JOURNAL OF TELECOMMUNICATIONS AND INFORMATION TECHNOLOGY

4/2017

(*Contents Continued on Back Cover*)

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

Circulation: 300 copies Sowa – Druk na życzenie, www.sowadruk.pl, tel. 22 431-81-40

Ministry of Science and Higher Education
Republic of Poland

Improvement of language quality; Assigning DOIs; Subscription to the plagiarism detection system – tasks financed under 556/P-DUN/2017 agreement from the budget of the Ministry of Science and Higher Education under the science dissemination fund.

JOURNAL OF TELECOMMUNICATIONS AND INFORMATION TECHNOLOGY

Preface

This issue is not devoted to a single subject, and includes papers on subjects belonging to several distinct fields of communications and information technology:

– radio communications,
– optical networks,
– information technology, software and data processing, and
– postal services.

Papers related to radio technologies begin with the *An Improved Downlink MC-CDMA System for Efficient Image Transmission* by M. S. Bendelhoum, A. Djebbari, I. Boukli-Hacene and A. Taleb-Ahmed. The authors attempt to optimize compression of images transmitted over a multi-carrier radio link having a limited bandwidth and varying transmission quality affected by noise. After extensive computer simulations, they have found that image compression using the Discrete Wavelet Transform (DWT) technique, in conjunction with the set Partitioning in Hierarchical Trees (SPIHT) coder gives the best results when the compressed image is sent over a Multi-Carrier Code Division Multiple Access (MC-CDMA) wireless network with a limited signal-to-noise ratio, where selective degradation of one of parallel data streams is frequently experienced. Monochrome photograph compression rates of up to 90% are possible.

The next two papers deal with different aspects of the beamforming technology being currently introduced in wireless networks.

The Multi-User Multiple Input Multiple Output (MU-MIMO) technology is very promising for high capacity wireless networks, including Wireless Local Area Networks (WLANs) defined in the IEEE 802.11ac standard, and for the future 5G networks. However, the full advantage of MU-MIMO can be experienced only with proper user selection and scheduling. User scheduling is done after acquisition of Channel State Information (CSI) from all users, and the number of CSI requests grows with number of active users, resulting in a rising CSI overhead and in degradation of the overall network throughput. In the paper titled *QoS-based Joint User Selection and Scheduling for MU-MIMO WLANs*, D. S. Rao, and V. B. Hency present the Joint User Selection and Scheduling (JUSS) scheme, comparing its performance

to other Medium Access Control (MAC) protocols. It was found that JUSS enhances network throughput and prevents contention during CSI feedback period, rendering it superior to other protocols.

The next paper, *Synthesis and Failure Correction of Flattop and Cosecant Squared Beam Patterns in Linear Antenna Arrays* by H. Patidar, G. K. Mahanti, and R. Muralidharan is devoted to the synthesis of flat-top and cosecant squared beam patterns using the firefly algorithm, and single-fault situations in antenna arrays, which distort the desired emission pattern, and the ways to automatically correct it by re-setting operating parameters of other working antennae to make the whole antenna system fault-tolerant. This "recovery" process was simulated in Matlab, with emphasis on reduction of side lobe level, ripple and the reflection coefficient. Simulations showed a successful application of the firefly algorithm for this purpose. Such a recovery enables uninterrupted operation of complex antenna systems with acceptable performance levels, without having to wait for repair or replacement of the faulty component.

In their paper *The Alive-in-Range Medium Access Control Protocol to Optimize Queue Performance in Underwater Wireless Sensor Networks*, V. Raina, M. K. Jha and P. P. Bhattacharya present work on optimization of a radio system operating in a special, hostile environment: an underwater wireless sensor network, composed of multiple fixed sensors and a mobile "sink" (interrogator) device collecting data from them in a wireless manner. Seawater is a loss-intensive medium, while both the sensors and the sink operate with severe power limitations, being usually battery-powered. For this purpose, the Alive-in-Range Medium Access Control (AR-MAC) protocol was adopted, with reduction in duty cycle, precise time scheduling of active/sleep cycles of the sensors, and monitoring of the mobility of the sink node, accompanied by selection of appropriate queues and schedulers to save power and prevent loss of priority data.

The field of optical communications is represented in this issue by two papers devoted to improving the reliability of communications over optical fibers and in a free space. In both cases, the applications are in access networks and operating distances are relatively short.

While the Fiber To The Home (FTTH) technology, most often in the PON (Passive Optical Network) variant, is generally considered as the best option for fixed broadband access in terms of transmission performance, FTTH networks are not immune to outages caused by cable cuts. In the paper *Availability Analysis of Different PON Models*, K. Radoš and I. Radoš analyze the protection of feeder fiber paths in FTTH-PON access networks to improve network resilience to cable cuts, which tend to be the main source of network failures in the urban environment. However, while protection by adding a spare feeder fiber between the central office (CO) and PON splitter, and including spare OLT active equipment at the CO in some cases, can improve service availability for demanding customers, a significant extra investment in spare fibers laid along separate routes is required. The authors compare the effectiveness of PON protection schemes standardized by ITU-T in several network scenarios. One conclusion is that protection of a short feeder fiber, say 300 m, common in dense urban networks, does not significantly improve service availability, because cable failure rates are proportional to their length.

Reliability of Free Space Optics (FSO) links, sometimes used for short-distance, line-of-sight communications, can be seriously degraded by unfavorable weather conditions like rain, haze and fog. The paper *Relay-assisted WDM-FSO System: A Better Solution for Communication under Rain and Haze Weather Conditions* by N. Dayal, P. Singh, and P. Kaur includes an analysis (relying on computer models) of how the availability of an FSO link can be improved by inclusion of relays (repeaters or optical amplifiers) to compensate for additional attenuation caused by fog, rain, etc. As in the case of the FTTH network described in the previous paper, this improvement requires additional expenditures on equipment and installation work.

The next group of five papers covers diverse IT and software issues.

First, a novel variant of cloud computing, involving a cloud of smartphones and other mobile devices instead of a set of stationary data centers, is proposed by L. Siwik, D. Kala, M. Godzik, W. Turek, A. Byrski, and M. Kisiel-Dorohinicki in their paper *Mobile Cloud for Parallel and Distributed Green Computing*. While mobile devices are definitely no match, in terms of computing power and 24/7 availability, for dedicated servers and data centers employed in more conventional cloud computing, they have certain advantages, such as location awareness, potentially large number of devices comprising the cloud, some sensor

functionality and the ability of ad-hoc assembly to work on a local, time-critical problem. An actual computing cluster was constructed using dedicated software and its scalability and efficiency were measured. Whether the idea will catch on, nobody knows, but it is definitely new.

Two subsequent papers are devoted to problem solving and optimization issues.

The *Monte Carlo Tree Search Algorithm for the Euclidean Steiner Tree Problem* by M. Bereta presents a novel Monte Carlo Tree Search algorithm for solving the problem of minimal Euclidean Steiner tree on a plane. The goal is to find connections between all the given points (terminals) on a plane, such that the sum of the lengths of edges is as low as possible, while an addition of extra points (Steiner points) is allowed. This is a very important problem in the design and optimization of telecom cable networks, water, gas and electric power distribution systems, etc., where the construction expenses, demand for materials, failure rates, and maintenance costs are approximately proportional to the total length of routes in a given network. The new algorithm combines Monte Carlo Tree Search with proposed heuristics and works better than both the greedy heuristic and pure Monte Carlo simulations. The results of numerical experiments for randomly generated and benchmark library problems (from OR-Lib) are presented and discussed.

The main drawback of the Batch Back Propagation (BBP) algorithm, widely used in neural network training, is that training is slow and that several parameters must be adjusted manually. The BBP algorithm also suffers from saturation training, finding a local minimum instead of a global one. M. S. Al Duais and F. S. Mohamad in their paper *Dynamically-adaptive Weight in Batch Back Propagation Algorithm via Dynamic Training Rate for Speedup and Accuracy Training* present an attempt to improve the speed of training and avoid the saturation effect. A Dynamic Batch Back Propagation (DBBPR) algorithm with a dynamic training rate is introduced. Results of Matlab simulations show that the new algorithm is much better than BBP in terms of training speed and accuracy.

Software testing is getting more and more complicated and time consuming with expansion of the size and functionality of the programs assessed. One of the widely adopted methods is the so called "mutant testing", where small, random changes (mutations) are made to the code which is later executed in parallel with the unmodified version, and their operation is compared. This technique is analyzed by L. T. My Hanh, N. T. Binh, and K. T. Tung in the paper titled *Parallel Mutant Execution Techniques in Mutation Testing Process for Simulink Models*". The authors propose three strategies for parallel execution of mutants on multicore machines using the Parallel Computing Toolbox (PCT) with the Matlab Distributed Computing Server, and demonstrate that the computationally intensive software testing schemes, such as mutation, can be facilitated by employment of parallel processing.

The popularity of online services involving money and sensitive data: banking, commerce, database access, remote work, system management, etc., brings with it the familiar and dreadful issue of computer crime and password theft. Various methods of generating and entering more secure passwords are being proposed and tested, including rather complicated biometric techniques. However, there is a trade-off between security and complexity (and error rate) of a given password scheme, so a relatively simple solution is being looked at as well. One of them is entering a conventional password being a classic string of characters, but with a variable (long and short) duration of breaks between them. The corresponding password recognition process is time-sensitive, and stealing the character string alone (as performed by existing spying software) is not enough for a successful login, and the attack is immediately detected. This technique is presented by K. W. Mahmoud, K. Mansour and A. Makableh in the paper *Detecting Password File Theft using Predefined Time-Delays between Certain Password Characters*. However, currently this is only a proposal in need of verification in a real environment, and one must state that writing a spyware capable of recording relative time spacing between characters will be relatively simple.

While traditional postal services appear to be more and more outdated in the 21st century, being steadily displaced by e-mail and other forms of electronic communications, they are still in demand, in particular for some business and advertising applications. For business customers, it is the quality of postal service which counts, and this quality needs to be monitored using standardized methods. The last paper *Evolution of Measurement of the Single Piece Mail Transit Time* by R. Kobus and F. Raudszus presents how the measurement of the transit time (the main parameter of interest in judging the quality of domestic and

cross-border postal services) of priority mail for home and small business customers in the E.U. has been performed and how it evolved since its introduction in 1994. The test is performed with a set of letters, posted and received by an independent test panel, which are addressed to multiple geographical segments. However, test details must be adjusted in the case of small member states with limited volume of postal traffic and special local conditions. Testing can be made cheaper and more accurate if the test letters are replaced with monitoring of real mail in transit. Alas, mail processing is not completely automated yet, making such a move premature.

Krzysztof Borzycki
Associate Editor

https://doi.org/10.26636/jtit.2017.preface4

Paper

An Improved Downlink MC-CDMA System for Efficient Image Transmission

Mohammed Sofiane Bendelhoum[1,2], Ali Djebbari[2], Ismail Boukli-Hacene[3], and Abdelmalik Taleb-Ahmed[4]

[1] *Department of Technologie, Sciences Institute, University Center Nour Bachir of El-Bayadh, Algeria*
[2] *Telecommunications and Digital Signal Processing Laboratory, Djillali Liabes University, Algeria*
[3] *Biomedical Laboratory, Abou Bekr Belkaid University, Tlemcen, Algeria*
[4] *Laboratoire LAMIH UMR CNRS 8530, Valenciennes et du Hainaut Cambrésis University, Valenciennes, France*

https://doi.org/10.26636/jtit.2017.110917

Abstract—Image compression is an essential stage of the data archiving and transmitting process, as it reduces the number of bits and the time required to complete the transmission. In this article, a study of image transmission over the Multi-Carrier Code Division Multiple Access (MC-CDMA) downlink system is presented. The solution proposed relies on source coding combined with channel coding. The Discrete Wavelet Transform (DWT) method is used in conjunction with the SPIHT coder to compress the image, then the data generated is transmitted with the MC-CDMA technique over a noisy channel. The results show that image transmissions performed over MC-CDMA using the SPIHT model are better than the traditional approach like MC-CDMA in the AWGN channel.

Keywords—data compression, MC-CDMA, MSSIM, performance evaluation, PSNR, SNR, SPIHT.

1. Introduction

One of the most important results of Shannon's work [1] was that source coding and channel coding can be treated separately without sacrificing fidelity [2]. Traditionally, the source and channel codes are designed separately and then cascaded together [3].

Multi-carrier modulation such as Orthogonal Frequency Division Multiplexing (OFDM) has a promising future as a new technology in next-generation wireless communication systems [4], [5]. Combining OFDM with Code Division Multiple Access (CDMA) results in the Multi-carrier CDMA system [6], [7]. This system has received much attention among researchers. However, it suffers from Multiple Access Interference (MAI) in a multi-user setting, which decreases the overall Bit Error Rate (BER) performance [5]. Multi-user Detection (MUD) techniques have been introduced to mitigate MAI to improve the system performance [8], [9].

Several studies in transmission of images and multimedia using MC-CDMA technology are considered [5], [10], [11]. In [10], a study of real-time image traffic over a radio link using the wavelet transform (WT) to compress images and a CDMA link to transfer these images over wireless communication networks was presented. Images compressed with JPEG2000 are transmitted at low SNRs with the MC-CDMA technique and are considered in [11]. An efficient approach for transmitting images over MC-CDMA systems was examined in [5]. Furthermore, several coding techniques have been investigated for efficient transmission of images with MC-CDMA over wireless channels [10], [12], [13]. Despite the efficiency of these coding techniques, they add much redundancy to the transmitted data, which increases the bandwidth and reduces the channel utilization [5].

This paper studies an unage transmission efficiency scheme relying on the MC-CDMA system using discrete wavelet transforms (DWT) and set partitioning in hierarchical trees (SPIHT) coding in the Additive White Gaussian Noise (AWGN) channel. The results are compared with the traditional MC-CDMA technique. In the simulation conducted, the images are issued with the use of both techniques, and their reception is objectively evaluated by means of PSNR and MSSIM parameters. The transmission of images is done with a different SNR.

The remainder of the paper is organized as follows: in Section 2, a description of decomposition of the image using DWT is presented. Section 3 exposes the SPIHT encoder. The parameters used for judging the quality of images are introduced in Section 4. In Section 5 the MC-CDMA model used for transmission and reception is given. Section 6 presents experimental and performance results. Finally, Section 7 offers conclusions and remarks.

2. Wavelet Transform

The emergence of wavelets has led to the convergence of linear expansion methods used in signal processing and applied mathematics. Subband coding methods and their associated filters are closely related to the wavelet concept [13].

A wavelet is a foundation for representing images in various degrees of resolution. Wavelet transform is a mathematical function dividing an image into two various frequency components matching the resolution. The wavelet transformation methodology has been used because of the disadvantages of Fourier transformation [14]. Wavelet transformation has been classified, as mentioned earlier, as DWT and continuous wavelet transforms (CWT). A wavelet is represented as a multi-resolution level, where each analysis is implemented through high-pass and low-pass filters, where each high-pass filter is based on wavelets and low-pass filter is based on scaling functions. The wavelet transform function is based on the conversion of a one-dimensional function into two-dimensional spaces involving translation and dilation parameters related to time and scale factors [15].

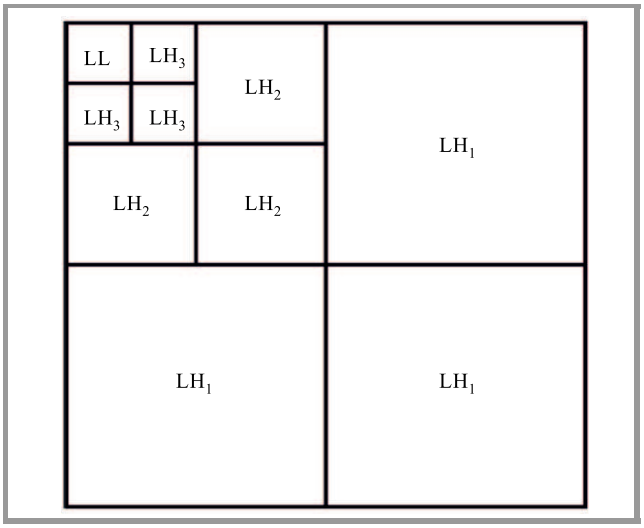

Fig. 1. Pyramidal structure of 3-level wavelet decomposition.

Wavelet analysis of an image can be viewed in the frequency domain as partitioning it into a set of subbands, where each partitioning step is obtained by applying the 2D wavelet transform. One level of 2D wavelet transforms results in four sets of data (wavelet coefficients) that correspond to four 2D frequency subbands. For these four subbands, if the original image data is on the zero-decomposition level (scale), we use the following notation on the k-th decomposition level: HHk (high–high or diagonal details), HLk (high–low or horizontal details), LHk (low–high or vertical details), LLk (low–low or approximation). The LLk subband is also referred to as image approximation, as it represents the image on a lower scale, while to other subbands are referred as to image details. Wavelet decomposition is dyadic when only the LLk subband is further transformed. It results in a new set of subbands: HHk+1, HLk+1, LHk+1, LLk+1. Dyadic decomposition used in image compression will thus generate a hierarchical pyramidal structure, as shown in Fig. 1. If the dyadic decomposition of N levels is performed (low–low subband transformation performed N times), the result will be 3 N+1 subbands [16].

3. SPIHT Coding Scheme

In 1996, Said and Pearlman introduced the SPIHT coding scheme [17]. SPIHT is viewed as one of the most efficient wavelet-based image compression algorithms available in the literature [17], [18]. It can be a progressive transmission and it may be relevant for the lossless compression area of interest. It takes advantage of the spatial orientation of the zerotree coding that was proposed by Shapiro [19] and has been shown to provide better performance [20]. In addition, SPIHT can compress images in a lossless manner, as well as with a loss [21].

SPIHT [22] is one of the most advanced schemes available, even outperforming the state-of-the-art JPEG 2000 in some situations. The basic principle is the same – progressive coding is applied, processing the image respectively to a lowering threshold. The difference is in the concept of zerotree (spatial orientation trees in SPIHT).

This is an idea that takes into consideration relations between coefficients across subbands at different levels [17]. The first idea is always the same. If there is a coefficient at the highest level of the transform in a particular subband which is considered insignificant against a particular threshold, it is very probable that its descendants in lower levels will be insignificant too. Therefore, we can code quite a large group of coefficients with one symbol.

Figure 2 shows how a spatial orientation tree is defined in a pyramid constructed with recursive four subbands splitting. The coefficients are ordered hierarchically. According to this relationship, the SPIHT algorithm saves many bits that specify insignificant coefficients [23].

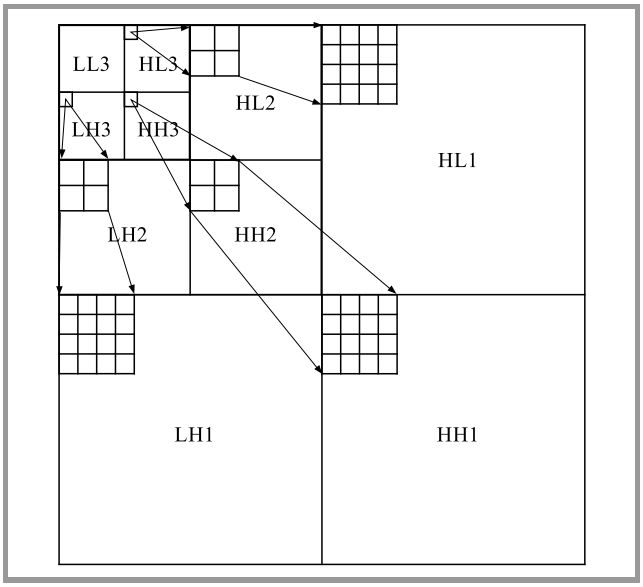

Fig. 2. Parent-child relationship.

The SPIHT flowchart is presented in Fig. 3. In the first step, the original image is decomposed into subbands. Then, the method finds the maximum and the iteration number. In the second step, the method puts the wavelet coefficients into a sorting pass that finds all significant values and encodes

their sign. In the third step, the significant coefficients that were found in the sorting pass are put into the refinement pass that uses two bits to exactly reconstruct the value for closing to the real value [24].

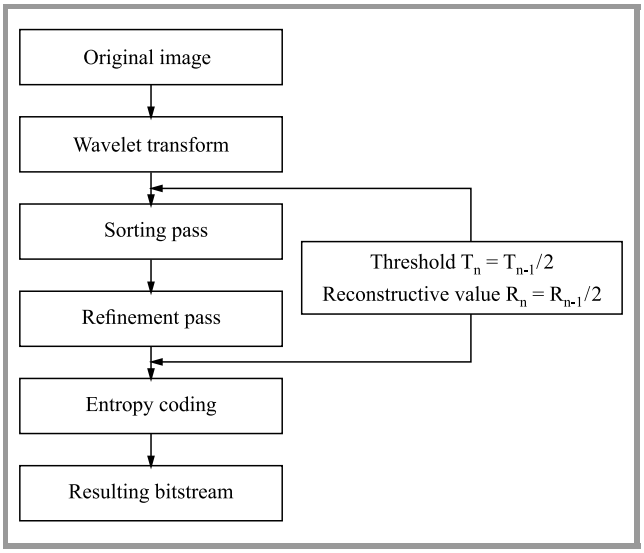

Fig. 3. Flowchart of SPIHT wavelet transform.

4. Quality Evaluation

The peak signal to noise ratio (PSNR) is the most commonly used reconstruction quality measure in image compression:

$$\mathrm{PSNR} = 10 \log \frac{(Dynamics\ of\ image)^2}{MSE}. \tag{1}$$

Mean square error (MSE) which requires two $M \times N$ gray scale images I and $\hat{I}$ where one of the images is considered as a compression of the

$$\mathrm{MSE}^2 = \frac{1}{M \times N} \sum_{i=1}^{N} \sum_{j=1}^{M} \left(I(i,j) - \hat{I}(i,j)\right)^2. \tag{2}$$

Usually an image is encoded by 8 bits. It is represented by 256 gray levels, which vary between 0 and 255, with the extent or dynamics of the image being 255.

The PSNR measurement gives a numerical value on the damage, but it does not describe its type. Moreover, as is often noted in [25], [26], it does not quite represent the quality perceived by human observers. For different applications where images that are degraded must eventually be examined by experts, traditional evaluation remains insufficient. For this reason, objective approaches are needed to assess the image quality. Then, we evaluate a new paradigm to estimate the quality of images, specifically the ones compressed by wavelet transform based on the assumption that the human visual system (HVS) is highly adapted to extract structural information. The similarity compares the brightness, contrast and structure between each pair of vectors, where the structural similarity index (SSIM) between two signals x and y is given by [27], [28]:

$$\mathrm{SSIM}(x,y) = l(x,y)\, c(x,y)\, s(x,y). \tag{3}$$

Finally, the quality measurement can provide a spatial map of the local image quality, which gives more information on the image quality degradation. For application, we require a single overall measurement of the whole image quality that is given by the following formula:

$$\mathrm{MSSIM}(I,\hat{I}) = \frac{1}{M} \sum_{i=1}^{M} \mathrm{SSIM}(I,\hat{I}), \tag{4}$$

where I and $\hat{I}$ are respectively the reference and degraded images respectively, I_i and $\hat{I}_i$ are the contents of images at the i-th local window. M is the total number of local windows in the image. The MSSIM values exhibit greater consistency with the visual quality.

5. Model MC-CDMA System

In general, there are three types of hybrid schemes, e.g. Multi-Carrier Direct Sequence CDMA (MC/DS-CDMA), Multicarrier-CDMA (MC-CDMA) and Orthogonal Frequency Code Division Multiplexing (OFCDM) [29]. The basic MC-CDMA signal is generated by a serial concatenation of classical Direct-Sequence CDMA (DS-CDMA) and Orthogonal Frequency Division Multiplexing (OFDM). Each chip of the direct sequence spread data symbol is mapped to a different subcarrier. Thus, with MC-CDMA, the chips of a spread data symbol are transmitted in parallel on different sub-carriers, in contrast to serial transmission with DS-CDMA [5], [30]. The MC-CDMA system model using for transmission data image is divided into four stages, including the image data compression stage using DWT and SPHIT, the BPSK modulation stage, the spreading stage using the Walsh code, and the OFDM modulation stage, as shown in Fig. 4.

At the transmitter, the bit stream generated after compression step $b_i^{(k)}$ is modulated by BPSK $B_i^{(k)}$. The BPSK modulated information bits of each user $k = 0, \ldots, k-1$, are spread by the corresponding spreading code, where Walsh-Hadamard codes are applied. In the downlink system, the spread data of all active users are added synchronously and mapped on to the L_c sub-carriers [7]. The sequence, $B_i^{(k)}$ is applied to a serial-to-parallel converter and multiplied by the k with the user specific spreading code of Walsh $C_i^{(k)}$ of length L [5], [7], In [31], [7] confirmed that the Walsh-Hadamard codes are the best choice for the downlink, also for the minimization of multiple access interference.

$$C_i^{(k)} = \left(C_0^{(k)}, C_1^{(k)}, \ldots, C_{L-1}^{(k)}\right)^T. \tag{5}$$

The chip rate of the serial spreading code of Walsh $C_i^{(k)}$ before serial-to-parallel conversion is [7]:

$$\frac{1}{T_c} = \frac{L}{T_B}. \tag{6}$$

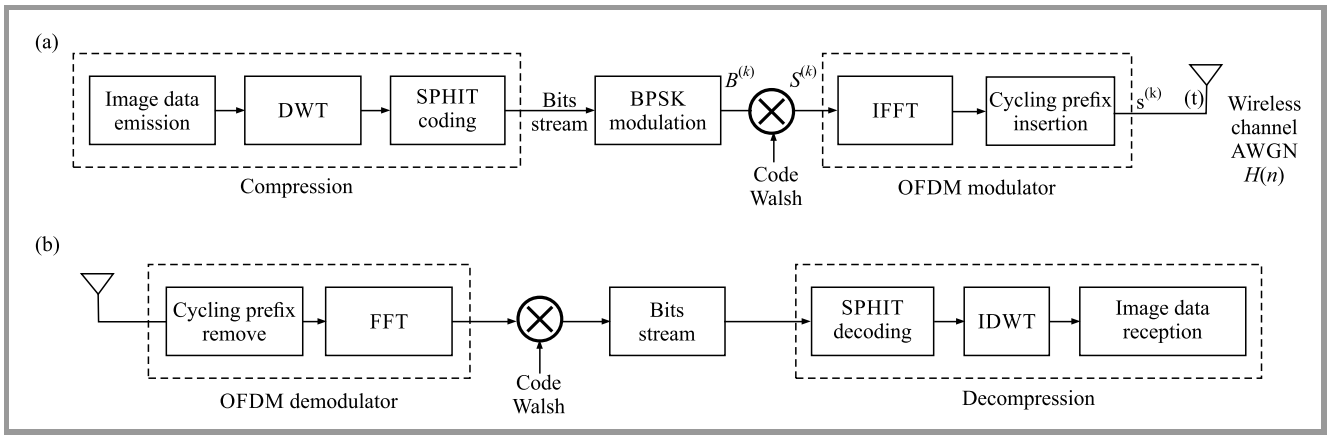

Fig. 4. The MC-CDMA system model for image transmission: (a) transmitter, (b) receiver.

The chip rate is L times higher than the data symbol rate $\frac{1}{T_B}$. The complex valued sequence obtained after spreading can be expressed in a vector notation as [7]:

$$s_i^{(k)} = B_i^{(k)} \cdot C_i^{(k)} = \left(s_0^{(k)}, s_1^{(k)}, \ldots, S_{L-1}^{(k)}\right)^T. \tag{7}$$

Then $s_i^{(k)}$ is modulated by the OFDM modulator, the input signal of the OFDM operation is statistically independent [7]. The result is $S_i^{(k)}$. The advantage of the frequency domain channel model is that the IFFT and FFT operation for OFDM and inverse OFDM can be avoided, and the fading operation results in one complex-valued multiplication per subcarrier [7]. The spectral containment of the channels is better since they are not using cyclic prefix [32] and it reduces the Inter-Symbol Interference (ISI) effect.

Signal H is obtained by adding the $S_i^{(k)}$ from K users:

$$H = \sum_{1=0}^{k} S_i^{(k)}. \tag{8}$$

The result H is applied to a parallel-to-serial converter. The transmitted vector T_X is given by:

$$T_X(t) = H_c + n, \tag{9}$$

where H_c is the data channel vector and n is the noise AWGN vector.

The steps used to transmit data images using the MC-CDMA model can be summarized as follows:

1. The data image is transformed by DWT in the same dimension as the original image.
2. The result of the step 1 is associated with SPIHT coding to generate the bits stream to be transmitted also at a different bit rate of R_c.
3. It is assumed that all subchannels are identically modulated using binary phase-shift keying (BPSK) [33].
4. The most important characteristic of the sequence in the MC-CDMA system is the zero-cross correlation property. The most widely used orthogonal code is the Walsh-Hadamard code [12], which is generated by using the Hadamard transform defined as:

$$C_0 = [0], \tag{10}$$

$$C_{2m} = \begin{bmatrix} C_m & C_m \\ C_m & \overline{C_m} \end{bmatrix}. \tag{11}$$

This transform gives matrix, C_m, for $m = 2^i$ matrix, where i is an integer. For example, consider a case of $n = 4$:

$$C_4 = \begin{bmatrix} C_{2m} & C_{2m} \\ C_{2m} & \overline{C_{2m}} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}. \tag{12}$$

5. An OFDM modulation step is performed on the binary data.
6. At the receiver, after OFDM demodulation step. The vector is multiplied by the $C_i^{(k)}$ code Walsh for k-th user, and we applied the SPIHT decoding method. The inverse discrete wavelet transform (IDWT) is used to receive the data image.

   Finally, the decompressed image is given at the transmitted bit rate of R_c.

6. Results and Discussion

After various applications intended to show the actual performance of the method proposed, we will make a comparison between the following, different types of wavelet trans-

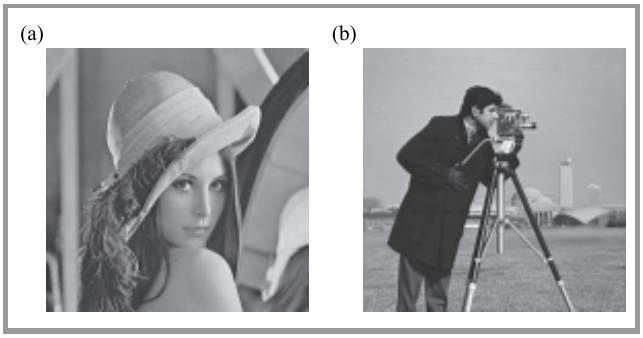

Fig. 5. Typical test pictures: (a) Lena, (b) cameraman.

forms: Daubechies (db5), symelets (sym9), Coiflets (coif5), Bi-orthogonal (bior4.4) and Cohen-Daubechies-Feauveau (CDF9/7). They are associated with an SPIHT coder in the Subsection 6.1. The Lena and Cameraman images (Fig. 5) with 128×128 pixels (grayscale) are encoded (8 bits per pixel) for each application. In the step compression step, we vary the bitrate from 0.125 to 1 and transmit them via the MC-CDMA model, at different SNRs – from 5 to 20 dB – in 5 steps. The quality of the images transmitted and received is evaluated objectively with PSNR and MSSIM parameters for different bit rates R_c.

Subsection 6.3 gives the performance of the MC-CDMA system in the AWGN channel with multiple users. The simulation values are depicted in a graph/plot of BER vs. Eb/N0 (in dB), which is the energy per bit per noise spectral density [12], [13]. It defines the SNR per bit and it is an important measurement to evaluate and compare different digital communication systems.

The CDF9/7 wavelet is implemented with a SPIHT coder. Tree 128×128 images are used: IRM1, IRM2 and IRM3. The original image is compressed with different bit rates R_c and transmitted at different SNRs for 10 to 17 dB in 1 dB steps. The image transmitted and reconstructed at different SNRs is estimated with PSNR and MSSIM factors.

In calculations and numerical simulations, the same images as those treated in [5] are used to evaluate the performance of the simulation model presented.

6.1. Wavelets Association with SPIHT in MC-CDMA

The average PSNR and MSSIM values for the Lena image at different R_c are shown in Tables 1–5, enabling the judgment of quality of the SPIHT compressed image after transmission, as compared to its quality before transmission.

A plot of the variation of PSNR and MSSIM for the Lena image with different SNRs at R_c constant is shown in Figs. 6–7.

Table 1
PSNR and MSSIM values for the Lena image at different R_c and SNR with bior4.4 and SPIHT

Parameter of evaluation	R_c [bpp]	Before transmission	Transmission with SNR [dB]			
			5	10	15	20
PSNR [dB]	0.125	30.54	27.56	27.70	30.47	30.54
MSSIM		0.662	0.076	0.164	0.661	0.662
PSNR [dB]	0.25	31.19	27.51	28.59	29.03	31.19
MSSIM		0.749	0.066	0.117	0.405	0.749
PSNR [dB]	0.5	32.53	28.01	27.70	29.83	32.53
MSSIM		0.840	0.081	0.138	0.557	0.840
PSNR [dB]	0.75	34.35	27.36	28.03	29.65	34.35
MSSIM		0.899	0.059	0.214	0.537	0.899
PSNR [dB]	1	35.76	28.91	27.76	29.09	35.76
MSSIM		0.927	0.028	0.177	0.415	0.927

Table 2
PSNR and MSSIM values for the Lena image at different R_c and SNR with coif2 and SPIHT

Parameter of evaluation	R_c [bpp]	Before transmission	Transmission with SNR [dB]			
			5	10	15	20
PSNR [dB]	0.125	30.48	28.76	29.04	30.27	30.48
MSSIM		0.663	0.095	0.114	0.646	0.663
PSNR [dB]	0.25	31.32	29.19	30.70	31.17	31.32
MSSIM		0.753	0.070	0.126	0.746	0.753
PSNR [dB]	0.5	32.63	29.11	29.24	31.40	32.63
MSSIM		0.849	0.120	0.121	0.743	0.849
PSNR [dB]	0.75	34.44	29.15	29.55	31.68	34.44
MSSIM		0.902	0.133	0.138	0.520	0.902
PSNR [dB]	1	35.90	29.16	29.60	32.55	35.90
MSSIM		0.928	0.135	0.135	0.827	0.928

Table 3
PSNR and MSSIM values for the Lena image at different R_c and SNR with with sym9 and SPIHT

Parameter of evaluation	R_c [bpp]	Before transmission	Transmission with SNR [dB]			
			5	10	15	20
PSNR [dB]	0.125	30.34	28.30	28.48	28.94	30.34
MSSIM		0.655	0.030	0.149	0.408	0.655
PSNR [dB]	0.25	31.25	28.40	28.52	30.16	31.25
MSSIM		0.750	0.039	0.152	0.635	0.750
PSNR [dB]	0.5	32.69	28.48	28.59	29.52	32.69
MSSIM		0.847	0.053	0.138	0.501	0.847
PSNR [dB]	0.75	34.56	28.05	28.10	28.82	34.56
MSSIM		0.903	0.054	0.142	0.327	0.903
PSNR [dB]	1	36.02	28.58	29.00	31.22	36.02
MSSIM		0.938	0.079	0.143	0.761	0.938

Table 4
PSNR and MSSIM values for the Lena image at different R_c and SNR with bior4.4 and SPIHT

Parameter of evaluation	R_c [bpp]	Before transmission	Transmission with SNR [dB]			
			5	10	15	20
PSNR [dB]	0.125	30.54	27.56	28.38	29.54	30.54
MSSIM		0.672	0.076	0.185	0.661	0.672
PSNR [dB]	0.25	31.35	28.16	29.27	30.52	31.35
MSSIM		0.766	0.037	0.120	0.690	0.766
PSNR [dB]	0.5	32.85	28.69	27.87	30.77	32.85
MSSIM		0.858	0.050	0.206	0.695	0.858
PSNR [dB]	0.75	34.74	27.25	29.73	30.54	34.74
MSSIM		0.911	0.042	0.087	0.650	0.911
PSNR [dB]	1	36.37	27.87	28.37	30.29	36.37
MSSIM		0.939	0.040	0.134	0.664	0.939

Table 5
PSNR and MSSIM values for Lena image
at different R_c and SNR with CDF9/7 and SPIHT

Parameter of evaluation	R_c [bpp]	Before trans-mission	Transmission with SNR [dB]			
			5	10	15	20
PSNR [dB]	0.125	30.44	28.12	28.98	30.29	30.44
MSSIM		0.680	0.032	0.063	0.655	0.680
PSNR [dB]	0.25	31.59	27.52	27.94	31.10	31.59
MSSIM		0.783	0.072	0.108	0.764	0.783
PSNR [dB]	0.5	32.96	27.14	28.25	30.14	32.96
MSSIM		0.870	0.047	0.081	0.631	0.870
PSNR [dB]	0.75	35.19	27.44	27.81	31.29	35.19
MSSIM		0.917	0.073	0.093	0.738	0.917
PSNR [dB]	1	36.92	27.85	29.27	29.51	36.92
MSSIM		0.944	0.050	0.073	0.480	0.944

For all figures the variations of PSNR and MSSIM, increases proportionally with the SNR. For the lower SNR (10 dB) the images that are received are of very poor quality (low MSSIM). In the case of SNR greater than 10 dB, the quality of images that are received increases until they reach the maximum value obtained without a transmission (SNR = 20 dB).

As far as the association of different types of wavelets (db5, coif2, sym9, bior4.4 and CDF9/7) with the SPHIT encoder at different bit rates (Rc) is concerned, let's say that the CDF9/7 wavelet provides the most important PSNR and MSSIM values compared to those obtained by other wavelets for a SNR = 20 dB [20] (except for $R_c = 0.125$ bpp, where PSNR = 30.44 dB).

The CDF9/7 wavelet will be subsequently retained for various images with different R_c transmission in a telecommunications MC-CDMA system. The received images will be evaluated objectively by PSNR and MSSIM.

6.2. Analysis of MC-CDMA Performance in a Wireless Channel

Figure 8 shows the BER performance comparison for multi-user cases in the AWGN channel with the MC-CDMA model used and traditional MC-CDMA. In theory, BER lowers with increasing Eb/N0. It is observed that BER performance decreases as Eb/N0 increases. This is because

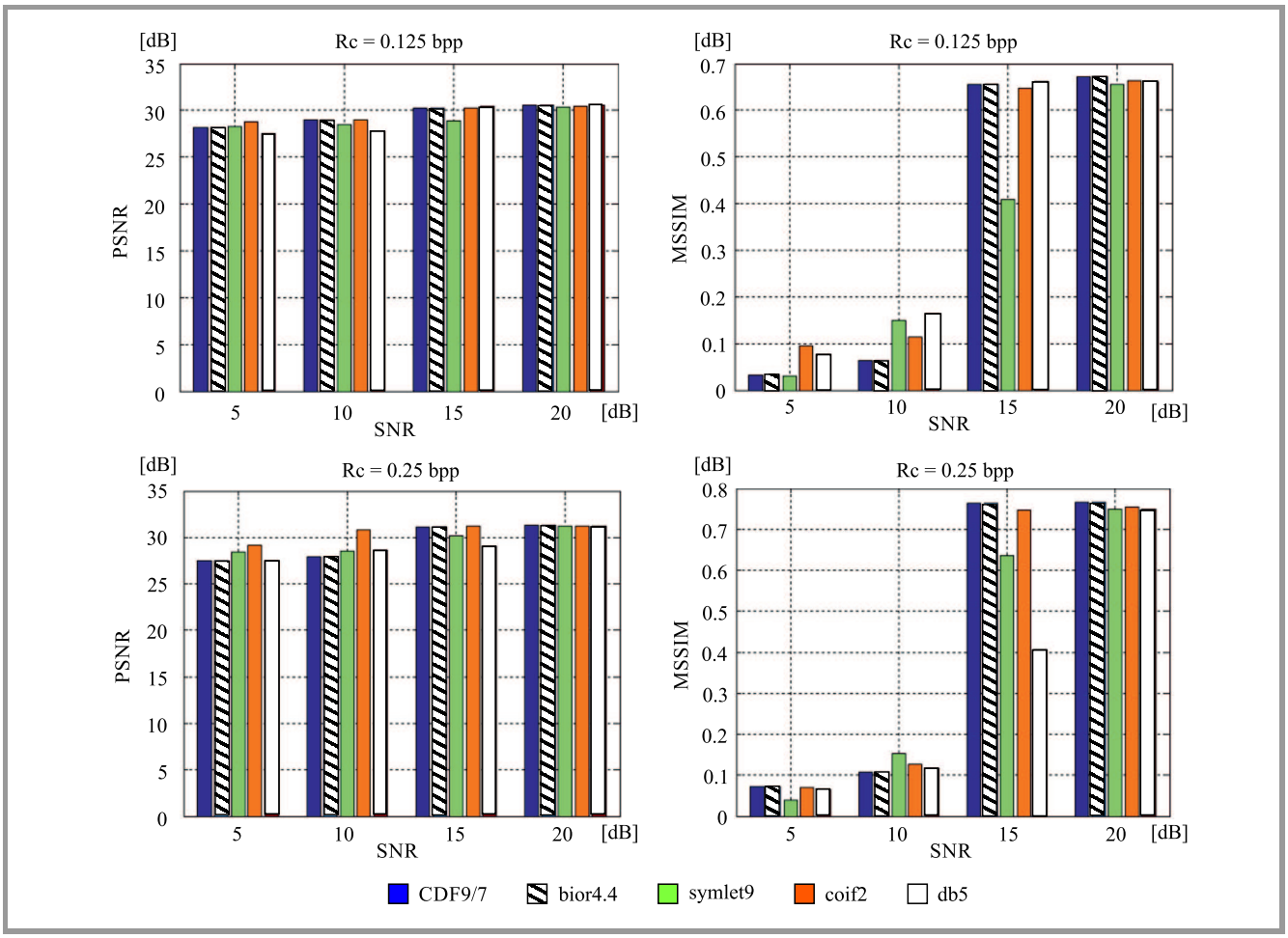

Fig. 6. Variation of PSNR and MSSIM for the Lena image with 0.125 and 0.25 bpp. (See color pictures online at www.nit.eu/publications/journal-jtit)

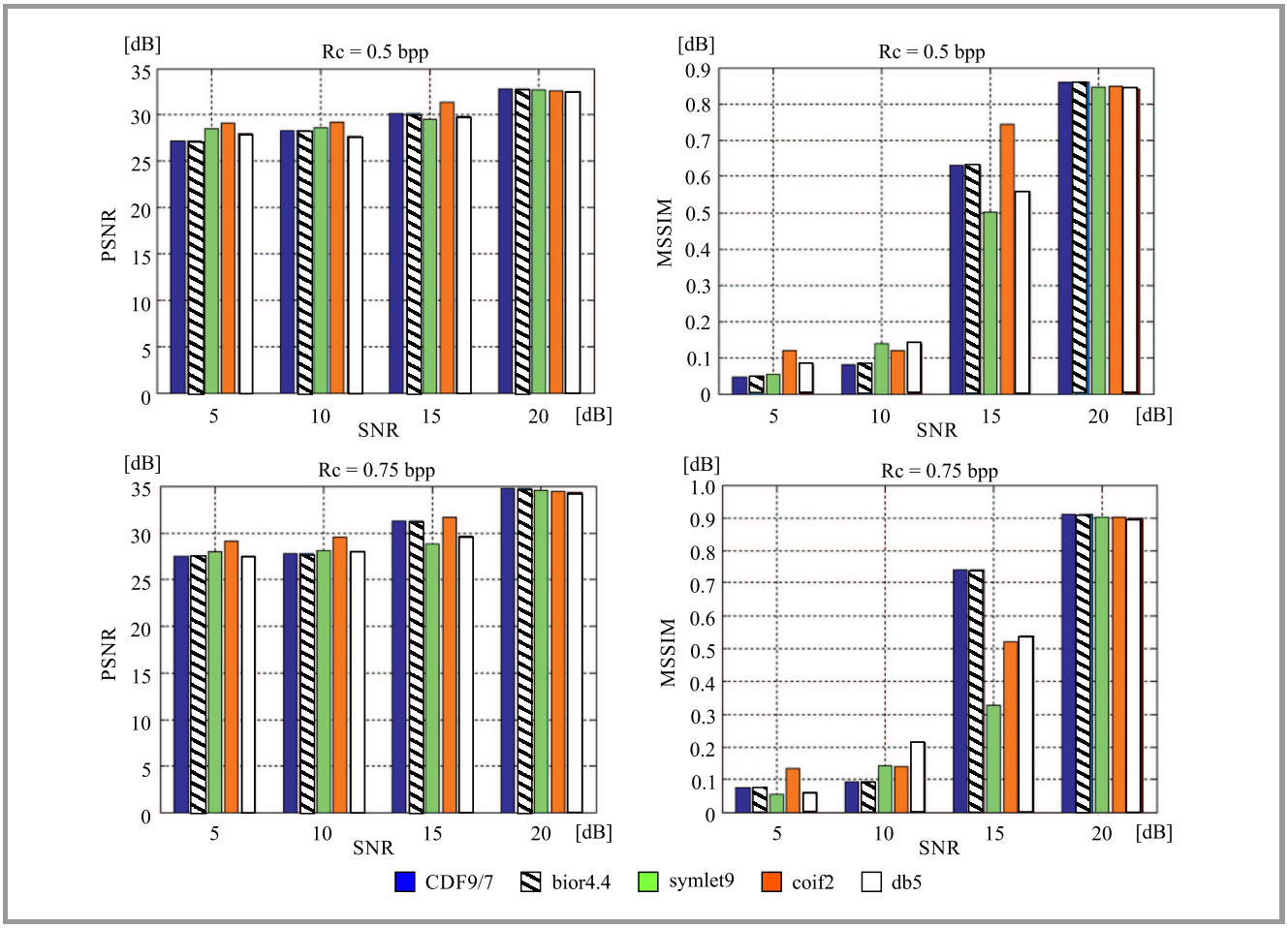

Fig. 7. Variation of PSNR and MSSIM for the Lena image with 0.5 and 0.75 bpp.

a lower AWGN noise level can be obtained with a higher Eb/N0. This is due to the fact that BER performance is only affected by AWGN noise [12], [13].

In the case of a low SNR, the two transmission systems show the same performance. As SNR increases, as it is clear than this transmission scheme offers better performance in terms of BER terminology. The BER results show that the proposed transmission scheme is based on the SPHIT coder coupled to MC-CDMA, and is superior to that obtained from the traditional MC-CDMA.

In the MC-CDMA model proposed the SNR increases to 15 dB, in comparison with the traditional MC-CDMA technique, where the SNR reaches 22 dB, which means a gain of about 7 dB.

6.3. Performance Analysis of CDF9/7 with SPIHT coder MC-CDMA System

Several experiments were performed to test the performance and the robustness of the medical images transmitted through the MC-CDMA system using the CDF9/7 and SPHIT coder. The results of the simulation are presented in Tables 6–8 for all image users (IRM1, IRM2 and IRM3).

The IRM1 image transmission via a wireless MC-CDMA system with the image compressed at 0.125 bpp needs the SNR = 15 dB to achieve the maximum PSNR and MSSIM values before the transmission. On the other hand, for the $R_c = 0.25$, 0.5, 0.75 and 1 bpp we need the SNR = 16 dB for a perfect transmission in our MC-CDMA system. The same remarks are noted in the case of IRM3 images.

For the transmission of the IRM2 image with $R_c =$ 0.125 bpp and 0.25, the SNR = 15 dB is necessary, but for R_c bpp greater than 0.25 all results show that the SNR = 16 dB can perfectly transmit all images.

Examples of the received images are shown in Figs. 9–11 – for SNR = 14, 15 and 16 dB.

Using human eyesight only, we can see that the images recovered are not identical with the originals, once the SNR is lowered to 14 dB. One can observe that the best quality is obtained at SNR = 16 dB, in this case the MC-CDMA model is considered as a reliable wireless communications system. The PSNR of the received images are exceeds 30 dB, which means that we can also transmit an image using $R_c = 0.75$ bpp and obtain a good level of visual quality (MSSIM $\geq$ 0.8) of the received image.

All results obtained by transmitting the images over the MC-CDMA model are better than those relying on the traditional MC-CDMA technique.

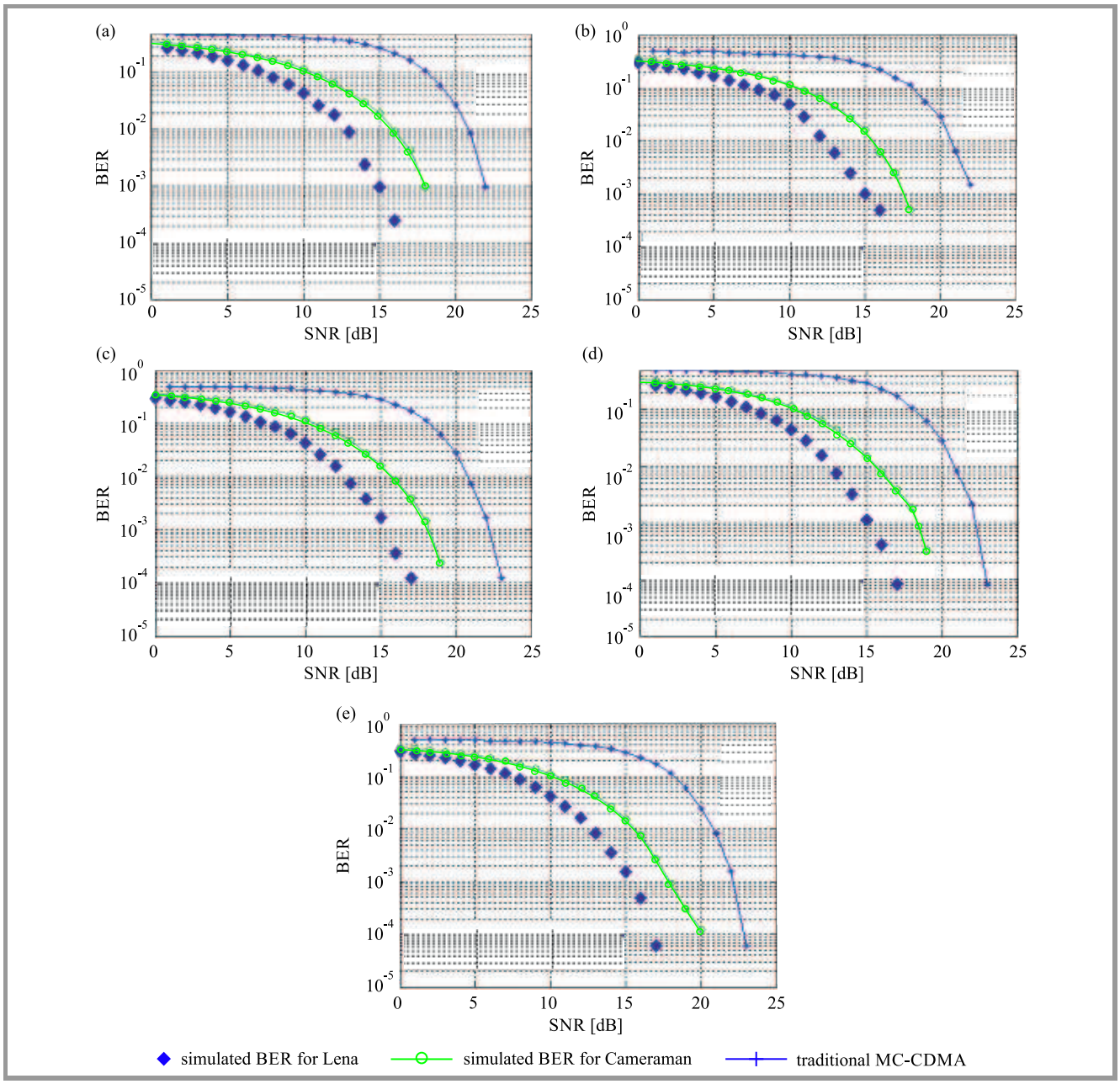

Fig. 8. BER vs. Eb/N0 for multi-user environment in AWGN channel with Lena and Cameraman image coded at: (a) 0.125 bpp, (b) 0.25 bpp, (c) 0.5 bpp, (d) 0.75 bpp, (e) 1 bpp (MC-CDMA proposed), and a single user in traditional MC-CDMA [30].

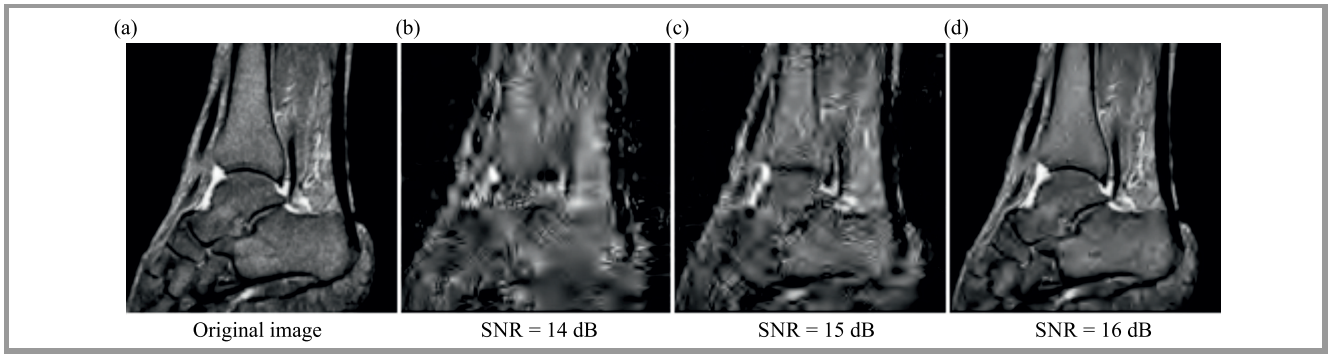

Fig. 9. The received IRM1 images with MC-CDMA using R_c = 0.75 bpp over AWGN channel: (a) original image, (b) PSNR = 28.77 dB, (c) PSNR = 29.81 dB, (d) PSNR = 32.67 dB.

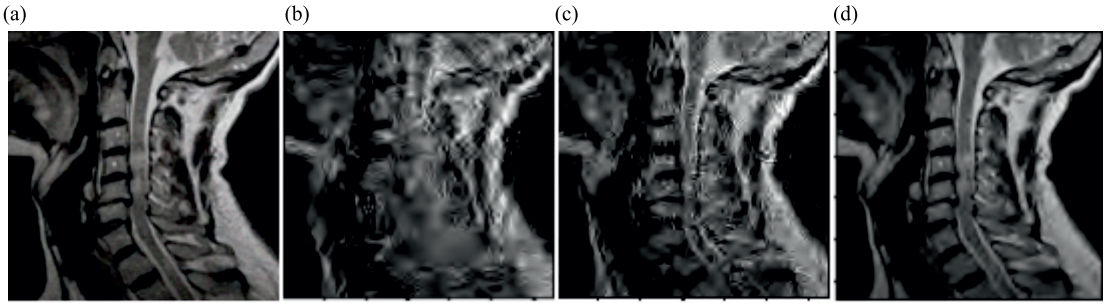

Fig. 10. The received IRM2 images with MC-CDMA using R_c = 0.75 bpp over AWGN channel: (a) original image, (b) PSNR = 28.61 dB, (c) PSNR = 29.35 dB, (d) PSNR = 32.43 dB.

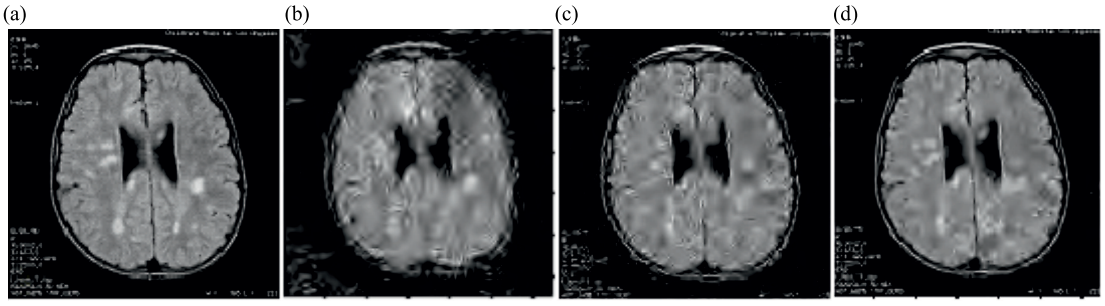

Fig. 11. The received IRM3 images with MC-CDMA using R_c = 0.75 bpp over AWGN channel: (a) original image, (b) PSNR = 29.88 dB, (c) PSNR = 31.87 dB, (d) PSNR = 32.52 dB.

Table 6

Performance results of PSNR and MSSIM values for the IRM1 image at different R_c and SNR

Parameter of evaluation	R_c [bpp]	Before transmission	Transmission with SNR [dB]							
			10	11	12	13	14	15	16	17
PSNR [dB]	0.125	29.86	27.48	27.93	28.15	28.48	29.54	29.86	29.86	29.86
MSSIM		0.545	0.130	0.137	0.172	0.258	0.491	0.545	0.545	0.545
PSNR [dB]	0.25	30.42	27.58	28.41	28.66	28.69	28.85	30.10	30.42	30.42
MSSIM		0.669	0.133	0.143	0.196	0.274	0.342	0.621	0.669	0.669
PSNR [dB]	0.5	31.64	26.96	27.88	28.22	28.73	28.89	29.79	31.64	31.64
MSSIM		0.783	0.126	0.139	0.164	0.872	0.315	0.555	0.783	0.783
PSNR [dB]	0.75	32.67	27.77	27.91	28.02	28.52	28.77	29.81	32.67	32.67
MSSIM		0.852	0.143	0.149	0.153	0.199	0.300	0.571	0.852	0.852
PSNR [dB]	1	33.83	27.72	28.87	28.91	29.11	29.22	29.87	33.83	33.83
MSSIM		0.890	0.110	0.155	0.202	0.228	0.444	0.627	0.890	0.890

7. Conclusions

In this paper, an efficient image transmission system using the downlink MC-CDMA model is presented. The solution proposed transmits images over wireless AWGN channels using DWT associated with the SPIHT coder. According to Shannon's theorem [1], the combination of the optimum source coder and optimum channel coder can create an optimum system, therefore joint design of source coder and channel coder is necessary to optimize communication systems [10]. Therefore, the CDF9/7 wavelet is used with the SPIHT coder to optimize data image compression. We can also transmit images with $R_c = 0.75$ bpp, which equals to the compression rate 90.63% of the original image.

Table 7
Performance results of PSNR and MSSIM values for the IRM2 image at different R_c and SNR

Parameter of evaluation	R_c [bpp]	Before transmission	Transmission with SNR [dB]							
			10	11	12	13	14	15	16	17
PSNR [dB]	0.125	29.48	27.62	27.71	27.68	28.73	29.14	29.48	29.48	29.48
MSSIM		0.546	0.111	0.121	0.167	0.352	0.420	0.546	0.546	0.546
PSNR [dB]	0.25	30.02	27.41	28.20	28.43	28.81	29.90	30.02	30.02	30.02
MSSIM		0.695	0.102	0.122	0.145	0.201	0.667	0.695	0.695	0.695
PSNR [dB]	0.5	31.41	27.58	28.01	28.23	28.84	29.22	30.08	31.41	31.41
MSSIM		0.822	0.078	0.114	0.120	0.290	0.336	0.716	0.822	0.822
PSNR [dB]	0.75	32.43	27.47	27.82	27.96	28.45	28.61	29.35	32.43	32.43
MSSIM		0.877	0.087	0.115	0.145	0.259	0.406	0.607	0.877	0.877
PSNR [dB]	1	33.59	27.54	27.66	28.57	29.07	29.60	30.49	33.45	33.59
MSSIM		0.912	0.093	0.110	0.136	0.220	0.306	0.753	0.906	0.912

Table 8
Performance results of PSNR and MSSIM values for the IRM3 image at different R_c and SNR

Parameter of evaluation	R_c [bpp]	Before transmission	Transmission with SNR [dB]							
			10	11	12	13	14	15	16	17
PSNR [dB]	0.125	30.13	27.75	28.41	29.56	28.48	29.80	30.13	30.13	30.13
MSSIM		0.527	0.148	0.169	0.320	0.258	0.464	0.527	0.527	0.527
PSNR [dB]	0.25	30.59	27.39	27.87	28.35	28.69	28.61	28.87	30.59	30.59
MSSIM		0.630	0.142	0.167	0.208	0.274	0.231	0.343	0.630	0.630
PSNR [dB]	0.5	31.64	27.86	28.12	28.72	28.73	29.95	30.24	31.64	31.64
MSSIM		0.746	0.153	0.181	0.301	0.872	0.537	0.622	0.746	0.746
PSNR [dB]	0.75	32.52	27.59	27.81	28.96	28.52	29.88	31.87	32.52	32.52
MSSIM		0.829	0.123	0.181	0.391	0.199	0.531	0.795	0.829	0.829
PSNR [dB]	1	34.29	27.21	27.57	29.20	29.11	29.51	30.17	34.29	34.29
MSSIM		0.881	0.140	0.177	0.233	0.228	0.263	0.443	0.881	0.881

For a perfect reception of the image, we need the SNR = 17 dB (for natural images). On the other hand, for medical images, the SNR = 16 dB is necessary. This enables to use an optimized wireless communications system.

References

[1] C. E. Shannon, "A mathematical theory of communication", *The Bell System Tech. J.*, vol. 27, no. 3, pp. 379–423, 1948.

[2] C. E. Shannon, "Coding theorems for a discrete source with a fidelity criterion", *IRE Int. Conv. Rec.*, pp. 142–163, 1959.

[3] H. Keang-Po and M. K. Joseph, "Image transmission over noisy channels using multicarrier modulation", *Signal Process.: Image Commun.*, vol. 9, no. 2, pp. 159–169, 1997.

[4] J. Latal *et al.*, "Experimental measurement of thermal turbulence effects on modulated optical beam in the laboratory", in *Proc. 19th OptoElectron. and Commun. Conf. and 39th Australian Conf. on Optical Fibre Technol. OECC/ACOFT 2014*, Melbourne, Australia, 2014, pp. 455–457.

[5] E. M. El-Bakary *et al.*, "Efficient Image Transmission with Multi-Carrier CDMA", *Wireless Pers. Commun.*, vol. 69, no. 2, pp. 979–994, 2013 (doi: 10.1007/s11277-012-0622-6).

[6] H. Schulze and C. Luders, *Theory and Applications of OFDM and CDMA*. New York: Wiley, 2005.

[7] K. Fazel and S. Kaiser, *Multi-Carrier and Spread Spectrum Systems*. Chichester: Wiley, 2003.

[8] S. Hara and R. Prasad, "Overview of multicarrier CDMA", *IEEE Commun. Mag.*, vol. 35, no. 12, pp. 126–133, 1997 (doi: 10.1109/35.642841).

[9] S. Verdu, *Multiuser Detection*. Cambridge, UK: Cambridge University Press, 1998.

[10] P. P. Dang and P. M. Chau, "Robust image transmission over CDMA channels", *IEEE Trans. on Consumer Electron.*, vol. 46, no. 3, pp. 664–672, 2000.

[11] T. Kathiyaiah and T. H. Oh, "Robust JPEG2000 image transmission through low SNR Multi-Carrier CDMA system in frequency-selective Rayleigh fading channels: a performance study", *Int. J. Wavelets Multiresolut Inform. Process.*, vol. 9, no. 2, pp. 283–303, 2011.

[12] T. H. Oh and G. T. Kim, "Image transmission through MC-CDMA channel: an image quality evaluation", *Int. J. Wavelets Multiresolut Inf. Process.*, vol. 6, no. 6, pp. 827–849, 2008.

[13] K. Ramchandran, M. Vetterli, and C. Herley, "Wavelets subband coding and best bases", *Proceedings of the IEEE*, vol. 84, no. 4, pp. 541–560, 1996.

[14] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Pearson-Prentice Hall, 2005.

[15] A. S. Vijendran and B. Vidhya, "A hybrid image compression technique using wavelet transformation – MFOCPN and Interpolation", *Global J. of Com. Science and Technol.*, vol. 11, no. 3, pp. 57–62, 2011.

[16] N. Sprljan, S. Grgic, and M. Grgic, "Modified SPIHT algorithm for wavelet packet image coding", *Real Time Imag.*, vol. 11, no. 5–6, pp. 378–388, 2005.

[17] A. Said and A. P. William, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees", *IEEE Trans. Circuits and Syst. for Video Technol.*, vol. 6, no. 3, pp. 243–250, 1996 (doi: 10.1109/76.499834).

[18] A. Bouridane *et al.*, "A very low bit-rate embedded color image coding with SPIHT", in *Proc. IEEE Int. Conf. on Acoust., Speech and Sig. Process. ICASSP 2004*, Montreal, Quebec, Canada, 2004, pp. 689–692.

[19] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients", *IEEE Trans. on Signal Process.*, vol. 41, no. 12, pp. 3445–3462, 1993.

[20] I. Boukli-Hacene, M. Beladghem, and A. Bessaid, "Lossy compression color medical image using CDF wavelet lifting scheme", *I. J. Image, Graphics and Sig. Process.*, vol. 11, pp. 53–60, 2013 (doi: 10.5815/ijigsp.2013.11.06).

[21] T. Brahimi, A. Melit, and F. Khelifib, "An improved SPIHT algorithm for lossless image coding", *Digit. Sig. Process.*, vol. 19, no. 2, pp. 220–228, 2009.

[22] S. G. Miaou, S. T. Chen, and S. N. Chao, "Wavelet-based lossy-to-lossless medical image compression using dynamic VQ and SPIHT coding", *Biomed. Eng. Appl. Basis Commun.*, vol. 15, no. 6, pp. 235–242, 2003.

[23] Y. Y. Chen and S. C. Tai, "Embedded medical image compression using DCT based subband decomposition and modified SPIHT data organization", in *Proc. 4th IEEE Symp. on Bioinform. and Bioengin. BIBE 2004*, Taichung, Taiwan, 2004 (doi: 10.1109/BIBE.2004.1317339).

[24] M. Beladgham *et al.*, "MRI image compression using Biorthogonal CDF wavelet based on lifting scheme and SPIHT coding", in *Proc. 4th Int. Conf. on Elec. Engin. CIGE'10*, Bechar, Algeria, 2010, pp. 225–232, 2010.

