International Journal of Advances in Telecommunications Electrotechnics, Signals and Systems

a publication of the International Science and Engineering Society

Vol. 5, No. 2
2016

ISSN: 1805-5443

www.ijates.org

IJATES²

International Journal of Advances in Telecommunications Electrotechnics, Signals and Systems

a publication of the International Science and Engineering Society

Vol. 5, No. 2, 2016 ISSN: 1805-5443

Editor-in-Chief

Jaroslav Koton, Brno University of Technology, Czech Republic

Co-Editors

Ondrej Krajsa, Brno University of Technology, Czech Republic
Norbert Herencsar, Brno University of Technology, Czech Republic

Editorial Board

Oguzhan Cicekoglu, Bogazici University, Turkey
Sergey Ryvkin, Trapeznikov Institute of Control Sciences Russian Academy of Sciences, Russian Federation
Hongyi Li, Bohai University, China
Emilia Daniela Bordencea, TU Cluj-Napoca, Romania
Albert Abilov, Izhevsk State Technical University, Russian Federation
Joze Guna, University of Ljubljana, Slovenia
Jaroslav Koton, Brno University of Technology, Czech Republic
Ondrej Krajsa, Brno University of Technology, Czech Republic
Danilo Pelusi, University of Teramo, Italy

Aims and Scope

The International Journal of Advances in Telecommunications, Electronics, Signals and Systems (IJATES²) is an all-electronic international scientific journal with the aim to bring the most recent and unpublished research and development results in the area of electronics to the scientific and technical societies, and is supported by the ISES (International Science and Engineering Society, o.s.). The journal's scope covers all the aspects of telecommunication, signal processing, theory and design of circuits and systems for electronics.

The IJATES² is ready to publish experimental and theoretical full papers and letters submitted by prospective authors. Paper submitted for publication must be written in English and must follow a prescribed format. All papers are subjected to a critical peer-review prior to publication.

The IJATES² is an open access journal which means that all content is freely available without charge to the user or his/her institution. Users are allowed to read, download, copy, distribute, print, search, or link to the full texts of the articles in this journal without asking prior permission from the publisher or the author. This journal provides immediate open access to its content on the principle that making research freely available to the public supports a greater global exchange of knowledge.

www.ijates.org

Copyright © 2012-2016, by ISES, o.s.
All the copyright of the present journal belongs to the International Science and Engineering Society, o.s.

CONTENTS

Vol. 5, No. 2, 2016 ISSN: 1805-5443

The role of the complex textural microstructure co-occurrence matrices, based on Laws' features, in the characterization and recognition of some pathological structures, from ultrasound images

Delia Mitrea, Sergiu Nedevschi, Mihail Abrudean, Radu Chifor and Radu Badea

***Abstract*—The non-invasive diagnosis, based on ultrasound images, is a challenge in nowadays research. We develop computerized, texture-based methods, for automatic and computer assisted diagnosis, using the information obtained from ultrasound images. In this work, we defined the co-occurrence matrix of complex textural microstructures determined by using the Laws' convolution filters and we experimented it in order to perform the characterization and recognition of some important anatomical and pathological structures, within ultrasound images. These structures were the colorectal tumors and the gingival sulcus, the properties of the latter being important concerning the diagnosis and monitoring of the periodontal disease. We determined the textural model of these structures, using the classical and the newly defined textural features. For the automatic recognition, we used powerful classifiers, such as the Multilayer Perceptron, the Support-Vector Machines, decision-trees based classifiers such as Random Forest and C4.5, respectively AdaBoost in combination with the C4.5 algorithm.**

***Keywords*—Complex Textural Microstructure Co-occurrence Matrix (CTMCM), classification performance, non-invasive diagnosis, texture, ultrasound images.**

I. Introduction

As they frequently affect the population of the developed countries and constitute a lethal disease, the colorectal tumors represent an important issue nowadays. The most reliable methods for diagnosis, the biopsy, the colonoscopy and the endoscopy are invasive or dangerous. The ultrasonography is a non-invasive method that is also inexpensive, safe, having the possibility of repeatability. In ultrasound images, the colorectal tumors have the characteristics of heterogeneity, bowel wall thickening and increased vascularity [1]. An eloquent example of a colorectal tumor is depicted in Fig. 1(a). We developed computerized, texture-based methods, for the non-invasive diagnosis of the colorectal tumors, based on ultrasound images. The Inflammatory Bowel Diseases (IBD) constitute chronic affections that frequently affect the population in the modern era, consisting mainly of the bowel layer inflammation, resembling the colorectal tumors in many situations, in ultrasound images [2]. For this reason, we compared the colorectal tumors with the IBD in our work.

The periodontal disease (periodontitis) is extremely widespread in the population nowadays. Gingival bleeding is highly prevalent among adults, and advanced stages of the periodontal disease affect 10% to 15% of them [3]. The size of the gingival sulcus (the possible space between the tooth and the gum, lined by sulcular epithelium) is an important property to be assessed in order to monitor the evolution of the periodontal disease, as the gingival sulcus becomes larger with the disease evolution). In ultrasound images, the gingival sulcus has a hypoechogenic, homogeneous aspect, an eloquent example being illustrated in Fig. 1(b). The delimitation of the gingival sulcus from the neighboring regions (teeth and gingival tissue) is difficult to be performed, within ultrasound images, by the human eye or by employing usual image analysis methods. We aim to find appropriate features in order to characterize the visual aspect of the gingival sulcus within high-frequency ultrasound images, this being a preliminary step for performing an accurate segmentation of this structure.

Texture is an important property of the body tissues, able to provide subtle information concerning both the tumor structure and the gingival sulcus aspect within ultrasound images [4]. Texture-based methods, in combination with classifiers, were widely used in the scientific research nowadays [5], [6], [7], [8], [9]. In our previous research, we defined the textural model of some important affections, consisting of the most relevant textural features for their characterization and of the specific values that corresponded to the relevant textural features: arithmetic mean, standard deviation, probability distribution [10]. In this work, we defined and implemented the complex textural microstructure co-occurrence matrix (CTMCM) of order two and three. The third order CTMCM was computed at a single resolution, as well as in a multiresolution manner. We assessed the role of the CTMCM matrix upon the improvement of the textural model of the colorectal tumors, respectively in order to accurately characterize the gingival sulcus within high-frequency ultrasound images.

The paper is organized as follows: first, the state of the art

Manuscript received on November 22, 2015, revised February 29, 2016.

Delia Mitrea and Sergiu Nedevschi are with the Technical University of Cluj-Napoca, Computer Science Department, Baritiu Str., No. 26-28, 400027, Cluj-Napoca, Romania, (corresponding author e-mail: Delia.Mitrea@cs.utcluj.ro).

Mihail Abrudean is with the Technical University of Cluj-Napoca, Automation Department, Baritiu Str., No. 26-28, 400027, Cluj-Napoca, Romania.

Radu Badea is with the Medical Imaging Department, Iuliu Hatieganu University of Medicine and Pharmacy, V. Babes Str., No. 8, Cluj-Napoca, Romania.

Radu Chifor is with the Department of Preventive Dentistry, Iuliu Hatieganu University of Medicine and Pharmacy, V. Babes Str., No. 15, 400012 Cluj-Napoca, Romania.

10.11601/ijates.v5i2.144

is provided in the next section; then, we define the CTMCM matrices and we describe the feature selection and classification methods implemented in this work. We describe the experiments performed in order to assess the role of the CTMCM in the automatic diagnosis of the colorectal tumors, respectively in order to perform the characterization and recognition of the gingival sulcus within ultrasound images. At the end, we formulate the conclusions and provide the bibliographic references.

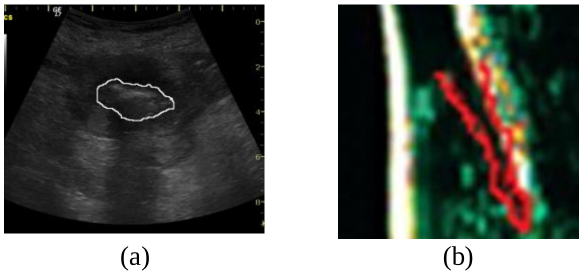


Fig.1. (a.) Colorectal tumor in ultrasound image (marked contour); (b.) Gingival sulcus in high-frequency ultrasound image (marked contour)

II. THE STATE OF THE ART

The most often implemented texture-analysis methods in the domain of pathological structure recognition from medical images were the second order GLCM matrix and the associated Haralick features, the Run-Length matrix [5], the fractals [6], the Wavelet [7] and Gabor transforms [8], used in combination with the k-nn classifiers, Bayesian classifiers [8], Artificial Neural Networks (ANN), Fisher Linear Discriminants, or Support Vector Machines (SVM) [5]. The resulting accuracy of these methods was around 90%. Concerning the automatic diagnosis of the colorectal cancer, in [9] the authors employed the Grey Level Co-occurrence Matrix (GLCM) in combination with morphologic features referring to shape and orientation, in order to distinguish the malignant and bening tissues in the case of the patients affected by colorectal cancer. These features were computed based on biopsy slides. The maximum obtained accuracy in this case was about 90% [9]. The generalized superior order co-occurrence matrices based on grey levels or edge orientations were defined in [10], [11]. A Local Binary Pattern (LBP) Co-occurrence Matrix of order two was defined in [12] and assessed on various datasets from the Brodatz texture collection, the resulted average accuracy being 94.30%. Also, a "Texture and Texture Orientation Co-occurrence Matrix" based on the computation of edge orientations, as well as on texton detection using specific filters, was defined and experimented in [13], yielding an average accuracy of 95.5%. There not exist significant approaches referring to co-occurrence matrices of the textural microstructures, determined by using the Laws' convolution filters. In [14], we defined the simple textural microstructure co-occurrence matrix, computed after the application of each Laws' filter. In this work, we study the role that the Complex Textural Microstructure Co-occurrence Matrix (CTMCM) has in the characterization and automatic diagnosis of some important structures within ultrasound images: the colorectal tumors, respectively the gingival sulcus, as it appears in the case of the periodontal disease.

III. THE PROPOSED METHODS

In order to assess the role of the CTMCM matrices in the recognition of the colorectal tumors, we *first applied the newly defined texture* analysis *methods*, then we performed *feature selection using specific techniques* and, *at the end*, we applied *supervised classifiers* in order to evaluate the *classification performance*. The techniques corresponding to this methodology are detailed below.

A. The newly defined texture analysis methods

The Complex Textural Microstructure Co-occurrence Matrix (CTMCM) was determined through the methodology described below, consisting of the following steps: (1) First, we associated feature vectors to the pixels in the region of interest, consisting of the results obtained after applying the 2D Laws' convolution filters for detecting levels, edges, spots, waves, ripples and also combined microstructures (L5L5, E5E5, S5S5, W5W5, R5R5, S5R5, R5S5) [15]. (2) Then, we applied an improved *k*-means clustering method, in the following manner: we started from a minimum number of centers (*k*=50); this number was increased by splitting the corresponding centers; a center was split in two other centers, if the standard deviation of the items within the corresponding class (cluster) overpassed the threshold equal with 3/4 of the average standard deviation of all the existing classes. The newly resulted centers were computed as being 1/2 of the old center, respectively 3/2 of the old center. (3) All the labels of the pixels from the ROI were re-assigned after splitting the old centers. The condition for the algorithm to finish was the convergence, the maximum number of centers being established to 200. The solution of the algorithm (the optimal solution) corresponded to the minimum value of WCSS (Within Cluster Sum of Squared Errors) [3]. Thus, the definition of the Complex Textural Microstructure Co-occurrence Matrix – CTMCM is provided in (1):

$$\begin{aligned}
&C_D(t_1,t_2,..,t_n) =\#\{((x_1,y_1),(x_2,y_2),(x_3,y_3),..,(x_n,y_n)):\\
&A(x_1,y_1)=t_1, A(x_2,y_2)=t_2,..,A(x_n,y_n)=t_n,\\
&|x_2-x_1|=|\vec{d}x_1|,\ x_3-x_1|=|\vec{d}x_2|,..,|x_n-x_1|=|\vec{d}x_{n-1}|,\\
&|y_2-y_1|=|\vec{d}y_1|,|y_3-y_1|=|\vec{d}y_2|,..,|y_n-y_1|=|\vec{d}y_{n-1}|,\\
&\operatorname{sgn}((x_2-x_1)(y_2-y_1))=\operatorname{sgn}(\vec{d}x_1\cdot\vec{d}y_1),..\\
&,\operatorname{sgn}((x_n-x_1)(y_n-y_1))=\operatorname{sgn}(\vec{d}x_{n-1}\cdot\vec{d}y_{n-1})\}
\end{aligned} \tag{1}$$

In (1), *#S* is the number of elements of the set *S* and

$$\vec{d}=((\vec{d}x_1,\vec{d}y_1),(\vec{d}x_2,\vec{d}y_2),..,(\vec{d}x_{n-1},\vec{d}y_{n-1})) \tag{2}$$

are the displacement vectors. „*A*" stands for the attribute associated to each pixel, while t_1, t_2,..., t_n are the values of the textons (cluster labels) obtained after the application of the improved *k*-means clustering algorithm. We computed the CTMCM matrix of order two and three and we determined the corresponding Haralick parameters, in a similar way as described in [10], [11]. The CTMCM matrix was determined in two situations: a) when taking into consideration all the Laws' convolution filters; b) when considering only the selected textural microstructures, corresponding to the S5S5, R5R5, S5R5 and R5S5 convolution filters, as the spot and ripple frequency provided the best results in our previous work [10]. We computed the CTMCM matrix of order two and three. For the CTMCM of order two, the following directions were considered: 0°, 90°, 180°, and 270°. For the CTMCM of order three, the current pixel was considered in the central position and together with the other two pixels they were either collinear, or formed a right angle triangle (the current pixel being in the position of the right angle). We considered the following orientations for the two displacement vectors: (0°, 180°), (90°, 270°), (45°, 225°), (135°, 315°) for the case of collinear pixels; (0°, 90°), (90°, 180°), (180°, 270°), (0°, 270°), (45°, 135°), (135°, 225°), (225°, 315°), (45°, 315°), for the right angle triangle case. The displacement vectors had the absolute value 2, in both cases. We determined the CTMCM matrices for all the considered direction combinations of the displacement vectors, the final features resulting as an average between the Haralick features of the individual matrices. We also determined *the third order CTMCM matrix at multiple resolutions*, obtaining the *Multiresolution Complex Textural Microstructure Co-occurrence Matrix (MCTMCM)* in the following manner: (1) first, we applied the Laws' convolution filters; (2) then, we applied the improved *k*-means clustering algorithm; (3) we applied the Haar Wavelet transform recursively, twice, on the image resulted at the previous step; (4) we computed the third order CTMCM matrix on each component, at two resolution levels. In this situation, we took into account all the Laws' convolution filters (at step (1)). Besides the Haralick features derived from the CTMCM matrices, we also considered, in our experiments, the following textural features: the Haralick parameters of the second and third order GLCM, respectively of the second and third order EOCM, edge and gradient based features, the autocorrelation index, fractal-based textural features (the Hurst index), the frequency of the textural microstructures, the entropy determined at two resolution levels, after applying the Wavelet transform [4].

B. The selection of the relevant features

In order to select the relevant textural features, we applied specific methods, which yielded the best results in our experiments, these being Correlation based Feature Selection (CFS) [16] combined with genetic search, which assigned a merit to each group of features with respect to the class [16] respectively the Gain Ratio Attribute Evaluation [17], combined with the Ranker method [17]. The final relevance index for each feature was obtained by performing the arithmetic mean between the individual relevance measures provided by each method.

C. Classification performance evaluation

For establishing the role of the CTMCM features in the context of the colorectal tumors' automatic diagnosis, respectively of the gingival sulcus characterization and recognition, we used the classifiers of Multilayer Perceptron (MLP) [18], of Support Vector Machines (SVM) [18], as well as the decision trees based algorithm Random Forest (RF) [13]. For the same purpose, we also implemented the AdaBoost meta-classifier, in conjunction with the C4.5 algorithm [18], this combination being well known for its performance. In order to assess the classification process, we used the recognition rate (classification accuracy), the sensitivity (TP rate), the specificity (TN rate) and the area under ROC (AUC) [18]. The strategy of cross-validation with 5 folds was adopted in this context.

IV. EXPERIMENTS AND DISCUSSIONS

During the experiments, we used 65 cases of colorectal tumors, respectively 65 cases of Inflammatory Bowel Diseases (IBD). The corresponding images were acquired using a Logiq 7 ultrasound machine at the same settings: 5.5 MHz frequency, gain of 78, respectively depth of 16 cm. Also, 50 high frequency ultrasound images representing the gingival sulcus and the surrounding tissues, belonging to 50 patients affected by periodontal disease, were taken into account for the experimental set. These images were acquired by a DermaScan C Cortex Technology®, Denmark device, at a 20 MHz frequency. The textural features were determined, after the conversion to grey-scale, within the regions of interest selected by the user inside the considered pathological structures, using our modules, implemented in Visual C++. The methods for the selection of the relevant textural features, respectively the classifiers, were implemented using the Weka 3.6 library. Concerning the classification methods, the John's Platt Sequential Minimal Optimization Algorithm (SMO) [17] of Weka 3.6, was applied for implementing the SVM method, with a polynomial kernel, the input data being also normalized. For the classifier of Multilayer Perceptron (MLP), we considered the specific method of Weka 3.6 (MultilayerPerceptron), containing, in the single hidden layer, a number of nodes equal with *a* = (*number_of_features* + *number_of_classes*)/2, the learning rate being tuned 0.2, respectively the momentum *α* being 0.8. The Random Forest (RF) classification method, with 10 trees, was implemented as well. The AdaBoostM1 meta-classifier of Weka was also employed using the J48 method (the equivalent of the C4.5 algorithm) as a basic learner. For classification performance evaluation, cross-validation with 5 folds was applied [17].

A. The role of the CTMCM matrix in the recognition of the pathological structures

1) Colorectal tumors/IBD differentiation

- *The case when all the Laws' features were considered*

The set of the relevant textural features for the characterization of the colorectal tumors, obtained by using the methods mentioned in Chapter III, are depicted in (3). Here, we notice the presence of the second order CTMCM features homogeneity and contrast, respectively of the third order CTMCM contrast, expressing the heterogeneity of the malignant tumors relatively to the IBD case, respectively the complex structure in grey levels of the colorectal tumors.

{GLCM_variance, Autocorrelation_index,
Wavelet_Entropy3, GLCM5_Entropy, GLCM5_variance,
EOCM3_Homogeneity, EOCM3_Energy,
EOCM3_Entropy, EOCM3_Contrast,
EOCM3_Homogeneity, ***CTMCM_Homogeneity***,
CTMCM_Contrast, ***CTMCM3_Contrast***} (3)

The classification performance parameters obtained in the case of the comparison between the colorectal tumors and IBD are depicted in Table I. The maximum recognition rate, of 98.33%, was obtained in the case of the SVM classifier.

We notice the increased values of the specificity (TN rate) obtained for the first three classifiers (SVM, MLP and RF).

TABLE I. THE CLASSIFICATION PERFORMANCE PARAMETERS OBTAINED IN THE CASE WHEN THE ENTIRE SET OF THE LAWS' FEATURES WAS CONSIDERED, FOR THE DIFFERENTIATION BETWEEN THE COLORECTAL TUMORS AND IBD

	Rec. Rate	TP Rate	TN Rate	AUC
SVM	**98.33%**	**96.7%**	**98.5%**	**98.33%**
MLP	**96.66%**	**93.3%**	**98.33%**	**96.66%**
RF	**95%**	**93.3%**	**96.7%**	**95%**
AdaBoost + J48	**93.3%**	**96.7%**	**90%**	**93.3%**

We also compared the classification accuracies obtained in the case when using only the old textural feature set with those obtained in the case when using the entire set of textural features (including the CTMCM features). As it results from the next figure (Fig. 2), in the case when considering the entire set of Laws' features, the newly resulted recognition rates overpassed the old recognition rates, in all cases.

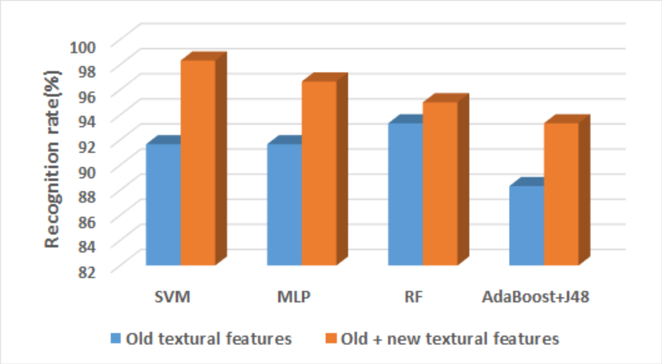


Fig. 2. The comparison between the recognition rates resulted for the old and new textural feature sets, when all the Laws' features were considered, in the case of the differentiation between colorectal tumors and IBD

- *The case when only the selected Laws' features were considered*

The relevant textural features obtained in this case are provided in (4). We notice the presence of the CTMCM based homogeneity, energy and contrast, as well as of the third order CTMCM based correlation and contrast, denoting the heterogeneity and complex structure of the tumoral tissue, respectively differences in granularity between the tumoral tissue and the non-tumoral one (through the third order CTMCM correlation).

{GLCM_variance, Autocorrelation_index,
Directional_grad_magnitude, Directional_grad_variance,
GLCM5_Entropy, GLCM5_Variance, EOCM3_
Homogeneity, EOCM3_Energy, EOCM3_Entropy,
EOCM3_Contrast, EOCM3_Variance,
GLCM3_Homogeneity, GLCM3_Entropy,
GLCM3_Correlation, ***CTMCM_Homogeneity***,
CTMCM_Energy, ***CTMCM_Contrast***,
CTMCM3_Correlation, ***CTMCM3_Contrast}*** (4)

The values of the classification performance parameters obtained in this case are depicted in Table II. As it results from Table II, the maximum recognition rate obtained in this situation, of 97.5%, corresponded to the RF classifier, being slightly lower than the maximum accuracy obtained in the previous case. We notice the increased values of the specificity (TN rate) in this case as well.

TABLE II. THE CLASSIFICATION PERFORMANCE PARAMETERS OBTAINED IN THE CASE WHEN THE SELECTED LAWS' FEATURES WERE CONSIDERED, FOR THE DIFFERENTIATION BETWEEN THE COLORECTAL TUMORS AND IBD

	Recogn. Rate	TP Rate	TN Rate	AUC
SVM	**97.33%**	**94%**	**98.7%**	**97.3%**
MLP	**96.66%**	**91.7%**	**98.7%**	**98.1%**
RF	**97.5%**	**95.3%**	**98.7%**	**99.8%**
AdaBoost + J48	**91.81%**	**86.3%**	**96.3%**	**97.4%**

When performning the comparison of the entire set of the textural features (containing both the old and the newly defined features) with the old textural feature set, we noticed an increase in classification accuracy for the first mentioned set, in most of the situations. This result is illustrated in the figure below (Fig. 3).

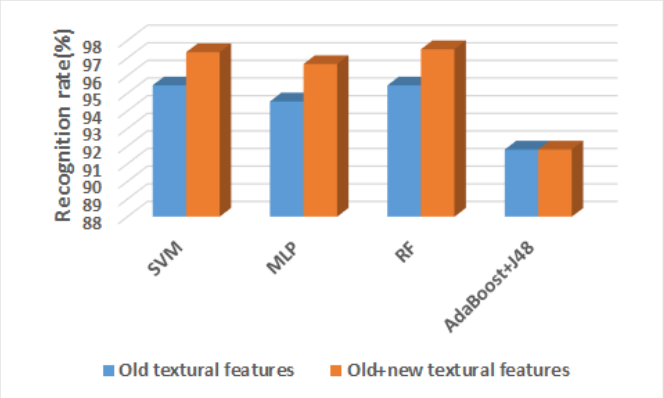


Fig. 3. The comparison between the recognition rates resulted for the old and new textural feature sets, when only the selected Laws' features were considered, in the case of differentiation between colorectal tumors and IBD

2) *The differentiation between the gingival sulcus and the neighboring areas*

- *The case when all the Laws' features were considered*

The set of the relevant textural features for the differentiation between the gingival sulcus and the surrounding areas is provided in (5):

{ GLCM_Energy, Autocorrelation_index,
Edge,orientation_variability,
Wavelet_Entropy7_lh, Wavelet_Entropy7_hl,
Wavelet_Entropy7_hh, Wavelet_Entropy8_hh,
Laws_level_mean, Laws_level_frequency,
Laws_ripple_mean, Laws_ripple_frequency,
CTCM_Homogeneity, ***CTMCM_Entropy***,
CTMCM_Correlation, ***CTMCM3_***
Homogeneity, ***CTMCM3_Energy***,
CTMCM3_Entropy, ***CTMCM3_Correlation***} } (5)

We notice, in (5), the presence of the second order CTMCM features, as well as of the third order CTMCM features. The homogeneity, energy and entropy, derived from the second and third order CTMCM matrices, stand for the more homogeneous and hypoechogenic nature of the gingival sulcus region, in comparison with the sourrounding areas. The correlation computed from the second, respectively from the third order CTMCM, together with the autocorrelation index, denote differences in granularity between the gingival sulcus and the neighboring regions.

The values of the classification performance parameters, obtained after the selection of the relevant textural features, in the case of separation between the gingival sulcus and the surrounding regions, are provided in Table III. The maximum recognition rate, of 93.05%, the maximum specificity, of 91.7%, as well as the maximum AUC, of 97.4%, resulted in the case of the RF classifier. The maximum specificity, of 94.4%, resulted in both cases of the RF and MLP classifiers.

TABLE III. THE CLASSIFICATION PERFORMANCE PARAMETERS OBTAINED WHEN ALL THE LAWS' FEATURES WERE CONSIDERED, IN THE CASE OF GINGIVAL SULCUS RECOGNITION

	Recogn. Rate	TP Rate	TN Rate	AUC
SVM	**86.11%**	**83.3%**	**88.9%**	**94.7%**
MLP	**91.66%**	**94.4%**	**88.9%**	**92.1%**
RF	**93.05%**	**94.4%**	**91.7%**	**97.4%**
AdaBoost + J48	**88.88%**	**88.9%**	**88.9%**	**91.6%**

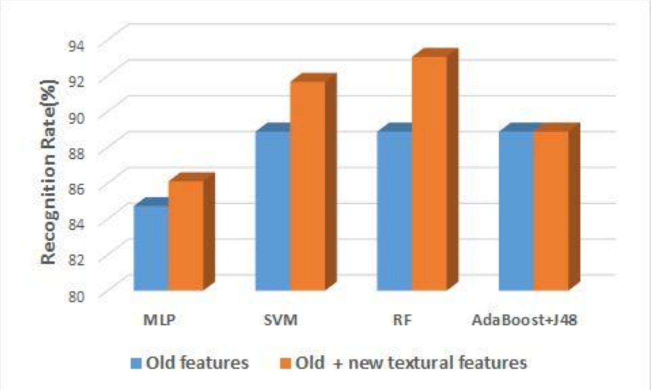


Fig. 4. The comparison between the recognition rates resulted for the old and new textural feature sets, when all the Laws' features were considered, in the case of the differentiation between the gingival sulcus and the surrounding areas

The discrimination power due to the previously existing textural features was compared with that which was due to the set formed by the old textural features and by the newly defined textural features. This comparison is illustrated in the fourth figure. From Fig. 4, we notice that the recognition rate which is due to the newly defined textural features is superior to that provided only by the old textural features in most of the situations, excepting the case of the AdaBoost meta-classifier combined with the J48 basic learner, when the two computed accuracies are equal.

- *The case when only the selected Laws' features were considered.*

The set of the relevant textural features selected in this case is illustrated in (6). We notice the presence of the textural features derived from the CTMCM matrix: the entropy computed from the second and third order CTMCM, respectively the homogeneity computed from the third order CTMCM, stand for the differences in heterogeneity between the structure of the gingival sulcus and the surrounding areas. The contrast derived from the third order CTMCM matrix denote the less complex structure of the gingival sulcus, respectively the more complex structure of the surrounding areas (gum and teeth). We also remark the first order statistics referring to the arithmetic mean and frequency of the simple textural microstructures detected by applying the Laws' convolution filters: levels, edges, spots, waves and ripples.

{ GLCM_Energy, GLCM_Entropy,
Autocorrelation_index,
Edge_orientation_variability,Wavelet_Entropy7_ll,
Wavelet_Entropy8_ll,Wavelet_Entropy8_lh,
Laws_level_mean, Laws_edge_mean, (6)
Laws_spot_mean, Laws_wave_mean,
Laws_ripple_mean, Laws_ripple_frequency,
CTMCM_Entropy, ***CTMCM3_Homogeneity***,
CTMCM3_Entropy, ***CTMCM3_Contrast***}

The classification performance parameters obtained in this situation, after the selection of the relevant textural features in the case of the differentiation between the gingival sulcus and the neighboring tissues, are illustrated in Table IV.

TABLE IV. THE CLASSIFICATION PERFORMANCE PARAMETERS OBTAINED IN THE CASE WHEN ONLY THE SELECTED LAWS' FEATURES WERE CONSIDERED, IN THE CASE OF GINGIVAL SULCUS RECOGNITION

	Recogn. Rate	TP Rate	TN Rate	AUC
SVM	**90.32%**	**91.7%**	**88.9%**	**97%**
MLP	**90.27%**	**97.2%**	**83.3%**	**88.7%**
RF	**91.75%**	**97.2%**	**86.1%**	**96.4%**
AdaBoost + J48	**91.66%**	**94.4%**	**88.9%**	**89%**

Fig. 5 illustrates the comparison between the recognition rates obtained when considering only the old textural feature set, respectively the set formed by the old and new textural features. As it results from Fig.5, there is always an increase in accuracy for the feature set that includes the newly defined textural features. The best accuracy, of 91.75%, resulted in the case of the RF classifier, being smaller than the maximum accuracy (93.05%) resulted in the case when all the Laws' features were considered in order to define the CTMCM matrix The highest sensitivity, of 97.2% resulted in both cases of MLP and RF classifiers, the highest specificity, of 88.9% resulted in the cases of the SVM classifier, respectively of the combination between the AdaBoost meta-classifier and the J48 learner, while the highest recognition rate, of 97%, resulted in the case of the SVM classifier. We notice that all the values for the recognition rate were above 90% in this situation.

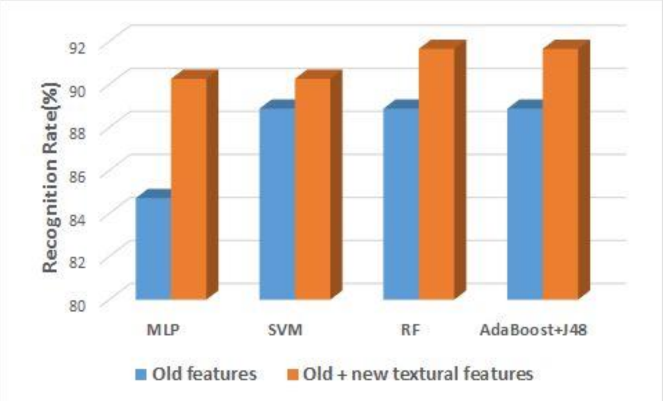


Fig. 5. The comparison between the recognition rates resulted for the old and new textural feature sets, when the selected Laws' features were considered, in the case of the differentiation between the gingival sulcus and the surrounding areas

B. The role of the multiresolution CTMCM (MCTMCM) in the recognition of the pathlological structures

1) Colorectal tumors/IBD differentiation

The set of the relevant textural features obtained in this case is illustrated in (7). We can remark that also the third order MCTMCM based features: variance, contrast and correlation are included in this set. These features characterize the heterogeneity and the complex grey level structure of the malignant tumors. We can also notice that the relevant MCTMCM features resulted at both resolution levels: at the first level, on the component obtained by applying the combination of the low-pass filters and at the second level, on most of the components.

{Wavelet_Entropy2, Wavelet_Entropy6_ll, Dir_grad_variability, GLCM5_ Entropy, EOCM3_ Coreelation, GLCM3_Energy, ***MCTMCM3 _Variance2, MCTMCM3_ Contrast3_ll, MCTMCM3_Correlation2 _ll, MCTMCM3_ Homogeneity3_lh, MCTMCM3_ Contrast3_hl, MCT MCM3_Homogeneity4_ hh***} (7)

TABLE V. THE CLASSIFICATION PERFORMANCE PARAMETERS OBTAINED IN THE CASE WHEN THE MCTMCM FEATURES WERE CONSIDERED, WHEN DIFFERENTIATING BETWEEN COLORECTAL TUMORS AND IBD

	Recogn. Rate	**TP Rate**	**TN Rate**	**AUC**
SVM	**96.36%**	**96.4%**	**96.4%**	**96.4%**
MLP	**95.45%**	**94.5%**	**96.4%**	**99.4%**
RF	**95.45%**	**92.7%**	**98.2%**	**98.8%**
AdaBoost + J48	**93.63%**	**94.5%**	**92.7%**	**94.1%**

Concerning the classification performance resulted for the relevant feature set depicted in (7), the maximum recognition rate, of 96.36%, was obtained in the case of the SVM classifier, as it results from Table V. We can also notice the increased values of the specificity obtained in this case, as well as the high AUC.

At the end, we compared the classification accuracies obtained for two datasets: for the dataset containing only the old textural features, respectively for that containing both the old and new textural features. The improvement due to the entire feature set is obvious for three classifiers, while for the RF classifier, only a slight improvement was noticed, as it results from the next figure (Fig. 6).

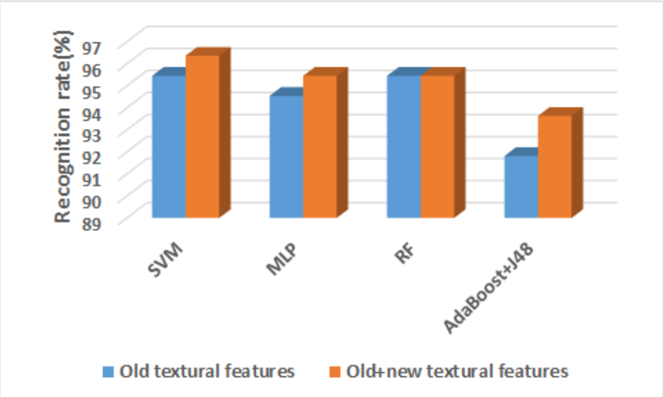


Fig. 6. The comparison between the recognition rates resulted for the old and new textural feature sets, when the MCTMCM features were considered, in the case of the Colorectal tumors/IBD differentiation

2) The differentiation between the gingival sulcus and the neighboring areas

The set of the relevant textural features for the differentiation between the gingival sulcus and the surrounding tissues, obtained when taking into account the multiresolution third order CTMCM attributes, is illustrated in (8). We notice that, in (8), there are multiple features derived from the MCTMCM matrix of order three. Thus, we remark the presence of the third order MCTMCM homogeneity, corresponding to most of the components on all the considered resolution levels, as well as of the third order MCTMCM entropy and energy, denoting the more heterogeneous structure of the tissues that surround the gingival sulcus; of the third order MCTMCM correlation, denoting differences in granularity, at multiple resolutions, between, the considered classes of tissues, of the third order MCTMCM contrast and variance, denoting the more increased structural complexity of the tissue classes that surround the gingival sulcus.

