillance application. Many of the proposed video surveillance system are expensive and lack the capability of cognitive monitoring system (such as no image analysis) and ability to send warning signal autonomous in real-time and before the incidents happen. Furthermore, it is difficult and might take a long time for the human to locate the suspects in the video after the incidents did happen. The problem may get more completely in the larger scale surveillance system.
The next generation video surveillance systems expected not only to solve the issues of detection and tracking but also to solve the issue of human body analysis. In the literature, it can be found many references in development. In such area, the CVS aims to offer meaningful characteristics like automatic, autonomy, real-time surveillance such as face recognition, suspects object, target detection, and tracking using cooperative smart cameras. Many face recognition systems have a video sequence as the input. Those systems may require being capable of not only detecting but also tracking faces. Face tracking is essentially a motion estimation problem. Face tracking can be performed using many different methods, e.g., head tracking, feature tracking, image-based tracking, model-based tracking. These are different ways to classify these algorithms.

2.2. Model of CVS System

In this section we introduce the system model of the video surveillance system. Video surveillance system has been used for monitoring, real-time image capturing, processing, and surveillance information analyzing. The infrastructure of the system model is divided in three main layers: mobile agents that are used to track suspect objects, cognitive video surveillance management (CVS), and protocol for communication as shown in Fig. 1. Each end device, smart camera, covers a certain zone or cell. Smart camera used for collecting parameters of human face.
In the system model has been introduced two communication protocols. The first protocol is used for agent-to-agent communication protocol. The protocol is based on messages exchange as shown in Fig. 1. The goal is to update the agents. The second protocol is used for communication between CVS and mobile agent.

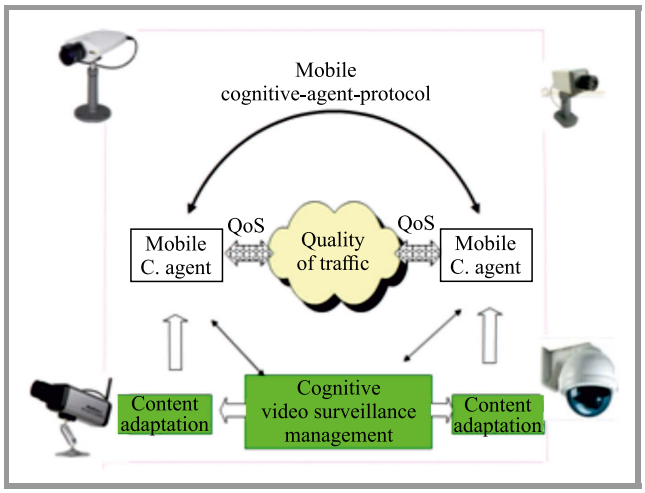


Fig. 1. CVS system model concept.

Mobile agents are placed in smart camera stations and aims to track the suspect object from smart camera station to others. Mobile agent offers various characteristics, e.g. negotiation, making decision, roaming, and cloning.
CVS provide the mobile agent with information. Based on received information mobile agents make decision when and where to move to next smart camera station.
In order to track moving objects, two strategies are used. The first is based on messaging protocol (msg_protocol) informing the mobile agent about the position of the suspect objects. The second strategy uses the protocol to help the mobile agent to roaming from point to others.

2.3. CVS Methodology

CVS uses a database of images. Pixels are described by a set of binary sequences. Each sequence presents certain properties (color). The database is divided into two separate sets of pixels – the training set and the test set. In both there are pixels, which belong to a certain family of colors (attributes) and sequence TP, which do not belong TN:

$$TP = X = \{X_1, X_2, \ldots, X_n\},$$

$$TN = Y = \{Y_1, Y_2, \ldots, Y_n\}.$$

Each image is then divided into frames $X_1 \ldots X_n$, a frame being a subset of n pixel from the sequence. The number of pixel in each frame is a variable and is dynamically set to obtain optimal results:

$$\begin{aligned} X_1 &= \{x_1^1, x_2^1, \ldots, x_n^1\}, \\ X_2 &= \{x_1^2, x_2^2, \ldots, x_n^2\}, \\ &\vdots \\ X_n^m &= \{x_1^m, x_2^m, \ldots, x_n^m\}. \end{aligned}$$

For example if a certain frame is comprised of 200 segments, the frames might consist of pixels 1 to 10, 2 to 11, 3 to 12, etc. Statistical methods are then applied to find correlation between a certain properties of the frame.

The basic logic of statistical differentiation of pixel is known and widely used in many prediction systems.

$$J = X \oplus Y,$$

$$J = \begin{cases} 1 & \text{if } x \neq y \\ 0 & \text{otherwise}. \end{cases}$$

A large number of correlating factors is defined by CVS and grouped in sets. A number is linked with each correlating factor. Each factor is then turned into a single number which represents the strength of the correlation factors for each frame with respect to the probability that this frame belongs to the certain family or not. As a result there are large number of frames, for each pair of a frame we have a number which is correlated to the probability that this frame belongs to a certain attribute (color similarity) or does not belong:

$$J = \{J_1^1, J_2^2, J_3^3, J_4^4 \ \ldots\}$$

or after optimization of J:

$$J_{Prediction}(J_1^*) - J_{demand} + k \cdot (\Delta J).$$

In addition to the statistical method logical XOR multiplication of matrices is applied to enrich the number of frames, which are potentially contributing to the prediction model. CVS can be implemented in a dynamic environment. When the training databases are modified, the prediction mechanism is modified as well with improved prediction capabilities.

2.4. Security and Privacy Outdoor

Modern cities offer various kinds of public places, and are created for different targets, i.e. public places for students and others on academic campuses, for visitors to historical sites, and for families and tourists. Public open spaces that are supported by various kinds of modern information communication technologies are called Cyber Parks [9]. Such places providing connectivity services to users on their personal computers, smart phones, tablets, and other mobile end-devices. Many users use Internet technologies for storing private data. Furthermore, Internet technologies are used for communication in business, the military, medicine, education, and government and public services. Over the last decade, as well, crime in virtual life has increased. Cyber attacks are performed through Internet networks that target individual machines, mobile devices, communications protocols, or smartphone application services. Cyber attacks are performed by spreading malware, by creating phishing websites, and by other means [6].
To implement information security policies and safety in Cyber Parks [10], security models are needed that lay out guidelines for securing information and communication. Cyber Park security models are based on formal models of access rights to smart phone applications and web services. In addition, an adaptive agent recognizes the applications that being used, and a mobile agent platform [11]–[15] creates mobile agents to serve the Cyber Park visitors. By

monitoring the behavior of users, detection systems ensure information privacy.
A mobile agent aims to fulfill user's preferences based on a dynamic environment. The mobile agent's structure is divided to three parts, as follows:

- Source code – the program consists of several classes to define the agent's behavior. In the source code, the backbone of the agent is created, which contains the basic rules. The agent then grows and develops itself according to the requirements of its environment;
- State – the agent's internal variables enable it to resume it is activities when it is found to be in one of the following states: offline (sleeping, in an evolution process), online (awake), busy, waiting (standby), or dead;
- Attributes – attributes consist of information describing the agent, its movement history, its resource requirements, and authentication keys.

In order to mediate useful tasks, a communication model to establish communication between mobile end users and the Cyber Park service provider is used. The agents in the system should be able to understand each other, and they should use the same message transport protocol. Messages are a data oriented communication mechanism, generally used to transfer data between processes. Communication is either asynchronous or synchronous.

2.5. Concept of Secured Information

Authentication refers to process of obtaining a confirmation that a person who is requesting a service, is a valid user. It is accomplished via the presentation of an identity and credentials, such as passwords, tokens, digital certificates, and phone numbers. To increase information security, users need a password to log in. The system starts the identification process and creates a mobile agent for each user, as shown in Fig. 2. The mobile agent is responsible for communication security in the system.

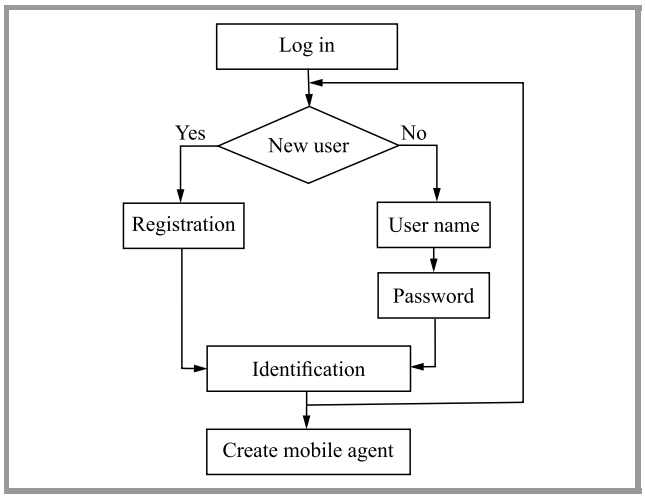


Fig. 2. Block diagram of authentication process.

Messages are a data oriented communication mechanism. Request-response mechanism is used to transfer data between a user end device and a service provider:

- inform message – includes the mobile device ID and the kind of information requested,
- re-inform message – includes information about Cyber Park resources,
- request message – includes the sender's name, a time stamp that indicates the time the request message was generated, the receiver's name and the requested resource,
- response message – includes the sender's name, a time stamp, and the requested resource.

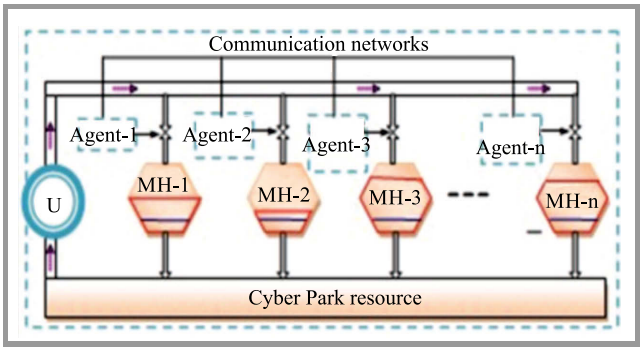


Fig. 3. Resources control.

Figure 3 illustrates the Cyber Park's resources and services. To increase information security communication between the mobile end user and service should be aware. It is necessary to checking the identity of the communication parties before establishing communication and allowing users access to information. Some users will follow a conventional scheme to access secure information. Namely, they access Cyber Park services with a password. For every user is allocated a mobile agent called a home agent. The home agent creates a new PIN number to access services. This number is a password shared between a user and a system that aims to authenticate the user.

3. Biometric Bid Data

Everyone uses several passwords to login to various systems and services. From simplicity and security viewpoint users demand new ways that replaces the passwords. Biometrics fills the user preferences and provide faster and easier. Biometrics is methods of recognizing a person based on physiological or behavioral characteristics. The recognizing process is based on measured features such as, face, fingerprint, hand geometric, iris, retinal, signature and voice. Authentication and security of biometrics data are very important issue. It refers to the collection of any kind of information about biological system, physiological or behavioral attributes. These data about humans is used to identify specific individual actions. Biometric security

is a security mechanism used to authenticate privacy data and to provide access to various IoT devices based on verification of an individual's physical characteristics. It is the strongest physical security technique used for identity verification. There are several algorithms, which convert the plain text data into cipher text data. These types of algorithms are known as encryption and decryption of data. It is used to protect data and it cannot be used for anyone except for the recipient.

3.1. Cyber Security and Privacy

It is important for Cyber Park visitors to keep their location secret. The privacy approach aims to protect private position information, as shown in Fig. 4. The mobile agent works to hide the identity of the user and his or her activity while the location for the user is visible. This prevents a cyber attacker from detecting the users location.

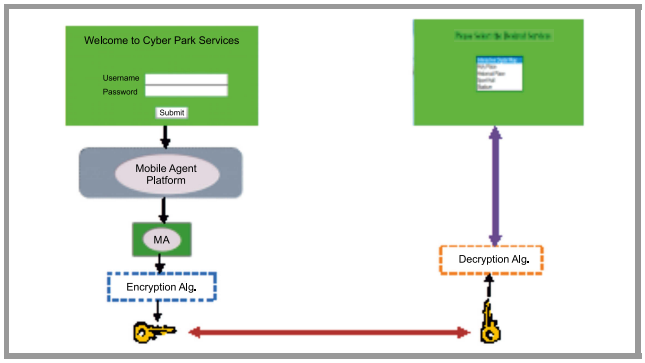


Fig. 4. Information security.

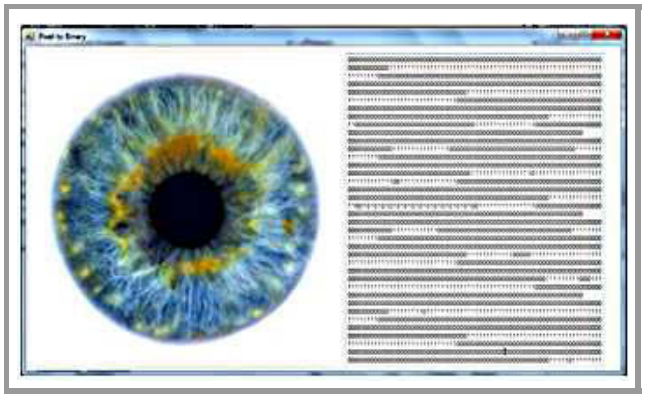


Fig. 5. Image representation in binary system.

The author has used the object oriented programming language C# to present the image in binary system as shown in Fig. 5. Hence, binary vectors are implemented in Waikato Environment for Knowledge Analysis (WEKA) platform, which implements many machine learning and data mining algorithms. As shown in Fig. 6 the image analysis in visual form is based on color classification. WEKA considers the color of the image. The colors are represented in binary system. WEKA clusters the binary vectors.

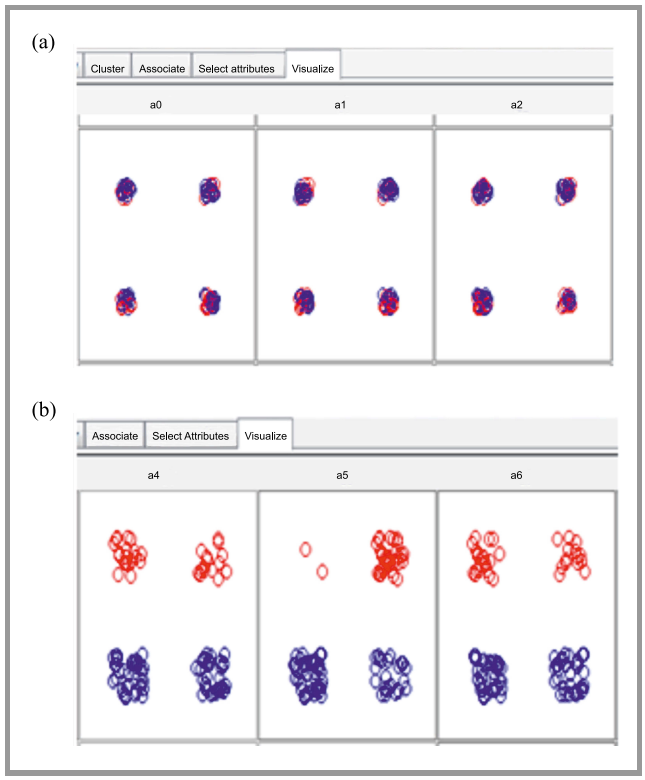


Fig. 6. WEKA platform: (a) image analyzis, (b) color classification.

4. Conclusion

The Internet and mobile technology are growing rapidly, and the data accumulated over twenty years have becomes big data. We have considered security big data indoor and outdoor, which is generated by IoT devices. The privacy approach aims to protect private position data. The mobile agent works to hide the identity of the user and his or her activity in Cyber Park (outdoor) services while the location for the user is visible. This prevents a cyber attacker from detecting the users location.

References

[1] D. Che, M. Safran, and Z. Peng, "From big data to big data mining: challenges, issues, and opportunities", in *Database Systems for Advanced Applications*, Berlin: Springer, 2013, pp. 1–15.

[2] S. Madden, "From databases to big data", *IEEE Internet Comput.*, vol. 16, no. 3, pp. 4–6, 2012 (doi: 10.1109/MIC.2012.50).

[3] K. Michael and K. W. Miller, "Big data: new opportunities and new challenges", *Computer*, vol. 46, no. 6, pp. 22–24, 2013 (doi: 10.1109/MC.2013.196).

[4] J. Raiyn, "Using cognitive radio scheme for big data traffic management in cellular systems", *Int. J. of Inform. Technol. & Manag.*, vol. 14, no. 2-3, 2015.

[5] J. Raiyn, "Toward developing real-time online course based interactive technology tools", *Adv. in Internet of Things*, vol. 4, no. 3, pp. 13–19, 2014 (doi: 10.4236/ait.2014.43003).

[6] J. Raiyn, "A Survey of cyber attack detection strategies", *Int. J. of Secur. & Its Appl.*, vol. 8, no. 1, pp. 247–256, 2014.

[7] C. A. Steed *et al.*, "Big data visual analytics for exploratory earth system simulation analysis", *Computers & Geosciences*, vol. 61, pp. 71–82, 2013 (doi: 10.1016/j.cageo.2013.07.025).

[8] N. Khan, I. Yaqoob, I. A. T. Hashem, Z. Inayat, W. K. M. Ali, M. Shiraz, and A. Gani, "Big data: Survey, technologies, opportunities, and challenges", *Scientif. World J.*, pp. 1–18, 2014 (doi: 10.1155/2014/712826).

[9] J. Raiyn, "Information security and safety in Cyberpark", *Global J. of Adv. Engin.*, vol. 2, no. 8, pp. 73–78, 2015.

[10] J. Raiyn, "Modern information and communication technology and their application in Cyberpark", *J. of Multidiscip. Sci. & Technol.*, vol. 2, no. 8, pp. 2178-2183, 2015.

[11] M. Wooldridge and N. R. Jennings, "Intelligent agents: Theory and practice", *The Knowl. Engin. Rev.*, vol. 10, no. 2, pp. 115–152, 1995.

[12] S. Russel and P. Norvig, *Artificial Intelligence: A Modern Approach*, Englewood Cliffs, NJ: Prentice Hall, 1995.

[13] M. Luck, V. Marik, O. Stepankova and R. Trappl, Eds., *Multi-Agent Systems and Applications. 9th ECCAI Advanced Course ACAI 2001 and Agent Link's 3rd European Agent Systems Summer School, EASSS 2001, Prague, Czech Republic, July 2-13, 2001. Selected Tutorial Papers*, *LNAI*, vol. 2086. Springer, 2001.

[14] T. Springer, T. Ziegert, and A. Schill, "Mobile agents as an enabling technology for mobile computing applications", *Kuenstliche Intelligenz*, vol. 14, no. 4, pp. 55–61, 2000.

[15] Z. Lin and K. Carley, "Proactive or reactive: An analysis of the effect of agent style on organization decision-making performance", *Intell. System in Account. Finance and Manag.*, vol. 2, no. 4, pp. 271–287, 1993.



Jamal Raiyn received the M.Sc. degree in Mathematics and Computer Science from Hannover University in Germany, in 2000 and he finished the Ph.D. study at Leibniz University of Hannover in Germany. In 2010 he finished his Postdoctoral at the Technion in Israel. Since 2002 he is Assistant Professor in Computer Science Department at the Al Qasemi Academic College in Israel. Since 2010 he is the Head of Computer Science Department at Al Qasemi Academic College in Israel. Since 2014 he is the Dean of the Faculty of Science at Al Qasemi Academic College.