[25] Z. Xiong, K. Ramchandran, and M. Orchard, "Space frequency quantization for wavelet image coding", *IEEE Trans. on Image Process.*, vol. 6, no. 5, pp. 677–693, 1997.

[26] K. A. Navas, M. L. Aravind, and M. Sasikumar, "A novel quality measure for information hiding in images", in *Proc. IEEE Comp. Society Conf. on Comp. Vision and Pattern Recogn. Worksh. CVPRW'08*, Anchorage, AK, USA, 2008 (doi: 10.1109/CVPRW.2008.4562985).

[27] M. Doaa and F. Abou-Chadi, "Image compression using block truncation coding", *J. of Selec. Areas in Telecommun.* (*JSAT*), pp. 9–13, 2011.

[28] W. S. Geislerand and M. S. Banks, "Visual performance" in *Handbook of Optics*, vol. III, M. Bass, Ed. McGraw-Hill, 1995.

[29] A. B. Watson and L. Kreslake, "Measurement of visual impairment scales for digital video", *Proc. of SPIE*, Human Vision and Electronic Imaging VI, 79, vol. 4299, pp. 79–89, 2001 (doi: 10.1117/12.429526).

[30] M. S. Salih *et al.*, "A proposed improvement model of MC-CDMA based FFT in AWGN channel", in *IEEE Int. Conf. on Electro/Inform. Technol.*, Chicago, IL, USA, 2007, pp. 517–520 (doi: 10.1109/EIT.2007.4374460).

[31] S. Nobilet, J.-F Helard, and D. Mottier, "Spreading sequences for uplink and downlink MC-CDMA systems: PAPR and MAI minimization", *European Trans. on Telecommun.*, vol. 13, pp. 465–474, 2002.

[32] K. Fazel and S. Kaiser, *Multi-Carrier and Spread Spectrum Systems*. New York: Wiley, 2008.

[33] A. Gupta, P. Nigam, and V. Chaurasia, "Removal of cyclic prefix in adaptive OFDM for dynamic spectrum access using DWT and WT", *Int. J. of Engin. Sci. & Res. Technol.*, vol. 3, no. 3, pp. 1187–1191, 2014.

Mohammed Sofiane Bendelhoum received a B.Sc. degree in Electronics Biomedical in 2004, and an M.Sc. in 2008, from Abou Bakr Belkaid University of Tlemcen, Algeria. He is currently pursuing his Ph.D. degree at the Djillali Liabes University of Sidi Bel Abbes, Algeria. His main research interests include the image processing, medical image compression, wavelets transform, wireless communication systems and signal processing.
E-mail: sbendelhoum13@gmail.com
Department of Technologie, Sciences Institute
University Center Nour Bachir of El-Bayadh
El-Bayadh, 32000, Algeria

Ali Djebbari received a B.Sc. in 1988, from University of Sciences and Technology of Oran (USTO, Algeria), an M.Sc. in 1991, from Sidi Bel Abbes University, and a Ph.D. in 1997, from USTO. Since 1991, he works at the Department of Telecommunications at the University of Sidi Bel Abbes and performs his research at the Telecommunications and Digital Signal Processing Laboratory. His research interests include signal processing for telecommunications, communication over multipath and fading channels, wireless networks, channel coding for communication networks. He has more than 60 publications in specialized journals with impact factors.
E-mail: adjebari2002@yahoo.fr
Telecommunications and Digital Signal Processing Laboratory
Djillali Liabes University
Sidi Bel Abbes, 22000, Algeria

Ismail Boukli-Hacene obtained a B.Sc. in Electronics, an M.Sc. in Electronic Biomedical, and a Ph.D. degree in Electronic Biomedical from University of Tlemcen, Algeria, His research interests are image processing, medical image compression, wavelets transform, wireless communication systems and optimal encoder. He has more than 20 publications in specialized journals with impact factors.
E-mail: ismaill80@yahoo.fr
Biomedical Laboratory
Abou Bekr Belkaid University
Tlemcen, 13000, Algeria

Abdelmalik Taleb-Ahmed received his Ph.D. in Electronics and Microwaves from the University of Lille1 in France in 1992. From 1992 to 2004, he was an Associate Professor at the University of Littoral, Calais. He is currently a Professor at the University of Valenciennes in the department GE2I, and does his research at the LAMIH FRE CNRS 3304 UVHC. His research interests include signal and image processing, image segmentation, prior knowledge integration in image analysis, partial differential equations and variational methods in image analysis, image compression, multimodal signal processing, medical image analysis, including multimodal image registration. He is an author of more than 200 publications.

E-mail: abdelmalik.tale-ahmed@univalenciennes.fr
Laboratoire LAMIH UMR CNRS 8530
Valenciennes et du Hainaut Cambrésis University
Valenciennes, 59313, France

Paper

QoS-based Joint User Selection and Scheduling for MU-MIMO WLANs

Dasari Srinivasa Rao[1] and Victor Dhason Berlin Hency[2]

[1] *Dept. of ECE, GMRIT, Rajam, Andhra Pradesh, India*
[2] *School of Electronics Engineering, VIT University, Chennai, Tamil Nadu, India*

https://doi.org/10.26636/jtit.2017.112217

Abstract— The shift in Multi-User Multiple Input Multiple Output (MU-MIMO) has gained attention due to its wide support in very high throughput Wireless Local Area Networks (WLANs) such as the 802.11ac. However, the full advantage of MU-MIMO can be utilized only with proper user selection and scheduling. Also, providing Quality of Service (QoS) support is a major challenge for these wireless networks. Generally, user scheduling is done with the acquisition of Channel State Information (CSI) from all the users. In MU-MIMO based WLANs, the number of CSI request increases with the number of users. This results in an increased CSI overhead and in degradation of the overall throughput. Most of the proposals in the literature have not addressed the contention in the CSI feedback clearly. Hence, in this paper a Joint User Selection and Scheduling (JUSS) scheme is discussed and its performance is evaluated in terms of throughput, delay, packet loss and fairness. In the performance comparison some wellknown Medium Access Control (MAC) protocols are considered. The proposed scheme not only enhances throughput, but also avoids contention during CSI feedback period.

Keywords— 802.11ac, CSI, delay, MMSE, throughput.

1. Introduction

Downlink (DL) MU-MIMO [1] in 802.11ac enables simultaneous transmissions from Access Points (APs) to the user stations (STAs) with the aid of parallel streams. The current WLAN supports up to eight spatial streams [2] at the transmitter side and allows multiple users to communicate simultaneously. Hence, there is a large increase in data rates compared to legacy 802.11 WLANs. MU-MIMO uses beamforming (BF) to direct information towards the desired users. MU-MIMO is the key revolutionary technology for all next generation WLANs [3] and cellular networks [4]. The performance of these networks depends mainly on proper user selection and scheduling. The scheduling is usually performed with the help of Channel State Information (CSI) feedback obtained from the selected users. However, obtaining CSI from all the users incurs a large overhead, and increases linearly with the channel sounding frequency and the number of user stations. Hence, there is an extensive requirement to reduce the CSI overhead.

To fully realize the benefit of MU-MIMO and guarantee the required QoS [5], it is important to acquire updated CSI from all the users. Hence, there exists a trade-off between efficiency of the scheduler and the CSI overhead. Generally, the Access Point (AP) limits the number of users based on CSI feedback. The best user CSI and suitable channels need to be obtained before the user is scheduled. In 802.11ac [6] user selection is performed randomly. However, selecting the best user group requires CSI feedback from all the users. In thr MU-MIMO scenario, the AP has to process multiple user requests simultaneously and has to schedule them in a parallel manner to maximize the overall throughput. Determining the possible user set is a problem, however, and it depends on the precoding method adopted at the AP. Although Zero-Forcing (ZF) receiver [1] is a simple technique that mitigates the multiuser inference, it has the disadvantage of noise enhancement. Hence, to avoid noise enhancement, the minimum mean square error beamforming (MMSE BF) can be used instead. Here, the MMSE-BF precoder is used to maximize throughput and minimize multi-user interference.

In this paper, a joint user selection and scheduling (JUSS) strategy is proposed for MU-MIMO networks to maximize downlink capacity and improve the overall throughput of the system. JUSS follows the same mechanism as 802.11ac [6], the AP involves user stations in the in polling process to procure CSI feedback. Here, scheduling is performed in parallel with active user identification and selection. Also, the contention during the CSI feedback phase will be reduced. Specifically, JUSS provides a practical solution for user scheduling. First, the AP polls the users randomly and obtains the CSI from one of the users. By using this information, the AP prepares a channel trace and broadcasts it to the users along with the polling frame. Upon receiving this frame, users can determine their individual channel gains. Then, the users respond with individual gains in the form of user response packets. By using this information, the AP prepares an ordered subset of users for obtaining CSI reports. The first user with the chosen subset is allowed to be scheduled for the MU-MIMO transmission. This process will be repeated until the AP receives all CSI reports. In the proposed JUSS, multiple stations are allowed to transmit the response packets at the AP to identify the user set to be scheduled. To avoid contention during CSI feedback, the AP identifies the user group to be scheduled based on response packets. In addition, the proposed scheme is extended onto some well-known fair scheduling

algorithms, such as round robin (RR) and proportional fair (PF), to attain fairness among the users.

The major contributions of this paper are summarized as follows. First, some well-known user selection schemes like semi-orthogonal user selection (SUS), random user selection (RUS), and random beamforming are reviewed. Second, the design of the proposed joint user selection and scheduling protocol is discussed. Then, the concepts to realize the protocol are briefly explained. Lastly, the performance of JUSS is evaluated using Matlab. The simulation results show that the JUSS scheme obtains better throughput, and improved fairness over the 802.11ac MAC protocol. Also, the delay and packet loss comparison are made with the existing protocols.

The rest of the paper is organized as follows. In Section 2, the background and related work concerning the problem are presented. Section 3 gives the details about the proposed scheme. The concepts discussed include CSI feedback, scheduling mechanism, and MU-MIMO transmission cycle of the protocol. Then, Section 4 evaluates the performance of the scheme with various parameters applied. Finally, Section 5 concludes the paper.

2. Background and Related Work

2.1. CSI Feedback Mechanism

The MU-MIMO technique is widely adopted in current 802.11 WLANs [3] because of its several dominant features. It offers high data rates [2] to the users at high signal-to-noise ratios. The AP is able to communicate simultaneously with multiple users providing spatial multiplexing. In addition, it offers considerable capacity gains [1] compared to the wireless communications system. The fundamental requirement for MU-MIMO based WLANs is to obtain CSI feedback before transmission. Generally, CSI acquisition [7] is performed by estimating a training sequence known by both the transmitter and the receiver In the uplink MU-MIMO system, the AP usually extracts the uplink CSI from the PHY preambles of the frames received. In the case of the downlink transmission, in turn, the acquisition of CSI cannot be performed directly.

CSI feedback schemes are classified depending on the computation of CSI at the two ends [8]. The first one involves implicit feedback, where the AP computes the CSI by estimating training sequences sent from the stations, while the other one – explicit feedback – means that stations calculate the CSI by estimating the training sequence sent from the AP, and then STAs feedback the calibrated CSI to the AP. Implicit feedback experiences less overhead compared to the explicit variety. However, in a practical wireless LAN system, channel variations and interference seen at the user stations is quite different from what is seen at the AP. Hence, the antennas at the AP are to be calibrated to eliminate the distortion if the implicit feedback scheme is used. On the other hand, the explicit feedback scheme provides higher CSI resolution at the cost of an increased overhead.

In literature, MAC control frames are usually extended to support CSI feedback, while an Explicit Compressed Feedback (ECFB) scheme [6] is implemented in IEEE 802.11ac to schedule and compress the amount of CSI feedback. The basic transmission cycle of a 802.11ac WLAN is shown in Fig. 1. It does not matter which CSI feedback scheme is applied. Both in the case of implicit and explicit schemes, the number of CSI requests in an MU-MIMO system increases with a growing number of users. This, in turn, affects the performance of the system.

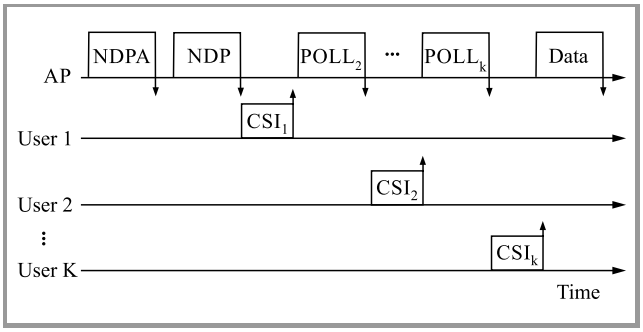

Fig. 1. 802.11ac MU-MIMO transmission cycle.

As evidenced by the discussion presented above, the major problem with CSI acquisition consists in a channel sounding interval and overhead due to feedback from one or more stations. Most publications have discussed the feedback overhead reduction using compression techniques [9]. However, compression leads to a loss of throughput in MU-MIMO transmissions. The quantization of feedback information is already adopted by Long Term Evolution (LTE) and MU-MIMO based Wi-Fi systems. Though extensive work [9], [10] has been done with regard to quantizing bits, there is still some room to optimize the feedback content. In addition to compression and quantization, sending the feedback information to the AP in an adaptive manner is another alternative enabling to reduce overload rates.

One such possible solution is provided in [11], where users with signal-to-noise ratios (SNRs) above a certain threshold value need to report the CSI to the AP. However, these techniques may not result in higher channel gains, as they ignore the relation between the user subchannels. Subchannels refer to the group of sub-carriers in orthogonal frequency division multiplexing (OFDM). In [12], the author presented a subchannel access approach, where the subchannels are allocated to users with the highest SNR to maximize the transmission rate and to keep the throughput of each user as high as possible. Since the scheduler works based on the SNR measurements received from the physical layer, most of the time is spent on the identification of the best user with the largest SNR. The size of the subchannel allocation table increases with the number of users and adds complexity to the scheduler.

Another challenge faced by MU-MIMO wireless LANs involves the frequency of CSI requests. The accuracy of the

system will be high if the feedback period is shorter than the coherence time. But, at the same time, this leads to a huge overhead at the AP. Hence, the frequency of CSI requests should be optimized. The authors in [13] and [14] have shown the impact of the channel sounding interval on the performance of the system. The key objective of the papers discussed above is to reduce the CSI feedback overhead with the increase in the number of users. It can be achieved with the help of adaptive CSI feedback mechanisms.

2.2. User Selection/Grouping Process

Nowadays, Wi-Fi systems are being deployed in crowded scenarios like airports, stadiums and buildings, where the number of concurrent users to be served is sufficiently large. In such situations, identifying, selecting and grouping the active users is a major task. In the MU-MIMO environment an AP with M antennas may be able to select a subset of M users out of all K users in a network. The problem of selecting users depends on the MAC scheduler and on the information provided by the PHY layer. User selection requires CSI estimates from all user stations. But even with the availability of full CSI knowledge at the AP, it is still difficult to determine the optimal user set that maximizes the transmission rate, specifically when the number of K users is large. Hence, an efficient user selection/grouping algorithm has to be devised to identify and select the user STAs to be co-scheduled.

In [15], the author discussed a Semi-orthogonal User Selection (SUS) algorithm, which achieves the maximum sum-rate with low complexity. The threshold is set to satisfy orthogonality and the users with the highest channel quality are selected.

Another popular algorithm, Orthogonality Probing based User Selection (OPUS) [16] has presented a user selection algorithm that is scalable and enables the user to evaluate its orthogonality with existing users. The author also discussed the distributed contention mechanism that singles out the best user to feedback its CSI. It outperforms conventional user selection schemes in terms of throughput and fairness.

In 802.11ac each user is selected randomly [6] with equal probability and requires M CSI feedbacks. Most publications have considered sounding all users before the transmission, whereas in [14] the author provided the flexibility to choose the subset of users that were sounded most recently. This exploited another direction and recently in [17] the author coupled the sounding, selecting and scheduling of users to maintain throughput and fairness performance.

In 18], the author proposed a novel orthogonality evaluation mechanism that enables each user to obtain its own CSI. This algorithm is known as signpost. Signpost also realizes a 2D prioritized contention mechanism to choose the best user efficiently by using both time and frequency domain resources. Signpost is a scalable user selection algorithm that is suited for uplink MU-MIMO WLAN transmissions. In this protocol, for each contention round, arbitrary probing directions are transmitted as channel hint to the user stations. The user stations check the orthogonality using these arbitrary directions and contend for the channel without sending the feedback to the access point. Hence, with zero CSI overhead the user competes for the resources.

Another recent uplink MU-MIMO WLAN protocol, known as optimal user selection (OUS) is proposed in [19]. OUS takes throughput and fairness into consideration and formulates the complex scheduling problem. OUS also considers the correlation among the users and provides throughput fairness solution to the user selection problem. It studies the impact of grouping the users based on throughput and fairness. To gain a better understanding of the above schemes, a summary of various state-of-art user selection and scheduling schemes with consideration of key design issues is provided in Table 1.

Table 1
Performance metrics of CSI feedback schemes

Scheme/ algorithm	Design issues considered			
	Through-put	Fair-ness	Over-head	Com-plex-ity
SUS [15]	✓	✓	High	Low
OPUS [16]	✓	✓	Low	High
802.11ac+ [17]	✓	✓	Low	High
Signpost [18]	✓	✓	Low	High
OUS [19]	✓	✓	–	Low

2.3. User Scheduling Schemes

When there are users contending for the same resource, their throughput may be improved by scheduling the users with fair channel quality. So, the increase in performance is obtained by exploring multi user diversity among the users. Due to this, the throughput of low rate users will be affected. System performance must be maximized without compromising fairness among the users. In user scheduling, the most challenging task is to determine the optimal user set because of the large search space. Hence, determining the optimal user set and scheduling them based on the QoS requirements improves the overall throughput.

With the introduction of MU-MIMO, most of the research regarding WLANs is now focusing on solving the issues affecting MU-MIMO based WLANs. In this way, multi user scheduling has gained attention. In [20], the author presented a novel MAC design with opportunistic MU-MIMO scheduling based on channel sounding feedback. Here, the packets being selected are based on transmission duration and the type of traffic. In [21], an efficient and heuristic MU-MIMO transmission method was proposed and compared to the beamforming based approach. The authors of [22] investigated the MU-MIMO transmission in WLAN by extending the MAC protocol with training functionalities to support efficient multi-user transmission. In [23]

the author verified the combined effect of packet aggregation with scheduling and has shown improved throughput performance at the cost of higher delay. In [24] the author presented a distributed multi-user scheduling scheme, which enhances the total throughput under many situations, compared with both contention-less and contention-based schemes. In all the publications discussed above, fairness among the user flows is not considered. Fairness issues were studied in [25], where a multi-user proportional fair scheduling scheme was proposed to schedule multi-user transmissions while providing a high degree of fairness.

3. JUSS Design

3.1. System Model

In this section, we consider the proposed MU-MIMO system with MMSE [26] as the precoder. The AP consists of M antennas and K-single antenna user stations that communicate with each other as shown in Fig. 2. The data symbols are transmitted through M antennas, and assume the MIMO channel satisfies the Rayleigh fading conditions: h_k are zero mean unit variance complex Gaussian channel vector. w_k is the beamforming weight corresponding to the linear precoder. MMSE precoder eliminates multi-user interference and provides better noise immunity compared to conventional ZFBF receiver.

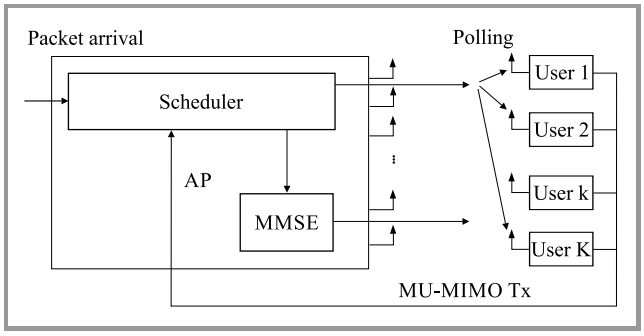

Fig. 2. MU-MIMO DL system model.

In MMSE precoding, since independent information is transmitted through different antennas the symbols are uncorrelated which means the precoding vector of one user is orthogonal to other users. Let S be the group of selected users, then the precoding matrix $\mathbf{W}(S)$ is given by:

$$\mathbf{W}(S) = P_d[P_d\mathbf{H}(S)^*\mathbf{H}(S) + \sigma_n^2\mathbf{I}]^{-1}\mathbf{H}(S)^*, \qquad (1)$$

where P_d represents the power of the transmitted data symbols, σ_n^2 is the noise variance, $\mathbf{H}(S)$ and $\mathbf{H}(S)^*$ the channel matrix, conjugate transpose matrix respectively.

3.2. Design Overview

In the proposed scheme, the CSI feedback and scheduling stages are considered jointly. The main idea is illustrated in Fig. 3. At the start of a transmission, the AP polls the first user (user 1 in the diagram) and it responds with its CSI. Then, the AP immediately puts user 1 in the scheduled user set S. User 1 is selected based on the AP queuing policy. Based on the first user CSI, the AP prepares a channel trace which is transmitted to all users. The channel trace typically contains the effective channel gain and the channel probe direction. This is termed as dual alignment metric. This information will be sent as the help of this information, the users can know contend for resources by transmitting the contention announcement (CA) frame. Then, the user with the best alignment metric will be announced as the winner in the contention round. In the example shown, user 2 is declared as the winner of that round. This step will be repeated until the AP successfully receives all M CSI reports or the feedback timeout is notified. In order to gain better understanding, the step-by-step procedure for the MU-MIMO downlink transmission is explained below:

1. Consider an AP with M antennas and K single antenna user stations. Assume that the channel is Gaussian with zero mean and unit variance (i.e. h_k satisfies i.i.d.).
2. The AP computes the precoding matrix ($\mathbf{W}(S) = \mathbf{H}(S)^*$) the pseudo-inverse, where S is the group of selected users. Now, AP starts polling the users.
3. At the beginning, AP polls the user 1 and it replies with its CSI, by making use of this information the AP prepares a channel trace.
4. Then, AP broadcasts the channel trace to the remaining users.
5. Upon receiving the channel trace from the AP, each user calculates individual gains and the user with the effective gain and the lowest correlation (i.e. better orthogonality) will be added to the set (may be user 2 in the example).
6. Repeat the steps 3–5 until the AP acquires all M CSI reports.

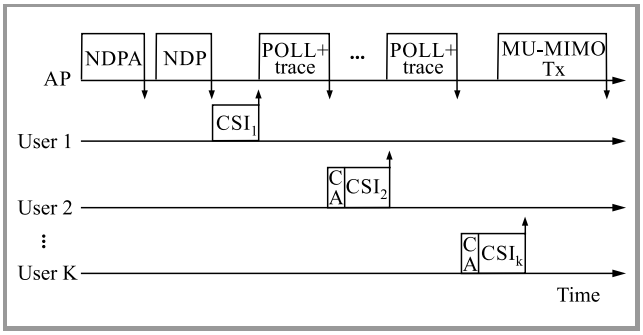

Fig. 3. MU-MIMO CSI feedback mechanism for JUSS.

The key idea behind the proposed scheme is that instead of entailing the AP to obtain the CSI from all the users, JUSS runs a distributed user selection and scheduling algorithm, where users participate in scheduling decisions to improve

the downlink capacity. As a result, the AP undergoes limited CSI feedback exchanges.

3.3. DL Feedback Mechanism

The user scheduling in the proposed scheme utilizes the channel gain obtained from the individual user stations during the feedback stage. Based on the information obtained from the user station, the AP prepares a channel trace that is basically composed of two preference metrics. This is termed as dual alignment metric. The first one gives the channel gain magnitude of the user station, and the other metric gives the probe direction of the desired user station. We shall look into the concept of these metrics in detail. For each user k, compute v_k the component of h_k orthogonal to the subspace spanned by $\{v_{(1)}, v_{(2)}, \ldots, v_{(|S|)}\}$:

$$v_k = h_k - \sum_{j=1}^{|S|} \frac{h_k v_{(j)}^*}{\|v_{(j)}\|^2} v_{(j)} = h_k \left(\mathbf{I} - \sum_{j=1}^{|S|} \frac{v_{(j)}^* v_{(j)}}{\|v_{(j)}\|^2} \right). \quad (2)$$

The effective channel gain of user k is denoted as v_k, it is seen that $\{v_{(j)}, 1 \leq j \leq |S|\}$ is a set of orthogonal vectors in $C^{1\times M}$, S is the set of users. The largest projection component is generally selected as the best user in this algorithm. Another important metric that is required to single out the best user is channel direction. After obtaining the channel state vector from the first user, the AP computes the possible directions to probe the signal space, which is orthogonal to selected users. The MU-MIMO MMSE system precodes the data symbols towards the desired users by suppressing the noise and interference. Let M be the number of transmit antennas on the access point and N be the number of client users. There will be a maximum of M rounds to feedback CSI. Each client user computes the alignment metric between the channel state vector h_k and the i-th probe vector p_i:

$$g_k = \frac{\|h_k p_i\|^2}{\|h_k\|^2}. \quad (3)$$

The orthogonality requirement for p_i is given by:

$$\left.\begin{array}{l} h_k \cdot p_i = 0,\ k \in S \\ P = \mathrm{Null}\mathbf{H}_{n-1} \end{array}\right\}, \quad (4)$$

where P is the probe vector and n is the number of contending rounds. Using the above metrics, the active users contend for the channel during the MU-MIMO transmission. The key advantage of this scheme is that the next best user will be decided among the users with the help of the dual alignment metric in a distributed manner. The objective of the scheme is to select the best user set with orthogonality and minimize the contention using an efficient feedback mechanism.

3.4. Contention Mechanism

Although the dual alignment metric computed in the above section is meant to single out a preferred set of users, contention will take place if more than one user contends for the channel at the same time. This is called feedback contention. Hence, to avoid feedback contention and to minimize the collision, an efficient feedback contention mechanism is required. For a MU-MIMO network with M antenna Access Point and N client users the contention mechanism works as follows. Recall each user can locally compute the dual alignment metric such that the user with strong orthogonality is selected. To avoid the contention among the users the AP selects the users by allowing them to transmit in slot durations. So, the users contend during different contention times. The users who might think their metric is best will contend for resource at different slots by transmitting CA packet. Upon receiving the request from the users, the AP singles out the best user and puts includes them in the selected user set. If any user hears contention announcement on a time slot, it stops transmitting and waits for a random amount of time for the next slot. For each round one user will be announced as contention winner and there will be maximum of M contention rounds. In JUSS, the number of contending users will be lower, because each user has to contend for the channel based on their individual alignment metric. So, each user actively participates in the scheduling process and, hence, the AP avoids the need to acquire CSI from all users. The fundamental difference between JUSS and 802.11ac is that the former uses a fixed CA duration for each CSI feedback and the AP modifies the polling frame by transmitting channel trace along with probing data. Also, to extend the fairness among the users, round robin and proportional fair scheduling algorithms are adopted in JUSS. These issues, along with throughput, delay and packet loss analysis, are discussed in the following section.

4. Performance Evaluation

For comparison purposes, we considered some well-known algorithms, namely 802.11ac, SUS with different α values [14] and the recent 802.11ac+ [16]. All simulations are performed using Matlab. The simulation parameters are shown in Table 2. The number of AP antennas is set to 4 and the number of serving users can equal up to 25. The OFDM PHY is specified to operate in the 5.2 GHz operating band. Each trace contains subcarrier (52 subcarriers) and transmission power ranges from 23 to 25 dBm. The total channel bandwidth is 160 MHz (two 80 MHz channels). The data traffic is generated in terms of packets from the AP and users. The proposed MAC is implemented by considering M transmit antennas (M = 2, 4) and K user stations (K varies up to 25).

4.1. Throughput Performance

To illustrate the throughput gain of user selection/scheduling over 802.11ac, Figure 4 shows the throughput of the proposed MMSE beamforming scheme for $M = 2, 4$ access point antennas. It is clearly seen that throughput of the downlink system degrades as the number of polls increases. It also affects the overall throughput.

Table 2
Simulation parameters

Parameters	Values
Carrier frequency	5.2 GHz
Channel bandwidth	160 MHz
Number of subcarriers	52
Modulation scheme	16 QAM
Payload length	512-1472 bytes
Tx power	23 dBm
MAC protocol	Extended CSMA/CA

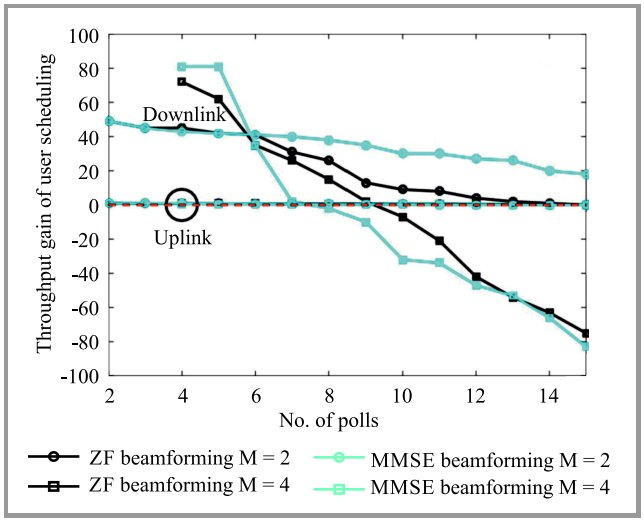

Fig. 4. Throughput gain of user selection/scheduling over 802.11ac. (See color pictures online at www.nit.eu/publications/journal-jtit)

The performance of any scheduling scheme is generally affected by an increase in the number of users. Although the increase in the number of scheduled users will benefit user diversity, the CSI feedback overhead may ultimately limit the gain. The impact of the user rate on the downlink throughput is shown in Fig. 5. The downlink and system throughputs of the three protocols are obtained. It is shown that the JUSS scheme performs well when compared to 802.11ac and SUS. It achieves a high throughput gain over 802.11ac and SUS respectively.

4.2. Throughput-Fairness Performance

The system throughput of 802.11ac+ and JUSS protocols is compared based on the round robin and proportional fairness criteria. It is shown in Fig. 6 that the system throughput of round robin and proportional fair JUSS is much higher than 802.11ac and better than RR-802.11ac+, PF-802.11ac+. JUSS and its fair approaches show similar performance, although they adopt different user selection criteria. Particularly, the system throughput of JUSS is far above that of 802.11ac. In Fig. 7. the downlink throughput attained by each user is plotted. From the plot, it is clear that the JUSS scheme achieves throughput that is higher than in 802.11ac and 802.11ac+.

The proposed scheme shows also that users with low SNRs achieve good throughput as well. This is possible because of MMSE precoding in the user selection process. The proposed scheme also achieves fair throughput over all users. It can be seen in the graph that 802.11ac does not show fair performance in the case of all users, hence the users with high SNRs will benefit the most. However, in the proposed scheme, the user grouping is done based on the alignment metric available at the AP.

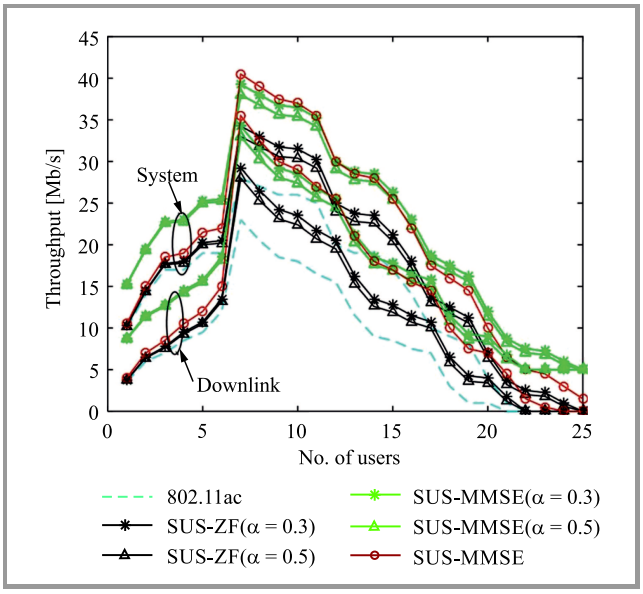

Fig. 5. System and downlink throughput with respect to the number of users.

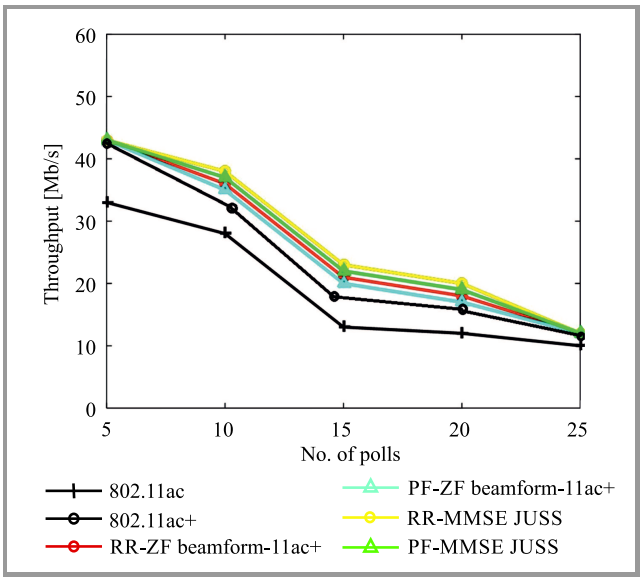

Fig. 6. System throughput comparison with respect to the number of polls.

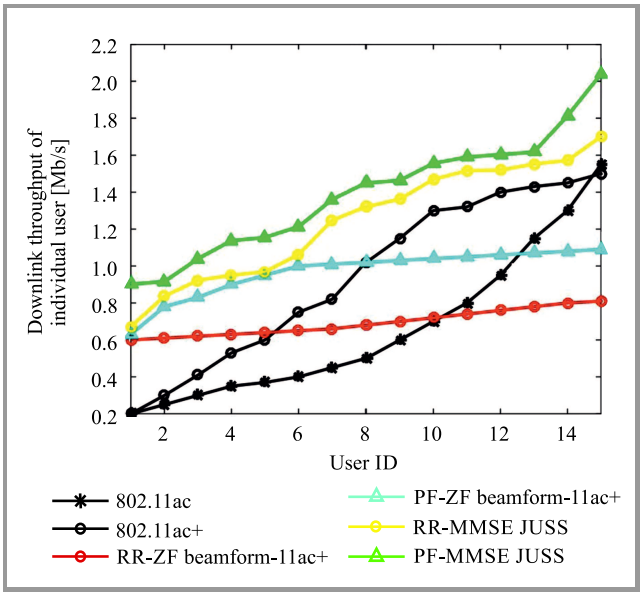

Fig. 7. Downlink throughout comparison of 802.11ac, 802.11ac+ and JUSS.

4.3. Delay Performance

Figure 8 gives the packet delay with respect to the number of interfering users. It is clearly shown that the percentage of packets lost for JUSS is low compared to ZF beamforming. The proposed feedback contention mechanism reduces packet loss due to collisions. In Fig. 9, the delay performance for ZF beamforming and MMSE beamforming scheduling schemes are provided. It indicates that the proposed scheme incurs less delay compared to the ZF beamforming method.
Some important conclusions about the results are given below.

- at $M = 4$ transmit antennas, our proposed scheduling scheme achieves better throughput gain over conventional ZF beamforming schedulers,
- the throughput comparison with 802.11ac, SUS and 802.11ac+ shows that, JUSS outperforms the existing schemes,
- it also provides fairness among the users along with throughput performance,
- in addition, it is also shown that the proposed scheme gives less delay and reduced packet loss compared to the ZF beamforming method.

5. Conclusions

In this work, an MMSE based MU-MIMO DL protocol for WLANs is proposed. The proposed JUSS scheme uses a dual alignment metric to select the preferred user set and schedule the MU-MIMO transmission collectively. The main advantage of this scheme consists in the fact that it eliminates the need to collect CSI from all users and reduces the feedback overhead. MMSE not only reduces

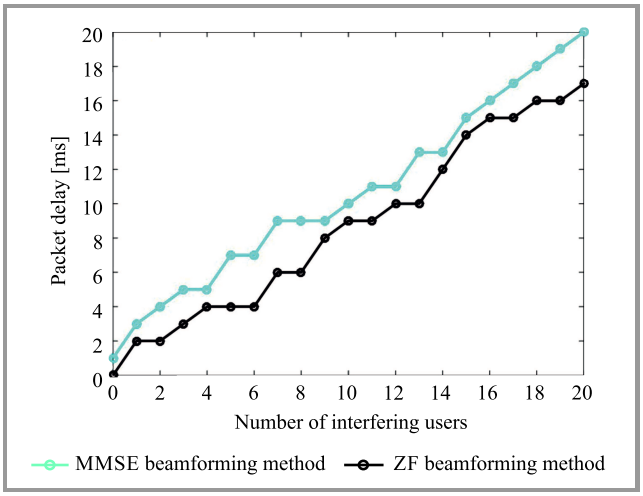

Fig. 8. Packet delay vs. number of interfering users

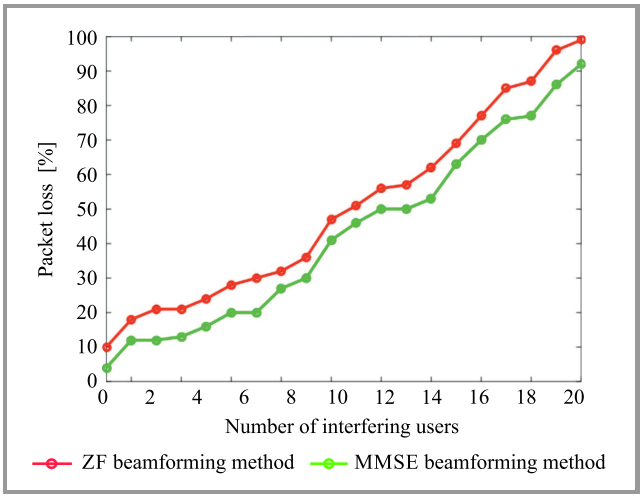

Fig. 9. Packet loss vs. number of interfering users.

interference, but also suppresses the noise present in the information. For performance comparison protocols like 802.11ac, SUS and recent 802.11ac+ are considered. To evaluate the performance of the proposed scheme throughput, fairness, delay and packet loss metrics are considered in this paper. The results obtained show that JUSS gives a better throughput performance compared to 802.11ac, SUS and 802.11ac+ protocols. Also, JUSS provides good results in terms of QoS performance measures like fairness, delay and packet loss.

References

[1] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*, 2 ed. Cambridge: Cambridge University Press, 2005.
[2] Cisco, "802.11ac: The Fifth Generation of Wi-Fi", in *Cisco White Paper*, 2012, p. 125.
[3] E. Parahia and R. Stacey, *Next generation wireless LANs: 802.11n and 802.11ac*, 2 ed. Cambridge: Cambridge University Press, 2013.
[4] Evolved Universal Terrestrial Radio Access (E-UTRA), Downlink Multiple Input Multiple Output (MIMO) enhancement for LTE Advanced (Release 11). 3GPP TR V11.0.0., 2015 [Online]. Available: http://www.3gpp.org

[5] D. Srinivasa Rao and V. Berlin Hency, "QoS based Radio Resource Management Techniques for Next Generation MU-MIMO WLANs: A Survey", *J. of Telecom. Electr. and Comp. Engin.*, vol. 8, no. 1, pp. 97–105, 2016.

[6] IEEE, IEEE Std P802.11ac: Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: enhancements for very high throughput for operation in bands below 6 GHz, 2013.

[7] R. Liao, B. Bellalta, M. Oliver, and Z. Niu, "MU-MIMO MAC protocols for wireless local area networks: A survey", arXiv: 1404.1622v2 (doi: 10.1109/COMST.2014.2377373).

[8] H. Lou, M. Ghosh, P. Xia, and R. Olesen, "A comparison of implicit and explicit channel feedback methods for MUMIMO WLAN systems", in *Proc. 2013 IEEE 24th Ann. Inter. Symp. on Person., Indoor, and Mob. Radio Commun. PIMRC*, London, UK, 2013, pp. 419–424 (doi: 10.1109/PIMRC.2013.6666172).

[9] X. Xie, X. Zhang, and K. Sundaresan, "Adaptive feedback compression for MIMO networks", in *Proc. of the 19th Ann. Int. Conf. on Mob. Comput. & Network. MobiCom'13*, Miami, FL, USA, 2013, pp. 453–464 (doi: 10.1145/2500423.2500465).

[10] D. Love, R. Heath, V. Lau, D. Gesbert, B. Rao, and M. Andrews, "An overview of limited feedback in wireless communication systems", *IEEE J. on Selec. Areas in Commun.*, vol. 26, no. 8, pp. 1341–1365, 2008.

[11] V. Hassel *et al.*, "A threshold-based channel state feedback algorithm for modern cellular systems", *IEEE Trans. on Wireless Commun.*, vol. 6, no. 7, pp. 2422–2426, 2007.

[12] A. B. Makhlouf and M. Hamdi, "Dynamic multiuser sub-channels allocation and real-ime aggregation model for IEEE 802.11 WLANs", *IEEE Trans. on Wireless Commun.*, vol. 13, no. 11, pp. 6015–6026 (doi: 10.1109/TWC.2014.2353611).

[13] G. Redieteab, L. Cariou, P. Christin, and J. F. Hélard, "PHY+MAC channel sounding interval analysis for IEEE 802.11ac MU-MIMO", in *Proc. 9th IEEE Int. Symp. on Wireless Commun. Syst. ISWCS 2012*, Paris, France, 2012, pp. 1054–1058 (doi: 10.1109/ISWCS.2012.6328529).

[14] O. Bejarano, E. Magistretti, O. Gurewitz, and E. W. Knightly, "MUTE: sounding inhibition for MU-MIMO WLANs", in *Proc. 11th Ann. IEEE Int. Conf. on Sensing, Commun., and Network. SECON*, Singapore, Singapore, 2014, pp. 135–143 (doi: 10.1109/SAHCN.2014.6990336).

[15] T. Yoo, N. Jindal, and A. Goldsmith, "Multi-antenna downlink channels with limited feedback and user selection", *IEEE J. on Selec. Areas. in Commun.*, vol. 25, no. 7, 2007, pp. 1478–1491 (doi: 10.1109/JSAC.2007.070920).

[16] X. Xie and X. Zhang, "Scalable user selection for MU-MIMO networks", in *Proc. IEEE Int. Conf. on Comp. Commun. INFOCOM 2014*, Toronto, ON, Canada, 2014, pp. 808–816 (doi: 10.1109/INFOCOM.2014.6848008).

[17] K. Lee and C. Kim, "User scheduling for MUMIMO transmission with active CSI feedback", *EURASIP J. on Wireless Commun. and Network.*, vol. 112, 2015 (doi: 10.1186/s13638-015-0331-4).

[18] A. Zhou, T. Wei, X. Zhang, M. Liu, and Z. Li, "Signpost: Scalable MU-MIMO signaling with zero CSI feedback", in *Proc. of the 16th ACM Int. Symp. on Mob. Ad Hoc Network. and Comput. MobiHoc'15*, Hangzhou, Zhejiang, China, 2015, pp. 327–336 (doi: 10.1145/2746285.2746286).

[19] Y. Zhou, A. Zhou, and M. Liu, "OUS: Optimal user selection in MU-MIMO WLANs", in *Proc. Int. Conf. on Comput., Network. and Comm. ICNC 2016*, Kauai, HI, USA, 2016 (doi: 10.1109/ICCNC.2016.7440681).

[20] S. Wu, W. Mao, and X. Wang, "Performance study on a CSMA/CA-based MAC protocol for multi-user MIMO wireless LANs", *IEEE Trans. on Wireless Commun.*, vol. 13, no. 6, pp. 3153–3166 (doi: 10.1109/TWC.2014.042314.131407).

[21] S. Huang, H. Yin, J. Wu, and V. C. M. Leung, "User selection for multiuser MIMO downlink with zero-forcing beamforming", *IEEE Trans. on Vehic. Technol.*, vol. 62, no. 7, pp. 3084–3097, 2013 (doi: 10.1109/TVT.2013.2244105).

[22] M. X. Gong, E. Perahia, R. Want, and S. Mao, "Training protocols for multi-user MIMO wireless LANs, in *Proc. IEEE 21st Int. Symp. on Person. Indoor and Mob. Rad. Commun. PIMRC 2010*, Istanbul, Turkey, 2010, pp. 1218–1223 (doi: 10.1109/PIMRC.2010.5672041).

[23] B. Bellalta, J. Barcelo, D. Staehle, A. Vinel, and M. Oliver, "On the performance of packet aggregation in IEEE 802.11ac MU-MIMO WLANs", *IEEE Commun. Lett.*, vol. 16, no. 10, pp. 1588–1591, 2012.

[24] S. Tang, "Distributed multiuser scheduling for improving throughput of wireless LAN", *IEEE Trans. on Wireless Commun.*, vol. 13, no. 5, pp. 2770–2781. 2014 (doi: 10.1109/TWC.2013.040214.130707).

[25] V. Valls and D. J. Leith, "Proportional fair MU-MIMO in 802.11 WLANs", *IEEE Wireless Commun. Lett.*, vol. 3, no. 2, pp. 221–224, 2014 (doi: 10.1109/WCL.2014.020314.130884).

[26] V. Jones and H. Sampath, "Emerging technologies for WLAN", *IEEE Commun. Mag.*, vol. 53, no. 3, pp. 141–49, 2015 (doi: 10.1109/MCOM.2015.7060496).

Dasari Srinivasa Rao received his Masters in Engineering (communication systems) from Thiagarajar College of Engineering, Madurai. Now, he is pursuing a Ph.D. from VIT University, Chennai. He has nearly eight years of experience in teaching. He is currently working as an Asst. Professor at Dept. of ECE, GMR Institute of Technology, Rajam, India. His research interests include wireless communications, next generation networks, and coding techniques.

E-mail: srinivasa.dasari@gmail.com
Department of ECE
GMR Institute of Technology
Rajam, Andhra Pradesh, India

Victor Berlin Hency received a B.E. in Electronics and Communication Engineering from Manonmaniam Sundarana University. She received a M.E. in Applied Electronics from Sathyabama University. She received a Ph.D. degree in Information and Communication Engineering from Anna University. She is an Associate Professor at the school of Electronics Engineering, VIT University Chennai Campus. Her research interests include wireless networks and IoT.

E-mail: berlinhency.victor@vit.ac.in
School of Electronics Engineering
VIT University
Chennai, Tamil Nadu, India

Paper

Synthesis and Failure Correction of Flattop and Cosecant Squared Beam Patterns in Linear Antenna Arrays

Hemant Patidar[1], Gautam Kumar Mahanti[1], and Ramalingam Muralidharan[2]

[1] *National Institute of Technology, Durgapur, India*
[2] *Caledonian College of Engineering, Oman*

https://doi.org/10.26636/jtit.2017.113417

Abstract—This paper deals with the synthesis of flattop and cosecant squared beam patterns using the firefly algorithm which is based on metaheuristics. This synthesis is followed by the correction of the radiation patterns when unfortunate malfunctioning of the individual elements in the array occurs. The necessary attention is given to the recovery process, with due emphasis on reduction of side lobe level, ripple and the reflection coefficient. Simulation in Matlab shows a successful employment of the firefly algorithm in producing voltage excitations of the good elements necessary for the recovered patterns. The performance of the firefly algorithm in failure correction is validated by duly comparing it with a standard benchmark.

Keywords—cosecant squared pattern, failure correction, firefly algorithm, flattop beam pattern, linear array, side lobe level.

1. Introduction

Past decades depict the importance of various shaped beam patterns, like the flattop pattern or the cosecant squared pattern, in a variety of applications relating to telecommunications. Many methods are being developed for the generation of these kinds of patterns and, in addition, importance is also attached to the various parameters that have evolved out of the radiation patterns. The most significant parameter is the side lobe level (SLL) referring to the ratio of the peak of the main lobe to the peak of the side lobe in the antenna array radiation pattern [1]. It is also the most important factor in the optimization of antenna array patterns. The higher SLL level, the bigger deterioration in the pattern that will cause failures in one or more of the antenna elements. These failures are represented by the zero excitations elements. These failures affect not only the SLL, but also many other parameters, like beam width or reflection coefficient. The reflection coefficient (RC) describes how much of an electromagnetic wave is reflected by an impedance discontinuity or a mismatch in the communication medium. The RC value equals the ratio of the amplitude of the reflected wave to the incident wave. Failures disturb the signal and force a correction to the patterns, irrespective of the beams shape.

The best correction method is to replace hardware, if the antenna array is easy to reach. This leads to a further increase in cost and difficulty. On the other hand, there are solutions relying on digital beam forming, using evolutionary algorithms, where – by modifying the excitations of good (unfailed) elements – the failures can be corrected in such a way that the recovered radiation pattern resembles that of the original pattern in terms of radiation pattern parameters as well as other parameters.

Literature review reveals many algorithms used in the past that proved to be helpful. Depending on their concepts and characteristics, each is proving to be better than the other.

Rodriguez and Ares [2] recovered the original pattern by changing the amplitude and phase distributions separately for both the sum and shaped beam patterns, successfully using the simulated annealing technique. Yeo and Lu in [3] presented numerous examples of single and multiple element failure corrections using beam forming weights, representing them directly by a vector of complex numbers. Lozano *et al*. [4] synthesized the patterns by finding the optimal configuration of the array factor roots, using the simulated annealing technique in both sum and flat-topped beam patterns. Rodriguez *et al*. applied the genetic algorithm [5] in restoring the array pattern with three failures, by changing the amplitude distribution of less than a third of the unfailed array elements. Yeo and Lu in [6] utilized particle swarm optimization for the radiation patterns correction during the failure of a few elements in the array.

Active impedance [7] affects the patterns and its corresponding parameters in terms of mutual impedance and self-impedance. Synthesis of antenna arrays with a shade of mutual coupling have been reported in [8]–[10]. To quote a few works in the past concerned with shaped beam patterns, Zhou *et al*. in [11] realized a 10-element flattop beam shaped linear antenna array, which gave a well-proportioned power distribution in the required zone using the genetic algorithm. Lei *et al*. [12]] presented a cosecant squared beam pattern in a wide band integrated linear printed array using a modified least square method by matrix inversion, which can be directly used in surveillance radar applications.

This paper proposes a synthesis of the flat top beam pattern and the cosecant squared beam pattern, using amplitude and

phase excitations, with a two-bit phase shifter for the flat top beam and continuous phase for the cosecant squared beam.

One of the recent evolutionary firefly algorithms (FA) [13]–[15] has been used here to generate excitations, due to its successful past usage in the field of antenna arrays [15]. The novelty of the approach presented in this paper consists in the fact that the failure correction process is performed for different shaped beam patterns, and active impedance is duly taken into consideration during the processing of both far-field patterns. The proposed algorithm is compared with the well-known Particle Swarm Optimization (PSO) algorithm [16]–[17].

2. Problem Formulation

The far field pattern in x-y plane for the linear antenna array constructed of parallel half wavelength (λ) dipoles equally spaced at a distance $d = 0.48\,\lambda$ is given by:

$$FFP(\phi) = \left[\sum_{n=1}^{N} C_n e^{j(n-1)kd\cos\phi}\right] \cdot ElementP(\phi), \qquad (1)$$

where n represents the element number, N is the total number of array elements, ϕ is the angle measured from the x axis to the far-field point, k is the wave number and C_n is the excitation current of n-th element.

The element pattern of each dipole $ElementP$ is assumed to be omnidirectional within the plane taken into consideration, and is substituted by a value equal to unity. The expression for the self and mutual impedances are taken from [7]. A sum pattern is produced in the broadside direction using the current excitations obtained with the help of the mutual coupling impedance matrix.

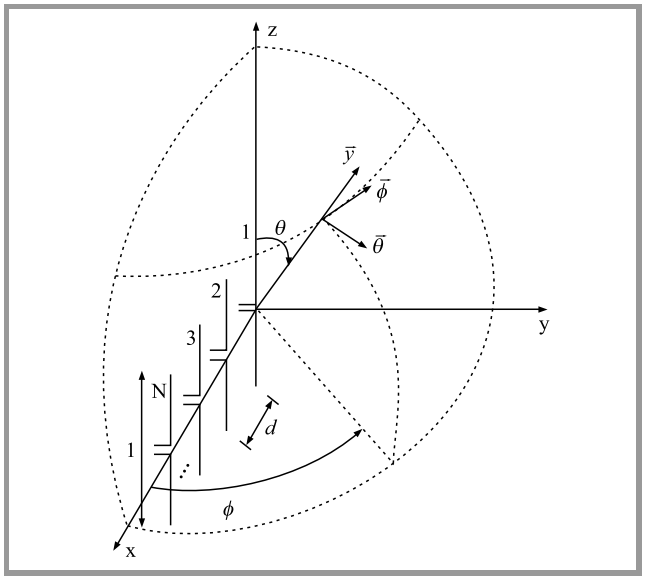

Fig. 1. Geometry of a linear antenna array with elements in the x axis and placed parallel to z axis.

Considering the characteristic impedance Z_0 of the feed network to be 50 Ω, the reflection coefficient RC (in dB) at the input of the a-th dipole antenna is given by:

$$\mathrm{RC} = 20\log\frac{|Z_a^A| - Z_0}{|Z_a^A| + z_0}, \qquad (2)$$

where Z_a^A is the active impedance of the elements. From among all the elements in the array, the maximum of the RC value is obtained. The active impedance of any failed element is considered to be zero. The problem is to obtain a new set of voltage excitations of the remaining non-failed elements such that the corrected pattern resembles similar to that of the original pattern in terms of the expected radiation pattern parameters, i.e. SLL, ripple in the flat top portion of the pattern, and RC. Similarly, for the restoration of the cosecant squared pattern, a new set of voltage and phase excitations for the remaining good elements is generated in such a way that the corrected restored pattern using the newly generated excitations resembles the original pattern in terms of radiation pattern parameters.

The fitness function for generation of the original and the corrected flattop beam pattern is:

$$Fitness = wt_1F_1^2 + wt_2F_2^2 + wt_3F_3^2, \qquad (3)$$

where:

$$F_1 = \begin{cases} \mathrm{SLL_o} - \mathrm{SLL_d} & \text{if } \mathrm{SLL_o} > \mathrm{SLL_d} \\ 0 & \text{if } \mathrm{SLL_o} \le \mathrm{SLL_d} \end{cases},$$

$$F_2 = \begin{cases} |\mathrm{ripple^o_{max}}| - \mathrm{ripple^d_{max}} & \text{if } |\mathrm{ripple^o_{max}}| > \mathrm{ripple^d_{max}} \\ 0 & \text{if } |\mathrm{ripple^o_{max}}| \le \mathrm{ripple^d_{max}} \end{cases},$$

$$F_3 = \begin{cases} \mathrm{RC^o_{max}} & \text{if } \mathrm{RC^o_{max}} > \mathrm{RC^d_{max}} \\ 0 & \text{if } \mathrm{RC^o_{max}} \le RC^{\mathrm{d}}_{\mathrm{max}} \end{cases}.$$

In Eq. (3), ripple refers to the difference between the maximum and the minimum value of the far-field pattern in the desired flat top sector pattern. The "o" prefix stands for the obtained values and "d" refers to the desired values with the "max" suffix referring to the maximum obtained value. The fitness function for generation of the original and the corrected cosecant squared pattern is:

$$Fitness = wt_1F_1^2 + wt_2F_2^2 + wt_3F_3^2 + wt_4F_4^2, \qquad (4)$$

where:

$$F_1 = \begin{cases} \mathrm{SLL_o} - \mathrm{SLL_d} & \text{if } \mathrm{SLL_o} > \mathrm{SLL_d} \\ 0 & \text{if } \mathrm{SLL_o} \le \mathrm{SLL_d} \end{cases},$$

$$F_2 = \begin{cases} |\mathrm{rpl^o_{max}}| - \mathrm{rpl^d_{max}} & \text{if } |\mathrm{rpl^o_{max}}| > \mathrm{rpl^d_{max}} \\ 0 & \text{if } |\mathrm{rpl^o_{max}}| \le \mathrm{rpl^d_{max}} \end{cases},$$

$$F_3 = \begin{cases} \mathrm{RC^o_{max}} - \mathrm{RC^d_{max}} & \text{if } \mathrm{RC^o_{max}} > \mathrm{RC^d_{max}} \\ 0 & \text{if } \mathrm{RC^o_{max}} \le \mathrm{RC^d_{max}} \end{cases},$$

$$F_4 = \begin{cases} |\mathrm{rpc^o_{max}}| - \mathrm{rpc^d_{max}} & \text{if } |\mathrm{rpc^o_{max}}| > \mathrm{rpc^d_{max}} \\ 0 & \text{if } |\mathrm{rpc^o_{max}}| \le \mathrm{rpc^d_{max}} \end{cases}.$$

In Eq. (4), rpl and rpc refers to the difference between the maximum and the minimum value of the far-field pattern in the desired flat portion (90 to 100°) and the cosecant squared beam portion (100 to 145°) in the pattern.

3. Firefly Algorithm

The firefly algorithm [13]–[14] is based on the social behavior of fireflies or lightning bugs living in tropical and moderate regions and known for their conspicuous usage of bioluminescence during twilight to attract their corresponding mates or prey. Only three rules related with fireflies are used here. All the fireflies are attracted to each other irrespective of their gender. The degree of attractiveness is proportional to their brightness. The darker ones will move towards the brighter ones. However, the apparent brightness decreases as their mutual distance increases. If no fireflies are available whose brightness would be higher than a given value, the other fireflies will start moving randomly.
The steps involved in this algorithm are given as below.

Initialization. The algorithm starts with the initialization of the location of M fireflies in D dimensional search space, with the searching boundary given by:

$$x_{md}(0) = \text{rand}_{md}(0,\ 1)(x_{md}^{upper} - x_{md}^{lower}) + x_{md}^{lower}, \quad (5)$$

where $m = 1, 2, \ldots, M$ and $d = 1, 2, \ldots, D$. The x_{md}^{upper} and x_{md}^{lower} are the corresponding upper limit and lower limits of the d-th variable in the overall population with rand being the uniformly distributed random value.

Light intensity. The next step is to determine the light intensity (brightness) of each firefly at the current iteration using the fitness function for optimization. This intensity is inversely proportionate to the fitness function for every firefly.

Updating of the locations. Each firefly moves towards a brighter firefly and updates its position for the next iteration. The attractiveness factor β is relative and is also judged by other fireflies. In other words, it will vary with the distance between two fireflies. In addition, light intensity decreases with the distance from its source, and light is also absorbed by the medium. Thus, it is obvious to say that the light intensity varies according to the inverse square law. For a given medium with a constant light absorption coefficient, the light intensity I varies with the distance r, $I = I_0 e^{-\gamma r}$, where I_0 is the original light intensity level. To use the combined influence of absorption as well as the inverse square law, the light intensity is approximated using the Gaussian form as $I = I_0 e^{-\gamma r^2}$.
As the attractiveness of a firefly is proportionate to the light intensity observed by other fireflies, the firefly attractiveness β is $\beta = \beta_0 e^{-\gamma r^2}$, where β_0 is the attractiveness at $r = 0$. The attractiveness between the fireflies is:

$$x_m = x_m + \beta_0 e^{-\gamma r_{mp}^2}(x_m - x_p) + \alpha \varepsilon_m, \quad (6)$$

where γ is the light absorption coefficient whose value depends on the characteristic features of the medium but is fixed in a given medium, α is the randomization parameter varying between 0 and 1, ε_m is a vector of random numbers drawn from an uniform of Gaussian distribution, and the attractiveness between the two fireflies m and p is given by the product of β_0 and $e^{-\gamma r_{mp}^2}$ terms.
The Cartesian distance between two m and p fireflies is given at x_m and x_p by:

$$r_{mp} = \| x_m - x_p \| = \sqrt{\sum_{d=1}^{D} (x_{m,d} - x_{p,d})^2}. \quad (7)$$

The algorithm states that the brightest firefly cannot move in any direction, whereas the remaining fireflies modify their location accordingly at current generation. This process is used to obtain the overall best result at the end of the final specified number of iterations. The brightness of the brightest firefly is regarded as the final best fitness value.
The firefly algorithm parameters are as follows: randomness = 1, minimum value of $\beta = 0.2$ and absorption coefficient = 1. The settings for PSO are: inertial weight = time-varying inertia weight changing randomly between U (0.4, 0.9) with iterations, time-varying acceleration coefficients c_1 (t) and c_2 (t) to be 1.495 over the full range of the search.

4. Simulated Results

4.1. Flat Top Beam Pattern

A linear antenna array containing 24 elements with spacing of 0.48 λ along the x axis and situated parallel to z axis is taken into account for simulation purposes. The voltage excitation ranges from 0 to 1 with the binary phase shifter generating only 0 and 180° phases. Moreover, the choice of the initial population is random and the population size is 48. The original pattern is obtained using FA and is scheduled for 10 runs with 1000 iterations. For the generation of the original pattern, the values of wt_1, wt_2 and wt_3 are kept at 5, 15 and 1 respectively. The corrected pattern is run 20 times with 1500 iterations each. And for the corrected pattern, the values they are kept at 1, 3 and 3. The choice of these values denotes the importance of the concerned designed parameter associated therewith. Seven elements out of 24 (29%) are taken as failed: V(3) = 0, V(6) = 0, V(11) = 0, V(14) = 0, V(17) = 0, V(20) = 0, and V(22) = 0. The choice of defective elements is random.

Table 1 shows that the original pattern without any failures is generated by using the firefly algorithm as per the desired ones. A random choice of 7 element failures deteriorates the pattern resulting in a very poor SLL, as well as ripples and RC. The two algorithms are made to engage in generating the necessary amplitude and the binary phases of the remaining unfailed elements and the results obtained through the algorithms shows that the FA has flared far

Table 1
Desired and obtained values of parameters for the flattop beam pattern

Parameters	Desired values	Patterns			
		Original FA	Damaged	Corrected using FA	Corrected using PSO
SLL [dB]	–25	–25.278	–5.5891	–24.9873	–23.8248
Maximum ripple (70 to 110°) [dB]	±0.5	0.40283	4.3745	0.54535	1.529
Maximum reflection coefficient	0.25	0.24786	4.3745	0.25793	0.53846
Mean	–	–	–	32.12	175.83
Standard deviation	–	–	–	11.82	62.72
Computation time [s]	–	5661	–	16854	17032

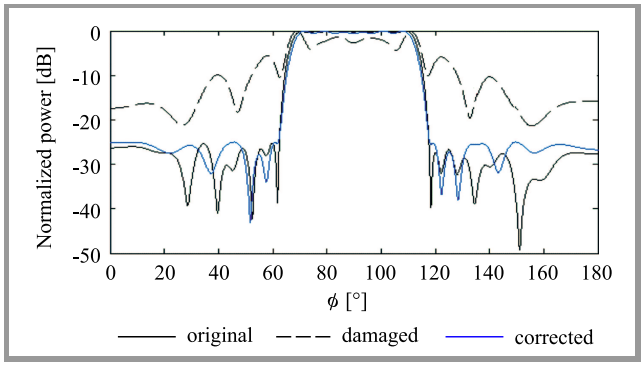

Fig. 2. Original, damaged and corrected normalized power pattern versus ϕ for flattop pattern using FA.

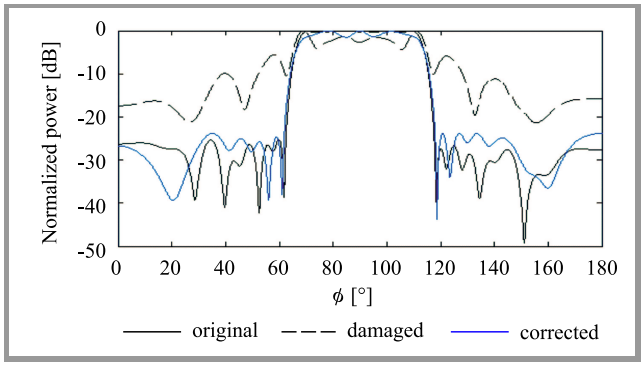

Fig. 3. Original, damaged and corrected normalized power pattern versus ϕ for flattop pattern using PSO.

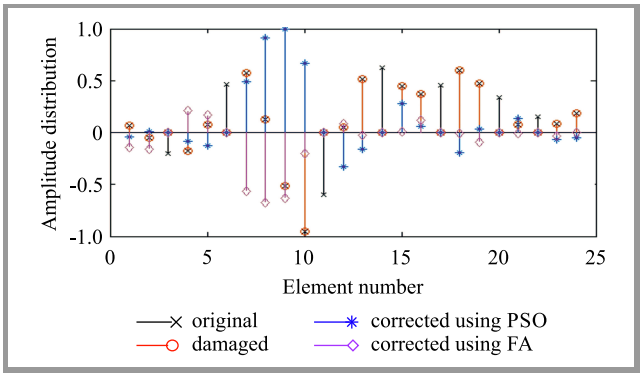

Fig. 4. Amplitude excitations vs. number of elements.

better than the PSO algorithm. Even though FA could not fully meet the needs, it resulted in producing parameters that are very close to the desired values. A magnitude deficit of 0.0127 dB in SLL, 0.0454 dB in maximum ripple and 0.00793 in RC is definitely a very low value and this shows a good agreement between the desired and corrected values obtained by using FA. A comparison with PSO also reveals that FA has flared far better than PSO in terms of all parameter values. This is sufficient enough to prove that the former algorithm is better than PSO. The patterns are shown in graphical form in Figs. 2 and 3. Figure 4 shows the corresponding amplitude distributions. A negative value in amplitude distribution represents a 180° phase.

Table 2
Desired and obtained values of parameters for the cosecant squared beam pattern

Parameters	Desired values	Patterns			
		Original FA	Damaged	Corrected using FA	Corrected using PSO
SLL [dB]	–20	–21.2463	–6.3317	–19.691	–19.3237
Maximum ripple (90 to 100°) [dB]	±0.5	0.2024	3.0963	0.53363	1.6616
Maximum reflection coefficient	0.25	0.25698	0.34199	0.27632	0.36964
Maximum ripple (100 to 145°) [dB]	1	0.99394	5.1404	1.3294	1.8336
Mean	–	–	–	81.64	129.42
Standard deviation	–	–	–	68.47	79.83
Computation time [s]	–	4200	–	25578	26428

4.2. Cosecant Squared Beam Pattern

A linear antenna array of 18 elements along the x axis and situated parallel to the z axis is taken into account for simulation purposes. The element spacing is 0.48 λ. The original pattern is run 10 times with 1000 iterations each and the population size is 36. The corrected pattern is run 20 times with 1500 iterations each and the population size is 72. For the generation of both the patterns using FA, the values of wt_1, wt_2, wt_3 and wt_4 are kept uniform. Five elements out of 18 are taken as failed: V(3) = 0, V(5) = 0, V(8) = 0, V(13) = 0, V(17) = 0. The choice of the defective element is random. The voltage excitation ranges from 0 to 1 and phase ranges from –180 to 180°.