{GLCM_Energy, GLCM_Contrast, Wavelet_Entropy7_ll, Wavelet_Entropy7_hh Wavelet_Entropy8_hh, ***MCTMCM3_Correlation3, MCTMCM3_Variance4, MCTMCM3_Contrast2_ll, MCTMCM3_Homogeneity4_ll, MCTMCM3_ Energy1_lh, MCTMCM3_ Contrast3_lh, MCTMCM3_Variance2_lh, MCTMCM3_ Entropy2_hl, MCTMCM3_ Contrast1_hl, MCTMCM3_Contrast4_hl, MCTMCM3_Homo geneity2_hl, MCTMCM3_ Homogeneity3_hl, MCTMCM3_Homogeneity2_hh, MCTMCM3_ Homogeneity3_hh, MCTMCM3_Homogeneity4_hh*** } (8)

In Table VI, the values of the classification performance parameters, obtained when providing the set of the relevant textural features illustrated in (8), at the inputs of the considered classifiers, are depicted. The maximum recognition rate, of 92.85%, as well as the maximum AUC, of 96.5%, resulted in the case of the RF classifier. We also notice the increased value of the sensitivity (96.4%), obtained in the case of the MLP classifier.

TABLE VI. THE CLASSIFICATION PERFORMANCE PARAMETERS OBTAINED IN THE CASE WHEN THE MCTMCM FEATURES WERE CONSIDERED, WHEN DIFFERENTIATING BETWEEN THE GINGIVAL SULCUS AND THE NEIGHBORING AREAS

	Recogn. Rate	**TP Rate**	**TN Rate**	**AUC**
SVM	**90.28%**	**90.3%**	**90.3%**	**90.3%**
MLP	**91.07%**	**96.4%**	**85.7%**	**91.1%**
RF	**92.85%**	**92.9%**	**92.9%**	**96.5%**
AdaBoost + J48	**90.28%**	**85.7%**	**92.9%**	**91.7%**

The comparison between the classification accuracies resulted when considering only the old textural features, respectively when taking into account both the old and the newly defined textural features is illustrated in Fig.7. An accuracy increase for all the classifiers, corresponding to the case when also the newly defined textural features were taken into account, can be noticed in this figure.

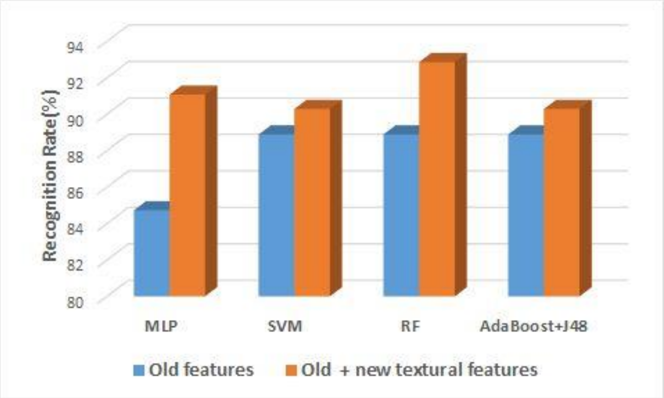


Fig.7. The comparison between the recognition rates resulted for the old and new textural feature sets, when the MCTMCM features were considered, in the case of the differentiation between the gingival sulcus and the surrounding tissues

V. Conclusions

The Complex Textural Microstructure Co-occurrence Matrix (CTMCM) generally led, in the case of the considered pathological structures, to a classification performance improvement, compared with the situation when using only the old textural features. The best results were obtained when all the Laws' features were considered, but satisfying results were provided also when taking into account the CTMCM matrix based only on the selected Laws' features, respectively the third order MCTMCM based features, especially in the case of the gingival sulcus characterization and recognition. The best obtained classification accuracy was 98.33% in the case of the recognition of the colo-rectal tumors and 93.05% in the case of the gingival sulcus recognition. Thus, in the first case, the result was better than the already obtained results in the domain, while in the second case the result was comparable with the state of the art results.

More extended datasets will be considered in our future work, in order to further validate the CTMCM and MCTMCM methods. Other types of multiresolution textural features will be considered as well, such as those based on the Gabor transform [4].

References

[1] D. Martínez-Ares, I. Martín-Granizo Barrenechea, J. Souto-Ruzo, J. Yáñez López, A. Pallarés Peral and J. L. Vázquez-Iglesias, "The value of abdominal ultrasound in the diagnosis of colon cancer", *Revista Espanola des Enfermedades Digestiva*, vol. 97, no.12, pp.877-886, 2005.

[2] L. Ruess "Inflammatory bowel disease in children and young adults: correlation of sonographic and clinical parameters during treatment", *American Journal of Roentgenology*, vol.175, no.1, pp.79-84, 2000.

[3] A. Nanci, D. Bosshardt, "Structure of periodontal tissues in health and disease", *Periodontology 2000*, vol.40, no.1, pp. 11–28, 2006.

[4] A. Meyer-Base, *Pattern recognition for medical imaging*, Elsevier, 2009.

[5] H.Sujana, S.Swarnamani, "Application of Artificial Neural Networks for the classification of liver lesions by texture parameters", *Ultrasound in Medicine & Bio*logy, vol. 22, no. 9, pp. 1177 –1181, 1996.

[6] T.Chikui, "Sonographic texture characterization of salivary gland tumors by fractal analysis", *Ultrasound in Medicine & Biology*, vol. 31, no. 10, pp.1297- 1304, 2005.

[7] H.Yoshida, D.Casalino, "Wavelet packet based texture analysis for diferentiation between benign and malignant liver tumors in ultrasound images*", Physics in Medicine & Biology*, vol. 48, pp. 3735-3753, 2003.

[8] A.Madabhushi, "Automated Detection of Prostatic Adenocarcinoma from High-Resolution Ex Vivo MRI", *IEEE Transactions on Medical Imaging*, pp.1611-1626, 2005.

[9] K. Masood, "Co-occurrence and morphological analysis for colon tissue biopsy classification", *Proceedings of the 4th International Workshop on Information Technology*, Pakistan, pp. 211-216, 2006.

[10] D. Mitrea, P. Mitrea, S. Nedevschi, R. Badea, "Abdominal tumor characterization and recognition using superior order cooccurrence matrices, based on ultrasound images", *Computational and Mathematical Methods in Medicine*, 17 pages, 2012.

[11] D. Mitrea, S. Nedevschi, M. Socaciu, R. Badea, "The Role of the Superior Order GLCM in the Characterization and Recognition of the Liver Tumors from Ultrasound Images", *Radioengineering*, vol. 21, no. 1, pp. 79-85, 2012.

[12] B. Sujatha, V. Kumar, P. Harini, "A new logical compact LBP Co-Occurrence Matrix for Texture Analysis", *International Journal of Scientific & Engineering Research*, vol. 3, no. 2, pp. 1-5, 2012.

[13] B. Sujatha, C. Sekhar Reddi, "Texture classification using texton co-occurrence matrix derived from texture orientation", *International Journal on Soft Computing and Engineering*, vol. 2, no. 6, pp. 18-23, 2013.

[14] D. Mitrea, S. Nedevschi, R. Badea, "The Role of the Textural Microstructure Coocurrence Matrices in the Classification of the Abdominal Tumors", *10th IEEE ICCP 2014 Conference*, T.U.Cluj-Napoca, pp. 187-190, 2014.

[15] K. I. Laws, „Rapid texture identification", *SPIE Vol. 238, Image Processing for Missile Guidance*, pp.76-380, 1980.

[16] M. Hall, "Benchmarking attribute selection techniques for discrete class data mining", *IEEE Transactions On Knowledge and Data Engineering*, vol. 15, no. 3, pp. 1-16, 2003.

[17] Weka3, Data Mining Software in Java, 2014, Online: http://www.cs.waikato.ac.nz/ml/weka.

[18] R. Duda, *Pattern Classification*, Wiley Interscience, 2003.

Design, Implementation and Testing of a Tiny Multi-Threaded DNS64 Server

Gábor Lencse and András Gábor Soós

***Abstract*—DNS64 is going to be an important service (together with NAT64) in the upcoming years of the IPv6 transition enabling the clients having only IPv6 addresses to reach the servers having only IPv4 addresses (the majority of the servers on the Internet today). This paper describes the design, implementation and functional testing of MTD64, a flexible, easy to use, multi-threaded DNS64 proxy published as a free software under the GPLv2 license. All the theoretical background is introduced including the DNS message format, the operation of the DNS64 plus NAT64 solution and the construction of the IPv4-embedded IPv6 addresses. Our design decisions are fully disclosed from the high level ones to the details. Implementation is introduced at high level only as the details can be found in the developer documentation. The most important parts of a through functional testing are included as well as the results of some basic performance comparison with BIND.**

***Keywords*—DNS, DNS64, domain names, IPv4, IPv6, IPv6 transition.**

I. Introduction

Due to the depletion of the public IPv4 address pool [1] the ISPs (Internet Service Providers) will not be able to assign public IPv4 addresses to their new clients. Reference [2] classifies the possible IPv4 address sharing mechanisms and discloses their tradeoffs. From among them, many Hungarian ISPs have chosen to give private IPv4 addresses to the clients and use CGN (Carrier Grade NAT). However, this solution limits the reachability of the clients from the outside world, and does not support their transition to IPv6, which one must happen once (sooner or later). In our opinion, the deployment of IPv6 is the forward looking solution for the shortage of public IPv4 address. The new clients will get IPv6 addresses only and they can communicate with the native IPv6 servers directly, but the majority of the Internet servers still use IPv4 only. The combination of a DNS64 [3] service and a NAT64 [4] gateways is a suitable solution which enables the IPv6 only clients to communicate with IPv4 only servers [5]. We agree with the authors of [2] that: “The only actual address sharing mechanism that really pushes forward the transition to IPv6 is Stateful NAT64 (Class 4). All other (classes of) mechanisms are more tolerant to IPv4.” Therefore we expect that (because NAT64 needs it) DNS64 will become a widespread used service during the upcoming phase of the IPv6 transition. To use this solution, a DNS64 server has to be set as the DNS server in the IPv6 only computers. When a client program (e.g. web browser) requests a domain name resolution for the domain name of a server which it wants to connect to, then the DNS64 server acts like a proxy: it uses the normal DNS system to find out the IP address. If the DNS64 server gets an IPv6 address from the DNS system then it simply returns the IPv6 address to the client. However, if it gets no IPv6 address but only IPv4 address (recall that it happens in the vast majority of the cases today) then it synthesizes a so called IPv4-embedded IPv6 address [6] and it returns the synthesized IPv6 address to the client. In this case, the communication of the IPv6 only client and the IPv4 only server will happen with the help of a NAT64 gateway. See more details later in this paper.

There are a number of free software [7] (also called open source [8]) DNS64 implementations, e.g. BIND, Unbound, PowerDNS or TOTD but even the smallest of them, TOTD has about 10,000 lines of source code (excluding the source of SWILL, its built-in web server) [9]. In this paper, we propose *MTD64*, a tiny *M*ulti-*T*hreaded *D*NS*64* server, which one is very small in code size (less than 1300 lines of source code) but it is still flexible and convenient. The aim of our work is to provide a simple DNS64 implementation which has clear and disclosed design decisions and well documented source code to give a chance for others to improve it by adding further functionalities or changing some of the used solutions to more efficient ones. The software is planned to be developed mainly by university students under the supervision of the first author of this paper, but our free software license allows anyone to join by making an own fork of the source code. At its current stage, MTD64 is not meant to be used as a DNS64 server in real-life networks, but it is rather meant to be a base point for further developments and to serve also as a testbed for comparison of the efficiency of different possible solutions (e.g. different caching policies). Our long term goal is to develop a production quality DNS64 server step by step.

The design decisions of MTD64 were originally disclosed in our conference paper [10], which one is now extended with high level details of implementation and with the documentation of testing including a thorough functional testing and a basic performance comparison with BIND.

The remainder of this paper is organized as follows. First, the theoretical background is introduced to the reader: the DNS message format, the operation of the DNS64+NAT64 solution and the construction of the IPv4-embedded IPv6 addresses are described. Second, our design decisions are presented from the high level ones to the details. Third, the implementation is described at high level including the source files and their roles as well as the operation of the program in a nutshell. Fourth, a detailed functional testing of our DNS64 implementation is done including a short performance testing, too. Fifth, our future plans are summarized. Finally, our conclusions are given.

Manuscript received November 6, 2015, revised March 14, 2016.

G. Lencse is with the Department of Networked Systems and Services, Budapest University of Technology and Economics, 2 Magyar tudósok körútja, H-1117 Budapest, Hungary (phone: +36-20-775-82-67; fax: +36-1-463-3263; e-mail: lencse@hit.bme.hu).

A. G. Soós was with Telenor Hungary, 1 Pannon út, H-2045 Törökbálint, Hungary (e-mail: soos.gabor.andras@gmail.com).

doi: 10.11601/ijates.v5i2.129

<table>
<tr><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th><th>15</th></tr>
<tr><td colspan="16">Transaction ID</td></tr>
<tr><td>QR</td><td colspan="4">OPCODE</td><td>AA</td><td>TC</td><td>RD</td><td>RA</td><td colspan="3">Z</td><td colspan="4">RCODE</td></tr>
<tr><td colspan="16">QDCOUNT</td></tr>
<tr><td colspan="16">ANCOUNT</td></tr>
<tr><td colspan="16">NSCOUNT</td></tr>
<tr><td colspan="16">ARCOUNT</td></tr>
</table>

Fig. 1. DNS message *Header* section format.

<table>
<tr><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th><th>15</th></tr>
<tr><td colspan="16">QNAME
(variable length)</td></tr>
<tr><td colspan="16">QTYPE</td></tr>
<tr><td colspan="16">QCLASS</td></tr>
</table>

Fig. 2. DNS message *Question* section format.

II. Theoretical Background

A. Format of DNS Messages

The DNS64 server has to work with various DNS messages: it must interpret, forward, prepare or synthesize them. Therefore we give a brief summary of the DNS message format [11].

DNS messages between a client and a server usually travel over UDP because both the requests and replies are usually short and sending them over UDP is much faster than establishing a TCP connection using the three-way handshake before the client-server communication and closing it at the end using the four-way handshake. If some of the messages happen to be lost then they can be resent.

1) Top level structure

A DNS message is built up by five sections: its *Header* section is always 12 bytes long and it is followed by four variable length sections (some of them may be empty): *Question*, *Answer*, *Authority*, *Additional*.

2) Header section format

The *Header* section can be further subdivided as shown in Fig. 1. The 16-bit *Transaction ID* field is used by the client to identify the answer of the server for different questions. It is generated by the requester (client) and it is copied by the server into the corresponding reply. The *QR* bit specifies whether this message is a query (0), or a response (1). The *OPCODE* field is used by the originator of the query to specify the kind of the query and it is copied by the server into the answer. Only the 0 value is of practical interest for us, it means standard query. The *AA* bit is valid only in responses and it signals if the answer is authoritative. The *TC* bit signals if the DNS message was truncated due to the limitations of the MTU of the transmission channel. The usage of the TC bit is clarified in section 9 of [12]. “The TC bit should not be set merely because some extra information could have been included, but there was insufficient room.” It also states that: “When a DNS client receives a reply with TC set, it should ignore that response, and query again, using a mechanism, such as a TCP connection, that will permit larger replies.”

→ *The DNS64 server program should not set the TC bit for leaving out some of the Additional RRs at the end of the message.*

The *RD* bit is used by the requester to ask recursive query. The *RA* bit is used by the server to signal if recursion is available. All four bits of the *Z* field must be set to 0 in all queries and responses (it is reserved for future use). The *RCODE* field of the responses specifies the error code: 0 value means no error. The *QDCOUNT* field specifies the number of entries in the *Question* section. In practice, clients send only one question in a DNS message. The *ANCOUNT*, *NSCOUNT* and *ARCOUNT* fields specify the number of resource records in the *Answer*, *Authority* and *Additional* sections, respectively.

<table>
<tr><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th><th>15</th></tr>
<tr><td colspan="16">NAME
(variable length)</td></tr>
<tr><td colspan="16">TYPE</td></tr>
<tr><td colspan="16">CLASS</td></tr>
<tr><td colspan="16">TTL
(32-bit)</td></tr>
<tr><td colspan="16">RDLENGTH</td></tr>
<tr><td colspan="16">RDATA
(variable length)</td></tr>
</table>

Fig. 3. DNS message *RR* format for *Answer*, *Authority*, *Additional* sections.

3) Question section format

The *Question* section contains *QDCOUNT* number of entries (usually 1). An entry follows the format shown in Fig 2. The variable length *QNAME* field contains the domain name using special encoding (see: Domain name encoding and message compression). The *QTYPE* filed specifies the *RR* (Resource Record) type by 16-bit long binary vales. Some examples are:

- *A* (0x01) – IPv4 ***A***ddress
- *AAAA* (0x1C) – IPv6 Address (***4*** times size of ***A***)
- *CNAME* (0x05) – ***C***anonical ***NAME*** (alias)
- *MX* (0x0F) – ***M***ail e***X***changer
- *NS* (0x02) – ***N***ame ***S***erver
- *PTR* (0x0C) – used for reverse mapping (***P***oin***T***e***R***).

The *QCLASS* field contains the 0x01 16-bit binary value for denoting the *IN* (Internet) class. The other theoretically possible values for *CH* (Chaos) or *HS* (Hesiod) are not used.

4) Resource record format

The *RR* (Resource Record) format – used in the *Answer*, *Authority* and *Additional* sections – is shown in Fig. 3. The first three fields correspond to that of the Question section. The 32-bit unsigned integer in the *TTL* (Time to Live) field specifies the time interval in seconds while the *RR* may be cached. The 16-bit unsigned integer in the *RDLENGTH* field gives (in octets) the length of the *RDATA* field, which contains the octets of the given resource (e.g. the 4 octets of the IPv4 address or the 16 octets of the IPv6 address).

5) Domain name encoding and message compression

The domain names stored in the *QNAME* or *NAME* fields follow special encoding. A domain name is built up by so called *label*s separated from each other by “.” characters. The labels must be no longer than 63 characters. When domain names are encoded in DNS messages, the first character gives the length of the first label and then the characters of the first label follow. After that, a character stands that specifies the length of the next label and the characters of the next label follow, etc. Finally, a zero character after the last label signals the end of the domain name. Fig. 4 illustrates the encoding of the domain name

5	w	h	a	l	e	3	h	i	t	3	b	m	e	2	h	u	0

Fig. 4. DNS encoding of the **whale.hit.bme.hu** domain name.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	OFFSET													

Fig. 5. The structure of a pointer.

3	w	w	w	C0	36

Fig. 6. Compressed encoding of the **www.hit.bme.hu** domain name using the fact that the domain name shown in Fig. 4 starts at offset 0x0030.

whale.hit.bme.hu.

The addition of *pointer*s to this encoding scheme makes possible an efficient compression if there are repetitions of entire domain names or label sequences at the end of the domain names in DNS messages. A pointer is a two octet sequence where the first two bits of the first octet are ones, see Fig. 5. Note that the length of a label is at most 63 octets, therefore the first two bits of the octet expressing its length are always zeros, thus a pointer can be easily distinguished from a label. The *OFFSET* field of the pointer specifies the offset of the pointed label sequence from the beginning of the DNS message. Let us demonstrate it with an example. If the domain name in Fig. 4 starts at offset 0x0030 in a DNS message then we can compress the **www.hit.bme.hu** domain name in the same DNS message as it is shown in Fig. 6. The beginning three **w** characters are encoded in the usual way and then follows the 0xC0 value. The "11" values of its first two bits show that this is a pointer and the octet is to be interpreted together with the next one. The value of the offset field is 0x0036, which points to the second label of the domain name in Fig. 4.

→ *The DNS64 server program must be able to handle correctly this encoding and compression scheme.* (See later its consequences: the server program must be able to decode the domain name for logging purposes and it must also be able to modify the pointer if the pointed *RR* is moved within the DNS message.)

B. Operation of the DNS64 + NAT64 Solution

The operation of the DNS64 + NAT64 solution is demonstrated in Fig. 7. It shows a scenario where an IPv6 only client communicates with an IPv4 only web server. The DNS64 server uses the 64:ff9b::/96 *NAT64 Well-Known Prefix* for generating *IPv4-embedded IPv6 address*es. A prerequisite for the proper operation is that packets towards the 64:ff9b::/96 network are routed to the NAT64 gateway (routing must be configured that way). Let us follow the steps:

1. The client asks its DNS server (which one is actually a DNS64 server) about the IPv6 address of the **www.hit.bme.hu** web server.
2. The DNS64 server asks the DNS system about the IPv6 address of **www.hit.bme.hu**.
3. No IPv6 address is returned.
4. The DNS64 server then asks the DNS system for the IPv4 address of **www.hit.bme.hu**.
5. The 152.66.148.44 IPv4 address is returned.
6. The DNS64 server synthesizes an *IPv4-embedded IPv6 address* by placing the 32 bits of the received 152.66.148.44 IPv4 address after the 64:ff9b::/96 prefix and sends the result back to the client.
7. The IPv6 only client sends a TCP SYN segment using the received 64:ff9b::9842:f82c IPv6 address and it arrives to the IPv6 interface of the NAT64 gateway (since the route towards the 64ff9b::/96 network is set so in all the routers along the path).
8. The NAT64 gateway constructs an IPv4 packet using the last 32 bits (0x9842f82c) of the destination IPv6 address as the destination IPv4 address (this is exactly 152.66.248.44), its own public IPv4 address (198.51.100.10) as the source IPv4 address and some other fields from the IPv6 packet plus the payload of the IPv6 packet. It also registers the connection into its connection tracking table (and replaces the source port number by a unique one if necessary). Finally it sends out the IPv4 packet to the IPv4 only server.
9. The server receives the TCP SYN segment and sends a SYN ACK reply back to the public IPv4 address of the NAT64 gateway.
10. The NAT64 gateway receives the IPv4 reply packet. It constructs an appropriate IPv6 packet

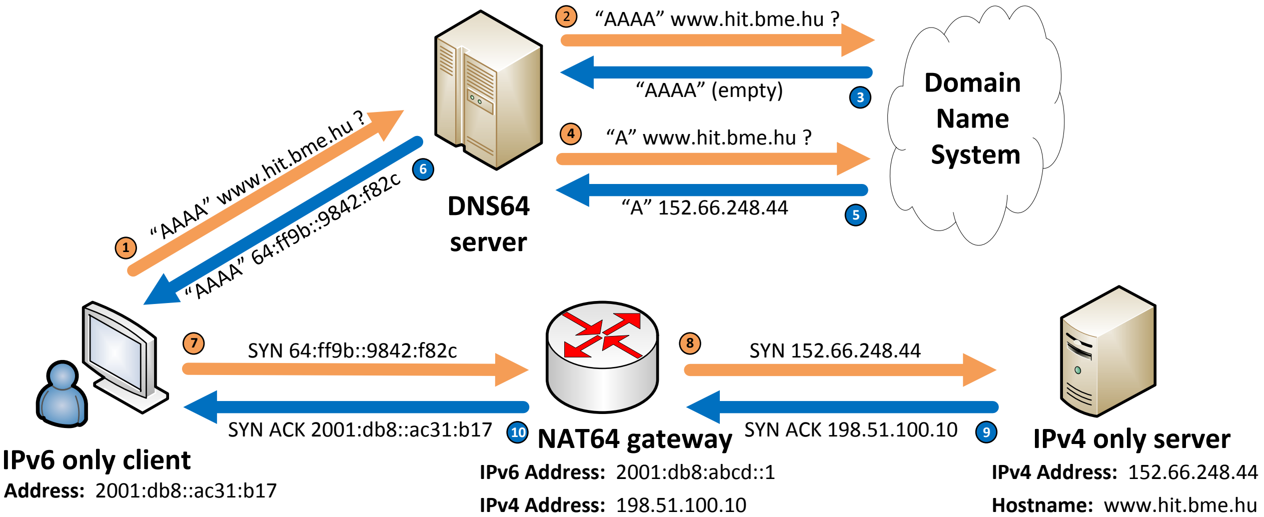


Fig. 7. The operation of the DNS64+NAT64 solution: an IPv6 only client communicates with and IPv4 only server.

using the necessary information from its state table. It sends the IPv6 packet back to the IPv6 only client.

The communication may continue on. It seems to the clients that it communicates to an IPv6 server. Similarly, the server "can see" an IPv4 client. If it logs the IP addresses of the clients than it will log the public IPv4 address of the NAT64 gateway.

Most client-server applications can work well with the DNS64+NAT64 solution. See more information about the application compatibility in: [13]–[15].

In practice, the world wide usage of the *NAT64 Well-Known Prefix* has several hindrances, see sections 3.1 and 3.2 of [6]. Therefore the network operators allocate a subnet from their own network for this purpose. It is called *Network Specific Prefix* (NSP).

→ *The DNS64 server must enable the user to set the appropriate prefix for synthesizing the IPv4-embedded IPv6 address.*

C. Construction of the IPv4-Embedded IPv6 Addresses

The construction of the IPv4-embedded IPv6 addresses is defined in [6]. When using *Network-Specific Prefix*, the network administrator has to decide the size of the prefix. There are some constraints:

- The prefix size must be exactly one of 32, 40, 48, 56, 64 or 96.
- The 64-71 bits of the IPv6 address must be 0.
- The 32 bits of the IPv4 address are stored right after the prefix but the above mentioned 0 bits have to be left out (or jumped over).
- If there are unused bits at the end of the IPv6 address then they must be filled with 0-s.

→ *The DNS64 server should be able to check the prefix size and accept only the permitted ones.*

D. Operation Requirements for the DNS64 server

The DNS64 server is set as the normal DNS server of the client.

→ *Therefore the DNS64 server must be able to act as a proxy for any other requests than the AAAA records (e.g. MX).*

Even though DNS64 is intended as an IPv6 transition solution for the IPv6 only clients, the clients might use dual stack.

→ *Therefore A record requests and their replies must also be forwarded untouched.*

III. Design Decisions

A. Design Principles

Our intention was to create a DNS64 server program that can be a viable alternative to the existing free software DNS64 implementations. Its attributes must include ease of use, high performance and ease of modification. In our position, a program like this should be:

- simple and therefore short (in source code)
- fast (written in C, at most some parts in C++)
- extensible (well structured and well documented)
- convenient and flexible in configuration
- free software under GPL or BSD license

B. High Level Design Decisions

1) Forwarder or recursor

A DNS server may operate in two modes. If it works as a *recursor* then it performs the recursion itself: starting from a top level DNS server it performs a series of iterative queries until it receives an authoritative answer. If it works as a *forwarder* then it acts like a proxy: forwards the queries to another DNS server and simply returns its answer to the client. (It may also cache the information.) As for the before mentioned four free DNS64 implementations, BIND and PowerDNS can act as both recursor and forwarder. TOTD can act as a forwarder only. Unbound can be either of them if it is used as a DNS server only, but it may perform the DNS64 functionality only in the case if it is set as a recursor.

We decided that MTD64 will operate as a forwarder only. It complies with the principle of simplicity.

2) Caching

On the one hand caching may significantly improve the performance of a DNS server, but on the other hand it seriously increases complexity. In addition to that the most common desktop operating systems, i.e. the different versions of Windows and Linux use DNS caching, thus they do not send the subsequent requests of the clients concerning the same domain name to the DNS server. However, if the DNS64 server is used by several clients then many of them may send requests for the same set of domain names thus caching is very likely to be beneficial.

We decided to omit caching from the first version of MTD64 because implementing it would have required more time than it was available during the final project (MSc thesis) of the second author of this paper. It is planned to be added later as a separate project for another student.

3) Storing the requests or not

When the DNS64 server receives a request from the client and forwards it to the DNS system, the DNS64 server should preserve the information about the client while waiting for the reply to be able to send back the reply (or the synthesized IPv6 address) to the client. The requests from the different clients may arrive in high number therefore an expandable data structure should be chosen e.g. linked list, balanced or unbalanced trees. Their operations (insert, find, delete) involve programming complexity and the operations may involve significant time complexity if the data structure has high number of elements. Unfortunately there is a trade-off between the programming complexity and the speed. E.g. the operations of the linked list are simple but their time complexity is $O(n)$, where n denotes the number of elements in the data structure. The time complexity of the operations of the balanced trees is $O(\log n)$, but their operations require more programming work. For more information see [16] and its references.

We decided not to store explicitly the client information but start a new thread for each request. It means the information is stored on the stack in the local variables (and on the heap in dynamically allocated data structures held by pointers). We hope that this solution will not fight back through high memory consumption but it will turn out during performance testing. As a positive consequence of our decision, MTD64 will be able to utilize all the CPU cores of the server.

4) Programming language and program structure

The C++ programming language was chosen mainly for its thread handling. Only one class is used: its tasks are to store the parameters set by the user and to make them available by member function calls. The majority of the source code is written in the C language to be as fast as possible. One main source code file contains the most important operation of the server program and two separate ones contains the code for loading and storing the settings. They all include the same single header file. See more details later on.

5) Configuration file format

Simple text format was chosen. The configuration file is line oriented: a keyword is followed by the values for the given setting. Both "#" and "//" can be used for denoting comments.

6) Logging

The MTD64 program uses the standard syslog facility for logging. The program uses multiple log levels and the amount of the logged information can also be set by the user in the configuration file of MTD64.

7) License

The GPL v2 license was chosen. It ensures that the derivatives of MTD64 will remain also free software.

C. Important Design Details

1) DNS servers and selection between them

Multiple name servers may be set. They can be added by using multiple lines. Also the configured name servers from the (Linux) operating system can be loaded. Two DNS server selection modes are supported. *Round Robin* uses the first one from the list while it replies on time. If time-out occurs, than it takes the next one from the list. *Random* chooses one randomly for every request. Note that this solution makes it possible to use the DNS servers balanced or unbalanced: e.g. one of them is specified 10 times and the other one is specified 20 times.

The random DNS server selection mode will also be useful when testing the performance of the MTD64 software: multiple DNS servers can be used so that their performance will not limit the performance of the MTD64 software.

2) DNS message length

DNS messages carried over the UDP transport protocol are limited to 512 octets. A DNS server may return multiple RR entries in its answer, thus its size may be close to 512 octets. When IPv4-embedded IPv6 addresses are synthesized from IPv4 addresses it results in a 16-4=12 octets growth for each IP address. Therefore care must be taken to the 512 octet limit. As certain programs may handle larger datagrams and others may not, we decided to entrust the decision to the user. Therefore the maximum length of the response of the MTD64 server can be set in the configuration file. If a resource record does not fit in the specified size of DNS reply message, the program leaves out the resource record and also logs the event. It does not set the TC bit, because by doing so it would force the client to repeat the query by using TCP, see [12].

3) Client and DNS server IP version

The IP version for the client side is obvious: the IPv6 only clients use IPv6. What IP version should be used to reach the DNS system? Theoretically the request for the "AAAA" record might also be sent over IPv6, but we found a safe and simple choice to use always IPv4. (It simplifies both the setting of the DNS servers in the configuration file and the communication with them.)

4) Order of questions and answers

Section 5.1.8 of [3] states that: "The DNS64 MAY perform the query for the AAAA RR and for the A RR in parallel, in order to minimize the delay." However, this possible speed up has its price in assembling and sending always two questions instead of one[1] as well as taking care for which one the answer has already arrived, therefore we decided not to do this, but rather follow the order shown in Fig 7.

5) Preparation of the answers to the clients

If the question of the client was different than an "AAAA" record (e.g. "A" record, "MX" record, etc.) or the client asked for an "AAAA" record and the DNS system responded with an "AAAA" record than it is enough to forward its reply to the client. (It can be done without any changes, because even the Transaction ID is matching since MTD64 forwarded the request of the client untouched to the DNS system which one also kept the Transaction ID.) When an "AAAA" record must be synthetized from an "A" record, we saw two possible ways for completing this task:

1. The complete reply can be assembled step-by-step "from scratch" using the information piece-by-piece from the reply of the DNS system. (It requires a lot of steps, see the fields of the DNS messages.)
2. The reply can be built in larger chunks by copying as long as possible memory areas from the reply of the DNS system.

The second one was chosen to achieve higher speed. The size of the chunks is limited by the occurrences of the "A" records: the 4 octet long IPv4 addresses have to be replaced by the synthesized IPv6 addresses which requires 16 octets space. Special care must be taken for the domain names containing pointers whether they have to be adjusted. (Recall that the RRs in the DNS answers also contain the questions with specially encoded and possibly compressed the domain names.)

D. Further Design Details

The presentation of all the design details would exceed the limitations of this paper. They are included in the "Programmer Documentation". For those who would like only to use MTD64, we recommend the "User Documentation". They can be found together with the commented source code on GitHub [17]. We also present a simple configuration file in the appendix, to give an impression of how flexibly MTD64 can be configured.

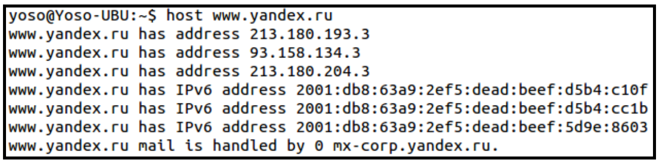
IV. Implementation

As both programmer documentation and commented source code are available on GitHub [17], we give only a high level overview here.

A. Source Files and their Responsibilities

There is a single header file **`header.h`** containing all the necessary system header file includes, the definition of the **`ConfigModule`** class and some function prototypes. It is included by all program files.

[1] We note that more and more Internet hosts will have IPv6 addresses in the future and, therefore, the "A" record will have to be asked less frequently.



```
yoso@Yoso-UBU:~$ host www.yandex.ru
www.yandex.ru has address 213.180.193.3
www.yandex.ru has address 93.158.134.3
www.yandex.ru has address 213.180.204.3
www.yandex.ru has IPv6 address 2001:db8:63a9:2ef5:dead:beef:d5b4:c10f
www.yandex.ru has IPv6 address 2001:db8:63a9:2ef5:dead:beef:d5b4:cc1b
www.yandex.ru has IPv6 address 2001:db8:63a9:2ef5:dead:beef:5d9e:8603
www.yandex.ru mail is handled by 0 mx-corp.yandex.ru.
```

Fig. 8. Output of the functional testing command.

The **config_load.cpp** file contains a single function **load_config()**, which is responsible for loading all parameters from the **settings.conf** configuration file and storing them in an instance of the **ConfigModule** class.

The member functions of the **ConfigModule** class are defined in **config_module.cpp** and their task is to set and retrieve all data members of the class containing configuration information as well as to synthesize *IPv4-embedded IPv6 address*es.

The main source file is called **dns64server.cpp** and it contains some string and error handling functions, the **main()** function and the **send_response()** function which one is responsible for the lion's share of the tasks of MTD64.

A workable sample configuration file **settings.conf** and a **Makefile** are also provided.

B. *Operation of MTD64 in a Nutshell*

The **main()** function calls the **load_config()** function to read the parameters, opens an UDP socket for receiving DNS queries and then starts an infinite loop, in which it receives DNS queries and starts a separate thread for handling each of them. The executed code of the thread is the **send_response()** function. It checks and records if an "AAAA" record was requested by the client. Then it forwards the query to the appropriate DNS server (determined by the settings) keeping everything (including the *Transaction ID*) untouched in it. Next, it sets the appropriate timeout (defined by the user in the configuration file) using the **setsockopt()** socket handling function and calls the **recfvrom()** function for receiving the reply from the DNS server. (If timeout occurs then it resends the query by at most as many times as the number set by the user in the configuration file, but now we do not go into deeper details.) If finally a response is received then it checks in its own records, if an "AAAA" record was requested by the client. If yes, and no "AAAA" record was received in the answer of the DNS server then it prepares a DNS query for an "A" record by modifying the query type in the preserved DNS query message from the client. Then it sends the query to the appropriate DNS server and receives its reply in the above mentioned manner (using timeout and resending the query if necessary). If it receives at least one "A" record then it modifies the message from the DNS server by replacing all "A" records with synthesized "AAAA" records (note that there may be more than one of them) taking care also for modifying pointers if necessary. It also considers the maximum DNS message length allowed by the standard (512 octets) or set for some other value by the user. (It may be necessary to omit some records at the end of the DNS message.) Finally, it returns either the untouched reply of the DNS server (if no change was necessary) or the modified one – containing one or more *IPv4-embedded IPv6 address*(es) – to the client. Note that the *Transaction ID* was always left unchanged thus the reply will meet the expectation of the client.