E-mail: raiyn@qsm.ac.il
Computer Science Department
Al Qasmi Academic College
Baqa Al Gharbiah, Israel

Paper

Using Polymatrix Extensive Stackelberg Games in Security – Aware Resource Allocation and Task Scheduling in Computational Clouds

Agnieszka Jakóbik[1] and Andrzej Wilczyński[1,2]

[1] *Tadeusz Kościuszko Cracow University of Technology, Cracow, Poland*
[2] *AGH University of Science and Technology, Cracow, Poland*

Abstract—In this paper, the Stackelberg game models are used for supporting the decisions on task scheduling and resource utilization in computational clouds. Stackelberg games are asymmetric games, where a specific group of players' acts first as leaders, and the rest of the players follow the leaders' decisions and make their decisions based on the leader's actions. In the proposed model, the optimal schedules are generated under the security criteria along with the generation of the optimal virtual machines set for the scheduled batch of tasks. The security criteria are defined as security requirements for mapping tasks onto virtual machines with specified trust level. The effectiveness of the proposed method has been verified in the realistic use cases with in the cloud environment with OpenStack and Amazon Cloud standards.

Keywords—cloud computing, resource optimization, Stackelberg equilibrium, Stackelberg games.

1. Introduction

The problem of optimal resource allocation and utilization in computational clouds (CC) remains challenging research task in today's parallel computing. The considered computational infrastructure contains a large set of virtual machines (VMs) implemented in physical resources in distributed private and public cloud clusters. In public clouds, the customer can use any number of VMs and pay for used resources. The cloud provider collects the tasks from the users, analyzes the customers' requirements and tries to allocate the virtual and physical resources based on such requirements. Therefore, the users' requests for the access to the cloud services and resources may be dynamic. The users may change the number of VMs, which can be utilized. Therefore, the system must ensure the proper scheduling policies for such dynamic environment.

Task scheduling in the dynamic cloud environments is a complex process of contains multiple stages. Tasks may be scheduled as batch with (dependent) or without (independent) correlations among them [1]. In the scheduling process, the following special security-related users' requirements may be taken into account: selection of an appropriate amount of resources for utilization, safety, security, and intrusion detection. It may be also assumed that idle time for the resource should be minimal. Such conditions improve the energy awareness in scheduling, tasks execution, and building of the green cloud architectures. The set of VMs may be selected based on the estimation of the cost of scheduling, maximal utilization of the memory, and bandwidth, service access, etc.

This paper proposes a new model of selection and allocation of VMs in batch scheduling. In that model we focus on cloud services select based on the users' requirements. The scheduling model is based on central scheduling unit. The polymatrix extensive Stackelberg games are used for calculation of the optimal choosing strategies for the available cloud resources.

The model has been verified in a simple experimental analysis based on use cases specified for two sets of VMs implemented in the cloud clusters in OpenStack Racspace platform and Amazon Cloud.

The paper is organized as follows. In Section 2 the backgrounds of virtualization of the cloud resources and task scheduling in cloud systems are presented. In Section 3 security characteristics are defined, which must provide each of cloud environment. In Sections 4 and 5 related work is reviewed, considered problems are defined and tasks representation and models are explained. In Section 6 we present polymatrix extensive Stackelberg Games models. In Sections 7 and 8 the application of the Stackelberg Games as the support of the decision processes in the optimization of the VMs parameters are presented. In Section 9 the results of the application of the proposed model in realistic use case are shown. Finally, the paper is concluded in Section 10.

2. VMs Optimization and Task Scheduling in CC Systems

In CC infrastructure, VMs are the main target "resources" available and selected for scheduling of the computational tasks. They may be created from the images of VMs on demand. VMs can be configured with the characteristics specified in Tables 1 and 2. Tasks are mapped into VMs by using various scheduling procedures. In the case of batch

scheduling, the tasks are grouped into batches. For each consecutive batch, a different workload can be calculated based on the number of atomic numerical operations per second estimated for each task in the batch. It can be interpreted as task complexity measure.

Table 1
Selected Amazon Cloud VM instances

Model	vCPUs	CPU credits per hour	Mem [GB]
t2.nano	1	3	0.5
t2.small	1	12	2
t2.xlarge	4	54	16
t2.2xlarge	8	81	32
m4.16xlarge	64	120	256

Table 2
Selected OpenStack cloud VM instances

Flavor	vCPUs	Disk [GB]	RAM [GB]
General1-1	1	20 SSD	1
General1-2	2	40 SSD	2
General1-4	4	80 SSD	4
General1-8	8	160	8

In CC, the pay-as-you-go model of provisioning of the cloud resources is used. It means that the cloud user pays only for the utilization of those cloud resources, which are used for the task execution. Therefore, for each batch of tasks, different number of VMs may be employed. The total cost of the utilization of the cloud resources must be calculated for each batch. In the case of centralized scheduling, the characteristics of the cloud computational nodes must be specified before the start of the scheduling process. As major scheduling measure the makespan [2] can be used calculated for the batch of tasks.

3. Security Demands in Task Scheduling and Task Execution in CCs

Security remains one of the main challenges and crucial issues in CC [3]. Strong security requirements specified by the users may be the reason of high costs of the utilization of the cloud services. In the case of considering security issues as additional scheduling and resource allocation criteria, the execution time of tasks can be longer due to execution of some specific additional operations before or after the task calculation. Such "additional operations" can be defined as:

- task integrity checking,
- using cryptography algorithm for decoding the necessary data, that was previously ciphered for security,
- using cryptography algorithm for coding the data to be stored,
- additional identity checking procedures during sending data from one cloud cluster to another.

Additionally, different tasks may need different security requirements. In some cases, the access to data and resources are restricted by law [4]–[6]. For instance, the confidential medical data processing requires higher level of security than free stock photograph processing. Therefore, in the presented paper, an individual security level parameter is defined for each task. The task can be sent to the resource, which "guarantees" to keep the proper security level.
In this paper, we propose the model, in which the trust levels of VMs are specified based on the Federal Information Processing (FIPS) and ISO/IEC 19790 standards [7]:

1. trust level 1 – at least one approved algorithm or security function shall be used;
2. trust level 2 – system equipped with role-based authentication: cryptographic module authenticates the authorization of all sides of communication;
3. trust level 3 – identity–based authentication mechanisms: a cryptographic module authenticates the identity of sides of communication and verifies that their IDs are authorized to assume a specific role and perform a certain services. The entry or output of every plaintext has to be processed inside a module using a trusted path from other interfaces. Plaintext may be entered into or sent in output from the cryptographic module in encrypted form;
4. trust level 4 – the highest level of security. Penetration of the cryptographic module enclosure from any direction. Very low probability of cryptography procedures failure. The immediate suppression of all operations.

The example of realization of these services for OpenStack platform may be found in [8] and [9]. The relevant services for Amazon Cloud are given by cloud provider.

4. Related Works and Problem Definition

Although many scheduling methods in cloud environments have been proposed so far, only some of them are based on the game theory. In this section, we survey the most important developments in this area.
Ananth and Chandrasekaran [10] defined model in which the cloud resource utilization function and profit function for the service provider are maximized. The other considered optimization criteria are the minimization of the deadline violation and makespan estimated for available resources. The overall scheduling problem is defined as multi-objective task. The proposed schedulers are based on the game theory and genetic algorithms. The Pareto optimal solutions are estimated by using the non-dominated sorting genetic algorithm.

Geethanjali *et al.* in [11] defined the problem of the recognition of the wrong data generated by the service and resource providers. The aim of each cloud service provider is to maximize his profits. The authors assume that they may generate and distribute the incorrect information about the offered services. The authors show the scheduling mechanism for real time tasks to gain timing constraint, which ensures that the information distributed by the provider is truthful. It means that the minimal costs of the resource utilization and service usage are guaranteed. The authors use auction game with the Nash equilibrium as solution. In their game only one provider – the winner of the game – can receive all submitted tasks for calculation.

In [12], Qiu *et al.* proposed mechanism to manage the contact between the broker and private clouds to ensure that benefits of both the broker and the private clouds are maximized. They proposed a model with two-stage Stackelberg game. Leader decides and proposes pricing for renting all type of VMs for each private cloud, and each private cloud confirms the amount of VMs and the possibility of their rent. The main difference between that model and the model presented in this paper is that in [12] the authors assume the possibility that some task may not be scheduled. The tasks are organized in queues for distribution amount virtual resources. In this paper, we propose central scheduler instead tasks queuing. There are no privileged tasks. Tasks are treated as equally important. Additionally, all tasks must be scheduled to the machines. In this proposal the payoff functions may take several forms, such as energy spend on computation or minimum time for rebooting the system.

Shie *et al.* [13] use the game theory in their scheduling model for solving the resource competition game in the federated cloud. The Nash games are used. The results of experiments show that the cloud provider can get some extra profits by outsourcing resources if federated cloud has sufficient idle resources. In contrary to our proposal, the authors consider only Nash equilibrium case. These kinds of games do not allow making sequential decisions, which are possible in presented model. The model developed in this paper is about activities during tasks processing in the cloud system. This is the main reason of using Stackelberg games, in which the decisions are taken with certain succession. The cloud system is modeled by using the system of autonomous software agent (MAS) as the main unit, the Master and the farm of virtual resources (see Fig. 1). The ECT matrix model is used as the central scheduler [2]. Computing capacities of the resources must be specified for the calculation of the schedules. They are also the main constrain in the generation of the optimal schedule(-s). There are two other parameters specified. One of them is security level that VM may provide. Second parameter is the security requirements of the tasks that have been specified by the user. The security levels declared by the VMs should correspond to the security requirements of the tasks.

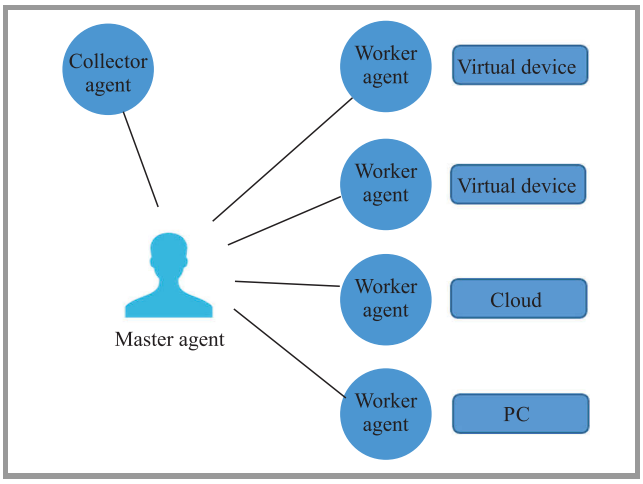


Fig. 1. Cloud model and hierarchy.

We consider two rounds of the games (Fig. 2) with three actions done automatically during each round of the game. Such a game also reflects different or competitive interests of game participants.

Table 3
The costs of renting of selected VM instances

Model	Price per hour
Amazon Cloud	
t2.nano	$0.0065
t2.small	$0.026
t2.xlarge	$0.188
t2.2xlarge	$0.376
m4.16xlarge	$3.83
OpenStack	
General1-1	£0.0035
General1-2	£0.007
General1-4	£0.014
General1-8	£0.028

5. Security Aware Cloud Resources and Tasks Modeling

The centralized scheduler considered in this paper, may be defined based on the general master-slave model (see Fig. 1). The master unit is responsible for task scheduling based on the resources available in the system. The sets of VMs can be implemented in each physical resource independently or based on the configuration of the other virtual resources.
The proposed model is suitable for cloud environment characterized as follows:

- a fully distributed environment with specified $n-1$ number of computational virtual resources,
- all tasks are generated by the system users and processed in batches,

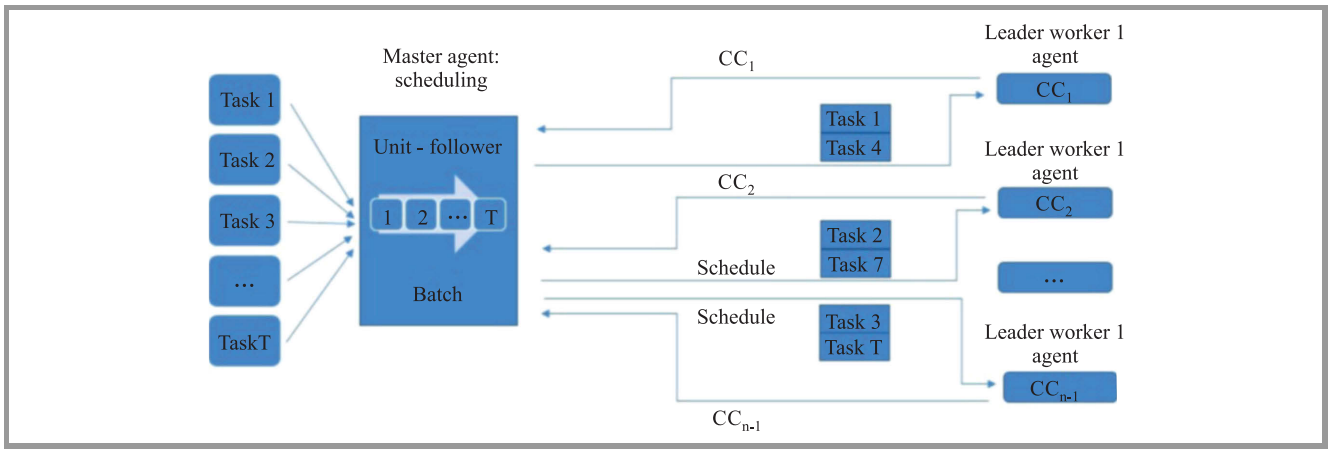


Fig. 2. Main game block diagram.

- each batch is composed of T tasks,
- a variety of computing capabilities, access modes, and response times is possible for all participating virtual resources,
- the performance of any given task on a single VM is neither related nor affected by the performance of any other VM,
- the number of VMs is fixed and remains constant during the execution of generated schedules,
- the scheduler sends the tasks one batch after another,
- there is no virtual resources without tasks – each resource has assigned at least one task, therefore must declare at least one VM,
- the tasks are independent,
- each task is calculated by one resource,
- all tasks in the batch have to be assigned and executed.

The following parameters should be specified for VM configuration:

- the number of units in the system: one master and $n-1$ slaves, that gives n units;
- proper task description for tasks $t = 1,\ldots,T$;
- security classes for tasks and virtual resources: $TASKsec = [Tsec_1,\ldots,Tsec_K]$ available K security classes for the tasks and $WORKsec = [Wsec_1,\ldots,Wsec_L]$ available L security classes for the virtual resources. For the clarity of presentation it is assumed, that: $TASKsec = WORKsec = [1, 2, 3, 4]$.
- security demand vector $SD \in \{1\ldots K\}^T$ for all the tasks in the batch. Each element of security demand vector SD_t corresponds to one class from $TASKsec$ vector and indicates that this task requires at least this particular security level;
- security capabilities of each resource are represented by trust level TL vector denoted by $TL = [tl_1,\ldots,tl_{n-1}] \in \{1\ldots L\}^{n-1}$. Each element of this vector indicates one class from $WORKsec$. Declaration of particular trust level means that the resource may operate at that trust lever or lower. For the clarity of presentation $SD, TL \in \{1, 2, 3, 4\}$;
- security classes for tasks and resource are specified as follows: the task t is scheduled for resource i if and only if $SD_t <= tl_i$ [9];
- workload vector for all T tasks in the batch $WL = [w_1,\ldots,w_T]$, expressed in giga floating point operations per second (GFLOPS);
- the computing capacity vector for each of $n-1$ virtual resources in GFLOPS $CC = [cc_1,\ldots,cc_{n-1}]$ [2];
- available VM types: VM_i the set of VM of each available type that may be created by resource number i;
- $CCVM_i$ the set of computer capacities of VM of each available type that may be created by i;
- proper mapping of security requirements at the master – slave level: SD into TL and WL into CC;
- cost for particular VMS is calculated per hour of working and formulates the cost vector: $costVM_i = [costVM_i(1),\ldots,costVM_i(M_i)]$, see Table 3.

6. Polymatrix Extensive Stackelberg Games Modeling

The n-player normal game Γ_n can be defined as [14]:

$$\Gamma_n = \left(N, \{S_i\}_{i\in N}, \{Q_i\}_{i\in N}\right), \tag{1}$$

where:

- $N = \{1,\ldots,n\}$ is the set of players,

- $\{S_1,\ldots,S_n\}$ ($cardS_i \geq 2;\ i=1,\ldots,n$) is the set of strategies for the players,
- $\{H_1,\ldots,H_n\}; H_i : S_1 \times \cdots \times S_n \to \mathbb{R}; \forall_{i=1,\ldots,n}$ is the set of payoff functions of the players.

The strategy of the player in the game is a plan of actions to make the game beneficial for him.
Stackelberg games are non-symmetric games, where one player (leader) or specified group of players (leaders) have the privileged position and plays first, and the rest of the players follow the leader and make their decisions based on the leader's actions [15]. Polymatrix game is the game where players' strategies form finite sets. The payoff of the player i is defined by the following multi–dimensional matrix [16]:

$$H_i = H_i(j_1,\ldots,j_n), \quad i \in N. \tag{2}$$

The class of mixed strategies of the player number i is denoted by

$$x_i = (x_1^i,\ldots,x_{p_i}^i), \tag{3}$$

where x_j^i is the probability that player i chooses the strategy $j \in \{1,,p_i\}$. Based on the strategy profile $x = (x_1,,x_n)$, the expected payoff of player i is calculated as:

$$H_i(x_1,,x_n) = \sum_{j_1=1}^{p_1} \ldots \sum_{j_n=1}^{p_n} H_i(j_1,,j_n) x_{j_1}^1 \cdot \ldots \cdot x_{j_n}^n, \quad i \in N. \tag{4}$$

Extensive-form n-player games are games where the above values may be changed when the next round begins.

7. Stackelberg Games in Cloud Systems

7.1. The Game Theory Approach for Scheduler and Virtual Resources

The schedule is calculated for each batch separately according to the tasks and computing capacities of the virtual resources. The master unit priority is to calculate the batch as soon as possible. The virtual resources unit is designed to find the set of VMs that fulfills declared computing capacity and minimize the computation cost. Therefore, the functionality may be considered as the Stackelberg game: virtual resources are behaving as the leaders playing computer capacity as the strategy and master unit is the follower responding the tasks splitting. Because tasks batches and their security demands may consume very much time, this is extensive game. The leader strategy is a finite type – there are finite VM types in the system to be chosen from. The follower strategy is also finite: there is finite number of possibilities for splitting given tasks for certain number of virtual resources. This is why the situation may be modeled as the polymatrix game.
Resource is charged for his services using pay-as-you-go cloud model. It means that the more tasks it gets the higher the payment will be. The payoff in the game is the highest if the cost of the VMs set is minimized keeping the computing capacity constant [15]. The computational cost of security operations (bias) may cause the offer less attractive for the follower. This is because the cost of computation is high, but it may be lowered.
Let us assume, that the set of players n is equal the number of virtual resources plus master. Then virtual resources may be numbered $i = 1, 2, \ldots, n-1$, the master unit is the player number n. The game is played according to the following rules, see Figs. 1–2:

- the leader is a set of virtual resources (the slaves),
- the follower is the central scheduling unit (the master),
- the leader proposing the computing capacity and decides first,
- the follower choosing the schedule and decides second,
- main game is ruling the capacity declaration and task scheduling,
- sub-game for virtual resources is also performed which allows finding the optimal resources for declared computing capacity and knowing task set considering the schedule.