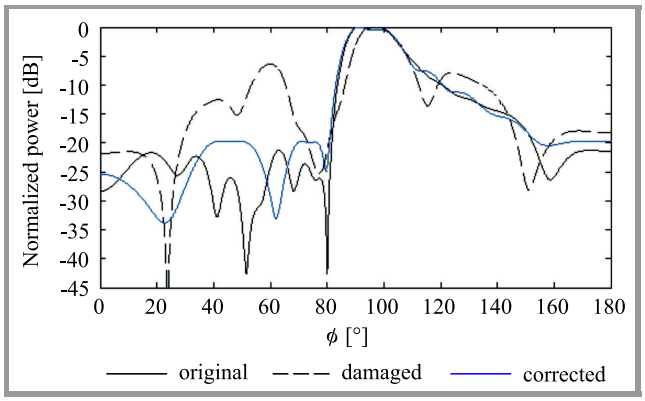

Fig. 5. Original, damaged and corrected normalized power pattern versus ϕ for cosecant squared pattern using FA.

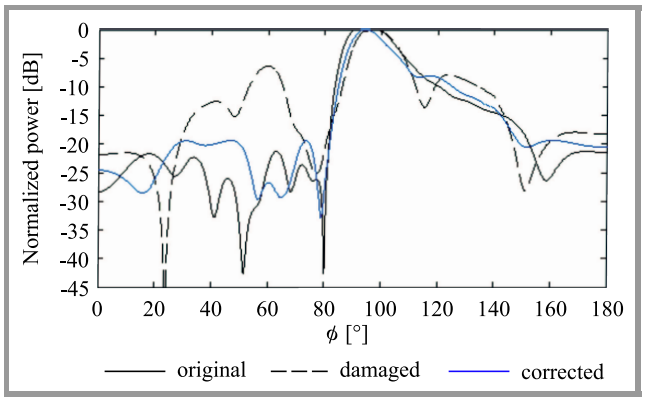

Fig. 6. Original, damaged and corrected normalized power pattern versus ϕ for cosecant squared pattern using PSO.

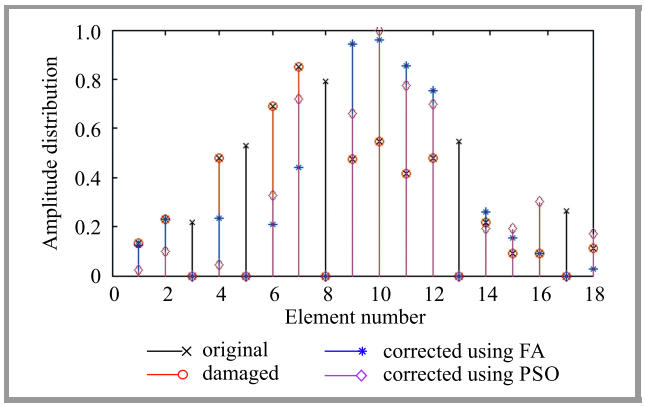

Fig. 7. Amplitude distribution vs. number of elements.

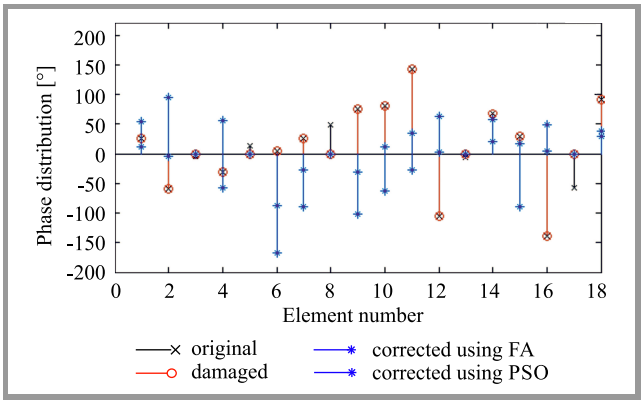

Fig. 8. Phase distribution vs. number of elements.

The original pattern is successfully generated by the FA algorithm. Both algorithms are made to engage in the recovery process and Table 2 shows that FA has once again flared better than PSO. A magnitude deficit of 0.309 dB in SLL, 0.03363 dB in ripple (flat), 0.02632 in RC and 0.3294 dB in ripple (shaped beam) is quite an acceptable one and this also shows that there is a good agreement with the desired and corrected values in the event of a failure. Table 2 also shows that FA has flared far better than PSO in the recovery process. The patterns are shown in Figs. 5 and 6. Figures 7 and 8 show amplitude and phase distributions.

5. Conclusion

The simulation performed with the use of Matlab proves successful usage of FA in most of the cases, and it edges better over the benchmark algorithm in restoring the radiation pattern and its associated antenna parameters. This work can be extended in the future to failure correction of shaped beam patterns in other geometries of antenna arrays.

References

[1] C. A. Balanis, *Antenna Theory: Analysis and Design*, 2nd ed. Singapore: Wiley (Asia), 2003.

[2] J. A. Rodriguez and F. Ares, "Optimization of the performance of arrays with failed elements using the simulated annealing technique", *J. of Electromag. Waves and Appl.*, vol. 12, no. 12, pp. 1625–1637, 1998.

[3] B. K. Yeo and Y. Lu, "Array failure correction with a genetic algorithm", *IEEE Trans. on Antenn. and Propagat.*, vol. 47, no. 5, pp. 823–828, 1999.

[4] M. V. Lozano, J. A. Rodríguez, and F. Ares, "Recalculating linear array antennas to compensate for failed elements while maintaining fixed nulls", *J. of Electromag. Waves and Appl.*, vol. 13, no. 3, pp. 397–412, 1999.

[5] J. A. Rodriquez, F. Ares, E. Moreno, and G. Franceschetti, "Genetic algorithm procedure for linear array failure correction", *Electron. Lett.*, vol. 36, no. 3, pp. 196–198, 2000.

[6] B. K. Yeo and Y. Lu, "Fast array failure correction using improved particle swarm Optimization", in *Proc. Asia Pacific Microw. Conf. APMC 2009*, Singapore, 2009, pp. 1537–1540.

[7] R. S. Elliott, *Antenna Theory and Design*. New York, USA: Prentice Hall, 1981.

[8] M. Thevenot *et al.*, "Synthesis of antenna arrays and parasitic antenna arrays with mutual couplings", *Int. J. of Antenn. and Propagat.*, vol. 2012, pp. 1–22, 2012.

[9] J. A. Rodriguez, F. Ares, E. Moreno, and G. Franceschetti, "Design of efficient, easily feed matched array antennas by joint optimization of excitations and element geometry: a pencil beam example", *Electron. Lett.*, vol. 34, no. 13, pp. 1280–1282, 1998.

[10] J. A. Rodriguez, F. Ares, and E. Moreno, "Feeding in-phase dipole arrays: a tutorial and a MATLAB program", *IEEE Antenn. and Propagat. Mag.*, vol. 47, no. 5, pp. 169–173, 2005.

[11] H. J. Zhou, Y. H. Huang, B. H. Sun, and Q. Z. Liu, "Design and realization of a flat-top shaped-beam antenna array", *Progress in Electromag. Res. Lett.*, vol. 5, pp. 159–166, 2008.

[12] J. Lei, G. Fu, L. Yang, and D.-M. Fu, "Wide band linear printed antenna array with low sidelobe cosecant square-shaped beam pattern", *Progress in Electromag. Res. C*, vol. 15, pp. 233–241, 2010.

[13] X. S. Yang, "Firefly algorithm, stochastic test functions and design optimization", *Int. J. of Bio-Inspired Comput.*, vol. 2, no. 2, pp. 78–84, 2010.

[14] X. S. Yang, "Firefly Algorithm, Lévy Flights and Global Optimization", in *Research and Development in Intelligent Systems XXVI*, M. Bramer, R. Ellis, and M. Petridis, Eds. London: Springer 2010, pp. 209–218.

[15] R. Muralidharan, A. Vallavaraj, and G. K. Mahanti, "Firefly Algorithm for failure correction of linear array of dipole antennas in presence of ground plane with mutual coupling effects", *Applied Computat. Electromag. Soc. J.*, vol. 30, no. 10, pp. 1122–1128, 2015.

[16] J. Kennedy and R. Eberhart, "Particle swarm optimization", in *Proc. of the IEEE Int. Conf. on Neural Networks*, Perth, Australia, 1995, pp. 1942–1948.

[17] D. W. Boeringer and D. H. Werner, "Particle swarm optimization versus genetic algorithms for phased array synthesis", *IEEE Trans. Antenn. Propagat.*, vol. 52, no. 3, pp. 771–779, 2004.

Hemant Patidar obtained his B.Eng. degree in Electronics & Communication Engineering from Swami Vivekananda College of Engineering, Indore, India in 2012. Since 2013 he has been working as a Ph.D. research scholar in the department of ECE, National Institute of Technology, Durgapur, India. His research interests include array antenna synthesis, evolutionary algorithms, electromagnetics.

E-mail: hemantpatidar08@gmail.com
Department of Electronics and Communication Engineering
National Institute of Technology
Durgapur, India

Gautam Kumar Mahanti obtained his B.Eng. in Electronics and Communication Engineering in 1988 from NIT, Durgapur, India, M.Sc. in Electronics Systems and Communication in 1994 from NIT, Rourkela, India, and Ph.D. (Eng.) from IIT, Kharagpur, India. He has more than 20 years of teaching and research experience. Presently he is working as a Professor at the Department of Electronics and Communication Engineering, National Institute of Technology, Durgapur, India. He is a senior member of IEEE, USA. His research area includes array antenna synthesis, evolutionary algorithms and electromagnetics.

E-mail: gautammahanti@yahoo.com
Department of Electronics and Communication Engineering
National Institute of Technology
Durgapur, India

Ramalingam Muralidharan obtained his B.Eng. in Electronics and Communication Engineering in 1995 and M.Sc. (Eng.) in Applied Electronics in 2004, and Ph.D. in 2014 in the field of antenna arrays. He has more than 15 years of teaching experience in teaching subjects related to electronics and communication engineering. Presently, he is working as a Senior Lecturer, at the Department of Electrical and Computer Engineering, Caledonian College of Engineering, Oman. His research areas include array antenna failure correction and evolutionary algorithms.

E-mail: muralidharan@caledonian.edu.om
Department of Electrical and Computer Engineering
Caledonian College of Engineering
Sultanate of Oman

Paper

The Alive-in-Range Medium Access Control Protocol to Optimize Queue Performance in Underwater Wireless Sensor Networks

Vikas Raina[1], Manish Kumar Jha[2], Partha Pratim Bhattacharya[1]

[1] *Mody University of Science and Technology, Lakshmangarh, India*
[2] *ABES Engineering College, Ghaziabad, India*

https://doi.org/10.26636/jtit.2017.112317

Abstract—Time synchronization between sensor nodes to reduce the end-to-end delay for critical and real time data monitoring can be achieved by cautiously monitoring the mobility of the mobile sink node in underwater wireless sensor networks. The Alive-in-Range Medium Access Control (AR-MAC) protocol monitors the delay of sensitive, critical and real-time data. The idea evolves as it involves reduction in duty cycle, precise time scheduling of active/sleep cycles of the sensors, monitoring the mobility of the sink node with the selection of appropriate queues and schedulers. The model for the path loss due to attenuation of electromagnetic wave propagation in the sea water is explained. The three-path reflection model evaluating reflection loss from the air-water and water-sand interfaces as a function of distance between sensors and water depth is introduced. The algorithms for effective path determination and optimum throughput path determination are elaborated. The results verify that implementation of the Alive-in-Range MAC protocol has reduced the total number of packets dropped, the average queue length, the longest time in queue, the peak queue length and the average time in queue significantly, making it relevant for critical and real-time data monitoring.

Keywords—Alive-in-Range MAC, effective path determination, mobile sink, optimum throughput path determination, underwater wireless sensor network.

1. Introduction

Unmanned underwater explorations have recently gained in popularity and are relied upon in monitoring aquatic environments (comprising ponds, lakes, rivers, oceans and reservoirs, etc.) for commercial utilization, scientific investigation and disaster prevention. An exemplary system for such a type of extensive monitoring is a network comprising hundreds of tiny autonomous, wireless sensors and vehicles distributed underwater, communicating with each other and performing collaborative tasks. They are known as underwater wireless sensor networks (UWSNs).

They augment our ability to observe and predict the aqueous environments by enabling many applications, such as tactical surveillance, assisted navigation, disaster prevention, oceanographic data collection, pollution monitoring and offshore exploration [1]. Although in the specific applications involving critical and real-time data monitoring performed for limited time periods, such as rescue and military operations, parameters like throughput, end-to-end delay, jitter and packets dropped are highly influential. The battery power available in the sensor nodes already deployed cannot be increased, but the battery power of sensor nodes intentionally deployed for the operation, as well as the amount of petrol/electricity in the autonomous underwater vehicles (AUVs) can be adjusted according to the tentative duration of the operations planned.

The low and distance-dependent bandwidth, high latency, floating node mobility, sparse deployment, high error probability, frequency dependent transmission range, 3-dimensional space and poor channel conditions lead to larger power consumption during transmissions, and thus multi-hop transmissions are not always attractive in underwater communication [1]. Therefore, the proposed AR-MAC protocol is capable of continuous monitoring of critical underwater events in a very cost-effective manner, as it utilizes relatively cheap, reduced function devices (RFDs) within the network only, avoiding multi-hop transmissions.

2. Related Work

Akyildiz *et al.* [1], [2] recognize the research challenges, mainly focusing on protocol issues, hardware issues and a cross-layer design approach for UWSNs. The elemental technology used in the physical layer of underwater acoustic sensor networks is acoustic communication, because optical signals and electromagnetic signals are unsuited due to high absorption, scattering and attenuation, respectively, in the aqueous environments. They are different from terrestrial sensor networks in many attitudes: 3-dimensional space, frequency dependent transmission range, floating node mobility, longer delay, low and distance dependent bandwidth, and many more. The propagation velocity of the acoustic waves underwater is approximately 1500 m/s and the available bandwidth over a distance of a few kilometers is up to 10 kHz [2].

Liang *et al*. [3] proposed a controlled mobile sink approach in wireless sensor networks (WSNs) to prolong the network lifetime. The least amount of time at each sojourn location of the mobile sink is calculated to minimize the packet loss and the problem is formulated as a mixed integer linear programming (MILP). The results showed the enhancement in network lifetimes, provided by various algorithms by varying the value of the total distance travel per tour.

Kartha *et al*. [4] have compared analytical models for delay performance to observe that the polling model is more effective in modeling the mobility-assisted data collection framework for UWSNs. The analytical models implemented are validated with an experimental setup developed using the NS-2 based AquaSim simulator. The outcomes illustrate that the proposed framework reveals superior performance in terms of energy efficiency, network lifetime and packet delivery ratio at the cost of increased message latency.

Sendra *et al*. [5] have performed a set of measurements of electromagnetic (EM) waves at 2.4 GHz in underwater environments. The water conditions are fixed and the behavior of EM waves as a function of several network parameters, such as the working frequency, data transfer rates and modulations, are analyzed. The output data rates are calculated at different frequencies and distances.

Elrashidi *et al*. [6] presented a comprehensive study of underwater propagation of EM waves for the WSN of 2.4 GHz. A model for the path loss due to attenuation of electromagnetic waves in sea and pure water is introduced. A bow tie antenna is used to compensate for the path loss. A reflection model is introduced to illustrate the impact of air-water and water-sand interfaces as a function of distance between sensors and water depth. The path loss is evaluated and compared as a function of resonance frequency at different distances between the sensors for pure and sea water.

Letre *et al*. [7] performed an analysis of throughput and delay of unslotted IEEE 802.15.4, offering high levels of reliability of a low power and low data rate protocol. The maximum throughput and minimum delay of the unslotted version of the protocol is analyzed at 2.4 GHz. The maximum throughput and minimum delay are calculated at different operating frequencies and at various address sizes. Finally, the comparison of bandwidth efficiency for 2.4 GHz and 868/915 MHz is performed and the minimum delay – as a function of payload size at 2.4 GHz – is evaluated.

Yu *et al*. [8] introduced a low jitter scheduling technique clarifying the dependence of TDMA scheduling for sensors, and point out the correlations between scheduling delays, overall quality control and focus on reducing jitter in scheduling. The values of normalized jitter at starting and ending time slots are evaluated.

Aoun *et al*. [9] consider the problem of modeling the transmission of real-time data from a single node of a WSN to the next hop or access point. A thresholding policy supported by an analytical model is introduced, and it is responsible for deciding whether to transmit a data packet or drop it and transmit the next one. A sensor system model, packet delivery ratio calculation and probability of an empty system are introduced. The packet delivery ratio gain with respect to the packet arrival rate is evaluated.

Liu *et al*. [10] have introduced a Doppler-assisted time synchronization scheme, where the mobile Doppler scaling factor is estimated to gather relative movement velocities of sensor nodes. A model is developed to establish the relationship between the joint and individual Doppler effects caused by sensor mobility and clock skew. The synchronization, data collection, velocity estimation, propagation delay estimation and linear regression phases are analyzed. The propagation delay with respect to velocity and clock skew is estimated. An evaluation is performed.

Yang *et al*. [11] exploit mobility of the sink node to transmit non-delay-sensitive data and avoid multi-hop transmissions. An area partitioning algorithm, a transmission mechanism based on superposition coding and a MAC protocol has been proposed. The simulation results produce optimum values of the minimum distance algorithm and the maximum throughput algorithm with respect to travel distance, transmission throughput, time spent in each group and energy consumption of the group at the different numbers of sensor nodes.

Tuan Le *et al*. [12] have approximated queuing analysis for IEEE 802.15.4 sensor networks. The objective is to optimize the number of nodes deployed with a QoS guarantee. A deployment optimization model has been proposed for the beaconless mode of operation, using the M/M/1 queuing model. Such parameters as total packet loss rate and end-to-end delay at different arrival rates and at various numbers of sinks with different data rates have been evaluated and compared between the NS-2 simulation model and the M/M/1 queuing model.

Paphitis *et al*. [13] have utilized a fluid dynamic model to estimate the queue formation rate in WSN. The precise estimation of the rate with which buffers of the nodes fill-up, can contribute to congestion control. Performance can be improved by developing a mechanism based on the estimate, to control congestion and overload. The traffic flow model, queue formation rate model and congestion triggering mechanism have been introduced. The results, such as the average received packets ratio, total energy consumption and average hop-by-hop delay at different data rates are evaluated and compared.

Mao *et al*. [14] have presented how the queueing theory is applied to evaluate the performance of WSN using queuing network models. The performance analysis has considered both kinds of data and routing packets. The proposed M/M/1/K analytical model has elaborated the coverage of signal transmission, open and closed queueing networks. The mean number of packets, mean response time and mean throughput at each sensor node for both data information and routing information networks has been evaluated.

Fan Ke *et al*. [15] have presented a method to determine the trade-off between packet transmission speed and battery life in WSNs. The important data packets have been assigned to high priority queues. A network lifetime pro-

longation approach, founded on the Priority Queue N strategy, has been proposed. The variation of probability levels in different states for different service rates with different probability of packet collision, at different queue thresholds has been evaluated. Finally, the N threshold value for the lowest power consumption value has been determined.

Vanithamani *et al*. [16] have analyzed the performance of queue-based scheduling schemes in WSNs. In the Dynamic Multilevel Priority scheduling scheme, in which each node has three levels of priority, queue and non-real time packets are assigned to the two other queues based on priority. Such parameters as end-to-end delay, average task waiting time, fairness, energy consumption and network lifetime have been evaluated as well.

Byun *et al*. [17] have proposed a control-based approach to duty cycle adaptation for WSNs. High performance has been achieved under variable traffic rates by using adaptive duty cycle control through queue management. The average queue length, delay, and power consumption, at different number of hops for the proposed algorithm, have been evaluated and compared.

Peters *et al*. [18] have examined the influence of the scheduling policy on feedback control over networks based on a hybrid communication protocol, which incorporates both contention-free and contention-based medium access. A number of scheduler-controller co-design algorithms have been compared that take both the contention-based and guaranteed parts of the protocol into account using an approximation to an infinite horizon linear quadratic cost function.

Pati *et al*. [19] have proposed a conflict-free query scheduling approach in WSNs, as conflict-free real-time query scheduling is one of the most difficult tasks. The results, such as response time for different scheduling algorithms with fixed deadlines and varying query rates, have been evaluated and compared.

Zhang *et al*. [20] have proposed scheduling with predictable link reliability for wireless networked control. Such a solution is required for wireless networked control, yet co-channel interference remains a major source of uncertainty in wireless link reliability. The Physical-ratio-k (PRK) model and PRK-S algorithm required to implement it have been introduced. The results ensured that PRK-S reduces the communication delay and increases the throughput more effectively than existing scheduling protocols. Such parameters as packet delivery reliability, settling time, median latency and throughput have been evaluated and compared for different algorithms.

Pati *et al*. [21] have proposed an Advanced Real-Time Query scheduling approach for WSNs. It works with three new schedulers: Advanced Non-Preemptive Query Scheduler, Advanced Preemptive Query Scheduler and Advanced Query Scheduler. The response time with fixed deadlines and varying query rates have been evaluated.

Peters *et al*. [22] have investigated the co-design of a scheduler and controller for feedback control over wireless industrial hybrid protocol networks. The possibilities and limitations of feedback control over hybrid networks that are based on IEEE 802.15.4 protocol are considered. The results illustrated that careful co-design of the scheduler-controller results in significant performance gains compared to round robin heuristics.

Shahid *et al*. [23] have used an improved low power scheduler for OSS-7 by implementing the open source DASH 7 stack. It has improved the performance up to 75%, on average, in terms of execution time, thus leading to reduced dynamic power consumption.

Chovance *et al*. [24] have presented a real-time scheduler in an operating system running on the ARM Cortex (M0, M3, M4) processors used in small mobile robotics, with the kernel response of around 1 ms. It can be used for WSNs due to its strong modularity, advanced sleep modes and event-driven programming capability.

Jhamshed *et al*. [25] have proposed a node-level optimal real-time priority based dynamic scheduling algorithm that schedules tasks according to their priorities, by changing the frequency with the use of dynamic voltage frequency scaling, and hence makes more efficient use of energy and helps in achieving QoS for delay sensitive traffic.

In this paper, a queuing model has been developed with a mechanism which utilizes the sink node velocity, the most suitable path with controlled mobility, low duty cycle, appropriate number of queues having optimum queue length, proper schedulers and, finally, the AR-MAC protocol has been implemented to improve network performance.

The rest of the paper is organized in the manner described below. Section 2 explains the related work. Section 3 illustrates the proposed AR-MAC protocol. Section 4 presents the preliminaries. Mathematical models are explained in Section 5. Section 6 contains an explanation of the simulation scenarios and algorithms. Sections 7 and 8 illustrate simulation results and contain their analysis. Section 9 concludes the paper and presents the future work.

3. The Alive-in-Range MAC Protocol

In AR-MAC as the sink node is continually mobile without sojourns. A centralized duty-cycle based MAC protocol is used to schedule the transmissions from the sensors. Normally, all sensor nodes are in the sleep state. As the sink traverses the most effective path, sensors coming into the range of the sink node become active and start transmissions. The maximum data transmission range equals 20 m. It is determined with respect to the effective power received at the sink node. In the example shown in Fig. 1, when the mobile sink node is at location 1, sensor nodes A and B are within range. After the sink moves into the range, sensor nodes A and B become active after sensing the beacons. At locations 2 and 3, sensors A, B, C and D are within range and become active. At position 4, sensor nodes A and B are out of range and enter the sleep mode, but nodes C and D are still active. The centralized approach to schedule active cycles eliminates collisions and reduces the synchronization

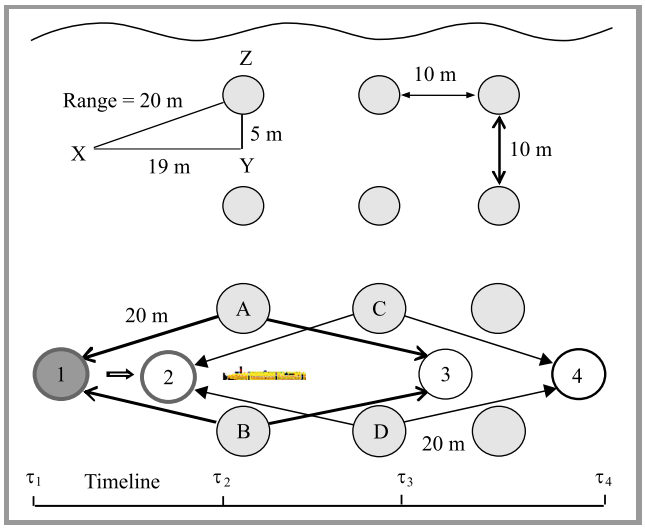

Fig. 1. Simulation architecture of the AR-MAC protocol.

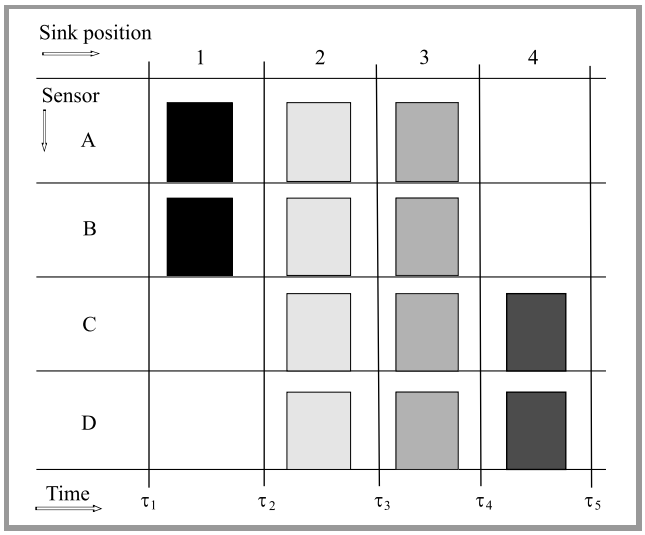

Fig. 2. The timing diagram of active and sleep scheduling in AR-MAC.

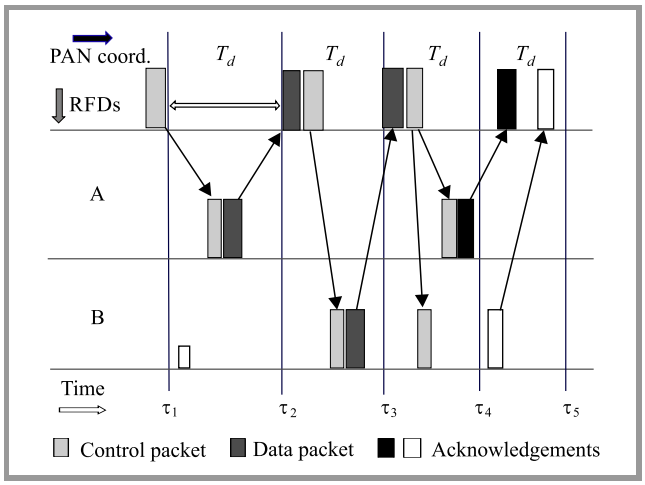

Fig. 3. The timing diagram of data transmission and reception in AR-MAC.

requirements among the sensors. Figure 2 presents the timing diagram of active/sleep scheduling. The process of data transmission and data reception of the proposed AR-MAC protocol with two sensors is shown in Fig. 3. It illustrates that, after the sink moves to the data collection position, it polls each sensor sequentially. A sensor with data to transmit does so immediately after it is polled, and, in the absence of data, sends a small packet to just acknowledge the receipt of the poll. Then the sink estimates the propagation delay for each sensor based on the measurement of the round-trip time. A sensor may be re-polled immediately if its data is not received correctly. Also, if no response is obtained in response to a poll, the sink may repeat polls subject to the limited number of retransmissions. After the uplink transmissions, the sink transmits the control messages to the sensors. The messages sent by the sink also contain the schedule according to which the sensors send back their acknowledgement packets (ACK).

In this paper an attempt has been made to optimize the queue performance by avoiding the use of relay nodes/full function devices (FFDs) to forward the data from sensor nodes acting as Reduced Function Devices (RFDs). It is achieved by using sensor nodes only for the deployment of UWSNs and by adjusting the speed of the mobile sink, so that it will collect data from all sensor nodes without the requirement of any sojourn. Moreover, the duty cycle of the sensor nodes is minimized as they become active only when the sink node is located within their transmission range. Otherwise, they are in sleep mode, which reduces energy consumption, minimizes collisions, diminishes the number of packets dropped and eliminates the formation of bottlenecks at the sink nodes.

In recent research, there are several bottleneck constraints affecting the mobility of a sink for the enhancement of network performance. These constrictions include the maximum distance between two consecutive movements, the maximum number of sojourn locations and the minimum sojourn time at each sojourn location. The distance-constrained mobile sink problem is dulled by controlling the mobility and by finding the most suitable path for the sink node to traverse the sensor network. The mobility of a sink node is beneficial for better balancing of energy consumption among the sensors.

4. Preliminaries

The preliminaries are as follows:

- an underwater wireless sensor network $U(S,E)$ is deployed comprising of n static sensors and a mobile sink, where S is the set of sensors and E is the set of radio links,
- there is no link between two static sensors as all the immobile sensors are RFDs,
- the sink and the sensor node are able to communicate if they are within the communication range of each other,

- the energy consumed by all communication operations is considered,
- it is assumed that by using radio positioning techniques, the sink is aware of the location of the sensors,
- all sensors have the same data generation rate r,
- the mobile sink offers a limited amount of energy which is sufficient for the completion of the monitoring operation, compared to the energy capacity of sensors,
- there is no sojourn location of the mobile sink, as it is moving continuously.

5. Network Models and Assumptions

5.1. Underwater Electromagnetic Wave Propagation

Conductivity, density and permittivity of water are different from and higher than those of air [17], [18]. Water differs from the air, as it has higher conductivity, higher density and higher permittivity. The value of relative permittivity of pure water is $\varepsilon_r = 79$, density is 1000 kg/m^3 and tangent loss is $\varepsilon_t = 0.924$ at 2.4 GHz. In the case of sea water $\varepsilon_r = 80.4$, density is 1033 kg/m^3 and $\varepsilon_t = 1.527$ at 2.4 GHz. The concentration of salt in the sea water determines the value of relative permittivity, which is normally 3%. In this paper the simulations are carried out for sea water only. Electromagnetic waves experience a very high path loss in underwater propagation. The Friis equation introducing the value of power received, in terms of transmitter and receiver antenna gains, transmitted power and path loss at the receiver, is expressed by the following formula:

$$\Psi_r = \Psi_i + G_i + G_r - \Theta_{path}\,, \tag{1}$$

where the gain of the transmitter and receiver antennas is denoted as G_t and G_r, the received power is Ψ_r, the transmitted power is Psi_t and the underwater path loss is Θ_{path}. All units are dBm. The expression for path loss (in dB) is:

$$\Theta_{path} = \Theta_0 + \Theta_m + \Theta_a\,. \tag{2}$$

The path loss in the air is denoted as Θ_0 and formulated as:

$$\Theta_0 = 20\log\frac{4\pi df}{\mathrm{c}}\,, \tag{3}$$

where c the speed of light in the air (m/s), the distance between transmitter and receiver in meters is denoted by d, the operating frequency is symbolized as f.
The path loss due to change in propagation medium Θ_m is:

$$\Theta_m = 20\log\frac{\lambda_0}{\lambda}\,, \tag{4}$$

where the wavelength of the signal in the air is signified as λ_0 and calculated as $\lambda_0 = \dfrac{\mathrm{c}}{f}$, the wave factor λ is calculated as $\lambda = \dfrac{2\pi}{\beta}$ and the phase shifting constant is denoted as β. Its value can be expressed as:

$$\beta = \omega\sqrt{\frac{\mu\varepsilon}{2}}\left(\sqrt{1+\left(\frac{\sigma}{\omega\varepsilon}\right)^2}\right)+1\,, \tag{5}$$

where ω is the angular frequency (rad/s), σ is conductivity in (S/m) and μ is permeability (H/m). The path loss due to attenuation in medium Θ_a is (in dB):

$$\Theta_a = 10\log \mathrm{e}^{-2\alpha d}\,, \tag{6}$$

where the attenuation constant is denoted by α. Its value can be calculated from:

$$\alpha = \omega\sqrt{\frac{\mu\varepsilon}{2}}\left(\sqrt{1+\left(\frac{\sigma}{\omega\varepsilon}\right)^2}\right)-1\,. \tag{7}$$

The parameters like propagation speed of EM waves v and the absorption coefficient ϑ are:

$$v \approx \frac{1}{\sqrt{(1+\chi_e)\times 8.85\times 10^{-12}\times\mu_r\times 4\pi\times 10^{-7}}}\,, \tag{8}$$

$$\vartheta = \frac{\sigma}{2}\times\sqrt{\frac{\mu_r\times 4\pi\times 10^{-7}}{(1+\chi_e)\times 8.85\times 10^{-12}}}\,, \tag{9}$$

where ε_0 and μ_0 are the permittivity and permeability of free space. The relative permittivity and permeability of sea water are denoted by ε_r and μ_r respectively.

5.2. Calculation of Reflection Loss of Water Boundaries

The reflection is caused by the boundaries of water bodies such as water-air and water-sand interfaces [18], as shown in Fig. 4. The value of the reflection coefficient, Γ can be expressed as:

$$\Gamma = \frac{\rho_2 v_2 - \rho_1 v_1}{\rho_2 v_2 + \rho_1 v_1}\,, \tag{10}$$

where the density of water is denoted as ρ_1 and of the other medium is signified as ρ_2. The wave velocity in water is denoted as v_1 and in another medium as v_2. The reflection

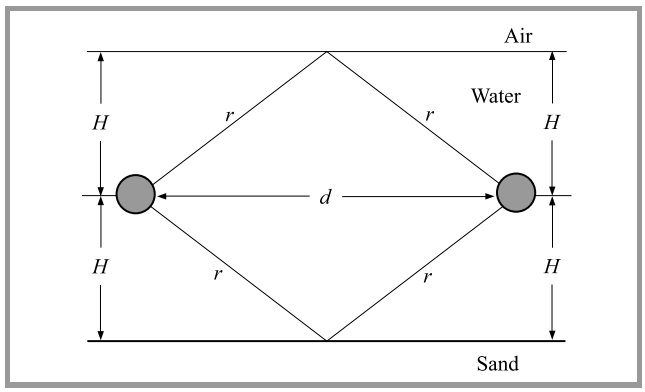

Fig. 4. Diagram representing the three-path reflection model.

loss from the surface and from the bottom is Θ_{ref} expressed as:

$$\Theta_{ref} = -V = -10\log V\,, \tag{11}$$

where V (in dB) is calculated as:

$$V^2 = 1 + \left(|\Gamma| e^{-\alpha\Delta(r)}\right)^2 - 2|\Gamma| e^{-\alpha\Delta(r)} \cos\left(\pi - \left(\phi - \frac{2\pi}{\lambda}\Delta(r)\right)\right), \tag{12}$$

where $\Delta(r)$ is the difference between r and d (in meters), $|\Gamma|$ and ϕ are the amplitude and phase of the reflection coefficient respectively and r is the reflected path length (in meters). The expression to calculate r is:

$$r = 2\sqrt{H^2 + \left(\frac{d}{2}\right)^2}\,. \tag{13}$$

5.3. Calculation of Throughput and Delay

The maximum throughput value is directly proportional to the payload bytes and inversely proportional to the delay experienced by each data packet [19]. The superframe structure of IEEE 802.15.4. is shown in Fig. 5. The precise

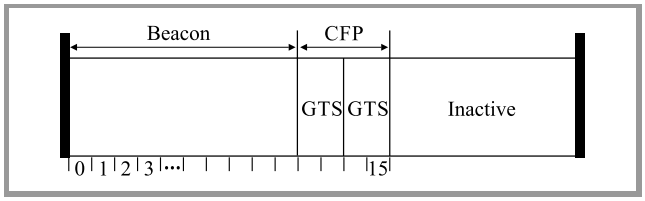

Fig. 5. Structure of the superframe in IEEE 802.15.4.

calculation of the delay and the determination of the number of payload bytes are very important to exactly calculate the maximum throughput. The total delay comprises of the

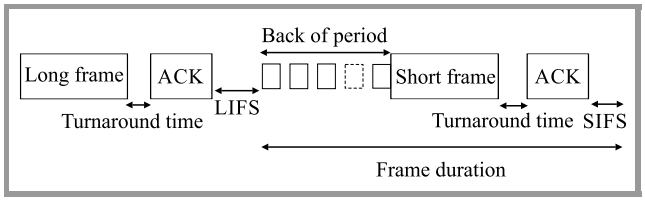

Fig. 6. The sequence of the IEEE 802.15.4 frames.

delay in data reception and the delay due to various frame sequences as shown in Fig. 6. Subsequently, the throughput ρ can be formulated as:

$$\rho = \frac{8 \cdot \varphi}{\delta(\varphi)}\,, \tag{14}$$

where φ denotes the payload bytes received from the network layer as shown in Fig. 7. The delay $\delta(\varphi)$ experienced by each packet can be expressed as:

$$\delta(\varphi) = \tau_{BO} + \tau_{frame}(\varphi) + \tau_{TA} + \tau_{ACK} + \tau_{IFS}(\varphi)\,, \tag{15}$$

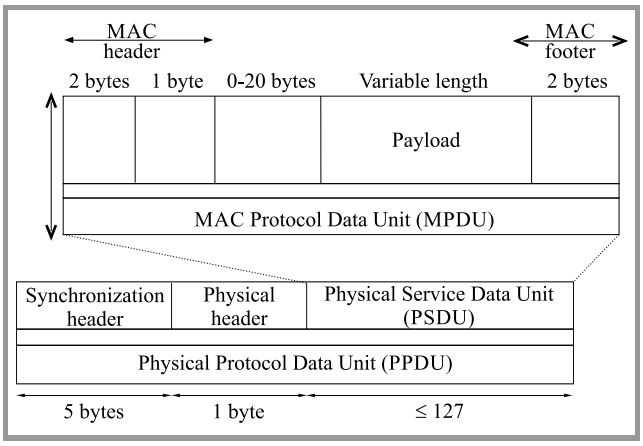

Fig. 7. Structure of the IEEE 802.15.4 frames.

where: τ_{BO} – back off period in s, $\tau_{frame}(\varphi)$ – transmission time for a payload of φ bytes in s, τ_{TA} – turnaround time in s, τ_{ACK} – transmission time for an ACK in s, $\tau_{IFS}(\varphi)$ – interframe sequence time in s, κ – payload size in bytes.

If the size of the MAC Protocol Data Unit (MPDU) is 18 bytes or less, then short interframe space (SIFS) is used for the interframe space (IFS). Otherwise, a long interframe space (LIFS) is used. The expression of the MAC Protocol Data Unit (λ) is given by:

$$\lambda = \phi + \mu + \kappa\,. \tag{16}$$

To calculate the total delay, it is important to calculate the time duration of its all components. The expression on the back off period is:

$$\tau_{BO} = \alpha \cdot \beta\,, \tag{17}$$

where α is the number of back off slots and β is the time for a back off slot.
The range of back off slots lies in the interval $(0,\, 2^{BE}-1)$, where BE means the *back off exponent* having a minimum value of 3. As, the communication channel is assumed as perfect the value of BE will remain the same. Therefore, the mean of the interval gives the number of back off slots which are: $\frac{2^{BE}-1}{1}$ or 3.5.
The total duration of the frame (in seconds) is:

$$\tau_{frame}(\varphi) = \frac{\psi + \phi + \Phi + \varphi + \mu}{\omega}\,, \tag{18}$$

where: ψ – length of the PHY header (6 bytes), ϕ – length of the MAC header (3 bytes), Φ – length of the information field of the MAC address (in bytes), μ – length of the MAC footer (2 bytes), ω – raw data rate (in b/s).

Equation (19) formulates the duration of an acknowledgement:

$$\tau_{ACK}(\varphi) = 8\frac{\psi + \phi + \mu}{\omega}\,. \tag{19}$$

The values of acknowledgement and turnaround times are 0, if acknowledgements are not used.

Finally, the throughput ρ (in b/s) and delay $\delta(\varphi)$ (in seconds) are expressed:

$$\rho \frac{8 \cdot \varphi}{m \cdot \varphi + n}, \tag{20}$$

$$\delta(\varphi) = m \cdot \varphi + n. \tag{21}$$

In the Eqs. (20)–(21), the values of m and n are influenced by the operating frequency of the channel, the length of data bytes in the payload (LISF or SIFS) and the length of the address. The values of parameters m and n are determined by using the curve fitting method considering the principle of least squares and by solving the respective normal equations to get the line of best fit.

5.4. Calculation of Jitter

To calculate jitter [20], let us assume the UWSN as a set M consists of n nodes, $M = \{M_1, M_2, M_3, \ldots, M_n\}$. Since each node periodically sends its status to the personal area network (PAN) coordinator or receives commands from it, we consider the set Γ' composed of n periodic tasks. Every task $\Gamma' j = (R_j, D_j, T_j) \in \Gamma'$ is characterized by the required number of timeslots R_j, a relative deadline D_j and the inter arrival time T_j. Effective tasks scheduling using adaptive active/sleep cycles is very important to reduce jitter. To avoid missing deadlines and to improve schedulability, the tasks are prioritized by various schedulers. When a task Γ'_j is scheduled at time slot s, the priority of this task P_j is calculated as:

$$P_j = D_j - s - T_j. \tag{22}$$

5.5. Calculation of Packets Dropped

In this paper, the Poisson process is implemented for the generation of packets at each sensor with parameter λ [21]. The exponentially distributed service time with parameter μ depends on the conduct of MAC protocols, and the influence of channel contention from other sensors. For the successful packet transmission, it should get completed within d time units after its arrival at the queue. For example, if a packet reaches at time t, it must get transmitted before its universal deadline $t + d$, where d is a fixed relative deadline. The probability of a successful packet transmission of the system is denoted as $PD(\theta)$ and its value is computed, where the packet dropping threshold, expressed in seconds, is denoted by θ. The number of packets present in the system, including the packets present in the transmitter immediately before a packet arrival is denoted by X. There are three states of the system experienced by a packet arriving in the system. These are empty, busy transmitter and full. The meaning of these states is explained as follows:

- *Empty* ($X = 0$) – the transmitter is available, and the buffer is empty as well;
- *Busy Transmitter* ($X = 1$) – the transmission of a packet is in progress but the buffer is empty;
- *Full* ($X = 2$) – the transmission is in progress and the buffer is full as well. In that case the reaching packet overwrites the packet already buffered.

The value of $PD(\theta)$ can be calculated depending on whether the packet arrives at an empty state or not. The service time experienced by the arriving packet is denoted by Q. The expression of $PD(\theta)$ is:

$$PD(\theta) = \Pi(X = 0)\,\Pi(Q \le d) + \\ + \big(1 - \Pi(X = 0)\big) \int_0^{d-\theta} \mu\, \mathrm{e}^{\mu t}\, \Pi(Q \le d - t)\, \mathrm{e}^{-\lambda t} \mathrm{d}t. \tag{23}$$

The transmission of an arbitrary packet Π, which arrives at an empty system, can be successful only if the required time Q is less than d. This explains the term $\Pi(X = 0)\Pi(Q \le d)$. If Π arrives when the system is busy in transmitting a packet ($X = 1$ or 2), the beginning of its transmission depends on the remaining transmission time t of the packet that is presently being transmitted. The density of the remaining service time is exponentially distributed as $\mu \mathrm{e}^{-\mu t}$. It is important to understand that these three conditions must be satisfied for the timely service of Π. First, the deadline duration is larger than θ when the remaining service ends, i.e. $d - t \ge \theta$. Second, it is explained by the factor $\Pi(Q \le d - t)$, which means that the service time is less than $t - d$. Third, the term $\mathrm{e}^{-\lambda t}$ explains it as there were no other arrivals during t time units. The representation of data packet arrivals and servicing is shown in Fig. 8.

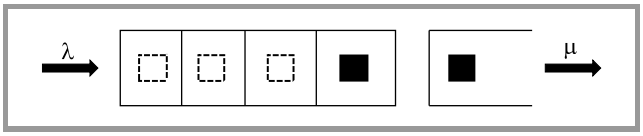

Fig. 8. The schematic representation of a queue.

5.6. Model of AR-MAC Protocol

Describing $f(t)$ as the transmitted signal sent from sensor node g to h, the expression of the transmitted signal is [22]:

$$f(t) = f(t + T_0), \tag{24}$$

where T_0 denotes the time duration of the signal. The received signal at sensor node h is defined as [22]:

$$z(t) = \mathrm{e}^{-j2\pi \frac{\varphi}{1+\varphi} f_c T_0} z\Big(t + \frac{T_0}{1+\varphi}\Big), \tag{25}$$

where φ is the Doppler scaling factor based on relative velocity between sensor nodes and the mobile sink, and the clock skew is denoted as δ. The summation of the signals that arrived along multiple physical paths is the received signal at the sink node and can be formulated as [22]:

$$Q_{hg}(t) = \sum_{k=1}^{N_k} A_k\, f_{gh}\big[(1 + \varphi_m)t - \tau_k)\big], \tag{26}$$

where $f_{gh}(t)$ is the message transmitted from sensor node g to h, with node g as the reference node. τ_k and A_k denote the delay and amplitude of the k-th path respectively. N_k signifies the number of paths. The joint Doppler scaling factor induced by the node mobility is thus $\varphi_m \cong \frac{v}{\sigma}$, where σ is the speed of electromagnetic waves in water and v is the relative velocity. The relative velocity of the two sensor nodes based on the joint Doppler scaling factor φ_m can be directly obtained from [22]:

$$v \cong \varphi_m \sigma = \big((1+\varphi_{gh})\phi - 1\big)\sigma\,, \tag{27}$$

the ϕ signifies clock skew and φ_{gh} is Doppler scaling factor for sensors g and h. Finally, the propagation delay is [22]:

$$\tau = \frac{\delta\sigma - \phi t_r(\sigma + \vartheta) - \mu}{2\phi\sigma}\,, \tag{28}$$

where ϑ is the velocity of the mobile sink (in m/s), $t_r = t_3 - t_2$ and $\delta = t_4 - t_1$ as t_1, t_2, t_3 and t_4 are different time instants, $\mu = \frac{1}{2\alpha t_r^2}$ as α denotes acceleration of the mobile sink node.

Let v be the velocity of the mobile sink. D_{ij} is the path length in meters from location i to j during, which the sensor nodes from S_i to S_j are active and is expressed as:

$$D_{ij} = v(\tau_j - \tau_i)\ \forall i, j\,. \tag{29}$$

The sensors are having different on/off time periods depending on the location and velocity of the mobile sink. The sensor number S_i active during the i-th time slot τ_i is expressed as:

$$\sum_{i}^{i+3} S_i = \tau_i, \quad \text{where} \quad i = 1\,, \tag{30}$$

$$\sum_{j}^{j+3} S_j = \tau_j, \quad \text{where} \quad j = i+3\,. \tag{31}$$

The expression for the n-th number of sensor is:

$$\sum_{n}^{n+3} S_n = \tau_n, \quad \text{where} \quad n = j+3\,. \tag{32}$$

The total simulation time T_{sim} is 670 s for 100 nodes and 1395 s for the 200 nodes:

$$T_{sim} = \overset{\tau_1}{\underset{0}{T_1}} + \overset{t_2}{\underset{\tau_1+1}{T_2}} + \ldots + \overset{t_n}{\underset{\tau_{n-1}}{T_n}}\,. \tag{33}$$

The duty cycle of each sensor node (RFD) is adjusted by varying the on/off time. In the proposed scenario a sensor node is active for 28 s, which means that for a 100 node network, with the total simulation time of 670 s, the duty cycle is about 4.1%.

Likewise, for a network of 200 nodes, having the total simulation time of 1395 s, the duty cycle is about 2%. The aim of the proposed AR-MAC is to minimize the average end-to-end delay, minimize jitter, and to improve other performance metrics. The aim to achieve a minimum average end-to-end delay is:

$$Aim = \min \delta(\varphi)\,. \tag{34}$$

5.7. Analysis of the Proposed Queuing Model

The proposed queueing model uses the slotted CSMA-CA channel access mechanism in a beaconed mode of operation. In order to transmit data, the device needs a guaranteed time slot during the contention-free period (CPF). If the channel is busy, during the contention access period the device waits for a random number of unit back-off periods ($\sum$) with the range of (0, $2^{BE}-1$) before performing the clear channel assessment (λ). The transmission takes place if the channel is at idle. Otherwise it has to wait for another random period before trying to access the channel again. The initial value of the back off exponent *BE* is equal to *macMinBE*.

Assuming that the channel is free and that the default *macMinBE* value is 3, the worst-case channel access time can be calculated as:

$$\int_{CSMA\text{-}CA} = \varsigma + \lambda \Rightarrow (2^3 - 1)\sum + \lambda \Rightarrow 2.368 \text{ ms}\,. \tag{35}$$

The duration of one symbol period is 16 μs, *aUnitBackoffPeriod* is equal to 20 symbol periods and the clear channel assessment detection time is equal to the duration of 8 symbol periods. The data frame transfer time denoted as $\int_{Data}$ of a channel having the data rate of (C) 250 kb/s is expressed in Eq. (36), where $\wp$ is the *MaxPHYPacketSize*.

$$\int_{Data} = \frac{\wp 8}{C}\,, \tag{36}$$

$$\int_{Data} = 4.064 \text{ ms}\,.$$

The size of an acknowledgement frame is 11 bytes. At the fundamental data rate of 250 kb/s, its transmission takes 0.352 ms. The transmission of acknowledgement uses the CSMA-CA mechanism. The transmission of an acknowledgement frame commences *aTurnaroundTime* symbols after the reception of the data frame, where *aTurnaroundTime* is 192 μs. This allows the device enough time to switch between transmitting and receiving modes. The maximum data payload is 127 bytes. The expression for effective service time using a stop-and-wait Automatic Repeat Request (ARQ) protocol can be calculated as:

$$\int_0 = \int_{CSMA\text{-}CA} + \int_{Data} + \int_{Turnaround} + \int_{Ack} + \int_{Prop}\,, \tag{37}$$

where $\int_{Prop}$ is the propagation time between the sensor node and the sink. The value of $\int_{Prop}$ is negligible and can be ignored in the calculation, and $\int_{Ack}$ is the acknowledgement transmission time.

In this model, the sensors relay the sensed data directly to the sink node using single hop. The transmission time through the single-hop can be calculated as:

$$\int_{s\_hop} = \int_{single\_hop} + \Delta\,, \tag{38}$$

where $\int_{single\_hop}$ is the transmission time taken to transmit one data frame through one hop. In this case, $\int_{single\_hop} = \int_0$, Δ is the processing time for routing.

The overall packet loss rate denoted as $\Re_{Totalloss}$, may constitutes of packet loss due to dropping from the queue with the rate of overflow ($Re_{Overflow}$), or due to channel errors. The packet error rate at physical layer is denoted as $\Re_{Phy}$. The rate of packet loss over the wireless channel, after U times of packet retransmissions, will be:

$$\Re_{Phyloss} = \Re_{Phy}^{U+1}. \tag{39}$$

Consequently, the overall packet loss rate will be:

$$\Re_{Totalloss} = \Re_{Overflow} + (1 - R_{Overflow})\Re_{Phyloss}. \tag{40}$$

Neglecting the value of $R_{Overflow} \times \Re_{Phyloss}$ as it is negligible; the total packet loss rate can be approximated as:

$$\Re_{Totalloss} = \Re_{Overflow} + R_{Phyloss}. \tag{41}$$

In this paper, the M/M/1 queuing model with the Poisson distribution arrival rate λ and the exponential service time μ is implemented. The probability density function (PDF) denoted as $g(t)$ of service time is:

$$g(t) = \mu e^{-\mu t}. \tag{42}$$

In the retransmission model, the PDF of the service time due to the packet transmission can be calculated as:

$$g(t) = \sum_{w=0}^{U} g\left(\frac{t}{w}\right) P(w), \tag{43}$$

where the probability of attempting w retransmissions of a single packet to succeed or to reach the retry-limit U is denoted as $P(w)$ and $g(\frac{t}{w})$ denotes the service time conditional PDF, given w retransmissions of a single packet. The $P(w)$ is:

$$P(w) = (1 - \Re_{Phy})\,\Re_{phy}^{w}. \tag{44}$$

The service time at the w-th retransmission is formulated as:

$$\mu = \mu_0 (w+1), \tag{45}$$

where μ_0 is the service time of system without an error in the wireless channel. The μ is 6.976 ms.

So, the service time conditional PDF, given that w retransmissions of a single packet, is:

$$g\left(\frac{t}{w}\right) = \mu\, e^{-\mu_w t} \tag{46}$$

from Eqs. (42)–(45), the service time due to the packet retransmission can be written as:

$$g(t) = \mu_0 e^{-\mu_0 t} + \mu_0 \sum_{w=1}^{U} \Re_{Phy}^{w} \left(\frac{1}{w+1} e^{\frac{-u_0}{w+1} t} - \frac{1}{w} e^{\frac{-\mu_0}{w} t} \right). \tag{47}$$

The mean value for service time $E\big(g(t)\big)$ is defined as:

$$E\big(g(t)\big) = \mu = \mu_0 \frac{1 - \Re_{Phy}}{1 - \Re_{Phy}^{U+1}}. \tag{48}$$

In this model every sensor node has a finite queue length denoted as Q. Therefore, the packet overflow drop rate is equal to the probability that a packet cannot find available room for storage, and is hence dropped.
With one packet under service, the overflow rate can be formulated as:

$$\Re_{Overflow} = \sum_{i=Q+1}^{\infty} (1-\rho)\rho^{i} = \rho^{Q+1}. \tag{49}$$

The total packet loss for the M/M/1 queuing model can be written as:

$$\Re_{Totalloss} = \left(\frac{\lambda_0}{\mu_0} \frac{1 - \Re_{Phy}^{U+1}}{1 - \Re_{Phy}} \right)^{Q+1} + \Re_{Phy}^{U+1}. \tag{50}$$

The average number of packets in the system is:

$$\overline{\Im} = \frac{\rho}{1-\rho} = \frac{\lambda}{\mu - \lambda}. \tag{51}$$

The average time spent in the system is:

$$\Delta = \frac{\overline{\Im}}{\lambda} \frac{\rho}{1-\rho} \frac{1}{\lambda} = \frac{1}{\mu - \lambda}, \tag{52}$$

where $\rho = \frac{\lambda}{\mu}$. The average spending time will be:

$$\Delta = \frac{1}{\mu_0 \cdot \dfrac{1 - P_{phy}}{1 - P_{Phy}^{U+1}} - \lambda}. \tag{53}$$

6. Simulation Scenario and Algorithms

Figure 9 shows the general scenario of UWSNs consisting of different types of sensor nodes and autonomous underwater vehicles used for collaborative monitoring. The

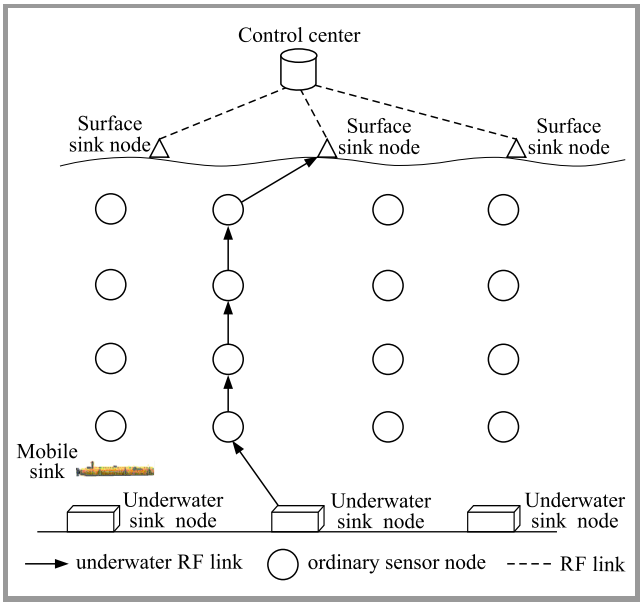

Fig. 9. General scenario of UWSN.

underwater sensors are linked with each other via microwave links. Further, the surface sink nodes are connected to the control center via RF links. The architectures are application dependent, either of the 3D or 2D variety. The ordinary sensor nodes sense and relay data using a direct link, or through a multi-hop path. The mobility of sensor nodes, as well as of sink nodes can be classified as controlled mobility and uncontrolled mobility. This paper implements radio links for both underwater and free space communication. The sensed data are transmitted through direct radio links and the mobility of the sink node is controlled.

The lawn mower trajectory of the mobile sink, as shown in Fig. 10, is considered after extensive simulations to reduce the requirement of multi-hop data transmissions. The network constitutes of RFDs, except for the mobile sink, to avoid multi-hop data transmissions and to reduce the depletion of battery power. This trajectory is most suitable, as the sink node is available for all sensors within their transmission range after the regular interval of times, but at the cost of an increased path length. The deteriorating effect of the increased path length is reduced by implementing an effective path determination (EPD) algorithm. The velocity of the sink node is selected so that each sensor node gets sufficient time to synchronize and to transmit the sensed data without the requirement of any sojourns.

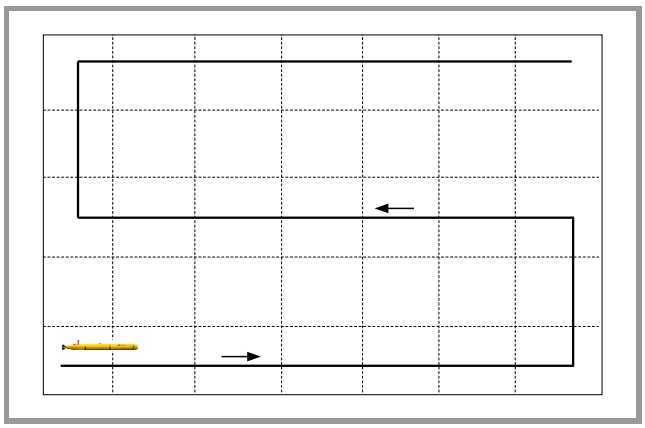

Fig. 10. Representation of the mobile sink's Lawn-mower trajectory.

The expression of transmission data rate r of data sent to sensor q denoted by r_q [23] can be written as:

$$r_q = B\log_2\left(1+\frac{p_q g_q(x^*,y^*)}{\sum_{m=q+1}^{n} P_m h_q(x^*,y^*)+N_0}\right) = l_q K_r^*, \quad (54)$$

where B is the available bandwidth, P_q is the power assigned to transmission to sensor q, $g_q(x^*,y^*)$ is the channel gain as a function of the distance, K_r^* is the ratio of rate of transmissions to the message length, l_q is the length of the message sent to sensor q, and N_0 is the expected ambient noise. To attain the transmission power P_i in the terms of x^*, y^* and K_r^*, Eq. (54) can be recursively solved to:

$$P_i(x^*,y^*,k_r^*) = (2^{L_q K_r^*}-1)\frac{N_0+\sum_{m=q+1}^{n} P_m(x^*,y^*,K_r^*)h_q(x^*,y^*)}{h_q(x^*,y^*)}. \quad (55)$$

Equation (55) illustrates the influence of all prime parameters on the power allotted to the transmission to sensor i. To determine the effective path length and the optimum throughput path, the sink needs to evaluate the maximum K_r possible over the entire network and then select the position with the highest K_r.

Algorithm 1 is proposed to find out the effective path length and Algorithm 2 to determine the maximum throughput path for the mobile sink. It is assumed that the sensors are properly anchored to limit their movement due to waves within the permissible limits to follow these algorithms.

The network scenario presented in this paper has implemented different queue types as FIFO, random early detection (RED), random early detection with I/O bit and weighted random early detection. Each of them is implemented with different queue sizes in bytes such as 20,000, 22,000, 25,000, 26,000, 27,000, 30,000, 50,000, 150,000, 200,000, 250,000, 500,000 and 1,000,000. The schedulers employed are differentiated as services weighted fair, round robin and self-clocked fair. The intent is to effectively schedule the transmission and reception of data packets.

Algorithm 1: Effective path length determination

State 1: The initial position of the mobile sink.
The initial position be P on the water surface.
Let the desired position is $\mathscr{P}$.
while $P \neq \mathscr{P}$ **do**
  $P = \mathscr{P}$
  State 2: Connections using radio channel
  Initialize the maximum range $= r_{\max}$.
  The location of the sensor k is L_k.
  The total number of sensors in the network is N.
  The point of the present position of the sensor k is L_k^m while sink and sensor node k distance is $D_{k,m}$.
  $D_{k,m} = \| L_k^m - P \|, \forall L_k^m \in P$.
  if $D_{k,m} \leq r_{\max}$ **then**
    Path found; break.
  end if
  State 3: Revaluate effective path and reallocate sensors
  Find P that minimizes max $\| L_k^m - P \| \, \forall P \in L_k^m$.
  if $P = \mathscr{P}$ and $D_{k,m} > r_{\max}$ for any m **then**
    No path found; break
  end if
end while
State 4: Location modification
Solve the Eq. (55) to attain the optimal location to achieve the maximum K_r for the sink node.

Algorithm 2: Optimum throughput path determination

Initialize: number of segments $S = 1$;
iteration time $q = N$, set segment $(k) = 1 \forall\, k$
Generate $n \times n$ distance matrix X and calculate throughput $\hbar$.
while $q > 0$ **do**
  Find the reference throughput $\hbar$ at distance $r_{\max} =$ 20 m.
  if $\hbar_{new} \geq \hbar$ **then**
    Optimum throughput path found. Delete entries in X corresponding to k.
    $S = S + 1; \quad q = q + 1$
  end if
end while
State: Recalculate optimum throughput path
while $q > 0$ **do**
  if $\hbar_{new} < \hbar$ **then**
    Optimum throughput path not found.
  end if
end while
Solve the Eq. (55) to attain the optimal location to achieve the maximum K_r for the sink node.

7. Simulations Results and Analysis

The simulation parameters are shown in Table 1. The simulations are performed using the Qualnet 6.1 software. Ten simulation rounds are performed and their average value is considered to obtain more accurate results. The flowchart demonstrating steps followed in the simulation process is shown in Fig. 11.

Table 1
Simulation parameters

No.	Parameter	Value
1	Total number on nodes	101, 201
2	Number of PAN coordinators	1, 1
3	Number of RFDs	100, 200
4	Communication protocols	
	RFDs to PAN coordinator	IEEE 802.15.4
	PAN to RFDs	IEEE 802.15.4
5	Operating frequency	2.4 [GHz]
6	Packet size	38 [bytes]
7	Packet interval	1 [s]
8	Simulation times	670, 1395 [s]
9	Battery	1200 [mAh]
10	Transmission range	20 [m]
11	Node connectivity	Protocol, frequency [GHz]
	RFDs to PAN	ZigBee, 2.4
	PAN to RFDs	ZigBee, 2.4
12	Terrain area	$110 \times 20 \times 110$ m $210 \times 20 \times 110$ m
13	Energy model	Mica-motes
14	Battery model	Linear
15	Optimum queue size	26,000 [bytes]
16	Queue type	FIFO
17	Scheduler type	DSWF
18	Application	Traffic generator
19	Speed of PAN	0.7164 [m/s]

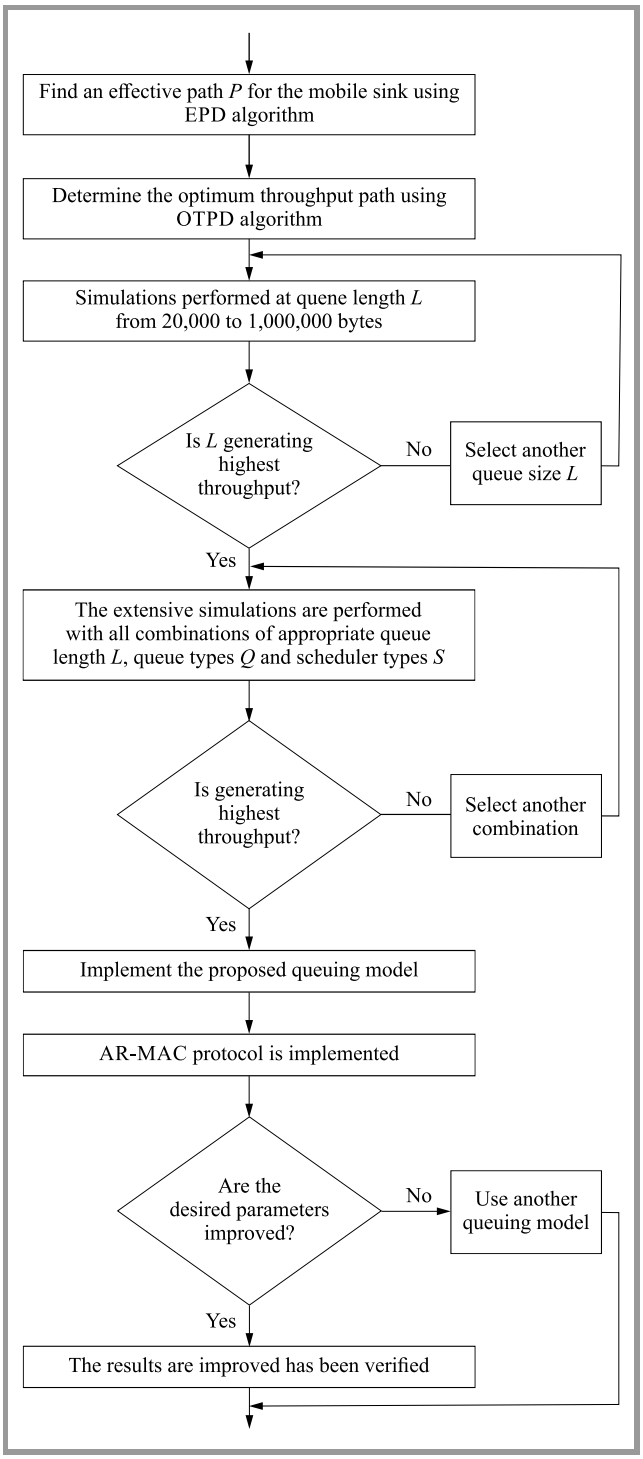

Fig. 11. Flowchart of the proposed model.