V. Testing

Unless stated otherwise, the settings of the sample configuration file were kept and the standard Linux **host** command was used for testing. The IPv6 address of the client was fe80::221:ccff:fe69:9f0a and the IPv6 address of the MTD64 server was: fe80::8e89:a5ff:fec5:5bef during all the functional and message length tests.

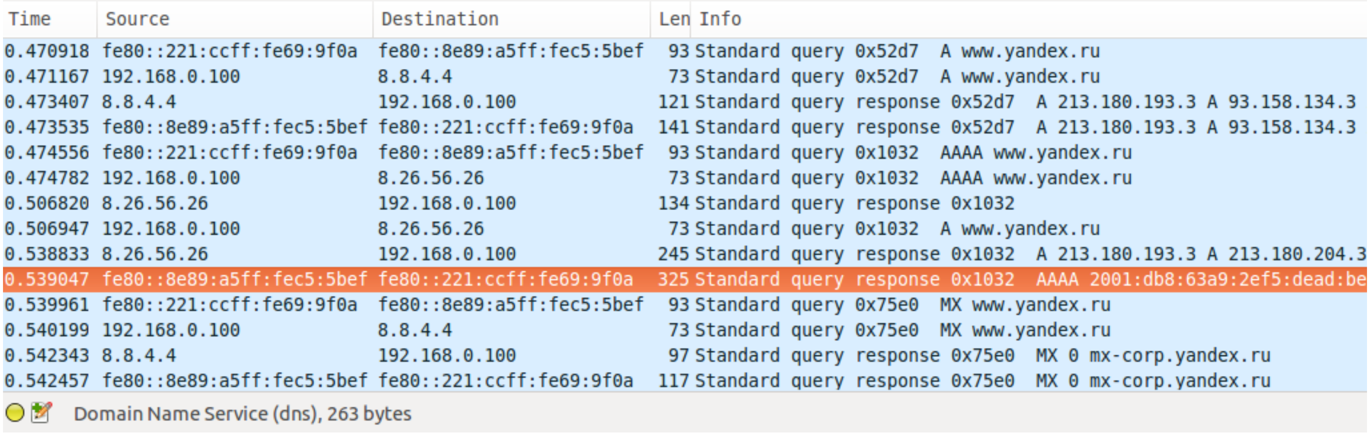
A. *Functional Testing*

1) Command line testing and observation

The **host www.yandex.ru** command was used for testing. As no other parameters were specified, the **host** command sent three queries and they asked "A", "AAAA" and "MX" records. The result of the command is shown in Fig. 8. It can be observed that the domain has three IPv4 addresses and no IPv6 address, therefore three IPv6 addresses were synthesized by MTD64 using the IPv4 addresses and the 2001:db8:63a9:2ef5:dead:beef::/96 prefix. Please note that the IPv4 and the corresponding *IPv4-embedded IPv6 addresses* are in a different order; we shall show its reason soon.

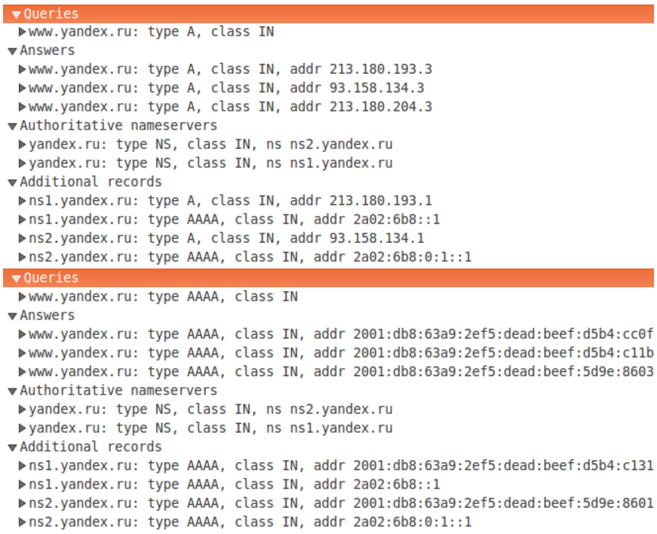
2) Wireshark capture analysis

The network traffic of the MTD64 server was captured by Wireshark during the execution of the **host** command for a more thorough checking of the operation of our DNS64 implementation. The captured packets are shown in Fig. 9. By observing the first four of them, we can see that MTD64



```
Time     Source                    Destination               Len Info
0.470918 fe80::221:ccff:fe69:9f0a  fe80::8e89:a5ff:fec5:5bef  93 Standard query 0x52d7  A www.yandex.ru
0.471167 192.168.0.100             8.8.4.4                    73 Standard query 0x52d7  A www.yandex.ru
0.473407 8.8.4.4                   192.168.0.100             121 Standard query response 0x52d7  A 213.180.193.3 A 93.158.134.3
0.473535 fe80::8e89:a5ff:fec5:5bef fe80::221:ccff:fe69:9f0a  141 Standard query response 0x52d7  A 213.180.193.3 A 93.158.134.3
0.474556 fe80::221:ccff:fe69:9f0a  fe80::8e89:a5ff:fec5:5bef  93 Standard query 0x1032  AAAA www.yandex.ru
0.474782 192.168.0.100             8.26.56.26                 73 Standard query 0x1032  AAAA www.yandex.ru
0.506820 8.26.56.26                192.168.0.100             134 Standard query response 0x1032
0.506947 192.168.0.100             8.26.56.26                 73 Standard query 0x1032  A www.yandex.ru
0.538833 8.26.56.26                192.168.0.100             245 Standard query response 0x1032  A 213.180.193.3 A 213.180.204.3
0.539047 fe80::8e89:a5ff:fec5:5bef fe80::221:ccff:fe69:9f0a  325 Standard query response 0x1032  AAAA 2001:db8:63a9:2ef5:dead:be
0.539961 fe80::221:ccff:fe69:9f0a  fe80::8e89:a5ff:fec5:5bef  93 Standard query 0x75e0  MX www.yandex.ru
0.540199 192.168.0.100             8.8.4.4                    73 Standard query 0x75e0  MX www.yandex.ru
0.542343 8.8.4.4                   192.168.0.100              97 Standard query response 0x75e0  MX 0 mx-corp.yandex.ru
0.542457 fe80::8e89:a5ff:fec5:5bef fe80::221:ccff:fe69:9f0a  117 Standard query response 0x75e0  MX 0 mx-corp.yandex.ru
```

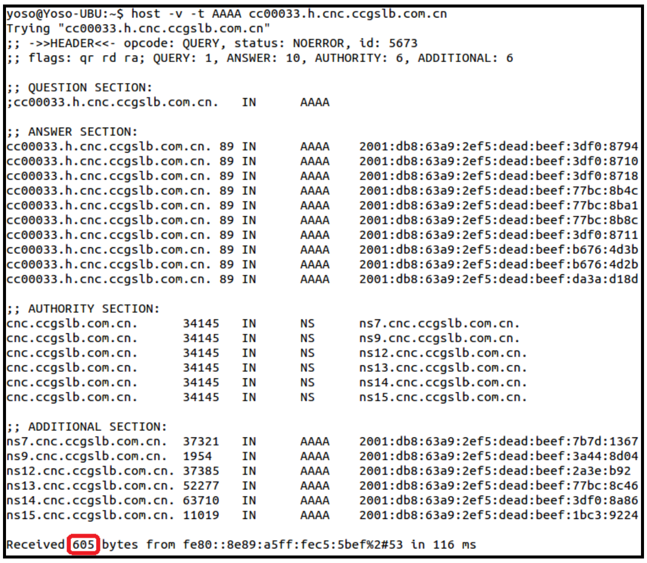
Fig. 9. DNS messages during the basic functional testing – captured and displayed by Wireshark.



```
▼Queries
  ▶www.yandex.ru: type A, class IN
▼Answers
  ▶www.yandex.ru: type A, class IN, addr 213.180.193.3
  ▶www.yandex.ru: type A, class IN, addr 93.158.134.3
  ▶www.yandex.ru: type A, class IN, addr 213.180.204.3
▼Authoritative nameservers
  ▶yandex.ru: type NS, class IN, ns ns2.yandex.ru
  ▶yandex.ru: type NS, class IN, ns ns1.yandex.ru
▼Additional records
  ▶ns1.yandex.ru: type A, class IN, addr 213.180.193.1
  ▶ns1.yandex.ru: type AAAA, class IN, addr 2a02:6b8::1
  ▶ns2.yandex.ru: type A, class IN, addr 93.158.134.1
  ▶ns2.yandex.ru: type AAAA, class IN, addr 2a02:6b8:0:1::1
▼Queries
  ▶www.yandex.ru: type AAAA, class IN
▼Answers
  ▶www.yandex.ru: type AAAA, class IN, addr 2001:db8:63a9:2ef5:dead:beef:d5b4:cc0f
  ▶www.yandex.ru: type AAAA, class IN, addr 2001:db8:63a9:2ef5:dead:beef:d5b4:c11b
  ▶www.yandex.ru: type AAAA, class IN, addr 2001:db8:63a9:2ef5:dead:beef:5d9e:8603
▼Authoritative nameservers
  ▶yandex.ru: type NS, class IN, ns ns2.yandex.ru
  ▶yandex.ru: type NS, class IN, ns ns1.yandex.ru
▼Additional records
  ▶ns1.yandex.ru: type AAAA, class IN, addr 2001:db8:63a9:2ef5:dead:beef:d5b4:c131
  ▶ns1.yandex.ru: type AAAA, class IN, addr 2a02:6b8::1
  ▶ns2.yandex.ru: type AAAA, class IN, addr 2001:db8:63a9:2ef5:dead:beef:5d9e:8601
  ▶ns2.yandex.ru: type AAAA, class IN, addr 2a02:6b8:0:1::1
```

Fig. 10. How MTD64 synthesizes IPv4-embedded IPv6 messages?

acted like a proxy when an “A” record was requested: the first message was sent from the client to the MTD64 server over IPv6. The second message contained the same query and is was sent from the MTD64 server to a public DNS server at 8.8.4.4 over IPv4. Similarly the response of the public DNS server was simply forwarded to the client by MTD64. The next six lines show the resolution of the query for an “AAAA” record. Similarly to the case of the query for the “A” record, MTD64 forwarded the request of the client to the public DNS server. Let us observe that the 0x1032 *Transaction ID* was also kept. However, MTD64 received an “empty” answer. Therefore, it sent a query for an “A” record using the same *Transaction ID*. We can observe that the public DNS server sent the same three IPv4 addresses as before but in a different order (now, the first two begin with the 213.180/16 prefix). This is the explanation of our earlier observation of the order change (based on the output of the host command shown in Fig. 8.) And let us also observe that whereas the length of the response of the public DNS server is 245 bytes, the length of the response of the MTD64 server is 325 bytes. The difference is 80 bytes. We shall explain its reason soon by a deeper analysis of the two messages. The last four lines show the resolution of the query for an “AAAA” record. Here, MTD64 acted as a proxy again, thus we do not go into details.



```
yoso@Yoso-UBU:~$ host -v -t AAAA cc00033.h.cnc.ccgslb.com.cn
Trying "cc00033.h.cnc.ccgslb.com.cn"
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 5673
;; flags: qr rd ra; QUERY: 1, ANSWER: 10, AUTHORITY: 6, ADDITIONAL: 6

;; QUESTION SECTION:
;cc00033.h.cnc.ccgslb.com.cn.   IN      AAAA

;; ANSWER SECTION:
cc00033.h.cnc.ccgslb.com.cn. 89 IN      AAAA    2001:db8:63a9:2ef5:dead:beef:3df0:8794
cc00033.h.cnc.ccgslb.com.cn. 89 IN      AAAA    2001:db8:63a9:2ef5:dead:beef:3df0:8710
cc00033.h.cnc.ccgslb.com.cn. 89 IN      AAAA    2001:db8:63a9:2ef5:dead:beef:3df0:8718
cc00033.h.cnc.ccgslb.com.cn. 89 IN      AAAA    2001:db8:63a9:2ef5:dead:beef:77bc:8b4c
cc00033.h.cnc.ccgslb.com.cn. 89 IN      AAAA    2001:db8:63a9:2ef5:dead:beef:77bc:8ba1
cc00033.h.cnc.ccgslb.com.cn. 89 IN      AAAA    2001:db8:63a9:2ef5:dead:beef:77bc:8b8c
cc00033.h.cnc.ccgslb.com.cn. 89 IN      AAAA    2001:db8:63a9:2ef5:dead:beef:3df0:8711
cc00033.h.cnc.ccgslb.com.cn. 89 IN      AAAA    2001:db8:63a9:2ef5:dead:beef:b676:4d3b
cc00033.h.cnc.ccgslb.com.cn. 89 IN      AAAA    2001:db8:63a9:2ef5:dead:beef:b676:4d2b
cc00033.h.cnc.ccgslb.com.cn. 89 IN      AAAA    2001:db8:63a9:2ef5:dead:beef:da3a:d18d

;; AUTHORITY SECTION:
cnc.ccgslb.com.cn.      34145   IN      NS      ns7.cnc.ccgslb.com.cn.
cnc.ccgslb.com.cn.      34145   IN      NS      ns9.cnc.ccgslb.com.cn.
cnc.ccgslb.com.cn.      34145   IN      NS      ns12.cnc.ccgslb.com.cn.
cnc.ccgslb.com.cn.      34145   IN      NS      ns13.cnc.ccgslb.com.cn.
cnc.ccgslb.com.cn.      34145   IN      NS      ns14.cnc.ccgslb.com.cn.
cnc.ccgslb.com.cn.      34145   IN      NS      ns15.cnc.ccgslb.com.cn.

;; ADDITIONAL SECTION:
ns7.cnc.ccgslb.com.cn.  37321   IN      AAAA    2001:db8:63a9:2ef5:dead:beef:7b7d:1367
ns9.cnc.ccgslb.com.cn.  1954    IN      AAAA    2001:db8:63a9:2ef5:dead:beef:3a44:8d04
ns12.cnc.ccgslb.com.cn. 37385   IN      AAAA    2001:db8:63a9:2ef5:dead:beef:2a3e:b92
ns13.cnc.ccgslb.com.cn. 52277   IN      AAAA    2001:db8:63a9:2ef5:dead:beef:77bc:8c46
ns14.cnc.ccgslb.com.cn. 63710   IN      AAAA    2001:db8:63a9:2ef5:dead:beef:3df0:8a86
ns15.cnc.ccgslb.com.cn. 11019   IN      AAAA    2001:db8:63a9:2ef5:dead:beef:1bc3:9224

Received 605 bytes from fe80::8e89:a5ff:fec5:5bef%2#53 in 116 ms
```

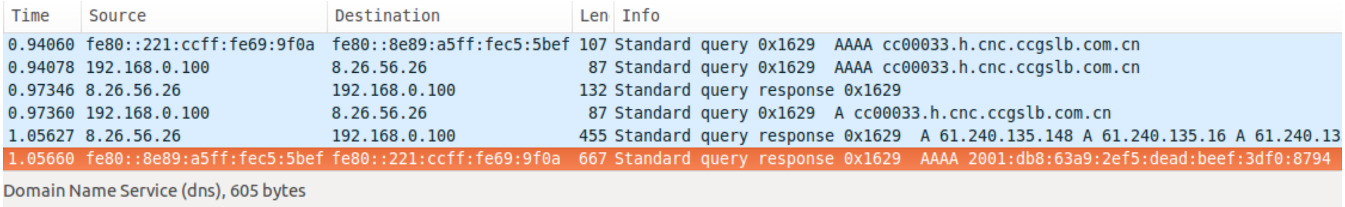
Fig. 11. A DNS message which is longer than 512 octets.

3) Deeper analysis of the messages

Now, we compare two messages: the one with the “A” records from the public DNS server to the MTD64 server and the one with the synthesized “AAAA” records from the MTD64 server to the client. They are shown in Fig. 10. The first message contains the query, the three answers as “A” records, the names of two authoritative name servers and four additional records. From the four additional records, two ones are of type “A” and the other two ones are of type “AAAA”. MTD64 transformed this message into the second one. How many differences can be observed? There are five of them: the three “A” records as answers and the two “A” records from among the additional records were transformed into “AAAA” records. Each transformation is responsible for a length growth by 12 octets. The IP headers are not displayed here but it can be checked in Fig. 9 that the first message travelled over IPv4 and the second one was sent over IPv6. The length of the standard IPv4 and IPv6 headers are 20 octets and 40 octets, respectively. Now, we have shown that the difference between the lengths of the two messages is exactly 5*12+20=80 octets.

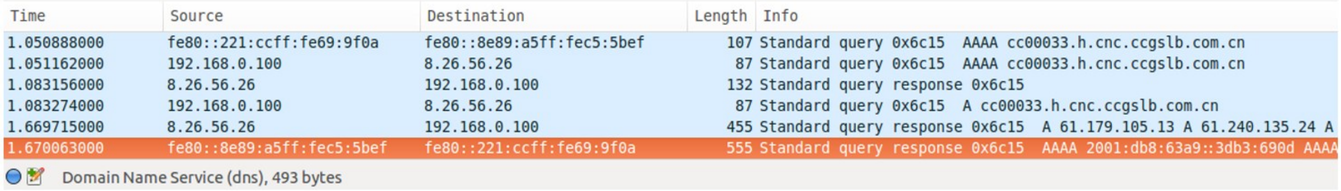
B. Testing DNS Message Length Issues

As we have just seen, the DNS64 functionality increases the length of the DNS messages. What happens if we reach



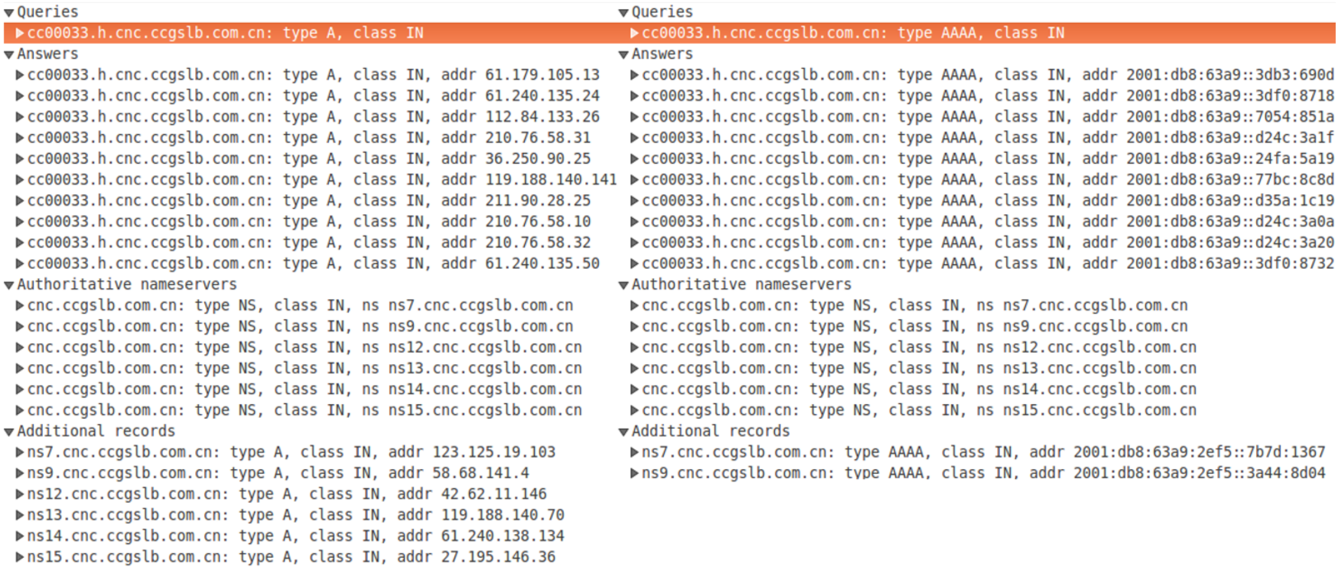
```
Time     Source                    Destination               Len Info
0.94060  fe80::221:ccff:fe69:9f0a  fe80::8e89:a5ff:fec5:5bef 107 Standard query 0x1629  AAAA cc00033.h.cnc.ccgslb.com.cn
0.94078  192.168.0.100             8.26.56.26                 87 Standard query 0x1629  AAAA cc00033.h.cnc.ccgslb.com.cn
0.97346  8.26.56.26                192.168.0.100             132 Standard query response 0x1629
0.97360  192.168.0.100             8.26.56.26                 87 Standard query 0x1629  A cc00033.h.cnc.ccgslb.com.cn
1.05627  8.26.56.26                192.168.0.100             455 Standard query response 0x1629  A 61.240.135.148 A 61.240.135.16 A 61.240.13
1.05660  fe80::8e89:a5ff:fec5:5bef fe80::221:ccff:fe69:9f0a  667 Standard query response 0x1629  AAAA 2001:db8:63a9:2ef5:dead:beef:3df0:8794
Domain Name Service (dns), 605 bytes
```

Fig. 12. DNS messages with 700 octets maximum length limit set – captured and displayed by Wireshark.



```
Time          Source                    Destination               Length Info
1.050888000   fe80::221:ccff:fe69:9f0a  fe80::8e89:a5ff:fec5:5bef    107 Standard query 0x6c15  AAAA cc00033.h.cnc.ccgslb.com.cn
1.051162000   192.168.0.100             8.26.56.26                    87 Standard query 0x6c15  AAAA cc00033.h.cnc.ccgslb.com.cn
1.083156000   8.26.56.26                192.168.0.100                132 Standard query response 0x6c15
1.083274000   192.168.0.100             8.26.56.26                    87 Standard query 0x6c15  A cc00033.h.cnc.ccgslb.com.cn
1.669715000   8.26.56.26                192.168.0.100                455 Standard query response 0x6c15  A 61.179.105.13 A 61.240.135.24 A
1.670063000   fe80::8e89:a5ff:fec5:5bef fe80::221:ccff:fe69:9f0a     555 Standard query response 0x6c15  AAAA 2001:db8:63a9::3db3:690d AAAA
   Domain Name Service (dns), 493 bytes
```

Fig. 13. DNS messages with standard 512 octets maximum length limit set – captured and displayed by Wireshark.



```
▼Queries
 ▶cc00033.h.cnc.ccgslb.com.cn: type A, class IN
▼Answers
 ▶cc00033.h.cnc.ccgslb.com.cn: type A, class IN, addr 61.179.105.13
 ▶cc00033.h.cnc.ccgslb.com.cn: type A, class IN, addr 61.240.135.24
 ▶cc00033.h.cnc.ccgslb.com.cn: type A, class IN, addr 112.84.133.26
 ▶cc00033.h.cnc.ccgslb.com.cn: type A, class IN, addr 210.76.58.31
 ▶cc00033.h.cnc.ccgslb.com.cn: type A, class IN, addr 36.250.90.25
 ▶cc00033.h.cnc.ccgslb.com.cn: type A, class IN, addr 119.188.140.141
 ▶cc00033.h.cnc.ccgslb.com.cn: type A, class IN, addr 211.90.28.25
 ▶cc00033.h.cnc.ccgslb.com.cn: type A, class IN, addr 210.76.58.10
 ▶cc00033.h.cnc.ccgslb.com.cn: type A, class IN, addr 210.76.58.32
 ▶cc00033.h.cnc.ccgslb.com.cn: type A, class IN, addr 61.240.135.50
▼Authoritative nameservers
 ▶cnc.ccgslb.com.cn: type NS, class IN, ns ns7.cnc.ccgslb.com.cn
 ▶cnc.ccgslb.com.cn: type NS, class IN, ns ns9.cnc.ccgslb.com.cn
 ▶cnc.ccgslb.com.cn: type NS, class IN, ns ns12.cnc.ccgslb.com.cn
 ▶cnc.ccgslb.com.cn: type NS, class IN, ns ns13.cnc.ccgslb.com.cn
 ▶cnc.ccgslb.com.cn: type NS, class IN, ns ns14.cnc.ccgslb.com.cn
 ▶cnc.ccgslb.com.cn: type NS, class IN, ns ns15.cnc.ccgslb.com.cn
▼Additional records
 ▶ns7.cnc.ccgslb.com.cn: type A, class IN, addr 123.125.19.103
 ▶ns9.cnc.ccgslb.com.cn: type A, class IN, addr 58.68.141.4
 ▶ns12.cnc.ccgslb.com.cn: type A, class IN, addr 42.62.11.146
 ▶ns13.cnc.ccgslb.com.cn: type A, class IN, addr 119.188.140.70
 ▶ns14.cnc.ccgslb.com.cn: type A, class IN, addr 61.240.138.134
 ▶ns15.cnc.ccgslb.com.cn: type A, class IN, addr 27.195.146.36
```

```
▼Queries
 ▶cc00033.h.cnc.ccgslb.com.cn: type AAAA, class IN
▼Answers
 ▶cc00033.h.cnc.ccgslb.com.cn: type AAAA, class IN, addr 2001:db8:63a9::3db3:690d
 ▶cc00033.h.cnc.ccgslb.com.cn: type AAAA, class IN, addr 2001:db8:63a9::3df0:8718
 ▶cc00033.h.cnc.ccgslb.com.cn: type AAAA, class IN, addr 2001:db8:63a9::7054:851a
 ▶cc00033.h.cnc.ccgslb.com.cn: type AAAA, class IN, addr 2001:db8:63a9::d24c:3a1f
 ▶cc00033.h.cnc.ccgslb.com.cn: type AAAA, class IN, addr 2001:db8:63a9::24fa:5a19
 ▶cc00033.h.cnc.ccgslb.com.cn: type AAAA, class IN, addr 2001:db8:63a9::77bc:8c8d
 ▶cc00033.h.cnc.ccgslb.com.cn: type AAAA, class IN, addr 2001:db8:63a9::d35a:1c19
 ▶cc00033.h.cnc.ccgslb.com.cn: type AAAA, class IN, addr 2001:db8:63a9::d24c:3a0a
 ▶cc00033.h.cnc.ccgslb.com.cn: type AAAA, class IN, addr 2001:db8:63a9::d24c:3a20
 ▶cc00033.h.cnc.ccgslb.com.cn: type AAAA, class IN, addr 2001:db8:63a9::3df0:8732
▼Authoritative nameservers
 ▶cnc.ccgslb.com.cn: type NS, class IN, ns ns7.cnc.ccgslb.com.cn
 ▶cnc.ccgslb.com.cn: type NS, class IN, ns ns9.cnc.ccgslb.com.cn
 ▶cnc.ccgslb.com.cn: type NS, class IN, ns ns12.cnc.ccgslb.com.cn
 ▶cnc.ccgslb.com.cn: type NS, class IN, ns ns13.cnc.ccgslb.com.cn
 ▶cnc.ccgslb.com.cn: type NS, class IN, ns ns14.cnc.ccgslb.com.cn
 ▶cnc.ccgslb.com.cn: type NS, class IN, ns ns15.cnc.ccgslb.com.cn
▼Additional records
 ▶ns7.cnc.ccgslb.com.cn: type AAAA, class IN, addr 2001:db8:63a9:2ef5::7b7d:1367
 ▶ns9.cnc.ccgslb.com.cn: type AAAA, class IN, addr 2001:db8:63a9:2ef5::3a44:8d04
```

Fig. 14. Comparison of the DNS reply with the A records (on the left) and the MTD64 reply with synthesized AAAA records (on the right). Some of the additional RRs were omitted due to the standard 512 octets DNS message size limit.

the 512 octets limit? To be able to test this situation, we had to find an appropriate domain name. An appropriate one can be found as follows: the **host -t AAAA www.gmw.cn** command results in several CNAME-s and one of them, namely **cc00033.h.cnc.ccgslb.com.cn** is suitable for this purpose.

1) Testing with 700 octets limit

The configuration file was modified as follows:

```
response-maxlength 700
```

Fig. 11 shows the results of the execution of the **host** command querying the "AAAA" record of the above mentioned domain name. The command also displayed the length of the DNS message: 605 bytes (marked by a red rectangle). Please note that the host command did not give an error message because of the DNS message was longer than 512 octets – even in the verbose mode (set by the **-v** option). Wireshark also displayed this message with no error, see Fig. 12. (The DNS payload size is also reported as 605 bytes in the bottom left corner of the figure.)

2) Testing with standard 512 octets limit

The configuration file was modified as follows:

```
response-maxlength 512
dns64-prefix 2001:0db8:63a9::/96
```

The same **host** command was issued again. Fig. 13 shows the Wireshark results. Now the payload length is only 493 bytes. Fig. 14 shows the reply of the public DNS server (with the "A" records) and the reply of the MTD64 server (with synthesized "AAAA" records). It can be seen that some of the additional RRs were omitted by MTD64 due to the standard 512 octets DNS message size limit.

C. Basic Performance Testing

During the review process of this journal paper, we have compared the performance of MTD64 to that of BIND, and it was found that MTD64 significantly outperformed BIND concerning the number of answered AAAA record requests per second [18]. However, as that paper is still under review (and therefore it is not citable yet), we have performed more measurements using different DNS64 server hardware to avoid copyright issues.

1) Test setup

The topology of our performance test network is shown in Fig. 15. A Raspberry Pi 2 Model B+ single-board computer was used as DUT (Device Under Test) to execute the DNS64 server programs to be compared. The **dns64perf** [19] test program was executed by a laptop computer for performance measurement. The authoritative DNS server was executed by a high performance desktop computer. The elements were interconnected by a Gigabit Ethernet switch. This setup was prepared so that the DUT be the bottleneck, thus its performance determined the overall performance of the test system.

2) Hardware and software parameters

For the repeatability of our measurements, we provide hardware and software details.

The authoritative DNS server was a desktop computer with 3.2GHz Intel Core i5-4570 CPU (4 cores, 6MB cache),

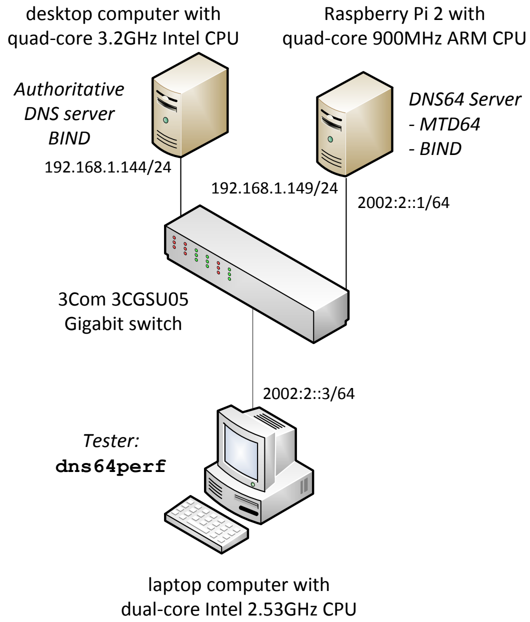


Fig. 15. Topology of the performance test network

TABLE I
BASIC PERFORMANCE COMPARISON OF MTD64 AND BIND

DNS64 Implementation		MTD64	BIND
Execution time of one experiment (ms)	average	65.87	122.27
	std. dev.	5.08	4.67
Replied AAAA queries per second		3886	2094

16GB 1600MHz DDR3 SDRAM, 250GB Samsung 840 EVO SSD, Realtek RTL8111F PCI Express Gigabit Ethernet NIC; Debian GNU/Linux 8.2 operating system, 3.2.0-4-amd64 kernel, BIND 9.9.5-9+deb8u3-Debian

The DUT was a Raspberry Pi 2 Model B single-board computer with 900MHz quad-core ARM Cortex A7 CPU, 1GB 400MHz LPDDR2 SDRAM, 16GB Kingston micro SD card, 100BaseTX Ethernet NIC; Debian GNU/Linux 8.0 operating system, 3.18.0-trunk-rpi2 kernel, BIND 9.9.5-9+deb8u2-Debian, MTD64 from [17] (Latest commit: January 4, 2015).

The tester device was a Dell Latitude E6400 series laptop with 2.53GHz Intel Core2 Duo T9400 CPU (2 cores, 6MB cache), 4GB 800MHz DDR2 SDRAM, 250GB Samsung 840 EVO SSD, Intel 82567LM Gigabit Ethernet NIC; Debian GNU/Linux 8.2 operating system, 3.2.0-4-amd64 kernel, **`dns64perf`** test program from [20].

The devices were interconnected by a 3CGSU05 5-port 3Com Gigabit Ethernet switch.

3) Testing method

The **`dns64perf`** program sent AAAA record queries for the domain names **`10-0-`*`b`*`-`*`c`*`.dns64perf.test`**, where variables *b* and *c* took their values from the [0..255] interval, thus altogether 65536 queries were sent. The domain names were resolved to the 10.0.*b.c* IPv4 addresses by the authoritative DNS server. As it is documented in [19], the **`dns64perf`** program organizes the 65536 name resolutions into 256 "experiments". During an experiment, variable *b* has a fixed value and the program can use several threads specified by the user. The program measures the execution time of the experiments and prints out the 256 results (in milliseconds). For further details, see [19]. For this time, 16 threads were used to send 16 queries concurrently in order to ensure high enough load.

4) Results

The result are presented in Table I. The *N* number or the replied AAAA record queries per second is calculated according to (1), where *T* denotes the execution time of one experiment (resolution of 256 AAAA record queries) specified in milliseconds.

$$N = \frac{256\,\frac{query}{exp} * 1000\,\frac{ms}{s}}{T\,\frac{ms}{exp}} \qquad (1)$$

The results are convincing: MTD64 could reply 3886 AAAA record queries per second whereas BIND could do only 2094. For further results on performance comparison, please see [18].

VI. FUTURE PLANS

A. Detailed Performance Analysis

We plan to test MTD64 under heavy load conditions to investigate its stability, CPU and memory requirements and also to check if it complies with the graceful degradation principle [21]. We also plan to compare its performance to the before mentioned free DNS64 server programs, namely BIND, TOTD, Unbound and PowerDNS with a similar test method which was used for their performance analysis in [22], [23], and [24].

We are especially interested in how the extensive use of threading influences the memory consumption of the program.

We consider that our current performance results and the result of [18] partially justify our design decisions but we need to perform further tests, especially concerning the effects of a possible DoS (Denial of Service) attack, when the attacker sends needless AAAA record requests to exhaust the resources of the server.

B. Implementing Further Functions

We plan to implement recursion, caching and concurrent look-up of "AAAA" and "A" records, too. We plan to add these functions one by one and compare the performance of the new software to the original one to check whether the additional complexity required by these functions results in speed-up or slow-down of the software.

Our long term plans include the support of TCP as transport protocol for DNS messages and after its inclusion, it will be possible to add also DNSSEC [25].

EDNS(0) makes it possible to use larger than 512 bytes message size over UDP, see section 4.3 of [26]. We will consider implementing this feature.

The tiny size of the source code makes it possible to oversee the program as a whole and thus to change its behavior and add functions as we find the best.

C. Expecting Feedback from the Users

MTD64 was released as free software, sharing the source code and documentation on GitHub [17]. The program can be used, modified and redistributed under the GPLv2 license. We would like to warn our potential users that the software is not yet ready to be used in production systems, but it can be tested and/or further developed.

Any questions, comments, suggestions, experiences, test reports are welcome by the authors of this paper.

D. The Development of MTD64 is Kept Going

Dániel Bakai has taken over the further development of our DNS64 implementation in 2015. He made a fork and named the new version mtd64-ng. We plan to report his results soon.

VII. CONCLUSION

We have introduced all the necessary details about the DNS message format, the operation of the DNS64+NAT64 solution and the construction of IPv4-embedded IPv6 addresses.

We have disclosed our design principles for a high performance, easy to use and modify DNS64 server.

We have fully described our design decisions from the top level ones to the details.

We have summarized the most important implementation details in this paper and also published the source code and documentation of our multi-threaded DNS64 server (called MTD64) on GitHub as a free software under the GPLv2 License.

We have conducted a thorough functional testing and also checked the DNS message size issues.