Therefore, the following 5 steps have to be executed in each round of the game:

1. Batch of tasks is loaded to the master unit.
2. For all virtual resources i trust level tl_i is declared and computing capacities cc_i are calculated using the polymatrix game.
3. The follower calculates the schedule for all virtual resources. The schedule splits the given batch into virtual resources. The follower calculates also his strategy S_n and finds his payoff H_n. Then, the tasks are sent to virtual resources.
4. Each virtual resource is performing the sub-game: the follower computes his security bias b and strategy for choosing available VMs. Then, after setting the appropriate VMs set, tasks are computed.
5. return to step 1.

7.2. The Leader's Behavior

Let the

$$VM_i = [VM_i(1),\ldots,VM_i(M_i)] \tag{5}$$

be the number of VM of each available type that is created by resource number $i = 1,2,\ldots,n-1$. For instance, it can be read from Table 1. $VM_2 = [VM_2(1), VM_2(2), VM_2(3), VM_2(4), VM_2(5)] = [3,2,0,1,4]$ that the second resource has 5 VM types, and the profile is $3 \cdot$ t2.nano VMs, $2 \cdot$ t2.small, $0 \cdot$ t2.large, $1 \cdot$ t2.2 $\cdot$ large,

$4 \cdot$ m416large; $M_1 = 4$ [15]. Let us assume that the number of available machines lies in range of

$$VM_i(1) \in [1, m_i(1)], \ldots, VM_i(M_i) \in [1, m_i(M_i)]. \quad (6)$$

Let the

$$CCVM_i = [CCVM_i(1), \ldots, [CCVM_i(M_i)] \quad (7)$$

be the set of computer capacities of VM of each available type that may be created by resource number $i = 1, 2, \ldots, n-1$. If VMs are working in the parallel mode then

$$cc_i = VM_i(1)\, CCVM_i(1) + \cdots + VM_i(M_i)\, CCVM_i(M_i). \quad (8)$$

We denote by $CCVM_i^{min}$ the smallest possible computing capacity available per resource i. Then computing capacity may be declared from the range:

$$\begin{aligned} &\left[1 \cdot CCVM_i^{\min},\ m_i(1)\, CCVM_i(1) + \ldots \right. \\ &\left. + m_i(M_i)\, CCVM_i(M_i)\right] = \left[cc_i^{\min},\ cc^i \max\right]. \end{aligned} \quad (9)$$

We consider the possible computing capacities in ascending order. Then resource i may declare one of p_i possible values:

$$cc_i^1, cc_i^2, \ldots, cc_i^{p_i}. \quad (10)$$

The security capability of each resource i is represented by trust level tl_i and it is constant in time. It does not change during the game round.

7.3. The Follower Behavior for Obtaining the Schedule

We used the Expected Time to Compute (ETC) matrix model for calculation of the schedules [17]:

$$ETC[t][i] = \frac{w_t}{cc_i}, \quad (11)$$

that results to

$$ETC = \left[\frac{w_t}{cc_i}\right]_{i=1,2,\ldots,n-1}^{t=1,2,\ldots,T} \quad (12)$$

for each of i-th resource and t-th task. Scheduling objective as far as the performance of the task calculating is the makespan that equals the time when the latest task is done:

$$makespan = \min_{S \in Schedules} \left\{ \max_{j \in Tasks} C_j \right\}, \quad (13)$$

where C_j is the time when task j is finalized, *Tasks* are all tasks submitted to system and *Schedules* is the set of all possible schedules. In addition, security demand *SD* vector that describes security demands for each task indicates if the task may be assigned to machine represented by trust level TL vector.

If we have $n-1$ virtual resources and T tasks in the batch, the number of all possible schedules is much higher than considering security demands and trust levels.

7.4. Main Game Model

During main stage of the Stackelberg game, the user of the virtual resources have to decide about their computing capacities. Then the master calculates the schedule. After the tasks are sent to the virtual resources, each of resource is optimizing his set of VMs, see Fig. 2.

The possible mixed strategy for the player defined by see Eq. (3) is the probability that he chooses certain computing capacity defined by Eq. (10). The player $i = n$ (the follower) chooses certain schedule from all $p_n = s$ possible schedules using pure strategy model, that is $x_n \in \{0, 1\}$.

The payoff matrices from Eqs. (2) and (4) are defined in the form of:

- the makespan for known declared computing capacities and chosen schedule is:

$$H_n(j_1, , j_n) = makespan(j_1, , j_n), \quad (14)$$

- the schedule independent utility function reflects the situation when each certain resource strategy influence the payoff of the other virtual resources:

$$H_i(j_1, , j_n) = u_i(j_1, , j_{n-1}),\ i = 1, 2, \ldots, n-1, \quad (15)$$

- the schedule independent utility function when virtual resources payoffs are independent from other virtual resources decisions:

$$H_i(j_1, , j_n) = u_i(j_i),\ i = 1, 2 \ldots, n-1. \quad (16)$$

These payoff functions may be time dependent, round dependent or history dependent. Additional designations were omitted for clarity of presentation.

The solution of the Stackelberg game is called Stackelberg equilibrium [14]. In such a case, each follower observes the leader's strategy x and responds with strategy $f(x) : x \to y$ that is optimal with respect to his expected payoff. We can define two types of Stackelberg equilibrium points: Strong Stackelberg Equilibrium (SSE) and Weak Stackelberg Equilibrium (WSE). SSE assumes that the follower breaks ties in favor of the defender. It means that he chooses his optimal strategy, which is also optimal from the leader's perspective. WSE assumes that the follower chooses the worst strategy from the leader's perspective [18].

Let $x = (x_1, x_2, \ldots, x_{n-1})$ be the leader strategy and $f(x) = x_n$ be the follower response. Then the game is as follows:

- for fixed leader strategy x the follower solves the linear problem to find his optimal response:

$$\min_{x_n} \sum_{j_1=1}^{p_1} \ldots \sum_{j_n=1}^{s} makespan(j_1, , j_n) x_{j_1}^1 \cdot \ldots \cdot x_{j_n}^n, \quad (17)$$

with constraints that means that every pure strategy is possible:

$$\sum_{j_n=1}^{j_n=s,\ j_n>=0} x_{j_n}^n = 1; \quad (18)$$

- the leader finds the strategy x that maximizes his utility, under the assumption that the follower used optimal response x_n:

  1. in cooperation mode:

$$\max_x \sum_{j_1=1}^{p_1} \dots \sum_{j_n=1}^{s} H_i(j_1,,j_n) x_{j_1}^1 \cdot \dots \cdot x_{j_n}^n = \\ = \max_x \sum_{j_1=1}^{p_1} \dots \sum_{j_{n-1}=1}^{p_{n-1}} u_i(j_1,,j_{n-1}) x_{j_1}^1 \cdot \dots \cdot x_{j_{n-1}}^{n-1}, \tag{19}$$

  with assumption that each mixed strategy is possible:

$$\sum_{j_1=1}^{j_1=p_1} x_{j_1}^1 = 1, x_{j_1}^1 \in [0,1],$$

$$\sum_{j_2=1}^{j_2=p_1} x_{j_2}^2 = 1, x_{j_2}^2 \in [0,1],$$

$$\dots$$

$$\sum_{j_{n-1}=1}^{j_{n-1}=p_{n-1}} x_{j_{n-1}}^{n-1} = 1, x_{j_{n-1}}^{n-1} \in [0,1]; \tag{20}$$

  2. in non-cooperation mode:

$$\max_x \sum_{j_1=1}^{p_1} \dots \sum_{j_n=1}^{s} H_i(j_1,,j_n) x_{j_1}^1 \cdot \dots \cdot x_{j_n}^n = \\ = \max_x \sum_{j_i=1}^{p_i} u_i(j_i) x_{j_i}^i, \tag{21}$$

  with assumption that each mixed strategy is possible:

$$\sum_{j_i=1}^{j_i=p_i} x_{j_i}^i = 1, x_{j_i}^i \in [0,1]. \tag{22}$$

The optimal strategy, denoted by $[x_1^1, x_2^1, \dots, x_{p_1}^1] = [\frac{1}{24}, \frac{1}{6}, \frac{1}{12}, \dots, \frac{1}{3}]$, means that user of the resource number 1 should declare the lowest possible computing capacity with probability $\frac{1}{4}$, next consecutive computing capacity with probability $\frac{1}{6}$, and so on, and maximum computing capacity with probability $\frac{1}{3}$.

7.5. Sub-game Model

When computing capacities were chosen and the schedule is known, the set of VM may be set. The strategy of VMs capacities declaring is given by the Eq. (5) with constraints defined by Eqs. (6) and (8). The economical cost of the all created VMs is calculated per hour:

$$costVM_i = [costVM_i(1), \dots, costVM_i(M_i)]. \tag{23}$$

Therefore, for each resource i, one hour of working (for the set of VMs having declared computing capacity) costs:

$$cost(i) = VM_i(1)\, costVM_i(1) + \dots \\ \dots + VM_i(M_i) = Cost_i(M_i). \tag{24}$$

Let the $W(i)$ be the workload of all T_i tasks scheduled for resource i. Each resource needs to meet the time deadline given by ETC matrix, see Eq. (12). It result to

$$time(i) = \frac{W(i)}{cc_i} \tag{25}$$

seconds for computing given tasks. The cost of computing given tasks is:

$$time(i) = \frac{cost(i)}{3600}. \tag{26}$$

The aim of the sub-game is to find the VM configuration what cost the least. To do it, we have to to solve to problem:

$$argmin_{[VM_i(1),\dots,VM_i(M_i)} \dots \tag{27}$$

$$\frac{W(i)}{cc_i} \cdot \frac{VM_i(1)\, costVM_i(1)}{3600} + \dots \\ \dots + \frac{VM_i(M_i)\, costVM_i(M_i)}{3600} \tag{28}$$

with constraints (6) and (8).

8. Search for Strategies and Equilibrium

The following conditions have to be satisfied to find Strong Stackelberg Equilibrium (SSE) [15]:

- the leader plays his best-response strategy,
- the follower plays his best-response strategy,
- the follower breaks ties in favor of the leader.

Game equilibrium may be estimated when all players are choosing the best responses. The best response is possible to find when the relevant optimization problem has the solution. Therefore, it depends on the payoff functions shapes and form. From practical point of view, the numerical methods are used to calculate proper strategies. Not for every game the exact solution may be found.
Calculating leader strategy means finding optimal computing capacities. It is possible if there exists the solution of problem (18) over the convex set $[0,1]^{n-1}$.
Follower strategy is calculated by solving problem (17) and (18). Finding optimal schedule is very time consuming process, therefore the suboptimal schedules are used. In this paper we used genetic algorithm for solving this problem. The detailed solution may be found in [9].
Solving sub-game is much simpler. There is not many types of VMs. Resource may choose finite number from a finite amount of VM types. This is why the solution always exists. It may be calculated using brute-force methods.

9. Numerical Example

To illustrate the presented idea, we consider the example Amazon Cloud and OpenStack VMs. The set VM is presented in Tables 1 and 2. The prices for instances are listed in Table 3. The available number of VM of each type and their computing capacities are presented in Table 4. The batch of tasks consists of 8 tasks, see Table 5. The assumed trust level offered by the virtual resources is $tl_1 = 1$, $tl_2 = 3$.

Table 4
The tested VM instances

Model	Max no.	Strategy	Capacity [GFLOPS]
t2.nano	$m_1(1) = 2$	$MV_1(1)$	$CCVM_1(1) = 3$
t2.small	$m_1(2) = 3$	$MV_1(2)$	$CCVM_1(2) = 3$
t2.xl	$m_1(3) = 2$	$MV_1(3)$	$CCVM_1(3) = 12$
t2.2xl	$m_1(4) = 3$	$MV_1(4)$	$CCVM_1(4) = 24$
m4.16xl	$m_1(5) = 2$	$MV_1(5)$	$CCVM_1(5) = 147$
Gen1-1	$m_2(1) = 3$	$MV_1(1)$	$CCVM_2(1) = 3$
Genl1-2	$m_2(2) = 3$	$MV_1(2)$	$CCVM_2(2) = 5$
Gen1-4	$m_2(3) = 2$	$MV_1(3)$	$CCVM_2(3) = 10$
Genl1-8	$m_2(4) = 2$	$MV_1(4)$	$CCVM_2(4) = 21$

Table 5
The tested batch of 8 tasks

Workload	Security demand
$w_1 = 18$	$sd_1 = 1$
$w = 922$	$sd_2 = 1$
$w_3 = 54$	$sd_3 = 1$
$w_4 = 62$	$sd_4 = 2$
$w_5 = 68$	$sd_5 = 2$
$w_6 = 41$	$sd_6 = 1$
$w_7 = 67$	$sd_7 = 2$
$w_8 = 85$	$sd_8 = 1$

The maximal and minimal capacities are:

$$cc_1^{\min} = 3,$$
$$cc_1^{\max} = 2\cdot 3 + 3\cdot 3 + 2\cdot 12 + 3\cdot 24 + 2\cdot 147 = 405, \quad (29)$$

$$cc_2^{\min} = 3$$
$$cc_2^{\max} = 3\cdot 3 + 3\cdot 5 + 10\cdot 2 + 21\cdot 2 = 86. \quad (30)$$

Considering configurations given by (6) and (7) the possible computing capacity vector may be calculated as:

For $a = 0\ldots 2, b = 0\ldots 3, c = 0\ldots 2, d = 0\ldots 3, e = 0\ldots 2$

$$capacity(a,b,c,d,e) = CCVM_i(1)a + CCVM_i(1)b + CCVM_i(1)c + CCVM_i(1)d + CCVM_i(1)e; \quad (31)$$

$$[cc_i^1, cc_i^2 \ldots, cc_i^{p_i}] = sort\Big(unique\big(capacity(a,b,c,d,e)\big)\Big), \quad (32)$$

where *sort* is sorting procedure. It places the given set of values in the increasing order without repetitions.

For $i = 1,2$

$$p_i = length\Big(sort\big(unique(capacity(a,b,c,d,e))\big)\Big), \quad (33)$$

where *sort* is sorting the vector in descending order, *unique* is erasing repetitive values.

The user of resource one chooses for $p_1 = 114$ computing capacities. The user of resource two chooses one of $p_2 = 72$ computing capacity, see Table 6.

Table 6
Possible computing capacities

$cc_1^{\min}$			.....					$cc_1^{\max}$	
3	6	9	...	177	180	...	399	402	405
$cc_2^{\min}$			.....					$cc_2^{\max}$	
3	5	6	...	47	48	...	81	83	86

The assumed payoff functions are defined as [19]:

$$u_i(j_i) = 400 + 40cc_1(j_1) - \big(cc_1(j_1)\big)^2, \quad (34)$$

for $i = 1,\ j_1 = 1,2,\ldots,114$, and

$$u_i(j_i) = 200 + 70cc_1(j_1) - \big(cc_1(j_1)\big)^2, \quad (35)$$

for $i = 2, j_2 = 1,2,\ldots,72$.

The optimization problem to solve for user of resource 1 is:

$$\max_x \sum_{j_1=1}^{114} \Big(400 + 40cc_1(j_1) - \big(cc_1(j_1)\big)^2\Big)\cdot x_{j_1}^1, \quad (36)$$

with assumption that each mixed strategy is possible:

$$\sum_{j_1=1}^{j_1=114} x_{j_1}^1 = 1, x_{j_1}^1 \in [0,1]. \quad (37)$$

This means that resource 1 tries to find computing capacity that results in maximum CPU credits per CPU unit.
The optimization problem to solve for user of resource 2 is assumed as:

$$\max_x \sum_{j_2=1}^{72} \Big(200 + 70cc_1(j_1) - \big(cc_1(j_1)\big)^2\Big)\cdot x_{j_2}^2 \quad (38)$$

with assumption that each mixed strategy is possible:

$$\sum_{j_2=1}^{j_2=72} x_{j_2}^2 = 1, x_{j_2}^2 \in [0,1]. \quad (39)$$

The objective of resource 2 is to find such computing capacity so that the VMs have the best memory usage per CPU unit.

Matlab *linprog* solver [20] was used for solving both above simple linear programs defined by linear equality constraints. Resource 1 declared capacity $cc_1 = 405$ and resource 2 declared $cc_2 = 48$.

The number of possible schedules without security demands is big. If we consider security demands, then tasks 1, 2, 3, 6 and 8 may be assigned to the resource 1. In such a case, resource 2 may calculate all the given tasks. The number of possible schedules with security demands is only 5. The ETC matrix takes the form:

$$ETC = \begin{bmatrix} 1/405 & 0 \\ 0 & 1/48 \end{bmatrix} \cdot \begin{bmatrix} 18 & 92 & 54 & 62 & 68 & 41 & 67 & 85 \\ 18 & 92 & 54 & 62 & 68 & 41 & 67 & 85 \end{bmatrix}. \quad (40)$$

Considering security constraints:

$$ETC = \begin{bmatrix} 0.044 & 0.22 & 0.13 & \infty & \infty & 0.10 & \infty & 0.20 \\ 0.37 & 1.91 & 1.12 & 1.29 & 1.41 & 0.85 & 1.39 & 1.77 \end{bmatrix}. \quad (41)$$

where ∞ time indicates that scheduling particular task on the chosen resource is impossible.

The possible secure schedules and their makespans are the following:

1. $[t_1, t_2, t_3, t_6]$, $[t_4, t_5, t_7, t_8]$,
   makespan=max{0.044+0.22+0.13+0.10+0.20,
   1.29+1.41+1.39+1.77}= max{0.694, 5.86} = 5.86,
2. $[t_1, t_2, t_3, t_8]$, $[t_4, t_5, t_7, t_6]$,
   makespan=max{0.044+0.22+0.13+0.2,
   1.29+1.41+0.85+1.39 }= max{0.59, 4.94} = 4.94,
3. $[t_8, t_2, t_3, t_6]$, $[t_4, t_5, t_7, t_1]$,
   makespan=max{0.2+0.22+0.13+0.1,
   1.29+1.41+1.39+0.37}= max{0.65, 4.46} = 4.46,
4. $[t_8, t_1, t_3, t_6]$, $[t_4, t_5, t_7, t_2]$,
   makespan=max{0.2+0.044+0.13+0.10,
   1.29+1.41+1.39+1.91}= max{0.47, 6} = 6,
5. $[t_8, t_1, t_2, t_6]$, $[t_4, t_5, t_7, t_3]$,
   makespan=max{0.2+0.044+0.22+0.1,
   1.29+1.41+1.39+1.12}= max{0.56, 5.21} = 5.21.

The optimal schedule is schedule no. 3. For this is schedule the workload for resource 1 is

$$W(1) = w_8 + w_2 + w_3 + w_6 = 85 + 92 + 54 + 41 = 272$$

and for resource 2 is

$$W(2) = w_4 + w_5 + w_7 + w_1 = 62 + 68 + 67 + 18 = 215.$$

After solving problem (32), (33) considering prices from Table 3 and Tab 4, for resource 1:

$$argmin_{[VM_1(1),\ldots,VM_1(5)]} \quad (42)$$

$$\frac{272}{405} \cdot \frac{VM_1(1)\,0.0065 + \ldots + VM_1(5)\,3.83}{3600} \quad (43)$$

and resource 2:

$$argmin_{[VM_2(1),\ldots,VM_2(4)]} \quad (44)$$

$$\frac{215}{72} \cdot \frac{VM_2(1)\,0.0035 + \ldots + VM_2(4)\,0.028}{3600}, \quad (45)$$

with constraints (6) and (8), we obtain the optimum set of VMs:

$$VM_1 = [VM_1(1), VM_1(2), VM_1(3), VM_1(4), VM_1(5)] = \\ = [2,3,2,3,2] \quad (46)$$

and

$$VM_2 = [VM_2(1), VM_2(2), VM_2(3), VM_2(4)] = \\ = [0,2,2,2]. \quad (47)$$

The next round of the game may be based on different assumed payoff functions. The payoffs may depend on the existing VMs, therefore they be the time dependent. Other possibility is that the payoff function of resource 1 may depend on the last VMs set of resource 2. In such a case, the future strategy of resource 1 is influenced by the former strategy of resource 2.