To validate the influence of queue size, queue type and scheduler type, performance parameter is calculated first to find out the most suitable queue size, as shown in Fig. 12, and to find out the most appropriate combination of types of queues and schedulers with the most efficient queue size, as

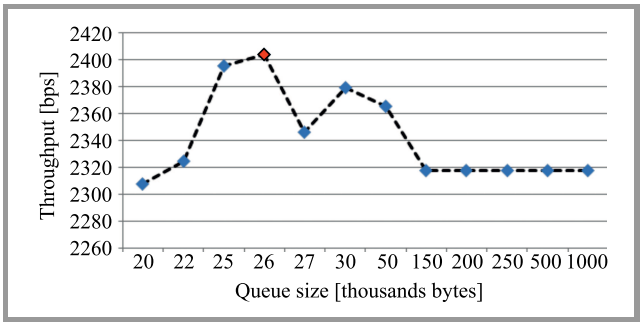

Fig. 12. Performance evaluation at different queue sizes.

shown in Fig. 13. The throughput at different queue lengths shown in Fig. 12 and the throughput for various combinations of queue types and scheduler types at the queue length of 26,000 bytes is considered as the performance parameter. It is observed that the maximum throughput is achieved at the queue size of 26,000 bytes, using a round robin scheduler with a random early detection queue. A more precise analysis of individual parameters, such as packets de-queued, packets dropped, packets queued, average queue length, longest time in queue, peak queue length, average time in queue and total number of packets dropped, is still required to finalize the selection of most the suitable queue and scheduler types.

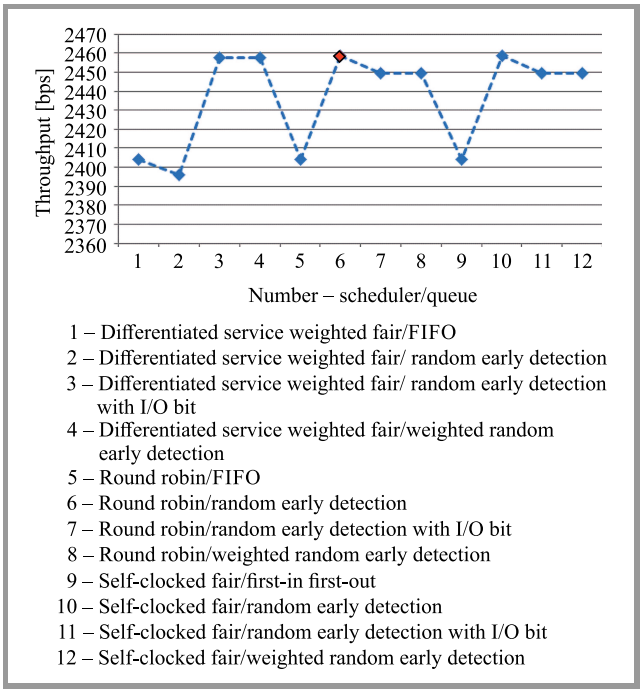

Fig. 13. Performance evaluation using different schedulers and queues of 26,000 bytes.

An extensive analysis of results, presented in Figs. 14 to 20, indicates that differentiated service-weighted fair schedulers, paired with the FIFO queue, at the queue size of 26,000 bytes, are chosen for the final simulation.

The number of packets de-queued at different queue sizes is presented in Fig. 14. The number of de-queued packets is the lowest at the queue size of 20,000 bytes. With the further increase in queue size it also decreases at the size of 26,000 bytes. Finally, above the queue size of 200,000 bytes, it becomes stable at the same value that is achieved at the size of 26,000 bytes.

The number of packets dropped at different queue sizes is shown in Fig. 15. It can be seen that at a low queue size the number of packets dropped is higher and decreases with the increase in queue size. At the size of 500,000 and 1,000,000 bytes, its value becomes almost negligible.

Fig. 14. Number of packets de-queued.

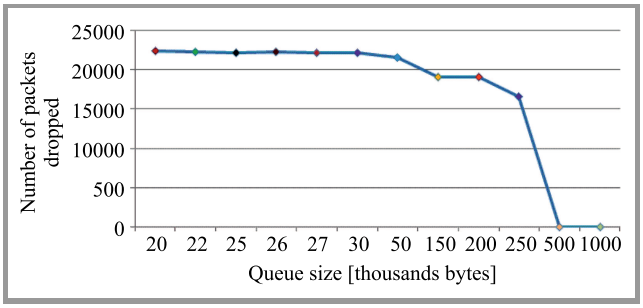

Fig. 15. Number of packets dropped.

The number of packets queued is directly proportional to the queue size (Fig. 16). With an increase in queue size, the capacity to queue the data packets also increases, which prevents the packets from being dropped.

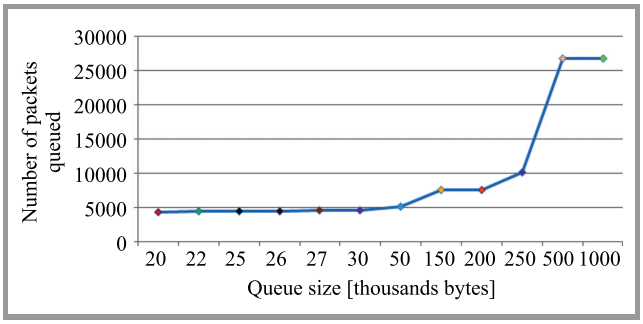

Fig. 16. Number of packets queued.

The average queue length for different combinations of different types of queues and schedulers at the queue size of 26,000 bytes is shown in Fig. 17. It is largest for the FIFO and smallest for the RED queue.

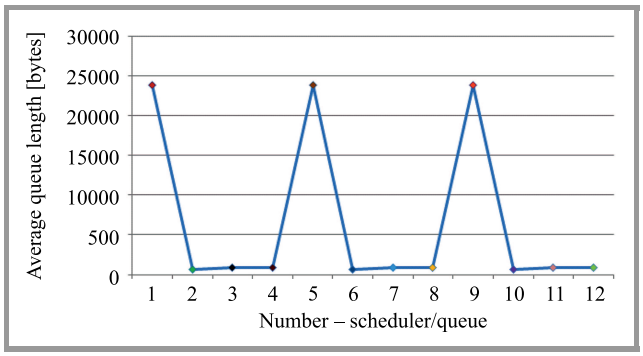

Fig. 17. The average queue length. The *x* axis legend is from Fig. 13.

The longest time spent by the data packets in the queues is presented in Fig. 18. The longest time taken is in the FIFO queue, and the shortest time is in the RED queue.

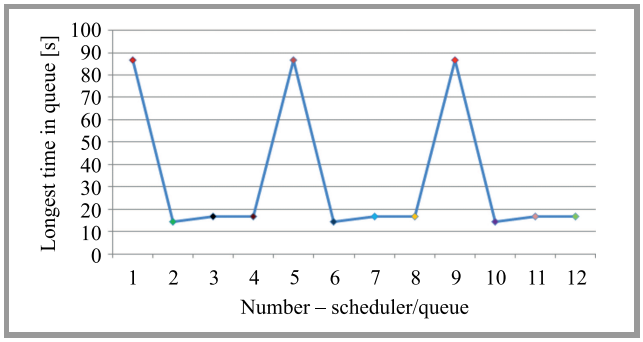

Fig. 18. The longest time in queue. The *x* axis legend is from Fig. 13.

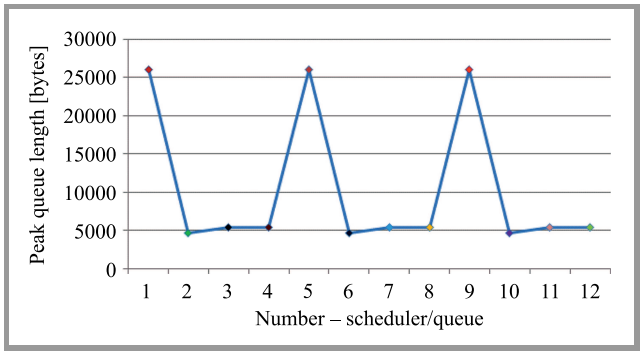

Fig. 19. The peak queue length – see legend box in Fig. 13.

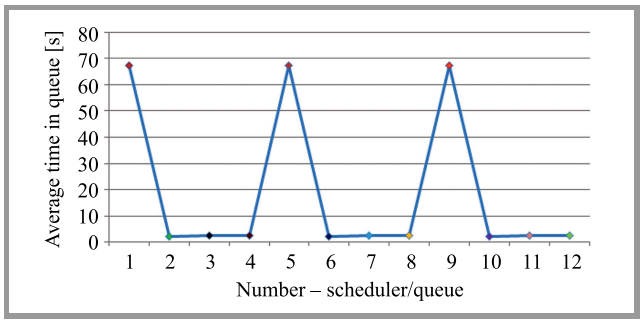

Fig. 20. The average time in queue – see legend box in Fig. 13.

The peak queue length is evaluated and compared in Fig. 19. The highest value is for the FIFO, which gets reduced for WRED, reduced even further for the RIO, and the minimum value is achieved for the RED queue.
The average time in the queue is evaluated and presented in Fig. 20. The longest time in the queue is for the FIFO, and the shortest time is for the RED queue.

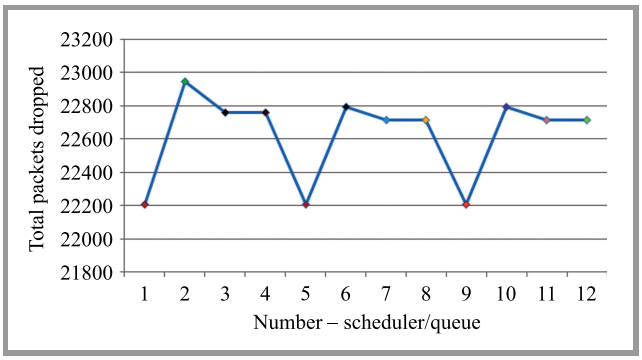

Fig. 21. The number of total packets dropped – to decipher *x* axis values, see legend box in Fig. 13.

The number of total packets dropped in the combination of different queues and schedulers is analyzed in Fig. 21. The results show that the combination of different schedulers with the FIFO queue has the lowest number of packets dropped, equaling to 22,200 packets.

8. The Comparison of IEEE 802.15.4 and AR-MAC

The evaluation and analysis of different parameters, such as packets queued, packets de-queued, average queue length, longest time in a queue, peak queue length and average time in a queue, results in the determination of the most appropriate combination. Implementation of the Differentiated Service Weighted Fair scheduler with the FIFO queue at the queue size of 26,000 bytes is considered for the simulations with the proposed queuing model and the AR-MAC protocol.
Evaluation and comparison of the total number of packets dropped for IEEE 802.15.4 and AR-MAC protocol is shown

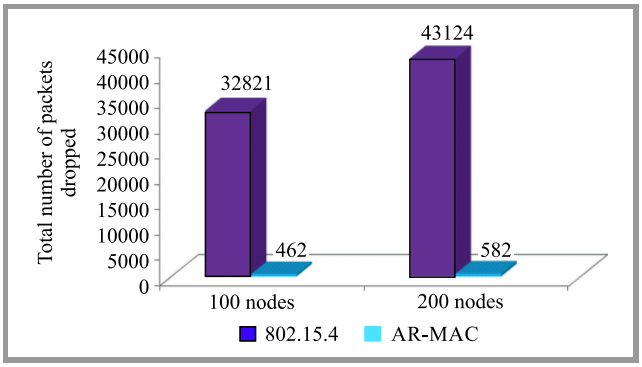

Fig. 22. The comparison of total number of packets dropped.

in Fig. 22. The result shows a significant reduction in the packets dropped when using the AR-MAC protocol in the networks of 100 and 200 nodes.

The average queue length is calculated and compared in Fig. 23. Utilization of the AR-MAC protocol significantly reduces the average queue length for the networks of 100 and 200 nodes. It will decrease the end-to-end delay, as the waiting time of the data packets is significantly reduced.

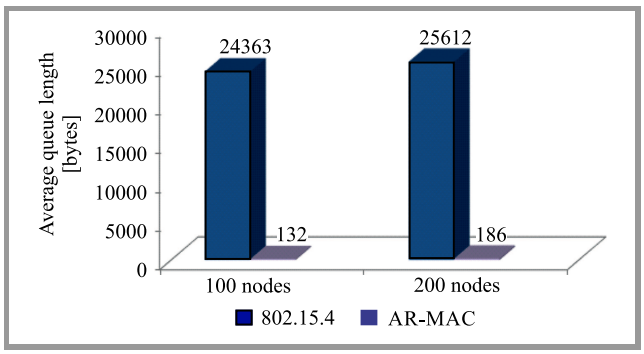

Fig. 23. The comparison of average queue length.

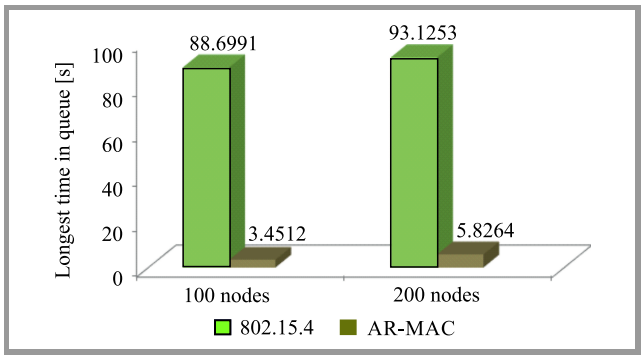

Fig. 24. The comparison of longest time in queue.

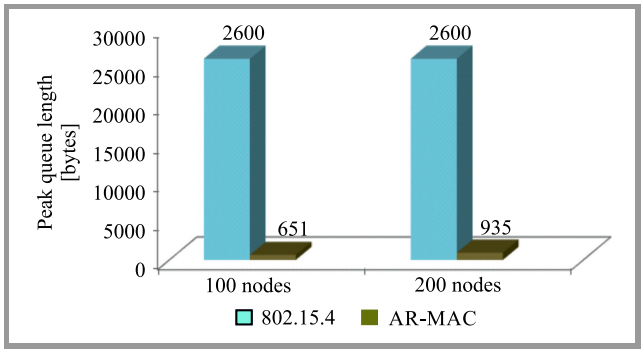

Fig. 25. The comparison of peak queue length.

The longest time in the queue is evaluated and compared in Fig. 24. The performance of AR-MAC is still better than that of the IEEE 802.15.4 protocol for the networks of 100 and 200 nodes.

The peak queue length for IEEE 802.15.4 and AR-MAC for the networks of 100 and 200 nodes is compared in Fig. 25. It is negligible for the AR-MAC protocol.

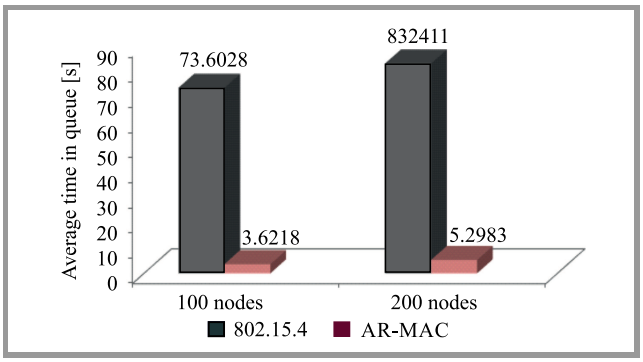

Fig. 26. The comparison of average time in queue.

The average time in the queue, which contributes to the end-to-end delay, is negligible for the AR-MAC protocol, as shown in Fig. 26. It has been verified that the AR-MAC protocol is appropriate to optimize the performance of schedulers and queues in the sink and sensor nodes.

9. Conclusion and Future Work

In this paper, we have implemented the AR-MAC protocol with effective path and optimum throughput path determination algorithms after the determination of queue size, queue type and scheduler type. The aim of optimizing the performance of queues has been successfully achieved along with the improvement in other performance metrics. Simulation results validate the proposed protocol under various application requirements and network conditions. Implementation of efficient scheduling algorithms to enhance the output of the network layer, which leads to optimized performance of the MAC layer, is a very effective technique. The aim is to prevent the overflow of queues – resulting in congestion and formation of bottlenecks – to eliminate the loss of data packets carrying important information.

In the future, we intend to develop a queuing model for the static sink UWSNs.

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] I. F. Akyildiz, D. Pompili, and T. Melodia, "State of the art in protocol research in underwater acoustic sensor networks", *ACM Mob. Comput. Commun. Rev.*, vol. 11, no. 4, pp. 11–22, 2007.

[3] L. Weifa, L. Jun, and X. Xu, "Prolonging network lifetime via a controlled mobile sink in wireless sensor networks", in *Proc. of IEEE Global Telecommun. Conf. GLOBECOM 2010*, Miami, FL, USA, 2010 (doi: 10.1109/GLOCOM.2010.5683095).

[4] J. K. Jalaja and J. Lillykutty, "Delay and lifetime performance of underwater wireless sensor networks with mobile element based data collection", *Int. J. of Distrib. Sensor Netw.*, vol. 8, no. 1, pp. 1–22, 2015.

[5] S. Sendra, J. Lloret, J. J. P. C. Rodrigues, and J. M. Aguiar, "Underwater wireless communications in freshwater at 2.4 GHz", *IEEE Commun. Lett.*, vol. 17, no. 9, pp. 1794–1797, 2013.

[6] E. Ali, E. Abdelrahman, A. Majed, and E. Khaled, "Underwater wireless sensor network communication using electromagnetic waves at resonance frequency 2.4 GHz", in *Proc. of of the ACM 15th Commun. and Netw. Simulation Symp. CNS'12*, Orlando, FL, USA, 2012.

[7] B. Latré, P. De Mil, I. Moerman, B. Dhoedt, and P. Demeester, "Throughput and delay analysis of unslotted IEEE 802.15.4", *J. of Networks*, vol. 1, no. 1, pp. 20–28, 2006.

[8] K. Yu, M. Gidlind, J. Akerberg, and M. Bjorkman, "Low jitter scheduling for industrial wireless sensor and actuator networks", in *Proc. of the 39th Ann. Conf. of the IEEE Industrial Electron. Soc. IECON 2013*, Vienna, Austria, 2013, pp. 5594–5599.

[9] M. Aoun and A. Argyriou, "Queueing model and optimization of packet dropping in real-time wireless sensor networks", in *Proc. of the IEEE Global Commun. Conf. GLOBECOM 2012 – Commun. QoS, Reliab. and Model. Symp.*, Anaheim, CA, USA, 2012, pp. 1687–1691.

[10] J. Liu *et al.*, "DA-Sync: A Doppler-Assisted time synchronization scheme for mobile underwater sensor networks", *IEEE Trans. on Mob. Comput.*, vol. 13, no. 3, pp. 582–595, 2014.

[11] H. Yang and B. Sikdar, "A mobility based architecture for underwater acoustic sensor networks", in *Proc. IEEE Global Telecommun. Conf. GLOBECOM 2008*, New Orleans, LA, USA, 2008.

[12] N. Tuan Le, S. Woong Choi, and Y. Min Jang, "Approximate queuing analysis for IEEE 802.15.4 sensor network", in *Proc. 2nd Int. Conf. on Ubiquit. and Future Netw. ICUFN 2010*, Jeju Island, Korea (South), 2010, pp. 193–198.

[13] S. Charalambos, V. Vasos, and P. Aristodemos, "Estimating queue formation rate in Wireless Sensor Networks using a fluid dynamic model", in *Proc. 20th IEEE Symp. on Comp. and Commun. ISCC 2015*, Larnaca, Cyprus, 2015, pp. 544–548.

[14] N. Van Mao and V. Que Son, "Applying queuing theory to evaluate performance of cluster wireless sensor networks", in *Proc. Int. Conf. on Adv. Technol. for Commun. ATC 2015*, Ho Chi Minh City, Vietnam, 2015.

[15] J.-F. Ke, W.-J. Chen, and D.-C. Huang, "Life extend approach based on priority Queue N strategy for wireless sensor network", in *Proc. 11th In. Conf. on Heterogen. Netw. for Qual., Reliabil., Secur. and Robustness QSHINE 2015*, Taipei, Taiwan, 2015.

[16] S. Vanithamani and N. Mahendran, "Performance analysis of queue based scheduling schemes in wireless sensor networks", in *Int. Conf. on Electron. and Commun. Syst. ICECS 2014*, Coimbatore, India, 2014.

[17] H. Byun and J. Yu, "Adaptive duty cycle control with queue management in wireless sensor networks", *IEEE Trans. on Mob. Comput.*, vol. 12, no. 6, pp. 1214–1224, 2013.

[18] E. G. W. Peters, D. E. Quevedo, and M. Fu, "Controller and scheduler codesign for feedback control over IEEE 802.15.4 networks", *IEEE Trans. on Control Syst. Technol.*, vol. 1, no. 99, pp. 2459–2464, 2016.

[19] B. Pati, J. L. Sarkar, C. R. Panigrahi, and R. K. Verma, "CQS: A Conflict-free query scheduling approach in wireless sensor networks", in *Proc. 3rd Int. Conf. on Recent Adv. in Inform. Technol. RAIT 2016*, Dhanbad, India, 2016, pp. 13–18.

[20] H. Zhang *et al.*, "Scheduling with predictable link reliability for wireless networked control", in *Proc. 23rd Int. Symp. on Quality of Service IWQoS 2015*, Portland, OR, USA, 2015.

[21] B. Pati, J. L. Sarkar, C. R. Panigrahi, and M. Tiwary, "ARTQS: An advanced real-time query scheduling approach in wireless sensor networks", in *Proc. Int. Conf. on Green Comput. and Internet of Things ICGCIoT 2015*, Delhi, India, 2015, pp. 219–224.

[22] E. G. W. Peters, D. E. Quevedo, and M. Fu, "Co-design for control and scheduling over wireless industrial control networks", in *Proc. 54th IEEE Conf. on Decision and Control CDC 2015*, Osaka, Japan, 2015, pp. 2016–2030.

[23] M. H. Shahid and Sh. Masud, "Improved low power scheduler for OSS-7: An open source DASH7 stack", in *Proc. IEEE Int. Conf. on Electron., Circuits, and Syst. ICECS 2015*, Cairo, Egypt, 2015, pp. 645–648.

[24] M. Chovanec and P. Sarafin, "Real-time schedule for mobile robotics and WSN applications", in *Proc. Federated Conf. on Comp. Sci. and Inform. Syst. FedCSIS 2015*, Lodz, Poland, pp. 1199–1202.

[25] M. A. Jamshed *et al.*, "An energy efficient priority based wireless multimedia sensor node dynamic scheduler", in *Proc. 12th Int. Conf. on High-capacity Opt. Netw. and Enabling/Emerging Technol. HONET 2015*, Islamabad, Pakistan, 2015, pp. 147–150.

Vikas Raina completed his B.E. in Electronics and Communication Engineering and M.Tech. in microelectronics and VLSI Design from Rajiv Gandhi Technical University, Bhopal, India. He is pursuing his Ph.D. in Underwater Wireless Sensor Networks from Mody University of Science and Technology, Lakshmangarh, Rajasthan. Mr. Vikas Raina has more than 11 publications in national and international journals and conferences.
E-mail: vikasraina.cet@modyuniversity.ac.in
Department of ECE, CET
Mody University of Science and Technology
Lakshmangarh, India

Manish Kumar Jha completed his B.E. in Electronics Engineering from Bangalore University and Ph.D. from Birla Institute of Technology, Mesra, Ranchi. He has more than 23 years of extensive experience in teaching and industry out of which he has spent more than 11 years as an academic with Birla Institute of Technology, Mesra Ranchi. Presently he is the Director of ABES Engineering College, Ghaziabad, India. Before joining ABES EC, he was working at Mody University of Science and Technology, Lakshmangarh, as Professor and Associate Dean. He is a very able administrator and a learned scholar. One Ph.D. has been awarded and several scholars are pursuing doctoral research under his guidance. He has guided more than 20 M.Tech. projects. Prof. Manish Kumar Jha has played many other roles over two decades of his career. A project concerned with design and simulation of a new energy efficient ML-MAC wireless sensor protocol, funded by ISRO, was successfully completed under his leadership and guidance. Prof. Manish Kumar Jha has more than 40 publications in national and international journals and conferences.
E-mail: manishkjhaa@gmail.com
ABES Engineering College
Ghaziabad, India

Partha Pratim Bhattacharya has 20 years of experience in teaching and research. He served many reputed educational institutes in India in various positions. At present, he is working as Professor in the 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. He has published more than 100 papers in reputed journals and conferences. His present research interests include mobile cellular communication, wireless sensor network and cognitive radio. He delivered several invited lectures and made many expert appearances on various television channels and All India Radio. Dr. Bhattacharya is a member of The Institution of Electronics and Telecommunication Engineers, India and The Institution of Engineers, India. He received, in 2005, the Young Scientist Award from the International Union of Radio Science. He is working as a reviewer in many reputed journals, such as 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. His name has been included in Marquis Who's Who in the World.

E-mail: hereispartha@gmail.com
Department of ECE, CET
Mody University of Science and Technology
Lakshmangarh – 332311, Rajasthan, India

Paper

Availability Analysis of Different PON Models

Katarina Radoš[1] and Ivan Radoš[2]

[1] *University of Split, Split, Croatia*
[2] *HT d.o.o. Mostar, Mostar, Bosnia and Herzegovina*

https://doi.org/10.26636/jtit.2017.116517

Abstract—Due to the increasing demands of individual users, Passive Optical Networks (PON) are a promising technology for future broadband access networks. Reliable access to network services is a very important feature, so the availability of the connection is becoming one of the most important requirements. Failure of the optical fiber occurring between the Optical Line Terminal (OLT) and the passive optical splitter can cause the services becoming unavailable for a large number of users, so it is necessary to prevent such an occurrence by providing backup resources – in this case a spare optical fiber. When constructing the spare path (protection fiber), it is important to keep in mind that, if possible, the working and the spare fiber should not be positioned within the same cable. Failure of the optical fiber between the passive splitter and the individual user also means a loss of service for that user. In that case, protection may be offered by adding spare fibers. The question is, however, whether such a solution is cost-effective, as it would lead to a significant cost increase in the construction of the access network. This paper presents the availability analysis conducted for different PON models.

Keywords—availability, failure, Passive Optical Network, protection fiber.

1. Introduction

The increasing number of Internet users and the growing bandwidth required by new applications, such as online games, telemedicine and distance learning, are factors that force operators to turn to new architectures. Future access networks must be capable of offering high bandwidth-per-user rates, with their capital and operating costs remaining at the lowest possible level. Therefore, optical access technologies are proposed as the best solution, with the Passive Optical Network (PON) being widely accepted as a promising technology for future broadband optical access networks that may be offered in various implementation scenarios, such as Fiber To The Home (FTTH) and Fiber To The Building (FTTB). One of the most important advantages of the PON network is that the Optical Distribution Network (ODN) consists of passive elements (optical fibers and passive optical splitters) only, which completely eliminates electromagnetic interference, improves reliability and availability of the system, and reduces the cost of maintenance [1]. In addition, the increasing importance of constant access to the Internet for people in everyday life leads to error management as an important challenge in future optical access networks. Individual users, despite requiring minimal interruptions, are unable to afford additional costs to improve reliability. So, operators should try to continue the provision of services in the case of a failures affecting the network, which requires additional costs associated with the deployment of various protection schemes. Availability-related requirements may nevertheless depend on the user's profile. Business users are looking for complete protection, covering end-to-end operations. Connection availability greater than 99.99% (4 nines) must be guaranteed, for example, for some business users, while most residential users may tolerate lower availability rates. However, costs incurred in order to offer additional protection layers may be significant, as such deployment involves doubling, resources such as fiber optics and optical line terminals. It was shown that the availability of 4 nines or more cannot be achieved without a protection path between the Central Office (CO) and end user. So, PON networks should be able to offer end-to-end support for some business users, when it is required [2].

This paper focuses on the implementation of the PON model, which will significantly improve the availability of the connection between the Optical Line Terminal (OLT) and end users. Of course, fiber protection extended from the optical splitter to the end user is not of the same importance as fiber protection between the OLT and the optical splitter, as an interruption between the OLT and the optical splitter will lead to a disconnection to all users that are connected to the OLT. At the same time, an interruption between the optical splitter and the Optical Network Unit (ONU) leads to a disconnection for that specific user only. Protection of any part of the network must be realized within 50 ms, i.e. in accordance with the international standard. The present paper shows also the impact of the fiber length on the availability ratio.

The remaining parts of this paper are organized in the following manner. Section 2 describes the basic PON model. The general PON protection schemes are shown in Section 3. Section 4 describes availability in general. Section 5 shows PON protection and availability models, and Section 6 contains calculations and comments.

2. Basic Model of Passive Optical Network

The basic PON model consists of the following elements: OLT, optical splitter and ONU.

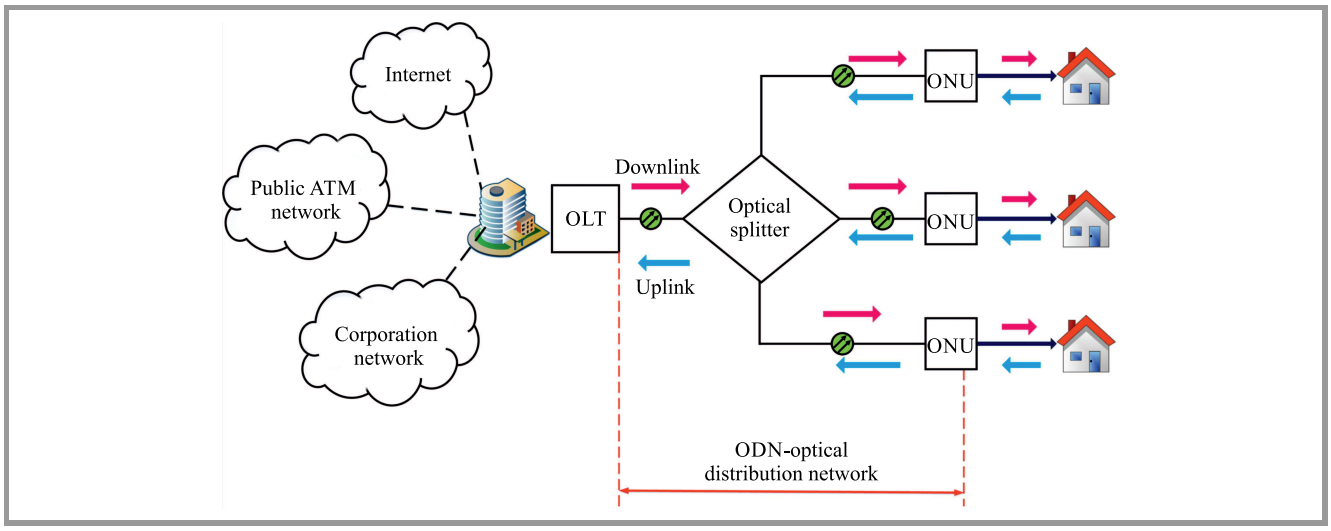

Fig. 1. The basic model of a passive optical network.

Figure 1 shows the basic PON model. Data transfer is always performed between the OLT and different ONUs via an optical splitter that performs multiplexing and demultiplexing of signals. OLT and ONU are active parts of the PON network, because they perform the electro-optical and opto-electrical signal conversion.

The OLT is located in the CO of the operator. It represents the interface between the public network and access network and also controls bidirectional flow of information via the ODN.

The OLT must be able to support transmissions over a distance of 20 km. The role of the OLT, in the downlink direction (from OLT to users), is to transmit data, voice and video form the public network, via a single-mode optical fiber, to ONUs. The OLT performs also the electro-optical conversion in the downlink direction [1], [2].

In the uplink direction (from users to OLT), in turn, the OLT accepts, performs opto-electrical conversion and distributes traffic from the users. Simultaneous transmission of specific types of services via the same optical fiber in the ODN is enabled by using different wavelengths for each direction. For downlink transmissions, PON uses the wavelength of 1490 nm for a combined voice and data traffic, and the wavelength of 1550 nm for video content. As far as uplink is concerned, PON uses the wavelength of 1310 nm for voice and data traffic. Video services are disabled in the uplink direction, because there are no video services which would send video data in this direction [2], [3].

The optical splitter is a passive bidirectional component with one input and multiple outputs. At the entrance of the splitter, the optical power of signals that are sent to the end users is divided evenly at the ratio of 1:N (commonly 1:32), where N is the number of end users connected to the optical splitter. Optical splitter requires no power supply and is maintenance-free.

ONUs are located at the user's premises and they serve as an interface with the end users. ONUs are connected via to the OLT via an optical splitter. The ONU performs opto-electrical conversion in the downlink direction [3], electro-optical conversion in the uplink direction, and packet filtering based on the destination address in the packet's header. The ODN consists of passive optical splitters and optical fibers. ODN consists of passive optical splitters and optical fibers.

3. General PON Protection Schemes

There are many different protection approaches, but they are confined, in principle, to the protection of the optical fiber (cable) and hardware used in optical access networks. Four types of protection are described in [4] and are based on the ITU-T Recommendation G.984.1, as shown in Fig. 2. Figure 2a shows the first type of protection in which the protection (spare) fiber is installed between the OLT and the optical splitter (type 1:N). No switching protocol is required for the OLT/ONU in Fig. 2a, since switching is only applied for optical fibers. The optical switch is located over the working and the protection fiber connected to the 1:N optical splitter.

Under normal conditions, the optical switch is in the bar state and traffic is carried over the working fiber. In case of a failure affecting the working fiber, the optical switch detects the loss of optical signal, passes into the cross state and redirects the traffic onto the protection fiber.

The second type of protection is shown in Fig. 2b. In that scenario, a spare OLT and a spare optical fiber are installed (marked with a dashed, bold, red line). No switching protocol is required since switching is carried out in the OLT only. The spare OLT is in the standby mode. In the case of a failure affecting the working fiber, the optical switch redirects traffic from the working OLT to the protection fiber. If the optical switch detects a loss of signal from the working OLT, it performs switching and redirects traffic from the spare OLT to the working fiber. Fiber interruption between the optical splitter and each ONU will not cause any

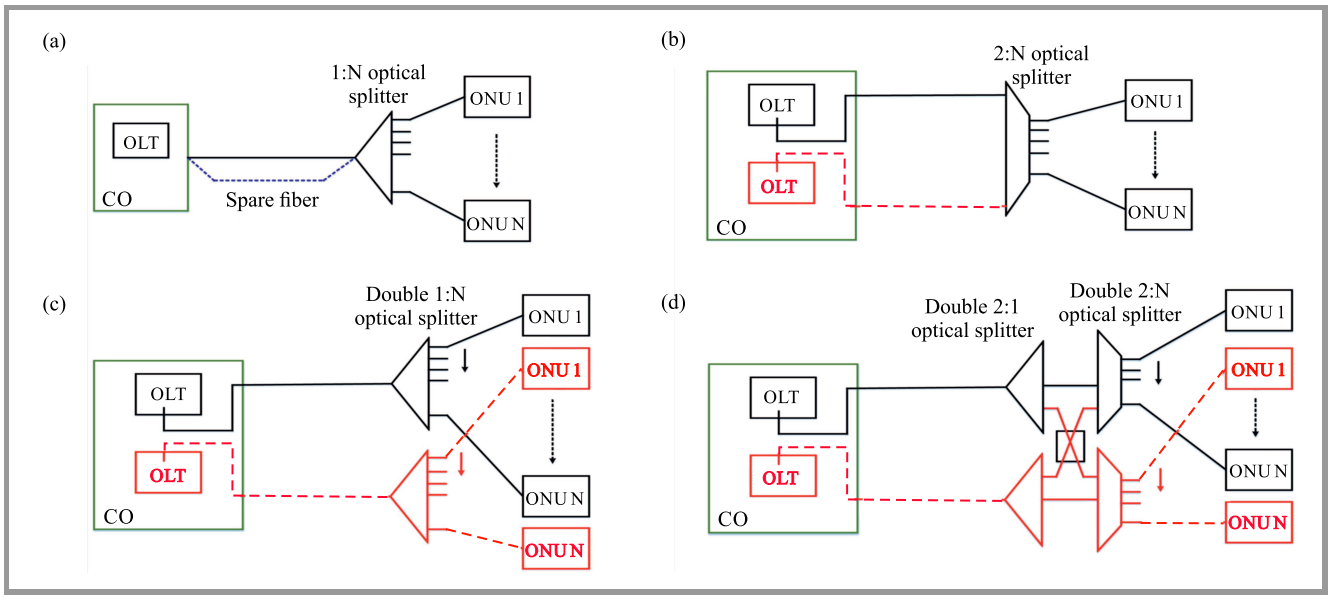

Fig. 2. General protection schemes by ITU-T. (For color pictures, see www.nit.eu/publications/journal-jtit)

switch reaction on the OLT side, so the concerned ONU affected by this kind of optical fiber interruption will not be protected [4].

Figure 2c shows the third protection scheme in which the OLT and ONU the are equipped with redundant modules. For this type of protection, 1:N optical splitters are needed. In this case the spare PON circuit that includes the OLT and the ONU can be activated in the case of a failure on the OLT or ONU side. To activate the spare PON circuit, constant synchronization between working and spare modules is required. With this type of protection, the network is able to survive all individual failures.

A network with complete redundancy, offering protection of equipment and optical fibers, is shown in Fig. 2d. This architecture also uses two 2:N passive optical splitters that connect N users with the geographically separated optical fibers. An additional optical switch with the controlling module is embedded in each ONU. This architecture can survive a failure of the OLT, the ONU's receiver and the passive splitter. It also can survive optical fiber failures between the OLT and the passive splitter, as well as between the passive splitter and the individual ONUs. Being the most reliable architecture, it is also the most expensive variant that requires highly complex management.

4. Availability in General

Availability A is often used to describe the performance of a system, and is defined as the probability of the system's proper operation at some point in time t, under condition that system was working properly at time $t = 0$, that it can enter fault states (maintenance or failure) which are always fixed and followed by the system's return to a working state [5]. The availability of a system within a period of time is also defined as the ratio between time in which the system operates correctly, and the total time elapsed. If the average time to failure (Mean Time To Failure – MTTF) and the average time to repair (Mean Time To Repair – MTTR) are known, availability can be calculated using the following formula:

$$A = \frac{\text{MTTF}}{\text{MTTF} + \text{MTTR}}. \tag{1}$$

Usually, MTTF is not known, so the term in the denominator is defined as the average time between failures (MTBF – Mean Time Between Failures) [5] and the availability can be written as:

$$A = \frac{\text{MTTF}}{\text{MTTB}}. \tag{2}$$

Unavailability U is the probability that is complementary to availability, i.e. $U = 1 - A$ [6], meaning that it is the probability of the system's failure to work properly at some point in time:

$$U = \frac{\text{MTTR}}{\text{MTTF} + \text{MTTR}} \cong \lambda \times \text{MTTR}, \tag{3}$$

where λ is the intensity of failure that is expressed in FIT, 1 FIT = 1 failure/10^9 h.

When reporting on system performance, unavailability is often expressed as the average time of failure (Mean Down Time – MDT) [6]:

$$\text{MDT} = 365 \times 24 \times 60 \times U \ [\text{min/year}]. \tag{4}$$

4.1. Basic Structures of Availability

There are two basic structures of availability: serial (non-redundant) and parallel (redundant) [7]. In the serial structure, the failure of any element of the system equals the failure of the entire system (the system will work only if

each element of the system is operating correctly). Serial structure of a system is shown in Fig. 3.
In light of the above, a path which consists of N elements is correct if each element is in the "correct" state, which may be represented by the following logical equation:

$$\mathrm{P}_{s,\mathrm{N}} = \mathrm{x}_1 \cap \mathrm{x}_2 \cap \ldots \cap \mathrm{x}_\mathrm{N}\,, \tag{5}$$

where $\mathrm{P}_{s,\mathrm{N}}$ is the probability of the proper operation of a serial structure of N elements, $\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_\mathrm{N}$ is the proper operation of each element. The path availability is expressed, in that case, by:

$$A(\mathrm{P}_{s,\mathrm{N}}) = p(\mathrm{x}_1 \cap \mathrm{x}_2 \cap \ldots \cap \mathrm{x}_\mathrm{N})\,. \tag{6}$$

If it is assumed that the failures of elements are independent, i.e., the failure of one element does not cause the failure of other elements, availability can be calculated in the following manner:

$$\begin{aligned} A(\mathrm{P}_{s,\mathrm{N}}) &= p(\mathrm{x}_1) \times p(\mathrm{x}_2) \times \cdots \times p(\mathrm{x}_N) = \\ &= \mathrm{A}_{x_1} \times \mathrm{A}_{x_2} \times \cdots \times \mathrm{A}_{x_\mathrm{N}}\,. \end{aligned} \tag{7}$$

Figure 4 shows the parallel system structure in which the failure of any element does not mean an interruption in communication (the system will work even if some of the elements are defective).
In most cases the structure comprises two elements, so the probability of proper operation of such a structure can be represented by [7]:

$$\mathrm{P}_{p,2} = \mathrm{x}_1 \cup \mathrm{x}_2\,. \tag{8}$$

In that case, path availability is

$$A(\mathrm{P}_{p,2}) = p(\mathrm{x}_1 \cup \mathrm{x}_2)\,. \tag{9}$$

Assuming that the failures of elements are mutually independent, availability can be calculated as follows:

$$\begin{aligned} A(\mathrm{P}_{p,2}) &= p(\mathrm{x}_1) + p(\mathrm{x}_2) - p(\mathrm{x}_1) \times p(\mathrm{x}_2) = \\ &= \mathrm{A}_{x_1} + \mathrm{A}_{x_2} - \mathrm{A}_{x_1} \times \mathrm{A}_{x_2}\,. \end{aligned} \tag{10}$$

5. PON Protection and Availability Models

In this study the standard PON models (i.e., the basic model and the protection model) defined by ITU-T are considered, and serve as a basis for availability calculations [4], [8], [10]. New models that are upgraded based on the ITU-T models are described as well. Models defined by ITU-T are shown in Figs. 5–8, and models worked out based on the ITU-T models are presented in Figs. 9–10. Distance

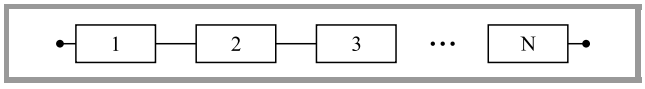

Fig. 3. Appearance of the serial system structure.

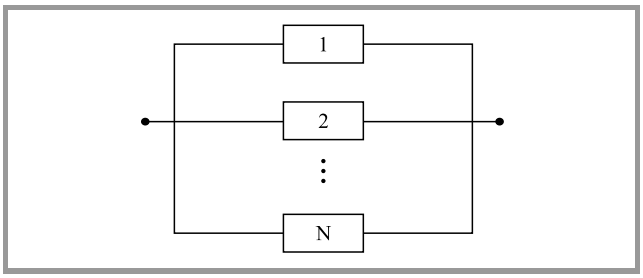

Fig. 4. Appearance of the parallel system structure.

between the OLT and the ONU equals, in all models, 20 km.
Figure 5 shows the basic PON model without protection, which consists of the OLT, FF (feeder fiber), the 1:N splitter, DF (distribution fiber) and the ONU.

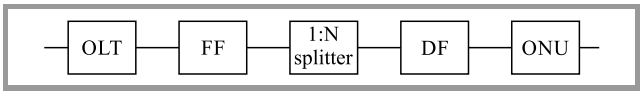

Fig. 5. Basic PON model.

The PON model with feeder fiber protection (one FF block is the working component and the other FF block the is protection component) is shown in Fig. 6. The system will function if there is at least one path that connects the start to the end and does not pass through a failed component.

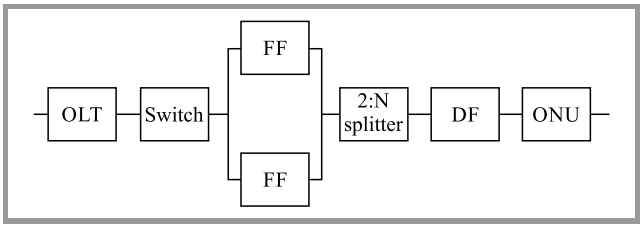

Fig. 6. Type 1 – based on scheme shown in Fig. 2a.

Figure 7 shows a redundant PON model which consists of two parallel structures. The first structure is made of the OLT, FF and the 1:2 splitter, and the other structure consists of the 2:N splitter, DF, and the ONU.

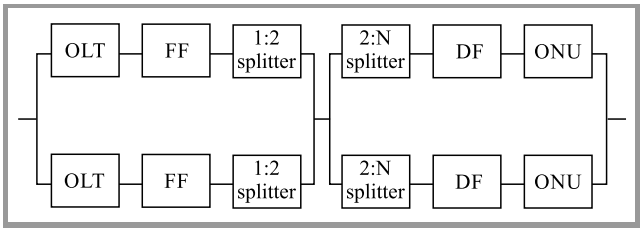

Fig. 7. Type 2 – based on Fig. 2d.

The PON model with a redundant OLT, FF and 1:2 splitter is shown in Fig. 8.
Based on previous 4 models, two new models that involve optical fiber protection leading to individual users (DF),

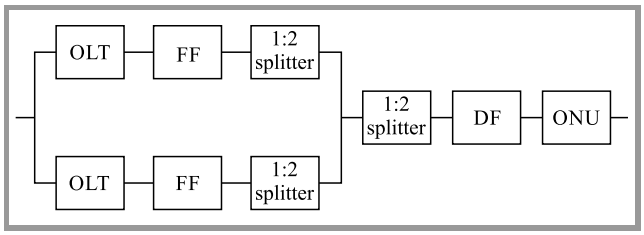

Fig. 8. Type 3 – based on Fig. 2b.

or ONU protection, are introduced. The reason for which the new models have been developed is that they are still concerned with the individual user, so the question remains whether it is profitable to install the protection as shown in type 2. The key components (OLT, FF, 1:2, 2:N) are protected because the failure of any of them disables the provision of the service to all users (ONUs). The model shown in Fig. 9 consists of a parallel structure made up of the OLT, FF and the 1:2 splitter, as well as a parallel structure made of the 2:N splitter, DF, and the ONU.

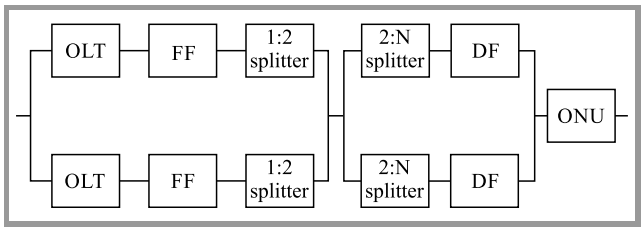

Fig. 9. Type 4.

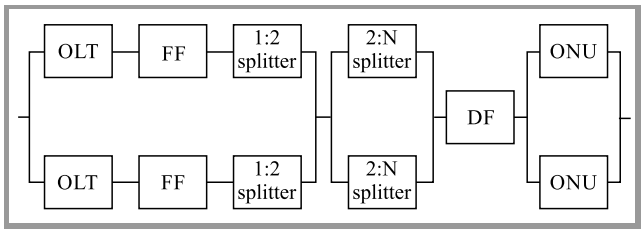

Fig. 10. Type 5.

Figure 10 shows the PON model in which the OLT, FF and the 1:2 splitter make up a parallel structure. The 2:N splitter and the ONU are redundant and separated.

6. Calculations and Comments

In this paper three FTTH deployment cases are analyzed [8]:

- Low customer density case for sparsely populated areas. In this case, FF is 15 km and DF is 5 km long;
- High customer density case for densely populated areas. FF is 18 km and DF is 2 km long;
- Extra high customer density case for very densely populated areas (large apartment blocks, e.g. in China, Russia, Poland, etc.). FF is 19.7 km and DF is 0.3 km long.

Firstly, the low customer density case will be explained. As already mentioned, FF is 15 km and DF is 5 km long in this case. Table 2 shows low customer density case for PON models. The parameters that are calculated for each model include total unavailability, total availability and total MDT. Unavailability rates for each component in the PON network are shown in Table 1 and are taken from [8], where N represents the number of ONUs.

Table 1
Unavailability of the components

Component	Unavailability
OLT (TDM PON)	$5.12 \cdot 10^{-7}$
ONU (TDM PON)	$1.54 \cdot 10^{-6}$
1:2 (2:2) splitter	$3.00 \cdot 10^{-7}$
1:N (2:N) splitter	$7.20 \cdot 10^{-7}$
Fiber [per km]	$1.37 \cdot 10^{-5}$
Switch	$1.20 \cdot 10^{-6}$

Before commenting on the results of calculations, it is important to say that the serial availability structure is calculated, for all models, according to Eqs. (7) and (10), where Eq. (7) is related to the serial structure, and Eq. (10) is related to the parallel structure.

Total unavailability for all models is obtained from the expression $U = 1 - A$. Furthermore, total MDT is calculated with the use of Eq. (4). The basic PON model is the ITU-T model without protection and represents the serial availability structure.

As shown in Table 2, total MDT for the basic model is 145.463 minutes per year. Model type 1 has FF protection which represents the parallel structure of availability that is calculated according to Eq. (10). In the case of failure of the working FF, all traffic is switched to the protection FF. Total MDT is better than in the case of the basic model and equals 38.113 min/year.

In type 2, all components of the network are redundant and it is a very well protected network, which MDT being very low (0.024 min/year). As shown in Table 2, MDT for type 2 is 5832 times lower than for the basic model, and 1528 times lower than for type 1.

Type 3 has three redundant components: the OLT, FF and the 1:2 splitter. This network is unavailable for 37.213 min/year. Better availability would be achieved if there was a redundant ONU, but ONU protection is not so important because if interruption of the optical fiber occurs between the OLT and the optical splitter, both the working and the redundant ONU will be deprived of the signal.

All components in type 4 are redundant except ONU. This model is very well protected with small MDT (0.834 min/year).

Type 5 has all redundant components except for DF, and, consequently, has a higher unavailability rate.

For the high customer density case, FF is 18 km and DF is 2 km, as already mentioned. Table 3 shows the values

Table 2
Low customer density case

Model	Total availability	Total unavailability	Total MDT [min/year]
Basic PON	0.999723243	$2.76 \cdot 10^{-4}$	145.463
Type 1	0.999927486	$7.25 \cdot 10^{-5}$	38.113
Type 2	0.999999953	$4.74 \cdot 10^{-8}$	0.024
Type 3	0.999929198	$7.08 \cdot 10^{-5}$	37.213
Type 4	0.999998413	$1.58 \cdot 10^{-6}$	0.834
Type 5	0.999931457	$6.85 \cdot 10^{-5}$	36.025

Table 3
High customer density case for densely populated areas

Model	Total availability	Total unavailability	Total MDT [min/year]
Basic PON	0.999723236	$2.76 \cdot 10^{-4}$	145.467
Type 1	0.999968567	$3.14 \cdot 10^{-5}$	16.521
Type 2	0.999999938	$6.19 \cdot 10^{-8}$	0.032
Type 3	0.999970279	$2.97 \cdot 10^{-5}$	15.621
Type 4	0.999998398	$1.60 \cdot 10^{-6}$	0.842
Type 5	0.999972539	$2.74 \cdot 10^{-5}$	14.433

Table 4
Extra high customer density case for very densely populated areas

Model	Total availability	Total unavailability	Total MDT [min/year]
Basic PON	0.99972323	$2.76 \cdot 10^{-4}$	145.470
Type 1	0.999991845	$8.15 \cdot 10^{-6}$	4.286
Type 2	0.999999927	$7.31 \cdot 10^{-8}$	0.038
Type 3	0.999993557	$6.44 \cdot 10^{-6}$	3.386
Type 4	0.999998387	$1.61 \cdot 10^{-6}$	0.847
Type 5	0.999995817	$4.18 \cdot 10^{-6}$	2.198

of total availability, total unavailability and total MDT for the high customer density case considered for each of the models. As one can see, total MDT for the basic PON is almost equal to that experienced in the low customer density case (see Table 2).

Total MDT for type 1 is 2.3 times lower than in the low customer density case (FF = 15 km), because the longer fiber (FF = 18 km) is protected in the high customer density case. Total MDT decreases because the unprotected distribution fiber is shorter and its unavailability is correspondingly lower. In type 2, total MDT in the high customer density case (DF = 2 km) is a little higher than in the low customer density case (DF = 5 km), because the shorter fiber is protected. Total MDT increases because the unprotected distribution fiber is longer.

Type 3 has approximately the same availability as type 1.

For type 4 the same is valid as for type 2 (the protection of the shorter fiber renders a higher total MDT, meaning that the network has a higher unavailability rate per year). For type 5 availability is approximately the same as for type 3.

Based on all the calculations presented above, one may conclude that type 2 is the best network model in both the low customer density and the high customer density scenarios, because all components are protected. Therefore, it has the lowest MDT, meaning that the network is unavailable for only 0.024 minutes per year in the low customer density case. Despite the best availability ratio, type 2 is expensive to construct and maintain, because all of its components are redundant. Type 4 has very good availability, although it is lower than in type 2. As far as cost-effectiveness is concerned, type 4 is cheaper than type 2 because the ONU is not redundant.

Compared to type 2 and type 4 models, type 5 has a worse availability ratio, but it is the cheapest because no spare fiber is necessary to protect the ONU.

As expected, basic PON model with serial structure has the worst availability because there are no redundant components.

Table 4 shows the values of total availability, total unavailability and total MDT for the extra high customer density

case of each model. In this case, results for basic PON, type 2 and type 4, are roughly equal to those in the second case (Table 3). Total MDT for type 1, type 3 and type 5 is lower than in the second case because the fiber protected is longer (FF = 19.7 km). In this case, DF protection makes no sense because we are dealing with a very densely populated area and relatively short distances (300 m), so the probability of failures is rather low.

7. Conclusions

This paper analyzes the availability of different PON models. Two deployment cases are shown, and it can be concluded, based on those cases, that type 2, with all redundant components, is the model characterized by the best availability and the lowest MDT. This model is also the most expensive scenario, as all its components are redundant. As mentioned above, ONU protection is not so important, because if interruption of the optical fiber occurs between the OLT and the optical splitter, both the working and the redundant ONU will be deprived of the signal. Type 4 does not have ONU protection but still offers very good availability. Type 5, which is considerably cheaper than models 2 and 4, offers availability that is satisfactory for individual users and is also a promising solution for the construction of PON networks.

It can be also concluded that a network in which the longer fiber is protected offers better availability (see type 5 in Tables 2 and 3). Therefore, one may state that if the length of the fiber increases, MDT decreases and vice versa. It is very difficult to obtain a model characterized by very good availability and cost-effectiveness.

References

[1] C. Lam, *Passive Optical Networks: Principles and Practice*. London: Elsevier, 2008.

[2] J. S. Asensi Pla, "Design of passive optical network", Master thesis, Brno University of Technology, May 2011.

[3] M. Kapov and J. Lorincz, *Local and Access Networks*, Split: FESB, 2013.

[4] ITU-T Recommendations: G983.1, 1998, and G984.1, 2003.

[5] L. Schwartz, D. Trstensky, and G. Cepciansky, "Reliability of telecommunications systems", University of Zilina, Slovakia, 2010, p. 193 (in Slovak).

[6] D. R. Shier, *Network Reliability and Algebraic Structures*. New York: Oxford University Press, 1991.

[7] I. Radoš, "Availability of self-healing ring networks based on wavelength division multiplexing", Ph.D. thesis, University of Split, 2012.

[8] L. Wosinska, J. Chen, and C. P. Larsen, "Fiber access networks: Reliability analysis and Swedish broadband market", *IEICE Trans. Commun.*, vol. E92-B, no. 10, pp. 3006–3013, 2009.

[9] W. A. Imtiaz, M. Waqas, P. Mehar, and Y. Khan, "Self-healing hybrid protection architecture for passive optical networks", *Int. J. of Adv. Comp. Sci. and Appl. (IJACSA)*, vol. 6, no. 8, 2015 (doi: 10.14569/IJCSA.2015.060819).

[10] ITU-T G-series Recommendations – Supplement 51, 2016.

Katarina Radoš received her M.Sc. in Communication and Information Technology from the University of Split, Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture (2016). Currently, she is working at the Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture, Split, Croatia. Her scientific research focuses on optical communications, optical access networks (PON), reliability and availability of optical networks.

E-mail: krados@fesb.hr
University of Split
Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture
Kroz Smrdečac 47
21 000 Split, Croatia

Ivan Radoš received the B.Sc. degree in Electrical Engineering from the University of Split, in 1983, M.Sc. degree from the University of Zagreb, in 2000 and Ph.D. from the University of Split, Croatia in 2012. In 1985 he joined the PTT (Post and Telecommunication) Tomislavgrad. Since 1992 he works at Department of Transmission Systems of the HT Mostar (Croatian Telecommunication). His research interests include: digital transmission systems, optical systems and networks, availability and reliability of telecommunication systems. He has published 10 papers in international conference proceedings, 5 papers in domestic journals and 8 papers in international journals.

E-mail: ivan.rados@hteronet.ba
HT d.o.o. Mostar
Kneza Branimira bb
88000 Mostar, Bosnia and Herzegovina

Paper

Relay-assisted WDM-FSO System: A Better Solution for Communication under Rain and Haze Weather Conditions

Navneet Dayal, Preeti Singh, and Pardeep Kaur

Department of Electronics & Communication Engineering, UIET, Chandigarh, India

https://doi.org/10.26636/jtit.2017.113917

Abstract— Among various conventional wireless communication systems, Free Space Optics (FSO) may be distinguished as well, which provides a good level security, high data rates, an enormous bandwidth and negligible susceptibility to electromagnetic interference. The main problem arises with unfavorable weather conditions, which affect the system's parameters and require an aversion to other paths. The weather conditions discussed are rain and haze. Elimination of their adverse effects is essential for establishing a better communication system. The routing path is a major issue, as its decreased parameters lead to the diversion of the transmitted signal or to its total loss. In this article, increased system path length and proficiency levels are demonstrated by using relays, which are assisted by Wavelength Division Multiplexing (WDM).

Keywords— Free Space Optics, Wavelength Division Multiplexing.

1. Introduction

The boost in the communication field is observed in the 700–1800 nm range of the spectrum. These high frequency waves are not only untraceable due to their narrow beam footprint, but also offer sufficient bandwidth. The basic communication system consists of a transmitter (LED or laser), a communication medium (air or vacuum), and a receiver which can be a photodiode or a PIN diode. The necessary condition for the Free Space Optics (FSO) system to work is Line of Sight (LOS) between transceivers [1]–[3]. If LOS is available, FSO outperforms all other systems by offering numerous benefits, including a lower bit error rate and lower cost, a higher signal-to-noise ratio, greater security excessive bandwidth, a sufficient data rate of up to tens of Gb/s, no interference with electromagnetic waves, a minimum Doppler spread etc. The maximum information capacity of any communication system is given by the Shannon Hartley's theorem.

$$C = B\log_2\left(1+\frac{S}{N}\right). \tag{1}$$

It presents the relation between bandwidth B (in Hz) and information capacity C in bits, where S is the average signal power, with N being the average noise power.

One can notice that capacity is directly dependent on bandwidth. Thus, it is correct to say that FSO operates as the first and last mile access gateway to communication networks. Having such huge advantages, FSO has made its way to all telecom-related fields and is used in a multitude of applications, such as disaster recovery, when all the connections are disrupted, as well as various defense operations, aircraft links, ground-to-airplane links, backhaul connectivity, and enterprise links. All of these applications rely on FSO. In spite of offering all well-known benefits and efficiency-related advantages, FSO is still suffering from a lot of drawbacks [4]. Due to turbulence resulting from a varying refractive index present in the different layers of the atmosphere, there is a loss of signal strength during propagation. These changes depend on the behavior of atmospheric molecules during any given weather situation. Therefore, as weather conditions change, the effect on signal changes accordingly.

Signal attenuation during heavy rain, haze, fog, storm, etc. is high compared to other conditions, so efforts should focus on making the system better. Such conditions cause laser light to either deviate or reflect, depending on the type of scattering effect offered by the molecules present and the wavelength transmitted. The loss that occurs in the path of the communication medium is due to many factors, such as scattering, absorption, pointing errors, scintillation caused by temperature and pressure changes resulting from the packet heating the air molecules. 1550 nm is referred to as the minimum absorption window. Performance of the system is judged using the Bit Error Rate

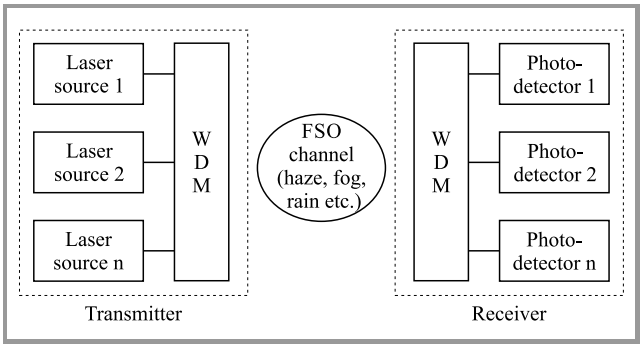

Fig. 1. Example of WDM-FSO system.

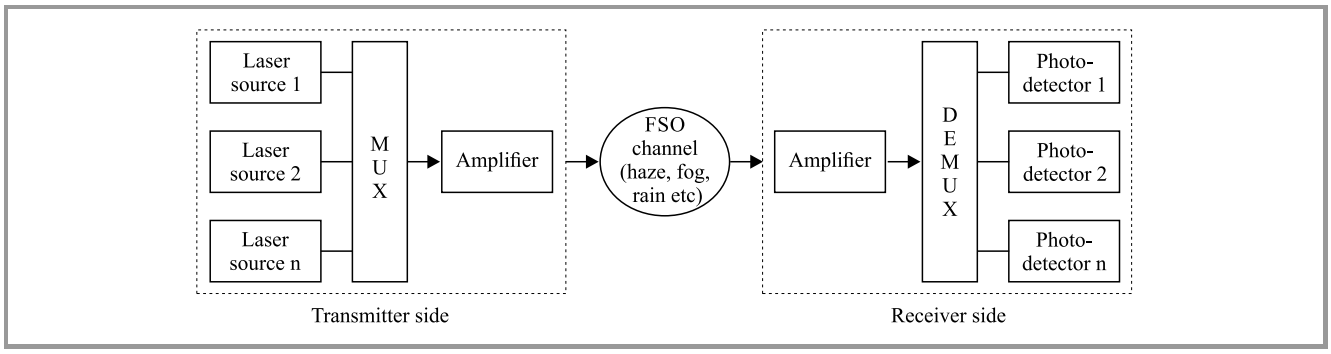

Fig. 2. An amplifier-assisted network shown along with the WDM system.

(BER) and the Q factor which depends on the transmitted power of the laser. The attenuation of light, as per the Beer-Lambert law, is:

$$T(R) = \frac{P(D)}{P(S)} = \mathrm{e}^{-AR}, \tag{2}$$

where $T(R)$ is the transmittance at a distance R, A is atmospheric attenuation, $P(D)$ is the power received at the receiver and $P(S)$ is the power at the transmitter.

WDM is a technique (Fig. 1) which is used to enhance the system's capacity, as number of wavelengths are accommodated within a single channel [5]–[6]. But due to the threat of non-linearity it possesses, the use of WDM systems requires different components, such as EDFA amplifiers, to ensure a good transmission range and SNR. EDFAs boost the signal and thus increase the SNR at the output [7]. The typical configuration is shown in Fig. 2. It describes the use of optical amplifiers at two positions, rather than using one for each user before WDM, thus decreasing the number of devices and ensuring good range and quality.

A new concept of relays has now been introduced to WDM-FSO. This combination is a good step in the direction to achieve optimum communication. The transmitter and the receiver act as the source and destination of the FSO network, while the air is the medium. It is there where relays are incorporated, as shown in Fig. 3. A relay is different than an optocoupler and behaves as a regenerator in the receiving circuit. Those devices which are activated by the incoming signal and are used to connect or disconnect a part of the circuit within the system.

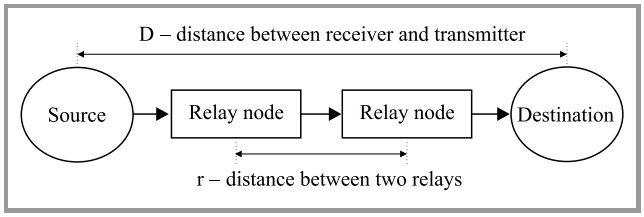

Fig. 3. Relay-assisted network.

Relays ensure complete isolation of the system from the external noise, increasing operation speed [8]–[9]. In optics, this isolation is supplied with a modulated laser. Relays allow to setup a virtual system between the transmitter and the receiver, so that the signal is received even if there is no LOS. Thus, it is referred to as a cooperative diversity system. Such a concept was first proposed by Acompora and Krishnamurthy in 1999, showing that the area coverage can be increased and outage probability can be minimized.

2. System Analysis

Rain is one of the most degrading factors disrupting FSO communication. The disparity in a rain droplet leads to multiple reflections and refractions. The specific attenuation due to rain is $\alpha = aRb$ (dB/km), where a, b are frequency dependent factors and R is the rainfall rate (mm/h) [10]. Rain drops have the size exceeding the wavelength of light signal, thus non-selective scattering predominates, but Mie scattering is still utilized to find the attenuation values. The expression for rain attenuation is: $\alpha_{\text{specific}} = 1.076R^{0.67}$ (dB/km).

The transmission of optical signal is also disrupted by attenuation caused by CO_2 and other gases present in the atmosphere. The attenuation value is given by Kim and Kruse:

$$\sigma = \frac{3.91}{V}\left(\frac{\lambda}{550\,\text{nm}}\right)^{-q}, \tag{3}$$

where V is the visibility (km), λ is the wavelength (nm) and q is the size of distribution of scattering particles (nm). The attenuation in Eq. 3 is the function of visibility only, which determines the defined particle size [11]. The values to be used in the above equation are given in Table 1.