During the basic performance testing, we have found that MTD64 significantly outperformed BIND when they were executed by a Raspberry Pi 2 Model B+ single-board computer.

Stability testing under heavy load conditions and a detailed performance analysis including comparison with several other free DNS64 implementations are planned future tasks.

We conclude that MTD64 may be useful also as a starting point for later development for anyone interested in.

ACKNOWLEDGEMENT

The development of the MTD64 server was the MSc thesis (final project) work of the second author at the Department of the Networked Systems and Services, Budapest University of Technology and Economics under the supervision of the first author.

REFERENCES

[1] The Number Resource Organization, “Free pool of IPv4 address space depleted” [Online]. Available: http://www.nro.net/news/ipv4-free-pool-depleted

[2] N. Skoberne, O. Maennel, I. Phillips, R. Bush, J. Zorz, and M. Ciglaric, “IPv4 address sharing mechanism classification and tradeoff analysis”, *IEEE/ACM Transactions on Networking*, vol. 22, no 2, April 2014, pp. 391–404.DOI: 10.1109/TNET.2013.2256147

[3] M. Bagnulo, A Sullivan, P. Matthews and I. Beijnum, “DNS64: DNS extensions for network address translation from IPv6 clients to IPv4 servers”, IETF, April 2011. ISSN: 2070-1721 (RFC 6147)

[4] M. Bagnulo, P. Matthews and I. Beijnum, “Stateful NAT64: Network address and protocol translation from IPv6 clients to IPv4 servers”, IETF, April 2011. ISSN: 2070-1721 (RFC 6146)

[5] M. Bagnulo, A. Garcia-Martinez and I. Van Beijnum, “The NAT64/DNS64 tool suite for IPv6 transition”, *IEEE Communications Magazine*, vol. 50, no. 7, July 2012, pp. 177–183. DOI: 10.1109/MCOM.2012.6231295

[6] C. Bao, C. Huitema, M. Bagnulo, M Boucadair and X. Li, “IPv6 addressing of IPv4/IPv6 translators”, IETF RFC 6052, 2010.

[7] Free Software Fundation, “The free software definition”, [Online]. Available: http://www.gnu.org/philosophy/free-sw.en.html

[8] Open Source Initiative, “The open source definition”, [Online]. Available: http://opensource.org/docs/osd

[9] F. W. Dillema, TOTD 1.5.3 source code, [Online]. Available: https://github.com/fwdillema/totd

[10] G. Lencse and A. G. Soós, “Design of a Tiny Multi-Threaded DNS64 Server”, in *Proc. 38th Internat. Conf. on Telecommunications and Signal Processing (TSP 2015)*, Prague, 2015, pp. 27–32. DOI: 10.1109/TSP.2015.7296218

[11] P. Mockapetris, “Domain names – implementation and specification”, IETF, November 1987. (RFC 1035)

[12] R. Elz and R. Bush, “Clarifications to the DNS Specification”, IETF, July 1997. (RFC 2181)

[13] N. Škoberne and M. Ciglarič, “Practical evaluation of stateful NAT64/DNS64 translation” *Advances in Electrical and Computer Engineering*, vol. 11, no. 3, August 2011, pp. 49–54. DOI: 10.4316/AECE.2011.03008

[14] V. Bajpai, N. Melnikov, A. Sehgal and J. Schönwälder, “Flow-based identification of failures caused by IPv6 transition mechanisms” in *Proc. 6th IFIP WG 6.6 Internat. Conf. on Autonomous Infrastructure, Management, and Security (AIMS 2012)*, Luxembourg, 2012, pp 139–150. DOI: 10.1007/978-3-642-30633-4_19

[15] S. Répás, T. Hajas and G. Lencse, “Application compatibility of the NAT64 IPv6 transition technology”, in *Proc. 37th Internat. Conf. on Telecommunications and Signal Processing (TSP 2014)*, Berlin, 2014, pp. 49–55. DOI: 10.1109/TSP.2015.7296383

[16] G. Lencse, “Investigation of event-set algorithms”, in *Proc. 9th European Simulation Multiconference (ESM'95)* Prague, 1995, pp. 821–825.

[17] A. Soós, “Multi-Threaded DNS64 server”, documentation and source code, [Online]. Available: https://github.com/Yoso89/MTD64

[18] G. Lencse, “Performance analysis of MTD64, our tiny multi-threaded DNS64 server implementation: Proof of concept”, review version available: http://www.hit.bme.hu/~lencse/publications/

[19] G. Lencse, “Test program for the performance analysis of DNS64 servers”, *Internat. J. of Advances in Telecomm., Electrotechnics, Signals and Systems*, vol. 4. no. 3. pp 60–65. Sep. 2015. DOI: 10.11601/ijates.v4i3.121

[20] G. Lencse, dns64perf source code, http://ipv6.tilb.sze.hu/dns64perf/

[21] NTIA ITS, “Definition of ‘graceful degradation’ ” [Online]. Available: http://www.its.bldrdoc.gov/fs-1037/dir-017/_2479.htm

[22] G. Lencse and S. Répás, “Performance analysis and comparison of different DNS64 implementations for Linux, OpenBSD and FreeBSD”, in *Proc. IEEE 27th Internat. Conf. on Advanced Information Networking and Applications (AINA 2013)*, Barcelona, 2013, pp. 877–884. DOI: 10.1109/AINA.2013.80

[23] G. Lencse and S. Répás, “Improving the Performance and Security of the TOTD DNS64 Implementation”, *Journal of Computer Science and Technology (JCS&T)*, ISSN: 1666-6038, vol. 14, no. 1, pp. 9–15. Apr. 2014.

[24] G. Lencse, S. Répás, “Performance analysis and comparison of four DNS64 implementations under different free operating systems”, *Telecommunication Systems*, in press, DOI: 10.1007/s11235-016-0142-x

[25] R. Arends, R. Austein, M. Larson, D. Massey, S. Rose, “DNS Security Introduction and Requirements”, IETF, March 2005. (RFC 4033)

[26] J. Damas, M. Graff, P. Vixie, “Extension Mechanisms for DNS (EDNS(0))”, IETF, April 2013. (RFC 6891)

Gábor Lencse received his MSc in electrical engineering and computer systems at the Technical University of Budapest in 1994, and his PhD in 2001.

He has been working for the Department of Telecommunications, Széchenyi István University in Győr since 1997. He teaches Computer networks, and the Linux operating system. Now, he is an Associate Professor. He is responsible for the specialization of the information and communication technology of the BSc level electrical engineering education. He is a founding and also core member of the Multidisciplinary Doctoral School of Engineering Sciences, Széchenyi István University. The area of his research includes discrete-event simulation methodology, performance analysis of computer networks and IPv6 transition technologies. He has been working part time for the Department of Networked Systems and Services, Budapest University of Technology and Economics (the former Technical University of Budapest) since 2005. There he teaches computer architectures and computer networks.

Dr. Lencse is a member of IEEE, IEEE Communications Society and IEICE (Institute of Electronics, Information and Communication Engineers, Japan).

András Gábor Soós received his Bsc and MSc in computer engineering from the Budapest University of Tecnology and Economics in 2013 and 2015, respectively. For his BSc thesis, he added DHCPv6 functionality to an open source GGSN software. For his MSc thesis, he has implemented a fully functional DNS64 server program called MTD64.

He worked for Telenor Global services as a Voice Core Network Operator Engineer providing second and third line support, being responsible for solving voice related issues in 2014 and 2015. Currently he is working at Avaya Global Support Services as a Backbone Engineer porividing third line enterprise support on Avaya Voice products.

APPENDIX: CONFIGURATION POSSIBILITIES OF MTD64

```
# Sample configuration file for MTD64, a tiny Multi-Threaded DNS64 server

// Uncomment the following line for name servers to be read from /etc/resolv.conf
#nameserver defaults

// Or you can add name servers manually
nameserver 8.8.8.8
nameserver 195.46.39.39

// Set DNS server selection mode
# selection-mode random // The given DNS servers will be used in random order
selection-mode round-robin // If a DNS server does not respond until timeout, the next one will be used

// Accepted IPv6 prefix length values are: 32, 40, 48, 56, 64, 96
dns64-prefix 2001:0db8:63a9:2ef5:dead:beef::/96

debugging yes  // Results in more verbose logging

# Sample settings for the timeout value of 1.35 sec
timeout-time-sec 1 // Maximum value is 32767
timeout-time-usec 350000 // Maximum value is 999999

# How many times will the DNS64 server try to resend a DNS query message if there is no answer
resend-attempts 2 // Maximum value is 32767

# This will set the maximum length of the IPv6 response message (UDP payload).
# Blocks which fall outside this value will be cut off.
# It is highly recommended not to change from 512 since it is the RFC standard.
# Some programs can accept UDP DNS response messages longer than 512 bytes.
# Note that only Answer, Authority, Additional blocks can be cut off.
# Queries block is going to be sent even if the message length is longer therewith
response-maxlength 512 // Accepted range for this setting is 0-32767
```

Presenting a Spatial-Geometric EEG Feature to Classify BMD and Schizophrenic Patients

Fatemeh Alimardani, Reza Boostani and Benjamin Blankertz

***Abstract*—Schizophrenia (SZ) and bipolar mood disorder (BMD) patients demonstrate some similar signs and symptoms; therefore, distinguishing those using qualitative criteria is not an easy task especially when these patients experience manic or hallucination phases. This study is aimed at classifying these patients by spatial analysis of their electroencephalogram (EEG) signals. In this way, 22-channels EEG signals were recorded from 52 patients (26 patients with SZ and 26 patients with BMD). No stimulus has been used during the signal recording in order to investigate whether background EEGs of these patients in the idle state contain discriminative information or not. The EEG signals of all channels were segmented into stationary intervals called "frame" and the covariance matrix of each frame is separately represented in manifold space. Exploiting Riemannian metrics in the manifold space, the classification of sample covariance matrices is carried out by a simple nearest neighbor classifier. To evaluate our method, leave one patient out cross validation approach has been used. The achieved results imply that the difference in the spatial information between the patients along with control subjects is meaningful. Nevertheless, to enhance the diagnosis rate, a new algorithm is introduced in the manifold space to select those frames which are less deviated around the mean as the most probable noise free frames. The classification accuracy is highly improved up to 98.95% compared to the conventional methods. The achieved result is promising and the computational complexity is also suitable for real time processing.**

***Keywords*—Bipolar Mood disorder, EEG classification, Noise Detection, Riemannian Geometric Mean, Schizophrenia, Weighting, Spatial topographic difference.**

I. INTRODUCTION

Diagnosing psychiatric disorders is very crucial because misdiagnosis leads to prescribe wrong medications and drives a patient into a worse situation. Schizophrenia (SZ) and bipolar mood disorder (BMD) are two of highly prevalent psychiatric diseases which show some similar clinical symptoms such as hallucination and delusion. Some researches have shown an acceptable (not promising) diagnosis rate in classification between these two groups of patients by analyzing their EEG features. In our previous work [9], we introduced a framework containing a feature extraction step followed by a feature selection phase and we have got the classification accuracy of 92.45% between the SZ and BMD patients. To the best of our knowledge there exists no more research to classify these groups of patients against each other so far. In a related work, Parvinnia et al. [10] improved the diagnosis accuracy between a group of SZ and control subject up to 95.32%. They first extracted several types of features from EEGs and then used an adaptive nearest neighbor to classify the features. Chun et al. [11] studied ERPs among schizophrenia, schizoaffective and bipolar (type-I) disorders and classified these groups up to 93.4% accuracy. Spatial resolution of their study is low. We aimed at this study to investigate EEGS in the spatial space. And we believe that spanning the inside information of one EEG signal to a higher dimensional space has this possibility to highlight the differences of various psychiatric groups better especially for SZ and BMD groups.

Generally, in the current literature on these two types of mental illnesses, first spectral based EEG features are extracted and then the classification process is executed. Recently, in the field of EEG-based brain computer interface classification, a new approach is presented in which the spatial information inside EEG is captured and used without these pre-processes [12-15]. According to promising performance of this novel framework we have motivated to investigate it for the problem of diagnosing between SZ and BMD groups. In aforementioned scenario, they have considered covariance matrices of each window of EEG (frame) as the descriptor of the brain state during the recording of that frame. In fact, instead of detecting event related synchronization/desynchronization (ERD/ERS) in ongoing EEG signals which is a known procedure in EEG-analysis applications [16], they spanned the covariance matrices of frames in the Riemannian space to make a topographical model of the brain state in the projected space.

Briefly, based on a non-Euclidean metric in the manifold space of covariance matrices, they developed a simple framework for the problem of single-trial EEG classification in a brain computer interface (BCI) application [13]. They also analyzed their method in other

Manuscript received November 22, 2015 and revised March 08, 2016.

F. Alimardani is the corresponding author and is with the Department of Computer Science and Engineering, Shiraz University, Shiraz, Iran (corresponding author to provide phone: 049-15737972997; e-mail: alimardani.f@ gmail.com). Also with Institute for Advanced Studies in Basic Sciences, GavaZang, Zanjan, Iran.

R. Boostani is with the Department of Computer Science and Engineering, Shiraz University, Shiraz, Iran (boostani@shirazu.ac.ir).

Present Address: Department of Computer Science and Engineering and Information Technology, School of Electrical and Computer Engineering, Zand Avenue, Shiraz, Iran, PO. BOX: 71348-51154, Tel/Fax: +98 711 6474605, Web: http://cse.shirazu.ac.ir

B. Blankertz is chair of the Neurotechnology group at Technische Universität Berlin in Berlin, Germany, heading the Berlin Brain-Computer Interfact (BBCI) project. email to benjamin.blankertz<AT>tu-berlin.de

.

doi: 10.11601/ijates.v5i2.143

applications such as: the event-related potential (ERP) and steady-state evoked potential (SSEP) based classifications, and led to achieve higher performance compared to state-of- the art approaches. This supremacy implies on the power of the manifold space in representing the EEG characteristics.

Here, first we demonstrate that spatial analysis of SZ versus BMD patients in the manifold space is informative and discriminative enough in order to setup a classification algorithm that we introduced in this paper. We also compare these spatial features for a group of control subjects versus SZ and BMD group. Secondly, in order to improve the accuracy of distinguishing between the patients a study has been done on noise detection and noise handling of EEG frames. As we know, noise and artifacts during EEG-recording decreases the performance of any EEG classification algorithm. And up to know, for the current geometric framework [15] in the field of EEG-analysis, no preprocessing is applied straightforward to eliminate noise; therefore, noise and artifacts are also projected into the manifold space. The goal of this study is to introduce a pre-processing step aimed at finding a confidence interval in the manifold space to mark a frame of EEG as the noise-free frame. Later, only these noise-free frames will participate in the classification phase where the geometrical features of the frames are utilized to classify the EEG signals of BMD patients from SZ ones.

In order to detect noisy frames, recently, the Riemannian geometric mean of frames has been used [15] in an online BCI system. They defined a Riemannian mean for the usual brain states and marked a frame as noise while it has a significant distance from the mean. We extended their idea in this paper. Since, the key point of the current analysis of EEGs in the manifold domain is the mean of covariance matrices, in this paper a weighted algorithm is introduced to define the mean-point more precisely.

For the classification step, one need to choose a suitable classifier in terms of good modeling of data characteristics additional to having a good accuracy level. Weighted nearest neighbor classifier has been appeared as an efficient method in machine learning problems where the distribution of classes are multi-modal [10], [17] meaning that samples are dispersed in the feature space. In these problems weighted nearest neighbor via its capability in capturing local information for data points produces more acceptable results while other classifiers which tuned a fix boundary in the form of linear or non-linear in their train phase, cannot make such a local and flexible decision. Here, by extending the nearest neighbor geometric framework to a weighted one, we achieved a more reliable classifier in order to improve the classification accuracy between the projected EEGs.

The rest of this paper is structured as follows. Section 2 presents the recorded data description. A short explanation of the current geometric framework in the literature is brought in Section 3. Section 4 shows that spatial geometric difference between SZ vs. BMD groups and also a control group (Ct) of normal participants is informative. Then Section 5 explains the contribution of this study in order to remove noisy frames geometrically, and our two proposed algorithms are described. Results are illustrated in Section 6 and the paper concludes in Section 7.

II. EEG DESCRIPTION

We recorded EEG signals from total 52 subjects consists of 26 patients with SZ and 26 patients with BMD. These subjects were selected from the pool of Pediatric Neurology outpatient Clinics of Hafez hospitals in Shiraz, Iran. They were diagnosed based on clinical and DSM-IV diagnostic criteria [1]. Patients were individually evaluated in the Clinical Neurophysiology Laboratory and EEG signals were recorded in three minutes length from 22 scalp electrodes (Fp1, Fp2, Fpz, F3, F4, F7, F8, FZ, C3, C4, CZ, T3, T4, T5, T6, P3, P4, PZ, O1, O2, A1, A2) with Scan-LT apparatus, according to the 10/20 international system, referred to linked A1+A2 electrodes. More details on subject specification are published in our previous paper [9]. In addition we also recorded EEG signals from a group of 26 age-matched healthy participants in the same idle condition. The criteria on selecting a control subjects was no experience of mental illnesses in their life. The EEG signals are passed through a Butterworth band pass filter (order 5) within the range of 8-30 Hz. Then, time regression method [2] is used to attenuate the artifact effect. Time frames of both lengths one/two seconds extracted from EEG-trials of each subject using successive rectangular windows with 50% overlap.

III. CURRENT RIEMANNIAN FRAMEWORK

Barachant et.al [12-15] described that covariance matrices of EEG trials carry high discriminative information [3]. They made a new set of data points to discriminate classes by considering covariance matrix of each time-frame as a sample. Let $X_i \in R^{n\times t}, i = 1,...,m$, be a time frame where n is the number of channels, t is the window length and m is the total number of frames; Then the covariance of X_i is:

$$C_i(X_i) = \frac{1}{t-1} X_i X_i^T. \qquad (1)$$

As we know from algebra, a set of positive definite covariance matrices $P(n) = \{C \in S(n), u^T Pu \succ 0, \forall u \in R^n\}$ follows a convex cone shape in the manifold space where $S(n)$ is the set of symmetric matrices on $R^{n\times n}$. Derivative at a point C (matrix) of $p(n)$ lies in the vector space T_C that called: tangent space at C. A local and smooth dot product on T_C is defined below:

$$\langle S_i, S_j \rangle_C = Tr(S_i C^{-1} S_j C^{-1}), \qquad (2)$$

where S_i and S_j are the projections of the C_i and C_j-two points of the manifold of $p(n)$- onto the T_C. This projection is done by using logarithmic map as an affine invariant metric on manifolds [4] and will induce the

Riemannian distance between two points on the manifold along with curve lines (geodesics) by:

$$\delta_r(C_i, C_j) = \left\| \log\left(C_i^{-1/2} C_j C_i^{-1/2}\right) \right\|_F . \qquad (3)$$

To discriminate the samples of different classes on the manifold of $P(n)$, most of the machine learning approaches needs to find the mean of samples in the projected space.

Exploiting the Riemannian definition of distance (3), the geometric mean of each class is the unique minima solution of the Karcher mean formula as:

$$\Theta(C_1,...,C_K) = \arg\min_{C\in P(n)} \sum_{i=1}^{K} \delta_r(C, C_i) , \qquad (4)$$

where $\Theta(.)$ is this minimum matrix and $\delta_r(.)$ is the geodesic value calculated by (3). The $\Theta(.)$ can be found by an iterative algorithm as described in [6]. Finding the geometric mean of each class, a nearest neighbor classifier as the lazy simplest learner has been used in [7] for the classification step and has shown acceptable results on single-trial EEG classification of BCI signals. Also, by considering the tangent space on the geometric mean of each class, to project all samples to the vector space, they used the commonplace classifiers such as linear discriminant on this space [12]. Reviewing the current Riemannian framework, leads us to the next sections that we will present our contributions to this framework.

IV. GEOMETRICAL DIFFERENCE BETWEEN SZ AND BMD GROUPS

As we mentioned in the introduction, we believe that spatial geometric analysis is promising in diagnosis between SZ and BMD. In order to verify our assumption, in this section, the aim is to reveal a topographic spatial difference between SZ and BMD groups also versus a group of control subjects. According to the methods that are described in the above section, geometric mean of the covariance matrices of a set of EEG time frames demonstrated a valuable feature in discriminating EEG signals up to now. In order to consider geometric means of SZ versus BMD group as the diagnosing criterion to distinguish between them; first one requires verifying that there is a spatial difference between the groups besides this difference does not come from the frontal part of the brain. Since we know SZ patients do more face contractions and eye movements rather than BMD subjects. In Fig.1 the topographic plot of geometric mean of the covariance matrices as the solution of the (4) is demonstrated for each electrode of our recording system. As we see, there is a meaningful difference in the left-parietal and occipital areas for the SZ group vs. BMD. This, support our claim that it is reasonable to do a spatial analysis. Furthermore, EEG signals from a group of 26 age-matched healthy participants are also analyzed and the topographic plot is shown the Fig.1 part (c). It can be seen that the interconnection pattern is different for the control group intuitively. In the lower part of the figure, we have plotted the difference between pair-wise groups. It is interesting that geometric difference of both groups with control group (Ct) is more located in the central and occipital parts, meaning that these areas are the most defected parts by the illness. Summarizing, the difference patterns appear to pinpoint a topographically relevant story. Left-parietal and occipital patterns have shown a good success in the literature for diagnosing these psychiatric illnesses and our study supports this pathophysiological implication as well.

V. NOISY FRAMES DETECTION BASED ON THE GEOMETRIC RIEMANNIAN MEAN

Artifact detection is critical in the problem of EEG classification. As described in section 2,in our experiment the patients were asked to sit and being relaxed (no mental task was needed) for three minutes. Here, an algorithm is introduced to detect and eliminate probable noisy frames trough studying the geometry of the covariance matrices of the EEG time frames.

Averaging is a traditional approach in the noise removal process [8]. Therefore, in the manifold of covariance matrices, the average across time frames for each subject is taken. Note that time frames are already zero-mean after passing through the Butterworth band pass filter in the pre-processing step. We first estimate the spatial-temporal property of the brain state for each subject with a multivariate Gaussian distribution as:

$$state_i \sim N(0, \Sigma_i) \quad i = 1,...,Tot , \qquad (5)$$

where Tot is the total number of subjects and Σ_i is the distribution of EEG time frames in the feature space. Based on (5), the only parameter which is needed to be determined per subject is Σ_i. Exploiting information geometry, we introduce to approximate Σ_i via the geometric mean of the covariance matrices of all time frames for the subject i^{th}. Let assume that there are *m* time frames per subject after windowing such as $X_k^i \quad k = 1,...,m$. Thus, the approximated geometric mean is:

$$\Sigma_i = \Theta(C_1,...C_m) , \qquad (6)$$

according to the (4) while each covariance matrix C_k, is found by (1). Instead of using all C_k s in the next step which is classification phase, our goal is to define a confidence interval for a time frame to be considered as a noise free frame. Our suggested algorithm in summarized below to calculate a marker value for each time frame.

1) Find geodesic distance between each time frame and Σ_i *:*

$$geodis(k) = \delta_r(C_k, \Sigma_i) \quad k = 1,...,m , \qquad (7)$$

2) Linear averaging of geodis

$$M_i = \frac{1}{m} \sum_k geodis(k) , \qquad (8)$$

3) Mark noisy frames

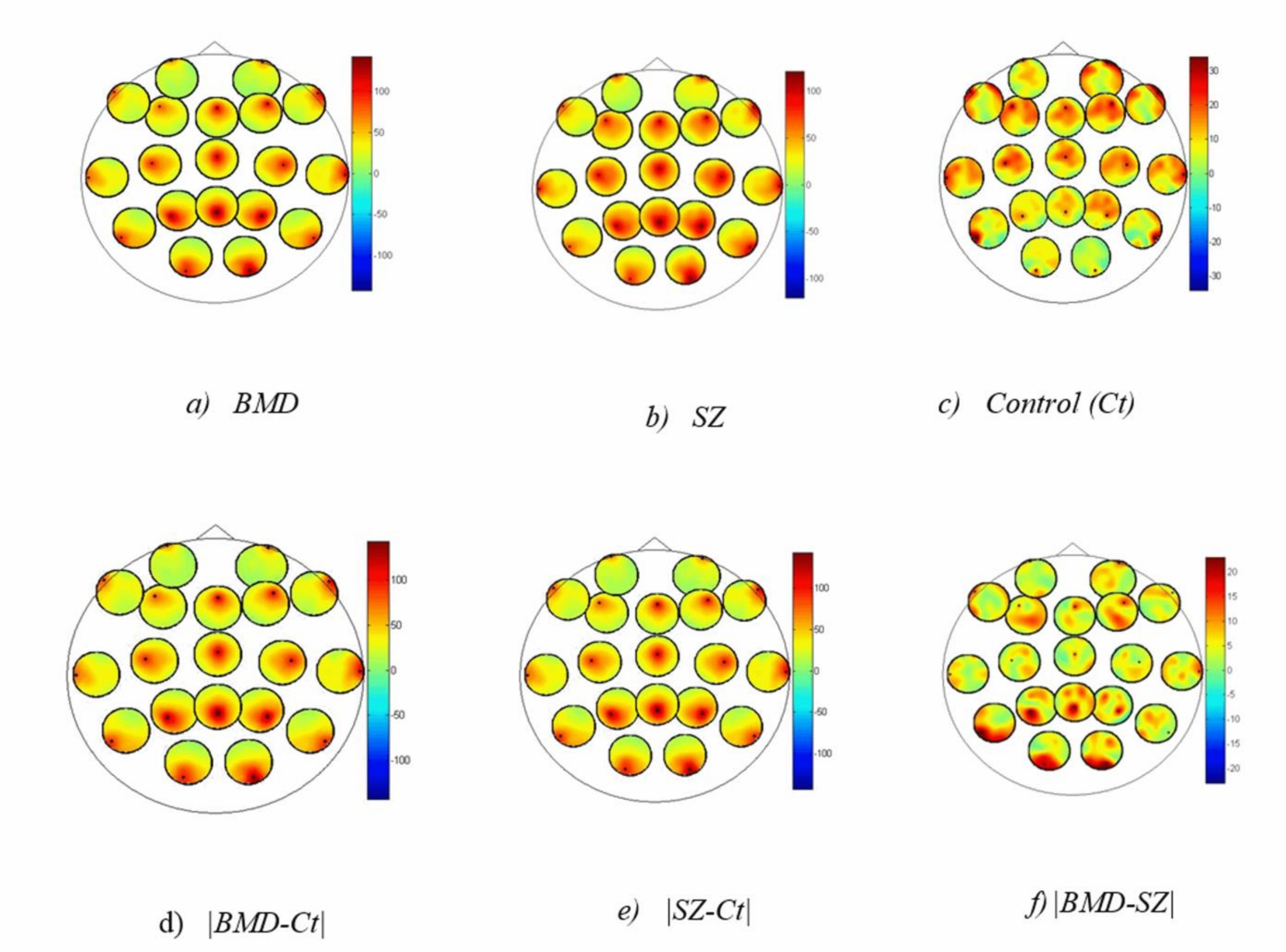
3-1) $$confInterval = M_i \pm 1.96\sigma_{geodis} \qquad (9)$$

3-2) If $geodis(k) \notin confInterval$ then

$mrk^i(k) = 0$ else $mrk^i(k) = 1$, for all *k*.

At the end, the multiplication of a time frame with its calculated marker value as: ($X_k^i * mrk^i(k)$) is given to the classifier.

This algorithm concludes in removing noisy frames with marker value equals to zero.



a) BMD

b) SZ

c) Control (Ct)

d) |BMD-Ct|

e) |SZ-Ct|

f) |BMD-SZ|

Fig 1 .Topographic plot of geometric mean of covariance matrices from EEG-frames for a) BMD, b) SZ and c) Ct group. Geometric mean difference of pair-wise groups is illustrated in d, e and f subplots. A left-parietal and occipital difference pattern for BMD vs. SZ subjects is obvious.

In the classification phase we have derived two different approaches. In both approaches, minimum distance to the geometric mean of classes will give the label for the test sample. First method, find the covariance points by applying (1) on $X_k^i * mrk^i(k)$ and then the geometric nearest neighbor classifier has been used. The scenario is described in the sub section *A* below. We called this method as Minimum Distance to Mean (MDM-NF) on Noise-Free frames.

In the second algorithm, a weighting mechanism is presented to find a balanced geometric mean according to the various amount of contribution for each frame to the geometric mean. The method is named: Weighted Minimum Distance to Mean (WMDM-NF) on Noise-Free frames and the weight is calculated as the inverse of the geodesic between a time frame and the mean that is calculated by MDM-NF algorithm.

In the nearest neighbor context, frames of all subjects belonging to the same group (label one/two) are used to calculate the mean matrix for each class. It means that cross-subject information has been summarized in the mean matrix. While the weight value for each time frame is estimated in such a way that it considers cross-frame information of each subject, separately. This conveys that the whole spatial information of data is combined inside the proposed weighted mean of the points in the space. We believe that this mean matrix is more accurate to be considered as a prototype for each group and consequently would result in a better classification. Our results in the next section support this claim.

A. *Minimum Distance to Mean (MDM-NF) on Noise-Free frames*

```
Input: $L^i, X_j^i \in R^{n\times t}$, $j = 1,...m$ for subject $i^{th}$ $i = 1,...,Tot$
As train frames belonging to two classes $L^i \in z = \{1, 2\}$ and
$X^{test}$ as test frame
Output: $Z^{test}$ label of test-frame
    1- For i=1:Tot
           $\forall k = 1,...,m \quad (frames)$
           a. Find $Y_k^i = X_k^i * mrk^i(k)$ based on (6)-(9)
           b. Find $C_k^i\left(Y_k^i\right)$ based on (1)
           End
    2- $Cov_z(1) = \Theta\left(C_k^i\right)$ $L^i \in \{z = 1\}$ based on (4)
    3- $Cov_z(2) = \Theta\left(C_k^i\right)$ $L^i \in \{z = 2\}$ based on (4)
    4- $L^{test} = \arg\min_z \delta_r\left(Cov_z(z), C^{test}\right)$ based on (3)
```

B. Weighted Minimum Distance to Mean (WMDM-NF) on Noise-Free frames

```
Input: $L^i, X_j^i \in R^{n\times t}$, $j=1,...m$ for subject $i^{th}$ $i=1,...,Tot$
As train frames belonging to two classes $L^i \in z=\{1,2\}$ and $X^{test}$ as test frame
Output: $Z^{test}$ label of test-frame
    1- For $i$=1:$Tot$
         $\forall k=1,...,m$ $(frames)$
         a. Find $Y_k^i = X_k^i * mrk^i(k)$ based on (6)-(9)
         b. Find $C_k^i\left(Y_k^i\right)$ based on (1)
         c. $weight^i(k)=\frac{1}{geodis(k)}$ based on (7)
         End
    2- $Cov_z(1)=\Theta\left(weight^i(k)*C_k^i\right)$ $L^i \in \{z=1\}$ based on (4)
    3- $Cov_z(2)=\Theta\left(weight^i(k)*C_k^i\right)$ $L^i \in \{z=2\}$ based on (4)
    4- $L^{test}=\arg\min_z \delta_r\left(Cov_z(z),C^{test}\right)$ based on (3)
```

VI. Results and Discussion

As mentioned above, one trial per subject is recorded and a set of time frames is generated for each subject using a window of the window size equals to one second with the purpose of preserving stationary assumption. Moreover, since in the most EEG applications, brain state is analyzed in time frames having two seconds length, our results are examined for this window size as well. Table. I presents the outcomes of the competitive approaches with window size equals to one second and Table.II, shows the accuracy for the window with length two. In general, it is hard for the subjects to get used to the conditions at the beginning of the session and near to the end of recording, they become impatient or exhausted a little bit. It means that the probability for a time frame to be contaminated with the movement, fatigue and other kinds of noise, is higher in the early time frames same as lately ones. Therefore, the second minute of recorded trials has been assessed as processing time-interval in addition to considering the whole session. But, this is a heuristic assumption and the amount of concentrating is a subject dependent value. To have a more robust result and remove dependency between train and test samples, leave-one patient-out cross validation (LOOCV) is implemented. The average accuracy of all subjects is reported. The accuracy rate is brought for two versions of WMDM method. In the standard WMDM-NF, weighting step is employed on time frames before covariance estimation. In our experiment we also investigate another version of the weighting approach entitled as WMDM-NF2 in which the weighting has been done after calculation of the covariance matrices. Consequently, the effect of weighting is evaluated in both temporal and spatial level. It can be seen that the performance is higher in the spatial-domain up to 98.95 %. The reason is that since the underlying processing domain is spatial the correspondence is much better if the weight function is applied to the of covariance matrices. Besides, since the weight is defined based on the characteristic of the covariance distribution, it is more reasonable to apply it on the covariance matrices.

TABLE I.
LOOCV-accuracy rate with window size equal to one second

Method	Just 2nd minute	2nd minute-end	Whole session
WMDM-NF	94.95±2.7	93.55±4.33	**97.56±2.41**
WMDM-NF2	96.06±3.66	96.25±3.02	**98.95±1.09**
MDM-NF	94.36±3.67	93.97±2.56	**95.85±2.95**

TABLE II.
LOOCV-accuracy rate with window size equal to two seconds

Method	Just 2nd minute	2nd minute-end	Whole session
WMDM-NF	95.42±2.28	97.97±2.9	**98.21±1.28**
WMDM-NF2	96.12±3.01	**98.75±1.06**	98.23±1.28
MDM-NF	94.32±2.5	97.72±2.09	**97.78±2.3**

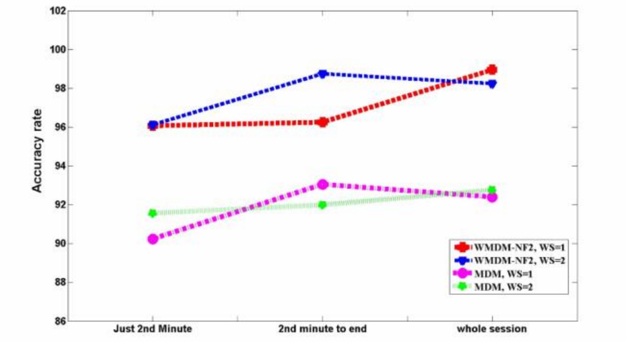


Fig. 2. . Comparison of reference method MDM [12] and our approach WMDM-NF2 in terms of classification accuracy. Sliding window is used to make epochs with both window sizes (WS) equals to one second and two seconds while various intervals of recorded data is considered.

Also it has been shown in these tables that considering the whole session gives better diagnosis because using the proposed noise-frame removal strategy; one can be relief of having extra noisy frames in the early and lately stages. Then we benefit from a general approach.

To show the effect of noise-frames elimination in comparison with the reference geometric method (MDM) in the literature, Fig. 2 Is plotted. This figure collates the accuracy of WMDM-NF2 as our best proposed algorithm applied to the various time intervals with the reference approach. The supremacy of the proposed method manifests its ability in estimating a more precise mean value geometrically. Also Fig. 3 exhibits the distribution of covariance matrices along four randomly chosen electrodes: C3, T4, T6, and O1. It depicts spatial difference of two classes before and after noise removal in the first and the second row, respectively.

As we can see, implementing introduced algorithm concludes in a more dense conic space which is more suitable to be used in the classification phase.