For example, if resource 1 tries to find computing capacity that results in maximum CPU credits per CPU unit. Then, optimization problem to solve for resource 1 is:

$$\max_x \sum_{j_1=1}^{114} \frac{CPUCredits}{vCPU(j_1) \cdot hour(j_1)} x_{j_1}^1, \quad (48)$$

with assumption that each mixed strategy is possible. When the objective of resource 2 is to find such computing capacity so that the VMs have the very best memory per CPU unit:

$$\max_x \sum_{j_2=1}^{72} \frac{RAM(j_2)}{vCPU(j_2)} x_{j_2}^2, \quad (49)$$

under assumption that each mixed strategy is possible.

10. Conclusions and Future Work

The Stackelberg game model can be used for supporting the user of the CC decisions during renting VMs. Proposed game stages are related to the information flow process in cloud. The proposed model enables the usage of on demand recourse provisioning to minimize the computational cost. In the proposed games, the users of virtual resources decide first, so they have special privileges, furthermore, they choose computing capacities. The scheduling unit follows the leaders and decide based on the leader's actions. Third step is the sub-game for virtual resources.

The following elements have been optimized: VMs computing capacities due to virtual resources objectives, sched-

ule for the given task batch, optimum set for that schedule. Additionally, security aspects in the form of mapping tasks security requirements into VMs declared trust level are considered. In presented example for Amazon Cloud and OpenStack VMs we considered the instances provided by the largest cloud services providers. The number of virtual resources and their characteristics can be varied over time. The optimal strategy is calculated for each round of the game.

Different form of payoff function was used. Proposed model allows considering wide variety of virtual resources behavior. One virtual resource payoff may not depend on other virtual resources strategies, or they may be influential. The central scheduler that was used is based on ETC matrix approach and assumes fare share of number of tasks among virtual resources. However, the model may be used with any central scheduler.

The future work will focus on optimizing the trust level of the virtual resources. In this paper trust levels was assumed constant. The more elastic approach should be used.

References

[1] A. Ananth and K. Chandra Sekaran, "Game theoretic approaches for job scheduling in cloud computing: A survey", in *Proc. 5th Int. Conf. on Comp. & Commun. Technol. ICCCT 2014*, Allahabad, India, 2014 (doi: 10.1109/ICCCT.2014.7001473).

[2] A. Jakóbik, D. Grzonka, J. Kołodziej, and H. Gonzalez-Velez, "Towards secure non-deterministic meta-scheduling for clouds", in *Proc. 30th Eur. Conf. on Modell. and Simul. ECMS 2016*, Regensburg, Germany, 2016, pp. 596–602 (doi: 10.7148/2016-0596).

[3] K. Xiong, *Resource Optimization and Security for Cloud Services*. Wiley, 2014.

[4] "Security and Privacy Controls for Federal Information Systems and Organizations", SP 800-53 Rev. 4, National Institute of Standards & Technology, 2013 (doi: 10.6028/NIST.SP.800-53r4).

[5] "NIST Cloud Computing Security Reference Architecture", SP 500-299, National Institute of Standards & Technology, 2013.

[6] ISO/IEC 19790:2012 "Security requirements for cryptographic modules", International Organization for Standardization, 2012.

[7] ISO/IEC 19790:2012 "Information technology, Security techniques, Security requirements for cryptographic modules", ISO Council, Switzerland, Geneva.

[8] A. Jakóbik, "A cloud-aided group RSA scheme in Java 8 environment and OpenStack software", *J. Telecommun. and Inform. Technol.*, no. 2 pp. 53–59, 2016.

[9] A. Jakóbik, D. Grzonka, and F. Palmieri, "Non-deterministic security driven meta scheduler for distributed cloud organizations", *Simul. Modell. Practice & Theory*, 2016 [Online]. Available: http://dx.doi.org/10.1016/j.simpat.2016.10.011.

[10] A. Ananth and K. Chandrasekaran, "Cooperative game theoretic approach for job scheduling in cloud computing", in *Proc. Int. Conf. on Comput. and Netw. Commun. CoCoNet'15*, Trivandrum, India, 2015 (doi: 10.1109/CoCoNet.2015.7411180).

[11] M. Geethanjali, J. Sujana, and T. Revathi, "Ensuring truthfulness for scheduling multi-objective real time tasks in multi cloud environments", in *Proc. Int. Conf. on Recent Trends in Inform. Technology ICRTIT 2014*, Chennai, India, 2014 (doi: 10.1109/ICRTIT.2014.6996183).

[12] X. Qiu, C. Wu, H. Li, Z. Li, and F. C. M. Lau, "Federated private clouds via broker's marketplace: A Stackelberg-game perspective", in *Proc. 7th IEEE Int. Conf. on Cloud Comput. CLOUD 2014*, Anchorage, Alaska, USA, 2014 (doi: 10.1109/CLOUD.2014.48).

[13] M. Shie, C. Liu, Y. Lee, Y. Lin, and K. Lai, "Distributed scheduling approach based on game theory in the federated cloud", in *Proc. Int. Conf. on Inform. Science & Appl. ICISA 2014*, Seoul, South Korea, 2014 (doi: 10.1109/ICISA.2014.6847388)

[14] S. Tadelis, *Game Theory: An Introduction*. Princeton University Press, 2013.

[15] A. Wilczyński, A. Jakóbik, and J. Kołodziej, "Stackelberg security games: Models, applications and computational aspects", *J. Telecommun. and Inform. Technol.*, no. 3, pp. 70–79, 2016.

[16] V. Mazalov, *Mathematical Game Theory and Applications*. Wiley, 2014.

[17] J. Kołodziej, *Evolutionary Hierarchical Multi-Criteria Metaheuristics for Scheduling in Large-Scale Grid Systems*. Springer, 2012.

[18] J. Gan and B. An, "Minimum support size of the defender's strong Stackelberg equilibrium strategies in security games", in *AAAI Spring Symp. on Applied Computat. Game Theory*, Stanford, CA, USA, 2014 [Online]. Available: http://www.ntu.edu.sg/home/boan/papers/AAAISS14b.pdf

[19] A. Mas-Colell, M. D. Whinston, and J. R. Green, *Microeconomic Theory*. Oxford University Press, 1995.

[20] MathWorks, Documentation [Online]. Available: https://www.mathworks.com/help/optim/ug/linprog.html



Andrzej Wilczyński is an Assistant Professor at Cracow University of Technology and Ph.D. student at AGH University of Science and Technology. The topics of his research are multiagent systems and cloud computing.

E-mail: and.wilczynski@gmail.com
AGH University of Science and Technology
Mickiewicza st 30
30-059 Cracow, Poland

Tadeusz Kościuszko Cracow University of Technology
Warszawska st 24
31-155 Cracow, Poland

Agnieszka Jakóbik (Krok) – for biography, see this issue, p. 64.

Paper

Data Fixing Algorithm in Radiosonde Monitoring Process

Piotr Szuster

Faculty of Physics, Mathematics and Computer Science, Cracow University of Technology, Cracow, Poland

Abstract—Earth surface monitoring can give information that may be used in complex analysis of the air conditions, temperature, humidity etc. Data from a vertical profile of the atmosphere is also essential for accurate thunderstorm forecasting. That data is collected by radiosondes – telemetry instruments carried into the atmosphere usually by balloons. Sometimes, due to the hostile conditions of upper troposphere, incorrect data can be generated by radiosonde sensors. In this paper, a new algorithm is developed for fixing the incorrect data, i.e. missing or out of specific range values. The proposed algorithm was tested both on benchmarks and real data generated by radiosondes. About 70% of significantly damaged test data volume was recovered. Up to 100% of real data was fixed.

Keywords—aerology, data fixing, data fusion, meteorological data, monitoring, radiosonde.

1. Introduction

Radiosonde monitoring delivers important information about convective available potential energy (CAPE), storm relative helicity (SRH), lifted condensation level (LCL) or wind shears at different altitudes [1]. The combination of specific values of these parameters can indicate high probability of the development of a tornadic supercell storm (high CAPE in area of high SRH accompanied by low LCL). In Figs. 1 and 2 the most important thermodynamic and kinematic parameters related to the tornadic supercell are presented, which has happened on August 15, 2008 near Strzelce Opolskie, Poland.

Thermodynamical data generated by radiosondes can be represented in graphical Skew-T log-P diagrams (shown in Fig. 1) and Stuve diagrams [2]. The hodograph [3] is shown in Fig. 2. Hodograph is a diagram for graphical representation of the wind velocity vectors.

Information delivered by radiosonde monitoring is important for providing meteorological warnings and post incidental case studies. Upper air monitoring is also source of data for the numerical weather prediction models, which are based on the concept of the characteristics of the atmosphere as a fluid. That data is also useful in the analysis of a vertical profile of the atmosphere. The general equations of fluid dynamics and thermodynamics are used for the estimation of atmosphere states at the specific time slots [4]. Those equations are sensitive on the input data errors. The quality of that data should be as high as possible and numerical weather prediction models solve those dynamic and thermodynamic equations. The obtained numerical results are presented to forecasters in a form of maps and charts in order to aid the process of the weather forecast. The

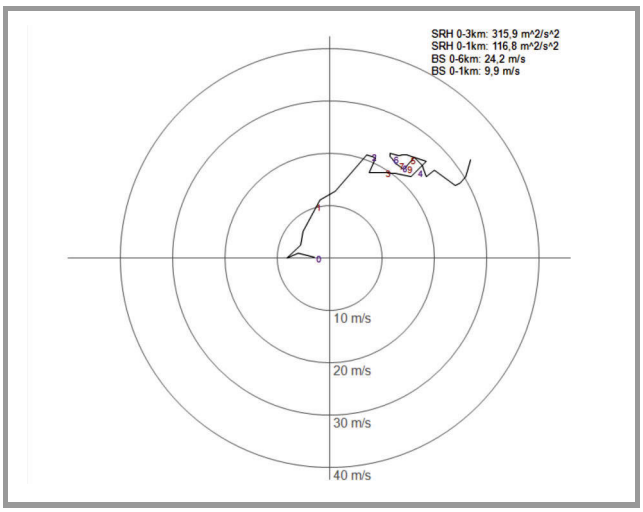


Fig. 1. Data from Poprad-Ganovce radiosonde monitoring, gathered on August 15, 2008, plotted on hodograph. Characteristic for a tornado shape of wind profile is visible.

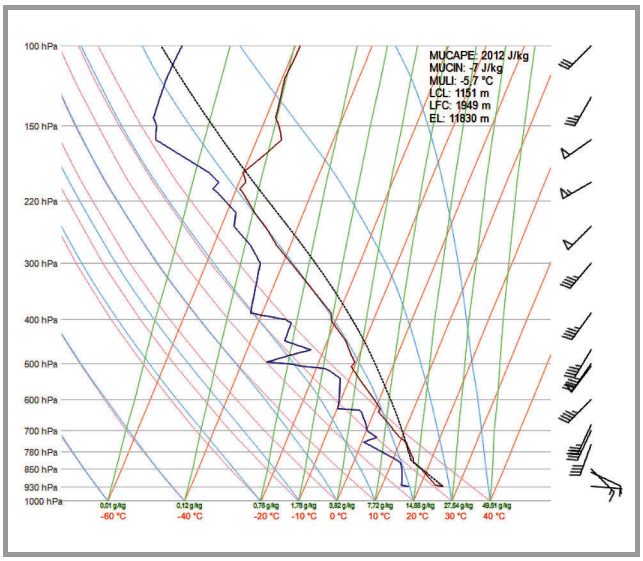


Fig. 2. Data from Poprad-Ganovce radiosonde monitoring, gathered on August 15, 2008, plotted on Skew-T log-P diagram. Large convective available potential energy is clearly visible. (See color pictures online at www.nit.eu/publications/ journal-jtit)

map presented in Fig. 3 defines the spatial distribution of the various parameters used in solving the above-mentioned analytical models.

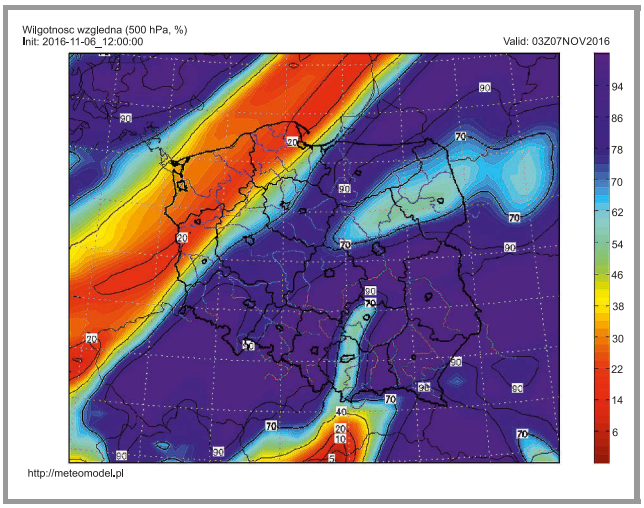


Fig. 3. Spatial distribution of relative humidity at 500 hPa height level for Poland. Numerical forecast generated by GFS model for November 7, 2016 03Z. Initial conditions taken from November 6, 2016 12Z.

There are many sources of input data (settings) in numerical prediction models. Such data is generated during the monitoring processes provided in irregularly distributed physical locations. Therefore, the generated data must be merged and analyzed based on the specified quality requirements. The major challenge is the large number of data sources, active monitoring processes executed simultaneously, and large data volumes. There is a need of the development of the new efficient algorithms and tools for data repairing and data fusion, which are necessary for providing data analysis. All those processes should be automated and independent on the administrators and all type of computational platforms.

In this paper an algorithm for detecting the damaged data records and repairing. Proposed methodology is based on fast interpolation data matrix structures stored at the data servers.

The paper is organized as follows. In Section 2 the data sources are specified together with the definition of the data matrices. In Section 3 the existing methods used in data healing are compared and in processing of the vast datasets. The computational and implementation aspects of the proposed models are discussed in Sections 4 and 5. Section 6 concludes the paper.

2. Data Characteristics

Radiosonde upper air monitoring is the most popular method of the data gathering for the analysis of weather conditions in the whole troposphere. In order to provide the proper analysis of weather condition, there is a need to obtain the complete data from such monitoring process without incorrect (missing or out of range) values. It becomes necessary to take a close look into radiosonde monitoring data that is distributed by the University Wyoming in the public repository [5]. In that repository, the aerological data is collected in most of countries. Radiosonde monitoring is often interpreted as sounding process, and it is performed twice a day. The data gathered during the monitoring can be saved as raws records, text list, and plain text. The graphical representations of such data structures are tephigrams and hodographs stored as GIF and PDF files.

The easiest format for the data analysis is the text list format presented in Table 1. In such case, the dataset is defined as a matrix. The columns represent the atmosphere attributes: pressure (P), height (H), temperature (T), dew point (D), relative humidity (RH), mixing ratio (ω), wind direction (A), wind speed (S), potential temperature (θ_a), equivalent potential temperature (θ_e) and virtual potential temperature (θ_v) [6]. In this paper, the attributes RH, ω, θ_a, θ_e and θ_v are ignored. Therefore, the number of columns of matrix is reduced to five. The exemplary values of such parameters are shown in the Table 2.

Table 1
The example of soundings data format after unnecessary cells reduction

P [hPa]	H [m]	T [°C]	D [°C]	A [°]	S [kt]
925.0	888	3.6	–15.4	0	0
897.0	1141	9.8		150	16
700.0	3167	–0.1	–5.0	165	16
500.0	5760			300	50
300.0	9290	–49.3	0.0	65	10
100.0	16110	–66.3		310	16

Table 2
The example of sounding data values after correction performed by proposed algorithm

P [hPa]	H [m]	T [°C]	D [°C]	A [°]	S [kt]
925.0	888.0	3.6	–15.4	0	0
897.0	1141.0	9.8	–14.1	150.0	16.0
700.0	3167.0	–0.1	–5.0	165.0	16.0
500.0	5760.0	–10.2	–10.2	300.0	50.0
300.0	9290.0	–49.6	–49.6	65.0	10.0
100.0	16110.0	–66.3	–66.3	310.0	16.0

Each row of the data matrix defines different altitude of the measurement performed by the ascending radiosonde. Rows are sorted in ascending order based on the altitude parameters and in descending order based on the pressure

parameters. Table 2 presents exemplary seven rows of the matrix with six cells each. The first row contains complete data from the monitoring and the range of the data values is correct. The second row has missing data in the fourth cell (dew point). The dew point data at the fourth and sixth rows are also missing. The fifth row does not contain both the temperature and the dew point parameters. The fourth row has missing temperature parameter.

2.1. Data Corruption

In existing weather forecasting systems, three main methods are implemented for the healing of the missing data of sounding, namely:

- the elimination of rows with incorrect values,
- manual repair of the damaged data,
- interpolation of the damaged data.

The correct data can be lost if all the rows, which contain incorrect values are removed. The rows can consist of both damaged and correct data.
A simple example of sounding with missing data is presented in [7].

2.2. Data Volume and Velocity

Data generated by different radiosondes is considered as representation of the atmosphere states at different time slots and different physical locations. In order to provide the fair analysis of the atmosphere in Poland, there is a need to collect the data from radiosonde monitoring provided in the neighbor countries, i.e. in Germany, Czech Republic, Slovakia, Ukraine, Belarus and Russia. The generated data matrix for such monitoring consists of up to 130 rows. Each row has six significant cells.
In most cases, the monitoring at aerological stations is provided twice a day (at 00Z and 12Z). By committing a single query data from one station, gathered within one month (maximum 31 days) can be requested. At time up to 62 soundings containing approximately 8060 rows can be obtained from Wyoming University's database. Finally, there are typically 62 soundings (about 50,000 values to check). Assuming that 25% of the values are damaged, there are approximately 12,500 cells to cure in order to heal soundings from a whole month.
Manual data repairing is a time-consuming process. In that case, every corrupted data record should be detected corrected. Software tools designed for manual data repairing require the involvement of the users and administrators who must assist the whole data healing. For large soundings with data matrices (over 100 rows), such process must be automated. The efficient method data healing can be based on the interpolation method.

3. Data Fixing Algorithms

In [8] and [9], the authors analyze the reasons of missing data and define the following three possible "data missingness" scenarios:

- missingness completely at random (MCAR),
- missingness at random (MAR),
- missingness not at random (MNAR).

Statistical analysis can be useful in solving the missing data problems [9]. The statistical approach shown in [9] can be classified according to the different criteria. First criterion is data shifting, which can be realized as data discarding or data retaining.
Three other methods defined as data discarding analysis:

- complete-case analysis,
- available-case analysis,
- non-response weighting.