Table 1
Values to be used in Eq. 3

V [km]	q [nm]	Visibility type
Over 50	1.6	High
6 ... 50	1.3	Average
1 ... 6	$0.16V + 0.34$	Haze
0.5 ... 1	$V - 0.5$	Mist
Below 0.5	0	Fog

2.1. Proposed System

The proposed system uses three components: transmitter, transmission medium and receiver. The transmitter has a pseudo-random bit sequence generator, continuous wave laser, NRZ pulse generator and Mach-Zehnder modulator. The medium is air using the FSO channel, and the receiver has a PIN detector and a Gaussian low pass filter. The relay nodes are placed in between the channel at the reception side, to enhance the system characteristics. The performance is measured using the BER analyzer and an optical power meter. The noise figure is maintained at 4 dB and the geometrical losses are considered. The system is analyzed by Optisystem 11.0 to have the lowest BER at 1550 nm for the 2.5 Gb/s data rate.

Table 2
FSO link optimized parameters

Characteristics	Values
Attenuation level	Haze, 2.37 dB/km Light rain, 3 dB/km Medium rain, 9 dB/km Heavy rain, 20 dB/km
Amplifier gain	30 dB
Laser power	30 dBm
Data rate	2.5 Gb/s
Channel spacing	0.8 nm
Aperture size	20 cm

The system is studied as a three-channel WDM system (the number of channels depends on the number of users, so it can be extended accordingly), optimized for different values of amplifier gain and power transmitted by taking into account the existing system, i.e. WDM with the use of optical amplifiers [7]. The beam divergence of the system is 2.5 mrad and the remaining optimized parameters are shown in Table 2.

The system design is illustrated in Fig. 4. It employs an optical detector to receive the incoming optical signal, a directly modulated laser which is used as a pump to boost the signal level at a synchronized level which is based on its extinction ratio, and an EDFA with 5 m length. Serial relays are a far better choice than parallel ones, as the latter solution is not good for longer path links. The noise also adds up if synchronization is not performed properly [12].

3. Simulation Results of WDM-FSO Amplifier System

The system studied not only decreases complexity of the network but also that of its implementation. The simulations show the performance of the proposed system using relays. The range of more than 50 km is noticed for light rain, 60 km for the haze condition, up to 20 km for medium rain and to 8 km in heavy rain at the highest data rate of 2.5 Gb/s as provided in Table 3.

The results for heavy rain and haze from the BER analyzer are shown in Fig. 5. The eye diagram for haze is given in Fig. 6. The laser power is kept at 30 dBm, and the optical amplifier gain is set to an optimized value. The output of the BER analyzer shows that the eye opening is good in the relay-assisted system, thus assuring better SNR at the receiver.

Evaluation of the system shows that for light and medium rain conditions the link distance extends up to the orders of 50 and 2 km respectively, with the BER value of $6.1033 \cdot 10^{-56}$ and $2.38721 \cdot 10^{-55}$ for acceptable transmission.

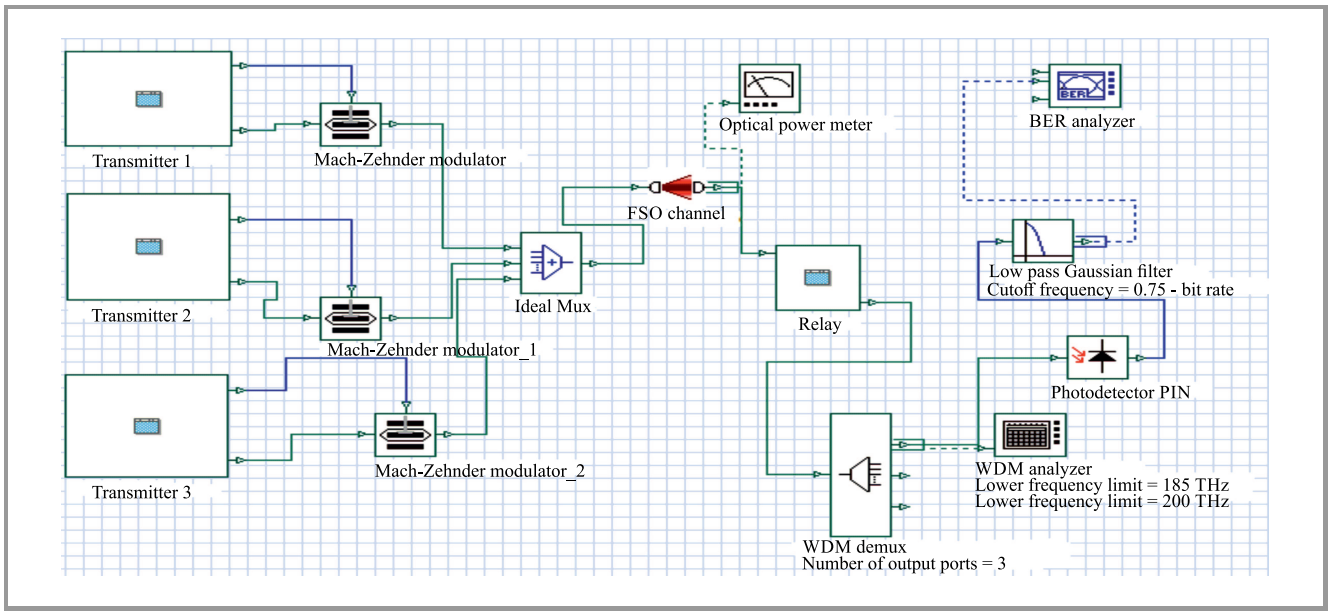

Fig. 4. Relay-assisted network. (See color pictures online at www.nit.eu/publications/journal-jtit)

Table 3
Comparison of WDM-FSO amplifier-assisted and relay-assisted system

Weather condition	Amplifier-assisted WDM system			Relay-assisted WDM system		
	Range [km]	Minimum BER	Q factor	Range [km]	Minimum BER	Q factor
Light rain	25	$2.7534 \cdot 10^{-21}$	9.39	50	$6.1033 \cdot 10^{-56}$	15.72
Medium rain	9.3	$2.70299 \cdot 10^{-20}$	9.14	20	$2.38721 \cdot 10^{-55}$	15.62
Heavy rain	4.5	$3.51199 \cdot 10^{-20}$	9.11	8	$6.2028 \cdot 10^{-56}$	15.71
Haze	23	$7.99541 \cdot 10^{-40}$	10.10	60	$2.3769 \cdot 10^{-54}$	15.69

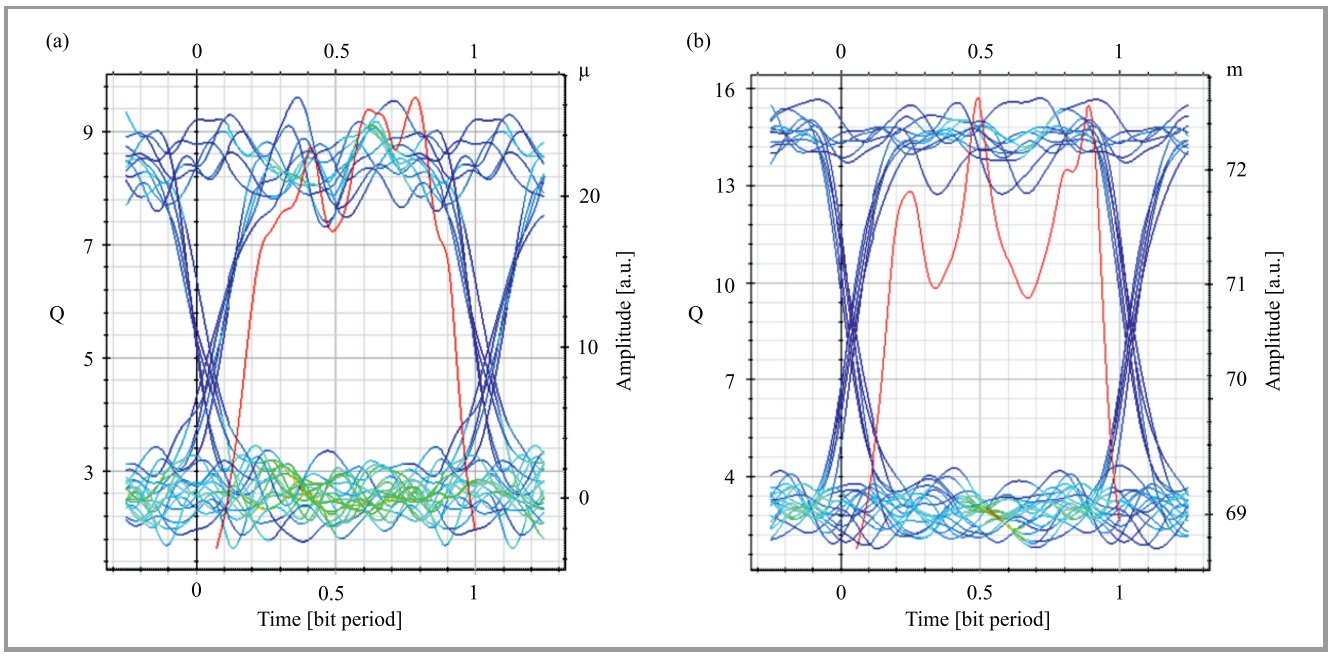

Fig. 5. BER output in heavy rain at 7 km using: (a) amplifier-assisted WDM system, (b) relay-assisted WDM system.

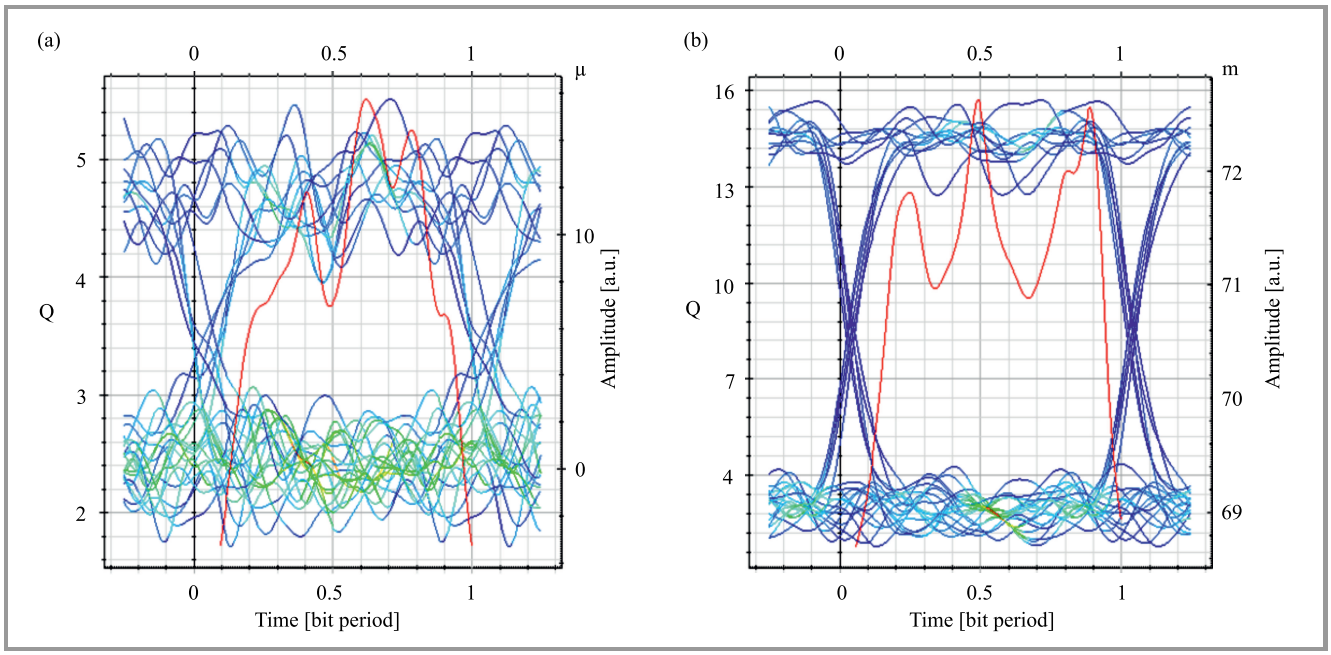

Fig. 6. BER output in haze condition at 60 km using: (a) amplifier-assisted WDM system, (b) relay-assisted WDM system.

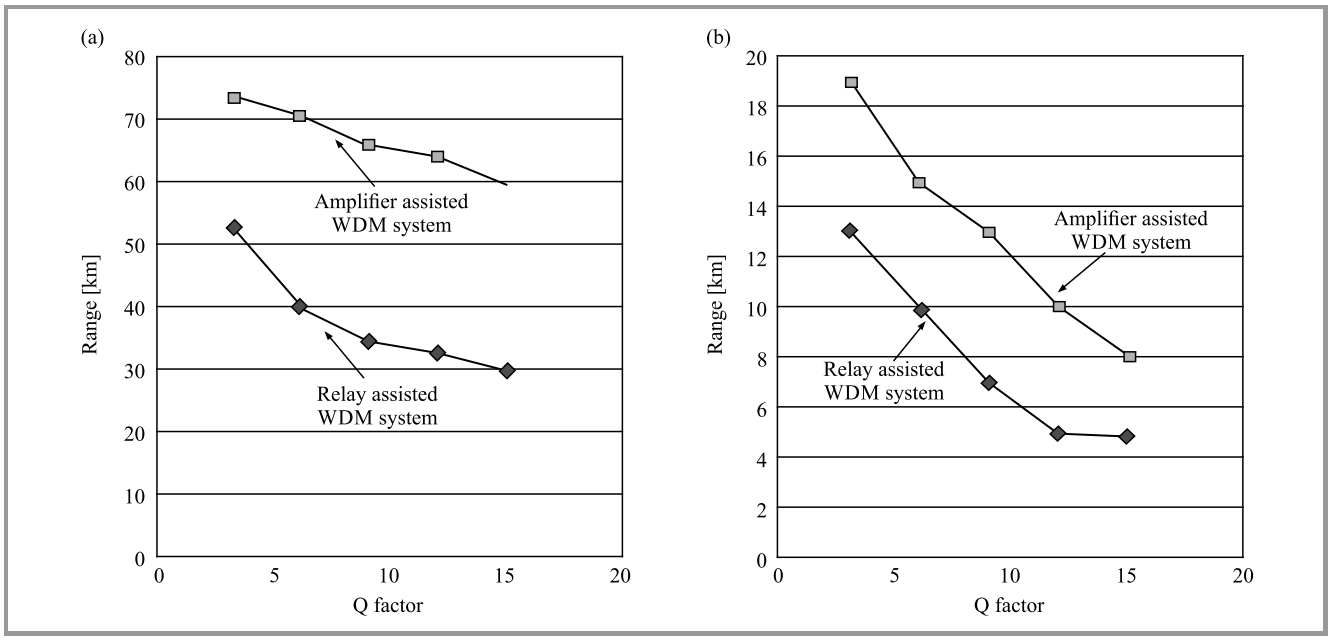

Fig. 7. Performance analysis of two systems in: (a) haze and (b) heavy rain scenarios.

For haze, the distance extends to 60 km having BER of $2.3769 \cdot 10^{-54}$, with performance remaining at an optimum level. Figure 7 describes the performance of the system's variation with the Q factor and the link range for an amplifier-assisted and a relay-assisted system. The graph shows an increasing rate of efficiency for the relay-assisted system with acceptable SNR, and a good transmission range.

3.1. Power optimization

In the next step of the research procedure an effort was made to optimize the power level of the CW laser in order to achieve enhanced performance (Table 4). Originally, the transmitted power used for the system equaled 30 dBm, but after optimization the same system could offer better performance at a low power level, also when compared to the existing solutions. It has been observed that at lower power, performance of the WDM relay-assisted system is better, and that SNR achieved is optimum for an acceptable transmission.

Table 4
Values observed within the system in various weather conditions at 2 dBm

Weather condition	BER	Q factor	Link range [km]	Bit rate [Gb/s]
Haze	$2.6005 \cdot 10^{-6}$	15.66	54	2.5
Light rain	$5.7865 \cdot 10^{-54}$	14.67	50	2.5
Medium rain	$1.5634 \cdot 10^{-45}$	15.80	17	2.5
Heavy rain	$6.6754 \cdot 10^{-8}$	15.73	8	2.5

4. Conclusion

In this article, a simulation of a WDM FSO serial relay assisted communication system is successfully performed. It has been observed that the performance of the system is affected, to a considerable degree, by haze and rain conditions. The results of the study show that the behavior of the system is outstanding in both types of weather conditions. For example, during haze and for the data rate of 2.5 Gb/s, the link range is 60 km with BER of $2.3769 \cdot 10^{-54}$ and an acceptable Q factor. However, the study shows that the range of up to 50 km is achieved for light rain, 20 km for medium rain and 8 km for heavy rain. Finally, the power level has been optimized for the system, 2 dBm is considered as a better solution here. At this level the performance achieved is better for any FSO system that is to be installed with its cost minimized.

References

[1] A. Malik and P. Singh, "Comparative analysis of point to point FSO system under clear and haze weather conditions", *Wireless Person. Commun.*, vol. 78, no. 3, pp. 67–70.

[2] A. Malik and P. Singh, "Free space optics: current applications and future challenges", *Int. J. of Optics*, vol. 2015, Sep. 2015, pp. 1–7 (doi: 10.115/2015/945483).

[3] I. I. Kim, B. McArthur, and E. J. Korevaar, "Comparison of laser beam propagation at 785 nm and 1550 nm in fog and haze for optical wireless communications", in *Proc. Information Technologies 2000*, Boston, MA, USA, 2000, *Optical Wireless Commun. III*, vol. 4214 (doi: 10.1117/12.417512).

[4] A. Hatziefremidis *et al.*, "Bit error rate analysis along a slanted path link between UAVs and ground atations", in *Proc. 15th Int. Conf. on Transpar. Optic. Networks ICTON*, Cartagena, Andalucía, Spain, 2013, pp. 1–4 (doi: 10.1109/ICTON.2013.6602799).

[5] M. Gregory and S. Badri-Hoeher, "Characterization of maritime RF/FSO channel", in *Proc, 2011 Int. Conf. on Space Optic. Syst. and Appl. OCSOS*, Santa, Monica, CA, USA, 2011, pp. 21–27.

[6] H. Maeda *et al.*, "Performance degradation of long-distance optical WDM networks due to bandwidth limitations of wavelength multiplexers and de-multiplexers", *IEEE Photon. Technol. Lett.*, vol. 11, no. 11, pp. 1509–1511, 1999.

[7] N. Dayal, P. Singh, and P. Kaur,"Performance enhancement in WDM-FSO system using optical amplifiers under different rain conditions", in *Proc. Int. Conf. on Intell. Commun., Control and Devices ICICCD*, Dehradun, Uttarakhand, 2015, pp. 293–298 (doi: 10.1117/12.417512).

[8] N. D. Chatzidiamantis *et. al.*, "Protocols for relay-assisted free-space optical systems", in *Proc. Glob. Telecomm. Conf. GLOBECOM 11*, Houston, TX, USA, 2011, pp. 1–6 (arXiv: 1105.3835v1).

[9] J. Jeyarani and D. S. Kumar, "BER analysis of serial relay-assisted FSO systems over strong atmospheric turbulence", in *Proc. 3rd Int. Conf. on Signal Process., Commun. and Network. ICSCN*, Chennai, Tamil Nadu, India, 2015 (doi: 10.1109/ICSN.2015.7219865).

[10] F. Nadeem, E. Leitgeb, M. S. Awan, and S. Chessa, "Comparing lifetime of terrestrial wireless sensor network by hybrid FSO/RF and only RF", in *5th Int. Conf. on Wireless and Mobile Commun. ICWMC*, Cannes, France, 2009 (doi: 10.1109/ICWMC.2009.29).

[11] H. Wu1 and M. Kavehrad, "Availability evaluation of ground to air hybrid FSO/RF links", *Int. J. of Wireless Inform. Networks*, vol. 14, no. 1, pp. 33–45, 2007.

[12] R. Boluda-Ruiz *et al.*, "MISO relay-assisted FSO systems over gamma-gamma fading channels with pointing errors", *Photon. Technol. letters*, vol. 28, no. 3, pp. 229–232, 2015 (doi: 10.1109/LPT.2015.2492622).

[13] M. A. Kashani, M. Safari, and M. Uysal, "Optimal relay placement and diversity analysis of relay-assisted free-space optical communication systems", *J. of Opt. Commun. Network*, vol. 5, no. 1, pp. 37–47, 2013 (doi: 10.1364/JOCN.5000037).

Navneet Dayal has received her M.E. in Electronics & Communication Engineering from UIET, Panjab University, Chandigarh. Her research focuses on wireless optical communication systems.

E-mail: navidayal1992@gmail.com
University Institute of Engineering & Technology
Department of Electronics & Communication Engineering
Panjab University
Chandigarh, India

Preeti Singh is working as an Assistant Professor at the Electronics & Communication Engineering Department in UIET, Panjab University, Chandigarh. She received her B.E. and M.E. degrees in Electronics and Communication Engineering. She obtained her Ph.D. degree in 2013. Her areas of interest include optical communication (wired and wireless), optical biosensors and cognitive neuroscience.
E-mail: preet_singh@pu.ac.in
University Institute of Engineering & Technology
Department of Electronics & Communication Engineering
Panjab University
Chandigarh, India

Pardeep Kaur is working as an Assistant Professor at the Electronics & Communication Engineering Department in UIET, Panjab University, Chandigarh. She received her B.E. and M.E. degrees in Electronics and Communication Engineering. She is pursuing her Ph.D. in wireless sensor networks. Her areas of interest include optical communication and wireless communication.
E-mail: pardeep.tur@gmail.com
University Institute of Engineering & Technology
Department of Electronics & Communication Engineering
Panjab University
Chandigarh, India

Paper

Mobile Cloud for Parallel and Distributed Green Computing

Leszek Siwik, Dawid Kala, Mateusz Godzik, Wojciech Turek, Aleksander Byrski, and Marek Kisiel-Dorohinicki

AGH University of Science and Technology, Krakow, Poland

https://doi.org/10.26636/jtit.2017.111817

Abstract—Mobile Computing and Mobile Cloud Computing are the areas where intensive research is observed. The "mobility" landscape (devices, technologies, apps, etc.) evolves so fast that definitions and taxonomies do not catch up with so dynamic changes and there is still an ambiguity in definitions and common understanding of basic ideas and models. This research focuses on Mobile Cloud understood as parallel and distributed system consisting of a collection of interconnected (and virtualized) mobile devices dynamically provisioned and presented as one unified computing resource. This paper focuses on the mobile green computing cloud applied for parallel and distributed computations and consisting of outdated, abandoned or no longer needed smartphones being able to set up a powerful computing cluster. Besides showing the general idea and background, an actual computing cluster is constructed and its scalability and efficiency is checked versus the results obtained from the virtualized set of smartphones. All the experiments are performed using a dedicated software framework constructed in order to leverage the no-longer-needed smartphones, creating a computing cloud.

Keywords—distributed computing, green computing, mobile cloud, mobile computing, parallel computing, pervasive computing.

1. Introduction

The omnipresence of mobile devices, smartphones, tablets and wearable devices calls for proposing new ways of leveraging their computing power especially that smartphones or phablets with hexa- or octa-core CPUs on board are not a rarity on the market. In fact, they are becoming the standard. Unfortunately, their immense computational power, unbelievable for the users of the Desktop PC only a little more than two decades ago, is utilized (wasted in fact) for such primitive tasks as social network integration, multimedia production, reality enhancement, etc.

Apparently, mobile gaming is a more demanding application, (especially with 3D and VR processing) but similarly to a situation where GP GPU devices or gaming consoles have been adapted to solving computing tasks [1], the question arises if the enormous computational power of mobile devices can be applied for scientific applications such as parallel and distributed computing. Obviously, there are some restrictions and limitations (the most trivial is the battery lifetime) and a singular mobile device is not so powerful to be applied for scientific processing, but what if we put hundreds or thousands of mobile devices together to set up an (ad hoc) cluster of computing devices, in order to perform certain computations? That is a very attractive and interesting idea, especially taking into consideration the number of available devices which together can potentially provide a really significant computational power. Additionally, mobile devices are "located in the environment" and equipped with sensors so during computation additional "observations" or measurements can be performed. What is more, mobile devices are always in constant and direct contact with the user, which significantly extends possible applications of such computational unit(s).

This is how the, so-called Ubiquitous Computing [2], Mobile Computing and Mobile Cloud Computing (MCC) fields of research came about and the latter two are also in the limelight of our research.

MCC is a kind of combination of mobile devices, (cloud) computing and (rich) communication. Recently, intensive research in this area has been observed but there is still ambiguity in definitions and common understanding, especially given that the market of mobile devices, technologies and apps evolves so fast that definitions and taxonomies do not catch up so dynamic changes.

In 2010 in [3] Mobile Computing was defined as "information at fingertips anywhere, anytime". We disagree, since meeting only one single condition that the user is constantly interconnected using his device and is able to open a browser, social media app or any other communication app to check out or to gain information about anything just under their "fingertip" is not a mobile computing, it is just Internet access "at fingertips".

So, what is mobile (cloud) computing? Well, the proper starting point for defining Mobile Cloud are Cloud and Cloud Computing notions. According to [4], Cloud "*is a type of parallel and distributed system consisting of a collection of interconnected and virtualized computers dynamically provisioned and presented as one or more unified computing resources based on service-level agreements established through negotiation between the service provider and consumers*".

Simultaneously, according to [5], Cloud Computing "*is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications and services) that can be rapidly provisioned and released with minimal management*".

On the basis of the above definitions in [6] Mobile Cloud Computing is defined as "*a rich mobile computing technology that leverages unified elastic resources of varied clouds and network technologies toward unrestricted functionality, storage, and mobility to serve a multitude of mobile devices anywhere, anytime through the channel of Ethernet or Internet regardless of heterogeneous environments and platforms based on the pay-as-you-use principle*".

The definition describing in the best way our research and perspective is the one of Cloud presented in [4] but in our case, it has to be adjusted taking the mobility into account. So, following [4] we define Mobile Cloud as *(a type of) parallel and distributed system consisting of a collection of interconnected (and virtualized) mobile devices dynamically provisioned and presented as one unified computing resource*.

Obviously, this requires the addition of a section saying that the resources are provisioned "*according to the service-level agreements established through negotiation between the service provider and consumers*" but since in our research we leave these aspects aside and focus on volunteer model, this part can be omitted in the definition.

Since, as mentioned above, intensive research in mobile (cloud) computing area has recently been performed, different models and approaches have been proposed [7]–[9].

Utilization of mobile devices in computing usually assumes that some of the computing tasks (or even most of them) will be off-loaded to a dedicated computing infrastructure connected wirelessly (see [10], [11]). An interesting concept of bi-location in an agent-based mobile cloud is presented in [12]–[13], where the actual configuration of the infrastructure may be perceived and efficiently mapped to the control-layer utilizing the notion of agency. A similar approach to make the computing application more portable is based on embedding the computing tasks into so-called weblets and deploying them in the mobile cloud [14]. An interesting realization of the Map-Reduce programming model on handheld devices is presented in [15]. A dedicated framework for supporting mobile-computing, using REST was also discussed in [16].

As one may see, researchers address and focus on many different aspects of mobile computing i.e. integration of "traditional cloud" with mobile devices [17], [18], code offloading [19], [20], energy wasting and battery lifetime [21], integration of cloud computing model with Internet of Things [22], frameworks for (mobile) distributed processing [23], security and user privacy [24]–[26], sensor utilization [27], heterogeneity [6], etc.

In this case, we focus on a green computing cloud [28] applied for parallel and distributed computations consisting of outdated, abandoned or no longer needed smartphones being able to set up a powerful computing cluster.

One has to keep in mind that the predicted number of smartphones should rise to the immense figure of 6.1 billion worldwide by 2020 [29], and producers will continue to tempt users to constantly replace their devices with newer versions, condemning the older ones, often still in a good shape, to extinction. Thus, scavenging the remains of the contemporary social civilization can lead to the construction of a truly renewable, easily configurable and powerful, yet completely "green" computing hardware.

To visualize the scale of wasted computing power, let us briefly analyze the issue. On the basis of our experiments, the average mobile device, computational power can be estimated as 0.37 GFlops. Assuming that the billion new phones introduced into the market every year replace old phones, we can conclude that at least 10 PFlops of potential computing hardware is wasted every year. This value is comparable with supercomputers from the top of the Top500 list [30].

In [31] a comprehensive study of leveraging the mobile devices in order to set up a green-computing appliance is presented. We follow and extend this idea exploring scalability and efficiency of both: a homogeneous and heterogeneous mobile computing cloud.

In Section 2 the concept of the framework is presented, followed by the development issues encountered during the implementation process. Next, the experimental verification of the appliance is shown, based on solving selected benchmark problems in a distributed environment. Finally the paper is concluded and future work is sketched.

2. Mobile Cloud Platform for Green Computing

Mobile Cloud Platform (MCP) is a platform for performing parallel computations on strongly distributed, heterogeneous yet massively available and increasingly powerful mobile devices used as a self-manageable computational unit. The main goal and task of the platform is making it possible to set up a cloud of mobile devices constituting a heterogeneous volunteer computational environment. An important assumptions is that when the (part) of computational task is realized on the device(s), it is possible to use their sensor if it is required or helpful for completing the task. Obviously, devices working on the same computational task can communicate with each other, distributing some parts of the task, collecting (partial) results, requiring some additional activities (using sensors), etc.

Two main conceptual assumptions of presented MCP are:

- self-manageability – understood here as the possibility of setting up the cloud, defining a new task, running tasks and collecting results without any additional servers,
- cross-platforming – the cloud can be set up of any mobile devices running any operating system (iOS, Android, Windows Phone, etc.).

Bringing the big picture of our mobile cloud idea it is assumed that the cloud that is set up takes the form of the hierarchical tree where every single node has n branches at most (see Fig. 1). The root node – in the center of the graph – is connected with five nodes, that form five different

branches of mobile devices. Some of them have further branches attached, etc.

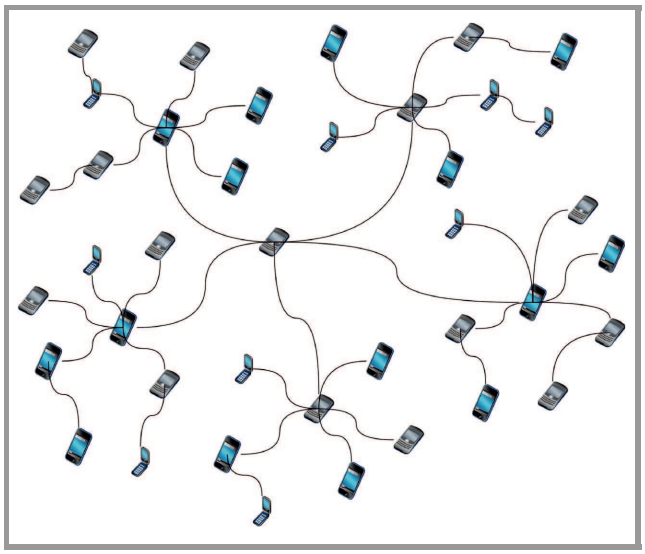

Fig. 1. Hierarchical structure of the mobile cloud.

Keeping the hierarchical structure in mind, the main operations on the cloud, i.e. joining the cloud and starting the computational task, are realized as follows on the conceptual level:

- joining the cloud by the device – the device willing to join the cloud sends a message to any of the devices being already part of the cloud. The message is being delegated to the device being the root of the cloud's tree. The root makes a decision where in the cloud's structure the new device should be joined. The decision is made taking into account the number of devices located in all (sub)trees. When the device has joined the cloud, the root device sends it information about its direct root;
- starting a computational task on the cloud – the device willing to start a new computational task on the cloud sends the message about the new task to all its neighbors. Next the message about the new task is (recursively) propagated among all the devices constituting the cloud. The device which is not able to send the message any further sends back the information if it is going to become the computational node for the new task or not. The message including the list of all the devices going to take part in solving the new task is being sent back to the devices that originated the new task. Next the device originating the new task sends the information to all the devices going to work on the new task that they are part of the virtual tree working on the new computational job.

As one may notice, the aforementioned assumptions require that the direct peer-to-peer communication among devices is available, which requires that either the devices are working within the same LAN or that all devices are publicly available (in the networking sense) so e.g. that they all utilize IPv6 protocols and addressing. There are some restrictions in this context at the moment since most of devices on the market utilize IPv4 protocols but for instance the T-Mobile operator made the IPv6 configuration as the default configuration on the U.S. market for all new devices with Android 4.3 or above from 2013. In December 2013, the list of devices configured for IPv6 included among others [32]: Samsung Galaxy Note 3, Galaxy Light, MetroPCS Samsung Mega, and Google Nexus 5.
Now, the set of IPv6 pre-configured devices looks as follows [32] (hit by December, 12th, 2016): Samsung Galaxy S5, HTC One M8, Samsung Note 3, Samsung Galaxy Light, Google Nexus 5, MetroPCS Samsung Galaxy Mega, Samsung Galaxy S4, Samsung Galaxy Note 2, Google Nexus 4, Samsung Galaxy S3 (latest firmware only), Samsung Galaxy S2 (with Android 4.0 update).
It means that the restrictions for mobile devices direct communication are going to be gradually eliminated and more and more devices available on the market will support the peer-to-peer communication. It is also worth remembering that more and more devices are constantly available in Wi-Fi networks so the direct communication restriction is also (at least partially) eliminated depending on the network configuration.

3. Selected Realization and Implementation Aspects

The easiest way to explain and show the most important realization aspects will be to present how the two most important operations, i.e. joining the cloud by the new device and starting new task, are realized.

3.1. Joining the Cloud by the New Device

When the new device wants to join the mobile cloud it first has to send *ConnectDeviceRequest* message to any device already being part of the cloud (see Fig. 2). *ConnectDeviceRequest* contains among others the IP address of the accessing device.

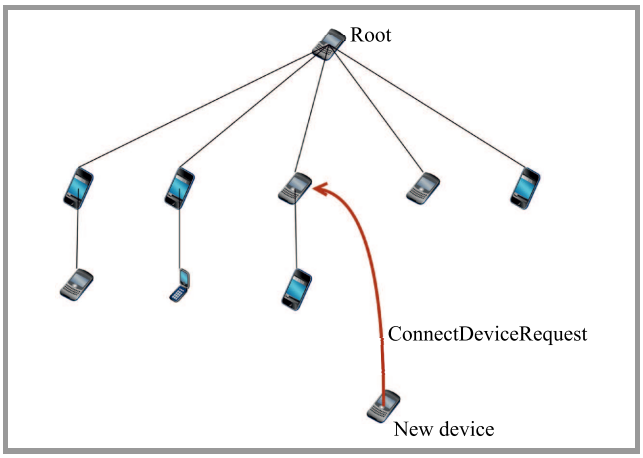

Fig. 2. Connect device request.

Next the device which received *ConnectDeviceRequest* propagates the message up to device being the root of the cloud at the moment.
Next, on the basis of the information about the number of devices accessible through each of its direct children, the root device propagates the *ConnectDeviceRequest* down to the subtree with the lowest number of devices.
The device becoming the direct parent of accessing device sends to the new device the *SaveDeviceConnectionRequest* (see Fig. 3). *SaveDeviceConnectionRequest* message contains among others the IP address of the parental device.

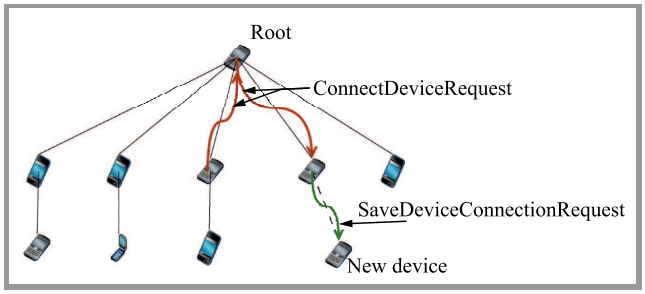

Fig. 3. Save device request.

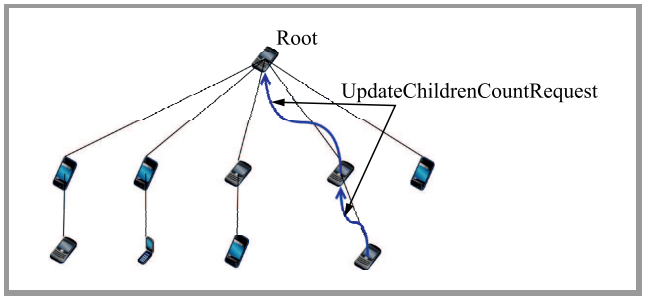

Fig. 4. Update children count request.

Finally, the newly connected device sends up to the parent (which is further propagated up to the root device) the *UpdateChildrenCountRequest* message (see Fig. 4). Every time, the message is sent up to the parental node, it contains the total number of devices in all subtrees on a given level. This way, whenever a new device attempts to access the cloud the cloud knows where it should be connected at the given cloud configuration to achieve a balanced number of devices in all subtrees.

3.2. Starting a New Task on the Cloud

Starting a new task to be computed on the cloud is realized as follows.
Firstly, the device willing to start the new task sends *StartTaskExecutionRequest* message. The message contains among the others: definition of the task to be run (the source code or meta definition), requirements regarding (geo)localization of the devices where the task should be run, information about sensors required to run the task on the single device etc.

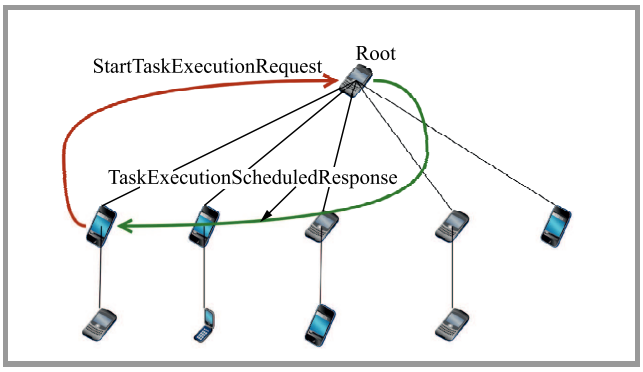

Fig. 5. Start task execution request.

When the device being part of the cloud receives the *StartTaskExecutionRequest* message, the unique UUID task identifier is set up and the *TaskExecutionScheduledResponse* message with the task identifier (see Fig. 5) is sent back.

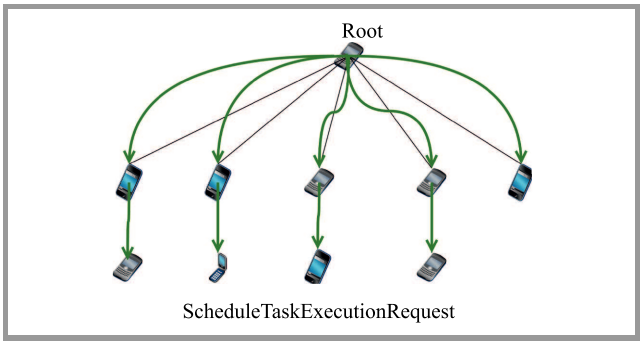

Fig. 6. Schedule task execution request.

Next, the device which has received the *StartTaskExecutionRequest* message propagates the message about the new computational task to all the neighbors as *ScheduleTaskExecutionRequest* message. *ScheduleTaskExecutionRequest* message is propagated through the whole cloud (see Fig. 6).
During the *ScheduleTaskExecutionRequest* message propagation, on the basis of the filters and required features contained in the task definition (required geolocation, sensors etc.) every single device makes a decision whether to participate in the realization of the task or not.
Devices that are not able to propagate the message any further (it is the leaf which did not initialize the propagation process) send the *RemoteTaskReadyRequest* message back to the device they received the *ScheduleTaskExecutionRequest* from.
This part of communication allows for propagation of the information about the devices which have accepted the new task. Along with propagation of the *RemoteTaskReadyRequest* through the cloud – in the message there are gradually aggregated identifiers (IP addresses) of the devices which decided to compute the new task (see Fig. 7).
The last part of starting and configuring the new task on the cloud is propagating the information about the topology of the devices participating in solving the new task.

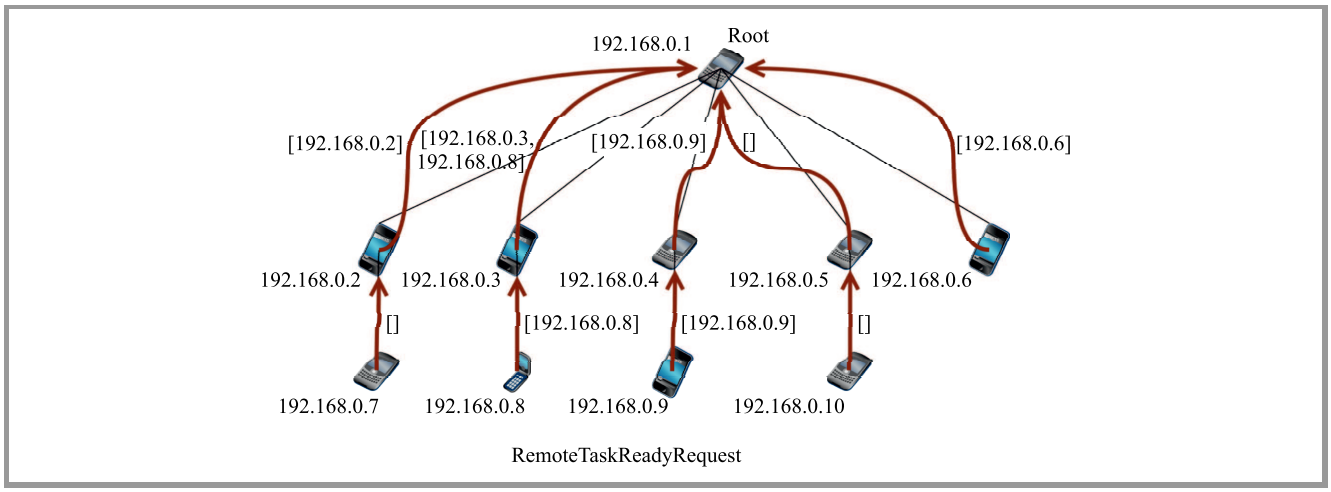

Fig. 7. Task ready request.

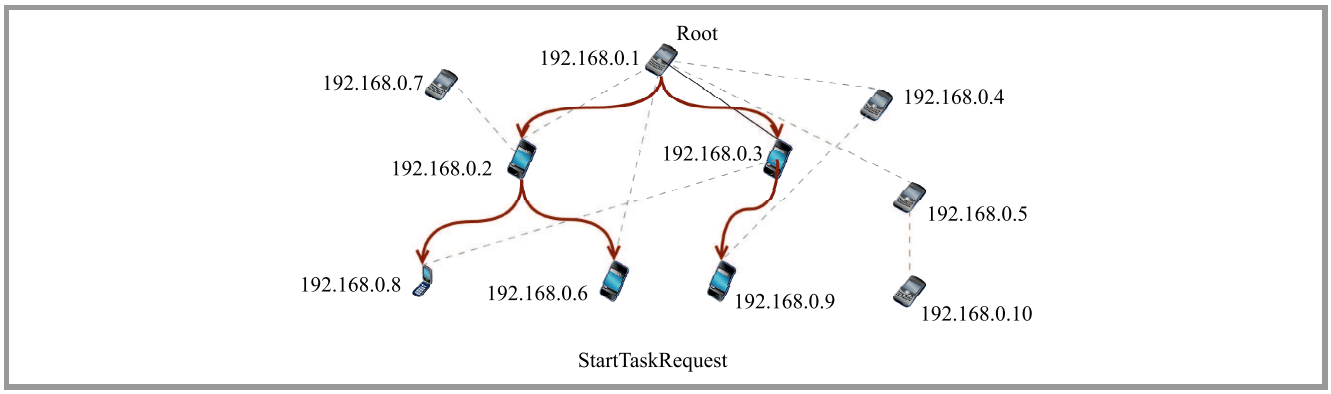

Fig. 8. Start task request.

It consists in creating a temporary, binary tree topology containing all devices which accepted realization of the new task. During this process, within aforementioned temporary topology, the *StartTaskRequest* message is propagated. The *StartTaskRequest* message contains an ordered collection of IP addresses of all the devices taking part in the realization of a new task (see Fig. 8). When the *StartTaskRequest* message is received, the device has all the information about the task definition and the configuration of the task realization environment so it is ready to start and run the task.

4. Experimental Results

The platform has been implemented and tested to verify experimentally its capacity to fulfill functional requirements, ability to perform distributed computations, and to assess its computational power.

The experiments involved heterogeneous devices of different brands, equipped with different hardware and controlled by different operating systems. Therefore, we are not hoping to get fully linear speedup of computations. However, showing that a cloud of cheap devices consuming very little energy can provide significant computational power seems to be a valuable result.

4.1. Functional Verification

The first stage of experiments was the functional verification, i.e. confirming that the platform, all its components, communication patterns and protocols work properly.

This experiment was performed using virtual mobile devices since it is easier to manage a set of virtual devices of the same type than a set of heterogeneous real devices and since using virtual rather than real devices does not negatively interfere with functional verification.

The first step was setting up a sample cloud. It was done using one of arbitrarily selected virtual devices running a simple user interface, thus making the process easier.

Next, we gradually added next devices to the cloud one by one. Devices can be joined to the cloud either by the graphical user interface on one of the devices being available in the same network or by the device that wants to join the cloud.

Next, the experiment of π estimation by Monte Carlo method with $3 \cdot 10^9$ evaluated points was repeated on the cloud consisting of a growing number of the same virtual Android mobile devices. The same experiment but with $3 \cdot 10^{10}$ number of points was also conducted. The times of computations in both cases are presented in Fig. 9.

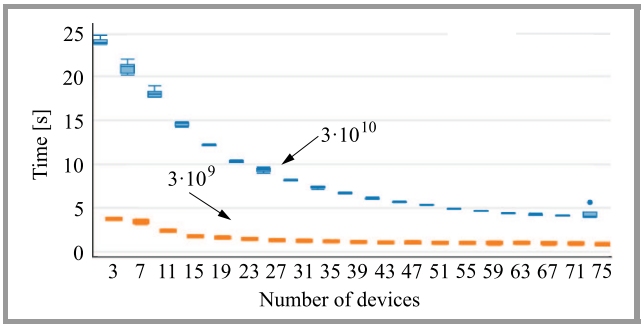

Fig. 9. Time of computations with a growing number of virtual mobile devices.

Since there are no differences among hardware and software specifications of the devices taking a part in computations, the classic picture of shortening computation time along with the growing number of computational units (i.e. virtual android mobile devices) was observed.
Figures 10 and 11 presents the speedup $S = \frac{T_1}{T_N}$ and the efficiency $E = \frac{S}{N}$ for estimating π by the Monte Carlo method with $3 \cdot 10^9$ and $3 \cdot 10^{10}$ number of points. The first conclusion from presented results is that since devices are the same, very typical characteristics of the speedup and the efficiency can be observed. The second conclusion is that the more difficult problem to be solved is, the higher the profit from the parallelization can be observed since when the instance of the problem is to small, communication overhead consumes potential profits from the parallelization.
The task executed during above experiments is relatively simple but at this point the goal was not to solve highly sophisticated or difficult problem but to confirm that all the components of the platforms work properly and that it is possible to set up a mobile cloud and to run computational tasks over there. The general conclusions coming from the experiments presented in this subsection is that a proper functioning of the proposed idea and implementation of particular parts of the platform and communication protocols could be observed and was confirmed. If so, the next step was setting up the cloud consisting of real heterogeneous mobile devices and running some computational tasks in it.

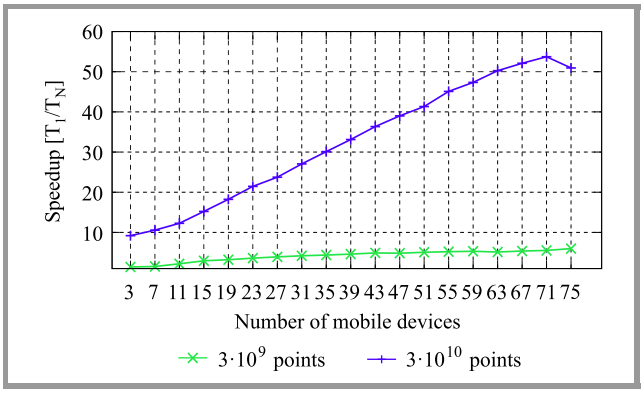

Fig. 10. Speedup as the function of the number of mobile devices for π value estimation by the Monte Carlo method with $3 \cdot 10^9$ and $3 \cdot 10^{10}$ points.

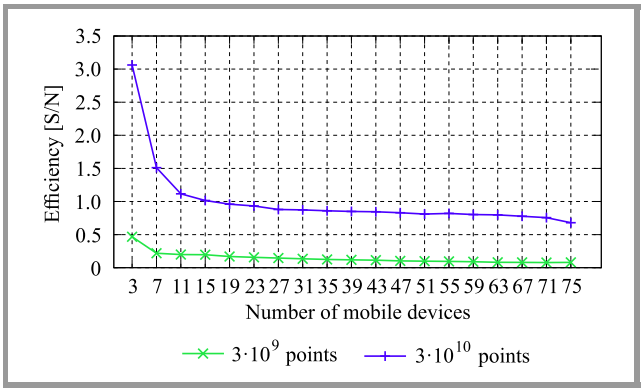

Fig. 11. Efficiency as the function of the number of mobile devices for π value estimation by the Monte Carlo method with $3 \cdot 10^9$ and $3 \cdot 10^{10}$ points.

4.2. Experimental Verification on Real Mobile Devices

In order to perform the tests with real devices, we collected several old and unused devices from our colleagues. We managed to collect 14 devices, however 5 of these failed to execute the client application of the platform or did not start at all. This is the first among the important issues which must be expected when building a system of this type. The main aim, that is putting together of the computing system out of trash, is an appealing idea. However, as it will be made clear, some real drawbacks were spotted. Of course, they were mostly caused by the technical condition of the collected hardware.

The first set of tests aimed at estimating the computational capabilities of particular devices. Each device formed a cloud composed of itself only. The task was again to determine of the π number by random selection of points within a square and a circle (Monte Carlo method). The number of points was $3 \cdot 10^9$. The tests were repeated 5 times. The results are presented in Fig. 12.
Significant differences in performance were expected to be observed (again, because of the fact that these devices were randomly collected). However, this simple experiment shows another major issue related to utilization of this kind of hardware. Six out of nine tested devices give repeatable results, which makes it possible to predict their behavior. The other 3 devices showed a very significant dispersion between different test runs. This was probably caused by the lack of stability of the operation systems loaded with different software installed earlier by the user. For the next experiments, we will thoroughly clean up the hardware and make it homogeneous from the point of view of operating system. This might be easily done using e.g. Cyanogen Mod[1] that removes all the unnecessary "bloat-ware" installed by the developer or the dealer. Moreover, this mod spans over multiple hardware configurations, so installation

[1]CYNG, Company Industries Computer Software Palo Alto, CA, https://cyngn.com

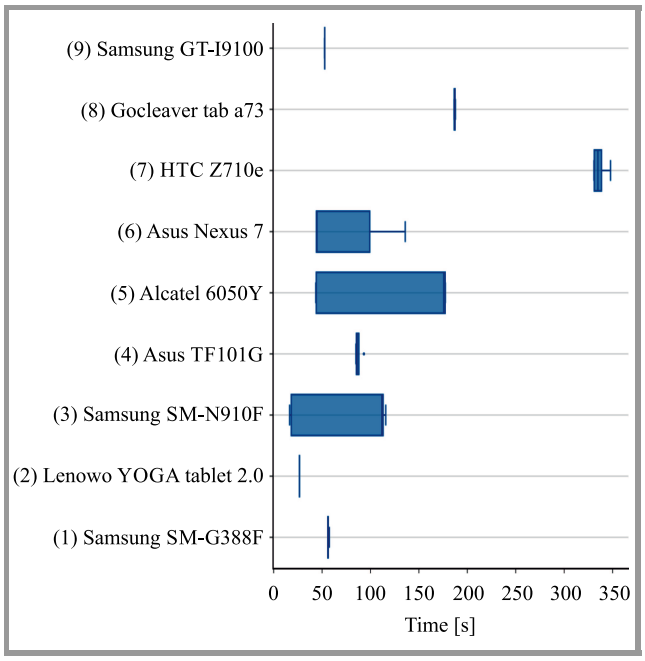

Fig. 12. Evaluation of performance capabilities of the considered devices.

of recent versions of Android systems can be realized even if the version is not supported using the older hardware.

In order to verify that, the computing platform can provide a significant increase in performance, a set of tests with a growing number of devices was performed. The computational task was the same as previously (estimating the value of π with the Monte Carlo method and with the number of points set to $3 \cdot 10^9$). The parallelization was achieved by assigning each device an equal subset of points to verify, so the tasks were of the same size.

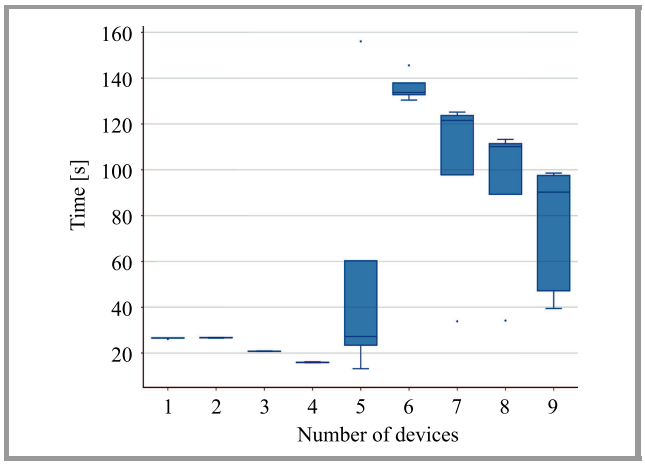

Fig. 13. Time of computations with a growing number of devices, starting from the fastest one.

The first approach was to add new devices, starting from the fastest one down to the slowest according to performance results presented in Fig. 12 – the order was: 2, 9, 1, 6, 3, 4, 5, 8, 7. The results are presented in Fig. 13.

Clearly, adding the third and fourth device increased the performance of the computing cloud. Adding the device no. 3 – Samsung SM-N910F – results in a huge drop in performance.

In the second set of tests, the devices were added in the inverted order. The results are shown in Fig. 14.

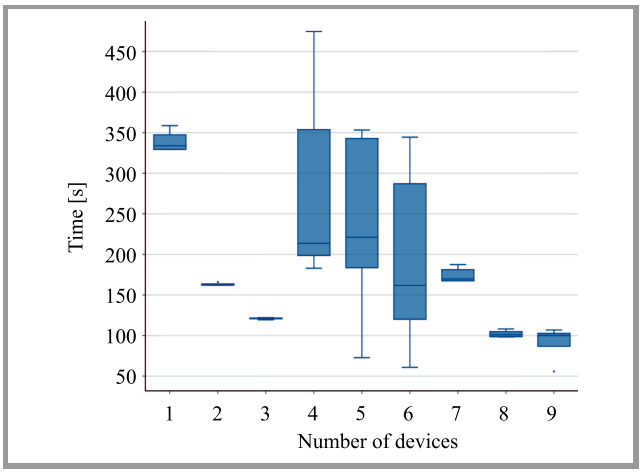

Fig. 14. Time of computations with growing number of devices, starting from the slowest.

Similarly, adding the second and the third device significantly decreases computing time. Along with the growing number of devices, a higher dispersion of results and a significant drop in performance can be observed. This issue can be addressed by extending filters in the task definition with minimal required computational power to be met by each device going to be a part of the computational task. This way modern devices with powerful CPUs would perform computing operations, whereas old devices would provide e.g. measurements and "observations" from available sensors.

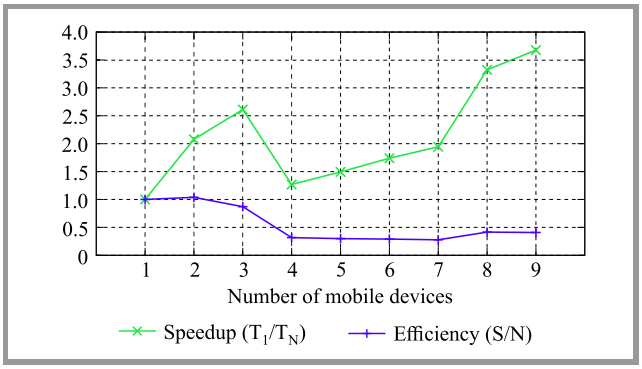

Fig. 15. Speedup (T_1/T_N) and efficiency (S/N) as the function of the number of real mobile devices in the cloud.

Figure 15 shows the speedup $S = \frac{T_1}{T_N}$ and the efficiency $E = \frac{S}{N}$ of the cloud set up from (obsolete) real devices. This time, in contrast to speedup and efficiency for the homogeneous environment presented in Figs. 10 and 11, characteristics are not so regular due to cooperation between

the devices with completely different hardware and software specifications so with totally different computational power and time needed to complete the assigned (part of the) task.

4.3. Estimating Mobile Cloud Computational Power

An important question one may ask regards the computational power of the cloud of cooperating mobile devices set up according to the model presented in this paper.
To estimate the power, the algorithm for π estimation was developed in C++ with MPI. Next, π estimation task was run separately on a single PC station with Intel i7-3370K CPU and on the cloud of the following twelve physical devices: HTC Sensation Z710e, JSR Soul, TCT Alcatel One Touch (x2), Samsung Galaxy Nexus, Samsung GT-I9505, Samsung GT-I9300, Samsung GT-P5100, Samsung GT P5313, LGE Nexus 4, Sony C6603, HTC Desire X. During the experiments task duration on both environments for the same problem sizes wasmeasured and used for further estimations.

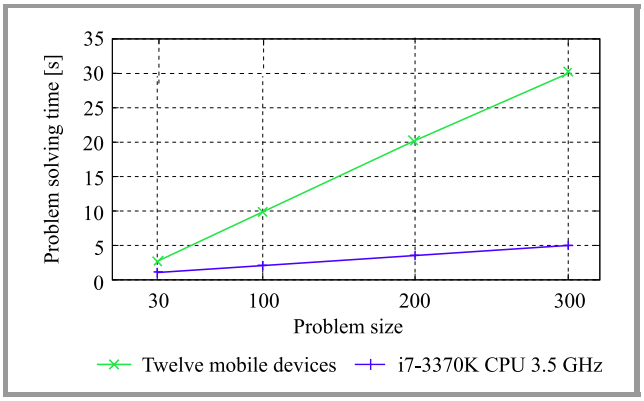

Fig. 16. Time of π estimation with the Monte Carlo method on mobile cloud and PC station with Intel i7-3370K @3.50 GHz CPU.

The results obtained on the mobile cloud and the PC station for 100, 200 i 300 million of points are presented in Fig. 16. Taking the average values obtained on both computational environments, it is possible to calculate coefficient M representing the efficiency ratio between mobile cloud and PC station.

$$M \approx \frac{1.75+3.60+5.35}{3} \div \frac{(10.10+20.25+30.61)\times 12}{3} \approx 0.0146$$

Using IntelBurnTest v2.54[2] the computational power of Intel i7-3370K CPU was set as 25.47 GFlops. So, taking both: the calculated computational power of the CPU used in the experiment and the calculated value of M coefficient, the computational power of a "theoretical" single mobile device working in the presented mobile cloud can be estimated at ~0,37 GFlops.

[2]IntelBurnTest is a tool for measuring CPU efficiency. It is available under: http://www.majorgeeks.com/files/details/intelburntest.html.

For comparison,

- to generate the same computational power as Tianhe-2 supercomputer [33][3] (33.9 PFlops) a mobile cloud consisting of 91.6 million mobile devices would have to be set up,
- to generate the computational power comparable to the power of the Bitcoin network c.a. 6.3 trillion mobile devices would have to be used which would be hardly possible,
- to generate the computational power comparable to the power of the BONIC grid c.a. 158.4 million of mobile devices would have to be used.

5. Conclusions and Future Work

Putting together a number of mobile devices, recovered for instance "from trash", makes it possible to set up easily a computing cloud that may be used for solving many different computational tasks – although the hardware setup and the operational system related issues may appear (and they did, as it was presented in this paper). Keeping in mind the rapidly growing computational power of single mobile device(s), the possible applications of such an appliance are indeed very broad. Devices with octa-core CPUs were not rarity even in 2014 [35] and today they are rather the standard [36].
The concept of collecting and connecting a large number of such devices in order to provide significant computational power seems valuable. The cost of building such a computing cluster is relatively low. Moreover, the cost of computations is also low – the devices are designed to consume as little energy as possible.
The most natural application of the proposed platform is of course volunteer computing leveraging donors' devices, however in this case two fundamental questions may be asked:

- what is the "business model" of the solution i.e. why should people agree to utilizing their (personal) devices for performing some external, "exotic" (computational) tasks, potentially blocking out sensors, consuming the battery power, consuming mobile data bandwidth etc. Well, it is for sure an important question but the same questions are valid in the case of any volunteer computing platform and since the goal of this paper is to present the concept, some realization aspects and preliminary experimental verification, it is out of its scope to discuss the business model;
- what is the type of tasks that could be (or even should be) efficiently run on a mobile cloud platform. Well, the number of mobile devices available on the market

[3]World's fastest supercomputer according to the TOP500 lists from June 2013 until November 2015. It was surpassed in June 2016 by the Sunway TaihuLight [34].

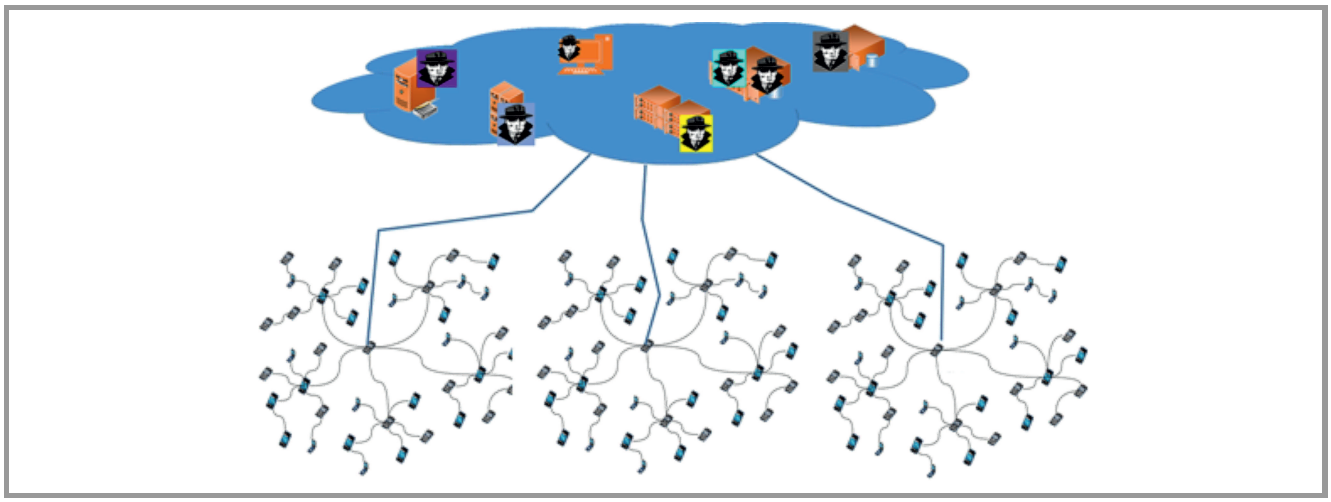

Fig. 17. The idea of cooperating mobile cloud and stationary computational environments.

is rapidly growing. Also, their computational power is greater and greater (quad-, hexa-, octa-core CPUs). Plus the same issues growing power regards memory, connectivity, battery, etc. So the answer can be simply "any". But the real advantage of the mobile devices used as the computational units is their personalization (your phone is always with you, and it is only yours) and localization in the real environment. So it seems that a particularly interesting task to be run on a cloud of the mobile devices would be a tasks related to environment sampling along with complex computations in the background, such as weather forecasting, air contamination monitoring and forecasting, distributed intelligent car navigation, and supporting public transportation system.

One of our further research and development directions is enhancing the presented mobile cloud with possibility of cooperating and spreading tasks to stationary computational units as presented in Fig. 17.

In this model, devices will be able to make decisions which tasks or which parts of particular tasks should be run on stationary computational environments (on the supercomputing center) and which of them should be run on the device. For instance, due to intensive cooperation with the end user, sampling and monitoring of the environment or intensive communication and cooperation among different devices and/or users are required to complete the given (part of) task.

Another very important issue is to work on the stability of the whole software configuration, therefore homogenization of the operating system will be considered (e.g. using Cyanogen Mod), tests will be rerun and extended, both for more hardware devices and for new problems.

Acknowledgements

This research was supported by AGH University of Science and Technology Statutory Fund.

References

[1] S. Mittal and J. S. Vetter, "A survey of CPU-GPU heterogeneous computing techniques", *ACM Computing Surv.*, vol. 47, pp. 69:1–69:35, 2015.

[2] U. Hansmann, L. Merk, M. S. Nicklous, and T. Stober, *Pervasive Computing: The Mobile World*. Springer, 2003.

[3] M. Satyanarayanan, "Mobile computing: The next decade", in *Proc. of the 1st ACM Worksh. on Mob. Cloud Comput. & Services: Social Networks and Beyond MCS'10*, San Francisco, CA, USA, 2010, pp. 5:1–5:6 (doi: 10.1145/1810931.1810936).

[4] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging {IT} platforms: Vision, hype, and reality for delivering computing as the 5th utility", *Future Gener. Comp. Syst.*, vol. 25, no. 6, pp. 599–616, 2009.

[5] P. Mell and T. Grance, "The NIST definition of cloud computing", 2011 [Online]. Available: http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf

[6] Z. Sanaei, S. Abolfazli, A. Gani, and R. Buyya, "Heterogeneity in mobile cloud computing: taxonomy and open challenges", *IEEE Commun. Surv. & Tutor.*, vol. 16, no. 1, pp. 369–392, 2014.

[7] A. Khan, M. Othman, S. Madani, and S. Khan, "A survey of mobile cloud computing application models", *IEEE Commun. Surv. & Tutor.*, vol. 16, no. 1, pp. 393–413, 2014 (doi: 10.1109/SURV.2013.062613.00160).