In another experiment, to show that taking the advantages of geometric information outperforms the standard linear averaging, the correctness of the suggested algorithms using Euclidean mean and also Euclidean metric in measuring distances employed on the whole session is displayed in Fig.4. The lack of capability of linear averaging in capturing the whole information is obvious in

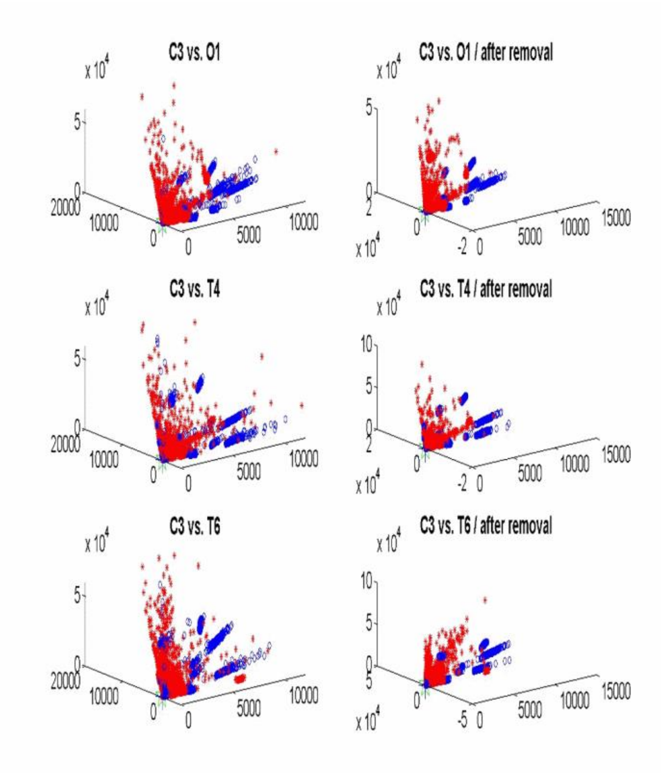


Fig. 3. The distribution of covariance matrices along piecewise electrodes [C3, O1, T4 and T6] in each subplot. Left column shows how frames scatter originally in the space and in the right column data distribution has been shown after eliminating noisy frames applying WMDM-NF.

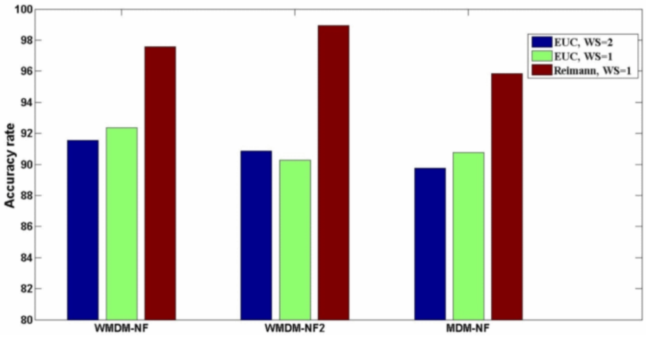
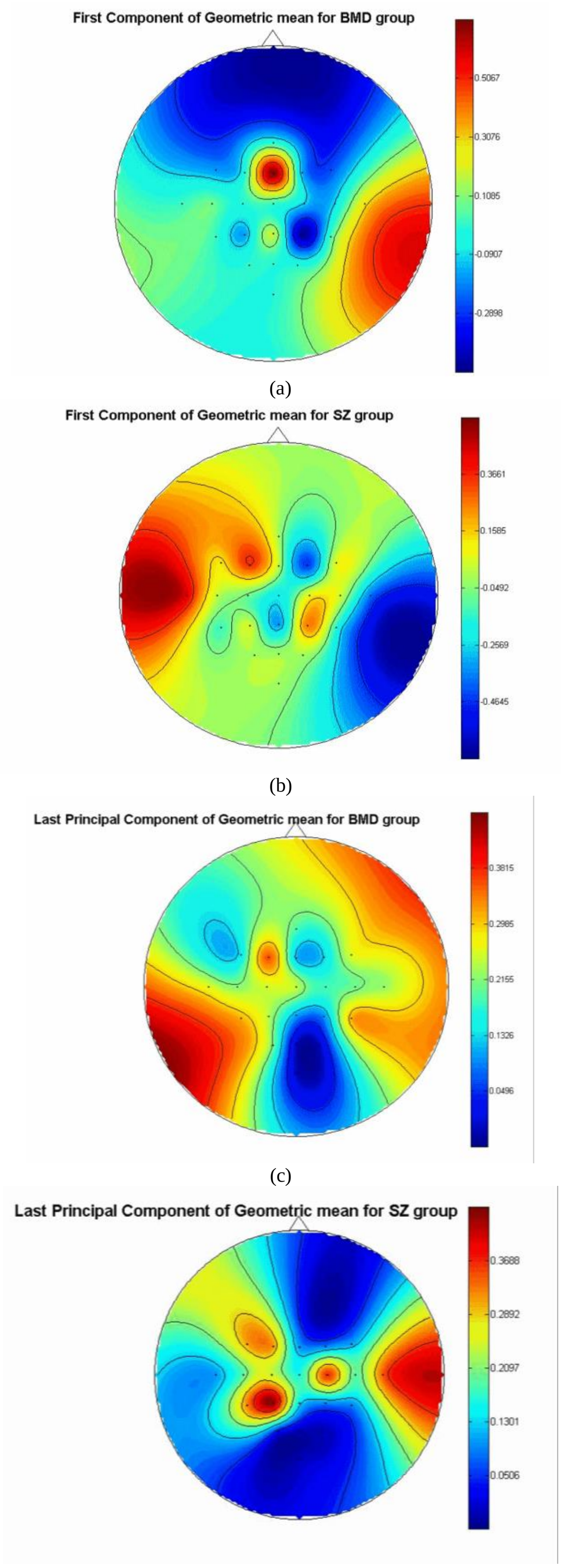


Fig. 4. Accuracy rate of classification using Euclidean metric/Euclidean mean (EUC) in proposed methods in both window sizes (WS), compared to Riemannian metric/ Riemannian mean. Here the best result according to window size in Riemannian analysis (WS=1) is assessed.

comparison with using geometric mean based on the geodesics.

In addition to above results that analyzes the performance of our algorithm in terms of accuracy, we also investigated the spatial pattern of our proposed geometric mean via topographic map to verify the potential of the $C_{WMDM-NF}$ as the definition of the mean of matrices in distinguishing among SZ and BMD group more intuitively. In this way, the eigenvector or principal component corresponding to the largest eigenvalue of $C_{WMDM-NF}$ for each of the groups is visualized as scalp topography, in the Fig.5. part a) and b). The ability of first component in discriminating between classes can be seen intuitively in this figure via looking at the diversity of the two topographies in the first two rows. It shows that the definition of the mean matrix $C_{WMDM-NF}$ is strong enough


First Component of Geometric mean for BMD group
0.5067
0.3076
0.1085
-0.0907
-0.2898
First Component of Geometric mean for SZ group
0.3661
0.1585
-0.0492
-0.2569
-0.4645
Last Principal Component of Geometric mean for BMD group
0.3815
0.2985
0.2155
0.1326
0.0496
Last Principal Component of Geometric mean for SZ group
0.3688
0.2892
0.2097
0.1301
0.0506

(a)

(b)

(c)

(d)

Fig. 5. This figure verifies that the proposed geometric mean of covariance matrices is discriminative via generating diverse pattern over the head. The eigenvectors or principal components corresponding to the largest eigenvalue of proposed $C_{WMDM-NF}$ are visualized as scalp topographies for BMD and SZ group in a) and b) respectively. The dissimilarity of the corresponding maps for the two classes can be seen. In c) and d) principal component corresponding to the smallest eigenvalue is brought for BMD and SZ respectively. Existence of the diversity even for the smallest eigenvalue verifies the meaningful usage of geometric mean.

to capture the information that we need in order to differentiate between the mental states of a patient from SZ/BMD group in a simple manner. It can be easily seen that the first principal component of this mean matrix differentiates active brain parts. For the BMD group right temporal-parietal areas are the most active parts, while for the SZ patients left temporal-frontal parts show stronger activity. Moreover, in the theory of covariance matrices principal components corresponding to the smaller eigenvalues are also of interest. The pattern of the last principal component of geometric mean of the BMD and SZ group is illustrated in Fig.5 parts c) and d). A large diversity between the groups for even the last component is obvious. This shows that our proposed spatial algorithm is a useful tool in the application that we analyzed.

Last but not the least, time complexity of applying the whole procedure on a test subject is just 543.57 seconds using Matlab R2008b on a device with CPU: Intel, 2 Duo 2.53 GHz, 4 GB internal memory RAM and Windows 8 pro. And it covers the claim of being suitable for online problems.

VII. Conclusion

This research presents a new algorithm to improve the classification of BMD from schizophrenic patients using their spatial geometry features. It relies on the covariance matrices as the descriptor of the brain state during the EEG recording. Two methods based on Riemannian geometry have been proposed which employed directly to the covariance matrices in order to reduce the effect of noise. The first method, named MDM-NF, is a modification of the Minimum Distance to Mean (MDM) algorithm proposed in [12] by defining a confidence interval for a time frame to be considered as noise free. This interval calculates based on geometry of second order statistic of EEG frames and is simple and effective. The second method, named WMDM-NF, is a weighted version of the previous algorithm in which after the elimination of noisy time frames from our set of time frames, a weight is adjusted for the remained frames to control their influence on the classifier's boundary for classification of a test frame. We applied weights to both time and spatial domains, termed as WMDM-NF and WMDM-NF2, respectively. Significant better results have been achieved by applying both proposed weighting approaches compared to the reference geometric framework (MDM). This improvement is mainly due to capability of our method in the handling of noisy frames. To show that benefitting manifold structure of covariance matrices will give more accuracy than using the standard linear averaging based on Euclidean metric, the results are also compared.

Although the framework is simple and it does not have any parameter to tune, its performance is satisfactory. Here the suggested methods are employed in diagnosing BMD patients from SZ; however, the whole scenario is applicable in other diagnostic procedure too. The proposed approach is promising according to its successful outcomes compared to the conventional methods.

References

[1] American Psychiatric Association, 1994. *Diagnostic and Statistical Manual of Mental Disorders*, 4th ed. American Psychiatric Association, Washington DC.

[2] P. He, G. Wilson, C. Russell, M. Gerschutz, "Removal of ocular artifacts from the EEG: a comparison between time-domain regression and adaptive filtering method using simulated data", *Med. Biol. Eng. Comp*. vol. 45, 2007, pp. 495–503.

[3] A. Barachant, S. Bonnet, M. Congedo, C. Jutten, "Classification of covariance matrices using a Riemannian-based kernel for bci applications," *Neurocomputing*, vol. 112, , 2013, pp. 172 – 178.

[4] X. Pennec, P. Fillard, N. Ayache, "A Riemannian framework for tensor computing", *Int. J. Comput. Vis*., vol. 66, no. 1, 2006, pp. 41–66.

[5] A. Fuster, A. Tristan-Vega, TD. Haije, CF. Westin, L. Florack, "A Novel Riemannian Metric for Geodesic Tractography in DTI", *in Computational Diffusion MRI and Brain Connectivity Mathematics and Visualization*, 2014, pp. 97-104.

[6] J. H. Manton, "A globally convergent numerical algorithm for computing the centre of mass on compact Lie groups", *in Proc. of ICARCV*, 2004, pp. 2211-2216.

[7] A. Barachant, S. Bonnet, M. Congedo, C. Jutten, "Common spatial pattern revisited by Riemannian geometry", *In IEEE International Workshop on Multimedia Signal Processing (MMSP)*, 2010, pp. 472-476.

[8] M. Marx, KB. Pauly, C. Chang, "A novel approach for global noise reduction in resting-state fMRI: APPLECOR", Neuroimage, vol. 64, 2013, pp. 19–31.

[9] F. Alimardani, R. Boostani, M. Azadehdel, A. Ghanizadeh, K. Rastegar, "Presenting a new search strategy to select synchronization values for classifying bipolar mood disorders from schizophrenic patients", *Engineering Applications of Artificial Intelligence*, vol. 26, no. 2, 2013, pp. 913–923.

[10] E. Parvinnia, M. Sabeti, M. Zolghadri Jahromi, R. Boostani, "Classification of EEG Signals using adaptive weighted distance nearest neighbor algorithm", *Journal of King Saud University – Computer and Information Sciences*, vol.26, 2014, pp. 1–6 .

[11] J. Chun, Z. N. Karam, F. Marzinzik, M. Kamali, L. O'Donnell, I .F. Tso, T. C. Manschreckd M. McInnis, P.J. Deldin, "Can P300 distinguish among schizophrenia, schizoaffective and bipolar I disorders? An ERP study of response inhibition", *Schizophrenia Research*, vol. 151, no. 1–3, December 2013, pp. 175–184.

[12] A. Barachant, S. Bonnet, M. Congedo, C. Jutten, "Riemannian geometry applied to BCI classification", *in Proc. 9th International Conference Latent Variable Analysis and Signal Separation (LVA/ICA)*,vol. 6365, Saint Malo-France, 2010, pp. 629-636.

[13] A. Barachant, S. Bonnet, M. Congedo, C. Jutten, "Multi-class brain computer interface classification by Riemannian geometry", *IEEE Transactions on Biomedical Engineering*, vol. 59, no. 4, 2012, pp. 920-928.

[14] F. Yger, "A review of kernels on covariance matrices for BCI applications", *IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, Sep 2013, pp. 1-6.

[15] A. Barachant, A. Andreev, M. Congedo, "The Riemannian Potato: an automatic and adaptive artifact detection method for online experiments using Riemannian geometry", *TOBI Workshop lV*, Sion- Switzerland, 2013.

[16] G. Pfurtscheller, C. Neuper, "Future prospects of ERD/ERS in the context of brain-computer interface (BCI) developments", *Progress in Brain Research*, vol. 159, 2006, pp. 433–437.

[17] F. Alimardani, R. Boostani, E. Ansari, "Feature selection SDA method in ensemble nearest neighbor classifier", *In Proc. of the Springer, 13th International Conference of Computer Science and Engineering*, Kish-Iran, March 2008, pp. 9–11.

Implementation of IEEE 802.15.4a Based UWB Systems for Coexistence with Primary Users

Çağlar Fındıklı, Serhat Erküçük, Mehmet Ertuğrul Çelebi

***Abstract*—Peaceful coexistence is a major implementation issue for both cognitive radios and ultra wideband (UWB) systems. Accordingly, the UWB impulse radio (UWB-IR) based Wireless Personal Area Network (WPAN) standard IEEE 802.15.4a has suggested using linear combination of pulse to limit interference to coexisting primary systems. In this paper, motivated by implementing the IEEE 802.15.4a based UWB-IR systems for peaceful coexistence, we consider the implementation of linear combination of pulses as suggested by the standard. Accordingly, we (i) design possible linearly combined pulses that conform to the standard requirements, (ii) consider coherent and noncoherent receiver structures that can be adapted for the physical layer of the IEEE 802.15.4a standard, (iii) investigate the effect of channel models on the system performance, and (iv) study the UWB-IR system performance in the presence of narrowband and orthogonal frequency division multiplexing (OFDM) based wideband primary systems with various bandwidths and subcarriers. The study shows that the UWB-IR system performance can be significantly improved by selecting suitable pulses for transmission and employing appropriate filtering techniques at the receiver when the primary system is active. For the implementation of IEEE 802.15.4a based UWB systems complying with coexistence requirements, the results of this study should be carefully considered.**

***Keywords*—Ultra wideband (UWB) systems, cognitive radios, coexistence, spectrum shaping, pulse design.**

I. Introduction

Cognitive radios [1] and ultra wideband (UWB) systems [2] have emerged as alternative technologies for efficient utilization of the spectrum. Both technologies are referred to as *secondary systems*, where they have to coexist with licensed (i.e., primary) systems without causing interference to them. While cognitive radios have to assess the availability of the spectrum by means of spectrum sensing [3]–[5] and use the frequency band only if the primary user is not active, UWB systems have to transmit with a low power spectral density (PSD) [6] to limit the interference level to coexisting primary systems. Despite UWB systems being perceived as underlay systems below noise spectrum level, many regulatory agencies worldwide have been cautious and have mandated detect-and-avoid (DAA) techniques in some bands to limit the interference level [7]. In the implementation of DAA, the UWB system initially has to perform spectrum sensing to determine whether the coexisting primary systems are active or not [8], [9]. As the next step, the UWB system either has to keep silent or lower its PSD level at the frequency band of the primary system for avoidance.

Due to the ultra-wide bandwidth nature (at least 500 MHz) of UWB systems, there may be multiple primary systems overlapping with the frequency band of a UWB system. On the other hand, one or more coexisting primary systems may be frequently active. Hence, keeping silent after the detection of one or more primary systems may cause limited operation capability for UWB systems. Alternatively, spectrum shaping and pulse design techniques have been widely considered in the literature. By utilizing the desired spectrum mask and reducing the PSD level at the desired frequency band, new pulses have been designed [10]–[12]. The common characteristic of these pulses is that the available spectrum is optimized with no restriction on the number of filter coefficients. However, UWB based systems are commonly accepted as low-cost systems with simple transceiver structures. Accordingly, the UWB impulse radio (UWB-IR) based Wireless Personal Area Network (WPAN) standard IEEE 802.15.4a [13] has suggested using linear combination of a few pulses, which is equivalent to using a few filter coefficients, for spectrum shaping purposes. That is, in the presence of an active primary system, the transceiver should generate a new pulse shape based on the aggregation of time-shifted and scaled versions of the original pulse. In [14], the authors have addressed generating pulses with notches at the desired frequencies by conforming to the restrictions in the standard based on a z-transform approach. However, the performance of the designed pulses has not been studied in the presence of an active primary system neither in [14] nor in optimum pulse design studies [10]–[12].

Parallel to pulse design techniques, the effects of narrowband and wideband licensed systems (also referred to as “interference” from the UWB communications perspective) on the UWB system performance have been studied [15]–[19]. In [15], jam resistance of UWB systems has been investigated for interferences with various bandwidths. In [16], the effects of GSM900, UMTS and GPS systems on the UWB system performance (and vice versa) have been studied. The authors have evaluated the performance of UWB systems employing differential-Rake (D-Rake) receivers in the presence of narrowband interference in [17]. In [18], an exact analysis has been derived for precisely calculating the bit error probability of a UWB-IR communication system coexisting with an IEEE 802.11a system. In [19], the performance of a

Ç. Fındıklı and M. E. Çelebi are with the Department of Electronics and Communications Engineering, Istanbul Technical University, Maslak, Istanbul, 34469 Turkey.

S. Erküçük is with the Department of Electrical and Electronics Engineering, Kadir Has University, Fatih, Istanbul, 34083 Turkey. E-mail: serkucuk@khas.edu.tr.

Manuscript received November 13, 2015; revised March 18, 2016; accepted March 21, 2016. This paper was presented in part at the 37th International Conference on Telecommunications and Signal Processing, Berlin, Germany, July 2014.

doi: 10.11601/ijates.v5i2.134

Multiband Orthogonal Frequency-Division Multiplexing (MB-OFDM) based UWB system has been analyzed in the presence of an IEEE 802.16 WiMAX system. The common approach in these studies is that the UWB systems employ pulses that do not take into account the interference level caused by UWB systems to the licensed systems. However, as mandated by the European and Japanese regulatory agencies, the UWB systems should transmit pulses with reduced power levels at the frequency bands occupied by licensed systems.

Motivated by implementing the IEEE 802.15.4a based UWB-IR systems for peaceful coexistence, we consider the implementation of linear combination of pulses as suggested by the standard. In [20], we have considered the presence of a narrowband interference and used only one type linearly combined pulse to assess the UWB-IR system performance. In this comprehensive study, we consider both narrowband and wideband primary users and employ all possible linearly combined pulses to determine which type of pulse may be preferred for implementation. Accordingly, we can summarize the contribution of the current study as follows. We:
(i) design possible linearly combined pulses that conform to the standard requirements,
(ii) consider coherent and noncoherent receiver structures that can be adapted for the physical layer of the IEEE 802.15.4a standard,
(iii) investigate the effect of channel models on the system performance, and
(iv) study the UWB-IR system performance in the presence of narrowband and OFDM based wideband primary systems with various bandwidths and subcarriers.

For the UWB-IR system performance, many practical scenarios such as the effects of transmitter-receiver structures, interference level, interference types, pulse types and the IEEE 802.15.4a channel models are investigated in detail. The results of this study are important as they demonstrate an alternative implementation of the IEEE 802.15.4a system complying with the regulatory agency mandates for coexistence, and yet achieving a reasonable system performance.

The rest of the paper is organized as follows. In Section II, IEEE 802.15.4a system model and the associated receiver structures are presented. In Section III, a modified transceiver structure that employs linear combination of pulses for coexistence is presented. In Section IV, simulation results are presented in order to assess the UWB system performance in the presence of narrowband and wideband interferences for various practical scenarios. Concluding remarks are given in Section V.

II. IEEE 802.15.4a System Model

In this section, the system model of the IEEE 802.15.4a based UWB impulse radios [13] is presented that can support both coherent and noncoherent data reception as given in [20]. The IEEE 802.15.4a standard uses combined binary phase shift keying (BPSK) / binary pulse position modulation (BPPM) for data transmission. While both the phase and position information can be detected by coherent detection, only the position information can be detected by noncoherent detection. The system model that supports both coherent and noncoherent data reception is explained as follows.

For reliable communications in a dense multipath environment, data transmission is achieved by burst of pulses, where each of the N_b consecutive pulses are transmitted within a chip time T_c and $T_b = N_b T_c$ is the burst duration. The symbol time $T_s = N_c T_c$, where N_c is the number of chips in a symbol, is much greater than the burst duration T_b ($T_s >> T_b$) in order to allow time hopping (TH) for multiple access (MA) and accommodate guard times to prevent intra- and inter-symbol interferences. With this symbol structure, the l^{th} symbol of the 1^{st} user that carries the position and phase information can be transmitted using the signal model

$$w_l^{(1)}(t)=\sum_{j=0}^{N_b-1} a_l^{(1)} s_j^{(1)} p\Big(t - lT_s - jT_c - d_l^{(1)}\delta_p - c_l^{(1)} T_b\Big) \quad (1)$$

where $w_l^{(1)}(t)$ is the waveform of the 1^{st} user's l^{th} transmitted symbol consisting of N_b consecutive pulses, $p(t)$ is the transmitted pulse with duration $T_p \leq T_c$, and $s_j^{(1)} \in \{\pm 1\}$ $\{j = 0, 1, \ldots, N_b - 1\}$ is a scrambling sequence specific to user-1 that is used to smooth the spectrum. $a_l^{(1)} \in \{\pm 1\}$ is the user phase information and can only be seen by the coherent receiver, whereas $d_l^{(1)} \in \{0, 1\}$ carries the user position information that can be seen by both coherent and noncoherent receivers, where $\delta_p = T_s/2$ is the position shift parameter. Accordingly, this combined modulation is regarded as BPSK/BPPM. $\{c_l^{(1)}\}$ are the TH integer values that scramble the position of the burst for multiuser interference suppression. The condition $c_{max} T_b + T_d \leq \delta_p$ should be satisfied in order to prevent inter-symbol interference, where c_{max} is the maximum TH shift integer value and T_d is the maximum channel delay spread.

In order to prevent inter-pulse interference and to specifically evaluate the effect of linear combination of pulses, we assume a single user scenario with a single pulse transmitted (i.e., $N_b = 1$) without loss of generality. Thus, the transmitted signal can be simplified to

$$w_l^{(1)}(t) = a_l^{(1)} p\Big(t - lT_s - d_l^{(1)}\delta_p\Big)\,. \quad (2)$$

In the presence of an active primary system, the received signal can be modelled as

$$r(t) = \tilde{w}_l^{(1)}(t) + J(t) + n(t) \quad (3)$$

where $\tilde{w}_l^{(1)}(t)$ is the received waveform of the 1^{st} user's l^{th} symbol, $J(t)$ represents the coexisting primary user, and $n(t)$ is the additive white Gaussian noise (AWGN) with two-sided power spectral density $N_0/2$. In fact, the transmitted pulse $p(t)$ and the primary signal $J(t)$ may be overlapping in the same frequency band. If the PSD level of the transmitted pulse is not low enough at the operating frequency of the primary signal, the regulatory agencies may not permit this pulse transmission. On the other hand, if both the UWB system and the primary system transmit simultaneously, performances of both systems are expected to degrade. In Section III, we will consider a modified transceiver structure to allow for coexistence.

The signal $\tilde{w}_l^{(1)}(t)$ in (3) is the waveform distorted by the channel $h(t)$ and is represented as

$$\tilde{w}_l^{(1)}(t) = w_l^{(1)}(t) * h(t) \tag{4}$$

where $*$ is the convolution operator. The equivalent channel model $h(t)$ can be given as

$$h(t) = \sum_{i=0}^{L-1} h_i \delta(t - \tau_i) \tag{5}$$

where h_i is the i^{th} multipath channel coefficient, τ_i is the delay of the i^{th} multipath component and $\delta(\cdot)$ is the Dirac delta function. Consistent with the earlier studies, it is assumed that the channel coefficients are normalized, i.e., $h(t) = \sum_{i=0}^{L-1} h_i^2 = 1$ to remove the path loss effect, and that the delays $\{\tau_i\}$ occur at integer multiples of the chip time T_c.

At the receiver, the information of user-1 transmitted by BPSK/BPPM can be detected either coherently or noncoherently.

A. *Coherent receiver*

The coherent receiver is a Rake receiver implemented using the delayed versions of the reference signal [2]. The output of the correlator corresponding to the i^{th} finger of the Rake receiver for the m^{th} PPM position can be given by

$$\begin{aligned} D_{i,m}^{(1)} &= \int_{-\infty}^{\infty} r(t) v_m(t - \tau_i) dt \\ &= \int_{-\infty}^{\infty} \Big(\tilde{w}_l^{(1)}(t) + J(t) + n(t)\Big) v_m(t - \tau_i) dt \end{aligned} \tag{6}$$

$i = 0, \ldots, L_0 - 1$ for $m \in \{0, 1\}$, where

$$v_m(t) = p(t - lT_s - m\delta_p) \tag{7}$$

is the reference signal and L_0 is the number of Rake fingers used. Assuming that the channel parameters can be predicted, a maximal-ratio combiner is used to combine the Rake receiver outputs as

$$D_m^{(1)} = \sum_{i=0}^{L_0-1} h_i D_{i,m}^{(1)} \tag{8}$$

to form the decision variables. Since $\{D_m^{(1)}\}$ carries the phase information as well, the data is recovered as

$$\begin{aligned} \max\Big\{|D_m^{(1)}|\Big\} = D_{d_l'}^{(1)} \quad &\Rightarrow \quad d_l' \\ \text{sign}\Big\{D_{d_l'}^{(1)}\Big\} \quad &\Rightarrow \quad a_l' \end{aligned} \tag{9}$$

where $|x|$ and sign$\{x\}$ denote the absolute value and the sign of x, respectively.

B. *Noncoherent receiver*

The noncoherent receiver is an energy detector with the decision variables $\{D_m^{(1)}\}$, where

$$\begin{aligned} D_m^{(1)} &= \int_{m\delta_p}^{m\delta_p + T_i} r^2(t) dt \\ &= \int_{m\delta_p}^{m\delta_p + T_i} \Big(\tilde{w}_l^{(1)}(t) + J(t) + n(t)\Big)^2 dt \end{aligned} \tag{10}$$

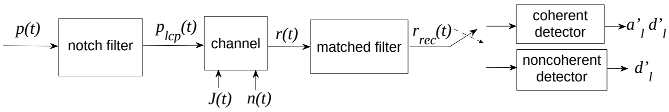


Fig. 1. Block diagram of the modified transceiver structure.

with $m \in \{0, 1\}$, which integrates the received signal energy for the duration of T_i. The position information is recovered by finding the maximum decision variable as

$$\max\{D_m^{(1)}\} = D_{d_l'}^{(1)} \Rightarrow d_l' \ . \tag{11}$$

III. Modified Transceiver Structure

In the case of an active primary system sharing the same frequency band, the UWB system has to take an action. The UWB system can either keep silent or use pulses that have low PSD level at the primary systems' frequency bands. If the primary system is active most of the time, keeping silent may decrease the operation time of UWB systems significantly. Hence, we will consider the implementation of the linear combination of pulses as suggested by the IEEE 802.15.4a standard to reduce the power level at the desired frequency band of a primary system. While this will allow the UWB system to transmit simultaneously with the primary system, we will consider a front-end filter matched to the linearly combined pulse at the receiver before coherent or noncoherent receiver processing. The modified transceiver structure is shown in Fig. 1.

In the following, we will present the primary user signal model, how the linear combination of pulses can be implemented, and how the matched filtering can be employed.

A. *Primary user signal model*

The primary user signal is considered to be either a narrowband or a wideband signal.

Narrowband signal: If the primary system is a narrowband system, then the signal is modelled as a single tone narrowband interference [17]

$$J(t) = \sqrt{2J_0} cos(2\pi f_j t + \theta_j) \tag{12}$$

with average power J_0, carrier frequency f_j and random phase θ_j uniformly distributed over $[0, 2\pi)$.

Wideband signal: For the wideband signal, we consider an OFDM signal of the form [18]

$$J(t) = \sqrt{\frac{2J_0}{N_s}} \sum_{n=0}^{N_s-1} b_n e^{j2\pi(n\Delta_f + f_j)t} \tag{13}$$

where N_s is the number of subcarriers, Δ_f represents the subcarrier frequency spacing, and b_n is the transmitted OFDM data symbol.

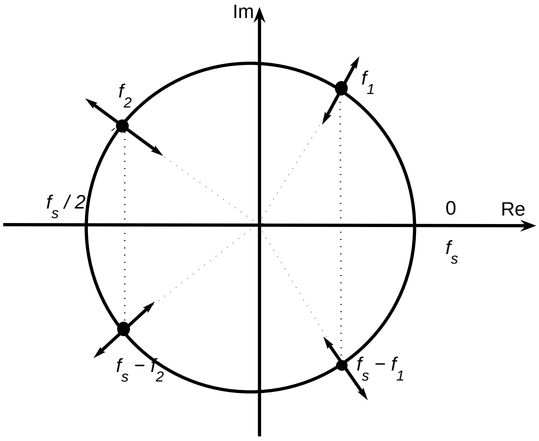


Fig. 2. Placing zeros on or near the unit circle when $N > 4$ pulses are used.

B. Linear combination of pulses

In order to generate a notch at the carrier frequency, f_j, of the primary user, the linear combination of pulses as defined in the IEEE 802.15.4a standard is considered. The new pulse shape, $p_{lcp}(t)$, is of the form

$$p_{lcp}(t) = \sum_{n=0}^{N-1} a_n \, p(t - \tau_n) \tag{14}$$

where $p(t)$ is a standard pulse used in the data transmission, $a_n \in [-1, 1]$ are the pulse coefficients, τ_n is the pulse delay and N is the number of pulses. According to the standard [13], the maximum number of pulses is limited by 4, and the pulse delays are restricted to $0 \leq \tau_n \leq 4$ ns with $\tau_0 = 0$. The new pulse shape given in (14) has the frequency domain representation

$$\begin{aligned} P_{lcp}(f) &= \sum_{n=0}^{N-1} a_n \, e^{-j2\pi f \tau_n} P(f) \\ &= C(f) \cdot P(f) \end{aligned} \tag{15}$$

where $C(f)$ is the code spectrum independent of the pulse spectrum $P(f)$. In order to generate notches at frequencies $\{f_j\}$ (and also at the integer multiples of $\{f_j\}$),

$$C(f) = \sum_{n=0}^{N-1} a_n \, e^{-j2\pi f \tau_n} \tag{16}$$

can be designed as suggested in [14]. Accordingly, by placing two zeros on the unit circle as a complex conjugate pair, a notch can be obtained at a desired frequency f_1 and at $f_s - f_1$ as shown in Fig. 2, where f_s represents the sampling frequency. In case more notch frequencies are desired, more conjugate pairs should be placed on the unit circle. However, this causes the number of pulses to be $N > 4$ in (14), which is not suggested by the standard. Therefore, by changing the locations of conjugate pairs (cf. Fig. 2) and yet placing them near the unit circle, tolerable magnitude values can be obtained at desired frequencies. This approach has the potential of approximating some coefficient values to zero when $N > 4$, however, is out of the scope of current research. Instead, we focus on generating a single notch frequency by placing zeros or multiple zeros on the unit circle so as to achieve notches with wider bandwidths. Limiting the number of pulses to $N = 4$ as in the IEEE 802.15.4a standard, new pulses that can be generated are presented next.

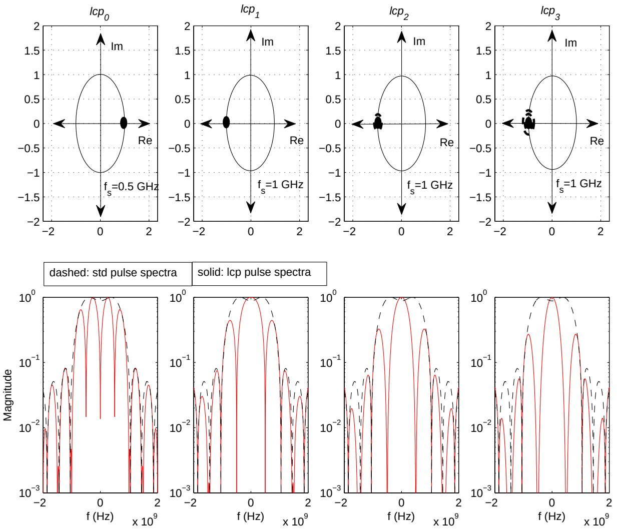


Fig. 3. Z-plane representations and corresponding magnitude spectra of linearly combined pulses.

C. Design of pulses with a single notch frequency

The pulses conforming to the standard specifications and generating a single notch at a desired frequency will be explained as follows. These designed pulses are obtained by placing zeros (or double/triple zeros) on the unit circle at the desired notch frequency. In Fig. 3, the z-plane representations and the corresponding magnitude spectra of the designed linearly combined pulses that have a notch at $f_j = 500$ MHz and the magnitude spectrum of a standard pulse are shown. Note that the standard pulses are 2ns-duration root raised cosine pulses [13], whereas the designed pulses[1] are the time-shifted and scaled aggregated pulses obtained by placing zeros on the z-plane. The frequency f_s corresponds to the sampling frequency.

Pulse lcp_0: A notch at the frequency $f_j = f_s$ (and also at the integer multiples of f_j) can be obtained by selecting the coefficients $a_0 = 1$, $a_1 = -1$ and the pulse delay $\tau_1 = 1/f_j$ in (14). The new pulse becomes

$$p_{lcp0}(t) = p(t) - p(t - 1/f_j) \, . \tag{17}$$

Pulse lcp_1: Similar to obtaining lcp_0, a notch at the frequency $f_j = f_s/2$ (and also at the odd integer multiples of f_j) can

[1] Note that the energy of the linearly combined pulse, $p_{lcp}(t)$, should be normalized to the energy of the standard pulse, $p(t)$, under the same transmission power constraint.

be obtained by selecting the coefficients $a_0 = 1$, $a_1 = 1$ and the pulse delay $\tau_1 = 1/f_j$ in (14). The new pulse becomes

$$p_{lcp1}(t) = p(t) + p(t - 1/f_j)\,. \tag{18}$$

Note that the differences between lcp_0 and lcp_1 are the a_1 coefficient and the sampling frequency f_s, which result from placing the zero at $z = 1$ (for lcp_0) and at $z = -1$ (for lcp_1) on the z-plane as can be seen in Fig. 3. This results in a larger mainlobe for lcp_1 compared to lcp_0 (cf. Fig. 3).

Pulse lcp_2: This pulse is obtained by placing double zeros at $z = -1$. Accordingly, a notch at the frequency $f_j = f_s/2$ (and also at the odd integer multiples of f_j) can be obtained by selecting the coefficients[2] $a_0 = 1$, $a_1 = 2$, $a_2 = 1$, and the pulse delays $\tau_1 = 1/f_j$ and $\tau_2 = 2/f_j$. The new pulse becomes

$$p_{lcp2}(t) = p(t) + 2p(t - 1/f_j) + p(t - 2/f_j)\,. \tag{19}$$

Note that the spectra of lcp_1 and lcp_2 are similar, where lcp_2 has a wider notch that can accommodate systems with wider bandwidths (cf. Fig. 3).