In complete-case analysis, all the rows or columns of radiosonde data matrix with missing or incorrect data are removed. This will result in the loss of correct information because if there is a row with one missing data, such row is removed from the data matrix.
In the available-case, the columns of data matrix are grouped into two categories: damaged and correct, and next the damaged values are removed. The remaining columns are suitable for the further analysis. In non response weighting, the columns classified as "correct" in complete-case analysis are reweighted (i.e. the values of the parameters are reweighted). Reweighting is made in order to restore the proper representation of the parameters. In most of the cases, the loss of data is very undesirable and data retaining methods can be more effective. Such methods are based on imputation – filling gaps in data records. The most popular imputation methods include:

- mean imputation,
- last value carried forward,
- information from related monitoring dependent,
- using indicator variables for missingness,
- imputation based on logical rules, and others.

In mean imputation, the missing data values are filled with mean values of remaining data. In last value carried forward, the gaps in data records are filled with the data from predecessor cells. In order to fix data information from related monitoring also could be used, but there is a need to recognize relations between them and a degree of misrepresentation. Those methods could be called information from related monitoring dependent. Methods that are using indicator variables are based on concept of creation of

extra category, that has an information about missingness. Some solution also come from logical rules, for example if there is no measurement, its value is set to zero.
The problem is more complex when more than one data variable is missing. One joint general model of imputation can be defined for all the variables. In addition, different models for each column could be applied.
All the methods mentioned above are based on the statistical subsets and performed in order to preserve the proper representation of the set and original probability distribution.
In the single sounding case, probability distribution of different values cannot be considered. There is only vertical distribution so all the methods are unsuitable. For example, mean imputation of filling missing values with mean can lead to nonsense. If mean of temperature during day is equal to -5°C, the missing value occurs at ground level and there is a summer, which is usually warm, it is leading to nonsense.

3.1. Traditional Interpolation Methods

Traditional interpolation methods can be also useful in data healing. In that case, reference points that have correct values are defined as interpolation nodes. The useful in data healing interpolation methods include:

- piecewise linear interpolation,
- Lagrange interpolation,
- spline interpolation.

The easiest way to interpolate data in tabular set is to replace a real function that delivers value of each pair of the set with a linear function [10]. In this case, there are two discrete sets: X – set of n arguments and Y – set of n values. Hence, n following samples can be generated:

$$(x_i, y_i): \quad x_i \in X, y_i \in Y, i \in 1, \ldots, n. \tag{1}$$

Let us define the function $f(x_i) = y_i$, which can be approximated by the linear function $g(x)$:

$$g(x) = ax + b. \tag{2}$$

For each set of the following parameters $x_i, y_i, x_{i+1}, y_{i+1}$ for $i \in 1, \ldots, n-1$, the following conditions are specified:

$$g(x_i) = y_i, \tag{3}$$

$$g(x_{i+1}) = y_{i+1}. \tag{4}$$

The function $g(x)$ can approximate $f(x)$ within the range $[x_i, x_{i+1}]$. Now the analytic formulas for a and b in the following way can be defined:

$$a = \frac{y_{i+1} - y_i}{x_{i+1} - x_i}, \tag{5}$$

$$b = y_i - a \cdot x_i. \tag{6}$$

The main advantages of using the piecewise linear interpolation can be formulated as:

- it requires only two complete samples to create the approximation for the range between them,
- it is of a low computing task and power efficient,
- it is easy for the implementation.

The main disadvantages of using the piecewise linear interpolation is:

- it is not smooth,
- for sparse sets and complex functions there can be large error occurring.

In the Lagrange interpolation [11], in order to define the n-th Lagrange interpolation polynomial, we have to:

- define the basis polynomials,
- define the Lagrange interpolation polynomial.

For $n+1$ discrete points (x_i, y_i) for $i \in 0, \ldots, n$ the basis polynomial $B_i(x)$ is defined as:

$$B_i(x) = \prod_{\substack{0 \le k \le n \\ k \ne i}} \frac{x - x_k}{x_i - x_k}. \tag{7}$$

Lagrange interpolation polynomial is defined in the following way:

$$L(x) = \sum_{i=0}^{n} y_i B_i(x). \tag{8}$$

The Lagrange interpolation is numerically unstable method due to Runge's phenomenon [12]. The computational complexity of that method is $O(n^2)$. All those makes this method rather ineffective in solving data-healing problems.
The third considered interpolation method is spline interpolation. It is based on the polynomials that have degrees lower than the number of samples. There are $n+1$ discrete points generated (x_i, y_i) for $i \in 0, \ldots, n$. The interpolation between all the pairs of the set is defined by polynomials $q_i(x)$ for $i \in 1, \ldots, n$. The following conditions should be satisfied:

$$\begin{cases} q_i'(x_i) = q_{i+1}'(x_i) \\ q_i''(x_i) = q_{i+1}''(x_i) \end{cases}. \tag{9}$$

The classical approach is to use cubic splines with third degree polynomials [13]. A spline interpolation is a very smooth and avoids Runge's phenomenon. In case of interpolation of aerological data smoothness is not desirable. For example large error can exist if there is sudden and large temperature inversion. In this case, conditions shown in Eq. (9) can lead to a large error.

In case of soundings there is a multivariate interpolation problem. The simplest and most promising way seems to be a usage of the piecewise linear interpolation.

3.2. Modeling Using Soft Computing

Another approach to solve missing data problem is to use soft computing components such as Support Vector Machine (SVM) or fuzzy logic.
The use of SVM model is described in [14]. General idea consists of few steps: at first we have to choose some measurements that do not have missing values, then select input, output attributes and decision attributes and finally apply regression model.
In [15] the authors proposed other methods of data imputation, i.e. *k*-nearest neighbor imputation (KNNI) that uses data from the most similar neighbors by imputing mean value from their monitoring. The second interesting way is to use value that comes from estimated distribution of missing value. That method is called hot deck imputation. The third manner of doing data curing is to use predictive model to estimate values that are missing. In that way attributes of missing value are used as the response attribute. The rest of them are used to create the model as an input. The authors also proposed a use of decision trees by using built-in approaches.
The soft computing methods mentioned above are more complex than the interpolation methods. Creating different predictive or supervised learning model for each sounding is in fact an ineffective way of data healing.

4. Proposed Methodology

The proposed algorithm use sounding in text list format. Then two stages correction process that consists is performed.

4.1. Validation Stage

At first stage validation is performed. Described in the first section data structure (matrix like) input is divided into m lines l_m. There has to be lexicographical order between all the lines in the set. Each line $l_k \quad k = 1,\ldots,m$ is split into n cells c_n. Cells that are not necessary (relative humidity, mixing ratio, thta, thte, thtv) are now removed so each line l_k consists of $n-5$ cells. That is lexical analysis part. In the next step each cell is verified if it has a value and if the value consists of permissible characters. That is syntax analysis. At this time the matrix of validity $X = m \times n-5$, the matrix of values $V = m \times n-5$ and $\vec{x}$ of invalid rows indexes are created:

$$X_{k,o} = S(l_k, o)\,, \tag{10}$$

where S is a syntax error function, l_k is k-th line and o is o-th cell in k-th line. When

$$l_k = [c_1,\ldots,c_{n-5}]\,, \tag{11}$$

then

$$S(l_k,o) = \begin{cases} 2 & \text{if } c_o \text{ consists of illegal characters} \\ 1 & \text{if } c_o \text{ is empty} \\ 0 & \text{otherwise} \end{cases}. \tag{12}$$

$V_{k,o}$ is set to zero if $S(l_k,o)$ is not equal to zero else $V_{k,o} = l_{k_o}$ then $X_{k,o} = R_o(V_{k,o})$ where R_o is o-th range error function. The o-th means that for different column of value matrix different correct range $[r_{o_l}, r_{o_u}]$ of values is defined.

$$R_o(V_{k,o}) = \begin{cases} 0 & \text{if } V_{k,o} \in [r_{o_l}, r_{o_u}] \\ 3 & \text{otherwise} \end{cases}, \tag{13}$$

For each k-th row of matrix X following sum is calculated:

$$\sum_{o=1}^{n-5} X_{k,o}\,, \tag{14}$$

and if the sum is greater than zero to vector $\vec{x}$ at the end element equals k is added.

4.2. Curing Stage

Each element of vector $\vec{x}$ contains index of the values matrix V row that is composed of cells that have damaged values. At this stage for each index stored in $\vec{x}$ algorithm is finding two nearest rows (curing rows) that have correct values that are damaged in cured row and at least one value that is correct in both cured and curing rows. The vector $\vec{u} = X_k$ is the nearest to $\vec{b} = X_{x_i}$ when difference between k and x_i is minimal.
Because there is the matrix of validity X that was created in previous stage, for each value x_i stored in $\vec{x}$ algorithm is analyzing each row of matrix of validity X:

$$\vec{b} = X_{x_i}\,. \tag{15}$$

Each value $\vec{b}_j$ of row $\vec{b}$ cells is now converted to Boolean form:

$$\vec{c}_j = B(\vec{b}_j) = \begin{cases} \text{true} & \text{if} \quad \vec{b}_j = 0 \\ \text{false} & \text{otherwise} \end{cases}. \tag{16}$$

Then for each false value $\vec{c}_j$ in $\vec{c}$ algorithm searches for the two nearest rows of X $\vec{f} = X_k, \vec{l} = X_l \wedge k \neq l \wedge k, l \in 1,\ldots,m$ that fits criteria:

$$\vec{f}_j = 0 \wedge \vec{l}_j = 0\,. \tag{17}$$

And after creation of vector of comparison $c\vec{m}p$ such as:

$$c\vec{m}p_j = \vec{c}_j \wedge B(\vec{f}_j) \wedge B(\vec{l}_j)\,. \tag{18}$$

The $c\vec{m}p$ has at least one true value. Two first vectors $\vec{f}, \vec{l}$ that are meeting mentioned requirements are chosen to be curing vectors.
For the chosen curing vectors $\vec{f} = X_k, \vec{l} = X_l$ algorithm is getting corresponding values vectors from V: $\vec{o} = V_k, \vec{p} = V_l$. Then if $l > k$ swapping is performed:

$$\vec{o} = \vec{p} \qquad \vec{p} = \vec{o}, \tag{19}$$

in order to maintain the ascending order. Then index z of the first cell of $c\vec{m}p$ that has true value is obtained and finally:

$$V_{x_i,j} = Vx_i,z \cdot \frac{\vec{p}_j - \vec{o}_j}{\vec{p}_z - \vec{o}_z} + \vec{o}_j - \vec{o}_z \cdot \left(\frac{\vec{p}_j - \vec{o}_j}{\vec{p}_z - \vec{o}_z}\right). \quad (20)$$

The missing value is interpolated by using of linear interpolation.

4.3. Algorithm Limitations

Developed algorithm has some limitations. The first limitation is that each damaged row of sounding has to have at least one correct value in order to make curing possible. The second limitation is that each sounding column has to have at least two correct values. At least one column row has to have lexicographical order.
Assuming that the number of damaged lines m will be significantly lower than the number of whole lines of sounding n $(m << n)$ the theoretical computational complexity will be $O(mn^2)$ in the worst case. The typical sounding gathered in the USA has up to 130 lines.

5. Algorithm Implementation

The developed algorithm was implemented in C# language and was integrated with Wyoming University Data Repository.

5.1. Validation Stage

The first step is to split each line into cells and to remove those that are not necessary.

```
p_ = line.Substring(1, 6);
p_ = p_.Replace(" ", String.Empty);
h_ = line.Substring(9, 5);
h_ = h_.Replace(" ", String.Empty);
t_ = line.Substring(16, 5);
t_ = t_.Replace(" ", String.Empty);
d_ = line.Substring(23, 5);
d_ = d_.Replace(" ", String.Empty);
a_ = line.Substring(46, 3);
a_ = a_.Replace(" ", String.Empty);
v_ = line.Substring(52, 4);
v_ = v_.Replace(" ", String.Empty);
```

The second step is to perform syntax analysis.

```
vp = tryParseDouble(p_, out p);
vh = tryParseDouble(h_, out h);
vt = tryParseDouble(t_, out t);
vd = tryParseDouble(d_, out d);
va = tryParseDouble(a_, out a);
vv = tryParseDouble(v_, out v);
```

Then correct range is checked and matrix of validity is created.

```
if (p < 0)
lineStatus.Add(LineStatus.PRES_ERR);
if (t < -273.15 || t > 100.0)
lineStatus.Add(LineStatus.TEMP_ERR);
if (d > t || d < -273.15 || d > 100.0)
lineStatus.Add(LineStatus.DEW_ERR);
if (a < 0 || a > 360)
lineStatus.Add(LineStatus.ANG_ERROR);
if (v < 0)
lineStatus.Add(LineStatus.VEL_ERROR);
if (!vp)
lineStatus.Add(LineStatus.PRES_MISSING);
if (!vh)
lineStatus.Add(LineStatus.HEI_MISSING);
if (!vt)
lineStatus.Add(LineStatus.TEMP_MISSING);
if (!vd)
lineStatus.Add(LineStatus.DEW_MISSING);
if (!va)
lineStatus.Add(LineStatus.ANG_MISSING);
if (!vv)
lineStatus.Add(LineStatus.VEL_MISSING);
```

5.2. Curing Stage

When the validation stage is completed then for each damaged cell in each damaged row indexes of curing rows are set to be out of range, index of healed row is copied to temporary variable and row of matrix of validity is converted to Boolean form.

```
bool[] cmp = null;
int np = -ylen-1; int nnp = -ylen-1;
int dmgInd = j;
bool[] cdmg;
cdmg = damagedColumns(sList[dmgInd]);
```

Then the first curing row is chosen to be checked.

```
for
(k = 0; k < soundingList[i].Count; k++){
bool[] dmg = null;
bool[] dmg2 = null;
if (k!=dmgInd){
dmg = damagedColumns(sList[k]);
```

Then the second curing row is chosen to be checked.

```
for
(l = 0; l < soundingList[i].Count; l++){
if (l != k&&l!=dmgInd){
dmg2 = damagedColumns(sList[k]);
int ccp;
```

And next comparison is the made.

```
int ccp;
bool[] comparison =
compareDamaged(dmg, dmg2,
cdmg, out ccp, i);
```

If both chosen rows meet requirements.

```
if (ccp >= 1 &&
Math.Abs(dmgInd - k)
< Math.Abs(dmgInd - np)
&& Math.Abs(dmgInd - l)
< Math.Abs(dmgInd - nnp))
```

Rows are marked as curing rows.

```
np = k;
nnp = l;
cmp = comparison;
```

Then swapping is performed.

```
int f, n;
if (np < nnp){
    f = np; n = nnp;
}
else{
    f = nnp; n = np;
}
```

Appropriate values are chosen to cure the damaged one.

```
double y1 = soundingList[i][f];
double y2 = soundingList[i][n];
double x1 = 0;
double x2 = 0;
double gx = 0;
```

Then the argument of linear function is chosen basing on information from vector of comparison.

```
for (int m = 0; m < cmp.Length; m++){
if (m != i && cmp[m] == true){
x1 = soundingList[m][f];
x2 = soundingList[m][n];
gx = soundingList[m][dmgInd];
break;
}
}
```

Finally, an interpolation is made and the value is inserted into values matrix.

```
double val =
MathHelper.interpolateLinear
(x1, y1, x2, y2, gx);
soundingList[i][j] = val;
```

Note that *i* is index of column and *j* is row index.

5.3. Tests

The proposed algorithm tests were divided into two stages. At the first stage they were performed on test data that was prepared for it (Table 3). Tests were performed as unit tests.

Table 3
The example of soundings dataset prepared for testing

P [hPa]	H [m]	T [°C]	D [°C]	A [°]	S [kt]
1014.0	96.0	1.0	0.3	250.0	8.0
850.0	1499.0	–6.3	–6.3	295.0	25.0
700.0	3002.0		–13.9	320.0	27.0
500.0	5520.0	–24.5		335.0	43.0
400.0	7100.0	–37.7	–41.1	340.0	45.0
300.0	9020.0		–57.4	345.0	56.0
200.0	11550.0	–58.9		330.0	39.0
100.0		–60.5	–87.5	330.0	31.0

Table 4
The example of soundings data set prepared for testing after correction made by proposed algorithm

P [hPa]	H [m]	T [°C]	D [°C]	A [°]	S [kt]
1014.0	96.0	1.0	0.3	250.0	8.0
850.0	1499.0	–6.3	–6.3	295.0	25.0
700.0	3002.0	–13.0	–13.9	320.0	27.0
500.0	5520.0	–24.5	–24.5	335.0	43.0
400.0	7100.0	–37.7	–41.1	340.0	45.0
300.0	9020.0	–50.9	–57.4	345.0	56.0
200.0	11550.0	–58.9	–73.7	330.0	39.0
100.0	14080.0	–60.5	–87.5	330.0	31.0

During the second stage of testing algorithm was connected to component that was responsible for the acquiring data from Wyoming University's website, so tests were performed on real data from radiosondes (Table 4). To make an outcome comparison, the algorithm was also connected to components of data edition and diagram generation created by the author. Also some vital thermodynamical and kinematical indices were calculated. The test sets for that stage were made from well-known soundings, which are related to famous severe weather situations. In that manner the comparison between actual outcome and expected values can be performed.

5.4. Test Results

The most of the real data examples were slightly damaged so the proposed algorithm recovered 100% of missing or damaged data in most cases. Test examples prepared for the first phase of testing were significantly damaged by the author. For example, the whole column was missing or some rows have only one correct value. There were also combinations of the missing columns and rows. About 70% of prepared data set examples were able to be used in processing, e.g. creating diagrams, computing thermodynamical indices, etc. "The example was able to be used"

means that after a correction, the outcome from processing was not misleading user – it does not result in incorrect assessment of the weather situation. In all the test cases runtime of algorithm is significantly lower than runtime of processing routines. In case of remote data usage, the time of the whole procedures was dominated by remote data obtaining time, thus confirmed proposed model's efficiency.

6. Conclusions

The developed algorithm that bases on linear interpolation in the most cases repairs all the available data. It can be classified as an imputation algorithm. It applies piecewise linear interpolation into multivariate model. During curing two the most proper values are chosen in order to heal a damaged cell so it is also related to the nearest neighbor idea. Due to its low runtime and easy understandable concept, it can be adapted to other fields that use data structures similar to that described in Section 2. The author is interested in data fusion, especially in fusion of meteorological data, so the algorithm will be used in cases that will require of a use of soundings data. The algorithm will be also used in the next versions of Sounding Decoder – the author's software solution dedicated to sounding processing.

References

[1] R. Thompson, "Explanation of SPC severe weather parameters", Storm Prediction Center, National Oceanic and Atmospheric Administration [Online]. Available: http://www.spc.noaa.gov/sfctest/help/sfcoa.html

[2] D. Dempsey, "The Stuve and Skew-T/Log-P thermodynamic diagrams", Dept. of Geosciences, San Francisco State University, 2009 [Online]. Available: http://funnel.sfsu.edu/courses/metr201/S12/handouts/stuve_skewt.html

[3] T. A. Apostolatos, "Hodograph: A useful geometrical tool for solving some difficult problems in dynamics", *Amer J. of Phys.*, vol. 71, no. 3, pp. 261–266, 2003 [Online]. Available: http://users.uoa.gr/∼pjioannou/mech1/SITES/hodograph.pdf (doi: 10.1119/1.1527948).

[4] W. C. Skamarock *et al.*, "A Description of the Advanced Research WRF Version 3", National Center for Atmospheric Research Boulder, CO, USA, Jun. 2008 [Online]. Available: http://www2.mmm.ucar.edu/wrf/users/docs/arw_v3.pdf

[5] "Atmospheric Soundings", University of Wyoming [Online]. Available: http://weather.uwyo.edu/upperair/sounding.html

[6] "Description of Sounding Columns", University of Wyoming [Online]. Available: http://weather.uwyo.edu/upperair/columns.html

[7] "33791 Kryvyi Rih monitoring at 12Z 01 Dec 2016", University of Wyoming [Online]. Available: http://weather.uwyo.edu/cgi-bin/sounding?region=europe&TYPE=TEXT%3ALIST&YEAR=2016&MONTH=12&FROM=0112&TO=0112&STNM=33791.