[8] M. Rahimi, J. Ren, C. Liu, A. Vasilakos, and N. Venkatasubramanian, "Mobile cloud computing: A survey, state of art and future directions", *Mobile Networks and Applications*, vol. 19, no. 2, pp. 133–143, 2014 (doi: 10.1007/s11036-013-0477-4).

[9] J. Samad, S. W. Loke, and K. Reed, "Mobile cloud computing", in *Cloud Services, Networking, and Management*, N. L. S. da Fonseca and R. Boutaba, Eds. Wiley, 2015, pp. 153–190 (doi: 10.1002/9781119042655).

[10] A. R. Khan, M. Othman, S. A. Madani, and S. U. Khan, "A survey of mobile cloud computing application models", *IEEE Commun. Surv. & Tutor*, vol. 16, pp. 393–413, 2014 (doi: 10.1109/SURV.2013.062613.00160).

[11] N. Fernando, S. W. Loke, and W. Rahayu, "Mobile cloud computing: A survey", *Future Gener. Comp. Syst.*, vol. 29, no. 1, pp. 84–106, 2013. Including Special section: AIRCC-NetCoM 2009 and Special section: Clouds and Service-Oriented Architectures.

[12] A. Byrski, R. Dębski, and M. Kisiel-Dorohinicki, "Towards an agent-based augmented cloud", *J. of Telecom. and Inform. Technol.*, no. 1, pp. 16–22, 2012.

[13] A. Byrski, R. Dębski, and M. Kisiel-Dorohinicki, "Agent-based computing in an augmented cloud environment", *Int. J. of Comp. Syst. Science & Engin.*, vol. 27, no. 1, pp. 7–18, 2012.

[14] X. Zhang, J. Schiffman, S. Gibbs, A. Kunjithapatham, and S. Jeong, "Securing elastic applications on mobile devices for cloud computing", in *Proc. of the 2009 ACM Worksh. on Cloud Comput. Secur. CCSW'09*, Chicago, IL, USA, 2009, pp. 127–134.

[15] P. R. Elespuru, S. Shakya, and S. Mishra, "MapReduce system over heterogeneous mobile devices", in *Software Technologies for Embedded and Ubiquitous Systems*, S. Lee and P. Narasimhan, Eds. *LNCS*, vol. 5860, pp. 168–179. Berlin, Heidelberg: Springer, 2009.

[16] J. H. Christensen, "Using restful web-services and cloud computing to create next generation mobile applications", in *Proc. of the 24th ACM SIGPLAN Conf. Companion on Object Oriented Programm. Syst. Lang. and Appl. OOPSLA 2009*, Orlando, FL, USA, 2009, pp. 627–634.

[17] X. Zhang, A. Kunjithapatham, S. Jeong, and S. Gibbs, "Towards an elastic application model for augmenting the computing capabilities of mobile devices with cloud computing", *Mob. Netw. & Appl.*, vol. 16, no. 3, pp. 270–284, 2011 (doi: 10.1007/s11036-011-0305-7).

[18] B. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, "Clonecloud: Elastic execution between mobile device and cloud", in *Proc. of the 6th Conf. on Comp. Syst. EuroSys 2011*, Salzburg, Austria, 2011, pp. 301–314.

[19] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing", *IEEE/ACM Trans. on Netw.*, vol. 24, no. 5, pp. 2795–2808, 2016 (doi: 10.1109/TNET.2015.2487344).

[20] E. Cuervo *et al.*, "Maui: Making smartphones last longer with code offload", in *Proc. of the 8th Int. Conf. on Mobile Systems, Appl., and Serv. MobiSys 2010*, San Francisco, CA, USA, 2010, pp. 49–62.

[21] K. Gai, M. Qiu, H. Zhao, L. Tao, and Z. Zong, "Dynamic energy-aware cloudlet-based mobile cloud computing model for green computing", *J. of Netw. and Comp. Appl.*, vol. 59, no. C, pp. 46–54, 2016 (doi: 10.1016/j.jnca.2015.05.016).

[22] A. Botta, W. Donato, V. Persico, and A. Pescapé, "Integration of cloud computing and internet of things: A survey", *Future Gener. Comp. Syst.*, vol. 56, pp. 684–700, 2016 (doi: 10.1016/j.future.2015.09.021).

[23] M. Shiraz, A. Gani, R. Khokhar, and R. Buyya, "A review on distributed application processing frameworks in smart mobile devices for mobile cloud computing", *IEEE Commun. Surv. Tutor.*, vol. 15, no. 3, pp. 1294–1313, 2013 (doi: 10.1109/SURV.2012.111412.00045).

[24] M. Gai, M. Qiu, L. Tao, and Y. Zhu, "Intrusion detection techniques for mobile cloud computing in heterogeneous 5G", *Secur. and Commun. Netw.*, vol. 9, no. 16, pp. 3049–3058, 2016 (doi: 10.1002/sec.1224).

[25] M. Alizadeh, S. Abolfazli, M. Zamani, S. Baharun, and K. Sakurai, "Authentication in mobile cloud computing: A survey", *J. of Netw. and Comp. Appl.*, vol. 61, pp. 59–80, 2016 (doi: 10.1016/j.jnca.2015.10.005).

[26] N. Kumar, K. Kaur, S. Misra, and R. Iqubal, "An intelligent RFID-enabled authentication scheme for healthcare applications in vehicular mobile cloud", *Peer-to-Peer Netw. and Appl.*, vol. 9, no. 5, pp. 824–840, 2016.

[27] C. Zhu, H. Wang, X. Liu, L. Shu, L. Yang, and V. Leung, "A novel sensory data processing framework to integrate sensor networks with mobile cloud", *IEEE Syst. J.*, vol. 10, no. 3, pp. 1125–1136, 2016.

[28] S. Murugesan and G. R. Gangadharan, Eds., *Harnessing Green IT: Principles and Practices*. Wiley, 2012.

[29] I. Lunden, "6.1B Smartphone Users Globally By 2020, Overtaking Basic Fixed Phone Subscriptions", Tecdhrunch, Jun. 2, 2015 [Online]. Available: https://techcrunch.com/2015/06/02/6-1b-smartphone-users-globally-by-2020-overtaking-basic-fixed-phone-subscriptions

[30] 48th edition of the TOP500, nov. 2016 [Online]. Available: https://www.top500.org/lists/2016/11/

[31] H. Ba, W. Heinzelman, C. A. Janssen, and J. Shi, "Mobile computing – a green computing resource", in *Proc. 2013 IEEE Wirel. Commun. and Netw. Conf. WCNC 2013*, Shanghai, China, 2013, pp. 4451–4456.

[32] "T-mobile IPV6 is here and now", Tech. Rep., T-Mobile, 2013–2016 [Online]. Available: https://sites.google.com/site/tmoipv6/lg-mytouch

[33] N. Hemsoth, "Full details uncovered on chinese top supercomputer", HPC Wire, 2013 [Online]. Available: https://www.hpcwire.com/2013/06/02/full_details_uncovered_on_chinese_top_supercomputer/ (accessed 2016-12-15).

[34] M. Feldman, "China tops supercomputer rankings with new 93-petaflop machine", Top500, 2016 [Online]. Available: https://www.top500.org/news/china-tops-supercomputer-rankings-with-new-93-petaflop-machine/ (accessed 2016-12-15).

[35] T. Florin, "10 of the best octa-core smarphones available now", Aug. 17, 2014 [Online]. Available: http://www.phonearena.com/news/10-of-the-best-octa-core-smartphones-available-now_id59431

[36] H. Bauer, Y. Goh, J. Park, S. Schink, and C. Thomas, "The supercomputer in your pocket", *McKinsley on Semiconductros*, no. 2, pp. 14–27, 2012 [Online]. Available: http://www.mckinsey.com/~/media/mckinsey/dotcom/client_service/semiconductors/issue%202%20autumn%202012/pdfs/the_supercomputer_in_your_pocket.ashx

Leszek Siwik obtained his Ph.D. in 2009 and works as an Assistant Professor in AGH University of Science and Technology. He is mostly interested in metaheuristics and multi-criteria optimization but also software engineering and mobile systems.

E-mail: siwik@agh.edu.pl
Department of Computer Science
AGH University of Science and Technology
30 Mickiewicza Av.
30-059 Krakow, Poland

Dawid Kala obtained his M.Sc. in 2016 from AGH University of Science and Technology. He is interested in software engineering and mobile systems.

E-mail: dawidkala@gmail.com
Department of Computer Science
AGH University of Science and Technology
Mickiewicza av. 30
30-059 Krakow, Poland

Mateusz Godzik obtained his M.Sc. in 2016 from Pedagogical University of Krakow and is currently a Ph.D. student at AGH University of Science and Technology. His interests cover cryptography, security and software engineering.

E-mail: godzik@agh.edu.pl
Department of Computer Science
AGH University of Science and Technology
30 Mickiewicza Av.
30-059 Krakow, Poland

Aleksander Byrski obtained his Ph.D. in 2007 and D.Sc. in 2013. He works as an Assistant Professor in AGH University of Science and Technology. He is interested in metaheuristics, agent-based systems, parallel and distributed computing.

E-mail: olekb@agh.edu.pl
Department of Computer Science
AGH University of Science and Technology
30 Mickiewicza Av.
30-059 Krakow, Poland

Wojciech Turek obtained his Ph.D. in 2010 and works as an Assistant Professor in AGH University of Science and Technology. He is interested in mobile robots, parallel and distributed computing and simulation.

E-mail: wojciech.turek@agh.edu.pl
Department of Computer Science
AGH University of Science and Technology
30 Mickiewicza Av.
30-059 Krakow, Poland

Marek Kisiel-Dorohinicki obtained his Ph.D. in 2001 and D.Sc. in 2013. He works as an Assistant Professor in AGH University of Science and Technology. He is interested in software engineering and agent-based systems.

E-mail: doroh@agh.edu.pl
Department of Computer Science
AGH University of Science and Technology
30 Mickiewicza Av.
30-059 Krakow, Poland

Paper

Monte Carlo Tree Search Algorithm for the Euclidean Steiner Tree Problem

Michał Bereta

Institute of Computer Science, Cracow University of Technology, Cracow, Poland

https://doi.org/10.26636/jtit.2017.122017

Abstract—This study is concerned with a novel Monte Carlo Tree Search algorithm for the problem of minimal Euclidean Steiner tree on a plane. Given *p* points (terminals) on a plane, the goal is to find a connection between all the points, so that the total sum of the lengths of edges is as low as possible, while an addition of extra points (Steiner points) is allowed. Finding the minimum Steiner tree is known to be np-hard. While exact algorithms exist for this problem in 2D, their efficiency decreases when the number of terminals grows. A novel algorithm based on Upper Confidence Bound for Trees is proposed. It is adapted to the specific characteristics of Steiner trees. A simple heuristic for fast generation of feasible solutions based on Fermat points is proposed together with a correction procedure. By combing Monte Carlo Tree Search and the proposed heuristics, the proposed algorithm is shown to work better than both the greedy heuristic and pure Monte Carlo simulations. Results of numerical experiments for randomly generated and benchmark library problems (from OR-Lib) are presented and discussed.

Keywords—Euclidean Steiner tree problem, MCTS, Monte Carlo Tree Search, UCT algorithm.

1. Introduction

One of the most recent achievements in the field of computational intelligence was the victory of the AlphaGo program in the game of Go against human champion Lee Sedol. The game of Go had been a challenge for computer scientists for many years. AlphaGo has been based on the mixture of deep learning algorithms [1] and the Monte Carlo tree search approach [2]. Monte Carlo tree search (MCTS) has proved to be a promising new search technique. Although the most spectacular successes of MCTS come from the field of playing agents, MCTS is a general search technique, and attempts have already been made to apply this technique to other problems, such as planning or optimization.

In this work, MCTS is applied to the problem of finding the minimal Steiner tree on the Euclidean plane (ESTP). ESTP is formulated as follows. Given p points called terminals, connect them so that the sum of lengths of edges is minimal. Additional intermediate points called Steiner points can be added. If adding Steiner points is not allowed, the problem becomes the task of finding the minimum spanning tree (MST) for the given set of terminals, which can be solved efficiently by algorithms such as Prim [4]. However, ESTP is np-hard [5]. Figure 1 shows a minimal spanning tree and a Steiner tree for a sample set of points. The sum of lengths of all edges is smaller for the Steiner tree than for the minimal spanning tree.

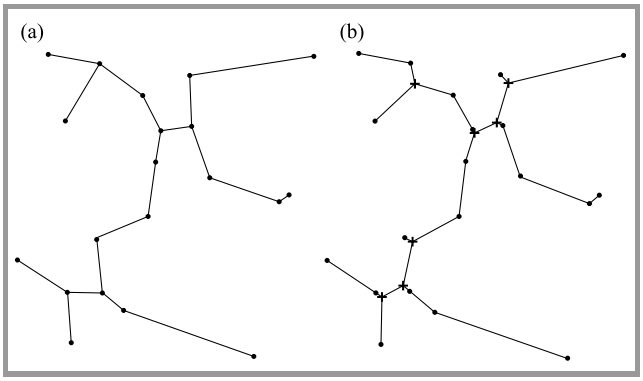

Fig. 1. A minimal spanning tree (a) and a Steiner tree (b) for a set of points; Steiner points are marked with "+".

Many practical problems require to solve a two-dimensional ESTP on a plane, or other versions of Steiner tree. Therefore, the issue evokes broad interest. Three-dimensional problems are also of practical concern. Such problems emerge in VLSI chip layout design, distribution network planning [6], wireless and sensor networks [7], robotics and molecular structure modeling [8]. A good review of Steiner tree applications can be found in [7], robotics and molecular structures modeling [8]. Another good review of Steiner trees applications can be found in [5]. Although exact algorithms exist that enable to solve ESTP on a plane, such as those proposed by Melzak [9], [10], Trietsch and Hwang [11] and Smith [12], they are time-consuming, and there is a need for new algorithms. New heuristics are still being proposed [13]. Evolutionary algorithms have also been proposed for various types of Steiner problems, such as [14]. The MCTS algorithm proposed for ESTP is, to the best of the author's knowledge, truly unique.

Motivations and contributions of this work are as follows. Firstly, it is the first time the Upper Confidence bound for Trees (UCT) algorithm has been applied to the Steiner problem. Secondly, while exact algorithms related to the Euclidean Steiner problem in 2D exist, the proposed algo-

rithm can serve as a heuristic to provide fast preliminary solutions.
Many Steiner problems, such as in higher dimensions, remain unsolved, and the proposed algorithm can be treated as the first attempt to approach these problems with MCTS.
The rest of the paper is organized in the following manner. In Section 2, MCTS is introduced as a general search algorithm together with its more specific version, which uses upper confidence bound (UCB) as a policy to guide the search process. In Section 3, the most important properties of Euclidean Steiner trees are described, which enable to formulate a simple heuristic and a greedy search technique. This heuristic is further combined with MCTS in Section 4 to propose a new algorithm for ESTP. Several possible improvements are also discussed. In Section 5, the results of numerical experiments are presented and analyzed. By using proper statistical procedures, the results of the proposed MCTS for ESTP are compared with the results of the greedy search method and simple Monte Carlo simulation. Additionally, the ability of the proposed algorithm to find the correct solutions is tested on the benchmark problems from OR-Lib [15], for which the exact solutions are known. Finally, the conclusions are drawn in Section 6.

2. Monte Carlo Tree Search

In many instances, searching for a solution to a given problem can be represented as traversing the nodes of a tree. In most cases it is impossible to visit all the nodes efficiently. Therefore, heuristics are used to make a decision about the most promising nodes that should be visited to maximize the reward (e.g. the probability of winning the game). In Monte Carlo (MC) simulations the approach is based on randomly sampling the search space and averaging the results. It has proved to be successful in many problems. In the game theory, it was proved that the process of MC simulation could accurately estimate the expected reward of an action [16]. The idea of combining Monte Carlo simulation with searching the tree was proposed for the first time in 2006 by Coulom [17]. The core idea of the proposed method is to take random samples in the search space of the given problem domain, and to build the search tree based on the results. The search tree itself is involved in the simulation, and its structure is constantly expanded by deepening those parts that seem to be more promising, based on MC estimates available at a given time. The whole process of MC tree search is usually described as consisting of four phases, which are described below.

Selection. After starting a new simulation run in the root of the tree, a child node is selected based on the so-called tree policy, and the selected node becomes the current node. This process is repeated until a leaf of the current tree is reached. Tree policy dictates which child node should be chosen for traversing next. It is crucial for the tree policy to balance between the exploitation of the most promising nodes (i.e. nodes with the biggest value of the expected reward) and exploration of other nodes, which might not have yet revealed their true potential.

Expansion. If the leaf that has been reached does not represent a final state, the tree is expanded by adding child nodes to the current leaf. Usually, all possible child nodes are added, and one of them is selected, possibly in a random way. In the future passes MCTS will prefer child nodes not yet visited or selected based on the tree policy (selection phase).

Simulation. If there are no further nodes to be visited at the current depth of the tree, the search process is simulated based on the so-called default policy, until the final state is reached and the value of the reward is known. The easiest way is to use a random simulation, although some simple, problem-specific rules may bring significant improvements. Using them may, however, decrease the simulation speed. During this step, no new nodes are added to the tree.

Backpropagation. The information about the received reward is backpropagated through all the nodes that were visited in the given simulation run, up until the root is reached and updated as well. Each node has to store at least information about the number of simulations that went through this node, and the average reward received from these simulations.

At the beginning, the search tree consists only of the root node. The tree is then built asymmetrically, as more promising nodes will be visited more often and new child nodes will be added in the more promising parts of the tree. Most researchers use a reward function, which assumes values from the [0, 1] range. The goal of MCTS is to find the decision, which maximizes the expected reward. It means that in the state described by the root node, the action leading to the root's child node maximizing the expected reward should be selected.
MCTS is well suited for problems in which the final state is guaranteed to be reached in a finite number of steps. It has several significant advantages. One of the most prominent ones is that there is no need to know any function to evaluate the intermediate states. This information is necessary only in the final state (e.g. after winning the game) and is then backpropagated through the nodes. This is unlike in the case of alpha-beta search. Also, MCTS can be stopped at any time and the current best solution can be given as a result. The general MCTS is presented as Algorithm 1. In the description, the term *terminal node* refers to the final state of the search (such as winning the game), and should not be confused with terminals in the Steiner problem.
The crucial part of the MCTS algorithm is the tree policy. The most widely applied approach is based on UCB proposed in [18] for a k-armed bandit problem. Each of the k arms of the bandit is defined by a random variable (interpreted as a reward), independent and identically distributed as the others, with an unknown expectation value. The goal is to develop (based on past rewards) a policy of selecting which bandit to play in order to maximize the reward. There is a straight-forward similarity to the problem of choosing a child node in MCTS. In [19] it was pro-

Algorithm 1: General MCTS

Input: root node *root*
Output: best child of *root*
while *enough computational resources* **do**
  leaf = SimulateWithTreePolicy(*root*)
  if *leaf is not terminal node* **then**
    $new\_node = \text{Expand}(leaf)$
    $reward = \text{SimulateWithDefaultPolicy}(new\_node)$
    $\text{UpdateTreeStatistics}(new\_node, root, reward)$
  else
    reward = Reward(*leaf*)
    $\text{UpdateTreeStatistics}(leaf, root, reward)$
  end
end
return $BestChild(root)$

posed to use UCB1 (a version of UCB) as the tree policy. Combing MCTS and UCB1 used as the tree policy results in an algorithm named Upper Confidence Bound for Trees (UCT). In this algorithm, in the selection step of the MCTS procedure, being in the current node, an i-th child node is selected to maximize the UCT value given as:

$$\text{UCT} = r_i + C\sqrt{\frac{2\ln n}{n_i}}, \tag{1}$$

where r_i is the average reward received in simulations going through i-th child node, n_i is the number of times i-th child node has been visited, n is the number of simulations going through the current (parent) node, and C is a constant value responsible for balancing between exploration and exploitation. In the case of UCT version of MCTS, reward value is expected to be within [0, 1]. From Eq. 1 it can be seen, that nodes with a higher average reward are preferred. However, their attractiveness decreases when they are visited many times (when n_i increases). It means that other nodes will be visited as well. In the special case when $n_i = 0$, this node is preferred before others. A significant finding of UCT is that given enough time and resources, the probability of selecting a suboptimal child node in the root converges to zero at a polynomial rate, with the number of simulations growing to infinity [19]. The UCT algorithm (i.e. Algorithm 1 with UCB1 used in "SimulateWithTreePolicy") will serve as a basis for the proposed algorithm for ESTP, as presented in Section 4.

3. Simple Heuristic for the Euclidean Steiner Tree Problem

In this section, the most important properties of Steiner trees is introduced and a measure of the quality of the candidate solutions is presented. Then, a special case of the problem for three terminal points is described. Based on its solution, a simple heuristic and a greedy search procedure are proposed to solve the ESTP for any number of terminals.

3.1. Basic Properties of Steiner Trees

Properties of Steiner trees have been studied for decades by many researchers. The readers interested in the detailed introduction are referred to [5]. Here, only the properties essential to the proposed algorithm are presented.

Property 1. Steiner minimal tree (SMT) for a set T of terminals has at most $p-2$ Steiner points. It is possible that no Steiner point (additional point) is necessary to connect the terminals with the minimum total length of edges. In this case, SMT equals the minimal spanning tree (MST) for the points in T.

Property 2. Degree property. In a Steiner tree, each Steiner point has exactly the degree of three (i.e. it is incident with exactly three edges).

Property 3. Angle property. Any two edges (from among three, see property 2), incident with the Steiner point, make exactly 120° with each other.

Figure 1 presents a minimal spanning tree and a Steiner tree. It can be observed that properties 1–3 are satisfied. It should be mentioned that a given set of Steiner points that satisfy the properties above, does not necessarily constitute an optimal solution. Properties 1–3 are the necessary conditions for a set of candidate Steiner points. It is possible to build a Steiner tree which satisfies the three properties and yet it is not the optimal tree.
The common criterion to compare the candidate solutions in ESTP is to calculate the so-called Steiner ratio for each candidate Steiner tree. The general procedure is as follows. Having a set of terminals T and a set of candidate Steiner points S, let $P = T \cup S$. Build the minimum spanning trees for T and for P. Then, the Steiner ratio is defined as:

$$\rho = \frac{L(MST_P)}{L(MST_T)}, \tag{2}$$

where $L(.)$ is the total sum of lengths of edges from the corresponding tree, MST_T and MST_P are the minimal spanning trees for the terminals and a set of points created by combining terminals with Steiner points, respectively. This criterion is minimized for the solution of the considered problem of finding the minimum Steiner tree. However, it can be calculated for any set of points S, even points which do not satisfy the required properties. For that reason, Steiner ratio is a useful criterion that can be used in any heuristic search algorithm.

3.2. Fermat Point

The task of finding the minimal Steiner tree was first introduced by Fermat as the problem of finding a point with a minimum distance from other three points that are given. Later the problem has been generalized by allowing any number of given points (terminals T) and any number of additional points (Steiner points S), while the task was to interconnect all terminals, achieving the minimum total length of all edges. The original question formulated by

Fermat can be given as follows. For a given set of three points on the Euclidean plane, A, B, and C, find point F (called Fermat point), such as for any other point F' the condition is satisfied:

$$|F'A| + |F'B| + |F'C| > |FA| + |FB| + |FC|. \quad (3)$$

This is, in fact, ESTP for a set of three terminals, and it is known from the property 1, that at most one Steiner point can be present in the optimal solution. This problem can be solved easily. Two situations are possible. If points A, B and C form a triangle in which all internal angles are smaller than 120°, then the only Steiner point is located at the Fermat point F of the triangle. The SMT is equivalent to the minimum spanning tree (MST) including three triangle points and the Fermat point. In the second case, if there is an internal angle equal to or greater than 120°, the Steiner point is incident with the triangle vertex (A, B or C) in which the two sides of the triangle meet at 120° degrees or more. The Steiner minimum tree (and the solution to the original Fermat problem at the same time) is the MST spanned on the three triangle points. The procedure of finding the Fermat point can be formulated in several ways. One of them can be given as follows.

Given three points A, B, and C on the Euclidean plane, if $\measuredangle ABC \geq 120°$ or $\measuredangle BAC \geq 120°$ or $\measuredangle ACB \geq 120°$, then return B, A or C as the Fermat point, respectively. Otherwise, select any two sides of the triangle ABC. Construct an equilateral triangle on each of the chosen sides. Find two lines, going through each new vertex of the constructed triangles and their opposite vertices of the original triangle. The Fermat point F is the point of intersection of the two lines.

3.3. Greedy Heuristic for ESTP

It can be observed that the Steiner tree is inherently connected with the procedure of building the MST. If the set of Steiner points is given, the Steiner tree can be found by calling one of the efficient algorithms for MST construction, such as the Prim algorithm [4]. However, this section proposes a simple heuristic, which considers the problem from another point of view.

The primary switch is to start the Prim algorithm without having the set of Steiner points given. Instead, the candidate Steiner points are added dynamically during the construction of the MST. The resulting procedure can be considered as a greedy optimization procedure. The main idea is based on the Prim algorithm. The Prim algorithm starts by selecting any of the given points. Then, in each step, a new point is added to the growing tree as the one with the minimum distance to any of the previously selected points. This step is repeated as long as all of the points are selected.

The proposed modification is as follows. The first two points are selected as in the original Prim algorithm. However, next steps are modified. A new point p is added to the growing MST by directly connecting it to its nearest neighbor p' or by connecting it through f (a Fermat point) to two points p' and p''. The choice is made in a greedy way, based on the total sum of lengths of edges in the partial tree. The formal description is given as Algorithm 2. The meaning of the correction steps is explained further in this section. In the algorithm, $|e|$ is the length of the corresponding edge e.

Algorithm 2: Greedy algorithm for Steiner tree

Input: Set of points T (terminals), *correct_all* (boolean value)

Output: Steiner tree ST for points in T, set of Steiner points SP

1. Set $E = \emptyset$ (E is the set of edges in ST), $SP = \emptyset$
2. Select any point p from T. Set $T' = \{p\}$ and $T = T \setminus \{p\}$
3. Select point p from T with the minimum distance to the point p' in T'. Set $T' = T' \cup \{p\}$ and $T = T \setminus \{p\}$. Add the edge (p, p') to E.
4. Select point p from T with the minimum distance to any point p' in T'.
5. Select any point p'' from T', so that $p'' \neq p'$ and $(p', p'') \in E$.
6. Calculate Fermat point f for the three points p, p' and p''.
7. **if** f is a valid Fermat point (i.e., not incident with p, p' or p'') and $|(p,p')| + |(p',p'')| > |(p,f)| + |(p',f)| + |(p'',f)|$ **then**
   remove (p', p'') from E,
   add (p,f), (p',f) and (p'',f) to E,
   set $T = T \setminus \{p\}$, set $T' = T' \cup \{p,f\}$, add f to SP,
   else
   add (p,p') to E, set $T = T \setminus \{p\}$, set $T' = T' \cup \{p\}$
   end
8. If $correct\_all = true$, correct solution (Algorithm 3)
9. **Repeat** from step 4 **until** $T \neq \emptyset$.
10. Correct solution (Algorithm 3).
11. Return $ST(T', E)$ and SP

3.4. Correction Procedure

After adding a new Fermat point as a Steiner point in Algorithm 2, some of the previously added Steiner points may no longer satisfy angle- and degree-related conditions required from all Steiner points. One example is presented in Fig. 2. Steiner point $F1$, inserted by the algorithm, satisfies the angle and degree properties, as it was calculated as the Fermat point for the three connected terminals (Fig. 2, step A). After the second Steiner point $F2$ is inserted (step B) as the Fermat point calculated for three points connected thereto, $F1$ no longer satisfies the angle condition. The situation is fixed using the proposed correction procedure, which is depicted in part C of Fig. 2. Note that during the correction procedure point $F2$ is also moved from its original position. The correction procedure is given as Algorithm 3. The main idea is to perform two actions repeatedly: 1) to remove all candidate

Algorithm 3: Correction procedure

Input: Set of points T (terminals), set of Steiner points SP (some of them can be invalid), ε – maximum allowed violation of angle condition for Steiner points.

Output: SP – set of valid Steiner points (i.e., satisfying the angle and degree conditions)

1. Let $P = T \cup P$ and MST_P be the minimum spanning tree for P.
2. Identify Steiner points in SP which have degrees not equal 3 in MST_P and remove them from SP; recalculate MST_P.
3. Repeat step 2 until no sp is removed in step 2.
4. Identify Steiner point sp in SP which violates the angle condition the most in MST_P. If this violation is more than ε, recalculate this point as a Fermat point of the triangle formed by the three points connected to sp (if the Fermat point is not valid, remove sp); recalculate MST_P.
5. If SP has been modified in step 4, go to step 2.
6. **Return** SP.

Steiner points which do not satisfy the degree-related condition, and 2) to recalculate the coordinates of those Steiner points which do not meet the angle-related condition (the recalculation is done by means of calculating new Fermat points). These two steps are repeated until no change is necessary.

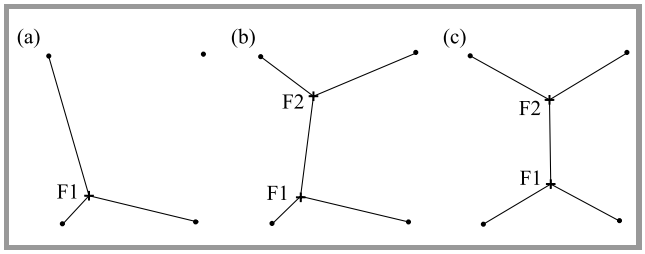

Fig. 2. Effect of the correction procedure.

A parameter *correct_all* is introduced in Algorithm 2 to limit the computational cost of the correction procedure. Only when it is set to $true$, the correction procedure will be called after adding each new Steiner point. Otherwise, it is called only at the end of the algorithm, after connecting all terminals. As revealed by tests (see Section 5), in practice, *correct_all* can be in most situations set to $false$ without decreasing the quality of the final solution significantly.

Algorithm 2 together with Algorithm 3 clearly constitute a greedy search procedure, as in step 7 of Algorithm 2, always locally better (shorter) connections are selected. This does not guarantee a global optimum. However, it can provide a feasible solution very fast. In fact, it is possible to run Algorithm 2 starting with each possible terminal as the starting point in this greedy procedure (point p selected in step 2 of Algorithm 2). Then, the best solution from all runs is chosen as the final solution. This simple greedy heuristic can be further extended by allowing not optimal local choices in step 7 of Algorithm 2.

4. Monte Carlo Tree Search for Euclidean Steiner Tree Problem

Algorithm 2 shares a common drawback with all greedy optimization algorithms. It may be trapped by local minima, as the choices that are optimal locally do not always lead to the global optimum. This section proposes two ways to relax the greediness and allow not optimal local choices with the hope to improve the solution.

4.1. Pure Monte Carlo Simulations for ESTP

The easiest way to relax the greediness of Algorithm 2 is to allow a random selection of one of the two options in step 7. The modified procedure is as follows. If the Fermat point f is not valid, connect the new point as in the original Prim algorithm; else, if the Fermat point is valid, randomly select (with equal probabilities) whether the new point is connected as in the original Prim algorithm, or through the Fermat point. After the last terminal point is connected, the found solution (Steiner tree and Steiner points) is remembered together with its Steiner ratio. The whole procedure is repeated for a given number of iterations. After each iteration, it is checked whether a better solution is found. Thus, each iteration is independent of others and can be easily parallelized. The proposed Monte Carlo simulation for ESTP is given as Algorithm 4. To simplify the notation, it is assumed that the set of terminals T is recovered to its original input state at the beginning of each iteration of the f or loop. Function RANDOM(0, 1) returns a random value from a uniform distribution from the [0, 1) range.

4.2. Hybridization of Prim Algorithm and MCTS

The drawback of Algorithm 4 is that the path leading to the solution (i.e. how the points were connected to the growing tree) is neither remembered nor used in subsequent iterations. In this section, it is proposed how the quality information about previous simulations can be utilized. The presented approach can be viewed as hybridization of the Prim algorithm and MCTS. It is important to clarify some potential ambiguities here. In fact, there are two trees in the proposed approach. The first one is the Steiner tree, which is the required solution of the problem at hand. During each iteration of Algorithm 4, a new Steiner tree is built and evaluated. The other tree in the proposed Algorithm 5 is the MCTS tree, which is a data structure used to remember the statistics about choices from previous iterations and to guide the search process.

To describe the proposed MCTS algorithm, one has to define the tree policy, the default policy, and the reward function, expanding the node and updating the statistics of the node. First, let us describe the proposed MCTS algorithm for ESTP in general. The simulation starts with only the root node in the MCTS tree. Expansion of the root node

Algorithm 4: Pure Monte Carlo simulation for Steiner tree

Input: Set of points T (terminals), *correct_all*, *number_of_simulations*

Output: Steiner tree ST_{best} for points in T, set of Steiner points SP_{best} corresponding to ST_{best}

1. $ST_{best} = MST_T$, $SP_{best} = \emptyset$, $\rho_{best} = 1$ (Steiner ratio)
2. **for** $i = 0$ *to number_of_simulations* **do**
   a. Set $E = \emptyset$ (E is the set of edges in ST), $SP = \emptyset$
   b. Select any point $p \in T$. Set $T' = \{p\}$, $T = T \setminus \{p\}$
   c. Select point $p \in T$ with the minimum distance to the point $p' \in T'$. Set $T' = T' \cup \{p\}$ and $T = T \setminus \{p\}$. Add the edge (p, p') to E
   d. Select point $p \in T$ with the minimum distance to any point $p' \in T'$.
   e. Select any point $p'' \in T'$, so that $p'' \neq p'$ and $(p', p'') \in E$.
   f. Calculate Fermat point f for the three points p, p' and p''.
   g. **if** f is a valid Fermat point (i.e., not incident with p, p' or p'') **AND** RANDOM$(0,1) < 0.5$ **then**
      remove (p', p'') from E,
      add (p, f), (p', f) and (p'', f) to E,
      set $T = T \setminus \{p\}$, set $T' = T' \cup \{p, f\}$, add f to SP,
   else
      add (p, p') to E, set $T = T \setminus \{p\}$, set $T' = T' \cup \{p\}$
   end
   h. **if** $correct\_all = true$ **then**
      correct solution (Algorithm 3)
   end
   i. **Repeat** from step 2.d **until** $T \neq \emptyset$.
   j. Correct solution (Algorithm 3).
   k. $\rho = L(MST_{T \cup SP}) / L(MST_T)$
   l. **if** $\rho < \rho_{best}$ **then**
      $ST_{best} = MST_{T \cup SP}$, $SP_{best} = SP$, $\rho_{best} = \rho$
   end
end
3. **Return** ST_{best} and SP_{best}

is performed by selecting the first terminal as in step 2b in Algorithm 4. Thus, the number of branches from the root node equals the number of terminals. On all following levels of the MCTS tree, expanding the node creates two branches at the most, one is for direct connection of the new terminal (as in the Prim algorithm), while the other for connection through the newly created Steiner point. In these cases in which the Steiner point cannot be created as a valid Fermat point, only one branch is created. While descending from the root, at each existing node, one of the two possible branches is selected based on the accumulated statistics in that node according to the common tree policy based on UCT (Eq. 1) described in Section 2. Thus, the process is performed differently than in step 2g of Algorithm 4. On the other hand, the default policy of the proposed MCST for ESTP is done exactly as in step 2g of Algorithm 4. It means, that when there are no statistics available on the current level of the MCTS tree, the new terminal is connected to the growing Steiner tree directly

Algorithm 5: UCT for Euclidean Steiner tree problem

Input: Set of points T (terminals), *correct_all*, *number_of_simulations*

Output: Steiner tree ST_{best} for points in T, set of Steiner points SP_{best}

1. $ST_{best} = MST_T$, $SP_{best} = \emptyset$, $\rho_{best} = 1$ (Steiner ratio of the best Steiner tree found), $root = \emptyset$
2. **for** $i = 0$ *to number_of_simulations* **do**
   a. Set $E = \emptyset$ (E is the set of edges in ST), $SP = \emptyset$

   Tree policy

   b. If $root = \emptyset$, expand *root*, the number of child nodes equals the number of terminals $|T|$. Each child node is associated with one terminal.
   c. Select *current_node* as the child node of the *root* maximizing its UCT value, giving priority to those child nodes which have not been visited yet. Set $T' = \{p\}$ and $T = T \setminus \{p\}$, where p is the terminal associated with *current_node*.
   d. **Repeat until** *current_node* is not a leaf in MCTS tree: Select *new_node* as the child node of *current_node* which maximizes its UCT value, giving priority to child nodes not yet visited. Let p be the terminal connected to the growing Steiner tree in a way described by node *new_node*. **If** *new_node* describes direct connection, set $T' = T' \cup \{p\}$ and $T = T \setminus \{p\}$, add the edge (p, p') to E, where p' is the point from T' closest to p (remembered in *new_node*); **Else If** *new_node* describes connection through Fermat point f, remove (p', p'') from E, add (p, f), (p', f) and (p'', f) to E, set $T = T \setminus \{p\}$, set $T' = T' \cup \{p, f\}$, add f to SP, where p'' has the same interpretation as in Algorithm 4, and is remembered in *new_node* together with p'. Set $current\_node = new\_node$. Correct solution if $correct\_all$ is $true$.

   Expanding

   e. **If** *current_node* is a leaf in MCTS tree, expand *current_node* as follows. Find point p in T with the closest distance to any point p' in T'. Select any point p'' from T', so that $p'' \neq p'$ and $(p', p'') \in E$. Create two child nodes of *current_node* in MCTS tree, one for the direct connection of p to p', the second for the case when p is connected to p' and p'' through f (Fermat point for points p, p' and p''), if f is valid.

   Default policy

   f. Execute steps 2d–2l from Algorithm 4 (it can update ST_{best} and SP_{best}).

   Update MCTS tree statistics

   g. Calculate reward value based on the Steiner ratio of the Steiner tree that has been built during this iteration: $reward = RewardFunction(\rho)$
   h. Update UCT value of each node in MCTS tree that has been visited during this iteration (according to Eq. 1).
end
3. **Return** ST_{best} and SP_{best}

or through a new Steiner point, as long as there is a valid Fermat point. The choice is made randomly with the probability of selecting either option equaling 0.5. Each iteration ends when the last terminal is connected to the Steiner tree being built during this iteration. The Steiner ratio is calculated, and its value can be used as the reward. However, it needs to be converted, as the original UCT algorithm requires the reward to be in the range of [0, 1]. Several possible ways of performing this conversion are described in the next section. Having the reward in the proper form makes it possible to use it for updating the UCT value of nodes in the MCTS tree according to Eq. 1. The proposed MCTS for ESTP is described more formally as Algorithm 5.

4.3. Reward Functions

The role of the reward functions proposed here is to provide a mapping from the Steiner ratios to the reward values that can be used in MCTS. In ESTP, the Steiner trees are evaluated using the Steiner ratio ρ, while in MCTS, there is a need to assign reward values for the leafs which are from the range of [0, 1]. Additional steps must be taken to satisfy this requirement. For the minimum spanning tree for the set of terminals, the Steiner ratio equals 1. Having any feasible solution of ESTP means that the Steiner ratio $\rho < 1$. It was proved [5], that for problems in the 2D Euclidean space, the minimum possible value of ρ is $\rho_{opt} = \frac{\sqrt{3}}{2} \approx 0.866$. Thus, is possible to define a mapping from the calculated ρ to the reward value that can be used in the UCT algorithm. The main idea is to assign high rewards (close to 1) for solutions with ρ close to ρ_{opt}, and low rewards (close to 0) for solutions with ρ close to 1. Thus, it is enough to define the reward functions on the domain $[\rho_{opt}, 1]$ as $reward(\rho) : [\frac{\sqrt{3}}{2}, 1] \rightarrow [0, 1]$. The proposed reward functions are as follows:

- $reward_{linear}(\rho) = \frac{1-\rho}{a}$ with example value of parameter $a = 7.45$, $reward_{Gauss}(\rho) = e^{-\frac{(\rho_{opt}-\rho)^2}{\sigma}}$ with example value of parameter $\sigma = 0.005$,
- $reward_{sigmoidal}(\rho) = 2 - \frac{2}{1+e^{-\beta(\rho-\rho_{opt})}}$ with example value of parameter $\beta = 40$,
- $reward_{log}(\rho) = 1 - a_2 \ln(1 + a_1(\rho - \rho_{opt}))$ with example value of parameters $a_1 = 1000$ and $a_2 = 0.204$,
- $reward_{cos}(\rho) = \cos\left(c(\rho - \rho_{opt})\right)$ with example value of parameter $c = 11.725$.

These functions differ in how they promote the decrease in the Steiner ratio. For example, the cosine reward function adds the reward faster for Steiner ratios close to 1 than it is the case with the linear function. The usefulness of these reward functions is verified, using numerical experiments, in the next section.

5. Experimental Studies

In this section, the results of the proposed algorithms applied to problems of various sizes are presented. Randomly generated problems, as well as problems with known optimal solutions from OR-Lib [15], are considered. The Friedman test (with Shaffer post hoc tests) is used, to compare the results. The testing procedure is as follows. Only the average results for each problem of each algorithm are considered by the Friedman test. All problems of a given size and all algorithms are taken up at once to discover potential differences among the average behaviors of the algorithms. Then, if any differences are found, a set of pair-wise post-hoc comparisons between algorithms is calculated.

The Friedman test is recommended where one wants to compare the general behavior of more than two algorithms on several test problems. This two-step procedure is regarded as a better approach than performing many pair-wise comparisons, such as the Wilcoxon test, which would increase the overall probability of making at least one type I error, which would erroneously discover differences between algorithms which, in fact, perform equally in general. More details about the testing procedure adopted can be found in [20]. All the tests were calculated using the Keel software package [21].

5.1. Results for Random Problems

Preliminary tests. In regard to random problems, for each problem size, twenty random problems are generated from a uniform distribution on $[0, 1]^2$. Each algorithm is run ten times on each problem. In the Friedman test, only the mean values of Steiner ratios are compared.

In the presentation of the results the following abbreviations are adopted:

- greedy1 – Algorithm 2 run $|T|$-times using each terminal point as the starting point in step 2 in Algorithm 2 and the parameter *correct_all* set to *true*;
- greedy2 – the same as algorithm greedy1 except the parameter *correct_all* is set to *false*;
- MC10k, MC100k – pure Monte Carlo simulation for Steiner tree (Algorithm 4) run with 10,000 and 100,000 iterations, respectively,
- UCT10k and UCT100k – UCT algorithm (Algorithm 5) with the linear reward function run with 10,000 and 100,000 iterations, respectively.

In the preliminary tests, the algorithms have been first applied to random problems with 20, 30, 40 and 50 terminals. The goal was to check how the ideas introduced scale with the number of terminals. Only the linear reward function is used and the $C = 1$ in Eq. 1. For the sake of space, only the conclusions, and not the detailed results, are provided here. For problems with 20 terminals, the *p-value* calculated from the Friedman test was 0.884. It means that the test does not discover any significant differences among the algorithms. The results of all algorithms are similar. For problems with 30 terminals, UCT and MC algorithms with 100k iterations were ranked as best. The *p-value* from the Friedman test was 0.00042, which means that significant

differences exist in the performance of the algorithms and Shaffer's post hoc procedures were calculated. They discovered significant differences between UCT100k and MC100k algorithms and UCT10k and MCT10k algorithms showing that 10k iterations are not enough for MCT and UCT algorithms. However, no significant difference was discovered between MCT100k and UCT10k, which can be interpreted as the first sign of the advantage of UCT over MC. Additionally, there were no significant differences discovered between any of the greedy and the winning UCT100k and MC100k algorithms.

Results for problems with 40 and 50 terminals showed that the Friedman *p-values* keep decreasing, demonstrating that the differences are more clear as the problems become more difficult. The proposed UCT algorithm with 100k iterations was always ranked as the best. However, considering Shaffer post-host comparisons, significant differences between UCT and MC algorithms are not discovered when given the same number of iterations (10k or 100k). The important difference between the winning UCT100k and the greedy 2 algorithm was signaled only once. On the other hand, the greedy algorithms perform significantly better than MC10k and UCT10k. It can be summarized that the majority of differences discovered in the average behavior of the algorithms stem from the poor performance of MC10k and UCT10k algorithms. Another important observation is that the greedy2 algorithm has never been significantly worse than greedy1. Based on this, in the remaining experiments, partial solutions found after each iteration of MC and UCT algorithms are also not corrected, and only the final solution is corrected.

Scaling the reward function and checking the influence of parameter C (50 terminals). There are two important parameters for UCT that may influence the performance of UCT, the parameter C from Eq. (1) and the reward function. C is responsible for tuning the importance of exploration vs. exploitation, thus setting it correctly is crucial. A set of 20 random problems with 50 terminals is considered here.

In this set of experiments, only the greedy1 algorithm is considered. All UCT algorithms and the MC algorithm were allowed to iterate over 100k iterations. The value of C is given after the underscore, i.e. $UCT\_C0.5$ means that the UCT algorithm with the linear reward function has been run with $C = 0.5$. The algorithm abbreviated UCT in the following results is the algorithm used in the previous set of experiments ($C = 1$ and linear reward function). However, algorithms $UCT\_C1.0$, $UCT\_C2.0$, $UCT\_C0.5$ and $UCT\_C0.1$ have been further modified by scaling the linear reward function. It has been observed that the best solutions found had never had the Steiner ratio lower than 0.96. The theoretical minimum for the Steiner ratio is $\rho_{opt} = \frac{\sqrt{3}}{2} \approx 0.866$, it is, however, only achieved for some special cases of regular grids of terminals. In the case of real life problems, the Steiner ratio for a given problem would be higher. In such cases, the reward function introduced and defined over the range $[\rho_{opt}, 1]$, does not use the full range of the domain. The idea here is to limit the domain of the reward function. The linear reward function has been scaled, i.e. its domain is only $[0.96, 1]$ and the parameter $a = 25$.

The Friedman ranks presented in Table 1, and the small *p-value* shows that, in fact, the setting of parameter C has a huge impact on the performance of the proposed UCT algorithm. The pair-wise comparisons revealed that by setting C to a proper value, the UCT algorithm performs significantly better than MC. Also, it can be observed that setting $C = 2$ significantly decreases the performance of the UCT algorithm. There were no significant differences discovered between the best-performing algorithms ($UCT\_C0.5$ and $UCT\_C0.1$) and the greedy algorithm. However, when the numbers of best solutions found are considered (showing the ability of the algorithms to find best-known solutions for the problems), it can be stated that the UCT algorithm with the proper value of C should be preferable. The influence of the scaling of the reward function is not clear from these experiments. On the one hand, the UCT version abbreviated simply as UCT (no scaling, $C = 1$) is ranked lower than any of the modified UCTs, except for the UCT with $C = 2$. On the other hand, no significant differences between UCT and any of the modified UCT algorithms have been discovered. To further investigate this issue, additional experiments have been calculated, as presented in the later part of this section.

Autoscaling the reward function (problems with 70 terminals). In this set of experiments, 20 random problems with 70 terminals are considered. Here we test the influence of the different reward functions introduced earlier. The number of MC and UCT algorithm iterations exceeded 150,000. All UCT algorithms use parameter $C = 0.1$. Algorithms abbreviated as *lin*, *cos*, *Gauss*, *log*, and *sigm* are the algorithms with reward functions described in Subsection 4.3. Algorithms with suffix *_sc* in their name are the algorithms with the scaled versions of the functions. The scaling procedure is more elaborate here. Instead of assuming one new lower limit for the Steiner ratio, the idea is to scale the reward functions based on the Steiner ratio of the best-known solution for each problem separately. In practice, the *greedy*1 algorithm is run as the first one for a given problem, and the Steiner ratio ρ_{greedy} of the solution found by this algorithm is used to calculate the new lower limit for the domain of the reward function used in UCT algorithms. In fact, ρ'_{opt} is calculated as $\rho'_{opt} = 0.995 \cdot \rho_{greedy}$. Then ρ'_{opt} is used to define the new scaled versions of the reward functions by calculating proper parameter values. The parameters of the scaled versions of the reward functions are calculated as follows:

- $a = \frac{1}{1-\rho'_{opt}}$ for the linear function,
- $\sigma = \frac{-(\rho'_{opt}-1)^2}{\ln 0.04}$ for the Gaussian function,
- $\beta = \frac{\ln(1/0.98-1)}{\rho'_{opt}-1}$ for the sigmoidal function,

Table 1
Friedman ranks and numbers of best solutions found for problems with 50 terminals. Entries marked in bold indicate algorithms which performed significantly worse than the best performing one; $p\text{-}value = 0.00225$

Algorithm	UCT_C0.5	UCT_C0.1	greedy 1	UCT_C1.0	UCT	UCT_C2.0	MC
Ranking	2.6	3.35	3.9	3.9	4.25	**4.6**	**5.4**
Best solutions	9	10	3	7	9	7	3

Table 2
Friedman ranks and numbers of best solutions found for problems with 70 terminals. Entries marked in bold indicate algorithms which performed significantly worse than the best performing one; $p\text{-}value = 6 \cdot 10^{-11}$

Algorithm	lin	lin_sc	cos	Gauss	log_sc	greedy 1	log	sigm_sc	cos_sc	sigm	MC	Gauss_sc
Ranking	2.8	3.2	3.65	4.05	5.55	**7.5**	**7.55**	**7.55**	**8.6**	**9**	**9.2**	**9.35**
Best solutions	7	7	7	6	3	4	2	2	3	0	1	3

- $a_2 = \frac{0.96}{\ln(1+(1-\rho'_{opt})a_1)}$,
- $a_1 = 1000$ for logarithm based function, $c = \frac{\pi}{2}(1-\rho'_{opt})$ for cosine function.

These derivations have been based on the assumptions, that the respective reward function should have values close to 0 and 1 at the limits of its domain.

Friedman ranks presented in Table 2 show that there are significant differences among the algorithms with different reward functions. Several important observations can be made based on pair-wise comparisons. MC and greedy algorithms are outperformed by the UCT algorithm, provided that the reward function is properly selected. UCT with linear, linear scaled, and cosine reward functions are significantly better than greedy and MC algorithms. UCT with the Gaussian reward function is also significantly better than MC. The superiority of UCT with these functions is also visible when considering the numbers of best solutions found.

On the other hand, MC and greedy algorithms are not outperformed by all of the UCT configurations, which shows that the proper choice of the reward function is crucial. At this point, a simple linear function seems to be the best option. Not only does it perform best, but it is also faster to calculate than the nonlinear functions. Surprisingly, scaling is never beneficial. Even if the scaled version of UCT is ranked higher than the original version, the difference is never significant. To explain this, let us ask the question about how fast a given algorithm finds the best solution in a given run (i.e. in which iteration). The statistics over all runs on all 20 problems with 70 terminals have been collected. It was observed, that UCTs based on scaled reward functions tend to converge prematurely, reaching the final solution early.

All results presented thus far can be summarized as follows. The proposed UCT algorithm depends on the proper choice of the parameter C and the reward function. When properly set, the proposed UCT algorithm performs significantly better than the greedy heuristic and the simple MC simulation. On the other hand, when not configured correctly, UCT may perform poorly. A good choice for C is 0.1, and a simple linear reward function can be recommended. The superiority of the UCT algorithm is becoming apparent when the dificulty of the considered ESTP grows.

5.2. Results for OR-Lib Problems

In this section, it is shown that the proposed algorithm is, in fact, able to find good quality solutions for ESTP. A set of benchmark problems from the OR-Lib library is used [15]. The optimal solutions for these problems are known and can be used to assess the results achieved by the proposed techniques. In the tests, the following files from OR-Lib have been used: estein10, estein20, estein30, estein40, estein50, estein60, estein70, estein80, estein90 and estein100. Those files have been taken from http://people.brunel.ac.uk/∼mastjjb/jeb/orlib/esteininfo.html. Each file contains 15 problems with the number of terminals corresponding to the number in the name of the file. The greedy algorithm (greedy1), MC and several UCT algorithms have been considered. The number of iterations in MC and UCT algorithms has been set to 250k. For UCT algorithms, $C = 0.1$. All reward functions have been used, however, without scaling.

All UCT and MC algorithms have been run ten times for each problem. Friedman ranks have been calculated as previously. The results obtained by the algorithms have been compared with the optimal solutions taken from the OR-Lib library.

For each algorithm, and for each problem, the best result from among ten runs is selected as the Steiner ratio ρ_{best}. Then, the relative error is computed as

$$err = \frac{\rho_{best} - \rho_{opt}}{\rho_{opt}} \cdot 100\%,$$

where ρ_{opt} is the Steiner ratio of the optimal solution for a given problem, taken from OR-Lib. For each algorithm, such errors were calculated for each of the 15 problems from the group of problems of a given size. The minimum, maximum, mean and standard deviation of these relative error values have been calculated. The whole procedure has been repeated for each file from OR-Lib,

Table 3
Friedman ranks for problems from OR-Lib with 90 terminals; $p\text{-}value = 6.4 \cdot 10^{-9}$

Algorithm	UCT_linear	UCT_cos	UCT_gauss	greedy 1	UCT_log	UCT_sigm	MC
Ranking	1.6	2.7333	3.2	**4.2667**	**4.5333**	**5.4667**	**6.2**

Table 4
Friedman ranks for problems from OR-Lib with 100 terminals; $p\text{-}value = 1.03 \cdot 10^{-6}$

Algorithm	UCT_lin	UCT_cos	greedy1	UCT_gauss	UCT_log	UCT_sigm	MC
Ranking	2.1333	2.6667	3.6	3.6	**4.5333**	**5.6667**	**5.8**

i.e. each group of 15 problems of a given size. For the sake of space, detailed Friedman ranks for problems with 90 and 100 terminals are only presented in Tables 3 and 4. Entries marked in bold indicate algorithms which performed significantly worse than the best performing one. For problems of all sizes only the summary and conclusions are given below.

The proposed algorithms can provide near optimal solutions for known problems from the OR-Lib library. On average, the best solutions found by the algorithms are never by more than 1% worse than the optimal ones, and UCT algorithms have shown to be able to provide the best results. Starting with the problems with 20 terminals, in the worst cases the best solutions found by the algorithms were never by more than 2% worse than the optimal ones (except for 2.2% for the greedy1 algorithm in the case of size 80 problems). In the case of problems with ten terminals, those errors exceeded 2% (they were lower than 3%) due to the existence of one EST problem which turned out to be difficult for all the algorithms. For all OR-Lib problems, Friedman ranks were consistent with the results from the previous section. For smaller problems, the average performance is not distinguishable. When the number of terminals increases, the UCT algorithm with a proper reward function can provide better results on average. For problems with ten terminals, all algorithms performed similarly, according to the Friedman test. For problems with 20 terminals, the Friedman test discovered significant differences ($p\text{-}value = 0.0174$), however, post-hoc procedures were not able to detect the sources of the differences.

For bigger problems (from 30 terminals up), significant differences were discovered. There are two sources of the differences. First, when the number of terminals grows, MC and greedy algorithms are consistently ranked lower than an instance of the UCT algorithm. The differences are not always discovered as significant, according to Shaffer's post-hoc procedures, showing that the proposed greedy approach is also interesting on its own. It loses, however, when the best solutions found are analyzed. The second source of the differences among algorithms is the selection of the reward function for the UCT algorithm. The results confirm that the choice of the reward function is crucial and there is no clear winner. The linear reward function is a good choice for problems with a bigger number of terminals. On the other hand, it performed worse for smaller problems. For example, for problems with 30 terminals it performed significantly worse than UCT with the sigmoidal reward function. The cosine function performed poorly for smaller problems, on the other hand, it provided, on average, the best solutions for problems with 100 terminals. This justifies the use of different reward functions. However, the linear reward function proves to be a good choice as computational costs are lower than for other reward functions.

Due to the limitations of this publication, a more detailed elaboration of the results is available from the author upon request.

6. Conclusions

A novel MCTS algorithm for ESTP on a plane has been proposed. A simple greedy heuristic has been introduced based on the Prim algorithm and the calculation of the Fermat point. It was utilized in MC simulations and the UCT-based algorithm, the most popular version of MCTS. To the best of the author's knowledge, it is the first time when the MCTS algorithm has been adapted for ESTP. Several reward functions have been proposed for use with UCT. The numerical experiments showed a high importance of the proper selection of the reward function. The proposed UCT algorithm can provide better results than simple MC simulations and greedy algorithms tested. It was shown, based on the benchmark problems from OR-Lib, that the proposed algorithms could provide near optimal solutions. The future research direction is to adapt the proposed MCTS to problems with a higher number of dimensions.

References

[1] D. Silver *et al.*, "Mastering the game of Go with deep neural networks and tree search", *Nature*, vol. 529, pp. 484–489, 2016.

[2] C. Browne and E. Powley, "A survey of Monte Carlo tree search methods", *IEEE Trans. on Intell. and AI in Games*, vol. 4, no. 1, pp. 1–49, 2012.

[3] A. Auger and O. Teytaud, "Continuous lunches are free plus the design of optimal optimization algorithms", *Algorithmica*, vol. 57, no. 1, pp. 121–146, 2010.

[4] R. C. Prim, "Shortest connection networks and some generalizations", *The Bell System Tech. J.*, vol. 36, pp. 1389–1401, no. 6, 1957.

[5] M. Brazil and M. Zachariasen, *Optimal Interconnection Trees in the Plane. Theory, Algorithms and Applications*, 1st ed. Springer, 2015.

[6] E. Miguez, J. Cidras, E. Diaz-Dorado, and J. L. Garcia-Dornelas, "An Improved Branch Exchange Algorithm for Large Scale Distribution Network Planning", *IEEE Power Engineering Review*, vol. 22, pp. 58–58, 2002.

[7] M. M. M. Brazil, C. C. Ras, and D. D. Da Thomas, "Relay augmentation for lifetime extension of wireless sensor networks", *IET Wirel. Sensor Sys.*, pp. 1–29, 2013.

[8] R. P. Mondaini and N. V. Oliveira, "Intramolecular structure of proteins as driven by Steiner optimization problems", in *Proc. 18th Int. Conf. on Syst. Engin. ICSEng 2005*, Las Vegas, NV, USA, 2005, vol. 1, pp. 490–491.

[9] Z. A. Melzak, "On the problem of Steiner", *Canadian Mathem. Bull.*, vol. 4, pp. 143–148, 1961.

[10] P. Winter and M. Zachariasen, "Euclidean Steiner minimum trees: An improved exact algorithm", *Networks*, vol. 30, no. 3, pp. 149–166, 1997.

[11] D. Trietsch and F. Hwang, "An improved algorithm for Steiner trees", *SIAM J. on Appl. Mathem.*, vol. 50, no. 1, pp. 244–263, 1990.

[12] W. D. Smith, "How to find Steiner minimal trees in euclideand-space", *Algorithmica*, vol. 7, no. 1–6, pp. 137–177, 1992.

[13] V. L. do Forte, F. M. T. Montenegro, J. A. de M. Brito, and N. Maculan, "Iterated local search algorithms for the Euclidean Steiner tree problem in n dimensions", *Int. Trans. in Oper. Res.*, vol. 23, no. 6, pp. 1185–1199, 2015.

[14] J. Barreiros, "A hierarchic genetic algorithm for computing (near) optimal Euclidean Steiner trees", in *GECCO 2003: Proceedings of the Bird of a Feather Workshops, Genetic and Evolutionary Computation Conference*, A. M. Barry, Ed. Chigaco: AAAI, 2003, pp. 56–65.

[15] J. E. Beasley, "OR-Library: distributing test problems by electronic mail", *J. of the Oper. Res. Soc.*, vol. 41, no. 11, pp. 1069–1072, 1990.

[16] B. Abramson, "Expected-outcome: A general model of static evaluation", *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 2, pp. 182–193, 1990.

[17] R. Coulom, "Efficient Selectivity and Backup Operators in Monte-Carlo Tree Search", in *Computers and Games: 5th International Conference, CG 2006, Turin, Italy, May 29–31, 2006. Revised Papers*, H. J. van den Herik, P. Ciancarini, and H. H. L. M. Donkers (Jeroen), Eds. Berlin Heidelberg: Springer, 2007, pp. 72–83.

[18] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem", *Machine Learning*, vol. 47, no. 2–3, pp. 235–256, 2002.

[19] L. Kocsis and C. Szepesvári, "Bandit based Monte-Carlo planning", in *Machine Learning: ECML 2006. 17th European Conference on Machine Learning Berlin, Germany, September 18–22, 2006 Proceedings*, J. Fürnkranz, T. Scheffer, and M. Spiliopoulou, Eds. Berlin Heidelberg: Springer, 2006, pp. 282–293, 2006.

[20] J. Derrac, S. Garcia, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms", *Swarm and Evol. Comput.*, vol. 1, no. 1, pp. 3–18, 2011.

[21] J. Alcala-Fdez, A. Fernandez, J. Luengo, J. Derrac, S. Garcia, L. Sanchez, and F. Herrera, "KEEL data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework", *J. of Multiple-Valued Logic and Soft Comput.*, vol. 17, no. 2–3, pp. 255–287, 2011.

Michał Bereta received his B.Eng. degree in Information Technology from Jyväskylä Polytechnic (Finland) in 2003 and his M.Sc. degree in Computer Modeling from Cracow University of Technology in 2004. He received his Ph.D. degree in Computer Science from the Institute of Computer Science of the Polish Academy of Sciences in 2008. Between 2010–2012 he was a Postdoctoral Fellow at the Department of Electrical and Computer Engineering, University of Alberta. He works at the Institute of Computer Science of the Cracow University of Technology. His research interests include computational intelligence, machine learning, data mining and pattern recognition.

E-mail: mbereta@pk.edu.pl
Faculty of Physics, Mathematics and Computer Science
Tadeusz Kościuszko Cracow University of Technology
24 Warszawska st
31-155 Cracow, Poland

Paper

Dynamically-adaptive Weight in Batch Back Propagation Algorithm via Dynamic Training Rate for Speedup and Accuracy Training

Mohammed Sarhan Al_Duais and Fatma Susilawati Mohamad

Department of Information Technology, Universiti Sultan Zainal Abidin, Terengganu, Malaysia

https://doi.org/10.26636/jtit.2017.113017

Abstract— The main problem of batch back propagation (BBP) algorithm is slow training and there are several parameters need to be adjusted manually, such as learning rate. In addition, the BBP algorithm suffers from saturation training. The objective of this study is to improve the speed up training of the BBP algorithm and to remove the saturation training. The training rate is the most significant parameter for increasing the efficiency of the BBP. In this study, a new dynamic training rate is created to speed the training of the BBP algorithm. The dynamic batch back propagation (DBBPLR) algorithm is presented, which trains with adynamic training rate. This technique was implemented with a sigmoid function. Several data sets were used as benchmarks for testing the effects of the created dynamic training rate that we created. All the experiments were performed on Matlab. From the experimental results, the DBBPLR algorithm provides superior performance in terms of training, faster training with higher accuracy compared to the BBP algorithm and existing works.

Keywords— artificial neural network (ANN), batch back propagation algorithm, dynamic training rate, speed up training, accuracy training.

1. Introduction

The batch back propagation (BBP) algorithm is commonly used in many applications, including robotics and automation. It has been used successfully in neural network training with a multilayer feed-forward network [1], [2]. The BP algorithm led to a tremendous breakthrough in the application of multilayer perceptions [3]. This method has been applied successfully in applications in many areas, and has an efficient training algorithm for the multilayer perception [4], [5]. Gradient descent is commonly used to adjust the weight through a change the error training, but it is not guaranteed to find the global minimum error, because the training is slow and converges easily to a local minimum [6]–[8].

The main problem of the BP algorithm is slow training; it requires a long learning time to obtain a result and there are several parameters that need to be adjusted manually, with highest saturation training [9], [10].

Current research on solving the slow training of the back-propagation algorithm is focused on the adaptation of parameters like the training rate that controls the weight adjustment along the descent direction [11]. We have improved the speed of the back propagation algorithm through adapting the training rate [12]. A new algorithm uses the square error function with a penalty for escaping from local minima. The weight is updated beside the penalty and the relationship between the training rate and penalty. The training rate is a fixed learning rate at 0.013 and the penalty parameter is set as 0.001. The results are compared with (standard back propagation) SBP.

The remaining portion of this paper is organized as follows. In Section 2 related work is presented. Section 3 presents proposed methods, while Section 4 shows experimental results. Section 5 covers discussion to validate the performance of the improved algorithm. Finally, Section 6 contains the conclusions.

2. Related Works

Abbas in [13] proposed a novel back propagation algorithm of ANN NBPNN that has a self-adaptive training rate. The experimental results show that NBPNN gave a more accurate result than the BP algorithm. In [14] a specific penalty to obtain the proportion of the norm of the weight or to prove the boundedness of the weights in the network training process is presented. The learning rate is set by an equation to be a small constant or an adaptive series. The initial weight is chosen in the range $[-0.5, 0.5]$; and the training rate is fixed to be a small constant: 0.05 or an adaptive series. The penalty factor is set as 0.001. The results show better convergence compared to existing work.

Authors in [15] improved the batch BPAP algorithm through their proposed dynamic training rate with a penalty. The structure of the algorithm is 2:2:1, using the sigmoid as the activation function. The weight was updated in the batch BPAP algorithm with bounded during training. From the experimental result, BPAP reaches a global minimum after the 1000th iteration. [16] provides the dynamic BP algorithm for training with a boundary. In this case,

the weight is updated under the effect of this boundary. The sigmoid function is used as the activation function. The boundary helps the BP algorithm for control of the weight update.

3. The Proposed Method

The data set is very important for verification to improve the BBP algorithm. In this study, all data are taken from UCI Machine Learning Repository [17].