Pulse lcp_3: Similar to obtaining lcp_2, this pulse is obtained by placing triple zeros at $z = -1$. With four pulse coefficients, which is the maximum allowable number to be used, the widest notch at the frequency $f_j = f_s/2$ (and also at the odd integer multiples of f_j) can be obtained by selecting the coefficients $a_0 = 1$, $a_1 = 3$, $a_2 = 3$ $a_3 = 1$, and the pulse delays $\tau_1 = 1/f_j$, $\tau_2 = 2/f_j$ and $\tau_3 = 3/f_j$. The new pulse becomes

$$\begin{aligned} p_{lcp3}(t) = p(t) &+ 3p(t - 1/f_j) + 3p(t - 2/f_j) \\ &+ 3p(t - 3/f_j)\,. \end{aligned} \tag{20}$$

This pulse is expected to accommodate wideband systems better compared to lcp_0, lcp_1 and lcp_2.

D. Modified receiver structures

Since the received signal contains the interference term $J(t)$ and the transmitted pulse shape is $p_{lcp}(t)$, the received signal should be matched filtered with $p_{lcp}(-t)$ before performing coherent or noncoherent detection. Accordingly, the signal at the output of the matched filter is

$$r_{rec}(t) = r(t) * p_{lcp}(-t). \tag{21}$$

The useful signal component of $r_{rec}(t)$ can be obtained from (3), (4) and (21) as $\tilde{w}_l^{(1)}(t) * p_{lcp}(-t)$, where $\tilde{w}_l^{(1)}(t)$ consists of time-shifted pulses $p_{lcp}(t)$. Therefore, the correlation-based coherent receiver should use

$$v_{mrec}(t) = v_m(t) * p_{lcp}(-t) \tag{22}$$

as the new reference signal to obtain the correlator outputs in (6). On the other hand, the matched filtered signal $r_{rec}(t)$ can be directly used in (10) for the noncoherent receiver.

In the following, the performances of the original IEEE 802.15.4a transceiver structure and the modified transceiver structure that allows for coexistence are compared for various practical scenarios.

IV. Results

The system performances are evaluated in terms of the bit-error rate (BER) with respect to varying signal-to-noise-ratio (SNR) and signal-to-interference-ratio (SIR) values. The SNR and SIR are defined as E_b/N_0 and E_b/J_0, respectively, where E_b is the bit energy. It is assumed that the standard pulse used is a root raised cosine pulse with a roll-off factor $\beta = 0.6$ and duration $T_p = 2$ ns as given in [13]. The linearly combined pulses are obtained from (17)–(20) and they generate a notch at $f_j = 500$ MHz, where there is either an active narrowband system as given in (12) or an active wideband system as given in (13). Channel models used (CM1, CM5 and CM8) are the standardized IEEE 802.15.4a channel models [21] with a channel resolution of $T_c = 2$ ns.

The UWB system performance is evaluated for both the coherent and noncoherent operation modes. As a benchmark, performances of the original (i.e., standard) pulse and the designed pulses are determined when there is no interference (i.e., primary system) in addition to assessing the system performances for various interference scenarios. The main factors that affect the UWB-IR system performance are listed below and will be investigated in detail.

- IEEE 802.15.4a channel models
- Pulse types (lcp_0, lcp_1, lcp_2 lcp_3) in coherent and noncoherent operation modes
- Interference types (narrowband and wideband with different bandwidths/subcarriers)

A. Effects of channel models

Assuming the presence of a narrowband interference, the performance of UWB system is investigated in different IEEE 802.15.4a channel models for various SIR and SNR values when a coherent receiver is used. In Fig. 4, the BER performances are plotted for various SIR values when SNR = 15dB and 5-tap partial Rake receivers are used. When a standard pulse is used and there is no prefiltering (i.e., no matched filtering at the receiver front-end), the BER performance of the UWB system is poor for all SIR values and channel models. Note that this case is also unacceptable from the primary system's perspective (i.e., high UWB interference level). When the linearly combined pulse (lcp_0) is used instead of the standard pulse, the corresponding correlator template at the receiver provides an inherent interference rejection capability although it is limited. When a prefilter (i.e., matched filter) is used as well, the narrowband interference is successfully suppressed at all SIR values. All these observations are valid for any selected channel model. On the other hand, the performances improve in the order of CM1 (i.e., residential line-of-sight (LOS)), CM5 (i.e., outdoor LOS) and CM8 (i.e., industrial Non-LOS (NLOS)). This is mainly due to the channel models having higher number of multipaths in the order of CM1, CM5, CM8, and therefore, the 5-tap Rake receiver not being able to collect significant pulse energy for CM5 and CM8 channel models. In order to improve the system performance, the number of Rake fingers should be increased. In Fig. 5, the BER performances are plotted for

[2] In the implementation, the coefficients $\{a_n\}$ are normalized to conform to the standard requirements.

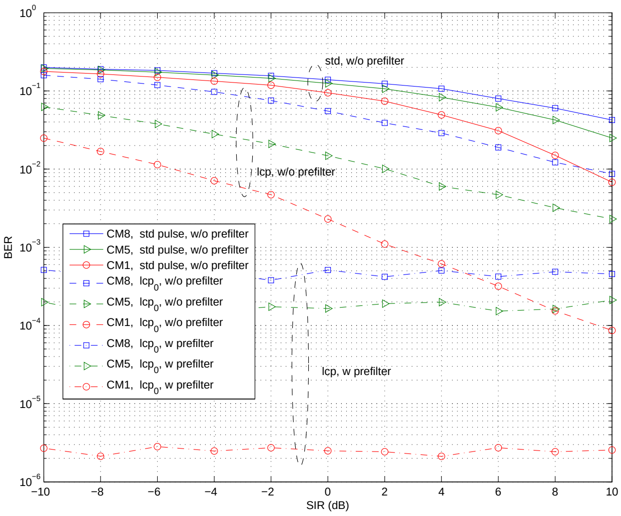


Fig. 4. BER performance of a coherent receiver for various SIR values and channel models when SNR = 15dB.

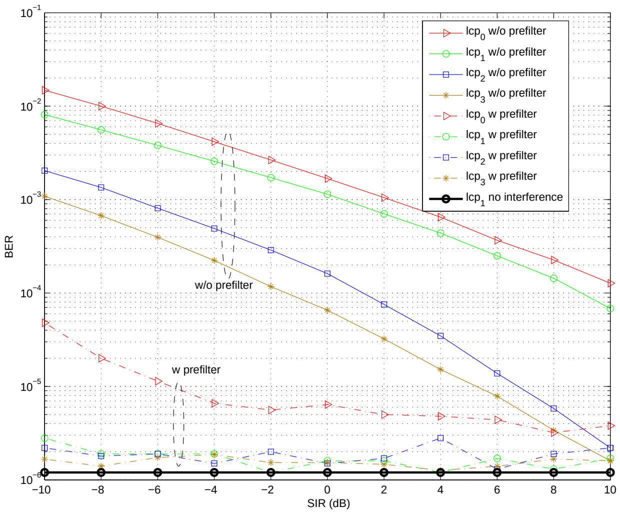


Fig. 6. BER performance of a coherent receiver in CM1 for various SIR values and pulse types.

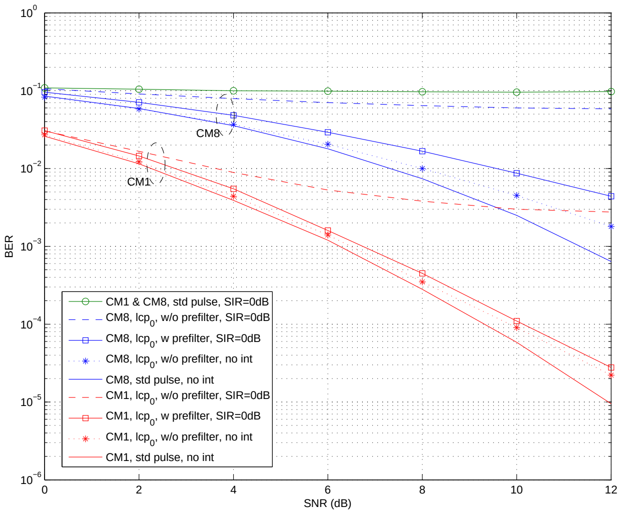


Fig. 5. BER performance of a coherent receiver for various SNR values and channel models.

various SNR values for the same 5-tap partial Rake receiver. As for the channel models, CM1 and CM8 are selected as the system shows the best and the worst performances, respectively. When a standard pulse is used, the performances are the best when there is no active primary system. However, if a narrowband system becomes active the BER performances degrade drastically for both CM1 and CM8. When lcp_0 is used, the performances are slightly worse than the standard pulse case when there is no interference. This can be explained by the duration of the linearly combined pulse becoming longer than $T_p = 2$ ns, which is also the assumed channel resolution. Hence, the performance degradation is due to the inter-pulse interference caused by the channel. If a narrowband system becomes active, while the linearly combined pulse with no prefiltering can provide some degree of interference suppression, including a front-end prefilter further improves the performances close to the no interference case for CM1 and CM8. As for comparative channel results, the system performance in CM1 is better than the one in CM8 as expected. For improving the system performance in CM8, more Rake fingers should be used at the receiver.

B. *Effects of pulse types*

Next, we consider the presence of a wideband interference and study the effect of pulse types on the UWB-IR system performance for both coherent and noncoherent receivers in residential LOS channel CM1. Initially, the coherent receiver performances of linearly combined pulses lcp_0, lcp_1, lcp_2 and lcp_3 are assessed in the presence of a wideband OFDM interference with 20 MHz bandwidth and 16 subcarriers for various SIR and SNR values, respectively. In Fig. 6, the performances are plotted for various SIR values when SNR = 15dB and 5-tap partial Rake receivers are used. The performance of lcp_1 when there is no interference serves as a benchmark. When there is an active primary system, the performances of lcp_1, lcp_2 and lcp_3 employing matched filters are similar to or slightly worse than the no interference case. This significant performance is due to filtering out the active licensed system successfully at the receiver end with the wide notches (cf. Fig. 3). On the other hand, lcp_0 with a prefilter performs poorly at lower SIR values because of its narrower notch width. In addition, without prefiltering the performances improve in the order of lcp_0, lcp_1, lcp_2 and lcp_3. This can be explained by more effective spectrum utilization and accommodation of a wider notch. In Fig. 7, the BER performances are plotted for various SNR values when SIR = 0dB and 5-tap partial Rake receivers are used for the same wideband interference. The performance of lcp_1 when there is no interference serves as a benchmark.

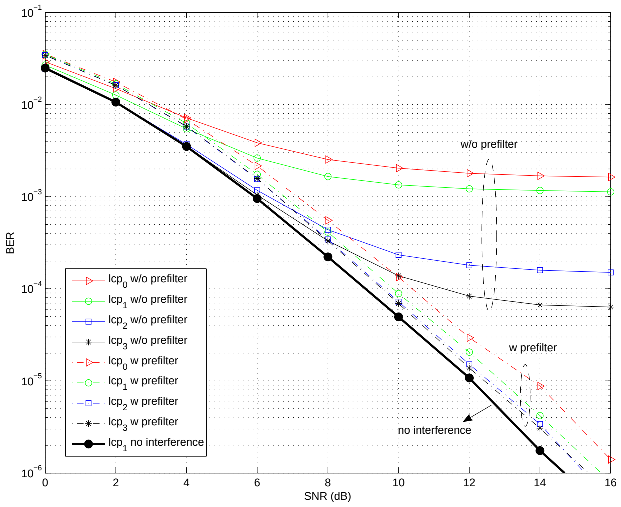


Fig. 7. BER performance of a coherent receiver in CM1 for various SNR values and pulse types.

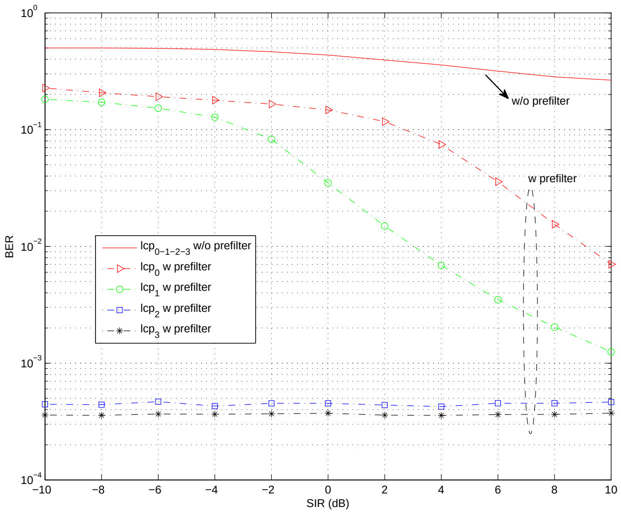


Fig. 8. BER performance of a noncoherent receiver in CM1 for various SIR values and pulse types.

While sharing the same band with an active primary system, lcp_1 used with front-end filtering attains about 0.5–1dB worse performance compared to the no interference case. This is also due to the prefiltering structure not being able to suppress the interference completely. When lcp_0 and lcp_1 are used with front-end filtering, lcp_1 performs slightly better than lcp_0 due to more effective spectrum shaping. When no prefiltering is employed, the performances get worse due to pulses' limited interference-rejection capability. It should also be noted that the performances are better in the order of lcp_3, lcp_2, lcp_1 and lcp_0 as expected.

Next, noncoherent receiver performances of linearly combined pulses lcp_0, lcp_1, lcp_2 and lcp_3 are assessed in the presence of the same wideband interference with the integration time of 16 ns for various SIR and SNR values, respectively. In Fig. 8, the BER performances are plotted for various SIR values when SNR = 30dB. When there is no prefiltering, the BER performance of the UWB-IR system is poor for all SIR values and all types of linearly combined pulses. Note that the similar performances of these pulses are shown with a single curve. Using prefiltering at the front-end, a linearly combined pulse improves the BER performance noticeably for lcp_0 and lcp_1, whereas using lcp_2 and lcp_3 with a prefilter at the front-end can suppress the interference efficiently independent from the SIR values due to having a wider notch width. In Fig. 9, the BER performances are plotted for various SNR values when SIR = 0dB. When there is no prefiltering, the BER performance of the UWB-IR system is poor for all SNR values and all types of linearly combined pulses due to integration of interference, noise and cross-terms. Using a prefiltering at the front-end, a linearly combined pulse improves the BER performance slightly for lcp_0 and lcp_1, whereas using lcp_2 and lcp_3 with a prefilter at the front-end can improve the system performance significantly due to having a wider notch width.

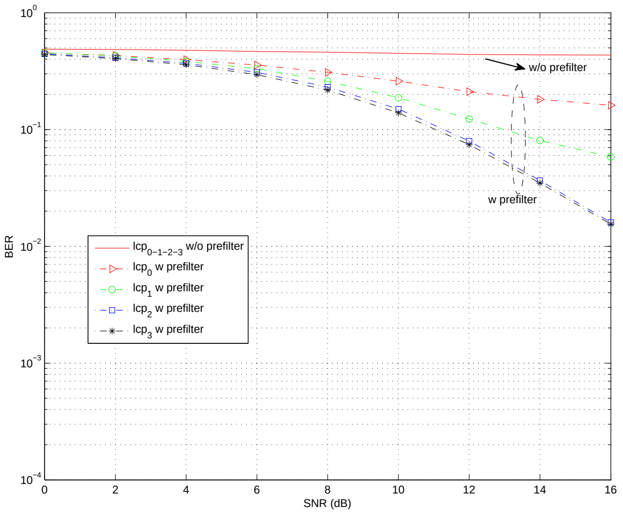


Fig. 9. BER performance of a noncoherent receiver in CM1 for various SNR values and pulse types.

As observed in this subsection, the selection of a linearly combined pulse is important in achieving a reasonable system performance for UWB-IR systems. In the presence of an OFDM interference with 20 MHz bandwidth and 16 subcarriers, it was observed that the pulses lcp_1, lcp_2 and lcp_3 achieved reasonable performances for coherent reception, whereas only lcp_2 and lcp_3 performed successfully in noncoherent reception. These performances depend on the bandwidth and the number of subcarriers of a wideband OFDM interference, and are investigated next.

C. *Effects of wideband interference*

The effects of bandwidths and number of subcarriers of a wideband OFDM interference are studied, respectively, in

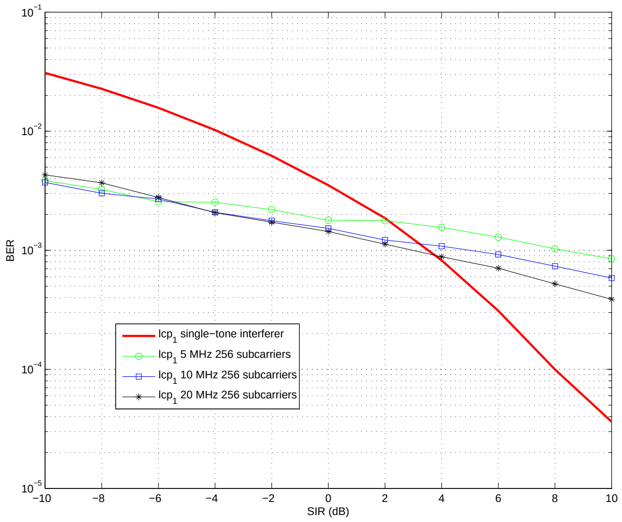


Fig. 10. BER performance of a coherent receiver using lcp_1 in CM1 for various SIR values and bandwidths.

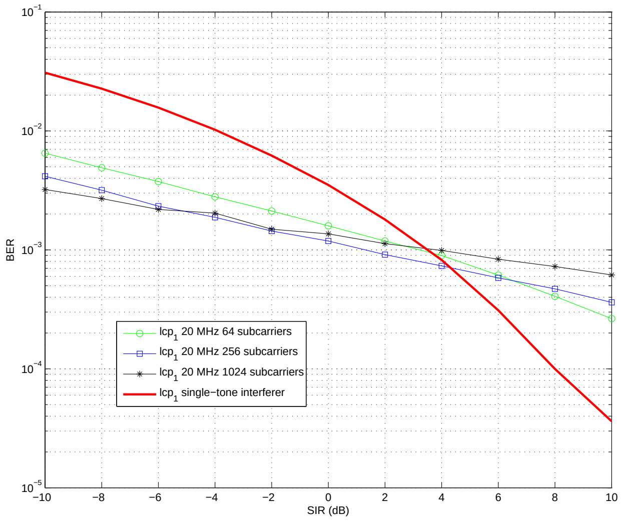


Fig. 11. BER performance of a coherent receiver using lcp_1 in CM1 for various SIR values and subcarriers.

Figs. 10 and 11. For both cases, the BER performances of lcp_1 are plotted for various SIR values when SNR = 15dB and 5-tap partial Rake receivers are used. The coherent receiver performance of lcp_1 is assessed without prefiltering. This consideration is a measure of inherent interference-rejection capability for given bandwidth and subcarriers. In Fig. 10, the effect of various bandwidths (5 MHz, 10 MHz, 20 MHz) for a fixed number of subcarriers (256 subcarriers) is investigated. The performance of lcp_1 with a single-tone interferer (narrowband interference) serves as a benchmark. It can be observed that, for most of the SIR values, the BER tends to decrease as the interferer bandwidth increases. This can be explained by the per-subcarrier interference PSD decreasing as the bandwidth increasing, where the average interference power is constant. A similar observation has been made in [19], where they analyzed the effect of WiMAX-OFDM interference on MB-OFDM systems. In Fig. 11, the effect of various number of subcarriers (64, 256, 1024) for a fixed bandwidth (20 MHz) is investigated. It can be observed that the UWB-IR system using a linearly combined pulse shows better performance in the presence of an interference with larger number of subcarriers for low SIR values. On the contrary, the UWB-IR performance is better for smaller number of subcarriers for high SIR values.

D. *Effects of random noise jamming*

The implementation of linear combination of pulses based on the assumption that the center frequency of the primary user (or interference) is known is the focus of the current research. While estimating the presence of primary users at different frequency bands is out of the scope of the current research and can be found in [22], some remarks should be made regarding random jamming. For a fixed power, if the bandwidth of the jamming signal is narrowband or wideband, then the spectrum should be listened to periodically to monitor the random jamming and to design the linearly combined pulses adaptively. On the other hand if the jamming noise has unlimited bandwidth, then its power spectral density will be very low due to the fixed power constraint. In that case, the UWB system will be able to operate without the need of linearly combined pulses, however, with some performance loss due to additive noise.

In this section, the performance of an IEEE 802.15.4a based system that can coexist with a narrowband or a wideband primary system has been investigated considering the realistic implementation issues such as practical receiver structures, pulse types and interference types in various channel models, SNR and SIR conditions. The presented results are important for determining appropriate pulse types and receiver structures when the operating frequency and the type of the active primary system are known. While this study focused on the performance of IEEE 802.15.4a based systems, possible performance degradation of primary systems should also be taken into account in the overall system design, which is out of scope of the current paper, however, is a future research topic of interest.

V. Conclusion

In this paper, we investigated possible implementations of linear combination of pulses and corresponding receiver structures in order that an IEEE 802.15.4a based UWB-IR system can operate in the same frequency band with a licensed narrowband or wideband system. Accordingly, a modified transceiver structure that allows for coexistence was presented and the UWB-IR system performance was assessed for various practical scenarios. The study showed that using a linearly combined pulse, the BER performances of coherent and non-coherent receiving structures may be slightly degraded when there is no active licensed system. However, if the licensed system becomes active, the performances can be significantly improved while not generating interference to the primary system. In addition, in the presence of a wideband interference, employing high order linearly combined pulses (e.g., lcp_2, lcp_3) can better compensate for the system performance due to having wider notches and more efficient spectrum shaping. In addition to coherent and noncoherent receiver structures and pulse types, the effects of channel models and wideband OFDM interference with various bandwidths and number of subcarriers on the UWB-IR system performance were also presented. The results of this study are important as the modified transceiver structure can achieve a reasonable system performance while complying with the European and Japanese regulatory agency mandates.

Acknowledgment

This research was supported by a Marie Curie International Reintegration Grant within the 7th European Community Framework Programme.

References

[1] J. Mitola and G. Q. Maguire, "Cognitive radio: making software radios more personal," *IEEE Personal Commun.*, vol. 6, pp. 13–18, Aug. 1999.

[2] M. Z. Win and R. A. Scholtz, "Ultra-wide bandwidth time-hopping spread-spectrum impulse radio for wireless multiple-access communications," *IEEE Trans. Commun.*, vol. 48, pp. 679–691, Apr. 2000.

[3] W. Zhang, R. K. Mallik, and K. B. Letaief, "Optimization of cooperative spectrum sensing with energy detection in cognitive radio networks," *IEEE Trans. Wireless Commun.*, vol. 8, pp. 5761–5766, Dec. 2009.

[4] Z. Quan, S. Cui, A. H. Sayed, and H. V. Poor, "Optimal multiband joint detection for spectrum sensing in cognitive radio networks," *IEEE Trans. Signal Proc.*, vol. 57, pp. 1128–1140, Mar. 2009.

[5] P. Paysarvi-Hoseini and N. C. Beaulieu, "Optimal wideband spectrum sensing framework for cognitive radio systems," *IEEE Trans. Signal Proc.*, vol. 59, pp. 1170–1182, Mar. 2011.

[6] Y. P. Nakache and A. F. Molisch, "Spectral shaping of UWB signals for time-hopping impulse radio," *IEEE J. Selected Areas Commun.*, vol. 24, pp. 738-744, Apr. 2006.

[7] European Commission, "2009/343/EC: Commission Decision of 21 April 2009 amending Decision 2007/131/EC on allowing the use of the radio spectrum for equipment using ultra-wideband technology in a harmonised manner in the Community," *Official Journal of European Union*, L 109, 9–13, Apr. 2009.

[8] S. M. Mishra, R. W. Brodersen, S. T. Brink, and R. Mahadevappa, "Detect and avoid: an ultra-wideband/WiMAX coexistence mechanism," *IEEE Commun. Mag.*, vol. 45, pp. 68–75, June 2007.

[9] S. Erküçük, L. Lampe, and R. Schober, "Joint detection of primary systems using UWB impulse radios," *IEEE Trans. Wireless Commun.*, vol. 10, pp. 419–424, Feb. 2011.

[10] X. Wu, Z. Tian, T. N. Davidson, and G. B. Giannakis, "Optimal waveform design for UWB radios," *IEEE Trans. Signal Proc.*, vol. 54, pp. 2009–2021, June 2006.

[11] I. Dotlic and R. Kohno, "Design of the family of orthogonal and spectrally efficient UWB waveforms," *IEEE Jour. Select. Topics Signal Proc.*, vol. 1, pp. 21–30, June 2007.

[12] Y. Wang, X. Dong, and I. J. Fair, "Spectrum shaping and NBI suppression in UWB Communications," *IEEE Trans. Wireless Commun.*, vol. 6, pp. 1944–1952, May 2007.

[13] IEEE Std 802.15.4a-2007, "Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)," 2007.

[14] S. Erküçük and B. A. Kaleli, "Linear combination of pulses for coexistence in the IEEE 802.15.4a standard," *IEEE Proc. SIU*, pp. 620–623, Apr. 2009.

[15] L. Zhao and A. M. Haimovich, "Performance of ultra-wideband communications in the presence of interference," *IEEE Jour. Select. Areas Commun.*, vol. 20, pp. 1684–1690, Dec. 2002.

[16] M. Hamalainen, V. Hovinen, R. Tesi, J. H. J. Iinatti, and M. Latva-aho, "On the UWB system coexistence with GSM900, UMTS/WCDMA, and GPS," *IEEE Jour. Select. Areas Commun.*, vol. 20, pp. 1712–1721, Dec. 2002.

[17] M. Di Renzo, F. Tempesta, L. A. Annoni, F. Santucci, F. Graziosi, R. Minutolo, and M. Montanari, "Performance evaluation of IR-UWB D-Rake receivers over IEEE 802.15.4a multipath fading channels with narrow-band interference," *IEEE Proc. ICUWB*, pp. 71–76, Sep. 2009.

[18] B. Hu and N. C. Beaulieu, "Effects of IEEE 802.11a narrowband interference on a UWB communication system," *IEEE Proc. ICC*, pp. 2818–2814, May 2005.

[19] C. Snow, L. Lampe, and R. Schober, "Impact of WiMAX interference on MB-OFDM UWB systems: analysis and mitigation," *IEEE Trans. Commun.*, vol. 57, pp. 2818–2827, Sep. 2009.

[20] Ç. Fındıklı, S. Erküçük, and M. E. Çelebi, "Performance of IEEE 802.15.4a systems in the presence of narrowband interference," *IEEE Proc. ICUWB*, pp. 395–399, Sep. 2011.

[21] A. F. Molisch et. al., "A comprehensive standardized model for ultrawideband propagation channels," *IEEE Trans. Antennas Propag.*, vol. 54, pp. 3151–3166, Nov. 2006.

[22] B. Yılmaz and S. Erküçük, "Detection of interdependent primary systems using wideband cognitive radios," *AEÜ Intl. J. Electron. Commun.*, vol. 67, pp. 926-936, Nov. 2013.

Çağlar Fındıklı received the B.Sc. degree in Electronics Engineering from Kadir Has University, Istanbul, Turkey and the M.Sc. degree in Telecommunications Engineering from Istanbul Technical University in 2009 and 2012, respectively. He is currently pursuing a Ph.D degree at Istanbul Technical University in Telecommunications Engineering. In addition to his academic career, he has worked for several software companies as software QA manager and consultant. His research interests include ultrawideband systems, emerging communication systems and the design of energy efficient systems.

Serhat Erküçük received the B.Sc. and M.Sc. degrees in Electrical Engineering from Middle East Technical University, Ankara, Turkey and from Ryerson University, Toronto, ON, Canada, in 2001 and 2003, respectively, and the Ph.D. degree in Engineering Science from Simon Fraser University, Burnaby, BC, Canada in 2007. He was an NSERC postdoctoral fellow at the University of British Columbia, Vancouver, BC, Canada until September 2008. Since then, he has been with Kadir Has University, Istanbul, Turkey, where he is currently an associate professor. His research interests are in physical layer design of emerging communication systems, wireless sensor networks, communication theory and signal processing for communications. Dr. Erküçük serves as an area editor for AEÜ - International Journal of Electronics and Communications. He is also a recipient of the Governor General's Gold Medal.

Mehmet Ertuğrul Çelebi received his B.Sc. degree from Istanbul Technical University, Istanbul, Turkey, in 1981, and his M.Sc. and Ph.D. degrees from Polytechnic University, Brooklyn, New York, USA, in 1984 and 1990, respectively. Since 1992, he has been a faculty member at the Department of Electrical and Electronics Engineering of Istanbul Technical University, Istanbul, Turkey, where he is a full professor. From 2001 to 2002 he was a Visiting Professor at New Jersey Institute of Technology, NJ, USA, and from 1995 to 1999 he was an Adjunct Lecturer at Galatasaray University, Istanbul, Turkey. His current research interests include cooperative communications, coding and decoding for massive MIMO, cognitive radio, and resource allocation in wireless systems.

Comparison of Bit Error Rate of Line Codes in NG-PON2

Tomas Horvath, Radek Fujdiak, Milan Cucka, Marie Dankova and Jiri Misurec

***Abstract*—This article focuses on simulation and comparison of line codes NRZ (Non Return to Zero), RZ (Return to Zero) and Miller's code for NG-PON2 (Next-Generation Passive Optical Network Stage 2) using. Our article provides solutions with Q-factor, BER (Bit Error Rate), and bandwidth comparison. Line codes are the most important part of communication over the optical fibre. The main role of these codes is digital signal representation. NG-PON2 networks use optical fibres for communication that is the reason why OptSim v5.2 is used for simulation.**

***Keywords*—Miller's code, RZ, NRZ, NG-PON2, Matlab, Simulations**

I. Introduction

The first standard above 1 Gbps was GPON. GPON standard was improved by ITU-T in 2003. The following standard was XG-PON, which has 10 Gbps bandwidth. In general, the transition from GPON to XG-PON had a disadvantage because the ISP needed to change PIN photodetector onto APD photodetector in ONU unit. All mentioned standards came from ITU-T. The second group, which has provided recommendation for passive optical networks, is an IEEE group. In general, standards of IEEE group are based on Ethernet technology. That is the reason why these standards dominate in Asian access networks. The latest standard NG-PON2 still covered by commonly line codes NRZ and RZ define in general requirements. Miller's code is mentioned only in appendix such as next possibility.

This paper deals with line codes in NG-PON2 standards for passive optical networks. On the other hand, nowadays only general requirements have been done [1]. In general, access networks have used the same line codes for a long time (from GPON to NG-PON2). In [2], Miller's code was mentioned such as the new possibility for using in the new optical networks.

This article presents a new way of using Miller's code in access networks for the latest standard. Miller's code is known as a decoder for delay–modulation coded Data [3]. The mentioned article was published in 1971. Nowadays, line codes are still from OOK (OnOff Keying) which are not the most bandwidth efficient. We designed Miller's coder in Matlab and implemented it into OptSim simulation application due to the comparison line codes: NRZ, RZ and Miller's coded in NG-PON networks.

The main contribution of this paper is the determination of the maximum bit error rate value in the transfer of 3 different line codes in NG-PON2 networks. This is followed by the research into effects of increasing spectral width of laser on bit error rate with different line codes.

The rest of this paper is structured as follows. Section 2 gives an overview of some other related works. Section 3 describes the line codes in our model. Section 4 shows the designed model and parameters of blocks. Section 5 discusses the results achieved and Section 6 concludes the paper.

II. State of Art

Many works related to a comparison of the line codes have been published in recent years. Works published these days are aimed at a comparison of line codes presented in simulation applications as a NRZ, RZ, etc.

Authors in [4] investigates the suitabillity of various data formats for optical XOR gate at 10 Gb/s. They focuse on different data formats for high–speed optical networks in XOR gate. NRZ, RZ, and Manchester modulation format was tested and verified. On the other hand, they used only simple line codes, where Miller's code is missing. In NG-PON2 networks the optical amplifiers have not been presented in distribution network. In the next publication [5] the influence of binary modulations was presented. They used NRZ, RZ, CSRZ, and CSRZ line codes for their research. On the other hand, Miller's code was not included. Further, the publication [5] used only 2.5 Gbps in simulation model. Our model has 40 Gbps which is a combination with four 10 Gbps transmission system.

Other articles [6], [7], deal with the same line codes as was mentioned above. None of them contain Miller's code, known as Sigma–Delta modulation, the articles compared only the base line codes, which are proposed in simulation networks. In general, Miller's code is able to improve a bit error rate in coexistence scheme and single technology network. The last article [8] deals with PMD (Polarization Mode Dispersion) for NG-PON, more precisely 10 Gbps network.

The main proposed reason of our article is the comparison of the current line codes with the known Miller's code from data backup field. Another one [9] deal with coexistency the three latest standards. On the other hand, the paper does not deal with influence of Miller's code in coexistence network.

Our work focuses on a design and implementation of well-known code for visible light communication [10], called Miller's code. The main contribution of this paper is a comparison of the included line codes with our implementation of Miller's code in OptSim v5.2 Miller's code brings better results for attenuation values.

Research described in this paper was financed by the National Sustainability Program under grant LO1401. For the research, infrastructure of the SIX Center was used.

Manuscript received April 12; 2016 revised May 20, 2016.

doi: 10.11601/ijates.v5i2.165

III. NG-PON2

We dealt only with the latest standard ITU-T G.989 called G-PON2. A general specification is available in [11]. This standard was improved in March 2013. First, the study group 15 had the main goal to select the main technology for these networks. The study group 15 had many technologies for their decision: WDM-PON (Wavelenght Division Multiplexing PON) [12], OFDM-PON (Orthogonal Frequency Division Multiplexing PON) [13], UDWDM-PON (Ultra Dense WDM PON) and TWDM-PON (Time and Wavelenght Division Multiplexing PON) [1]. In the final version TWDM-PON technologies have been selected. TWDM technology needs to be changed only at ONU side that means ISP (Internet Services Provider) should change ONU units, due to TWDM technology required the tuneable laser at customer side. On the other hand, there is no change in attenuation classes for passive optical networks according to [1] or [2], see in Tab. I.

TABLE I
ATTENUATION CLASSES FOR NG-PON2

	Nominal1	Nominal2	Extended1	Extended2
Minimum loss [dB]	14	16	18	20
Maximum loss [dB]	29	31	33	35

In general, TWDM-PON networks are the combination of four or eight channels, which are the same as in XG-PON but they use wavelength of NG-PON2 spectrum (see in Tab. II).

TABLE II
WAVELENGTHS FOR DOWNSTREAM COMMUNICATION (OLT TO ONU)

Downstream		
Channel	f [THz]	λ [nm]
1	187.8	1596.3389
2	187.7	1597.1894
3	187.6	1598.0408
4	187.5	1598.8931
5	187.4	1599.7463
6	187.3	1600.6004
7	187.2	1601.4554
8	187.1	1602.3113

IV. LINE CODE

Line codes also called digital baseband modulation is used for telecommunication systems. It is an amplitude and time–discrete signal which is specially tuned for the channel. Line coding is made from a signal which is represented by voltage and in digital expression zeros and ones. If a sender and a recipient communicates, this process is called coding lines. There are several types of encoding: polar, unipolar, bipolar, and Manchester encoding.

Selecting the appropriate code in the system design is an important step. The same encoding cannot be used for copper and fibre optic lines because optical lines cannot have negative values. This problem is solved by the permanent preload, tristate code or unipolar code. The next option is transfer from bipolar signal to binary form and transfer to suitable line code.

The line signal is given by several requirements, such as high levels of clock components, constant DC component, minimum bandwidth, simple error detection etc. Optical transmission has several requirements: the performance and linearity modulation characteristics of the radiation source, attenuation and bandwidth cable or noise characteristics of the detector. In coding with line code NRZ and RZ codes are commonly used for their simplicity. Long sequences of ones and zeros are excluded by a scrambler.