[8] T. D. Pigott, "A review of methods for missing data", *J. of. Educational Res. and Eval.*, vol. 7, no. 4, pp. 353–383, 2001.

[9] A. Gelman and J. Hill, *Data Analysis Using Regression and Multilevel/Hierarchical Models*, 1st ed. Cambridge University Press, 2006, pp. 529–543.

[10] T. Blu, "Linear interpolation revitalized", *IEEE Trans. on Image Process.*, vol. 13, no. 5, pp. 710–719, 2004 (doi: 10.1109/TIP.2004.826093).

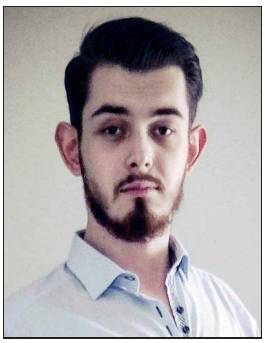
[11] J.-P. Berrut and L. N. Trefethen, "Barycentric Lagrange interpolation", *SIAM Rev.*, vol. 46, no. 3, pp. 501–517, 2004.

[12] C. Runge, "Über empirische Funktionen und die Interpolation zwischen aquidistanten Ordinaton", *Zeitschrift für Mathematik und Physik*, vol. 46, pp. 224–243, 1902 (in German).

[13] G. Wolberg, "Cubic spline interpolation: A review", Tech. Rep. CUCS-389-88, Columbia University, Computer Science Reports, New York, USA, 1988.

[14] F. Honghai *et al.*, "A SVM Regression Based Approach to Filling in Missing Values", in *Knowledge-Based Intelligent Information and Engineering Systems*, R. Khosla, R. J. Howlett, and L. C. Jain, Eds. *LNCS*, vol. 3683, pp. 581–587. Springer, 2005.

[15] E. Acuña and C. Rodriguez, "The treatment of missing values and its effect in the classifier accuracy", in *Classification, Clustering, and Data Mining Applications*, D. Banks *et al.*, Eds. Springer, 2004 [Online]. Available: http://sci2s.ugr.es/sites/default/files/files/TematicWebSites/MVDM///IFCS04r.pdf



Piotr Szuster graduated in Computer Science at Cracow University of Technology, Poland, in 2016. Currently, he is a research and teaching assistant at Cracow University of Technology and a Ph.D. student at AGH University of Science and Technology. The main topics of his research are Big Data, Data Fusion and Internet of Things.

E-mail: Piotr.Szuster@pk.edu.pl
Faculty of Physics, Mathematics and Computer Science
Cracow University of Technology
Warszawska st 24
31-155 Cracow, Poland

AGH University of Science and Technology
Mickiewicza av. 30
30-059 Cracow, Poland

Paper

My City Dashboard: Real-time Data Processing Platform for Smart Cities

Catalin-Constantin Usurelu[1] and Florin Pop[1,2]

[1] *Faculty of Automatic Control and Computers, University Politehnica of Bucharest, Bucharest, Romania*
[2] *National Institute for Research and Development in Informatics (ICI), Bucharest, Romania*

Abstract—In recent years, with the increasing popularity of IoT, the rapid growth of smartphone usage enabled by the increase adoption of Internet services and the continuously decreasing costs of these devices and services has led to a huge increase in the volume of data that governments can use in the context of smart city initiatives. The need for analytics is becoming a requirement for smart city projects such as city dashboards to provide citizens with an easy to understand overview of the city. As such, data should be analyzed, reduced and presented in such a way that citizens can easily understand various aspects of the city and use this information to increase quality of life. In this paper, we firstly present the context and the start of the design and implementation of proposed solution for real-time data processing in smart cities, mainly an analytics processing pipeline and a dashboard prototype for this solution, named My City Dashboard. We focus on high scalability and modularity of this platform.

Keywords—big data, data analytics, real-time processing, smart cities, statistics.

1. Introduction

Because the use of sensors is not always feasible given the inaccessibility of locations, lack of a complete understanding of where to best gather data from and costs, an avenue worth exploring is that of crowdsources initiatives. These entail citizen participations resulting in no costs (citizens don't directly benefit), the advantage of human decision making related to what data to collect and from where and sometimes better accuracy and fault detection compared to sensors [1]. One such approach is presented in [2] where the authors propose a crowdsourcing framework that lets user combine data collection, selection and assessment activities to allow local government to achieve complex goals [3]. The authors present a system where users submit queries that get transformed in a set of tasks that are further submitted to other users. Through the completion of tasks by the other users, such as collecting and assessing images of damaged roads, the query can be answered.

To evaluate city services, Motta *et al*. [4] propose a four-stage model in the design of City Feed – a crowd-sourced governance system. These stages are: publishing (provides government data), interacting (by social media and online service tools), transacting (service integrations) and evaluating. With the growing maturity, Quality of Service (QoS) increases. Briefly, the four-stage model is a roadmap of service evolution, that includes information display, online processing, online interaction, and holistic analysis. The system is composed of two parts: a transactional system (City Feed manager) responsible for processing citizen generated events, creating issues and de-duplicating them, and an analytic system (City Feed analyzer) that simply extract the data from the manager and uploads it to a data warehouse. The data are shown in different forms such as bar charts, and structured along different dimensions (e.g. location, time, event class, etc.).

ArcGIS, currently one of the most capable geographic information systems is another example that allows the processing of streaming events and generation of analytics with the help of the GeoEvent Server extension [5]. By defining both input and output connectors it can receive real-time event streams and push the analytics results to other systems (e.g. to a message queue). Nevertheless, it is limited to its analytics offering leading to extra work for users to use other analytics algorithms and it is also a commercial solution, not a free, open-source one.

Search-the-City [6] is a dashboard primarily concerned with processing large amounts of heterogeneous data (from sensors, cameras, social streams, user generated contented and data produced by city authorities) and displaying it in an easy to consume form. The dashboard's architecture is composed of two parts: a search layer and a visualization framework. The search layer is based on a Storm topology and the Terrier search engine. It is responsible for receiving data collected by edge servers (in the form of XML, RDF and Linked Data) and indexing. The visualization framework takes the concept of mash-up to a new level – the visual components themselves can communicate with each other. This is done by implementing the widgets as Java portlets thus giving them the ability to pass events between each other.

The Bandung Smart City dashboard [7] is a prototype project designed to help solve some of Bandung's – one of Indonesia's cities problems, caused by its fast-growing population. The authors propose architecture composed of sensor nodes that transmit data to processing servers over a classical TCP/IP Internet connection. The sensor nodes sample data using a specialized protocol to reduce energy

consumption. The servers themselves have a database used to store sensor data (although currently the platform only displays the last received value) and Geographical Information System (GIS) software. The result is a single dashboard that gives a summary of the current city-state.
In this context, the paper has the following contributions. First, an analytics architecture designed for city dashboards is presented. The existing solutions are analyzed and requirements for such architecture are provided. Then the analytics pipeline architecture together with statistics computation and clustering algorithms are described. The proposed architecture is evaluated with simulated date on Bucharest as a city example. Finally, the results of proposed algorithms on the city dashboard are presented.
The paper is structured as follows. Section 2 presents the existing solution and a lesson learned from all available approaches. Then, Section 3 presents the architecture and design consideration. Section 4 described the main used algorithms while Section 5 presents My City Dashboard architecture prototype. Section 6 introduces the methodology and experimental results. The paper ends with conclusions and future work in Section 7.

2. Related Work and Existing Solutions

In this section several existing solutions for city dashboard are analyzed.

2.1. Amsterdam City Dashboard

The Amsterdam City Dashboard [8] was briefly launched as a prototype in 2014 and is currently a work in progress. The dashboard has two main modes of displaying data:

- a map view capable of displaying both points representing discrete information types and paths representing statistics along that path, for example the average speed along a road;
- a partition view, where each partition displays a certain category on which city elements are projected. The categories on which the city elements are projected to are: transport, environment, statistics, economy, community, culture, and security. Each category presents a citywide statistics based on blocks of 24 hours with data refreshed every 10 s. Similar approaches were presented in [9] and [10].

The project is based on the City SDK project [11], more specifically the Linked Data API. The API aims to help government agencies open up data and provides the ability to collect data from different sources, annotate, link and make the information available and searchable. Also by using Linked Data, datasets can be easily linked or enriched with user provided information, for example reporting road blockages or alternative routes. The project also provides a developer page [12]. Main characteristics of the SDK:

- authentication – simple session creation (through the use of username and password over HTTPS) and deletion;
- formats – the SDK supports JSON-LD for Linked Data and GeoJSON for representing geographical information;
- endpoints – RESTful endpoints;
- resource types:
  - layers represent data sets,
  - objects can be contained on one on or more layers,
  - owners of layers;
- technologies used – Ruby as a programming language, Grape as a REST framework and PostgreSQL with the addition of the PostGIS add-on in order to add support for geographic objects.

Although this solution is perfect for collecting data and making it available, it doesn't address a few potential necessities:

- handling massive amounts of real-time streaming data;
- the linked data must be generated by the application implementing the SDK and more research has to be done in order to properly use the linked data concept. Not only that, but for a city dashboard use case where we are mostly interested in displaying statistics or applying different machine learning algorithms that would work better on the raw data, the concept of linked data would only add more complexity;
- no way of integrating existing applications such as city service apps or social apps like Twitter, Foursquare, Instagram, etc.;
- user registration;
- dashboard personalization per user.

2.2. Dublin Dashboard

The Dublin Dashboard [13] is part of The Programmable City project [14] and led by Professor R. Kitchin. A few of the most important motivational research questions R. Kitchin and his team [15] try to answer are: how city dashboards can change and influence the performance of a city, how can we display, structure, analyze and select information and is all this reproducible to compare a smart city – in this case Dublin – with other cities that would implement this platform, like benchmarking?
The Dublin Dashboard is an analytical dashboard pulls together data from data sources such as: Dublin City Council, Dublinked [16] – this platform provides most of the real-time data and static datasets, Central Statistics Office, Eurostat, government departments and several existing smart

Table 1

Analysis of existing solutions for city dashboards

Dashboard name	City data sources	Social media networks integration	Data source formats	Provides contextual user information	User log-in	User preferences support	Persistence of data	Analytics support	Future work and observed needs
Amsterdam City Dashboard	Real-time: city wide statistics on transport, environment, statistics economy, community, culture and security	No	JSON-LD, GeoJson	No	No	No	Yes	No	• Linked data is created manually • Not scalable (simple REST framework and an SQL database) • User registration support
Dublin Dashboard	Static: Dublin City Council, existing static datasets, Dublinked platform Real-time: Dublinked – APIs to existing apps	in progress – Facebook, Twitter, Flickr and Instagram	JSON, SML, CSV, XLS, etc.	No	No	No	No	No	• Opening new datasets • Cleaning and processing those data • Developing the site beyond data visualizations to include a roader set of data analytics, including modeling tools
London Dashboard	Static: government institutions Real-time: RSS channels, Twitter, CASA University's wether station, OpenStreetMap updates, Yahoo stocks etc.	Yes – Twitter (trends and special accounts)	HTML – web scrapping, XML, JSON, CSV	No	No	No	No	No	The integration of future data-sources: energy network, sensor data, crowdsourced data – FitBit, atmospheric data
Dubai Personal Dashboard	Real-time: building information and status, weather, prayer timings, video streams etc.	Yes – Twitter and Facebook (requires authentication)	N/A	Yes – also integrates a "My Family" module	Yes	Yes – dashboard customizability	No	No	N/A
Bandung Smart City Dashboard	Real-time: city sensors	Binary	No	No	No	Yes	No		• Analytical tools – precise evaluation and prediction • Decision system support tools
CityEye	Real-time: sensor data (environmental, city service providers), GPS data, citizen	Yes – sentiments analysis of feedback from Facebook and Twitter	N/A	Yes – provides graphs predictions in the user's vicinity	No	No	Yes	Yes – service and indirect sentiment analysis	Include additional data sources: real-time traffic data, pedestrian activity, local wind sensors, water quality sensors
Search-the-City Dashboard	Streaming data oriented – sensors, cameras and social media streams	Yes – Facebook, Twitter, Foursquare	XML, RDF, Linked Data	Yes – provides data from all data sources near chosen location	Yes – split for citizens and municipal administrators	Yes – dashboard customizability and indirect information extracted from social networks accounts and associated social graphs	Yes	Yes – query indexing and query personalization with data steaming from social accounts, event detection, sentiment analysis, contextual statistics	Find optimal combinations of environment generated content originated components from sensors and social networks, in order to provide new visual components

city and social applications (e.g. Twitter, Facebook, or applications which publish links to Dublinked). By using existing resources and applications it minimized duplicate effort.
Types of information it provides – static information, real-time information, time-series indicator data, and interactive maps.
The dashboard contains hundreds of data representations grouped in different modules. Some notable examples are:

- Dublin at a glance module – displays both overall statistics from the city (e.g. number of thefts in the city, or overall air quality index of Dublin) and information from key points in the city (e.g. current parking spots at certain locations or sound level of Blessington Street Basin). It also provides current top news;
- Dublin reporting module – provides links to sites (intuitively in the form of the frontpages of those target apps) that provide services for reporting: FixMyStreet and FixMyArea used for city related problem reporting and CityWatch, which aggregates and display sensor data received from citizens and also municipals;
- Dublin Near Me module – integrates (both in the context of the dashboard as a separate view or with links to separate sites providing the service) of apps such as: Dublin Community Maps (used to find amenities in Dublin), Rate My Area (used to rate city areas) or Vacant Spaces (an app used by users to indicate spaces in the city that are currently unused);
- Dublin RealTime – this module displays the following information: Dublin Environment Maps (displays air quality, ambient sound levels and water levels at certain key locations), City Traffic and Travel (displays available spaces at certain car parks, available bike stations – dynamically clustered on zooming in/out and travel times on certain routes), Maritime Traffic (link to an external site) and a Flight Radar (also a link to an external site);
- Dublin Apps module – is a list of smart city mobile apps that can be used in Dublin with links for easy installation;
- Dublin Social module – a work in progress modules, which suggests that it will integrate information from Facebook, Twitter, Flickr and Instagram.
- Modules providing maps with census, crime (displayed as clustered datasets), companies, housing, and planning information.

Overall, the Dublin Dashboard is the most advanced city dashboard that we could find, with a vast number of data sources, representations and overall smart city integrations. This is mostly the result of Dublin's smart city initiatives that have made it possible to develop many services and applications and open-up the data they provide to be used in new applications, in this case – the Dublin Dashboard.
In the future, the project aims to accomplish the following tasks [17]: opening new datasets, cleaning and processing those data, developing new applications and developing the site beyond data visualizations to include a broader set of data analytics, including modeling tools.
The current drawbacks we identified in this dashboard are: lack of user personalization (users cannot login and control what they want to see or receive more personalized information), no open-source initiative that could be used to allow public contribution to the dashboard, no way of integrating personal applications (e.g. personal Facebook account, or Smart Sports Watch applications that could be used to create key performance metrics related to city resident's health levels etc.) and as of yet, no way to create key performance indicators based on the received data sources as proposed in [17].

2.3. The London Dashboard

The London Dashboard [18] is another UK initiative, this time developed by the Centre for Advanced Spatial Analysis research center of the University College London. It was mainly developed in the first half of 2012 and has been in maintenance since.
The design of the dashboard is simple [19]: a service that collects data from various websites (web scrapping) and APIs, a website composed of three views (a map data is retrieved from a CSV based API, a module-based view and a grid view data is retrieved from a HTML based API).
The data is obtained mainly in 2 ways [20]: web scrapping in the case of sites that do not provide other means of accessing it such as ScotRails tube style line running and APIs returning data in XML/JSON/CSV such as BBC RSS (for local news), OpenStreeMap, RSS updates, Twitter (tweets from a list of accounts related to general news and university news and also top Twitter trends), Mappines (an app that aims to capture the mood of the population across UK), CASA's radiation detector, DEFRA's air pollution data, etc.
The dashboard currently obtains its data from the following sources [21]:

- Department for Environment Food and Rural Affairs,
- National Oceanic and Atmospheric Administration,
- OpenStreetMap (and Pawel's Static Maps API),
- British Broadcasting Corporation,
- London School of Economics,
- Yahoo! Developer Network,
- Port of London Authority,
- Transport for London,

- Yahoo! Finance,
- UCL CASA,
- MapTube,
- ScotRail,
- Twitter.

Each module that displays information from a certain data-source also has a counter notifying the user of the next update. This is mostly because the server caches responses from APIs in order to improve performance and go around rate limits (such as from Twitter).
The project also provides developers with a set of APIs [22] that expose the aggregated data of the input data-sources and that are used in the creation of the dashboard so that they can be consumed by external services (e.g. PigeonSim, the London Periodic Table, the London Data Table, Prism, etc.). The APIs are very simple and come in two MIME-type flavors: CSV and HTML.
A few of the main challenges in the future development of the project are as follows [20]:

- finding real-time and reliable data,
- integration of various heterogeneous services,
- issues in filtering and representing social networking data,
- obtaining environment related sensor data,
- news, events and community data sources are not easily available.

The project also expects the integration of future data-sources [19] such as from the energy network or sensor data sources that have not yet opened-up their data and also from crowd initiatives (e.g. Air Pollution Egg for atmospheric data and the FitBit One for personal mobility).
Also, an initial design proposal for the city dashboard [20] was that of being customizable to user preferences so that is another direction in which the dashboards development might head towards. The lack of database storage is also noted, but currently the dashboard's functionality does require analyzing stored data.
One thing that can be improved: provide users with the ability to understand the cause of certain indicators by visualizing data over recent time intervals.

2.4. Dubai Personal Dashboard

The Dubai Personal Dashboard [22] was launched by the Dubai Civil Defence (DCD) in October 2015 with the aim of allowing residents and visitors to visualize data, keep them informed and connected, make their daily city interactions more seamless, enable real-time public engagement and generally contribute to enabling Dubai to be the smartest city in the world.

The dashboard collects and displays data from public and private sources, and from social media networks (Facebook – personal feeds, Tweeter – government tweets). In order to enable seamless login, the dashboard uses Dubai's MyID service for single sign-on. The dashboard is module based, being formed from a mash-up of modules, each displaying a different concept. The modules are of 3 types:

- **General modules**: nearby fire stations, DCD newsletter, Gulf News, tweets from the government, weather, prayer timings;
- **Personalized modules**: building related information and alerts modules – requires the user to map their building ID; Facebook feeds – requires the user to sign-in with his account in the module; video streaming – required camera source and login information.
- **Reporting modules**: emergency button – for fire, police, ambulance etc; safety violation reporting – ability to create and send a safety violation report with detailed information, pictures and video.

The Dubai Personal Dashboard is one of the only dashboards that support personalized information. The dashboard does not currently have a lot of data sources and no complex city analytics so that is a direction it may take in the future.