3.1. Neural Network Model

The proposed ANN model is a three-layer neutral network that has an input layer, hidden layer and output layer. The input layer is considered to be $\{X_1, X_2, \ldots, X_i\}$, which represents the nodes. The nodes depend on the data types or attributes. The hidden layer is made of two layers of four nodes. The output layer is made of one layer with one neuron. Three biases, two of them, which is denoted by u_{0j}, v_{0k} and w_{0r}. Finally, the sigmoid function is employed as an activation function, which is linear for the output layer [18]. The neural network can be defined as I, T, W, A, where I denotes the set of input nodes and T denotes the topology of NN, which covers the number of hidden layers and the number of neurons. The set of weights by the activation function is as follows:

- L_h – first hidden layer for neuron h, $h = 1, \ldots, q$,
- LL_k – second hidden layer for neuron j, $j = 1, \ldots, p$,
- Y_r – output layer for neuron r,
- u_{ih} – the weight between neuron I in the put year and neuron h in the hidden layer,
- u_{0h} – the weight of the bias for neuron j,
- v_{hj} – the weight between neuron h from hidden layer z and neuron j from the hidden layer LL,
- v_{0j} – the weight of the bias for neuron j,
- w_{jr} – the weight between neuron k from the hidden layer LL and neuron r from the output layer L,
- w_{0r} – the weight of the bias for neuron r from the output layer,
- Δw – the difference between the current and new value in the next iteration,
- γ – the manual of training rate,
- γ_{dmic} – the dynamic training rate,
- $|e|$ – an absolute value of the error training,
- BBP – batch back propagation algorithm,
- DBBPLR – dynamic batch back propagation algorithm with dynamic training rate.

3.2. Dynamic Training Rate

One way to escape the local minimum and save training time in the BBP algorithm is by using a large value of γ in the first training. On the contrary, a small value of γ leads to slow training, but a smaller value of γ leads to the BBP algorithm having a slow convergence [19], [20]. Even a large γ is unlikely for training the BBP algorithm. The weight update between neuron k from the output layer and neuron j from the hidden layer is as follows:

$$\Delta w_{jk}(t+1) = w_{jk}(t) - \gamma \frac{\partial E}{\partial W_{jk}(t)}, \tag{1}$$

where $\Delta w_{jk}(t)$ is a weight change, the weight is updated for each epoch from Eq. 1, and slow training or fast training depends on a parameter that affects the updating of the weight. To enhance the BBP algorithm, which is given by the Eq. 1, to avoid the local minima and to avoid saturation training, we created the dynamic training rate:

$$\gamma_{\text{dmic}} = \sec(Y_r) + (1+k)^{|e|}, \tag{2}$$

where k is an average of the activation function, in this study is a sigmoid function.
The main idea is to keep the value of dynamic training rate positive for every epoch, to avoid the vibration of the value of training error e. We substitute γ_{dmic} from Eq. 2 into Eq. 1 to obtain:

$$\Delta w_{jk}(t+1) = w_{jk}(t) - \left[\sec(Y_r) + (1+k)^{|e|}\right] \frac{\partial E}{\partial W_{jk}(t)}. \tag{3}$$

The weight update is automatic for every layer under the effect of the dynamic training rate γ_{dmic}.

3.3. DBBPLR Algorithm

There are three stages of training BBP algorithm: forward phase, backward phase and feedback phase. In the feed-forward phase, each input unit x_i receives an input signal x_i and broadcasts this signal to the next layer until the output layer of the system. The backward pass phase is starting when the output of the last hidden layer reaches to end step then the start. The goal of the BBP algorithm is to get the minimum error training between the desired output and actual data, the all steps recorded as follows:

$$e_r = \sum_{r=1}^{n} (t_r - Y_r). \tag{4}$$

The local gradient for the output derivative of the activation function of Y is:

$$e_r = e_r f'(Y_{-inr}) f'(Y_{-inr}) = Y_{-inr}(1 - Y_{-inr}). \tag{5}$$

The weight correction term, used to update w_{jr} later is:

$$\Delta w_{jr} = -\left[\sec(Y_r) + (1+k)^{|e|}\right] \delta_r Y Y_j. \tag{6}$$

The bias correction term, used to update w_{0r} later is:

$$\Delta w_{0r} = -\left[\sec(Y_r) + (1+k)^{|e|}\right] \delta_r, \tag{7}$$

sending δ_r to the hidden units ($YY_j,\ j = 1,\ldots,p$) in the layer above we obtain:

$$\delta_{-inj} = \sum_{r=1}^{m} \delta_r w_{jr}\,. \tag{8}$$

The local gradient for the hidden layer (YY_j):

$$\delta_j = \delta_{-inj} f'(YY_{-inj})\,. \tag{9}$$

the weight correction term, used to update v_{hj} is:

$$\Delta v_{hj} = -\left[\sec(Y_r) + (1+k)^{|e|}\right] \delta_j Y_h\,. \tag{10}$$

The bias collection term to update v_{0j} later:

$$\Delta v_{0j} = -\left[\sec(Y_r) + (1+k)^{|e|}\right] \delta_j\,, \tag{11}$$

by sending δ_j to the hidden unit ($L_h,\ h = 1,\ldots,a$) in the layer above we obtain:

$$\delta_{-inh} = \sum_{j=1}^{b} \delta_j v_{hl}\,. \tag{12}$$

The local gradient of the hidden layer L_h:

$$\delta_h = \delta_{-inh} f'(L_{-inh}),\ f'(L_{-inh}) = L_{-inh}(1 - L_{-inh})\,. \tag{13}$$

The weight correction:

$$\Delta u_{ih} = -\left[\sec(Y_r) + (1+k)^{|e|}\right] \delta_h x_i\,, \tag{14}$$

and collate the bias weight corrective term used to update u_{0h}:

$$\Delta u_{0h} = -\left[\sec(Y_r) + (1+k)^{|e|}\right] \delta_h\,. \tag{15}$$

In feedback phase all the layers are adjusted simultaneously. The weight update is as follows: For each output layer $j = 0,1,2,\ldots,p,\ \ r = 1,\ldots,m$:

$$w_{jr}(t+1) = w_{jr}(t) + \left[\sec(Y_r) + (1+k)^{|e|}\right] \delta_r LL_j\,. \tag{16}$$

For bias:

$$w_{0r}(t+1) = w_{jr}(t) + \left[\sec(Y_r) + (1+k)^{|e|}\right] \delta_r\,. \tag{17}$$

For each hidden layer $LL_j,\ \ h = 0,\ldots,q,\ \ j = 1,\ldots,p$:

$$v_{hj}(t+1) = v_{hj}(t) + \left[\sec(Y_r) + (1+k)^{|e|}\right] \delta_j Y_h\,. \tag{18}$$

For bias:

$$v_{0j}(t+1) = v_{0j}(t) + \left[\sec(Y_r) + (1+k)^{|e|}\right] \delta_j\,. \tag{19}$$

For each hidden layer $L_h,\ i = 0,\ldots,n,\ h = 1,\ldots,q$:

$$u_{ih}(t+1) = u_{ih}(t) + \left[\sec(Y_r) + (1+k)^{|e|}\right] \delta_h x_i\,. \tag{20}$$

For the biases:

$$u_{ih}(t+1) = u_{ih}(t) + \left[\sec(Y_r) + (1+k)^{|e|}\right] \delta_h\,. \tag{21}$$

3.4. Implementation DBBPLR Algorithm

The BBP algorithm was implemented with a fixed value of the training rate from 0 to 1, and DBBPLR trained with the dynamic function of the training rate. There are no theories to determine the value of the limited error or condition. Anyway, the range of the limited error affects the training time [21]. In [22] the stop training is set by l to 10^{-5}. The convergence rate is very slow. It takes 500,000 epochs. In [23] the limited error by less than $3 \cdot 10^{-4}$. The convergence rate was very slow. It took 10,000 epochs.

0 : Read the initial weights.

1 : Read the number of neurons in the hidden layer.

2 : Read the pattern XOR 2 bit, obtain the target and limit the error E to 10^{-6}.

3 : Read the dynamic rate.

4 : While MSE $>$ limited error, repeat steps 4–15.

5 : For each training pair, repeat steps 5–15.

6 : Calculate the error training using Eq. 4.

7 : Compute the error signal δ_k at neuron k from Eq. 5.

8 : Calculate the weight correction for each Δw_{jr} and bias Δw_{0r} using Eq. 6 and 7.

9 : Send δ_k to LL_j and calculate the error signal δ_{-inj} and the local gradient of the error signal δ_j using Eq. 8 and 9.

10 : Calculate the weight correction for each Δv_{hj} and the bias Δv_{0j} using Eq. 10 and 11.

11 : Send δ_j to L_h and calculate the error signal δ_{-inh} and the local gradient of the error signal δ_h, using Eq. 12 and 13.

12 : For layer L_h calculate the weight correction for each Δu_{ih} and bias Δu_{0h} using Eq. 14 and 15.

13 : Update weight for each layer:

- output layer Y_r using Eq. 16 and 17,
- hidden layer LL_j using Eq. 18 and 19,
- hidden layer L_h using Eq. 20 and 21.

14 : Calculate the error training, time training and accuracy training.

15 : Test the conditional.

4. The Results

The accuracy training is measured by the following [25]:

$$\text{Accuracy} = \frac{1 - \text{absolut}(\text{T}_\text{i} - \text{O}_\text{i})}{\text{UP} - \text{LW}} \cdot 100\ [\%]\,,$$

where UP and LW are the upper bound and lower bound of activation function. Because the sigmoid function is used, the UP = 1 and LW = 0.

4.1. DBBPLR Algorithm Using XOR Problem

We run 10 experiments with DBBPLR algorithm given in Eq. 2 in Matlab 2012a. The experimental results are shown in Table 1.

Table 1
Average the performance of DBBPLR algorithm with XOR

Experiments	Time [s]	Epoch	Accuracy of training
Average	8.119	4426	0.9847
St. dev.	0.6614	0	$1.112 \cdot 10^{-16}$

As shown in Table 1 the back propagation algorithm enhances the performance of the training, and also reduces the training time. The average time of training is $t = 8.119$ s with the average epoch is 4426. The dynamic training rate has highest effects for increasing the accuracy of the training, whereas the average of accuracy training is 0.9807, it is close to 1. The training is shown in Fig. 1.

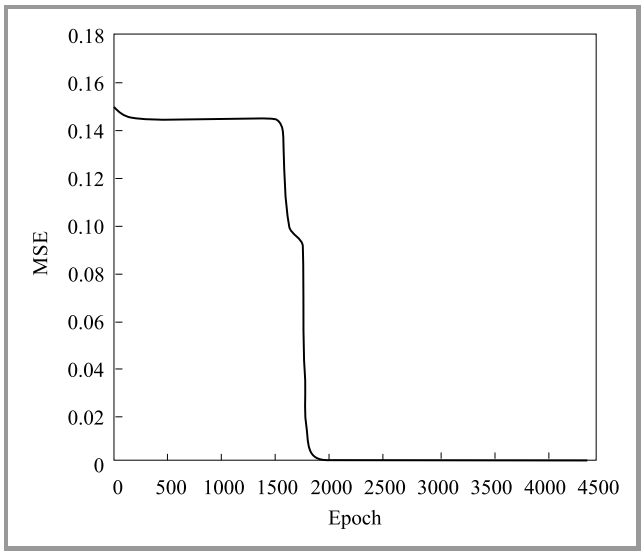

Fig. 1. Training curve for the DBBPLR algorithm.

The weight does not change before 1500 epochs, meaning that the DBBPLR algorithm is saturated, after which the training curve converges quickly to obtain the minimum error.

4.2. BBP Algorithm Using XOR

The simulation result of the BBP algorithm, given in Eq. 1 with trial or manual values for each training rate is tabulated in Table 2. The best performance of the BBP algorithm is achieved at $\gamma = 0.5$ when the training time is 35.7590 s. The worst performance of the BBP algorithm is achieved at $\gamma = 0.034$.

Meanwhile for Fig. 2 one can see that the BBP algorithm has the highest saturation training because the weight training slightly changes until 6,000 epoch and then starts to change.

Table 2
Average the performance of BBP algorithm with XOR

γ	Time [s]	Epoch
0.1	193.7690	86954
0.2	215.4940	43310
0.3	103.1070	28988
0.4	91.4450	21894
0.5	35.7590	17665
0.6	57.7380	14858
0.07	330.2740	124845
0.08	311.1590	109017
0.09	212.3080	96745
0.034	646.9530	260273
Average	219.8006	80455
St. dev.	171.4614783	71506.58607

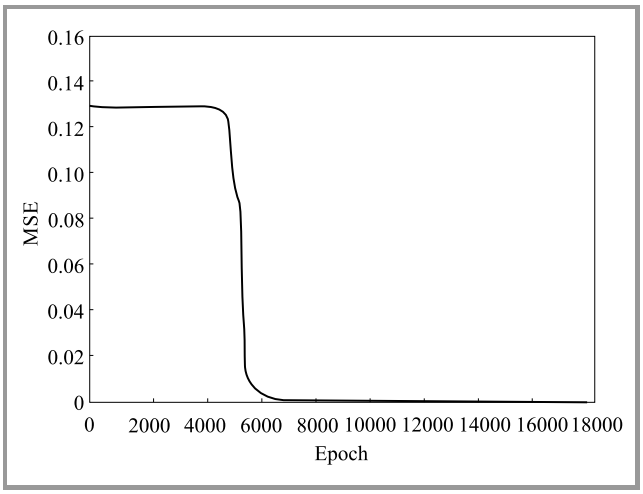

Fig. 2. Training curve of the BBP algorithm.

4.3. DBBPLR Algorithm with Balance Training Set

The balance data set is one of the best-known databases in pattern recognition. The data set has 625 patterns, which are used for training, and 375 patterns used for testing. Ten experiments were performed. The experimental results are given in Table 3.

The average training time is 11.284 s with 144 epochs. The average of the training accuracy is 0.999, it is close to one. This high accuracy indicates that the dynamic training rate helps the BBP algorithm to remove saturation training, to obtain faster training and to reach the global minimum training of the balance training set.

Table 3
The training performance of DBBPLR algorithm with balance

Experiments	Time [s]	Epoch	Accuracy of training
Average	11.284	144	0.999
St. dev.	0.925871	0	0

The DBBPLR algorithm has a flat spot until 600 epochs. After 600 epochs, the training curve convergence reduces the error training quickly. This observation means that the formulae created help the DBBPLR algorithm to reach the global minimum after 50 epochs.

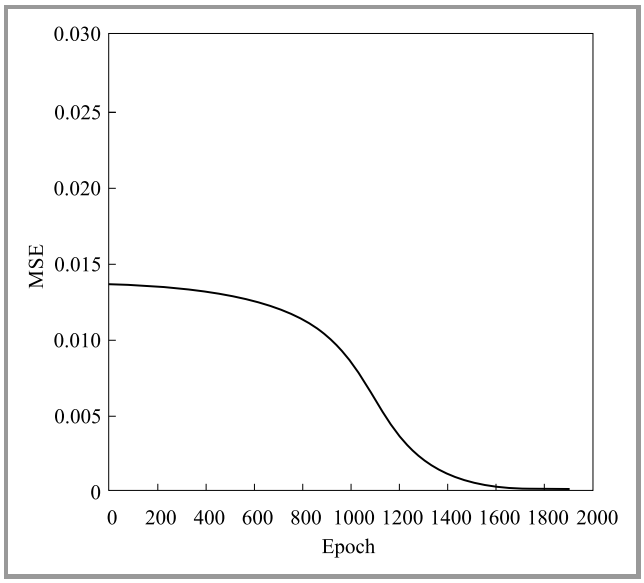

Fig. 3. Training curve of DBBPLR algorithm with the balance testing set.

4.4. Average the Performance of BBP Algorithm with Balance Training

The performance was tested using 375 patterns as a form of training. The experiments result is given in Table 4. The average training time is in the interval $30.9480 \leq t \leq 431.8660$ s, with 30.9480 s as the minimum training time and 431.8660 s as the maximum training time. The best performance of the BBP algorithm is achieved at $\gamma = 0.08$, whereas the average training time is 30.9480 s. The worst performance of the BBP algorithm is achieved at $\gamma = 0.01$. The BBP algorithm suffers the highest saturation when γ is 0.01.

4.5. DBBPLR Algorithm With Balance Testing Set

Table 5 shows results of testing the DBBPRL algorithm using the balance data testing set. The dynamic approach for training rate reduces the time required for training and

Table 4
Average the performance of BBP algorithm with XOR

γ	Time [s]	Epoch
0.01	431.8660	1907
0.02	105.5980	964
0.33	68.8110	655
0.045	55.5870	453
0.05	44.2000	414
0.068	43.7580	324
0.07	45.6230	317
0.08	30.9480	288
0.09	34.2550	258
Average	95.627	620
St. dev.	120.767	501.87

enhances the convergence of the time training. The average training time is 13.879 s at an average epoch of 273.

Table 5
Average the performance of DBBPLR algorithm with balance data testing set

Experiments	Time [s]	Epoch	Accuracy of training
Average	13.879	273	0.9963
St. dev.	0.570648	0	0

The training curve at approximately 10–50 epochs is a straight line with a flat spot, which means that the weight does not change for each epoch. In addition, the curve training begins to fall quickly.

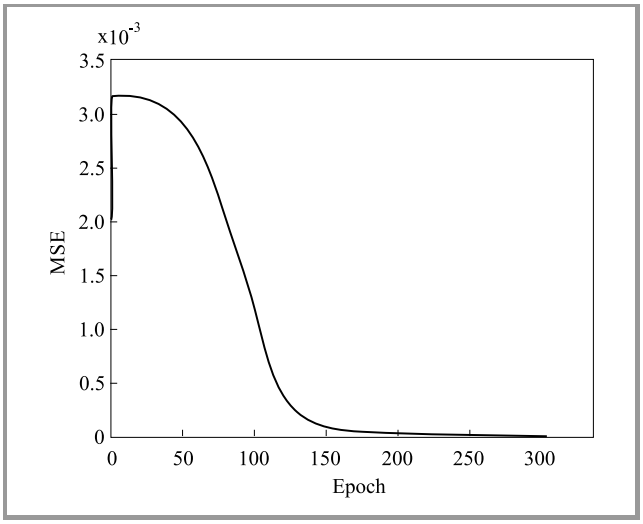

Fig. 4. Curve training of DBBPLR algorithm with the balance training set.

4.6. BBP Algorithm with Balance Testing Set

The BBP algorithm was tested using 250 patterns. The results are given in Table 6.

Table 6
The performance of training of BBP algorithm with balance testing set

γ	Time [s]	Epoch
0.1	448.9520	4084
0.2	213.3720	2080
0.3	181.1260	1418
0.4	68.5580	1091
0.5	54.6830	896
0.6	45.2700	768
0.7	40.7950	676
0.8	38.8610	609
0.9	71.4690	556
1	61.7610	514
Average	122.4847	1269
St. dev.	123.1680033	1043.628459

The best performance of the BBP algorithm was at $\gamma = 0.8$, where the BBP algorithm gives fast training at the same point. The range of the average training time is $38.8610 \leq t \leq 448.9520$ s. 38.8610 is a minimum training time; 448.9520 is the maximum.

4.7. DBBPLR Algorithm with Iris Training Set

The DBBPRL algorithm, given in Eq. 2, was run 10 times in Matlab 2012a. The experimental results are given in Table 7.

Table 7
Average the performance of DBBPLR algorithm with iris training set

Experiments	Time [s]	Epoch	Accuracy of Training
Average	1.637	121	0.99442
St. dev.	0.832629	60.93242	0.0004069

The average training time is $1.637 \approx 2$ s, with 121 epochs. The average of the accuracy training is 0.99455 the value of accuracy is close to 1. This high accuracy indicates that the dynamic training rate helps the back-propagation algorithm to remove saturation training, to obtain faster training and to reach the global minimum training. The standard deviation for time training is high value.

4.8. BBP Algorithm with Iris Training Set

The algorithm was tested with trial or manual values for each training rate. The results are shown in Table 8. The best performance of the BP algorithm is achieved at $\gamma = 0.5$ where the training time is 25.1360 s. The worst performance is achieved at $\gamma = 0.08$ when the training time is 191.7140 s.

Table 8
The performance of training of BBP algorithm with iris training set

γ	Time [s]	Epoch
0.1	74.2100	1458
0.2	25.1360	527
0.03	46.5020	1086
0.4	37.0980	861
0.06	75.9100	1890
0.7	29.0920	708
0.07	82.1590	2057
0.08	191.7140	4826
0.09	66.0930	1680
0.034	134.4380	3398
Average	71.43136364	1849
St. dev.	49.2239	1267.7981

4.9. DBBPLR Algorithm with Iris Testing Set

The performance of a proposed dynamic algorithm was tested using balance data from the iris data set. The data set has 150 patterns. 90 patterns were used for training and 60 patterns for testing. The structure of the algorithm considered is 4:2:1. The results are shown in Table 9.

Table 9
Average the performance of DBBPLR algorithm with iris testing set

Experiments	Time [s]	Epoch	Accuracy
Average	1.894	226	0.993
St. dev.	0.973410	98.77049154	0.002761159

The average training time is 1.894 s with average 226 epochs. The average of accuracy is 0.993. It indicates that the dynamic training rate helps the BBP algorithm to remove saturation training, to obtain faster training and to reach the global minimum training of the balance training set.

4.10. BBP Algorithm with Iris Testing Set

The performance was tested with 60 patterns (Table 10, next page). The best performance of the BBP algorithm is

achieved at $\gamma = 0.1$ whereas the training time is 21.2480 s. The worst performance of the BBP algorithm is achieved at $\gamma = 0.5$ when the training time is 108.355 s.

Table 10
The performance of training of BBP algorithm with iris testing set

γ	Time [s]	Epoch
0.1	21.2480	674
0.2	96.8510	3068
0.03	51.7230	1534
0.04	85.2280	1487
0.5	108.355	2069
0.06	59.7260	1826
0.07	39.7960	1157
0.08	61.0250	1776
0.09	77.9630	2217
0.034	48.6180	1499
Average	65.0533	1731
St. dev.	25.5128	612.2052

5. Discussion on Performance of the DBBPLR Algorithm

To verify or to validate the efficiency of the proposed algorithm, the performance of the improved DBBPLR algorithm was compared to the BBP algorithm based on certain criteria such as MSE, average time of training, and the number of epochs. The performance of the dynamic algorithm has been compared to the BBP algorithm in [26], [27]. The speed up training is calculated using the formula [28]–[30]:

$$\text{Speedup} = \frac{\text{Execution time of BBP}}{\text{Execution time of DBBPLR}}.$$

The number of an epoch is considered as a criterion used to compare the performance of the training. The comparison between the DBBPLR algorithm and BBP algorithm is presented in Table 11.

The dynamic algorithm provides superior performance over the BBP algorithm for all data sets. The range of the training time of the DBBPLR algorithm is $1.63686 \leq t \leq 13.8791$ s. This is a narrow interval, meaning that the DBBPLR algorithm reaches the global minimum in a short time and with few epochs. The range of training times of the BBP algorithm is $65.0533 \leq t \leq 122.4847$ s. This is a wide interval, meaning that the BBP algorithm has a long training time and a high level of training saturation. The DBBPLR algorithm is ≈ 44 times faster than the BBP algorithm at its maximum, and also the DBBPLR algorithm is ≈ 8 times faster than the BBP algorithm at its minimum.

Table 11
Speed up the DBBPLR algorithm versus BBP with various data set

Data set	DBBPLR		BBP		
	Average time [s]	Average epoch	Average time [s]	Average epoch	Speed up rate
XOR	8.119	4426	219.801	80455	27.074
Balance training	11.284	144	95.627	620	8.474
Balance testing	13.879	273	122.485	1269	8.825
Iris training	1.637	121	71.431	1849	43.639
Iris testing	1.894	226	65.053	1731	34.351

6. Conclusions

The DBBPLR algorithm gives superior training than BBP algorithm for all data set, whereas, the DBBPLR is 44 times faster than the BBP algorithm as maximum, and also the DBBPLR algorithm is 8 times faster than BBP algorithm. The dynamic training rate affected the weight for each hidden layer and output layer and eliminated the saturation training in the BBP algorithm. The dynamic DBBPLR algorithm provides superior performance of training, with higher accuracy compared to the BBP algorithm.

References

[1] H. Shao, J. Wang, L. Liu, L. D. Xu, and W. Bao, "Relaxed conditions for convergence of batch BPAP for feed forward neural networks", *Neurocomputing*, vol. 153, pp. 174–179, 2015 (doi: 10.1016/j.neucom.2014.11.039).

[2] H. H. Örkcü and H. Bal, "Comparing performances of back propagation and genetic algorithms in the data classification", *Expert Sys. with Applic.*, vol. 38, no. 4, pp. 3703–3709, 2011 (doi: 10.1016/j.eswa.2010.09.028).

[3] P. Moallem and S. A. Ayoughi, "Improving back-propagation VIA an efficient combination of a saturation suppression method", *Neural Network World*, vol. 2, pp. 207–223, 2010.

[4] S. M. Shamsuddin and F. I. Saman, "Three Term backpropagation algorithm for classification backpropagation algorithm (BP)", *Neural Network World*, vol. 4, no. 10, pp. 363–376, 2007 (doi: 10.1109/NABIC.2009.5393407).

[5] J. Ge, J. Sha, and Y. Fang, "A new back propagation algorithm with chaotic learning rate", in *Proc. 2010 IEEE Int. Conf. on Softw. Engin. and Serv. Sci.*, Beijing, China, 2010, pp. 404–407 (doi: 10.1109/ICSESS.2010.5552353).

[6] D. Xu, H. Shao, and H. Zhang, "A new adaptive momentum algorithm for split-complex recurrent neural networks", *Neurocomputing*, vol. 93, pp. 133–136, 2012 (doi: 10.1016/j.neucom2012.03.013).

[7] G. Yang and F. J. Qian, "A fast and efficient two-phase sequential learning algorithm", *Appl. Soft Comp.*, vol. 25, no. C, pp. 129–138, 2014 (doi: 10.1016/j.asoc.2014.09.012).

[8] S. Nandy, W. Bengal, and P. P. Sarkar, "An improved Gauss-Newtons method based back-propagation algorithm for fast convergence", *Int. J. of Comp. Appl. in Technol.*, vol. 39, no. 8, pp. 1–7, 2012.

[9] M. S. Al_Duais and F. S. Mohamad, "A review on enhancements to speed up training of the batch back propagation algorithm", *Ind. J. of Sci. and Technol.*, vol. 9, no. 46, pp. 1–10, 2016 (doi: 10.17485/ijst/2016/v9i46/91755).

[10] E. Noersasongko, F. T. Julfia, A. Syukur, R. A. Pramunendar, and C. Supriyanto, "A tourism arrival forecasting using genetic algorithm based neural network", *Ind. J. of Sci. and Technol.*, vol. 9, no. 4, pp. 1–55, 2016 (doi: 10.17485/ijst/2016/v9i4/78722).

[11] L. Wang, Y. Zeng, and T. Chen, "Back propagation neural network with adaptive differential evolution algorithm for time series forecasting", *Expert Sys. with Applic.*, vol. 42, no. 2, pp. 855–863, 2015 (doi: 10.1016/j.eswa.2014.08.018).

[12] L. Rui, Y. Xiong, K. Xiao, and X. Qiu, "BP neural network-based web service selection algorithm in the smart distribution grid", in *Proc. 16th Asia-Pacific in Netw. Oper. and Manag. Symp. APNOMS*, Hsinchu, Taiwan, 2014 (doi: 101109/APNOMS.2014.6996111).

[13] Q. Abbas, F. Ahmad, and M. Imran, "Variable learning rate based modification in backpropagation algorithm (MBPA) of artificial neural network for data classification", *Sci. Int.*, vol. 28, no. 3, pp. 2369–2378, 2016.

[14] H. Zhang, W. Wu, and M. Yao, "Boundedness and convergence of bach back-propagation algorithm with penalty for feed-forward neural networks", *Neurocomputing*, vol. 89, pp. 141–146, 2012 (doi: 10.1016/j.neucom.2012.02.029).

[15] H. Shao, J. Wan , L. Liu, D. Xu, and W. Bao, "Relaxed conditions for convergence of batch BPAP for feed forward neural networks", *Neurocomputing*, vol. 153, pp. 174–179, 2015 (doi: 10.1016/j.neucom.2014.11.039).

[16] Q. Feng and G. Daqi, "Dynamic learning algorithm of multi-layer perceptrons for letter recognition", *The Int. Joint Conf. in Neural Networks IJCNN*, pp. 1-6, Dallas, TX, USA, 2013.

[17] UCI Machine Learning Repository [Online]. Available: https://archive.ics.uci.edu/ml/index.html (accesed Oct. 11 2017).

[18] N. A. Hamid, N. M. Nawi, R. Ghazali, M. Najib, and M. Salleh, "Improvements of back propagation algorithm performance by adaptively changing gain, momentum, and learning rate", *Int. J. of New Comp. Archit. and their Applic. IJNCAA*, vol. 1, no. 2, pp. 889–90, 2011.

[19] G. Yang and F. J. Qian, "A fast and efficient two-phase sequential learning algorithm", *Appl. Soft Comput.*, vol. 25, pp. 129–138, 2013 (doi: 10.1016/j.asoc.2014.09.012).

[20] Y. Huang, "Advances in artificial neural networks – methodological development and application", *Algorithms*, vol. 2, no. 3, pp. 973–1007, 2009 (doi: 10.3390/algor2030973).

[21] A. E, Kostopoulos and T. N. Grapsa, "Self-scaled conjugate gradient training algorithms", *Neurocomputing*, vol. 72, no. 13–15, pp. 3000–3019, 2009 (doi: 10.1016/j.neucom.2009.04.006).

[22] Y. Li, Y. Fu, H. Liand, and S.-W. Zhang, "The improved training algorithm of back propagation neural network with self-adaptive learning rate", in *Proc. Int. Conf. on Comput. Intell. and Natur. Comput.*, no. 3, Wuhan, Hubei, China, 2009, pp. 1–4 (doi: 10.1109/CINC.2009.111).

[23] C.-C. Cheung, S.-C. Ng, A. K. Lui, and S. S. Xu., "Enhanced two-phase method in fast learning algorithms", in *Proc. Int. Joint Conf. on Neural Networks IJCNN*, Barcelona, Spain, 2010, pp. 1–7 (doi: 10.1109/IJCNN.2010.5596519).

[24] C. Yang and R. Xu, "Adaptation Learning rate algorithm of feed-forward neural networks", in *Proc. Int. Conf. on Inf. Engin. and Comp. Sci.*, Wuhan, Hubei, China, 2009, pp. 1–3 (doi: 10.1109/ICIECS.2009.5366919).

[25] N. M. Nawi, N. A. Hamid, R. S. Ransing, R. Ghazali, and M. N. M. Salleh, "Propagation neural network algorithm with adaptive gain on classification problems", *Int. J. of Datab. Theory and Applic.* vol. 4, no. 2, pp. 65–75, 2011.

[26] K. Wang, L. Zhuo, H. Lu, H. Guo, L. Xu, and Y. Zhang, "An improved BP algorithm over out-of-order streams for big data", in *Proc. Int. ICST Conf. on Commun. and Network.*, Guilin, Kuangsi, China, 2013, pp. 840–845 (doi: 10.1109/ChinaCom.2013.6694712).

[27] Q. Dai, Z. Ma, and Q. Xie, "A two-phased and ensemble scheme integrated ack propagation algorithm", *Appl. Soft Comput.*, no. 24, pp. 1124–1135, 2014 (doi: 10.1016/j.asoc.2014.08.012).

[28] S. Scanzio, S. Cumani, R. Gemello, R. F. Mana, and F. Laface "Parallel implementation of neural network training for speech recognition", *Patt. Recog. Lett.*, vol. 1, no. 11, pp. 1302–1309, 2010 (doi: 10.1016/j.patrec.2010.02.003).

[29] M. N. Nasr and M. Chtourou, "A self-organizing map-based initialization for hybrid training of feed-forward neutral networks", *Appl. Soft Comput.*, vol. 11, no. 8, pp. 4458–4464, 2011 (doi: 10.1016/j.asoc.2011.05.017).

[30] F. Saki, A. Tahmaasbi, H. Soltanian-Zadeh, and S. B. Shokouhi, "Fast opposite weight learning rules with application in breast cancer diagnosis", *Comp. in Biol. and Med.*, vol. 43, no. 1, pp. 32–41, 2013 (doi: 10.1016/j.compbiomed.2012.10.006).

Fatma Susilawati Mohamad is an Associate Professor in the Department of Information Technology, Faculty of Informatics and Computing, UniSZA. She has 18 years of experience in the academic field and has been actively involved in teaching, research, publications, professional services, consulting, and administration. Now she serves as a Deputy Dean at the Graduate School, UniSZA. She has graduated a B.Sc. (Information Systems) from Oklahoma City University, U.S.A. in 1997. She has continued his studies at Universiti Kebangsaan Malaysia and obtained her M.Sc. in Information Technology (Computer Science) in 2004. She received his Ph.D. in Computer Science from Universiti Teknologi Malaysia in 2012. Her specialization is in computer science focusing on pattern recognition and image processing.
E-mail: fatma@unisza.edu.my
Department of Information Technology
Faculty of Informatics and Computing
Universiti Sultan Zainal Abidin (UniSZA)
21300 Gong Badak, Kuala Nerus, Terengganu

Mohammed Sarhan Al_Duais is Ph.D. student in the Department of Information Technology, Faculty of Informatics and Computing. His field of research activity is artificial intelligence more specially, improvement the training of back propagation algorithm. He is an author in several scientific papers and publications in the field of arterial intelligence. He has completed the B.Sc. degree and master from Sana'a University.
E-mail: duais2025w@gmail.com
Department of Information Technology
Faculty of Informatics and Computing
Universiti Sultan Zainal Abidin (UniSZA)
21300 Gong Badak, Kuala Nerus, Terengganu

Paper

Parallel Mutant Execution Techniques in Mutation Testing Process for Simulink Models

Le Thi My Hanh, Nguyen Thanh Binh, and Khuat Thanh Tung

The University of Danang – University of Science and Technology, Vietnam

https://doi.org/10.26636/jtit.2017.113617

Abstract—Mutation testing – a fault-based technique for software testing – is a computationally expensive approach. One of the powerful methods to improve the performance of mutation without reducing effectiveness is to employ parallel processing, where mutants and tests are executed in parallel. This approach reduces the total time needed to accomplish the mutation analysis. This paper proposes three strategies for parallel execution of mutants on multicore machines using the Parallel Computing Toolbox (PCT) with the Matlab Distributed Computing Server. It aims to demonstrate that the computationally intensive software testing schemes, such as mutation, can be facilitated by using parallel processing. The experiments were carried out on eight different Simulink models. The results represented the efficiency of the proposed approaches in terms of execution time during the testing process.

Keywords—mutant execution, mutation testing, parallel processing, software testing.

1. Introduction

Software testing is an expensive process. It typically consumes more than half of the development budget [1], but it is an effective way to estimate the reliability of software. With the increasing expectations for software quality, developers are required to perform more effective testing on large and complex systems.

In this context, mutation testing has been used as a fault injection technique to measure the adequacy of test cases. This method adopts a "fault simulation mode". It has been advocated as a technique for generating test cases by inserting faults into an original program, and the effectiveness of a test suite is represented by its "mutation score". Thus, mutation testing is used to measure the robustness of a test suite. Though powerful, mutation testing is computationally intensive, as numerous mutants need to be produced and executed.

When a mutation is introduced to a large application, a huge number of mutants can be generated. Despite the existing techniques to reduce the costs of mutation analysis, the computational time required to apply mutation testing to large applications is still very long. The costs of mutation testing depend mainly on the number of mutants generated, as well as on the number of test cases. In a single sequential process, the total computational time of mutation testing includes the time spent generating the mutants and the time devoted to executing the tests against all the mutants and the original system (which must be executed at least once). The execution time E_t is always much higher than the generation time G_t. This is the reason why researchers have focused their efforts on reducing E_t. For example, mutant schemata [2] make it possible to speed up execution by including all mutants in a single file. This makes it possible to avoid continuous uploading of mutant files into memory, and thus launching a new process to execute each program version. Techniques such as random selection of mutants [3], selective or constrained mutation [4]–[7], and higher-order mutation [8] produce fewer mutants, which exerts a positive influence on the total execution time. Techniques such as byte-code translation [9], which remove the compilation-related tasks, may in turn reduce the time of processing.

In addition to the described techniques, parallel execution attempts to decrease the overall time by distributing the execution across different processors. This is to improve the performance of mutation testing without compromising the effectiveness of the process in which mutants and tests are executed on parallel processors. This method contributes to reducing the total time needed to perform the mutation analysis.

The size and complexity of the system under test determine the execution expense. The larger or the more complex a given application is, the more test cases are required to achieve the adequate coverage. It takes less time, hence, to generate mutants, but the execution time grows exponentially along with the number of mutants and test cases which, in turn, depends on the size of the application.

This paper presents a study of the parallel mutant execution technique, which is appropriate to reduce the computational cost of the execution phase. Three distribution strategies are proposed to parallelize this task.

The rest of this paper is organized as follows: Section 2 describes some related work on parallel mutation testing. Section 3 briefly introduces the Simulink environment, mutation testing for Simulink models, and a process of mutant generation and execution. In Section 4, three distribution algorithms are proposed to parallelize the execution phase. The experimental results are discussed in Section 5 and,

finally, Section 6 presents the conclusions and future work required.

2. Related Work

In the surveys on mutation testing, such as mutation testing cost reduction techniques, Jia and Harman [10], Mateo and Usaola [11] identified three directions of investigation: execution of mutants (i) in single instruction, multiple data machines (SIMD), (ii) in multiple instruction, multiple data (MIMD) machines, and (iii) with optimized serial algorithms.

Mathur and Krauser [12] were the first ones who performed mutation testing on a vector processor system. They suggested that vectorizable programs be created, each one incorporating several mutants of the same type. The authors hoped that a vector processor could execute the unified mutant programs and achieve a significant speed-up over a scalar processor. The proposed strategy had not been implemented yet, and the authors implied in their papers that only scalar variable replacement (SVR) type mutants are suitable for unification. A later paper by Krauser, Mathur, and Rego [13] proposed an approach for the concurrent mutant execution under SIMD machines. The authors suggested a strategy for efficient execution of mutants. Mutants of the same type are grouped together, and the groups are handled by different processors in a SIMD system. This strategy, however, has also not been implemented yet.

The second direction of research is based on MIMD machines. The work of Choi and Mathur [14] was the first study about the parallel mutant execution on these machines. These authors presented the Pmothra tool that is an adaptation of the Mothra tool [15] for the Ncube/7 Hypercube machine. It was based on a mutant generator, a mutant compiler, a mutant scheduler, several test case servers, and mutant executors. This tool used a dynamic distribution algorithm that executes a mutant when a node of the hypercube becomes available. Another important work was also proposed by Offutt *et al*. [16]. These authors presented the HyperMothra tool, an adaptation of the Mothra tool [13], to be executed in the Intel iPSC/2 hypercube machine with 16 processors. The HyperMothra tool was designed to generate mutants for Fortran systems with 22 mutation operators and to interpret them in parallel in the hypercube.

The last direction of research in the field of parallel mutation testing concerns optimized serial algorithms. Fleyshgakker and Weiss [17], [18] proposed some algorithms to reduce the number of executions and improve the efficiency of the mutation testing process. Although the algorithms were not implemented for parallel execution, the authors indicated that their structure made them easily parallelizable.

In the related works described above, all the mutation studies used programs written in Fortran. In recent years, programming languages, networks, and processors have evolved a great deal. Therefore, recent studies concerned with parallel mutation have adapted the existing cost reduction techniques to new programming languages such as Java. Mateo and Usaola [19] introduced Bacterio-P, which is a parallel extension of the mutation testing tool Bacterio [20] using Java-RMI for communicating among the nodes of the network. In addition, the authors presented five distribution schemes adopting dynamic and static distributions. Among these ones, the parallel execution with the dynamic ranking and ordering algorithm, which is a dynamic distribution algorithm based on factoring self-schedulling ideas [21], gave the best results. However, the mechanisms used in the communications are not the most adequate for high performance environments since a high degree of latency is introduced by this technology, and Java-RMI is much slower than MPI. To cope with this problem, Pablo *et al*. [22] proposed a dynamic distributed algorithm, known as EMINENT, to reduce the execution time associated with the classical mutation testing scheme. Their approach was implemented using the standard Message-Passing Interface (MPI) library to facilitate communications in high-performance environments. In another research, Saleh and Nagi [23] proposed the Hadoop Mutator framework, which is based on the MapReduce programming model to distribute and execute the mutant generation and the testing processes. Nonetheless, this approach follows a static schema, so it is not suitable for heterogeneous and dynamic environments.

As one may see, most studies on parallel mutation testing are applied to programming languages. To the best of our knowledge, this is the first work on parallel mutation testing concerning designs in the Simulink environment.

3. Mutation Testing for Simulink Models

Simulink [24] is a block diagram environment for multi-domain simulation and model-based design. It supports simulation, automatic code generation, continuous test and verification of embedded systems. Simulink provides a graphical editor, customizable block libraries and solvers for modeling and simulating dynamic systems. It is integrated with Matlab software, enabling to incorporate Matlab algorithms into models and export simulation results to Matlab for further analysis.

Simulink has been popularly used as a high-level system prototype or a design tool in many domains, including aerospace, automobile, and electronic industries.

Simulink models are of the data-flow variety and consist of three levels of granularity: whole models, subsystems, and blocks. Models contain systems, and systems contain other subsystems and blocks. Blocks originate from predefined block libraries (covering generic functions such as addition or logical operators, but also domains like fuzzy logic or network communication). Blocks are connected by lines that provide a mechanism to transfer signals across the connections, and have their own semantics. Blocks receive a specific number of input signals from which output signals are computed. The underlying internal representation of Simulink models is stored as text either in a Simulink

MDL file or an XML file in the case of newer versions of Simulink.
Simulink plays an increasingly important role in system engineering, while verification and validation of Simulink models are becoming ever more vital to users [25].
This paper is concerned with mutation testing for Simulink models that contain basic blocks in predefined libraries, such as commonly used blocks, continuous, discrete, logic and relational operations, math operations, sinks and sources.

3.1. Mutation Testing for Simulink Models

Mutation testing, which is a fault-based testing technique proposed by DeMillo *et al*. [26], focuses on measuring the quality of a test set according to its ability to detect specific faults. It works in the following way: a large number of simple faults is introduced to a program (or a model), one at a time. The resulting changed versions of the program (or model) under test are called mutants. Test data are then constructed to cause these mutants to fail. The effectiveness of the test set is measured by the percentage of the number of mutants killed over the total number of mutants. Since the number of mutants that can be generated is very large (the number is usually in the order of N^2, where N is the number of variable references in the program), many methods have been suggested to reduce the computational expenses of this testing technique.
In mutation testing, faults are introduced to the program (or model) under test by using mutation operators. They are well-defined rules to make syntactic changes in the original program (or model). They are designed based on the experience of the target language usage and the most common faults. Mutation testing is usually applied to programming languages such as C++, Java, and C#. In this study, we apply mutation testing to Simulink designs, using a set of mutation operators proposed in [27] and shown in Table 1.

Table 1
Mutation operators

Operator	Description
VNO	Variable negation operator
VCO	Variable change operator
TRO	Type replacement operator
CCO	Constant change operator
CRO	Constant replacement operator
SCO	Statement change operator
SSO	Statement swap operator
DCO	Delay change operator
ROR	Relational operator replacement
AOR	Arithmetic operator replacement
ASR	Arithmetic sign replacement
LOR	Logical operator replacement
BRO	Block removal operator
SRO	Subsystem replacement operator

By applying a mutation operator to the model under test, i.e. by inserting a single fault into the model, a faulty model is obtained, which is called a mutant. Then, test data should be generated to reveal the fault introduced.
A test suite is considered good if it contains tests that are able to distinguish a large number of these mutants from the original model. If a mutant can be distinguished from the original model by at least one of the test cases in the test set, the mutant is considered to be killed. Otherwise, the mutant is alive. Sometimes the mutants cannot be killed due to the semantic equivalence of the mutants and the original model. These mutants are called equivalent mutants. Worse still, determining whether a mutant is equivalent is generally undecidable [28], and so typically the decision is left for testers to establish manually.
The proportion between mutants killed and all non-equivalent mutants is called the mutation score and is formally defined as:

$$MS(P,Ts) = \frac{K}{T-E}, \tag{1}$$

where P is the program under test, Ts is the test suite, K is the number of mutants that have been killed, T is the total number of mutants, and E is the number of equivalent mutants.
The process of mutation testing for Simulink models consists in generating mutants, executing mutants, analyzing results and in the generation of test suites. If this process is performed manually, it will require too much time. Hence, we have designed and implemented a *MuSimulink* tool to automate this process. The design details of this tool are presented in [29].

3.2. Mutant Generation and Execution

This section presents, in detail, the process of generating and executing mutants for Simulink models. As with any automated mutation testing system, there are several important steps that a tester must follow. Because there is a large number of mutants generated for each model, it has been considered impractical to compile and store each mutant model separately. Therefore, MuSimulink has been built as an interpretive system. Instead of creating, compiling, and storing many separate models, the Simulink model is translated once into an intermediate form, and each mutant is stored in the form of a short description of the changes required to create the mutants. In MuSimulink, these descriptions are stored in records of a mutant description table (MDT). The testing process continues until a satisfactory mutation score is attained or is forced to stop due to time or economic constraints. The major steps of mutation testing are listed below and illustrated in Fig. 1.

1. **Mutant Generator**. The mutation testing process begins with the construction of mutants, which are automatically created through mutation operators. First of all, the original model O is submitted, analyzed, and parsed to create an intermediate form

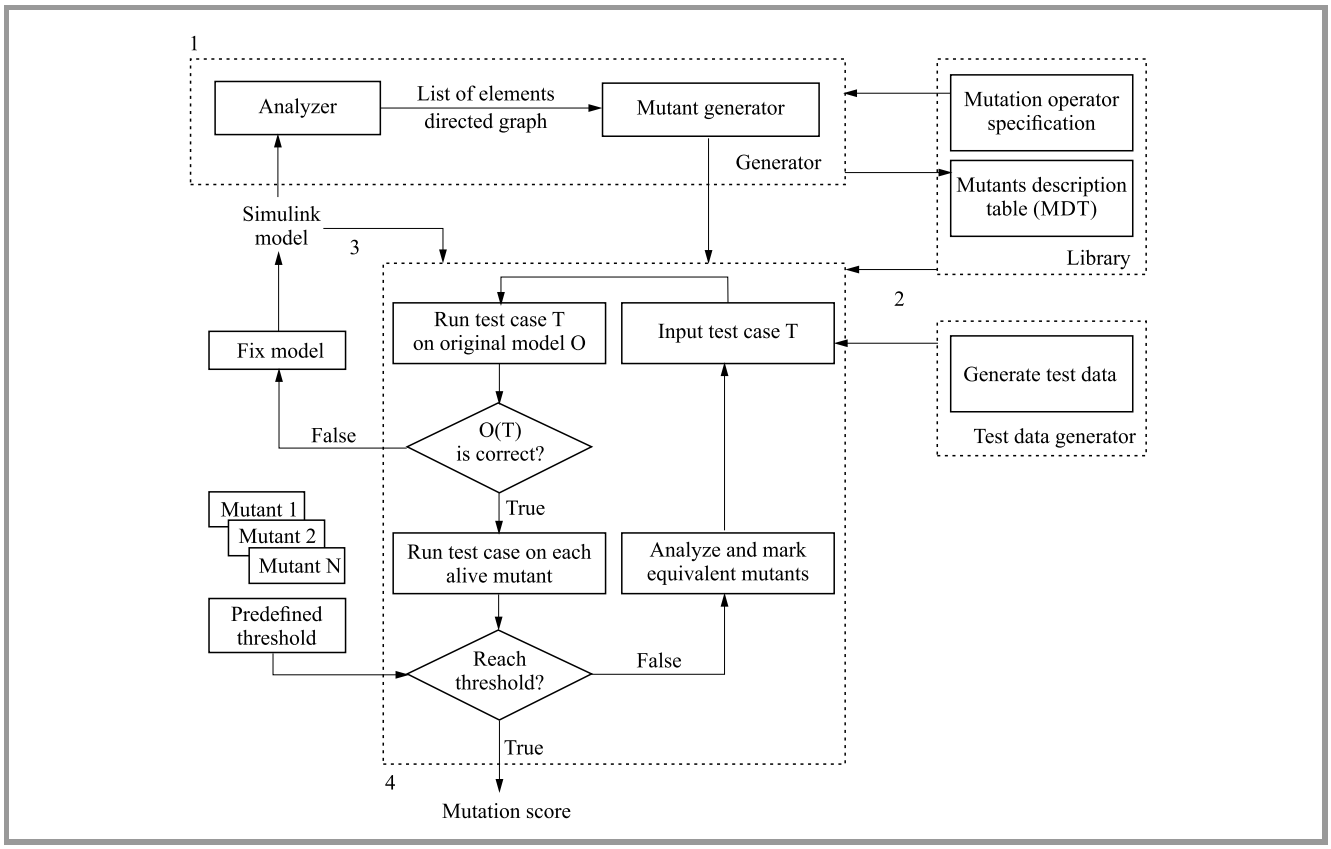

Fig. 1. MuSimulink mutation testing process.

ready for interpretation. One or more mutation operators will be selected to be applied to O. The testers typically use all mutation operators available. Based on the mutation operator specification, each of the selected operators is applied to O to produce the MDT that describes the M set of O mutants.

2. **Test data generator**. A set of test cases T is submitted. Each test case within T contains values for the input variables of O. T can be created manually by testers or generated automatically by the MuSimulink test data generator.
3. **Mutation analysis**. This task is to execute the model under test and its mutants against test cases. The goal is to determine how many mutants are killed by the tests. The results are then analyzed, and the mutation score is calculated to measure the mutation adequacy of the test suite.

The original model O is interpreted once for each test case in T by the MuSimulink tool. A set of expected outputs $O(T)$ is produced. The expected outputs of O can be examined at any point of time during the testing process to determine whether O is performed on T correctly. If any output is incorrect, a fault has been found, and the model must be fixed before the process restarts from step 1. If the output is correct, that test case is executed against each alive mutant within the set of mutants M. M is interpreted by MuSimulink using the input values for each test case in T. This produces a set of mutant outputs $M(T)$ that has at most $|M| \cdot |T|$ elements. Since mutants are not executed against new test cases after being killed, this set of mutant outputs will not be usually very large. In practice, T is usually small compared to M. Note that a typical test case will kill a large number of mutants, so the number of executions is usually much lower than $|M| \cdot |T|$. After that, each element of $M(T)$ is compared with the element of $O(T)$ generated from the same test data. If they are not the same, the mutant is killed. If, after output comparison, some mutants remain alive, either the test data in T are not adequate or the mutant is equivalent to O, and it can never be killed.

The results of testing are analyzed and, if necessary, further testing may be performed. If all mutants are killed (or reach the specified mutation score threshold), and all mutation operators have been applied to the original model O, then no further testing is necessary. If one or more non-equivalent mutants remain alive, then additional test cases should be added to T, and appropriate steps of the process should be repeated. If, as a result of testing, faults in O are uncovered, then O must be modified, and the testing process will be repeated as well. Existing test cases can usually be reused for the subsequent testing phases.

After all test cases have been executed against all mutants generated, each remaining mutant falls into one of two categories. The first one is composed of mutants that are functionally equivalent to the original model. The equiva-

lent mutant always yields the same output as the original model, so no test case can kill it. The second category consists of mutants that are killable, but the test set is insufficient to kill them. In this case, new test cases need to be added, and the process reiterates until the test set is strong enough to kill all mutants or until the specified mutation score threshold can be reached.

Steps are presented above to expose the inherent parallelism in mutation testing. The most computationally expensive parts of the mutation process are the execution of the original model, the mutant execution, the output comparison, and the test data generation. The mutant execution, which is performed once for each mutant and each test case, is considered to be an internal loop. The internal loop includes the interpretation of a mutant on a test case, the comparison of the mutant output with the output of the original model, and, if they differ, the killing of the mutant. The internal loop (shown in lines 11–23 of Algorithm 1, lines 14–29 of Algorithm 2, lines 16–31 of Algorithm 3) is by far the most computationally expensive part of mutation testing, and therefore it is the target of our parallelization efforts.

Algorithm 1. Mutant distribution strategy using the parfor paradigm

```
1: Read test data file T
2: Start N workers in Matlab
3: for each test case t in T do
4:     Execute original model O on t to produce Δ(O,t)
5:     if (abnormal termination O) then
6:         Send "error" message, close workers and finish
7:     else
8:         Record expected output Δ(O,t)
9:     end if
10:    Send t and expected output Δ(O,t) to workers
11:    for parallel each alive mutant m in M do
12:        Send the mutant information m to the worker
13:        Modify original model O to produce mutant model O'
14:        Interpret O' on t to produce Δ(O',t)
15:        if abnormal termination O' then
16:            Mark m as killed
17:        else if Δ(O',t) ≠ Δ(O,t) then
18:            Mark m as killed
19:        else
20:            Mutant m remains alive
21:        end if
22:        Update the killed mutant counter
23:    end for parallel
24: end for
25: Close all workers
26: Write execution outputs to a result file
```

Algorithm 2. The alternate-order mutant distribution strategy using Matlab's SPMD paradigm

```
1: Read test data file T
2: Start N workers in Matlab
3: for each test case t in T do
4:     Execute original model O on t to produce Δ(O,t)
5:     if (abnormal termination O) then
6:         Send "error" message, close workers and finish
7:     else
8:         Record expected output Δ(O,t)
9:     end if
10:    Send t and expected output Δ(O,t) to workers
11:    spmd in M
12:        k ← 0
13:        i ← N·k + labindex
14:        while i ≤ |M| do
15:            if mutant M(i) alive then
16:                Send infor. of mutant M(i) to worker labindex
17:                Modify original model O to produce mutant O'
18:                Interpret O' on t to produce Δ(O',t)
19:                if abnormal termination O' then
20:                    Mark M(i) as killed
21:                else if Δ(O',t) ≠ Δ(O,t) then
22:                    Mark M(i) as killed
23:                else
24:                    Mutant M(i) remains alive
25:                end if
26:            end if
27:            k ← k + 1
28:            i ← N·k + labindex
29:        end while
30:        Update the killed mutant counter
31:    end spmd
32: end for
33: Close all workers
34: Write execution outputs to a result file
```

4. Solutions Parallel to Mutant Execution

In a survey of parallel Matlab technologies [30], nearly 27 technologies were discovered. Many of them are defunct, while many others are currently under development, with a large user base and an active developer base. This paper uses the parallel computing toolbox (PCT) with the Matlab distributed computing server (MDCS), which is a novel technology.

While the core Matlab software itself supports multithreading, the PCT offers operations to run the Matlab code on multicore systems and clusters. The PCT provides functions for the parallel for-loop execution, creation/manipulation of distributed arrays, as well as message passing functions for implementing fine-grained parallel algorithms.

The MDCS enables to scale parallel algorithms to larger cluster sizes. The MDCS consists of the Matlab worker processes that run on the cluster and is responsible for parallel code execution and process control. Figure 2 illustrates the PCT and MDCS architecture.

The PCT also allows users to run up to 12 Matlab labs or workers on a single machine. This enables interactive development and debugging of parallel codes from a desktop. After parallel codes have been developed, they can be scaled up too much larger number of workers or labs in conjunction with the MDCS. Thus, the PCT addresses

Algorithm 3. The random-order mutant distribution strategy using Matlab's SPMD paradigm

```
1: Read test data file T
2: Start N workers in Matlab
3: for each test case t in T do
4:     Execute original model O on t to produce Δ(O,t)
5:     if (abnormal termination O) then
6:         Send "error" message, close workers and finish
7:     else
8:         Record expected output Δ(O,t)
9:     end if
10:    Send test case t and expected output Δ(O,t) to workers
11:    Get a list of alive mutants 𝓛 from M
12:    Randomly reorder list 𝓛
13:    spmd in 𝓛
14:        k ← 0
15:        i ← N·k + labindex
16:        while i ≤ |𝓛| do
17:            if mutant 𝓛(i) alive then
18:                Send infor. of mutant 𝓛(i) to worker labindex
19:                Modify original model O to produce mutant O'
20:                Interpret O' on t to produce Δ(O',t)
21:                if abnormal termination O' then
22:                    Mark 𝓛(i) as killed
23:                else if Δ(O',t) ≠ Δ(O,t) then
24:                    Mark 𝓛(i) as killed
25:                else
26:                    Mutant 𝓛(i) remains alive
27:                end if
28:            end if
29:            k ← k + 1
30:            i ← N·k + labindex
31:        end while
32:        Update killed mutants on M
33:    end spmd
34: end for
35: Close all workers
36: Write execution outputs to a result file
```

the challenge of getting codes to work well in a multicore system by enabling to select the programming paradigm that is most suitable for applications. The paper employs two most basic parts of these paradigms: parallel for-loops and Single Program Multiple Data (SPMD) blocks.

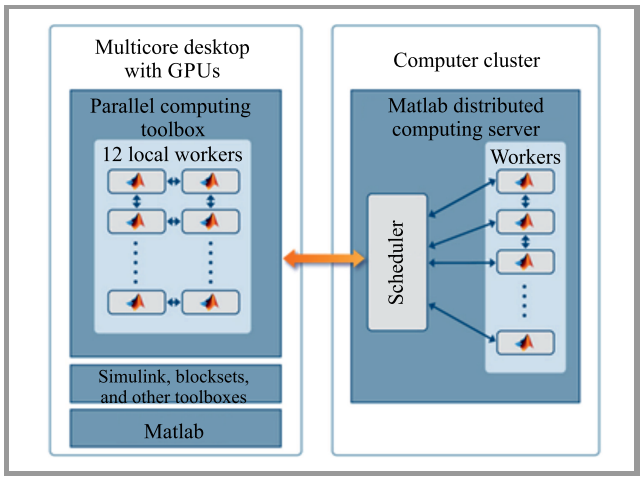

Fig. 2. The parallel computing toolbox and the Matlab distributed computing server.

4.1. Work Distribution Strategies

Parallel algorithms can be divided into two categories: task-parallel and data-parallel. Task-parallel algorithms take advantage of the fact that multiple processors can work on the same problem without communicating with each other. These algorithms can be used when the computations in a large loop are independent from each other and can be performed in any order without affecting the results. In such cases, multiple processors can analyze the subsets of the data simultaneously, without the need for inter-processor communication. Data-parallel algorithms typically involve some inter-processor communication. In such algorithms, the data are typically too large to be analyzed on a single processor. Therefore, parallel computing paradigms are used to distribute the data across processors, and each processor works on a smaller chunk of the same data. In such cases, there may be some communications required between different processors that involve the exchange of data to address boundary conditions. Based on how the data are distributed, each processor needs a small amount of data from its neighbor to complete the computations.

Since the size of T (the set of test cases) is usually small compared to the size of M (the set of mutants), our work uses the task-parallel approach. Parallelizing the mutant execution on multi-processor machines has been implemented by supplying each worker/lab with all test cases and a subset of mutants. The mutation testing using the parallel mechanism for mutants is a natural way to divide up the work because it does not necessarily guarantee an even distribution of work among workers/labs. Some mutants which are easily killed by a test case in the early stage of the mutant execution process will not be executed against most of the remaining test cases in the test set. On the other hand, equivalent mutants must be executed against all test cases, since they are not killed by any test case. Therefore, there is a wide volatility in the amount of execution time required for individual mutants. To achieve maximal speed-up, we should distribute mutants to workers/labs such that each worker/lab performs the same amount of execution. It is unfortunate that we have no way to know in advance how much execution time will be required for a mutant, or how many test cases need to be run against it. The optimum distribution of mutants, thus, cannot be determined.

The parallel mutation algorithm has two execution phases: original model execution and mutant execution. In the original model execution phase, workers/labs are not used, and only the host processor (Matlab client) computes and saves the expected outputs from the original model. In the mutant mode, the client begins with sending the startup information to the workers. For each test case, the input values and the expected outputs are sent to the workers, then once a worker interprets all its mutants on the test case, a counter of remaining alive mutants is sent back. If all the mutants within all the workers are killed (or mutation score reaches

a predetermined threshold value), then the algorithm takes an early exit and does not send more test cases to the workers. Otherwise, the next test case is sent. When all the test cases have been processed, the workers send the updated MDTs back to the client. There is recurring communication between the client and the workers. The client sends the test case information which includes the outputs of the original model and a list of elements referenced in the original model, and the workers send back the number of alive mutants.

Three dynamic distribution strategies are described below that attempt to balance the amount of work done by the workers. The distribution strategies are shown in Algorithms 1–3, where test data file T and mutant description table M are the inputs, while the output is the *Alive/Killed* list of mutants. In all algorithms, test data are delivered to workers sequentially by an outer loop (line 3 for all), while the inner iterations (line 11 in Algorithm 1, line 14 in Algorithm 2, and line 16 in Algorithm 3) take responsibility for executing mutants in parallel.

Algorithm 1 presents the first mutant distribution strategy using the Matlab parfor paradigm which divides the MDT list between the workers. Alive mutants are interpreted on each test case by the workers, and the process is referred to as internal loop iteration (lines 12–22 in Algorithm 1). The parfor statement is very simple to use due to the fact that it is based on the automatic data management, but the parfor iterations are executed in an unknown order. Thus, the effective distribution of mutants cannot be determined.

Algorithms 2 and 3 present two dynamic distribution strategies using Matlab's SPMD paradigm. For each test case, the first worker gets the first mutant in the MDT list, the second worker gets the next mutant, and so on. Dead mutants in the MDT list are not distributed to the workers. If there are N workers, the first N mutants will be assigned to the workers at a time. Then, the second N mutants will be assigned to the workers in the next iteration, and so on. Thus, if *labindex* is the index number of each worker, worker *labindex* gets mutants at position $N \cdot k + labindex$ in the MDT list (where k is from 0 to $|M|/N$ in turn, $|M|$ is the number of mutants). These two strategies distribute approximately the same number of mutants to each worker.

The second strategy, in Algorithm 2, is the distribution of MDTs in an alternate order of a list of all mutants. However, not all executions will take the same time because each mutant usually takes a different amount of time to run, so presumably, some processors will finish before others, and the total process time will be the time taken by the slowest processor. To avoid a large number of "hard to kill" mutants running on a worker, the third distribution strategy assigns each worker approximately the same number of mutants in a random order of the list of alive mutants. For each new test case, the MDT list will first be randomly reordered before alive mutants are delivered to the workers, and the parallel process is performed on this randomly reordered list (lines 11–12 of Algorithm 3).

5. Experimentation

5.1. Parallel Mutation Testing on a Single Multicore Machine

Three proposed distribution strategies were implemented in the MuSimulink tool [29], and experiments in this subsection were carried out on a single computer which uses the Intel Xeon E5520 2.27 GHz CPU with 8 GB RAM, and runs the Windows Server 2008 operating system. This computer has two processors with four cores each. Thus, to use all the cores, eight workers were run. For the distribution algorithms, a configuration parameter which needs to be established is the number of workers. In experiments, the MDT is generated for the models using mutation operators introduced in our previous work [27]. Each model under test was given 100 test cases generated randomly using MuSimulink's automatic test data generator. The original model is executed on the client, while the mutant ones are executed in parallel on eight workers using the different work distribution strategies (using parfor paradigm, alternate-order using SPMD, and random-order using SPMD).

The time of the execution phase within the mutation process is shown in Table 2, which describes the average time of ten runs per second for each model with regard to each distribution algorithm. The use of parallel strategies helps us reduce the execution time by up to 89.23% (from 4752.54 s down to 511.9 s).

Figure 3 is a chart that shows the speed-up achieved by MuSimulink using eight workers. Speedup for N workers is defined as the division between the serial execution time on one worker and the parallel execution time on N workers. *Speed-up* indicates by how much the execution time has been reduced. It may be seen from both Fig. 2 and Table 2 that the work distribution strategy 1 (using the

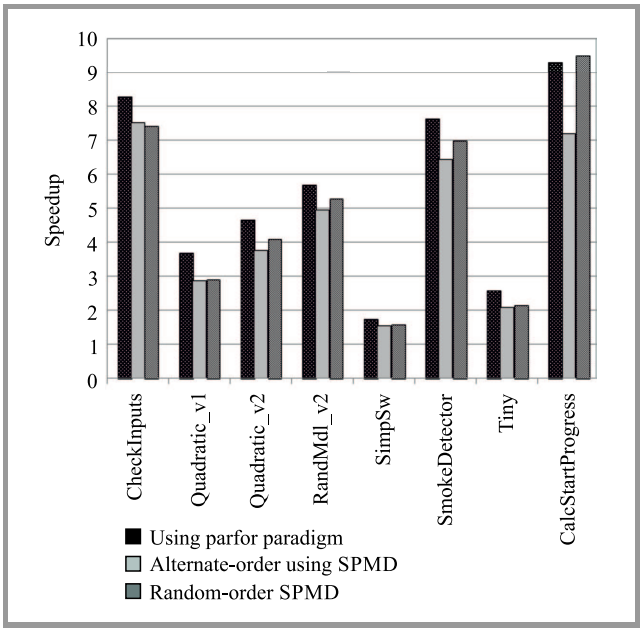

Fig. 3. *Speed-up* achieved using eight workers.

Table 2
The total execution time for each model with each distribution strategy

Model name	Mutant	Killed	MS	Time [s]			
				Serial	Using parfor paradigm	Alternate-order using SPMD	Random-order using SPMD
CheckInputs	154	130	83.77	2303.20	278.5	306.6	311.04
Quadratic_v1	161	129	90.43	636.94	173.2	221.37	220.18
Quadratic_v2	140	89	63.57	976.52	210.7	259.62	240.32
RandMdl_v2	188	138	73.40	1141.18	201.2	230.07	217.22
SimpSw	92	85	92.39	263.16	153.2	170.69	167.56
SmokeDetector	321	160	49.84	2685.83	353.3	418.08	385.11
Tiny	144	120	83.33	490.74	190.3	234.63	229.13
CalcStartProgress	458	183	39.96	4752.54	511.9	661.69	501.13

parfor paradigm) results in a much better speed-up than strategy 2 (alternate-order using SPMD), and strategy 3 (random-order using SPMD) is also marginally better than strategy 2 in general.

In the second distribution strategy, each worker receives the mutants located once at the fixed position in the list of mutants, before executing. Therefore, it is more likely a certain worker will always be available if all its mutants are killed soon. This situation causes an imbalance among workers in which some cores are overworked with hard-to-kill mutants, and so the second strategy has rendered the worst performance in comparison with the others. This drawback is partially overcome in the third strategy due to the random redistribution of alive mutants before delivering them to workers to execute with new test case. Hence, the execution time of the third strategy is shorter than that of the second one.