A. Return to zero - RZ

Encoding processes of zeros and ones use positive and negative pulses. It is important to return to the neutral state. This means that there is zero voltage. The voltage is used to synchronize the clocks of the sender and recipient and all without the use of a clock signal. For coding greater bandwidth is required.

The binary signal is encoded by rectangular pulse amplitude modulation with this type of code. In coding return to zero inverted (RZI) is also used. It is a method of mapping for transmission. There are two levels of RZI signal. The signal has a pulse if the binary signal is zero and if the binary signal has no pulse, it is one. This method uses serial infrared IrDA physical layer specification.

B. Non return to zero - NRZ

In this coding logical ones mean specific value and logical zeros mean another specific value. This can be a positive or negative voltage. The signal contains no other values. Non return to zero encoding cannot be used for synchronizing transmissions. If there is a need to synchronous transfer, it must add synchronization with RLL (Run Length Limited) or add clock signal.

WDM also called Wavelength division multiplexing is used for combining multiple signals on laser beams at various infrared wavelengths for transmission along fibre. Laser generates a different type of signals. Laser is modulated by independent set of signals. Wavelength sensitive filters are used at the receiving end.

WDM modulation is similar to FDM which uses radio frequencies, but in WDM IR portion of the electromagnetic spectrum is used.

In a WDM system, a signal in each channel has a different carrier frequency, and the model equation of the system can be obtained by setting the electric field q of the NLS equation [14]:

$$i\frac{\partial q}{\partial Z} + \frac{\beta_2}{2}\frac{\partial^2 q}{\partial T^2} + v|q|^2 q = 0, \tag{1}$$

in term of:

$$q = \sum_{j=1}^{n} q_j, \tag{2}$$

where q_j represents the electric field in the j-th channel having carrier frequency, and ω_j means two channels,

$$\begin{aligned} i\tfrac{\partial q_1}{\partial Z} - \tfrac{\beta_2}{2}\tfrac{\partial^2 q_1}{\partial T^2} + \left(|q_1|^2 + \alpha|q_2|^2\right) q_1 = 0, \\ i\tfrac{\partial q_2}{\partial Z} - \tfrac{\beta_2}{2}\tfrac{\partial^2 q_2}{\partial T^2} + \left(|q_2|^2 + \alpha|q_1|^2\right) q_2 = 0, \end{aligned}$$

four channels, $\alpha = 1$ (four channels are used in NG-PON2 topology with combining four 10 Gbit data source),

$$\begin{aligned}
i\frac{\partial q_1}{\partial Z} - \frac{\beta_2}{2}\frac{\partial^2 q_1}{\partial T^2} + \left(|q_1|^2 + |q_2|^2 + |q_3|^2 + |q_4|^2\right) q_1 = 0,\\
i\frac{\partial q_2}{\partial Z} - \frac{\beta_2}{2}\frac{\partial^2 q_2}{\partial T^2} + \left(|q_1|^2 + |q_2|^2 + |q_3|^2 + |q_4|^2\right) q_2 = 0,\\
i\frac{\partial q_3}{\partial Z} - \frac{\beta_2}{2}\frac{\partial^2 q_3}{\partial T^2} + \left(|q_1|^2 + |q_2|^2 + |q_3|^2 + |q_4|^2\right) q_3 = 0,\\
i\frac{\partial q_4}{\partial Z} - \frac{\beta_2}{2}\frac{\partial^2 q_4}{\partial T^2} + \left(|q_1|^2 + |q_2|^2 + |q_3|^2 + |q_4|^2\right) q_4 = 0.
\end{aligned}$$

C. *Miller's code*

Miller's code is used for Pseudorandom Binary Sequence (PBS). This code is known as delay modulation or modified frequency modulation. It contains rectangular pulses with two levels. Binary zeros cause no change of signal level. If this zero is followed by another zero, it means that transition to the other level takes place at the end of the first bit period. Binary ones mean transition from one level to the other in the middle of the bit period. Miller's code is primarily used for encoding radio signals and data backup. Frequency spectrum of the encoded signal contains less low-frequency energy than a non-return to zero signal and less high frequency energy than a biphase signal.

$$\begin{aligned}
S(\omega) = \frac{2}{\omega^2 T[17+8\cos 4\omega T]\cdot[23-2\cos\frac{\omega T}{2}-22\cos\omega T]}\cdot\ldots\\
\ldots\cdot\left[-12\cos\frac{3\omega T}{2} + 5\cos 2\omega T + 12\cos\frac{5\omega T}{2}\right] + \ldots\\
\ldots + \left[2\cos\frac{3\omega T}{2} - 8\cos\frac{7\omega T}{2} + 2\cos\frac{4\omega T}{2}\right].
\end{aligned}$$

V. OptSim

The simulation software has been developed by Synopsis. The main advantage of this application is a real time simulation of the real networks. Of course, all parameters are described via math equations which mean that some results are idealized. The application is able to simulate the following optical technologies: CWDM, DWDM, OTDM, FTTx etc. On the other hand, we can simulate analog signals of various modulation format too. The most important property of OptSim is cooperation with external applications Matlab and Spice, for example. Our solution of Miller's code was completely designed in Matlab.

VI. Simulation model

At first, we needed to design and implement Miller's code into OptSim application. Miller's code method (Delay method) is not implemented only in OptSim but also in other simulation tools. Our used simulation tool OptSim allows implementation and developing only Matlab environment possibility as functional matlab block. Matlab block in OptSim could have only electrical and optical input and output. It means for us that it is also not possible to use the logical sources (generators etc.), which are used for example with RZ or NRZ functional blocks. Miller's code, which is used for our simulation, was implemented in Matlab code. Our code provides also functional random generator, which is able to generate bits in necessarily speed for the system.

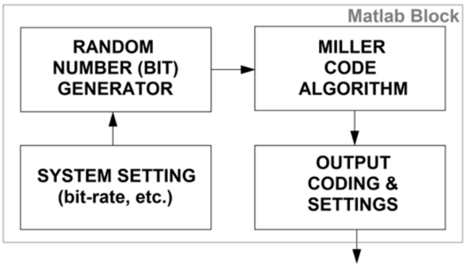
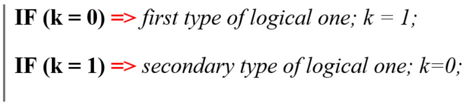

Matlab Block
RANDOM NUMBER (BIT) GENERATOR
MILLER CODE ALGORITHM
SYSTEM SETTING (bit-rate, etc.)
OUTPUT CODING & SETTINGS

Fig. 1. Matlab code schematic

Matlab code schematic can be seen in Fig. 1. The Matlab block has not inputs, because it was not possible to use logical sources for it and we create our source for the system. System settings include initiation data, bit-rate, a number of bit settings and also all other necessarily inputs and settings for our random generator and the whole system. The generator creates random bit sequences, which are coded by Miller's code algorithm. The output coding and settings only change Miller's code sequences to the right format for the OptSim simulator and it is also possible to choose there, for example, amplitude of the signal and some smaller and less important settings.



IF (k = 0) => *first type of logical one; k = 1;*

IF (k = 1) => *secondary type of logical one; k=0;*

Fig. 2. Main principal of coding

Miller's code has two types of logical one, it is necessary to hold some control variable, which will give order to a final sequence. The simple explanation of algorithm is in Fig. 2. The generator generates bits, which goes to Miller's code algorithm, where the Miller-code sequence is created. First, each bit (1 or 0) is extended by time and speed variable (10 Gbps) and after it is with the Miller code algorithm (Fig. 2) changed the logical 1 to the Miller logical one (to the first or second type). The technology NG-PON2 voids from combination four (40 Gbps) or (80 Gbps) sources based on ITU-T G.987. That means the wavelength multiplexer and demultiplexer must be included in real network and in our simulation model. We selected four 10 Gbps data source wavelength of NG-PON2 range, in concrete 1596.3389, 1597.1894, 1598.0408, and 1598.8931 nm.

For the NG-PON2 networks four attenuation plans in two groups were proposed: nominal1, nominal2, extended1, and extended2 (see Tab. 1.) For example, extended1 and extended2 attenuation classes required APD diode in ONU unit. On the other hand, we designed model only with APD diodes due to reflecting a recommendation. Our simulation model provides the total attenuation of 34.5 dB (in the last scenarios).

NG-PON2 networks have four parts. Each transmitting part contains the following properties: 10 Gbps bitrate with pseudorandom sequence, wavelength of NG-PON2 range, various line code, CW laser, and modulator. Topology can be seen in Fig. 3.

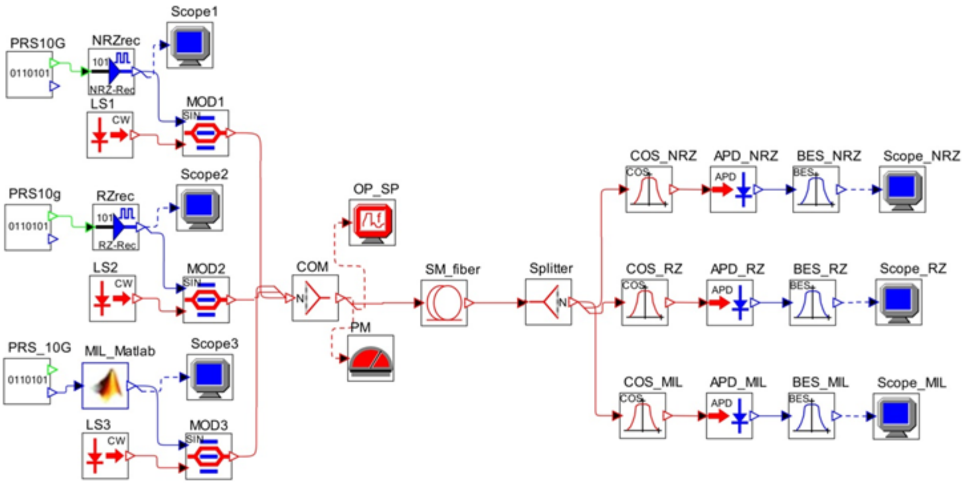


Fig. 3. Proposed topology for simulation with various line codes

As obvious in Fig. 3, we designed three optical sources with different line codes. First, we used NRZ line code, the second one was RZ. Both of them were carried out by Synopsis application OptSim. The third one for a comparison we need to implement Miller's code via Matlab.

Optical distribution network contained WDM combiner, single mode optical fibre with 20 km length, and splitter with different split ratio. The main goal of our article is the comparison of well-known line codes, which are used in the optical networks with Miller's code that were used at first in [3].

We created 2 scenarios with 8 values of split ratio (from 1:2 to 1:256 and the higher attenuation value) and 2 higher values than it was defined in E1 attenuation class (28 and 29.5 dB). The selected attenuation values could be seen in Tab. III.

TABLE III
THE ATTENUATION VALUES FOR SPLITTER IN ODN

Split ratio	Attenuation [dB]
1:2	3.6
1:4	7
1:8	9.6
1:16	13.2
1:32	17.4
1:64	20.3
1:128	23.5
1:256	27.1

Above mentioned merits are only for splitter, we need to add another attenuation of the rest component for total attenuation. In the final scenario we had $1 + 5 + 27.1 = 33.1$ dB. On the other hand, the recommendation ITU-T defined 35 dB as the maximum values. Due to a standard definition of attenuation maximum value, we had the last scenario with total attenuation of 35.5 dB in the optical distribution network.

VII. Simulation Results Discussion

All results are discussed with achieved BER values. As was already mentioned, the main parameter of our simulation is bit error rate. Another parameter is Q-factor, which describes the analog quality of digital signal with respect to SNR (Signal to Noise Ratio). Q-factor is defined by the following equation:

$$Q = \frac{I_1 - I_0}{\sigma_1 + \sigma_0} \quad [-]\,, \tag{3}$$

where I_1 logic level 1, I_0 logic level 0, σ_1 standard deviation of the logic level 1, σ_0 standard deviation of the logic level 0 [3].

When we know Q-factor, we can define BER with the following equation:

$$BER = \frac{1}{2} erfc\left(\frac{Q}{\sqrt{2}}\right) \approx \frac{\exp\left(\frac{-Q^2}{2}\right)}{Q\sqrt{2}} \quad [-]\,. \tag{4}$$

First of all, we designed a scenario for simulation of the maximum attenuation in ODN (Optical Distribution Networks) and we examined the values of bit error rate. As was mentioned in the previous chapter, final attenuation was 35.5 dB for our simulation model. The results of BER took a logarithm and the values were depicted in graph, which can be seen in Fig. 4.

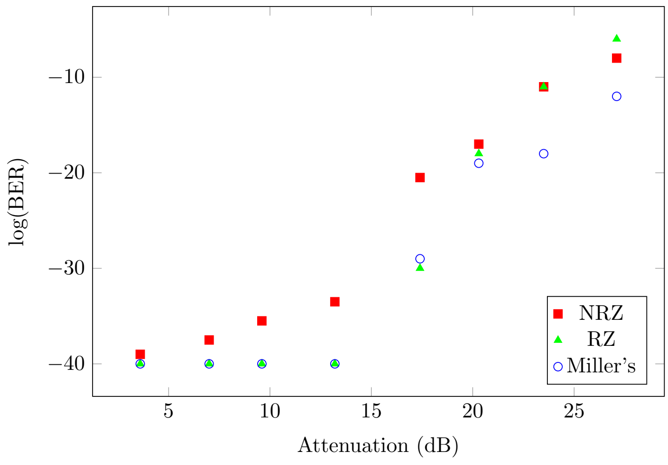


Fig. 4. Dependence of bit error rate on the attenuation value

In Fig. 4 it can be seen that the Miller's code has one of the best results for attenuation values. In particular, Miller's code had BER 10^{12} for 27.1 dB, NRZ had 10^{7}, and RZ had 10^{6}. In comparison, according to [1] only Miller's code should be accepted as applicably value. Source [1] defined the border BER value as 10^{12}, which should be accepted in NG-PON2 networks. As was mentioned above, NRZ and RZ line codes had these values for the lower split ratio 1:128 (23.5 dB). We did not implement the FEC (Forward Error Correction).

The comparison of the eye diagram for split ratio 1:64 for Miller's and NRZ line code can be seen in Fig. 5 Miller's code is represented in red and NRZ is in blue. From this eye diagram the following parameters should be compared: among others jitter and decision point. The values of jitter are 0.02 ns for NRZ and half of them 0.01 ns for Miller's code. In general, these values are completely sufficient because VoIP (Voice over Internet Protocol) required maximum jitter value of 150 ms. Another highlighted parameter is a decision point. In comparison with NRZ it has better value that Miller's code, but the eye diagram of NRZ is not so clear as Miller's code.

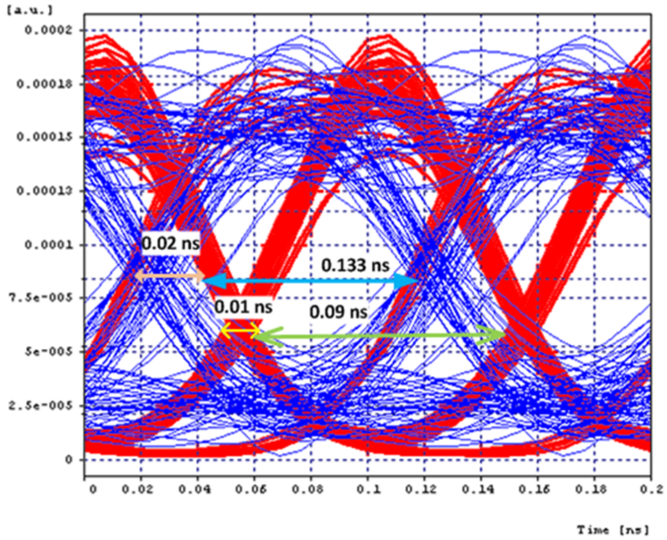


Fig. 5. The comparison of Miller's and NRZ eye diagram

Our scenario was divided into two parts, the results of the first part are described above. The second part of the scenario discovered the influence of FWHM on bit error rate. In simulation, the spectral width of the laser with 10 MHz (default value) step was increased to 100 MHz. The aim was to achieve, as in the previous simulation, the boundary values of bit error rate, BER. On the average our designed Miller's code has one of the best results for this simulation. For example, NRZ and RZ had from 30 MHz to 50 MHz linear character for another increasing spectral width BER rapidly dropped on the border value. The overall decline was 3 orders in bit error rate. The resulting graph of the simulation is shown in Fig. 5. Simulated topology was still the same as the previous one but for this scenario we used split ratio 1:128, optical power of laser was 6 dBm for all channel, and 20 km of optical fibre length. For the simulation topology must be the same, because of spectral with and their effect. If we have maximal split ratio, the results are incomparable and misguided. The next step was determination of the maximum bit error rate value in the transfer of three different line codes in NG-PON2 networks. This is followed by the research into effects of increasing spectral width of laser on bit error rate with different line codes Fig. 6.

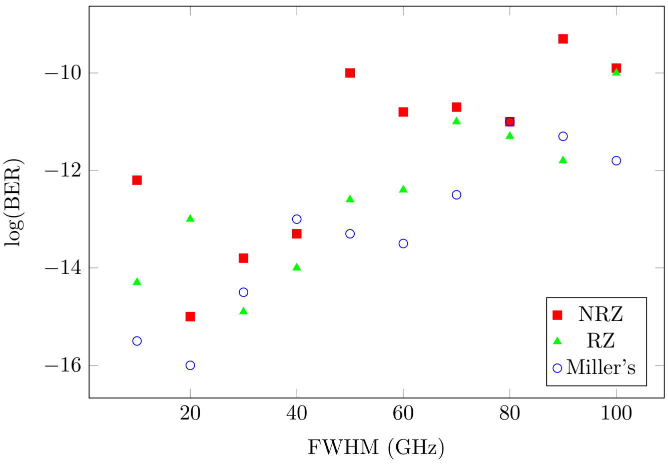


Fig. 6. Dependence of bit error rate on the spectral width of the laser

VIII. Conclusion

This article proposed a comparison of line codes with our implementation of Miller's code in OptSim. The next step was determination of the maximum bit error rate value in the transfer of three different line codes in NG-PON2 networks. This is followed by the research into effects of increasing spectral width of laser on bit error rate with different line codes. We created simulation with a variety of split ratios (from 1:2 to 1:256 and the higher attenuation value) and 2 higher values than were defined in E1 attenuation class (28 and 29.5 dB). First of all, we designed a scenario for simulation of the maximum attenuation in ODN (Optical Distribution Networks) and we examined the values of bit error rate. Final attenuation was 35.5 dB for our simulation model. Miller's code has shown one of the best results for attenuation values. In particular, Miller's code had BER 10^{12} for 27.1 dB, NRZ had 10^{7}, and RZ had 10^{6}. In comparison with other line codes, according to [1] only Miller's code has achieved the best results. All of these results were without forward error correction. FEC should also be implemented in future research.

References

[1] *Recommendation ITU-T G. 989. 1, 40–Gigabit–capable passive optical networks (NG-PON2): General requirements* 2013.

[2] *Recommendation ITU-T G. 989. 2, 40–Gigabit-capable passive optical networks 2 (NG-PON2): Physical media dependent (PMD) layer specification.* 2013.

[3] J. Lewin. Decoder for Delay-Modulation Coded Data, in *IEEE Transactions on Communications*, vol. 19, no. 5 pp. 719–723, 1971.

[4] P. Singh, D. K. Tripathi, and H. K. Dixit, Investigation of XOR operation in all-optical system with NRZ, RZ and Manchester modulation formats, in *2012 International Conference on Communications, Devices and Intelligent Systems (CODIS)*, pp. 417–420, 2012.

[5] V. Tejkal, M. Filka, J. Sporik, P. Reichert, and P. Munster, The influence of binary modulations in passive optical network based on WDM, in *Proc. 34th International Conference on Telecommunications and Signal Processing (TSP)*, Budapest, pp. 141–144, 2011.

[6] S. Yao, S. Fu, H. Wang, M. Tang, P. Shum, and D. Liu, Performance Comparison for NRZ, RZ, and CSRZ Modulation Formats in RS-DBS Nyquist WDM System, in *Journal of Optical Communications and Networking*, vol. 6, no. 4, p. 355, 2014.

[7] G. Bosco, A. Carena, V. Curri, R. Gaudino, and P. Poggiolini, On the use of NRZ, RZ, and CSRZ modulation at 40 Gb/s with narrow DWDM channel spacing, in *Journal of Lightwave Technology*, vol. 20, no. 9, pp. 1694–1704, 2002.

[8] P. Munster, R. Sifta, T. Horvath, V. Novotny, and M. Filka. Polarization mode dispersion in NG- PON, in *In Fourth Forum of Young Researchers. In the framework of International Forum Education Quality*. Izhevsk, Russia: Publishing House, ISBN: 978–5–7526–0649–6, pp. 364–367, 2014.

[9] J. Mullerova, D. Korcek, and M. Dado, On wavelength blocking for XG-PON coexistence with GPON and WDM-PON networks, in *Proc. 14th International Conference on Transparent Optical Networks (ICTON)*, Coventry, pp. 1–4, 2012.

[10] A. M. Cailean, B. Cagneau, L. Chassagne, M. Dimian, and V. Popa, Miller code usage in Visible Light Communications under the PHY I layer of the IEEE 802. 15. 7 standard, in *Proc. 10th International Conference on Communications (COMM)*, Bucharest, pp. 1–4, 2014.

[11] J. Yu, Z. Jia, P. N. Ji, and T. Wang, 40 Gb/s WDM Passive Optical Network with Centralized Lightwave Source, in *Proc. OFC/NFOEC 2008 - 2008 Conference on Optical Fiber Communication/National Fiber Optic Engineers Conference*, San Diego, CA, pp. 1–3, 2008.

[12] B. Liu, L. Zhang, X. Xin, and J. Yu, Constellation–masked secure communication technique for OFDM-PON, in *Optics Express*, vol. 20, no. 22, pp. 25161, 2012.

[13] Y. Luo, X. Zhou, F. Effenberger, X. Yan, G. Peng, Y. Qian, and Y. Ma, Time and Wavelength–Division Multiplexed Passive Optical Network (TWDM-PON) for Next-Generation PON Stage 2 (NG-PON2), in *Journal of Lightwave Technology*, vol. 31, no. 4, pp. 587–593, 2013.

[14] Y. Kodama, The Whitham Equations for Optical Communications: Mathematical Theory of NRZ, in *SIAM Journal on Applied Mathematics*, vol. 59, no. 6, pp. 2162–2192, 1999.

Tomas Horvath was born in Havirov, Czech Republic on March 7, 1989. He received his M.Sc.degrees in Telecommunications from the Brno University of Technology, Brno, in 2013. His research interests include passive optical networks (xPON), optoelectronics and BitTorrent protocol. Currently, he has been actually post graduate student at Brno University of Technology, Department of Telecommunications and his topic of dissertation thesis is Optimization services in FTTx optical access networks. Technicka 12 - FEEC, BUT, Czech Republic 60190 or horvath@feec.vutbr.cz.

Radek Fujdiak was born in Czech Republic 1987. He received Bachelor (Teleinformatics) and Master degree (Telecommunication and Information Technique) in the Brno University of Technology (BUT). Now he is working on his Doctor Degree in Teleinformatics from the same University (BUT). His research interest include the mathematics, cryptology, smart grid and sensor networks. Technicka 12 - FEEC, BUT, Czech Republic 60190 or xfujdi00@phd.feec.vutbr.cz.

Milan Cucka Cucka was born in Vyskov (Czech Republic) 1989. He received his Bachelor Degree in Teleinformatics in 2012 and his master degree in Telecommunication and Information Technique in 2014. This titles received at Brno University of Technology. Now he is studiing his doctoral degree in the same university. His research is focused on optical fibers, mainly optical distributed systems which used this fibers, mainly optical distributed systems which used this fibers. Technicka 12 - FEEC, BUT, Czech Republic 60190 or xcucka00@phd.feec.vutbr.cz.

Marie Dankova graduated from the Mathematical engineering, Faculty of Mechanical Engineering, Brno University of Technology in 2014. She is currently working towards her PhD in Teleinformatics at Faculty of Electrical Engineering and Communication, Brno University of Technology. Technicka 12 - FEEC, BUT, Czech Republic 60190 or xdanko05@phd.feec.vutbr.cz.

Jiri Misurec received Master degree (Radioelectronics), CSc degree (Radioelectronics) and Doc. degree (Teleinformatics) in the Brno University of Technology (BUT). Now he is working as Department Leader in Department of Telecommunications BUT. His research interest include the signals, PLC, telecommunications, smart grid, power grid networks, networking and sensor networks. Antoninska 548/1 - BUT, Czech Republic 60190 or misurec@feec.vutbr.cz.

Multi–GPU Implementation of Machine Learning Algorithm using CUDA and OpenCL

Jan Masek, Radim Burget, Lukas Povoda and Malay Kishore Dutta

***Abstract*—Using modern Graphic Processing Units (GPUs) becomes very useful for computing complex and time consuming processes. GPUs provide high–performance computation capabilities with a good price. This paper deals with a multi–GPU OpenCL and CUDA implementations of *k*–Nearest Neighbor (*k*–NN) algorithm. This work compares performances of OpenCL and CUDA implementations where each of them is suitable for different number of used attributes. The proposed CUDA algorithm achieves acceleration up to 880x in comparison with a single thread CPU version. The common *k*-NN was modified to be faster when the lower number of *k* neighbors is set. The performance of algorithm was verified with two GPUs dual-core NVIDIA GeForce GTX 690 and CPU Intel Core i7 3770 with 4.1 GHz frequency. The results of speed up were measured for one GPU, two GPUs, three and four GPUs. We performed several tests with data sets containing up to 4 million elements with various number of attributes.**

***Keywords*—Artificial intelligence, big data, comparison, CUDA, GPU, high performance computing, *k*-NN, multi–GPU, OpenCL.**

I. Introduction

Parallel computing is a way how to accelerate many algorithms, which are computationally intensive. These algorithms can be found in image, sound and video applications or simulations, data mining, security [1], forecasting systems, etc.

k-NN belongs to the algorithms of artificial intelligence and it is one of the most widely used algorithms in data mining applications. Algorithm can be used for the classification of many various problems from business or science. Sometimes there is a requirement to process large datasets with high dimensional data. These problems can take days to compute. Using parallel computing, these problems can be solved faster than using non–parallel implementation. GPUs have much more cores than CPU, so they can be used as better solution for parallelization. The next advantage to use GPUs is relatively low price due to their high performance. *k*-NN algorithm is a good candidate for GPU parallelization.

In this paper an OpenCL [2] and CUDA [3] accelerated version of *k*–Nearest Neighbor machine learning algorithm has been introduced. This work is based on our previous work [4]. The algorithm is very computationally intensive mainly when big datasets with high dimensional data have to be processed. The process can take hours or days. To solve this problem, we modified common *k*–NN algorithm to run on multiple GPUs. We used two common gaming dual–core graphic cards NVIDIA GeForce GTX 690 [5] with 2x3072 CUDA cores in total. The theoretical single precision computing performance is 11.24 TFLOPS for both devices. We also used these GPUs to speed up Viola–Jones object detector [6], which was also used in [7] [8] [9].

The main contribution of this paper is the creation of the OpenCL and CUDA versions of *k*–NN algorithm, which can be executed on several GPU cards in parallel. Using this relatively cheap hardware, it is able to speed up computation up to 880 times in comparison with CPU with 4.1 GHz frequency. A newly created algorithms were tested on dataset containing millions of elements with various number of attributes (4, 10, 100 and 1000 attributes) and then algorithms were together compared.

The rest of the paper is organized as follows: Section II describes other GPU implementations of *k*–NN algorithm. Section III describes *k*–NN algorithm. OpenCL and CUDA platforms are introduced in section IV. In section V our GPU implementation is described. Results and discussion are described in section VI. Section VII concludes this paper.

II. Related work

GPU computing has become very popular during last several years. There is also increasing need to process more amount of data with artificial intelligence. The next paragraph describes several articles dealing with CUDA implementations of *k*-NN GPU algorithm and various use cases of the algorithm.

In [10] a new brute force algorithm for building the k-Nearest Neighbor Graph is described. The proposed algorithm has two parts, where the first is for finding distances between the input vectors and the second part is for selection of *k* neighbors for each testing sample. Also new algorithm based on quick sort was implemented for quicker sorting of distance pairs. The algorithm achieves higher speed up, if the *k* variable is increasing.

The paper [11] compares GPU implementation of brute force *k*-NN with several CPU based implementations and the implementation of algorithms from ANN[1] library (A Library

Manuscript received on November 18, 2015, revised June 9, 2016.

This research work was supported by the project of the Technology Agency of the Czech Republic - project EPSILON no. TH01010277.

J. Masek is with BurgSys, a.s., Hnevkovskeho 30/65, 617 00 Brno, Czech Republic, phone: +420728947018; e-mail: masek@burgsys.com

R. Burget is with the Brno University of Technology, Faculty of Electrical engineering, Department of Telecommunications, Czech Republic, e-mail: burgetrm@feec.vutbr.cz

L. Povoda is with the Brno University of Technology, Faculty of Electrical engineering, Department of Telecommunications, Czech Republic, phone: 541146962; e-mail: xpovod00@stud.feec.vutbr.cz

M. K. Dutta is with Amity University, Department of Electronics & Communication Engineering, Sector-125, 201 301 Noida, India (phone: 0120-2445252 / 4713600; e-mail: mkdutta@amity.edu

[1] Available from URL: http://www.cs.umd.edu/˜mount/ANN/

doi: 10.11601/ijates.v5i2.142

for Approximate Nearest Neighbor Searching).

In [12] [13] several optimization techniques were applied to maximize the utilization of the GPU.

The work [14] describes MST (minimum spanning tree) problem, which is resolved by the combination of classical Boruvka MST algorithm and the *k*-NN graph structure. Achieved speed-ups were between 30 and 40 in comparison with CPU implementation.

In [15] authors describe how to use GPU *k*-NN algorithm for image processing (texture analysis). Their algorithm is 150 times faster than CPU version during processing synthetic data and up to 75 times faster during processing image data.

In [16], the LSH (Locality Sensitive Hashing) algorithm was used for *k*-NN computation. The results were demonstrated on large image datasets and achieved acceleration was 40 in comparison with CPU version.

Several comparison tests between OpenCL and CUDA frameworks were performed. In [17] authors performed 16 benchmark tests where CUDA achieves for about 30 % better performance than OpenCL. Further they tried OpenCL portability and they did not found differences in performance. In [18] authors compared executive time of CUDA Drive API with OpenCL platform where CUDA was for about 5 % faster than OpenCL.

The implementation of GPU *k*-NN algorithm into RapidMiner[2] data mining platform was described in [19]. The algorithm was created in JAVA programming language with using jcuda[3] library that is responsible for executing CUDA kernel from JAVA. Created algorithm achieves 170x speedup, but it depends mainly on number of used attributes (using more than 128 attributes decreases the speedup).

Our approach differs from using OpenCL platform instead of CUDA and our algorithm has several improvements in comparison with some approaches described in this paragraph. The main improvement is an option to run our algorithm on multiple GPUs in parallel. The created OpenCL kernel was partially vectorized and the algorithm was created without need to have some sorting algorithms. These improvements speed up the algorithm. Our solution was tested on very large data set, where the processing time was minutes against other works, where processing quite small datasets took seconds.

III. *k*-Nearest Neighbor algorithm

k–NN algorithm can be used for classification or regression. The principle of *k*–NN is shown in Fig. 1. The input of algorithm are training examples and testing examples. For each testing example, the distance (Euclidean, Manhattan, etc.) of attributes between testing and training example is computed. The distances are computed for the one testing example and all training examples. Then the distances are sorted according to their values. The training examples with *k* lowest differences are selected as the nearest neighbors. According to their classification classes, the testing example is classified. Usually the lower number of *k* value is set.

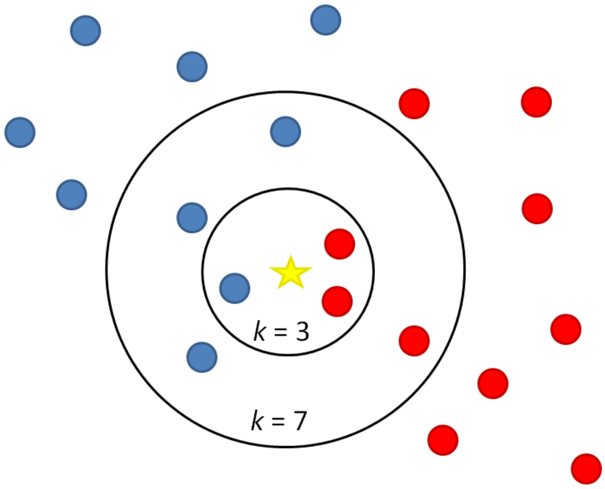


Fig. 1. The principle of *k*–NN algorithm.

IV. OpenCL and CUDA introduction

Nowadays there exist two platforms for GPU computing that are well used by many users. The first developed platform is CUDA [3] and the second is OpenCL [2]. CUDA is being more used but on the other hand CUDA can only be used with NVIDIA GPUs. OpenCL is being used less than CUDA but OpenCL can be performed on many various devices.

When compared these GPU platforms with common CPU solution, GPU hardware is much more specialized for intensive highly parallel computing. It can be seen from Fig. 2 and Fig. 3 where ALU (Arithmetic Logic Unit) elements are used for computing. The GPU hardware can process much more computing units in parallel than CPU.

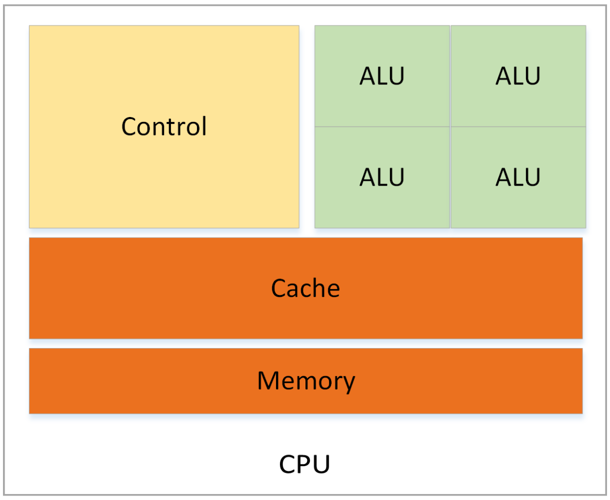


Fig. 2. Scheme of CPU

A. OpenCL

OpenCL (Open Computing Language) [2] is an open royalty-free standard determined for parallel programming of suitable devices like CPUs, GPUs and the other devices. OpenCL can solve many problems more efficiently than CPU.

[2] Available from URL: http://rapidminer.com

[3] Available from URL: http://www.jcuda.org/

In these days, the GPU computing is very popular and many applications have been developed in OpenCL.

There are two types of OpenCL code. The first type executed on CPU is called host part and the second one that is executed on OpenCL device is called device part. OpenCL kernel is executed in a device. The kernel can contain optimized code with OpenCL functions. OpenCL devices use SIMT (Single Instruction Multiple Threads) architecture. OpenCL device consists of Streaming Multiprocessors (SMs) where each of them contains many simple cores. These cores are able to do only simple operations, so OpenCL programming is more complex. Cores can execute many work-items (threads) in parallel. Work-items are grouped into work-group and they can mutually communicate and use the same (shared) local memory. The number of work-groups and work-items has to be set on the start of the process. OpenCL device contains on–chip and off–chip memories. On–chip (private, local) memories are faster than off-chip (global memory, constant memory, texture cache). However some of these memories can be fast too, because they are cached. [20]

B. CUDA

CUDA [21] is a parallel computing and programming platform and only newer NVIDIA graphic cards are supported by CUDA. Nowadays, there exist many GPU computing applications developed in CUDA for example deep learning algorithm [22] that is used for training neural network for image recognition.