2.5. Bandung Smart City Dashboard

The Bandung Smart City Dashboard [24] is a prototype project designed to help solve some of Bandung's – one of Indonesia's cities – problems, caused by its fast-growing population. By monitoring what happens in the city, it can help the government detect problems and find solutions as soon as possible. The dashboard is design to monitor indicators such as: temperature, air pollution, water pollution, traffic situation, economic indicators, energy supplies, number of citizens, number of vehicles, number of houses, etc.
The authors of [24] propose architecture composed of sensor nodes that transmit data to processing servers over a classical TCP/IP internet connection. The sensor nodes sample data using a specialized protocol in order to reduce energy consumption. The servers themselves have a database used to store sensor data (although currently the platform only displays the last received value) and Geographical Information system (GIS) software. The result is a single dashboard that gives a summary of the current city-state.
In the future, the authors propose adding more decision support system and analytical and add functionalities such as prediction.

2.6. CityEye

CityEye [25] is an urban visualization and management dashboard designed as a solution to the problem of urban

infrastructure and service transparency. The aim of the platform is to foster richer dialogue between users and urban service providers and give citizens an overall view on the state of the city.
The context of two cities was used to find current existing smart city platforms and problems associated with them: Barcelona (heterogeneity of data resulted from each urban service company resulted in integration problems leading to the need of data standardization) and Santander, where applications have failed to generate interest from citizens and companies.
The data sources included in CityEye can be divided into three categories: sensor data, data generated during the provision of services, and data generated by citizens (in the form of feedback and reports from applications or by email and sentiment analysis of Facebook and Twitter posts). Also, it has been proposed that some of these sensors can be embedded in urban maintenance vehicles, which combined with GPS capabilities can provide monitoring capabilities across the city.
There are still some problems identified in the development of the platform: data availability, existing sensor infrastructure, relationships between service providers and government, real-time traffic information, capture of pedestrian activity, water quality testing.

2.7. Search-the-City Dashboard

Search-the-City dashboard [6] is primarily concerned with processing large amounts of heterogeneous data from sensors, cameras, social streams, user generated contented and data produced by city authorities, and displaying it in an easy to consume form.
The dashboard's architecture is composed of two parts: a search layer and a visualization framework. The dashboard itself is implemented using an off-the-shelf solution for building portals Liferay. It allows building sites with role-based access control, single sign-on support, CMS functionalities and the ability to integrate Java portlets.
The search layer is based on a Storm topology and the Terrier search engine and has 3 sub-layers:

- a layer of edge servers that are responsible for acquiring sensor data and passing it to the storm topology in the form of standard formats (XML, RDF, Linked Data),
- a search layer based on the Terrier search engine responsible for indexing data retrieved from edge servers and answering queries,
- end-user applications layer, which submit queries to the search engine.

The search layer also treats social media information (from Facebook, Twitter and Foursquare) as normal sensor data allowing for the use of algorithms such as: sentiment analysis, event detection etc. Another interesting aspect is that the search layer can also return personalized results based on the user's social applications.
The visualization framework takes the concept of mash-up present in the previously presented dashboards to new level – the visual components themselves can communicate with each other. This is done by implementing the widgets as Java portlets thus giving them the ability to pass events between each other. This gives the possibility of doing a workflow in which a user enters a search query in a widget and upon receiving event data from the search layer. It distributes it to another widget that lists a search result set. Upon clicking on an item on the result set, events are transmitted to the other widgets resulting in them updating. The updated widgets are: a video widget with camera feeds near the location, a social media widget with posts geo-located near that location, a sensor widget with nearby sensor data and a maps widget that centers to that location. The result of a query for a specific location is an event that contains all the data related to that location such as that presented in the workflow above.

3. Architecture and Design Considerations

For the design of the analytics platform several criteria have been chosen:

- *high scalability*– resulting from the great quantity of data generated by sensor streams, crowdsourcing produced streams and social streams;
- *modularity* – the system would be required to use a large amount of sub-systems. These systems must be allowed to evolve independently with no or reduced impact on the rest of architecture;
- *pluggability* – new processing components, event streams, data sinks etc. can be attached or detached without affecting the rest of the system;
- the main technologies used must be able to interact with each other without requiring the development of custom communication modules or modifying the open-source projects;
- data must be presented in an easy to understand way, mainly using key performance indicators and aggregated data [26].

These requirements have led to the decision to use the following technologies for each layer of the architecture. The overall architecture is presented in Fig. 1. As possible use cases for the system we have considered the following stream types: temperature and noise. We currently provide identical functionality for these data types so only the results of noise analytics [27], [28] will be presented.

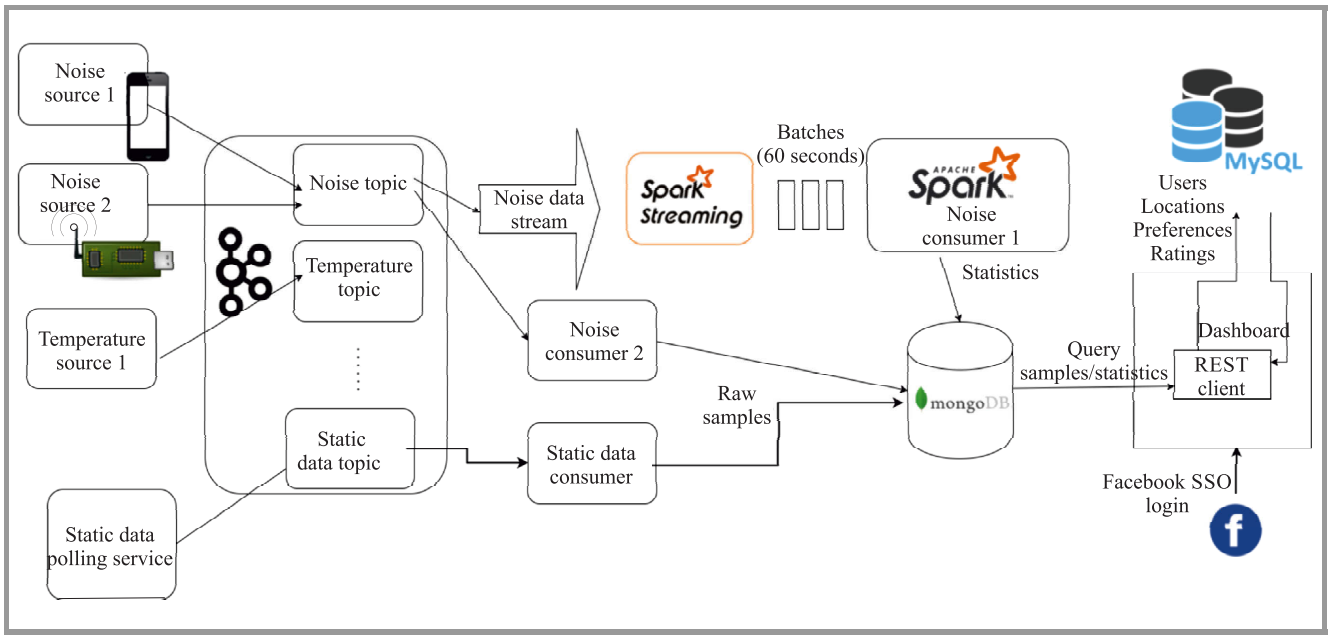


Fig. 1. Analytics pipeline architecture.

3.1. Acquisition Layer

For the ingestion layer, we have chosen Apache Kafka, which is a high-throughput, low-latency and massively scalable publish/subscribe message queue for handling real-time data feeds [29]–[31]. The other technology we have considered is RabbitMQ, one of the most known and used message queues. The decision was made because RabbitMQ does not support multiple consumers per queue as such resulting in low pluggability (we require that developers should be able to write whatever analytics modules they want without being impacted by other existing modules) and its performance is about 2 times lower than that of Kafka [32]. Also Kafka provides a Spark Streaming modules that allows us to easily integrate its output in Spark as we will see later.

3.2. Processing Layer

For the processing layer Apache Spark and Apache Flink have been considered. While Flink is more suited for streaming data, providing better streaming semantics support, the project is still in its infancy (the project is currently incubating). This lack of current support has impacted our decision mainly because there was no MongoDB connector that could easily stream data to our database thus requiring us to write files on disk and use the Hadoop Connector, which is not what we want. Also, while presented system is designed for real-time analytics we do not require pure stream processing because most of operations are done in small batches (e.g. 60 s batches).

3.3. Persistance Layer

Because we receive a high amount of data a highly scalable database. Also, considering that most of the data has a geographic source that is directly used in processing pipeline and in our visualization layer we required a database that supports GIS operations such as retrieving all the information in a defined area on the map. The main two technologies that have been considered are MongoDB and the PostGIS module for PostgreSQL. While both solutions provide the required operators only MongoDB is designed to scale for a high amount of read and particularly write operations.

3.4. Dashboard Layer

The dashboard layer is composed of a separate dashboard project in which we have plugged in the analytics functionality. The dashboard itself is built on top of a Service-Oriented Architecture (SOA) architecture composed of RESTful services. This layer, besides providing the UI also gives a proof of concept example of using the provided APIs for accessing and creating event, location and user resources. The location APIs are especially useful because they will be used in further research for providing a way to import and display location and event related data in the dashboard.

4. Algorithms

Two main algorithms have been implemented in the platform: an overall statistics algorithm and a clusterization algorithm used for aggregating the huge amount of data that would need to be display on the map and only compute statistics on data partitioned in these clusters.
The first algorithm is simple and composed of only a small list of steps as all the functionality is already provided by Spark (see Algorithm 1).

The clusterization algorithm is more complex and requires some description. The complexity does not derive from the algorithm itself but from mapping the algorithm in Spark (it is still a lot easier than manually implementing it).

Algorithm 1 Overall statistics computation

```
MC ← MongoConnector;
samples ← Kafka.readStream("noise");
parsed ← samples.map(deserializeFunc);
samples.foreachRDD(
procedure function(RDD) {
    if !rdd.isEmpty() then
        convRDD ← RDD.mapToDouble(mapFunc);
        min = convRDD.min();
        max = convRDD.max();
        avg = convRDD.average();
        count = convRDD.count();
        rddRow = NewRDDRow(min,max,avg,count);
        MC.save(rddRow);
});
```

Algorithm 2 Tile clustering algorithm.

```
samples ← Kafka.readStream("noise");
MC ← MongoConnector;
partitionedSamples ← samples.mapToKeyValuePair(
procedure function(sample){
    tileIdX ← convertToTileCoordinates(sample).X;
    tileIdY ← convertToTileCoordinates(sample).Y;
    key = tileIdX:tileIdY;
    value = sample.value;
    return (key, value);}
);
intermKeyValuePairs ← partitionedSamples.mapValue(
procedure function(sample){
    return (sample.value, 1); }
);
reduceResults ← intermediaryKeyValuePairs.reduce(
procedure function((key1, value1), (key2, value2)){
    return (key1 + key2, value1 + value2); }
);
reduceResults.foreachRDD(
procedure function(p){
    if !pairRDD.isEmpty() then
        Map< key, value > map = p.collectAsMap();
        for (key, value) ∈ map do
            tileIdX = parse(key).X;
            tileIdY = parse(key).Y;
            average = value.getComputedAverage();
            count = value.getComputedCount();
            rddRow = NewRDDRow(key, tileIdX,
                tileIdY, average, count,
                computeTialCoordinates());
            MC.setKey(key).save(rddRow);
});
```

The algorithm is based on the Cluster Griddy clustering type [33]. It works as follows: a reference point is chosen in geographic coordinates on the map. It is in the lower left of the map. A tiles size is chosen. Each data point is converted from geographic coordinates to tile-based coordinates, that is each point is assigned to a tile. Each tile is mapped to a unique key of the form $tileIdX : tileIdY$ in order to parallelize computation on Spark. For each tile, we compute the necessary statistics. On information retrieval, the tile coordinates are transformed in 4 geographical coordinates that represent that tile (see Algorithm 2).

5. *My City Dashboard* Architecture Prototype

In its current form, the city dashboard is based on a SOA architecture implemented using RESTful APIS in the Java Spring Boot Actuator framework.
The accessible endpoints are as follows:

- `/locations` – API used for creating and accessing locations,
- `/users` – API used for creating and accessing users,
- `/noise/samples` – API used for retrieving the total number of processed sensor samples,
- `/noise/latestAggrates` – API used for retrieving the current computed overall city aggregate data,
- `/noise/tiles` – API used for retrieving overlay tiles that create a heatmap representing sensor data on the map.

The user and location information are stored in a MySQL database and are accessed through the Hibernate ORM. The database tables used for storing user, location and relation entities are presented in Fig. 2.
The main interactions available currently in the dashboard are presented in the navigation diagram in Fig. 3.

6. Methodology and Experimental Results

The data was simulated using Bucharest as a city example. The geographic coordinates where generated using the coordinates of Bucharest's city center and a radius encompassing the city. The architecture was run on a virtual machine with 2 cores processor, 4 GB RAM and 20 GB SSD. The underlying hardware is a MacBook Pro 15 with Broadwell i7 2.5 GHz, 16 GB, 512 GB SSD. We present the results of our algorithms on the city dashboard.
In Fig. 4 we can see the results of running presented application. The tiles are 500 × 1000 meters in size and repre-

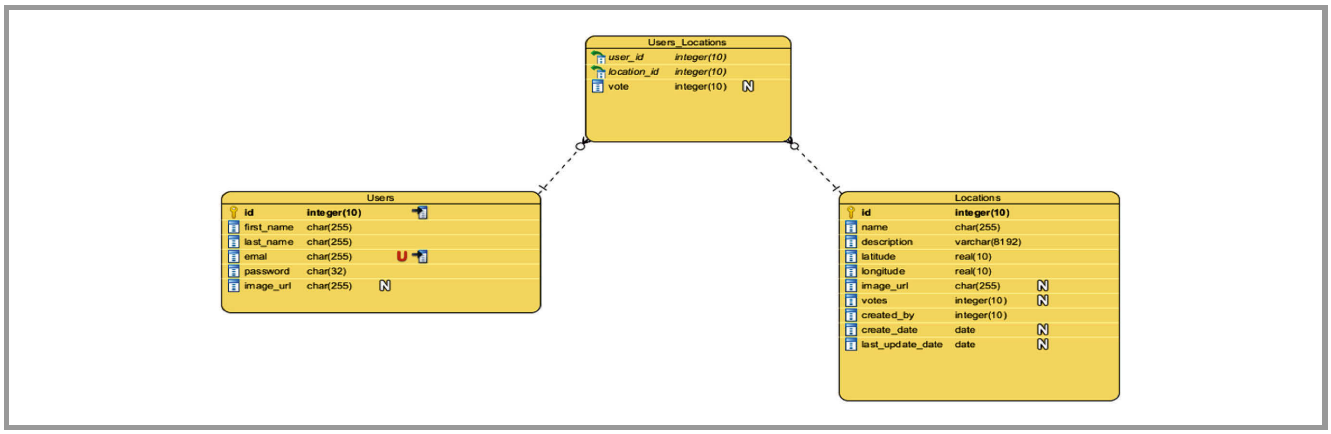


Fig. 2. Analytics pipeline architecture.

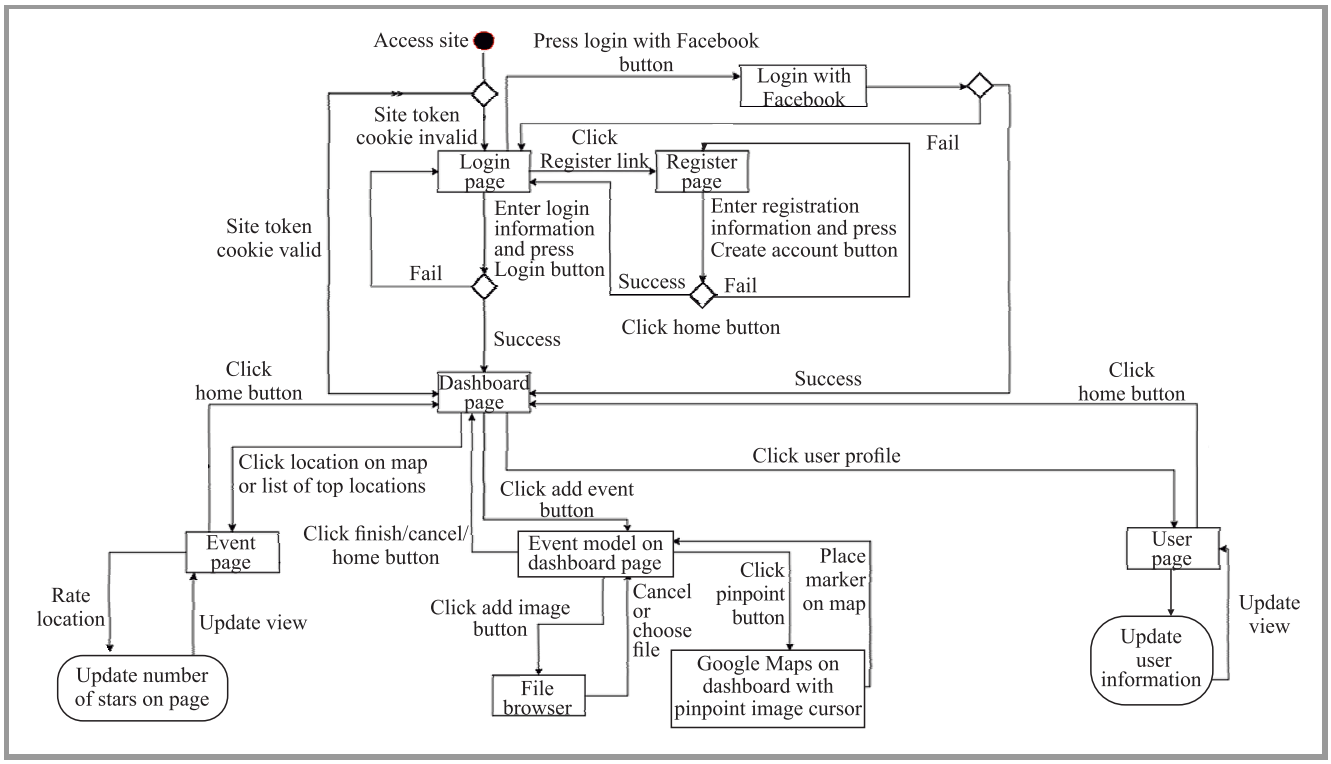


Fig. 3. Navigation diagram.

sent the average noise value over that region. Also, we can see the average noise statistics for the whole city. The overall processing speed of the platform results in a processing rate of approximately 523 sample/s.

While that is not a large value we have to keep in mind that 3 distributed systems and the dashboard were running on the same PC and in a virtual machine. In Fig. 5 we can see a more overall view of the city. In this running instance, only part of the city tiles received information.

In Fig. 6 we can see the complete tile rendering for a portion of the city. Also, to be noticed, one region in the upper left of the image is more green conveying the fact that that region is quiet compare to the rest of the city (at least in the time frame it was analyzed – we simulated the fact that the airport was shut down).

7. Conclusions

Our implementation and experiments were run with three test cases – city noise, temperature and pollution use cases. The results of our experiments are shown from two points of view: performance and visual representation. In terms of performance, despite the constricted running environment the results are promising. The visual representation is shown from 3 perspectives – a small map portion, a large map portion with partial data sources and a larger map portion completed covered with data sources where some zones stand out from the other.

Also, we have presented a first prototype of the dashboard with user login support, including SSO through an OAuth2 provider – Facebook to enable seamless login.