The second and third strategies can reduce the communication between workers and host, and the network traffic will be minimal because each worker receives fixed groups once, before starting the process of executing mutants for each test case. However, mutant model interpretation and execution times are much longer than the communication time. Meanwhile, the second and third algorithms use fixed groups of mutants, so not all executions will take the same amount of time. In such a case, some workers will finish before others and become available when they get many easy-to-kill mutants. This is the disadvantage of strategies 2 and 3 using the SPDM mechanism, while strategy 1 using parfor does not face this problem. In the first distribution strategy, the executions are split into smaller pieces, called tasks. The tasks are delivered to parallel workers several times until all the groups are delivered. In other words, the tasks are sent to the parallel workers on demand. The parfor mechanism has a smart scheduler that sends tasks to the free workers. When a worker finishes its executions, it receives a new task with its size depending on the number of remaining executions in the mutation process [31]. Hence, when the remaining unexecuted mutants are few, the task size is small. The loop of sending and executing mutants on the test case finishes if all tasks have been sent and run. The smart scheduler contributes to the reduction of the free time of a worker, as well as to the increase in the speed of mutant execution, so execution time of the strategy 1 is significantly reduced, in general terms, compared to that of the second and third strategies. Nonetheless, there are some rare exceptions such as the *CalcStartProgress* model, when the third strategy is better than the first one. As may be seen from the result of this model, the mutation score is low – it means that a lot of alive mutants exist, and these mutants are executed on all test cases in the test set. In this case, it is more likely that the number of hard-to-kill mutants on each worker is quite equal in both strategy 1 and strategy 3. Meanwhile, the first distribution strategy has to spend more time on communication, so the execution time in the strategy 1 is slightly longer.

In practice, a complex Simulink model with few mutants might run longer compared to a simple one with many mutants. This is shown in the *Serial* column of Table 2, where *CheckInputs* model with only 154 mutants shows the serial execution time being more than twice as long as that of *RandMdl v2* model with 188 mutants. The same applies also to the parallel mutant execution time. In general, the execution time will increase along with the growth of complexity of the models under test.

It would be more helpful if we could show how the distribution and communication times contribute to the total execution time. However, parallel computing toolbox is a high-level application programming interface (API), so we may only identify the total execution time, and there is no way to get the communication time between the host and workers. In the future work, therefore, we need to use other low-level APIs to ameliorate and further analyze the effect of communication time on the total mutant execution time.

5.2. Parallel Mutation Testing on Many Machines

The first experiment was conducted by parallelizing on only one multicore computer. To prove for the effectiveness of parallel programming with Matlab, we scaled up the distri-

bution strategy using parfor to much larger number of workers in the different environments using the MDCS. This ex- periment was performed in two environments with different characteristics, one of them being a homogeneous system (four computers with the same processor, memory and operating system) and the other a heterogeneous system (two computers with the different processors and memories), which are called DCS_A and DCS_B, respectively. The DCS_A environment is made up of four homogeneous computers (with 2.4 GHz Intel Core 2 Quad CPU Q6600 and 2 GB memory, running the Windows 7), which are configured for running four workers on each computer. The DCS_B environment is made up of two heterogeneous computers, including one computer (CPU Intel Xeon E5520 2.27 GHz and 8 GB RAM) running eight workers and one computer (AMD Operon Dual-Core 2.27 GHz, 4 GB memory, running the Windows Server 2008 operating system) running four workers.

Table 3
The total time in seconds for each model on two different environments using MDCS

Model	DCS_A	DCS_B
CheckInputs	74.99	189.35
Quadratic_v1	119.48	129.51
Quadratic_v2	133.21	148.67
RandMdl_v2	125.76	132.23
SimpSw	112.45	117.59
SmokeDetector	187.88	221.21
Tiny	126.87	173.48
CalcStartProgress	340.67	437.72

A drawback of PCT is that it can only get use of the maximum of 12 workers on a multicore machine. Hence, the purpose of this experiment is to show that parallel mutation testing can be executed on many machines with homogeneous and heterogeneous configurations by running the Matlab Distributed Computing Server. This ability is useful to execute mutation testing for large Simulink models.

As shown in Table 3, the parfor mechanism on the homogeneous combination of many machines (DCS A) with 16 workers takes much less time compared with that using 8 workers on a single multicore machine with the same configuration as shown in Table 2. It is also noted that the parallel execution for small models with a few mutants is less efficient than that for large models, because, in such cases, the communication cost is higher than the execution cost.

The goal of this subsection is too prove that mutation testing can be scaled up to many computers by running it on MDCS. Results in Table 3 have not yet concluded that the shorter execution time for the DCS_A configuration results from its homogeneity, or that it has more workers, as the number of workers in two different environments is not equal. The influence of the configuration type on the mutation execution time is out of scope of this paper, and it will be figured out more carefully in the future studies.

6. Conclusion

Parallel execution helps reduce the computational cost, which is one of the biggest problems in mutation testing, and increase the efficiency without compromising the effectiveness. Three different strategies were implemented by distributing different subsets of mutants to the workers, and an experimental comparison of these three distribution algorithms was conducted. The experimental results demonstrated that the distribution strategy of mutants using the parfor scheme is the best one.

As discussed above, the parallel execution is only useful for large models with a large number of mutants, so mutation testing should be done on sequential machines when models are small and moved to parallel machines only if the size of models requires much execution time.

Another problem is that the communication overhead is fairly high because the client broadcasts one test case at a time to all workers, and some small models do not require much time for mutant interpretation. To decrease the communication cost in all cases, test cases could be sent to the workers in blocks, since few large messages (n test cases at a time) are processed more efficiently than many small messages (one test case at a time) on multicore machines. This offers a great potential to improve the performance of MuSimulink in the future work.

Moreover, workers often sit idle waiting for the slowest node to finish executing mutants. If nodes are allowed to request work from the host rather than wait for the host to send the next test case, then the idle time could be significantly reduced. This demand-driven strategy may restrict overhead to the time necessary for communicating test case information.

Finally, mutants are assigned to processors before they are interpreted, and there is no way to redistribute mutants during interpretation if one or more processors become overworked. With dynamic load balancing, mutants can be reassigned during interpretation so that each processor performs approximately the same amount of work. In the future work, the study will be extended to validate this direction.

References

[1] B. Beizer, *Software Testing Techniques*, 2nd ed., Thomason Computer Press, 1990.

[2] R. H. Untch, A. J. Offutt, and M. J. Harrold, "Mutation analysis using program schemata", in *Proc. Int. Symp. on Software*, Cambridge, Massachusetts, 1993, pp. 28–30.

[3] K. N. King and A. J. Offutt, "A Fortran language system for mutation based software testing", *Software: Practice and Experience*, vol. 21, no. 7, pp. 685–718, 1991.

[4] E. F. Barbosa, J. C. Maldonado, A. Marcelo, and R. Vincenzi, "Toward the determination of sufficient mutant operators for C", *Software Test., Verif. and Reliabil.*, vol. 11, no. 2, pp. 13–136, 2001.

[5] A. J. Offutt, G. Rothermel, R. H. Untch, and C. Zapf, "An experimental determination of sufficient mutant operators", *ACM Trans. on Software Engin. and Methodol.*, vol. 5, no. 2, pp. 99–118, 1996.

[6] W. E. Wong and A. P. Mathur, "How strong is constrained mutation in fault deletion", in *Proc. Int. Computer Symp. ICS'94*, Hsinchu, Taiwan, Republic of China, 1994, pp. 515–520.

[7] W. E. Wong, J. C. Maldonado, M. E. Delamaro, and A. P. Mathur, "Constrained mutation in C programs", in *Proc. 8th Simpósio Brasileiro de Engenharia de Software SBES 94*, Curitiba, PR, Brazil, 1994, pp. 439–452.

[8] M. Polo, M. Piattini, and I. Garcia-Rodriguez, "Decreasing the cost of mutation testing with 2-order mutants", *Softw. Test. Verif. Reliab.*, vol. 19, no. 2, pp. 111–131, 2008.

[9] Y. S. Ma, J. Offutt, and Y. R. Kwon, "MuJava: an automated class mutation system", *Software Test., Verif. and Reliabil.*, vol. 15, no. 2, pp. 97–133, 2005.

[10] Y. Jia and M. Harman, "An analysis and survey of the development of mutation testing", *IEEE Trans. on Software*, vol. 37, no. 5, pp. 649–678, 2011.

[11] P. R. Mateo and M. P. Usaola, "Mutation testing cost reduction techniques: a survey", *IEEE Software*, vol. 27, no. 3, pp. 80–86, 2010.

[12] A. P. Mathur and E. W. Krauser, "Modeling mutation on a vector processor", in *Proc. 10th Int. Conf. on Software Engin. ICSE'88*, Singapore, 1988, pp. 154–161.

[13] E. W. Krauser, A. P. Mathur, and V. J. Rego, "High performance software testing on SIMD machine", *IEEE Trans. on Software Engin.*, vol. 17, no. 5, pp. 403–423, 1991.

[14] B. Choi and A. Mathur, "High-performance mutation testing", *J. of Syst. and Software*, vol. 20, no. 2, pp. 135–152, 1993.

[15] R. A. DeMillo, D. S. Guindi, K. N. King, W. M. McCracken, and J. Offutt, "An extended overview of the Mothra software testing environment", in *Proc. 2nd Worksh. on Softw. Test., Verif. and Anal.*, Banff, Alberta, Canada, 1988, pp. 142–151.

[16] A. J. Offutt, R. P. Pargas, S. V. Fichter, and P. K. Khambekar, "Mutation testing of software using a MIMD computer", in *Proc. Int. Conf. on Parallel Process. ICPP 1992*, Chicago, Illinois, USA, 1992, pp. 257–266.

[17] V. N. Fleyshgakker and S. N. Weiss, "Efficient mutation analysis: a new approach", in *Proc. Int. Symp. on Software Test. and Anal. ISSTA 1994*, Seattle, WA, USA, 1994, pp. 185–195.

[18] S. N. Weiss and V. N. Fleyshgakker, "Improved serial algorithms for mutation analysis", in *Proc. Int. Symp. on Software Test. and Anal. ISSTA 1993*, Cambridge, MA, USA, 1993, pp. 149–158.

[19] P. R. Mateo and M. P. Usaola, "Parallel mutation testing", *J. of Software Test., Verif. and Reliabil.*, vol. 23, no. 4, pp. 315–350, 2013.

[20] P. R. Mateo and M. P. Usaola,"Bacterio: Java mutation testing tool: A framework to evaluate quality of tests cases", in *Proc. of the Int. Conf. on Software Mainten. ICSM 2012*, Trento, Italy, 2012, pp. 646–649.

[21] S. F. Hummel, E. Schonberg, and L. E. Flynn, "Factoring: A method for scheduling parallel loops", *J. of Commun. of ACM*, vol. 35, no. 8, pp. 90–101, 1992.

[22] C. C. Pablo, G. M. Mercedes, and N. Alberto, "EMINENT: EMbarrassINgly parallEl mutatioN Testing", *Procedia Comp. Science*, vol. 80, pp. 63–73, 2016.

[23] I. Saleh and K. Nagi, "Hadoopmutator: A cloud-based mutation testing framework", in *14th Int. Conf. on Software Reuse ICSR 2015*, Miami, FL, USA, 2014, pp. 172–187.

[24] Matlab Inc. [Online]. Available: http://www.mathworks.com/products/simulink/ (accessed on March 10, 2017).

[25] K. Ghani, J. A. Clark, and Y. Zhan, "Comparing Algorithms for Search-based Test Data Generation of Matlab Simulink Model, in *Proc. 10th IEEE Congr. on Evol. Comput. CEC'09*, Trondheim, Norway, 2009, pp. 2940–2947.

[26] R. DeMillo, R. Lipton, and F. Sayward, "Hints on test data selection: help for practicing for programmer", *IEEE Computer*, vol. 11, no. 4, pp. 34–41, 1978.

[27] L. T. M. Hanh and N. T. Binh, "Mutation Operators for Simulink Models", in *Proc. of the 4th Int. Conf. on Knowl. and Syst. Engin. KSE 2012*, Danang, Vietnam, 2012, pp. 54–59.

[28] T. A. Budd and D. Angluin, "Two notions of correctness and their relation", *Acta Informatica*, vol. 18, no. 1, pp. 31–45, 1982.

[29] L. T. M. Hanh and N. T. Binh, "Automatic generation of mutants for simulink models", in *Proc. 16th Nat. Conf.: Selec. Problems About IT and Telecommun.*, Danang, Vietnam, 2013, pp. 339–346.

[30] A. Krishnamurthy, S. Samsi, and V. Gadepally, "Parallel Matlab techniques", in *Image Processing*, Y.-S. Chen, Ed. InTech, 2009 [Online]. Available: http://www.intechopen.com/books/image-processing/parallel-matalab-techniques

[31] G. Sharma and J. Martin, "MATLAB: A language for parallel computing", *Int. J. of Parallel Programm.*, vol. 37, no. 1, pp. 3–36, 2009.

Le Thi My Hanh is currently a lecturer at the Information Technology Faculty, University of Science and Technology, Danang, Vietnam. She gained her M.Sc. degree in 2004 and the Ph.D. degree in Computer Science at the University of Danang in 2016. Her research interests are about software testing and, more generally, application of heuristic techniques to problems in software engineering.
Email: ltmhanh@dut.udn.vn
The University of Danang
University of Science and Technology
54 Nguyen Luong Bang, Lien Chieu
Danang, Vietnam

Nguyen Thanh Binh graduated from The University of Danang, University of Science and Technology in 1997. He got a Ph.D. degree in Computer Science from Grenoble Institute of Technology (France) in 2004. He is currently associate professor of the Information Technology Faculty, The University of Danang, University of Science and Technology, Vietnam. He has been dean of Information Technology Faculty at The University of Danang, University of Science and Technology since 2010. He has been directing a research team since 2009. His research interests include software testability, software testing and software quality.
Email: ntbinh@dut.udn.vn
The University of Danang
University of Science and Technology
54 Nguyen Luong Bang, Lien Chieu
Danang, Vietnam

Khuat Thanh Tung received the B.Sc. degree in Software Engineering from University of Science and Technology, Danang, Vietnam, in 2014. Currently, he is participating in the research team at DATIC Laboratory, University of Science and Technology, Danang. His research interests focus on software engineering, software testing, evolutionary computation, intelligent optimization techniques and applications in software engineering.

Email: thanhtung09t2@gmail.com
The University of Danang
University of Science and Technology
54 Nguyen Luong Bang, Lien Chieu
Danang, Vietnam

Paper

Detecting Password File Theft using Predefined Time-Delays between Certain Password Characters

Khaled W. Mahmoud, Khalid Mansour, and Alaa Makableh

Department of Computer Science, Zarqa University, Zarqa, Jordan

https://doi.org/10.26636/jtit.2017.112517

Abstract—This paper presents novel mechanisms that effectively detect password file thefts and at the same time prevent uncovering passwords. The proposed mechanism uses delay between consecutive keystrokes of the password characters. In presented case, a user should not only enter his password correctly during the sign-up process, but also needs to introduce relatively large time gaps between certain password characters. The proposed novel approaches disguise stored passwords by adding a suffix value that helps in detecting password file theft at the first sign-in attempt by an adversary who steals and cracks the hashed password file. Any attempt to login using a real password without adding the time delays in the correct positions may considered as an impersonation attack, i.e. the password file has been stolen and cracked.

Keywords—access control, intrusion detection systems, network security, password protection.

1. Introduction

The wide spread of communication methods, especially through the Internet expedites various computational tasks and facilitates data sharing amongst people and organizations. Several methods were proposed in the literature to enhance data security and protect privacy. Customers' passwords are amongst the most important data to secure. According to what was recorded in the last few years about password file thefts, obtaining the maximum level of protection to secure password files and detecting password file thefts is still a major challenge in the digital system world [1].

As textual passwords are still widely used as an authentication mechanism and they are unlikely to disappear in the near future [2], [3], passwords should be selected and saved carefully. Many password file theft incidences were reported recently [4], [5], and big companies have been exposed to password database theft such as Yahoo, Hotmail, Social Security Administration [1]. This urges security community to develop methods to increase the immunity of passwords and effectively detects the theft of password files. Moreover, timely discovering the theft of password files is a very important issue. Storing passwords properly is a key issue in protecting them.

This paper proposes adding an extra level of security that puts the adversary at risk of being detected at the first login attempt. This helps companies to deny any access from adversary who is trying to impersonate users and access the system as a legitimate user. The proposed mechanisms are built on the password hardening technique that was presented in [6] and [7]. This password hardening technique depends mainly on the way users perform the sign-up step. When a user leaves relatively large time gaps between certain consecutive password characters, an adopted keying pattern is created and associated with the password. It allows to both harden the password and detect password file thefts.

The rest of this paper is organized as follows. Section 2 presents the common password file structures. Section 3 reviews the related work. In Section 4, the idea of password pattern is explained. Section 5 shows the proposed approaches for detecting password file theft. Finally, research conclusions and suggestions for further studies are discussed in Section 6.

2. Password File Structure

Password-based authentication systems require password files or tables to organize users' login information. In this critical file, which should be kept secret, users' IDs are stored in plaintext. However, the passwords are stored according to the security policy adopted by the used system. Three types of security policies exist:

- Plain text – some servers store passwords as a plaintext. In this case, security depends entirely on the password file secrecy (i.e. protecting the file from being stolen). Once the authentication server is compromised and the password file is stolen, the adversary can impersonate users and logs in using the correct passwords.
- Hashed passwords – hash function (or algorithm) is a function that can be used to transform the input data into a fixed size output (hash value) regardless the input size. This function is a one-way function because it is easy to get the result for any given input but it is very difficult or impossible to get the input backward given the result and the function. Many hash functions exist and some have many versions

such as: SHA1, SHA256, SHA512, MD2, MD4 and MD5. In this type of security policy, the result of hashing the original password $H(P)$ is stored in the password file. Unfortunately, this way is not sufficient against brute force and dictionary attacks. In case the list of hashed passwords is stolen, the attacker can apply offline brute-force attack to find a password with a hash value equivalent to the value stored in the stolen list. Later on, the adversary can impersonate the user and logs in using the correct password.

- Hash + salt – salt is a random data that is appended to the password before passing it to the hash function $H(p||s)$. Salt is used to increase the amount of cracking time by increasing the size of passwords space. In addition, it is used to overcome the fact that many user's accounts have the same password and so their hash values. Despite this enhancement, passwords are still under leakage by attackers.

3. Literature Review

Several techniques are proposed in the literature to either harden the cracking process or to discover the theft as soon as possible.

Injection "deceive" in password-based systems is a general trend that aims to hide the correct passwords and to enable fast discovery of password file thefts. For example, some approaches suggest to insert fake accounts in the passwords file or insert extra fake password files [8], [9], any use to those accounts or files indicates an intrusion [10]. More about those approaches are given next.

3.1. Kamouflage

Kamouflage is proposed as an architecture for building theft-resistant password managers [11]. It aims to protect local password database (e.g., stored on a laptop or a cell phone) by adding a list of statistically indistinguishable decoy passwords to hide the real list. Decoy sets are generated using certain rules based on the correct passwords. In case the attacker decrypts passwords, it is difficult for attackers to distinguish between the correct password set and the decoy sets.

3.2. Honeywords

Honeywords are sets of decoy passwords that are added to each user accounts to deceive the adversary [10]. Honeywords, which are similar to user-selected passwords, are stored in hashed form along with the correct password, i.e. the structure of the password file is altered by adding a list of candidate passwords for each user rather than a single password for each account. In this case, the adversary faces a list of candidate passwords for each user and this makes the process of distinguishing between honeywords and the correct user's password harder for the adversary who steals and compromises a password file. Consequently, using any of these honeywords to login will generate an alarm to the system administrator. This alarm implies that the passwords file was stolen and cracked and an illegal login attempt is recorded.

The honeywords system requires an extra secure server called honeychecker to store the indexes of correct passwords. The server is connected to the main server through a dedicated line. This server aims to improve the overall security by separating the information related to users' authentication process. During the login process, if the entered password is not in the list associated with the user then the login failed. If a match is found, then the index of this match is sent to the honeychecker for verification. If the index is correct, then the user is authenticated. Otherwise, the honeychecker will raise a silent alarm to the system administrator and return false to the main server to deny the login.

To ensure the effectiveness of the honeywords approach, honeywords should be generated in a manner that makes every sweetword (sweetwords list is a set of all honeywords plus the correct password) to seem strongly as a candidate password. This makes it difficult for the adversary to guess the correct password. In addition, if a wrong password is used, a password alarm can be raised by the system. Different methods to generate honeywords can be found in [10]. Some of these methods can be done without users' intervention while others require user intervention (i.e. by modifying the sign-up interface). More methods can be found in [12], [13].

An enhancement that was presented in [14] aims to increase the effectiveness of the honeywords system and to overcome more active attack scenarios by adding phone number as an extra information field for each user in the honeychecker database. Honeychecker uses phone number to communicate with the user in special cases such as password change or many invalid sign-in operations.

3.3. Paired Distance Protocol

The article in [13] proposed a new honeywords system based on paired distance protocol (PDP) with less storage overhead. In PDP, the user chooses a random string (RS) that will be appended to the password during registration. A distance chain – which is derived from RS – is used to replace the honeywords for each user account. Distance chain is a set of paired distances (separated by "–") between every two consecutive elements of RS. This distance is calculated using a secret honey circular list (hcl) that holds the keyboard letters and digits spread in random order (see Fig. 1). As an example, the distance chain for the random string $fa5$ is (20–22), i.e. the number of elements that is traversed in clockwise between f and a in hcl is 20 and the number traversed in clockwise between a and 5 is 22.

The password file contains the username, the hashed password and the distance chain for the RS. The honeychecker is still needed to store the username and the first character of the RS. During login, if the user submits an incorrect

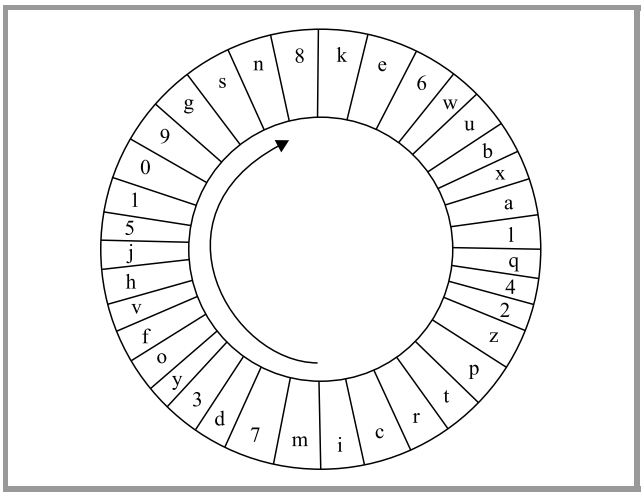

Fig. 1. Random order for 36 letters and digits in honey circular list.

password, then the system will reject the user login directly. If the password is correct, then the system will calculate the distance chain from the given RS and compare it with the stored distance chain in the password file. If no matches are found, then the login will be denied. If the derived distance chain gets matched with a stored one, then the system will match the first character in the given RS with the character stored in the honeychecker. If a match is found then the login is permitted. The success of PDP is associated with the high randomness in choosing RS. PDP provides security against multi-system attacks and DoS.

3.4. Two Password Files Model

The paper in [12] proposed a new system that simulates the honeywords system. However, the difference is that the proposed mechanism stores the user's authentication information in two password files F1 and F2 that are located in the main server.

- F1 is a two-column table. The first column is a list of all usernames sorted in an alphabetical order. A list of indexes that are randomly selected from the first column in F2 is stored in the second column beside each user. One of these indexes is the index of the correct password. The correct index is paired with the username and delivered to honeychecker.
- F2 is a two-column table. The first column is just an index column and the second column is the hash value of each account's password. Some of these accounts are decoy user accounts (or honeypots). They are created by the administrator during system initialization.

During user login, the system authenticates the user as follows:

1. The system goes through F2 in order to find the index of the given password.
2. If this index is not in the corresponding list of this user in F1 then the login fails directly.
3. Otherwise, the system checks if the returned account is a honeypot account. If yes, the system will raise an intrusion alarm.
4. If this account is not a honeypot account, then the returned index is sent to the honeychecker to check if it is the correct index. If no, the honeychecker will again raises an intrusion alarm.

3.5. ErsatzPasswords

ErsatzPasswords scheme is another solution provided in [4] to control the problem of password file theft and detect the leakage attempts. The proposed scheme uses a machine-dependent function (HDF) at the authentication server to harden the off-site password discovery. Adding this hardware supports the software system and makes the cracking process impossible without accessing the target machine physically. Moreover, a single password file identical to the traditional password file exists and no additional server is needed. Whenever the attacker attempts to crack the password file offline, he/she uncovers ersatzpasswords, i.e. fake passwords, which will raise an alarm in case of their use. The following steps are used during sign-up operation:

1. Each user presents his username, password p and ersatzpassword p^*.
2. Calculate the salt $s^* = \mathbf{HDF}(p) \oplus p^*$.
3. Calculate $B = H(\mathbf{HDF}(p) \oplus s^* || s^*)$.
4. B and the associated username are stored only in the password file.

During the login process, if the presented password is correct and B is equal to the stored value then the user is successfully authenticated. If the presented password was the ersatzpassword p^* (i.e. the password file was stolen and the hash values were inverted), then the result of $H(p^*||s^*)$ will be equal to the stored B and consequently an alarm will be raised. Finally, if there is no match, then this can be treated as an error login.

3.6. PolyPasswordHasher

PolyPasswordHasher is a software only solution proposed in [5] that reduces the possibility of cracking user's passwords by making them interrelate. The basic idea in this mechanism is to XOR a secret share with a salted password hash at the account creation time, where Shamir secret sharing [15] method is used to create the shares. The result of XORing is stored in a password file given that neither the share nor the salted password hash exists on the disk. To specify which share was used, an extra field called the share number is added to the password file. This scheme prevents the adversary from validating the hash value for password,

and the attacker cannot gain any information from compromising the password file because the share and the password hash value are not stored on the disk. However, if the adversary reached the arbitrary memory on the server then he can steal the secret.

3.7. Synergetic Authentication

The study in [16] proposed Synergetic Authentication (SAuth) protocol. A user can be authenticated by a site if and only if true credentials are used at that site and another vouching site voted for authenticating that user. For example, let user A has an account on the system S and also has an account on the system V. The login request of A to access S will not be accepted unless the system V sends vouching message to system S ensure the truth of A information. So, if an adversary compromised the database of the system S and tries to impersonate the user A his try will fail. The adversary must compromise both systems S and V simultaneously to success in this attack.
The design of SAuth system provides a mechanism that can detect the activities that seem irregular. If the vouching system received several user authentication requests ended with fail, this could be considered as an attempt to impersonate a certain user(s). The major limitation to this mechanism is when users have the habit of password reusing. To alleviate this risk, decoy passwords are proposed. The other limitation is the need for synchronizing between different services.

4. Hardening Passwords by Adding Delays between Keystrokes

Text passwords are common, easy and reliable authentication method. To ensure the desired goal of access protection, the creation of passwords and how they are stored need to be managed carefully. Several research papers were dedicated to harden passwords and make them stronger. Some of these methods verify not just the knowledge of the password, but also verify other credentials such as: the possession of a specific token, the current GPS location of the user [17], the fingerprints [18], the signature or something individuals can be characterized with such as keystroke dynamics biometrics [19], [20].
In this paper, a modified version of the method that was proposed in [6] to strengthen passwords is adapted and used in detecting passwords file theft. In Mansour's method, users need to type their passwords in certain rhythms during the sign-up process. During sign-up, the user is required to enter his password twice. The password is accepted only if the two trials return the same keying pattern. Otherwise, the user is instructed to repeat entering the password twice again. For a successful login, a user needs to key password with the same sign-up rhythm. A strong password is formulated based on classifying the delay times between consecutive password characters into either slow or fast based on a certain threshold value. The exact delay time values are not critical since the keying rhythm is classified into fast and slow regardless the exact delay times.
In the modified version and in order to simplify the registration process, the user is deliberately declaring the location(s) of large delay. The users are instructed to enter their passwords normally without leaving large delays and then select – for example by mouse – the large-delay location(s). A special sign-up interface is designed and some components (such as check boxes) to specify the delay positions are added. At sign-in, the login is accepted only if the user enters the correct password with large delay in the specified position(s). The user should leave a delay time at the selected position(s) that exceeds normal typing. As an illustration, if a user specifies a delay after the third character, then the system will not allow the user accessing the system unless he/she types in the correct password and leave relatively large time delay after the third character. Any accidental waiting time during typing password will be considered as a large delay and will lead to reject the login if this delay is in the wrong position. For more details refer to [7].
It should be clear that this approach of password hardening is different from password hardening based on keystroke dynamics. Authentication systems based on keystroke dynamic capture normal typing behaviors of users using long training session(s). In these systems, the users are authenticated only if they write the correct password using their way of writing (i.e. rhythm) [21]. In proposed system, users predefined the keystroke pattern rather than using his human patterns. Consequently, proposed system does not require training or analyzing the keying behaviors of users. In addition to hardening textual passwords, the above idea is extended to detect a possible password file theft. The next section shows how the adopted keying pattern data can be used to detect password file theft.

5. Password File Theft Detection

According to what was recorded in the last few years about password file thefts, obtaining the maximum level of protection to secure password files and detecting any violation over the password file is still a challenge in the digital system world [1]. In this section, three novel password file theft detection mechanisms are presented. The first two mechanisms require an auxiliary secure server, while the third one uses login server only.

5.1. Preliminaries

The following notations are defined for the i-th user: u_i is the username, p_{ij} is a honeyword number j that belongs to user i. The honeywords plus the correct password p'_i forms what we call sweetwords. k is the number of sweetwords, c_i is the index of the correct password for user i, and dp_i is the delay pattern. The delay pattern is a sorted comma-separated list of all positions where user i should leave

large delay. For example, if the large delays come after the fourth, the second and the seventh character, then the delay pattern is $<2,4,7>$. Finally, the *delaylist* is a list of time delays between every two successive password characters.
In proposed system, two types of servers can be used: login server and auxiliary server (or honeychecker). The login server maintains the password file while the auxiliary server maintains the authentication file, which stores data related to the correct passwords. We assume that a dedicated line for communication between honeychecker and the login server is used. In addition, both types of servers are able to raise an alarm if a password file disclosure is detected.

5.2. Honeychecker Server-based Mechanisms

In the first mechanism a traditional password file structure is used to store usernames and the hash value of correct passwords. The honeychecker server stores users' names and their associated delay patterns *dp*.
In case the password file was hacked and cracked, the system makes it more difficult for the adversary to access the system since the password has to be keyed in a certain pattern given that this pattern does not exist in the login server. With every accessing attempt, if the entered password was correct, its keying pattern *user-dp* is extracted and sent to the honeychecker server. If the pattern is correct, then it is considered a successful attempt, otherwise it is considered a possible password file theft, see Algorithm 1.

Algorithm 1. Login and user authentication algorithm for the first mechanism

```
u ← read the entered username
(p, delaylist) ← read the entered password and record
the delays between each successive character.
search the password file for the user u
if u is not stored in the password file then
   return login-fail
else
   p' ← get the correct password from password file
end if
if p ≠ p' then
   return login-fail
else
   user-dp ← extract-the-delays (delaylist)
   send u and user-dp to the honeychecker
   stored-dp ← get the stored delay pattern for this user
   from honeychecker
   if user-dp = stored-dp then
      return login-succeed
   else
      raises a theft alarm
   end if
end if
```

To extract the delay pattern from the collected delaylist, a detailed algorithm is given in [7]. It sorts the list of delays in an ascending order a long with the original position of each delay value. Then it computes the difference between every two-successive delay in the sorted list and returns the location of the maximum difference. All elements appear after this location is considered the user delay pattern.
This mechanism is characterized by its simplicity and effectiveness since neither extra storage space is needed to store honeywords nor extra effort is needed to design these words.
If the password was only correct, we cannot be sure that this case is a theft of password file since a legitimate user can enter his password without its correct pattern for various reasons such as fatigue and forget cases (false positive). However, when the system raises an alarm for possible file theft, the administrator needs to take further actions to check the current situation.
To further protect passwords in case of password file theft, the next mechanism augments honeywords with correct password in the password file.
The second mechanism uses password file with honeywords. With this mechanism, each user account in the password file is associated with a list of honeywords as shown in Table 1. Note that, the sweetwords are stored in the password file without any information related to the keystroke latencies.

Table 1
Passwords file structure stored in login server based on honeywords system

Username	Sweetwords
u_1	$\{p_{11}, p_{12}, p_{13}, p'_{14}, \ldots, p_{1k}\}$
u_2	$\{p_{21}, p_{22}, p_{23}, p_{24}, \ldots, p'_{2k}\}$
$\ldots$	$\ldots$
u_n	$\{p_{n1}, p'_{n2}, p_{n3}, p_{n4}, \ldots, p_{nk}\}$

The honeychecker (auxiliary secure server) is required in the authentication process. In addition to the username and the delay pattern *dp* a new column is added in the authentication file to store the index c_i of the correct password p'_i in the list of sweetwords (see Table 2).

Table 2
The authentication file structure stored in honeychecker

Username	Index (c_i)	Delay pattern (dp_i)
u_1	c_1	dp_1
$\vdots$	$\vdots$	$\vdots$
alice	3	3,6
$\vdots$	$\vdots$	$\vdots$
u_n	c_n	dp_n

As any normal login process, the user enters his credential information, i.e. username and password. The system captures username u, password p and the keystroke laten-

cies between every two successive characters. Firstly, the system checks whether the entered password p is in the sweetwords list. If p does not exist then the login is denied. Otherwise, the index of this sweetword (call it y) is sent to the honeychecker paired with the username u and the users' delay pattern that was extracted from delaylist. The honeychecker checks whether y matches the stored index c for this user. If no match is found then the honeychecker raises a theft alarm, else the honeychecker checks for the correctness of the delay pattern. If the password p is entered correctly (i.e. with relatively large time gap in the specific positions), the honeychecker returns a message to accept the login, otherwise a theft alarm is raised. This process is shown in Algorithm 2.

Algorithm 2. Login and user authentication algorithm for second mechanism

```
u ← read the entered username
(p, delaylist) ← read the entered password and record
the delays between each successive character.
search the password file for the user u
if u is not stored in the password file then
  return login-fail
else
  sweetwords ← get the list of sweetwords from login
  server
end if
if p ∉ sweetwords then
  return login-fail
else
  y ← get the index of p in sweetwords
  user-dp ← extract-the-delays (delaylist)
  send u, y and user-dp to the honeychecker
  (c, stored-dp) ← search the authentication file for the
  user u and return the index of the correct password
  and its delay pattern.
  if y ≠ c then
    raises a theft alarm
  else
    if user-dp = stored-dp then
      return login-succeed
    else
      raises a theft alarm
    end if
  end if
end if
```

According to the honeywords system presented in [10], when the attacker gets the password file and inverts the hashed values, any use of honeywords (wrong password) will raise a theft alarm. However if accidentally the correct password is picked, the attacker gets in. In proposed mechanism, a new layer of protection has been added. If a user accidentally tries the correct password without keying it according to the predefined delay pattern, the system raises a theft alarm. Consequently, the chance for password file theft detection is increased.

To have a more secure authentication system, the number of sweetwords per user suggested in [10] was 20 and can reach 1000 in some circumstances. In proposed mechanism, the number of sweetwords can be much less due to the fact that guessing the correct password does not guarantee successful access to the system. If the attacker was able to guess the correct password due to the existing small number of sweetwords, the second piece of information (i.e. delay pattern) gives a second layer of protection. Moreover, the problem of designing honeywords become easier for small set of sweetwords.

5.3. Login Server-based Mechanism

Some authentication systems suffered from increasing the required storage area such as adding $k-1$ honeywords for each user as in [10], or adding additional file as in [12]. The storage cost is increased as the number of users in the system increases. Hence, storage optimization becomes an issue. Moreover, an auxiliary server is also needed in the authentication process. Using this server is considered an extra storage cost. Eliminating this server may simplify the authentication process. In this section a new mechanism that requires no honeychecker server and no honeywords is introduced, where each password by itself works as a password and a honeyword at the same time. The trap is in a single password rather than a list of honeywords.

The passwords file (see Table 3) has the same structure as in traditional passwords file. It contains two columns of information: usernames and hashed passwords $h(uas_i)$ where *uas* stands for "user authentication string". The user authentication string is actually a concatenation of two parts: p'_i and dp_i. For example, if user i selects his password as me@me12 and the delay pattern was $<2,5,7>$, then uas_i= me@me12257.

Table 3
Password file structure

Username	Password
u_1	$h(uas_1)$
u_2	$h(uas_2)$
$\vdots$	$\vdots$
u_n	$h(uas_n)$

During the login process, the user enters his credential information (username and password). The password should be keyed in the predefined pattern. The first authentication step checks if the entered password p is equal to *uas*. If yes, raise a password file theft alarm. This step adds a security level that aims to detect password file theft and cracking as early as possible. If there are no penetration signs, the system extracts the user delay pattern from the delaylist and checks whether *h(p || user-dp) = h(uas)*. if yes then the login is succeeded otherwise the login is failed. This process is shown in Algorithm 3.

This mechanism is a simple one with no extra storage overhead; no need to k honeywords. Clearly, if the system

has n users then no more than n authentication values are stored. Moreover, no auxiliary server is needed. Additionally, it has a normal password file structure. Such normality makes the adversary believe that the cracking results are original passwords and this increases the possibility of trapping him.

Algorithm 3. Login and user authentication algorithm for the third mechanism

```
u ← read the entered username
(p, delaylist) ← read the entered password and record
the delays between each successive characters.
search the password file for the user (u)
if u is not stored in the password file then
   return login-fail
else
   h(uas) ← get the stored hashed password
end if
if h(p) = h(uas) then
   raises a theft alarm
else
   user-dp ← extract-the-delays (delaylist)
   if h(p || user-dp) = h(uas) then
      return login-succeed
   else
      return login-fail
   end if
end if
```

Finally, to further disguise the cracked passwords, dp can be encoded into alphabets and inserted in a position that depends on the average of dp. The insertion positions can vary for different users depending on their delay patterns.

6. Conclusion

The proposed mechanisms use a new strong password that is based on augmenting time delays between certain password characters. These delays are used in both accessing systems and detecting password file thefts. Two of the proposed methods use a honeychecker server while the third one applies a different mechanism that does not need a honeychecker server.

The three mechanisms can raise a theft alarm. Particularly, in case of correct password and wrong delay pattern. This can happen in two cases: the first one is when the attacker succeeds in getting the password and tries guessing the delay pattern, while the second case is when legitimate users do certain mistakes in the pattern during the login process. Many actions can be taken in these cases such as blocking the account or sending an alarm to the genuine user. In order to avoid raising an alarm for legitimate users, the system can be tuned such that the system can raise a silent alarm to the administrator for the first failed login but after a few failed login attempts, the system can take stronger actions.

As a future work, large scale empirical study is needed to analyze both the effectiveness of the proposed password hardening system and the ability of the three proposed mechanisms in detecting password file theft.

Acknowledgements

This research is funded by the deanship of scientific research at Zarqa University, Jordan.

References

[1] D. Mirante and J. Cappos, "Understanding Password Database Compromises Technical Report", Tech. Rep. TR CSE-2013-02, Polytechnic Institute of NYU, 2013.

[2] D. Florêncio, C. Herley, and P. C. van Oorschot, "An administrator's guide to internet password research", in *Proc. 28th Large Instal. Sys. Administr. Conf. LISA14*, Seattle, WA, USA, 2014, pp. 44–61.

[3] P. Jadhao and L. Dole, "Survey on Authentication Password Techniques", *Int. J. of Soft Comput. and Engin. (IJSCE)* vol. 3, no. 2, pp. 67–68, 2013.

[4] M. H. Almeshekah, C. N. Gutierrez, M. J. Atallah, and E. H. Spafford, "Ersatzpasswords: Ending password cracking and detecting password leakage", in *Proc. of the 31st Ann. Comp. Secur. Appl. Conf. ACSAC 2015*, Los Angeles, CA, USA, 2015, pp. 311–320.

[5] J. Cappos and S. Torres, "PolyPasswordHasher: Protecting Passwords in the Event of a Password File Disclosure", Tech. Rep., 2014 [Online]. Available: https://password-hashing.net/submissions/specs/PolyPassHash-v1.pdf

[6] K. Mansour, "Adopted keystroke rhythm for password hardening", in *Proc. 11th Int. Conf. on Passwords Passwords 2016*, Bochum, Germany, 2016.

[7] K. W. Mahmoud, "Elastic password: A new mechanism for strengthening passwords using time delays between keystrokes", in *Proc. 8th Int. Conf. on Inform. and Commun. Syst. ICICS 2017*, Irbid, Jordan, 2017, pp. 316–321.

[8] M. J. A. Mohammed Almeshekah and Eugene H. Spafford, "Improving Security using Deception", Tech. Rep. 203-13, Center for Education and Research Information Assurance and Security, Purdue University, West Lafayette, USA, 2013.

[9] F. Cohen, "The use of deception techniques: Honeypots and decoys", *The Handbook of Inform. Secur.*, vol. 3, no. 1, pp. 646–655, 2006.

[10] A. Juels and R. L. Rivest, "Honeywords: Making password-cracking detectable", in *Proc. of the 20th ACM SIGSAC Conf. on Comp. and Commun. Secur. CCS 2013*, Berlin, Germany, 2013, pp. 145–160.

[11] H. Bojinov, E. Bursztein, X. Boyen, and D. Boneh, "Kamouflage: Loss-resistant password management", in *Proc. 15th Eur. Symp. on Res. in Comp. Secur. ESORICS 2010*, Athens, Greece, 2010, vol. 6345, pp. 286–302.

[12] I. Erguler, "Some remarks on honeyword based password-cracking detection", *IACR Cryptology ePrint Archive*, vol. 2014, p. 323, 2014.

[13] N. Chakraborty and S. Mondal, "A new storage optimized honeyword generation approach for enhancing security and usability", *Comput. Res. Repository*, vol. abs/1509.0, p. 8, 2015 (arXiv:1509.06094).

[14] Z. A. Genc, S. Kardas, and M. S. Kiraz, "Examination of a New Defense Mechanism: Honeywords", *IACR Cryptol. ePrint Archive*, vol. 2013, p. 696, 2013.

[15] A. Shamir and A. Shamir, "How to share a secret", *Commun. of the ACM (CACM)*, vol. 22, no. 1, pp. 612–613, 1979.

[16] G. Kontaxis, E. Athanasopoulos, G. Portokalidis, and A. D. Keromytis, "Sauth: Protecting user accounts from password database leaks", in *Proc. of the 20th ACM SIGSAC Conf. on Comp. and Commun. Secur. CCS 2013*, Berlin, Germany, 2013, pp. 187–198.

[17] N. Abdelmajid and K. W. Mahmoud, "Global position system location-based authentication (KERBEROS AS AN EXAMPLE)", *ITEE Journal: Inform. Technol. and Elec. Engin.*, vol. 5, no. 3, pp. 13–18, 2016.

[18] A. K. Jain, A. Ross, and S. Prabhakar, "An introduction to biometric recognition", *IEEE Trans. on Circ. and Syst. for Video Technol.*, vol. 14, no. 1, pp. 4–20, 2004.

[19] P. S. Teh, A. B. J. Teoh, and S. Yue, "A survey of Keystroke dynamics biometrics", *The Scientific World Journal*, vol. 2013, Article ID 408280, p. 24, 2013.

[20] S. P. Banerjee and D. Woodard, "Biometric authentication and identification using Keystroke dynamics: A Survey", *J. of Pattern Recogn. Res.*, vol. 7, no. 1, pp. 116–139, 2012.

[21] P. Dholi and K. P. Chaudhari, "Typing pattern recognition using Keystroke dynamics", in *Mobile Commun. and Power Engin.*, vol. 296, pp. 275–280, 2013.

Khaled Walid Mahmoud received a B.Sc. in Computer Science from Jordan University in 1992, a M.Sc. in Computer Science (Artificial Intelligence) from Jordan University in 1998 and a Ph.D. in Print Security and Digital Watermarking from Loughborough University, UK, in 2004. This was followed by academic appointments at Zarqa Private University (Assistance Professor in Computer Science). His areas of interest include information security, digital watermarking, image processing, AI and Arabic language processing.
E-mail: k.w.mahmoud@zu.edu.jo
Department of Computer Science
College of Information Technology
Zarqa University
P.O. Box 132222, Zarqa 13132, Jordan

Khalid Mansour received his Ph.D. in Computer Science from Swinburne University in 2014. He is currently the head of CS Department at Zarqa University, Jordan. His research interests include automated negotiation in multi-agent systems, web services and information security.

E-mail: kmansour@zu.edu.jo
Computer Science Department
Zarqa University
P.O. Box 132222, Zarqa 13132, Jordan

Alaa Makableh received her M.Sc. degree in Computer Science from Zarqa University in 2016 and a B.Sc. in Computer Information Systems from Hashemite University (Jordan) in 2007. Her research interest is in information security especially knowledge-based authentication systems.

E-mail: alaa.magableh@gmail.com
Computer Science Department
Zarqa University
P.O. Box 132222, Zarqa 13132, Jordan

Paper

Evolution of Measurement of the Single Piece Mail Transit Time

Ryszard Kobus[1] and Frank Raudszus[2]

[1] *National Institute of Telecommunications, Warsaw, Poland*
[2] *Bundesnetzagentur, Mainz, Germany*

https://doi.org/10.26636/jtit.2017.110717

Abstract—The article presents the evolution of measurement of the transit time of priority mail methodology used for postal services quality assessment from the point of view of home and small business senders based on European postal standards. The paper also considers the possibility of test cost reduction by further changes in the measurements methodology.

Keywords—European standards, measurements of transit time, postal services.

1. Introduction

Delivery of single piece mail is a low-cost service addressed to private persons or small entrepreneurs enabling them to send paper documents or items. It is usually posted by letterboxes or postal offices and its route is not registered. Therefore, the service is not subject to complaint and if an item is lost or its delivery is delayed the client cannot receive compensation.
For many years, the quality of service has been monitored by measuring the transit time of test items. The European UNEX measurement system has been operating since 1994 and it tests the transit time of the cross-border priority mail.

2. The First European Standard

The European Commission emphasized the need to establish common rules for the development of community postal services and the improvement of quality of service (QoS) [1]. The Commission identified that the QoS measurement system should include:

- independent end-to-end measurement capabilities,
- a focus on cross-border delivery service performance,
- a single, uniform and reliable system for monitoring delivery service performance within the European Union.

The Commission acknowledged that the many postal traditions and different cultures in Europe would not allow the establishment of one common unified measurement system for domestic mail. Therefore, the national postal systems should have sufficient degrees of freedom to reflect local needs and peculiarities. On the other hand, they should fulfill a defined set of minimum requirements to satisfy the information interests of the national regulatory authority, postal customers as well as postal operators.
The first European standard defining the measurement system was drafted by technical committee CEN-TC331 and published in 2002. The EN 13850 [2] defines the measurement method of the transit time of end-to-end services for single piece priority mail and first-class mail for domestic and cross-border services. The standard was addressed to EU members, i.e. to countries with big postal flows. Standards [3] for second-class mail (EN14508) and for bulk mail (EN14534) measurements were published with a slight delay.
The transit time tests were based on measurements of letters posted and received by test panel, recruited and managed by independent research organization. The mailing should be distributed in at list 30 contractual postal areas. The standard does not define directly the size of the test sample and the size of the test panel but states that they should ensure the accuracy of 1% for domestic and 5% for cross-border measurements.
To ensure high accuracy of the test sample, it should be representative of the real mail. This is achieved by geographical stratification based on the following categories:

- urban – cities including their outskirts,
- rural – smaller cities and communes,
- local I – mail sent to and delivered within the same urban city,
- local II – mail sent within and between rural parts of the same catchment area,
- distance I – within a straight-line distance of 200 km,
- distance II – above a straight-line distance of 200 km.

The basic definitions of outskirts, urban, rural, local and distances are described in the EN standard. The domestic measurements may take national peculiarities into account, by agreement with the regulatory authority. By default, as the city is assumed to be a town above 50,000 inhabitants but in Poland it has been assumed that urban are cities above 10,000 inhabitants.
The sample of test items shall be representative of the real mail for at least the 10 geographical parameters based on: point of posting and point of delivery, for example:

- urban – urban – distance I,
- urban – urban – distance II,
- urban – rural – distance I,
- urban – rural – distance II, etc.

The test items shall have correctly written addresses in according to recommended template and with respect to addressee, delivery address with postcode and for cross-border mail the destination country. The test sample shall represent a statistical sample of the real mail for all discriminant mail characteristics. Generally, the following discriminant mail characteristics should be assumed as a minimum and shall be estimated through real mail studies:

- day of induction as day of the week,
- time of posting for urban areas (it is only important if more than 1 collection time is published),
- all typical methods of delivery.

In real conditions, some test templates can be considered negligible especially in domestic mail. The service delivery at P.O. boxes is often omitted. In Poland, the test mail is paid only by stamps, because other payment methods are rare.
The implementation of EN 13850 was difficult, hence the TC331 workgroup published an implementation guide which describes all stages of preparing and starting the measurements in detail. It should be noted that the application of the rule has limitations, for example:

- the EN 13850 European standard may not be suitable for the measuring of very small volumes of mail;
- a test flow cannot noticeably affect the overall mail traffic. Usually it is assumed that test flow cannot increase the total flow by more than 0.2%;
- the standard may not be suitable for operators with limited coverage. Therefore, the covered geographic area should be large enough so that it can be divided into 30 contractual postal areas;
- this standard is not applicable for measuring the end-to-end transit time distribution of large bulk mailers' services and hybrid mail, which require different measurement systems and methodologies.

The standard makes it possible to conduct measurements with or without electronic system for item tracking, and thus to validate results. In Europe, the transit time of cross-border letters is measured in the UNEX system [4], [5] with Radio Frequency Identification (RFID) active tags. The gates for events registration are installed in sorting centers in all Europe. Such a route tracking system can be used to measure transit time for each part, but this functionality is not covered by this standard.

Generally, the standard is used to measure the achievements of the postal operator in the form of two annual indicators of quality of service.

2.1. On-time Performance

The on-time performance is described as the percentage of postal items delivered within the defined service standard expressed in days. A report should present the level of on-time performance accuracy achieved in the test period.

2.2. Cumulative Distribution of Delivery Days

The cumulative distribution of delivery days is a percentage factor of mail delivered within a given period, from 1 to 10 days. All postal items delivered up to 30 days shall also be considered in the calculations.
It should be added that the resulting indicators cannot be used for direct comparison of achievements by other operators due to different conditions, i.e. geographic areas handled or postal flows, which can make the results incomparable.

3. EU Countries with Less Letter Flows

In 2004, a group of 10 new countries joined the EU. In this group, there are also countries with relatively small populations and thus with low mail flows. An additional problem is that the priority mail service does not enjoy great popularity. It requires reducing the random sample test mail. The developed A1 supplement [6] temporarily solved this problem, but in the meantime work was undertaken to develop a new edition of the standard.
The A1 proposes two methods to decrease the annual test:

1. Extending the period of measurement to two or three years. This way, the annual real mail volume can be smaller while keeping full accuracy. The disadvantage of this approach is that the results for reaching full accuracy after 2 or 3 years should not be reported until these years have passed. After that, the results should continue to be reported annually on a 2- or 3-year rolling basis.

2. Case based on accuracy. The accuracy depends on the sample size which should be as big as necessary to meet the accuracy requirement for each measurement result. If the real mail volume in a given field of study is lower than a certain threshold and the on-time performance is above a certain limit, then the sample size could be decreased.

Figure 1 shows that if on-time performance is higher than 85%, then the sample size can be decreased without loss of accuracy. The expected on-time performance can be estimated based on previous results.

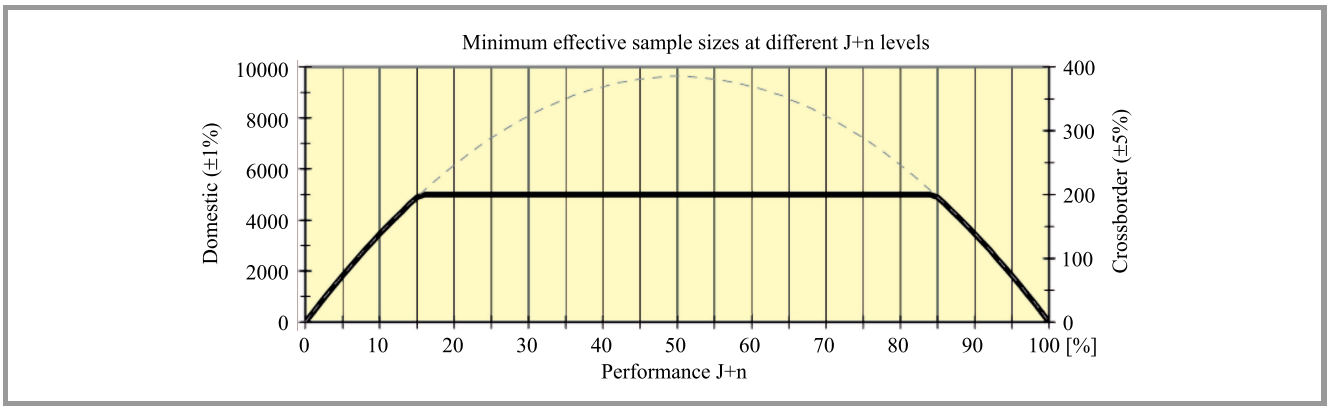

Fig. 1. Possibility of decreasing the sample size [6].

4. The Second Edition of EN 13850

In 2006, a full revision of standard EN 13850 [7] was started and huge modifications were planned. The aim was to increase the flexibility of standards through better adaptation to a random sample of actual traffic and to widen of application area.
The revision team published a survey asking about types of characteristics that are monitored in the real mail studies and their impact on measurements results, which also included questions about the type of geographical stratification used as well as how many cells were incorporated into the design. Results of this survey were implemented in the revision.

4.1. Highly Discriminant Mail Characteristics

The mailing characteristics have to be reviewed at least every three years. The list of possible criteria is given in the standard, and all of them should be checked whether they can be regarded as discriminant or not. The following list of possible characteristics that can be evaluated as a minimum:

1. Mail characteristic referring to the induction/delivery point:
   - type of geographical area by: urban, rural,
   - type of payment by: stamped, metered, postage paid,
   - type of induction by: mail street box, post office, collection from sender's premises, induction in sorting centers,
   - time of posting – only in the case of more than one collection per day,
   - type of delivery by: street address, P.O. box, delivery to receivers' premises.
2. Mail characteristics referring to the test letter itself:
   - formats by at least two modes,
   - weight steps by at least two modes,
   - addressing method by: hand written, typed,
   - weekday of induction.

One can note that the distance is no longer a discriminant mail characteristic in the presented list. Many postal systems use only big automated sorting centers. This means that the item sent locally and at a distance of approx. 200 km may travel on a similar route. Anyway, research conducted in Poland showed that for domestic mail distance has a significant impact on time performance.

4.2. Minimum Sample Size Accuracy

The revised standard gives the minimum sample size (MSS) of exactly 9,625 items, which shall be taken for a domestic measurement system. If expected performance level is greater than 50%, the minimum sample size may be reduced. For example, the 90% performance level can be taken by 3,500 items for domestic measurement system and only 1,850 items for 95% performance level.
The minimum sample size for 50 – 97.5% performance is given in the tables as separate values for domestic and cross-border measurement systems. Generally, the minimum sample size is estimated to ensure 1% accuracy for domestic system and 5% for cross-border. All possibilities to reduce the size of the annual test given in the A1 supplement were transferred, and even extended in second edition of standard. The four categories of countries are defined according to domestic and separately cross-border mail flows. For each category, the optimal solution is proposed. The example for the domestic system is:

- Category 1 – large size mail volumes, i.e. with total annual real mail volume above 500 million mail pieces. Measurement without restriction;
- Category 2 – medium size mail volumes, i.e. with total annual real mail volume of 200–500 million mail pieces. Measurement for countries above J+n[1] performance of 85% – fixed sample size of 4,950 and for countries below J+n performance of 85% the bounded sample size is recommended;

[1]The delivery of the n-th day after posting (J).

- Category 3 – small size mail volumes, i.e. with a total annual real mail volume of 1.5–200 million mail pieces. Measurement for countries above J+n performance of 90% – fixed sample size of 3,500 and for countries below J+n performance of 90% the bounded sample size is recommended;
- Category 4 – very small size flows, i.e. for all flows with volumes below 1.5 million mail pieces per year. Measurement: in case of domestic measurement systems, the test mail can increase the real mail volume in the total field of study by more than 0.25%, undermine the neutrality of such a measurement. Broadening the field of study is recommended here, for example by including further operators.

4.3. Calculation of Accuracy

Three methods of of accuracy estimation are proposed.

Normal approximation. In most cases, the normal distribution will be an appropriate approximation of the binomial distribution. This simple normal confidence interval is symmetrical and easy to use. There are some restrictions on the use for high performance levels when the normal approximation can work poorly even with moderate sample sizes.

But for domestic measurement systems with performance levels up to 96%, the normal confidence interval can be used. In this case, usually at least 50 delayed items are registered by the measurement system. Annex A of the standard explains this problem in detail.

Agresti-Coull approximation. This estimation method is an improved method based on the normal distribution and can be used for all sample sizes with at least 40 items, which is easy to meet. This adapted normal distribution confidence interval is asymmetrical and can be used without sophisticated statistical software.

Inverse beta approximation. This improved estimation method uses the inverse beta function. It is based on the beta distribution, which is the continuous form of the binomial distribution. The inverse beta function is implemented in many software packages for data analysis. For example it is part of Microsoft Excel edition 2010 or newer. The beta distribution confidence interval is asymmetrical and easy to use.

Geographical distribution of the panel. The panels of senders and receivers are dispersed over a geographical recruitment grid, based on postal areas served by the operators. The geographical distribution of the panel is carried out according to random sampling on the whole of the geographical area defined in the field of study. In this range, the standard is very flexible and describes rules for the geographical distribution of the panel both in small countries and big ones.

For operation in bigger countries, a big panel of over 90 panelists distributed in at least in 30 areas is dedicated. In this case, there is no important change compared to the previous edition.

For small countries, a small panel (10 to 90 panelists) and from 4 to 30 postal areas are proposed. The new thing is that in this case number of postal areas is related to the number of panelists and presented in the table. The weekly workload of panelists should be limited to the level of typical user.

The standard proposes a maximum load of 12 letters per week for any domestic sender and 12 letters for any domestic receiver. But the average load of receiver is restricted to 6 letters per week during their time of participation in the measurement period. For a business address a bigger load is allowed – up to 24 letters per week. The weekly workloads given above are the maximum. In many cases, especially in countries with smaller postage flows, the actual workloads should be much lower.

4.4. Design Basic

The design of the measurement system should ensure that the test letters are allocated as a representative sample of all single piece priority mail in the field of study. The best way to achieve a representative sample would be to take as a simple random sample of real mail letters and observe their transit time. Unfortunately given the high measurement accuracy requirement such a solution is unrealistic due to technical reasons. Instead, pre-fabricated test letters are used for measurements and they are sent and received by a group of selected panelists. This design approach requires that the test letters that are added to the existing real mail stream do not differ from it in each mail characteristics, which have a significant influence on the transit time result. The characteristics and modes, which are discriminant, depend in practice on the detailed operation of the dedicated postal system operator. Therefore, a factor which is important to one operator could be not important to others.

The standard describes how to check if a characteristic is discriminant. The test is based on comparison of at least two modes of the characteristic and evaluating their impact on transit time results.

The standard is based on real mail flows and its design is determined by a system of real mail studies which estimate these flows. The real mail studies are performed before or parallel to the first test measurement period. The real mail studies make it possible to:

- consider all single piece priority flows of a given field of study,
- collect statistics on single piece priority real mail flows and real mail characteristics according to requirements of the statistical design.

The real mail studies can be implemented either by the postal operators themselves or by an external body, but they have to be independently audited.

Under the standard the distribution of real mail flows corresponding to certain mail characteristics can be estimated using existing logistic or management data available in the

postal system. These alternative solutions can decrease the cost of design and accelerate the acquisition of necessary data. They may also ensure greater accuracy of estimation than in real mail studies if data is collected by the operator on regular basis.
The standard recommends that real mail studies should be performed at least once every three years. And it is a proper period if the postal market is stable but if postal market is fast-changing then the period should be shorter.

4.5. Implementing Standard

An implementation guide, which was earlier a separate document is now implemented to the standard as Annex H. It contains a scenario which describes successive stages of implementation, from the survey planning phase through its implementation, to reporting of results and auditing. Planning and implementation phases are very time-consuming. The preparation of the test mail survey can take 9 to 18 months. Considering together the measurement period and time of data analysis as well as reporting, the first regular report may be available even after 33 months.
Detailed rules related to all stages of measurements phases and auditing process have been discussed. Furthermore, the attached examples allow for better understanding of the requirements of the standard. Generally, particular attention was paid to the problems of measurements accuracy and corrective weighing.

4.6. Measurements in Multi Operators Environment

Initially, the standard was adapted to the measurements on the liberalized postal market, which in practice means that it should be ready to conduct measurements in the environment of many operators. To fulfil this condition each panelist sender should be able to freely choose the best postal operator for posting test letters. This option was removed from the final draft, because no country except Germany was interested in such a feature.

4.7. Monitoring the Cross Border Priority Mail

The high cost of services forces one to consider the possibility of replacing test letters by the monitoring of real mail transit. This cost effective method can be based on observation of existing components or added in a typical posting technological process. Based on such an assumption, a few testing methods were selected to make a detailed investigation.
The first method is based on information retrieved from sorting machines with proper software. The method seems not expensive, except for a few countries which sort manually cross border priority mail. In addition, there are issues with the compatibility of machines of different generations.
The second approach is based on reading the day of posting, in the destination postal office/sorting center. The date can be retrieved from the stamp, timestamp or franking image. Unfortunately, franking machines and digital stamps are not popular in some countries. In addition, the day printed by franking machines or placed on a digital stamp, is the accounting day of operation, which can be different than the day of posting. Additionally, the franking machines are not used by private senders and small businesses.
This solution does not require any additional components or installation in the postal infrastructure, nor any additional technological operations in the country of origin. But in destination countries new scanners are need or the existing ones need to be modified.
The third approach is based on passive RFID tags which are hidden or placed inside the envelope. The solution needs RFID scanners in the country of origin and destination to collect data. Fortunately, the cost of RFID scanning is low, while passive RFID tags are also cheap. The big issue is the tags secure placement on or inside the envelope in a confidential way with an additional label, whereas the efficiency of this solution can be low. Hence, the label should act as an essential element of the letter, for example as prepaid envelope or the postal stamp dedicated for cross-border mail. Additionally, passive tags need to have relatively large antennas to achieve the appropriate radio coverage. For example, a typical passive RFID tag (complies with ISO 18000-6c) for coverage above 5 m needs antenna as big as 100×25 mm. Such a large component is difficult to hide. Another problem is that some senders may try to damage the tag. Embedding the tag on a postage stamp can solve this, but then it remains to solve the tag resistance to damage by timestamp hammer.
Neither of the mentioned solutions ensures end-to-end measurements, forcing an addition of a correction factor related to delay of collection and delivery. Delay of collection is relatively easy to estimate based on detailed records from domestic transit time. The estimation of delay of delivery is possible only with test letters with RFID tags.
Generally, all the proposed solutions generate significant costs for operators and do not cover all letters. The mentioned approaches do not ensure end-to-end measurements which means that all of them need adding a correction figure related to delay of collection and delivery. Hence, none of them seem to be ideal.

5. Conclusion

Postal items with correspondence are increasingly often being replaced by electronic communication, i.e. voice transmission or e-mail, thus decreasing the letters volume. Therefore, the cost of measurements of quality of service should also go down. However, the choice of the optimal solution will require further testing.

References

[1] Directive 97/67/EC of the European Parliament and of the Council of 15 December 1997 on common rules for the development of the internal market of Community postal services and the improvement of quality of service, OJ L 015 of 21 January 1998 with modifications (Directive 2002/39/EC and Directive 2008/6/EC).

[2] EN 13850:2002, "Postal services – Quality of service – Measurement of the transit time of end-to-end services for single piece priority mail and first class mail", 25 July 2002.

[3] R. Kobus, "Zastosowanie norm pocztowych", *Telekomunikacja i Techniki Informacyjne*, no. 1-2, pp. 67–81, 2012 (in Polish).

[4] International Post Corporation [Online]. Available: https://www.ipc.be/en/operational-services/quality%20excellence/unex

[5] R. Kobus, "Zastosowanie RFID do lokalizacji przesyłek pocztowych", *Zeszyty Naukowe Uniwersytetu Szczecińskiego nr 598, Ekonomiczne problemy usług nr 58*, pp. 657–664, 2010 (in Polish).

[6] EN 13850:2002+A1, "Postal service – Quality of service – Measurement of the transit time of end-to end-services for single piece priority mail and first class mail", March 2007.

[7] EN 13850:2012, "Postal Services – Quality of Services – Measurement of the transit time of end-to-end services for single piece priority mail and first class mail", 27 October 2012.

Ryszard Kobus received his B.Sc. and M.Sc. degrees from the Faculty of Electronics of the Warsaw University of Technology in 1975. Mr. Kobus has been working at the National Institute of Telecommunications since 1975. He is a member of the Expert Technical Committee CEN/TC 331 specializing in postal services, and the deputy chairman of the Postal Service Committee PKN/TC 259. He is a co-author of many patented telecommunications solutions. His research interests include: telecommunications, measurements and evaluation of quality of telecommunications services, quality surveys, evaluation the quality of postal services, standardization.

E-mail: R.Kobus@itl.waw.pl
National Institute of Telecommunications
1 Szachowa St
04-894 Warsaw, Poland

Frank Raudszus received his M.Sc. in Social Sciences and his Ph.D. in Economics at the University of Osnabrück. He continued with post graduate studies of "European Integration" at the University of Saarbrücken. Since 1991 he has been working for the Federal Network Agency. He has been chair of the DIN German Working Committee "Postal Services" and since 2016 he has chaired the European Technical Committee CEN/TC 331 Postal Services. Since 2011 he has also been working at the UPU Standards Board.

E-mail: Frank.Raudszus@BNetzA.de
Bundesnetzagentur
Canisiusstraße 21
Postfach 80 01, 55003 Mainz, Germany

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.

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*)

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 4/2017