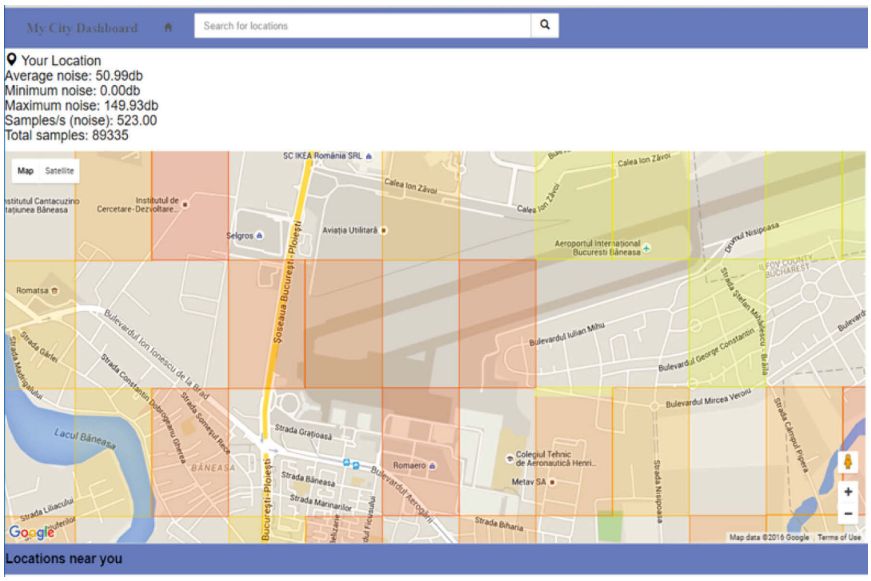


Fig. 4. City dashboard – analytics example 1.

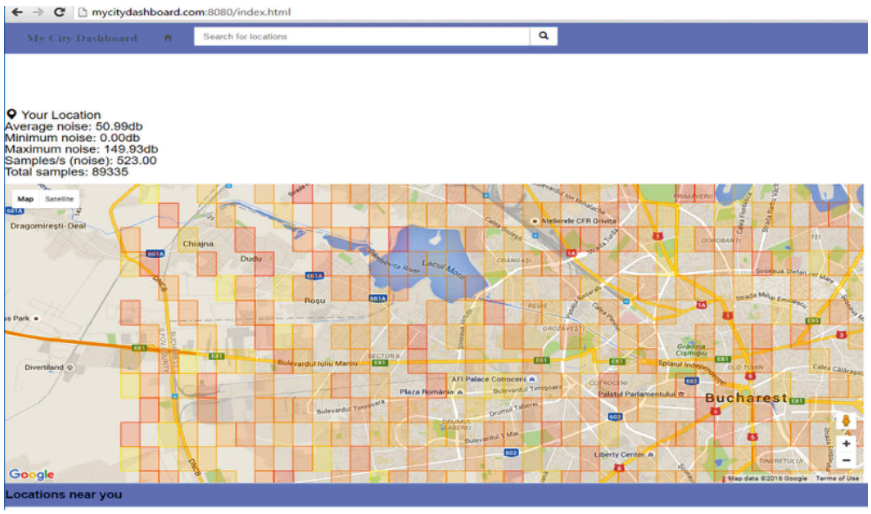


Fig. 5. City dashboard – analytics example 2.

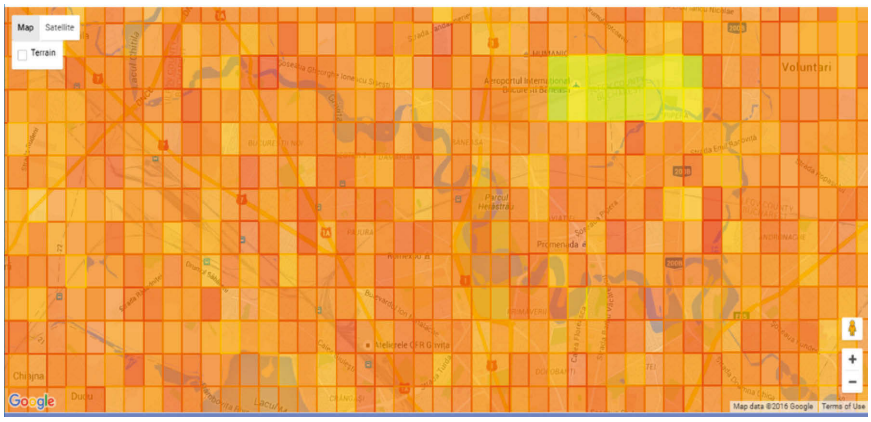


Fig. 6. City dashboard – analytics example 3.

Also, a navigation diagram was presented for various operations the user can currently do in the dashboard, such as creating interesting locations/events, find locations/events or event rating them. This functionality only serves as a proof of concept of the underlying APIs, which will be used to import and display data from external sources.
In the future, we will consider more data sources such as social streams or open data and provide additional processing capabilities to integrate these new sources.

Acknowledgements

The research presented in this paper is supported by the following projects: *DataWay*: Real-time Data Processing Platform for Smart Cities: Making sense of Big Data - PN-II-RU-TE-2014-4-2731; *MobiWay*: Mobility Beyond Individualism: an Integrated Platform for Intelligent Transportation Systems of Tomorrow – PN-II-PT-PCCA-2013-4-0321.

We would like to thank the reviewers for their time and expertise, constructive comments and valuable insight.

References

[1] M. D'Arienzo, M. Iacono, S. Marrone, and R. Nardone, "Estimation of the energy consumption of mobile sensors in WSN environmental monitoring application", in *Proc. 27th Int. Conf. on Adv. Inform. Netw. & Appl. Worksh. WAINA 2013*, Barcelona, Spain, 2013, pp. 1588–1593.

[2] K. Benouaret, R. Valliyur-Ramalingam, and F. Charoy, "CrowdSC: Building smart cities with large-scale citizen participation", *IEEE Internet Comput.*, vol. 17, no. 6, pp. 57–63, 2013.

[3] D. Merezeanu, C. Vasilescu, and R. Dobrescu, "Context-aware Control Platform for Sensor Network Integration in IoT and Cloud, *Studies in Informatics and Control*, vol. 25, no. 4, pp. 489–498, 2016.

[4] G. Motta, L. You, D. Sacco, and T. Ma, "CITY FEED: A crowdsourcing system for city governance", in *Proc. IEEE 8th Int. Symp. on Service Oriented System Engin. SOSE 2014*, Oxford, UK, 2014, pp. 439–445.

[5] ArcGIS GeoEvent Server, "Real-Time Mapping and Analytics" [Online]. Available: http://www.esri.com/software/arcgis/arcgisserver/extensions/geoevent-extension (accessed on Jan. 2017).

[6] A. Moralis, G. Perreas, A. Glaros, and D. Dres, "Search-the-City – A versatile dashboard for searching and displaying Environment and User Generated Content in the context of the future Smart City", in *Proc. Information Access in Smart Cities i-ASC 2014*, Amsterdam, The Netherlands, 2014 [Online]. Available: http://dcs.gla.ac.uk/workshops/iASC2014/papers/iasc2014_moralis.pdf

[7] S. Suakanto, S. H. Supangkat, and R. Saragih, "Smart city dashboard for integrating various data of sensor networks", in *Proc. Int. Conf. on ICT for Smart Society ICISS 2013*, Jakarta, Indonesia, 2013.

[8] Amsterdam City Dashboard [Online]. Available: http://citydashboard.waag.org (accessed on Jan. 2017).

[9] C. Fratila, C. Dobre, F. Pop, and V. Cristea, "A transportation control system for urban environments", in *Proc. 3rd In. Conf. on Emerg. Intell. Data and Web Technol. EIDWT 2012*, Bucharest, Romania, 2012, pp. 117–124.

[10] C. Gosman, T. Cornea, C. Dobre, F. Pop, and A. Castiglione, "Controlling and filtering users data in Intelligent Transportation System", *Future Gener. Comp. Syst.*, in press, 2015 (doi: 10.1016/j.future.2016.12.014).

[11] CitySDK Linked Data API [Online]. Available: http://dev.citysdk.waag.org (accessed on January 2017).

[12] CitySDK Linked Data API Source Code [Online]. Available: https://github.com/waagsociety/citysdk-ld (accessed on Jan. 2017).

[13] Dublin Dashboard [Online]. Available: http://www.dublindashboard.ie (accessed on Jan. 2017).

[14] The Programable City project [Online]. Available: http://www.maynoothuniversity.ie/ (accessed on Jan. 2017).

[15] R. Kitchin, Rob, T. P. Lauriault, and G. McArdle, "Knowing and governing cities through urban indicators, city benchmarking and real-time dashboards", *Regional Studies, Regional Science*, vol. 2, no. 1, pp. 6–28, 2015 (doi: 10.1080/21681376.2014.983149).

[16] Dublin data sharing platform [Online]. Available: http://dublinked.ie/ (accessed on Jan. 2017).

[17] Dublin Dashboard challenges [Online]. Available: http://smartdublin.ie/ smartstories/dublin-dashboard (accessed on Jan. 2017).

[18] London Dashboard [Online]. Available: http://citydashboard.org (accessed on Jan. 2017).

[19] F. Roumpani, O. O'Brien, and A. Hudson-Smith, "Creating, visualizing and modelling the realtime city" [Online]. Available: http://casa.oobrien.com/ misc/presentations/roumpani2012a.pdf (accessed on Jan. 2017).

[20] Smart Cities – City DashBoards DashBoards Lecture [Online]. Available: http://www.spatialcomplexity.info/files/2013/06/Session-5-Lecture-2.pdf (accessed on Jan. 2017).

[21] London Dashboard API [Online]. Available: http://oobrien.com/2012/06/citydashboard-the-api/ (accessed on Jan. 2017).

[22] Dubai Personal Dashboard [Online]. Available: https://pdb.24x7dcd.ae/portal/dashboards/personal-dashboard (accessed on Jan. 2017).

[23] Dubai Civil Defence presentation [Online]. Available: http://24x7dcd.ae/pdf/DCD-Personal-Dashboard-23-Oct-2015-English.pdf (accessed on Jan. 2017).

[24] S. Suakanto, S. H. Supangkat, Suhardi, and R. Saragih, "Smart city dashboard for integrating various data of sensor networks", in *Proc. Int. Conf. on ICT for Smart Society ICISS 2013*, Jakarta, Indonesia, 2013.

[25] D. Lee *et al.*, "CityEye: Real-time visual dashboard for managing urban services and citizen feedback loops", in *Proc. 14th Int. Conf. on Comp. in Urban Plann. & Urban Manag. CUPUM 2015*, Cambridge, MA USA, 2015.

[26] V. Serbanescu, F. Pop, V. Cristea, and G. Antoniu, "Architecture of distributed data aggregation service", in *Proc. IEEE 28th Int. Conf. on Adv. Inform. Netw. & Appl. AINA 2014*, Victoria, Canada, 2014, pp. 727–73.

[27] E. Barbierato, M. Iacono, and S. Marrone, "PerfBPEL: A graph-based approach for the performance analysis of BPEL SOA applications", in *Proc. 6th Int. Conf. on Perform. Eval. Methodol. & Tools VALUETOOLS 2012*, Cargèse, France, 2012, pp. 64–73.

[28] E. Barbierato, M. Gribaudo, M. Iacono, and S. Marrone, "Performability modeling of exceptions-aware systems in multiformalism tools", in *Proc. Int. Conf. on Anal. & Stoch. Model. Techniq. & Appl. ASMTA 2011*, Venice, Italy, 2011, pp. 257–272.

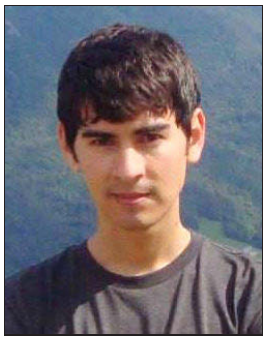
[29] C. Esposito, A. Castiglione, F. Palmieri, M. Ficco, and K. K. R. Choo, "A publish/subscribe protocol for event-driven communications in the Internet of Things", in *Proc. IEEE 14th Int. Conf. on Depend., Autonom. & Secure Computing DASC 2016, IEEE 14th Int. Con. on Perv. Intelligence & Computing PICom 2016, IEEE 2nd Int. Conf. on Big Data Intelligence & Computing DataCom 2016, IEEE Cyber Sci. & Technol. Congr. CyberSciTech 2016 (DASC-PICom-DataCom-CyberSciTec 2016)*, Auckland, New Zealand, 2016, pp. 376–383.

[30] C. Esposito, M. Ficco, F. Palmieri, and A. Castiglione, "A knowledge-based platform for Big Data analytics based on publish/subscribe services and stream processing", *J. of Knowledge-Based Syst.*, vol. 79, pp. 3–17, 2015 (doi: 10.1016/j.knosys.2014.05.003).

[31] C. Esposito, A. Castiglione, and K. K. R. Choo, "Challenges in Delivering Software in the Cloud as Microservices", *IEEE Cloud Comput.*, vol. 3, no. 5, pp. 10–14, 2016.

[32] J. Kreps, N. Narkhede, and J. Rao, "Kafka: A distributed messaging system for log processing", in *Proc. 6th Int. Worksh. on Netw. Meets Databs. NetDB 2011*, Athens, Greece, 2011.

[33] Cluster Griddy [Online]. Available: http://maplarge.com/visual/clustering (accessed on Jan. 2017).



Cătălin-Constantin Uşurelu is a master in Computer Science student within Computer Science Department, University Politehnica of Bucharest. His research interests are oriented on big data processing for smart cities application, data cleaning and aggregation, real time processing. He participated in Google Summer of Code 2013 with a Debian project. He has an excellent knowledge on NoSQL databases (e.g. MongoDB, DynamoDB), RESTful services, AngularJS, JMeter.
E-mail: catalin.usurelu@cti.pub.ro
University Politehnica of Bucharest
Faculty of Automatic Control and Computers
Computer Science Department
Splaiul Independentei 313, Sector 6
Bucharest 060042, Romania



Florin Pop is Associate Professor at the Department of Computer Science and Engineering at the University Politehnica of Bucharest. His main research interests are in the field of large scale distributed systems concerning scheduling and resource management (decentralized techniques, re-scheduling), adaptive and autonomous methods, multi-criteria optimization methods, grid middleware tools and applications development (satellite image processing an environmental data analysis), prediction methods, self-organizing systems, data retrieval and ranking techniques, contextualized services in distributes systems, evaluation using modeling and simulation (MTS2). He was awarded with two Prizes for Excellence from IBM and Oracle, three Best Paper Awards (in 2013, 2012, and 2010), and one IBM Faculty Award.
E-mail: florin.pop@cs.pub.ro
University Politehnica of Bucharest
Faculty of Automatic Control and Computers
Computer Science Department
Splaiul Independentei 313, Sector 6
Bucharest 060042, Romania

National Institute for Research and Development
in Informatics (ICI)
8-10, Mareşal Averescu
011455 Bucharest, Romania

Information for Authors

Journal of Telecommunications and Information Technology **(JTIT)** is published quarterly. It comprises original contributions, dealing with a wide range of topics related to telecommunications and information technology. **All papers are subject to peer review**. Topics presented in the JTIT report primary and/or experimental research results, which advance the base of scientific and technological knowledge about telecommunications and information technology.

JTIT is dedicated to publishing research results which advance the level of current research or add to the understanding of problems related to modulation and signal design, wireless communications, optical communications and photonic systems, voice communications devices, image and signal processing, transmission systems, network architecture, coding and communication theory, as well as information technology.

Suitable research-related papers should hold the potential to advance the technological base of telecommunications and information technology. Tutorial and review papers are published only by invitation.

Manuscript. TEX and LATEX are preferable, standard Microsoft Word format (.doc) is acceptable. The authors JTIT LATEX style file is available:
http://www.nit.eu/for-authors

Papers published should contain up to 10 printed pages in LATEX authors style (Word processor one printed page corresponds approximately to 6000 characters).

The manuscript should include an abstract about 150200 words long and the relevant keywords. The abstract should contain statement of the problem, assumptions and methodology, results and conclusion or discussion on the importance of the results. Abstracts must not include mathematical expressions or bibliographic references.

Keywords should not repeat the title of the manuscript. About four keywords or phrases in alphabetical order should be used, separated by commas.

The original files accompanied with pdf file should be submitted by e-mail: redakcja@itl.waw.pl

Figures, tables and photographs. Original figures should be submitted. Drawings in Corel Draw and PostScript formats are preferred. Figure captions should be placed below the figures and can not be included as a part of the figure. Each figure should be submitted as a separated graphic file, in .cdr, .eps, .ps, .png or .tif format. Tables and figures should be numbered consecutively with Arabic numerals.

Each photograph with minimum 300 dpi resolution should be delivered in electronic formats (TIFF, JPG or PNG) as a separated file.

References. All references should be marked in the text by Arabic numerals in square brackets and listed at the end of the paper in order of their appearance in the text, including exclusively publications cited inside. Samples of correct formats for various types of references are presented below:

[1] Y. Namihira, Relationship between nonlinear effective area and mode field diameter for dispersion shifted fibres, *Electron. Lett*., vol. 30, no. 3, pp. 262264, 1994.

[2] C. Kittel, *Introduction to Solid State Physics*. New York: Wiley, 1986.

[3] S. Demri and E. Orłowska, Informational representability: Abstract models versus concrete models, in *Fuzzy Sets, Logics and Knowledge-Based Reasoning*, D. Dubois and H. Prade, Eds. Dordrecht: Kluwer, 1999, pp. 301314.

Biographies and photographs of authors. A brief professional authors biography of up to 200 words and a photo of each author should be included with the manuscript.

Galley proofs. Authors should return proofs as a list of corrections as soon as possible. In other cases, the article will be proof-read against manuscript by the editor and printed without the author's corrections. Remarks to the errata should be provided within one week after receiving the offprint.

Copyright. Manuscript submitted to JTIT should not be published or simultaneously submitted for publication elsewhere. By submitting a manuscript, the author(s) agree to automatically transfer the copyright for their article to the publisher, if and when the article is accepted for publication. The copyright comprises the exclusive rights to reproduce and distribute the article, including reprints and all translation rights. No part of the present JTIT should not be reproduced in any form nor transmitted or translated into a machine language without prior written consent of the publisher.
For copyright form see: http://www.nit.eu/for-authors

A copy of the JTIT is provided to each author of paper published.

Journal of Telecommunications and Information Technology has entered into an electronic licencing relationship with EBSCO Publishing, the worlds most prolific aggregator of full text journals, magazines and other sources. The text of *Journal of Telecommunications and Information Technology* can be found on EBSCO Publishings databases. For more information on EBSCO Publishing, please visit www.epnet.com.

(*Contents Continued from Front Cover*)

Introduction to Big Data Management Based on Agent Oriented Cyber Security

J. Raiyn — Paper — 65

Using Polymatrix Extensive Stackelberg Games in Security – Aware Resource Allocation and Task Scheduling in Computational Clouds

A. Jakóbik and A. Wilczyński — Paper — 71

Data Fixing Algorithm in Radiosonde Monitoring Process

P. Szuster — Paper — 81

My City Dashboard: Real-time Data Processing Platform for Smart Cities

C.-C. Usurelu and F. Pop — Paper — 89

INSTYTUT ŁĄCZNOŚCI
PAŃSTWOWY INSTYTUT BADAWCZY

Editorial Office

National Institute
of Telecommunications
Szachowa st 1
04-894 Warsaw, Poland

tel. +48 22 512 81 83
fax: +48 22 512 84 00
e-mail: redakcja@itl.waw.pl
http://www.nit.eu

JOURNAL OF TELECOMMUNICATIONS AND INFORMATION TECHNOLOGY 1/2017