Common CUDA GPU uses same principles that are described in OpenCL section. There are only different names of therms (shown in Table I). In CUDA CPU is marked as a host and GPU is marked as a device.

V. Optimization *k*-NN for GPUs

Firstly, we tried to process large data sets in RapidMiner, but unfortunately the original CPU version of *k*-NN was too slow. So we decided to create GPU accelerated algorithm that can be executed from RapidMiner. Our implementation was created in JAVA programming language, because RapidMiner is also programmed in JAVA. The first step was to create OpenCL kernel that was created in C programming language with using OpenCL syntaxes. For mutual cooperation between OpenCL and JAVA, jocl[4] library was used [21]. According to OpenCL kernel we created CUDA version of this kernel using CUDA version 7.5 and jcuda[5] library that was used as JAVA wrapper.

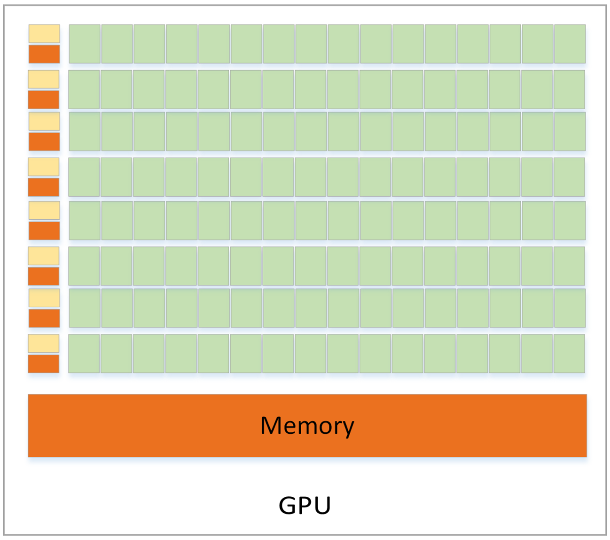


Fig. 3. Scheme of GPU

Training and testing data sets have to be transformed into float arrays before they are transmitted to GPU. We used OpenCL vector format called float4 that has a big advantage: it contains four float values that are processed in one step instead of four steps (for common float). So every training and testing example is saved into float4 array. We also optimized kernel with using local memory.

In our implementation, the classical principle of *k*-NN was a little bit modified. The differences are mutually compared during their computation and the lowest *k* differences are saved as final nearest neighbors. The algorithm 1 shows the principle of modified algorithm. After this modification, the algorithm can work faster for lower *k* values. When compared with CPU version, the classification results were the same.

A. Multi–GPU support

For multi–GPU support we created a JAVA library that is able to utilize all found GPUs. This library is available only for OpenCL. The library can automatically split input and output data and transmit them equally into all devices. It decreases amount of transmitted data. The next advantage is a very easy way, how to write code in JAVA with minimum knowledge of OpenCL. For multi–GPU support for CUDA platform we had to run split data into GPUs and start computing on each GPU in separated JAVA thread in parallel.

In case of *k*-NN algorithm, the training data vector had to be transmitted into all GPU devices. Testing data vector and vector with final predictions were splitted equally into all device due to lower load of GPU memory. We also tried the version of algorithm, where data were not splitted into GPU devices, but they were copied whole to each device. Differences between computing times of each version were negligible.

[4] Available from URL: `http://www.jocl.org/`
[5] Available from URL: `http://www.jcuda.org/`

TABLE I
CUDA and OpenCL terminology mapping

CUDA	OpenCL
Grid	NDRange
Thread Block	Work–group
Thread	Work item
Thread ID	Global ID
Block index	Block ID
Thread index	Local ID
Shared Memory	Local Memory
Local Memory	Private Memory

```
Input: training data (float4), testing data (float4)
Output: prediction vector
loading test sample from testing data;
foreach train sample do
    foreach attribute do
        compute distance between train and test samples;
        sum distances;
    end
    for $k = 0$ to number of neighbors do
        if sum of distances < distance for k neighbor then
            distance for k neighbor = sum of distances;
            shift distance values for other k distances;
            break;
        end
    end
end
counting number of neighbors for each label;
selecting label with highest number of neighbors;
setting prediction for test sample to prediction vector;
```

Algorithm 1: *k*-NN OpenCL algorithm

VI. Results and discussion

We performed several comparison tests to verify the functionality of our accelerated *k*-NN algorithms. The tests were performed with data sets containing different amount of elements and different numbers of attributes. Since the algorithm has been modified to have a good result for lower *k* parameter ($k = 5$), we also carried out several tests for higher values of *k* parameter ($k = 10, k = 20$). For a comparison between CPU and GPU versions, we used RapidMiner platform that consists of many machine learning and data–mining algorithms. First, we generated polynomial data set using one of the RapidMiner algorithms. Then this data set was divided into two parts. The training part contained 25 % of elements and testing part contained 75 % of elements. In the next step, several tests with different number of elements and different number of attributes were performed. We used CPU version of *k*-NN algorithm integrated in RapidMiner and our GPU versions of *k*-NN that were also executed in RapidMiner.

TABLE II
OpenCL - comparing for different *k* neighbors.

	$k = 1$	$k = 10$	$k = 20$
1 million, 10 attributes.	14.1 s	25.2 s	93.7 s
1 million, 4 attributes.	4.1 s	17.8 s	88 s

TABLE III
CUDA -comparing for different *k* neighbors.

	$k = 1$	$k = 10$	$k = 20$
1 million, 10 attributes.	16 s	21.7 s	88.5 s
1 million, 4 attributes.	3.8 s	16.2 s	78.8 s

Our measurements were performed in computer with CPU Core i7-3770 4.1 GHz (in boosted mode), 32 GB RAM and two dual-core NVIDIA GeForce GTX 690 [5] graphic cards that are very powerful in single precision mode. Every GTX 690 consists of two GPU cores and every GPU core has 8 Streaming Multiprocessors and each SM consists of 192 CUDA cores (1536 CUDA cores in total). The size of a GPU memory is 4096 MB with 6 GHz frequency. GTX 690 has theoretical performance 5.62 TFLOPS in single precision mode. When both GPU cards are used, the performance of system is 11.24 TFLOPS. The measured power consumption of one GPU GTX 690 was 300 watts.

The tests were performed with using one core of CPU, one GPU, 2 GPUs, 3 GPUs and 4 GPUs. The results show how much time each scenario took and they are described in the Table IV for OpenCL implementation and in Table V for CUDA implementation. In this case the measurements were performed for $k = 5$. As we can see from table some CPU computations can take days in comparison with GPU computation, where it takes minutes. Fig. 4 shows speed up of our OpenCL GPU implementation of *k*–NN algorithm. We can see that increasing amount of attributes can decrease speed up. Speed up can be also increased if higher number of elements is used. Scenarios for CUDA implementation are shown in Table V. The overall speed up of CUDA implementation of *k*–NN algorithm is shown in Fig. 5. The best speed up was achieved in scenario with 1 million of elements and 4 attributes where achieved speed up was 882 times. The comparison between CUDA and OpenCL implementations is shown in Fig 6. We can see that for scenarios with number of attributes 100 and 1000, CUDA was for about 3 % faster than OpenCL. For scenarios with 10 attributes OpenCL implementation was faster for about 11 %. And for 4 attributes CUDA was for about 18 % faster than OpenCL. These differences in speed up when 4 or 10 attributes are used, can be caused with using float4 data type for storing array of attributes where CUDA can handle much more better with 4 attributes in one float4 array than with 10 attributes in 3 float4 arrays where two elements in array are not used. When computing the average value of all scenarios, CUDA was for about 0.5 % faster.

The table II shows the results for using different values of *k* (measured for all GPUs). Our OpenCL implementation has been created to work effectively with the number of *k* neighbors lower than 10. Otherwise, the speed up of algorithm will be radically decreased. In comparison with CUDA implementation (see Table. III) we can see that CUDA is slightly faster than OpenCL implementation.

VII. Conclusion

The main contribution of this work is OpenCL accelerated implementations of *k*-Nearest Neighbor machine learning algorithm with using OpenCL and CUDA. The algorithm can be executed on multiple GPUs in parallel. We created the modified version of algorithm that achieves very good results for *k* neighbors lower than 10. We found that with using relatively cheap hardware (2x NVIDIA GeForce GTX 690), it is possible to compute 4 million elements (each has 10 attributes) in 3 minutes in comparison with using one single core CPU

TABLE IV
RESULTS OF COMPUTATION FOR OPENCL IMPLEMENTATION - $k = 5$.

	1 CPU	1 GPU	2 GPUs	3 GPUs	4 GPUs
0.4 million, 1000 attributes	17h 35min	843 s	434 s	295 s	228 s
2 million, 100 attributes	2d 4h 10min	2106 s	1056 s	707 s	532 s
1 million, 10 attributes	1h 55min	40.2 s	21.7 s	15.2 s	11.5 s
4 million, 10 attributes	1d 7h 29min	645 s	332 s	223 s	169 s
1 million, 4 attributes	1h 12min	21 s	11.1 s	7.5 s	5.8 s

CPU - Intel Core i7 3770@4.1GHz, L3 cache - 8192kB
4 GPU - 2x3072 cores, mem. 2x4096MB@6 GHz, GPU - 1019 Mhz

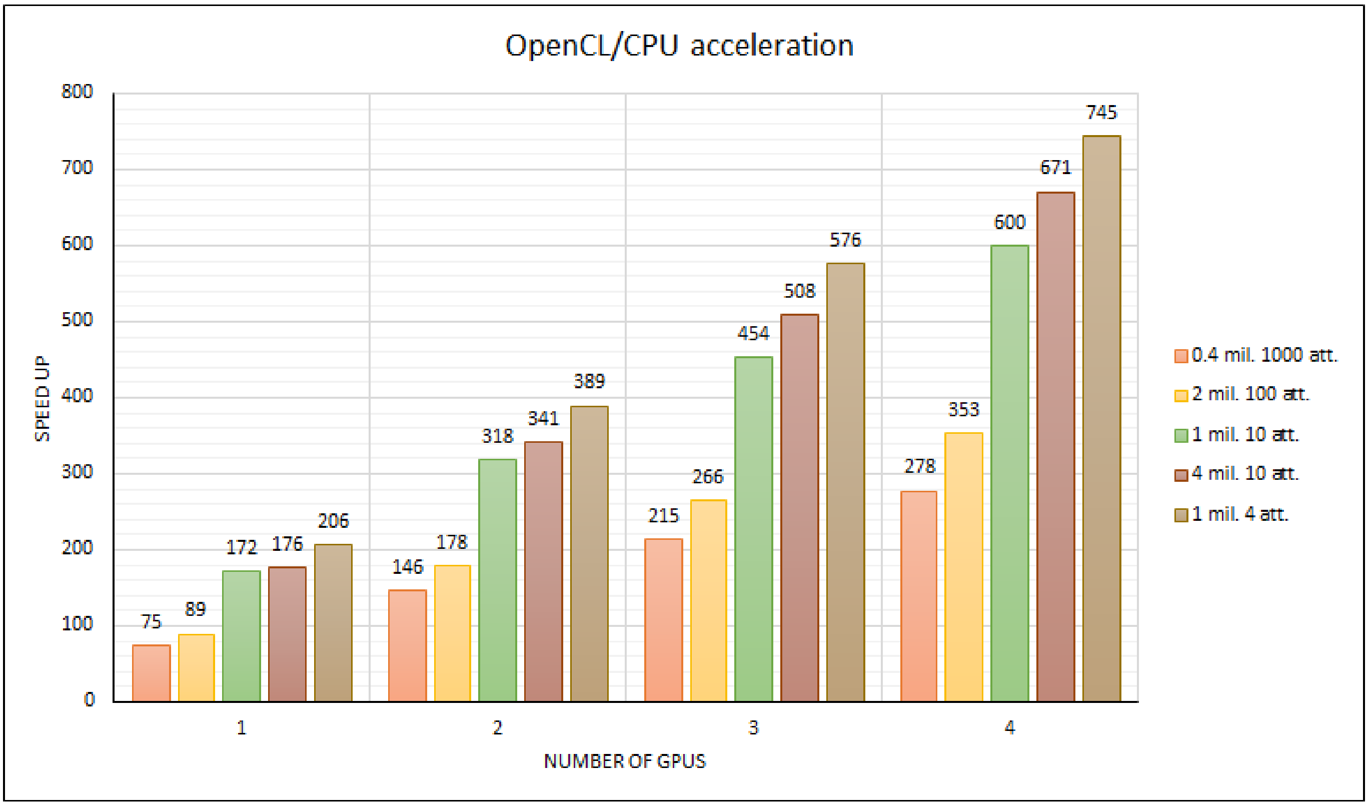


Fig. 4. Acceleration for OpenCL version of k-NN algorithm.

TABLE V
RESULTS OF COMPUTATION FOR CUDA IMPLEMENTATION- $k = 5$.

	1 CPU	1 GPU	2 GPUs	3 GPUs	4 GPUs
0.4 million, 1000 attributes	17h 35min	818 s	419 s	286 s	218 s
2 million, 100 attributes	2d 4h 10min	2038 s	1023 s	683 s	523 s
1 million, 10 attributes	1h 55min	47.3 s	23.9 s	15.9 s	12.6 s
4 million, 10 attributes	1d 7h 29min	757 s	377 s	252 s	196 s
1 million, 4 attributes	1h 12min	18 s	9.3 s	6.4 s	4.9 s

CPU - Intel Core i7 3770@4.1GHz, L3 cache - 8192kB
4 GPU - 2x3072 cores, mem. 2x4096MB@6 GHz, GPU - 1019 Mhz

(Intel Core i7–3770, 4.1GHz), where the computation took over 31 hours. The best achieved acceleration was up to 880x. Our algorithms were created in JAVA programming language and they have been implemented in to the RapidMiner data mining platform. The most time consuming part of algorithm has been created in OpenCL and CUDA. For mutual cooperation between JAVA and OpenCL or CUDA, the jocl and jcuda libraries were used. When compared OpenCL and CUDA implementations, CUDA has better results for data set containing 4 attributes or with 100 or 1000 attributes, but

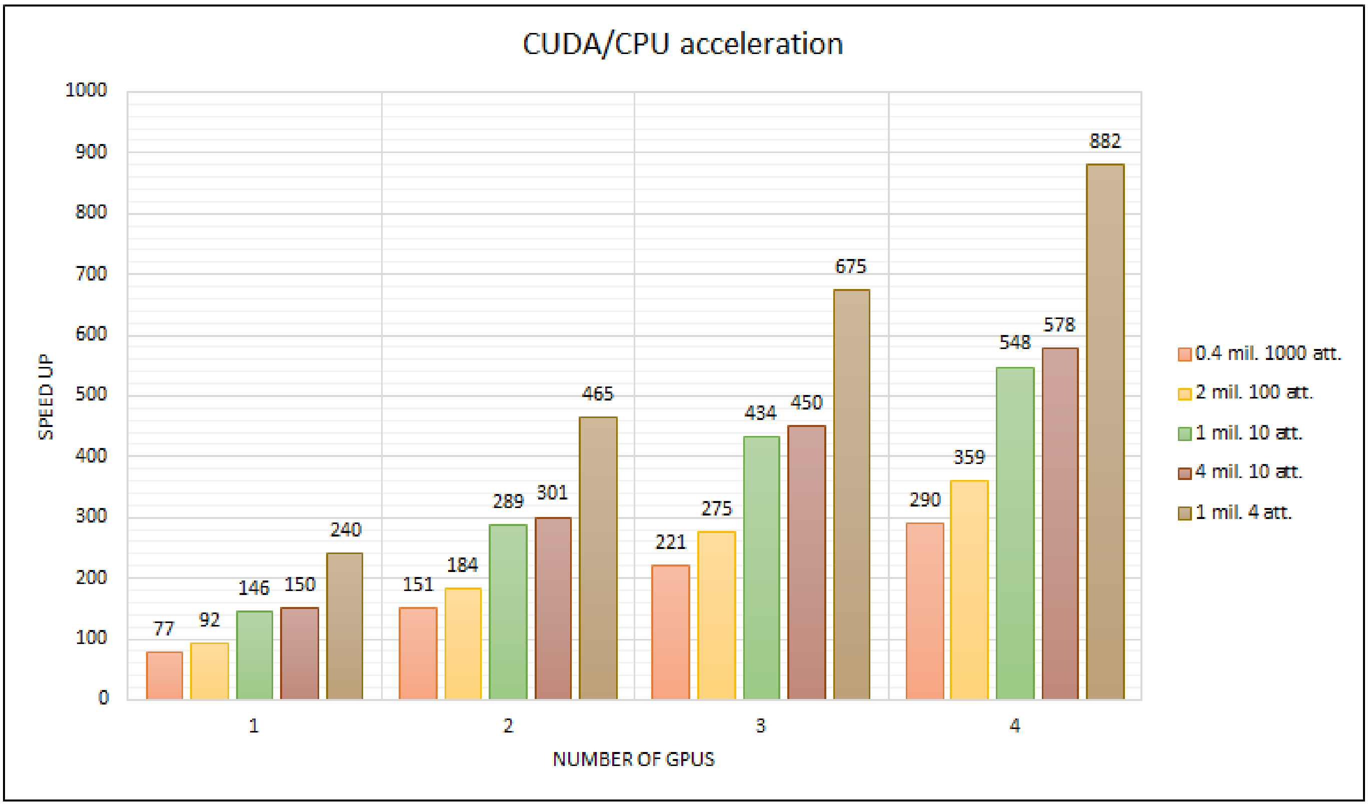


Fig. 5. Acceleration for CUDA version of *k*-NN algorithm.

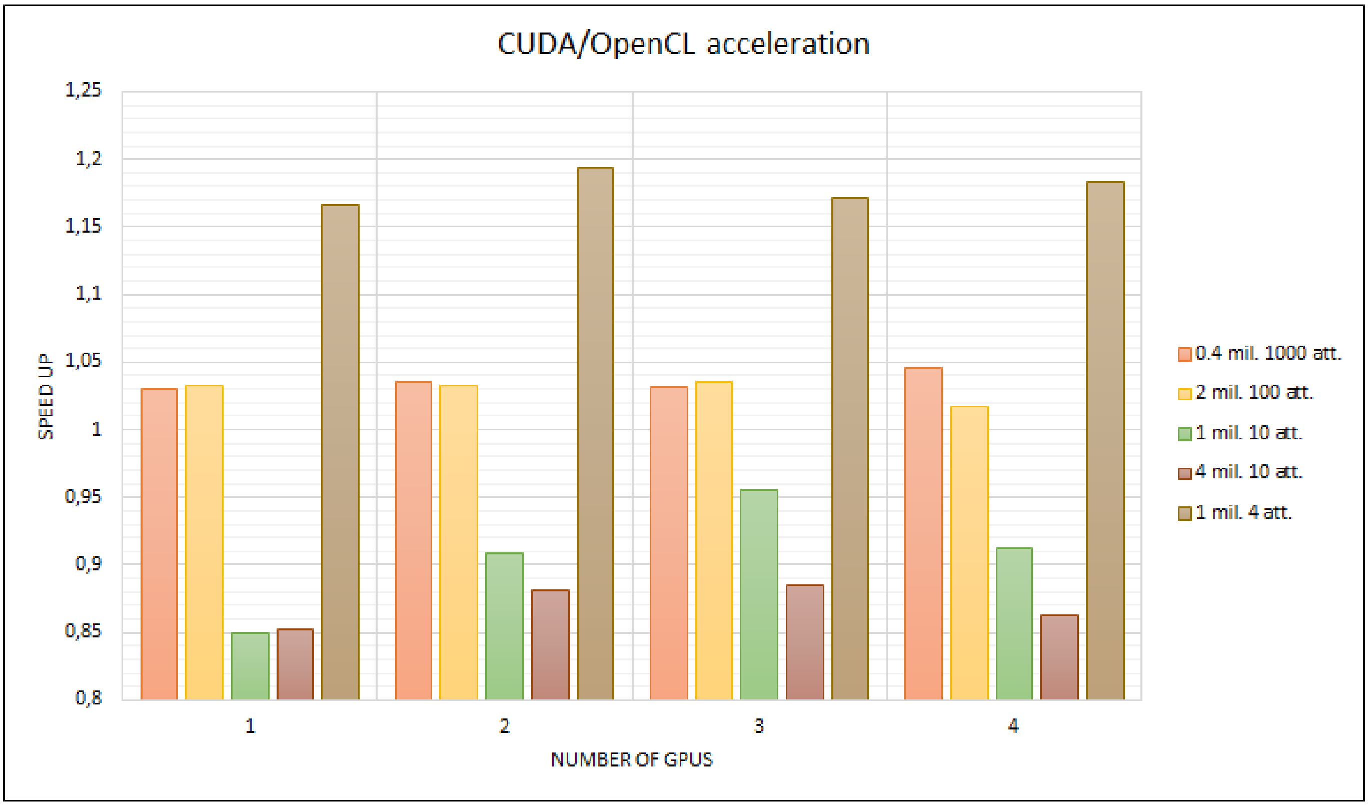


Fig. 6. Speed up comparison between CUDA and OpenCL implementations of *k*-NN algorithm.

on the other hand OpenCL has better result with using data set with 10 attributes. When compared overall results, both OpenCL and CUDA achieves similar speed up.

References

[1] J. Minar, K. Riha, H. Tong, "Intruder Detection for Automated Access Control Systems with Kinect Device,"*In 2013 36th International Conference on Telecommunications and Signal Processing (TSP),* 2013, pp. 826-829, ISBN 978-1-4799-0403-7.

[2] Khronos OpenCL Working Group,"The OpenCL Specification - Version: 1.1,"2011, Available: `http://www.khronos.org/registry/cl/specs/opencl-1.1.pdf`

[3] NVIDIA Inc,"CUDA Toolkit 7.5,"2015, Available: `https://developer.nvidia.com/about-cuda`

[4] J. Masek, R. Burget, J. Karasek, V. Uher, M.K. Dutta, "Multi-GPU implementation of k-nearest neighbor algorithm,"*In 2013 36th International Conference on Telecommunications and Signal Processing (TSP),* 2015, pp. 764-767, ISBN 978-1-4799-8497-8.

[5] NVIDIA, 2013, February 5. "GeForce Hardware."Available: `http://www.geforce.com/hardware/desktop-gpus/geforce-gtx-690`

[6] J. Masek, R. Burget, V. Uher, S. Guney,"Speeding up Viola–Jones algorithm using multi–Core GPU implementation,"*In 2013 36th International Conference on Telecommunications and Signal Processing (TSP),* 2013, pp. 808-812, ISBN: 978-1-4799-0402-0.

[7] K. Riha, J. Masek, R. Burget, R. Benes, E. Zavodna, "Novel method for localization of common carotid artery transverse section in ultrasound images using modified Viola–Jones detector,"*Ultrasound in Medicine & Biology,* 2013, pp. 1887-1902, ISSN 0301-5629.

[8] R. Burget, P. Cika, M. Zukal, J. Masek,"Automated Localization of Temporomandibular Joint Disc in MRI Images,"*In 2011 34th International Conference on Telecommunications and Signal Processing (TSP),* 2011, pp. 413-416, ISBN: 978-1-4577-1409-2.

[9] J. Masek, R. Burget, J. Karasek, V. Uher, S. Guney,"Evolutionary Improved Object Detector for Ultrasound Images,"*In 2013 36th International Conference on Telecommunications and Signal Processing (TSP),* 2013, pp. 586-590, ISBN: 978-1-4799-0402-0.

[10] I. Komarov, A. Dashti, R. D Souza,"Fast *k*-NNG construction with GPU-based quick multi-select,"2013.

[11] V. Garcia, E. Debreuve, M. Barlaud,"Fast k Nearest Neighbor Search using GPU,"*In IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops,* 2008, pp. 1-6.

[12] S. Liang, Y. Liu, Ch. Wang, L. Jian,"A CUDA-based Parallel Implementation of K-Nearest Neighbor Algorithm,"*In Cyber-Enabled Distributed Computing and Knowledge Discovery,* 2009, pp. 291-296.

[13] Q. Kuang, L. Zhao,"A Practical GPU Based KNN Algorithm,"*In Proceedings of the Second Symposium International Computer Science and Computational Technology (ISCSCT 09),* 2009, pp. 151-155, ISBN: 978-952-5726-07-7.

[14] A. Arefin, C. Riveros, R. Berretta, and P. Moscato,"kNN-Boruvka-GPU: a fast and scalable MST construction from kNN graphs on GPU,"*In ICCSA'12 Proceedings of the 12th international conference on Computational Science and Its Applications - Volume Part I,* 2012, pp. 71-86, ISBN: 978-3-642-31124-6.

[15] V. Garcia, F. Nielsen,"Searching High-Dimensional Neighbours: CPU-Based Tailored Data-Structures Versus GPU-Based Brute-Force Method,"*In Lecture Notes in Computer Science, Springer Berlin Heidelberg* 2009, pp. 425-436, ISBN: 978-3-642-01810-7.

[16] J. Pan D. Manocha,"Fast GPU-based locality sensitive hashing for k-nearest neighbor computation,"*In Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems,* 2011, pp. 211-220, ISBN: 978-1-4503-1031-4.

[17] F. Jianbin, A.L. Varbanescu, H. Sips,"A Comprehensive Performance Comparison of CUDA and OpenCL,"*In Conference of Parallel Processing (ICPP)* 2011, pp.216-225, ISBN: 978-1-4577-1336-1.

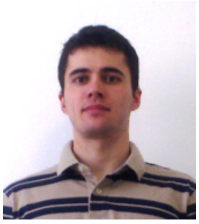
[18] Ch.L. Su, P.Y. Chen, CH.CH Lan,L.S Huang, K.H. Wu,"Overview and comparison of OpenCL and CUDA technology for GPGPU,"*Circuits and Systems (APCCAS), 2012 IEEE Asia Pacific Conference on* 2012, pp. 448-451, ISBN: 978-1-4577-1728-4.

[19] A. Kovacs, Z. Prekopcsak,"Robust GPGPU plugin development for RapidMiner,"*In RapidMiner Community Meeting And Conference - RCOMM 2012,* 2012.

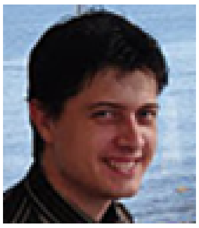
[20] A. Munshi, B. Gaster, T. Mattson, J. Fung, D. Ginsburg,"OpenCL Programming Guide,"2011.

[21] NVIDIA,"OpenCL Programming Guide for the CUDA Architecture,"2010, Available: `http://www.nvidia.com/content/cudazone/download/OpenCL/NVIDIA_OpenCL_ProgrammingGuide.pdf`

[22] NVIDIA Inc,"GPU-Based Deep Learning Inference: A Performance and Power Analysis,"2015, Available: `http://www.nvidia.com/content/tegra/embedded-systems/pdf/jetson_tx1_whitepaper.pdf`.



Jan Masek is Ph.D. student at the Department of Telecommunications, Faculty of Electrical Engineering, Brno University of Technology, Brno, Czech Republic. He obtained his MSc. in 2012 (Communications and Informatics). He is interested in image processing, data mining, parallel systems.



Dr. Radim Burget is associated professor at the Department of Telecommunications, Faculty of Electrical Engineering, Brno University of Technology, Brno, Czech Republic. He obtained his MSc. in 2006 (Information Systems) and his finished his Ph.D. in 2010. He is associated professor since 2014. He is interested in image processing, data mining, genetic programming and optimization.

Lukas Povoda is Ph.D. student at the Department of Telecommunications, Faculty of Electrical Engineering, Brno University of Technology, Brno, Czech Republic. He obtained his MSc. in 2014 (Communications and Informatics). He is interested in image processing, text processing, and genetic programming.

Professor Dr. Malay Kishore Dutta is professor at Amity University, Department of Electronics & Communication Engineering, where he is Additional Director (ADET), Joint Acting Head (ASET), Professor & HOD ECE. He is interested in Digital Watermarking & Encryption techniques of multimedia signals, Image Processing, Computer Vision & Pattern Recognition, Audio Signal processing, Feature extraction from audio signals and its application, Time-frequency analysis of Signals using STFT & Multi-resolution methods like Wavelet Decomposition, Medical Image Segmentation, Fundus Image analysis.

The HSBQ Algorithm with Triple-play Services for Broadband Hybrid Satellite Constellation Communication System

Anupon Boriboon and Siriwhaddhanah Pongpadpinit

Abstract—**The HSBQ algorithm is the one of active queue management algorithms, which orders to avoid high packet loss rates and control stable stream queue. That is the problem of calculation of the drop probability for both queue length stability and bandwidth fairness. This paper proposes the HSBQ, which drop the packets before the queues overflow at the gateways, so that the end nodes can respond to the congestion before queue overflow. This algorithm uses the change of the average queue length to adjust the amount by which the mark (or drop) probability is changed. Moreover it adjusts the queue weight, which is used to estimate the average queue length, based on the rate. The results show that HSBQ algorithm could maintain control stable stream queue better than group of congestion metric without flow information algorithm as the rate of hybrid satellite network changing dramatically, as well as the presented empiric evidences demonstrate that the use of HSBQ algorithm offers a better quality of service than the traditionally queue control mechanisms used in hybrid satellite network.**

Keywords—**Active Queue Management (AQM), Broadband Hybrid Satellite Constellation Communication System (BHSCCS), COMMStellation^{TM}, Hybrid Satellite – Blue Queue (HSBQ), Queue.**

I. Introduction

In the next-generation Global Area Network (GAN), satellite networks are expected to complement the coverage of the globalized networks in instances where they best serve the communication needs of users. In recent years, Micro Systems Canada Inc. (MSCI) [1] has been deployed to provide firm global coverage communications to the public. To increase the feasibility of success of commercial high-speed broadband data satellite networks, it is important for designers to consider both wired and wireless techniques integration together. The initiative of the COMMStellation^{TM} by MSCI is to supply, produce, deploy, and perform the COMMStellation^{TM} system on satellite constellation topology over Low Earth Orbit (LEO). The concept provides global backhaul and connects rural areas. Each COMMStellation^{TM} [2] satellite will provide high speed, high bandwidth communications to four individual user service provider terminals, with each satellite supporting a total data throughput of up to 8.8 Gbps (1.1 Gbps per user service, both uplink and downlink) over 1.6 GHz of bandwidth. For the ground side of the link both types are Trunk Stations and User Stations. Trunk Stations that always connect to fiber infrastructure will transfer data Internet service with mission concept is bent-pipe configuration. User Stations aims to be a new high-bandwidth Internet Service Providers (ISPs), allowing individual clients to connect to its satellite link on the globe. The COMMStellation^{TM} is designed to provide long-haul wireless broadband global area network communication in an area where optical fiber networks does not exist and large and/or island geographic areas [3].

The Internet wide application which is working with the rapid development of the satellite technique, the hybrid satellite network has been becoming a research focus. For the challenges, efficient utilization of the capital investment in the face of high uncertainty in traffic demand needs to be considered during the hybrid satellite network design stage. A queue sizes oscillations large range make the queue suffer, which will make early congestion notification hard and finally lead to congestion [4].

Recent research activities have placed much emphasis on various the queue weight techniques for the active queue management over satellite segment assuming bottleneck link are known. A variety of Hybrid Satellite – Blue Queue (HSBQ) algorithm aims to evaluate for triple-play service applications over Broadband Hybrid Satellite Constellation Communication System (BHSCCS) [5] using the network system based on COMMStellation^{TM} satellite system has been proposed in this research. This paper proposes a rate-based stabilization for HSBQ. This algorithm adjusts the queue weight, which is used to estimate the average queue length, based on the rate; moreover, it uses the changes of the average queue length to adjust the amount by the mark or drop of probability which is changed.

This paper is organized as follow: Section II describes the related works which is queue system, Section III is the proposed HSBQ algorithm model, the network topology system module in NS-2 is described in Section IV, Simulation results is presented in Section V, whereas our conclusions are drawn in Section VI. Simulation results show that this algorithm could maintain higher throughput

Manuscript received March 16, 2015, revised July 2, 2016.

Anupon Boriboon is with the Vincent Mary School of Science and Technology, Assumption University, Bangkok, Thailand (corresponding author to provide e-mail: anupon_b@hotmail.com).

Siriwhaddhanah Pongpadpinit is with the Department of Business Information System, Martin de Tours School of Management and Economics, Assumption University, Bangkok, Thailand (e-mail: winyu@scitech.au.edu).

doi: 10.11601/ijates.v5i2.169

than BLUE algorithm as the rate of hybrid satellite network system.

II. Related Works

In this section, we summarize related works. Queues represent locations where packets may be held (or dropped). Packet scheduling refers to the decision process used to choose which packets should be serviced or dropped. Buffer management refers to any particular discipline used to regulate the occupancy of a particular queue. In the common case where a delay element is downstream from a queue, the queue may be blocked until it is re-enabled by its downstream neighbor. This is the mechanism by which transmission delay is simulated. In addition, queues may be forcibly blocked or unblocked at arbitrary times by their neighbors (which is used to implement multi-queue aggregate queues with inter-queue flow control). Packet drops are implemented in such a way that queues contain a drop destination; that is, an object that receives all packets dropped by a queue. This can be useful to keep statistics on dropped packets [6], [7].

In the past few years numerous AQM algorithms have been developed and improved for solving the Internet congestion that appears in routers. The issues that should be considered in AQMs are various problems such as lock-out, global synchronization, and fairness. These AQMs used a concept mixture to solve all of the problems above. In analysis AQM schemes were classified to perform the analysis of AQMs with ease. Based on the classification, in general most of the AQMs employed only congestion metric to detect the congestion. However some of the AQMs required additional flow information other than the congestion metric to know the accurate status of the queue. Few of the AQMs required only spotting out the flow information. Considering these AQMs relevant to classification, the first category AQMs based only on congestion metric without flow information such as RED, DS-RED, LRED, HRED, Yellow, REM, BLUE were more simple and easy to design compared to the second category AQMs based on congestion metric with flow information. Therefore, the second category AQMs such as FRED, CHOKe, FAB, SFB also required extra overhead and implementation compared to the first category AQMs. The third category AQMs has a still greater complexity such as SRED, GREEN in identifying the flow information for calculating the marking probability [8].

III. Proposed the Hybrid Satellite – Blue Queue Algorithm

The proposal of AQM algorithm is to maintain high link utilization with low loss late and queuing delay in a wide range of network environments. Stable operation and quick response to changes in load are also desirable. This section outlines a proposed algorithm and explains its design in terms of these attributes. We begin with a general discussion of how the proposed algorithm provide control of queuing delay, then provides description of the adjustment of algorithm adapting to changes.

BLUE [9] uses packet loss and links idle events to manage congestion to give a very high throughput and high utilization with low queue length stability. For the better quality of high link utilization and throughput, it should be accessible by Load-based AQMs instead of Queue-based AQMs. Based on the research it indicates that the additional flow information is unnecessary to give attention. In the routers, the congestion is brining awareness get a better strength by Load-based AQMs. During, the comparison of congestion indicated the design of Queue-based AQMs is simple but some except for the parameter tunes for suitable of each problem compared with the other AQMs. In term of satellite high throughput and utilization, the Load-based AQMs performed better than Queue-based by flow information in AQMs.

In AQM schemes, comparative is based on performance metrics of [8], it appears as moderate of loss rate and low queue stability for the BLUE queue. Hence, we have to modify this point to improve BLUE algorithm to make it better for broadband hybrid satellite network system. Also, loss rates are low, and queue stability is higher. It aids to support more link utilization, and throughputs are very high.

Generally of probability which drop the packets before the queues overflow at the space nodes and/or ground nodes in system. An HSBQ (Hybrid Satellite – Blue Queue) is an active queue management algorithm for broadband hybrid satellite network system, which orders to avoid high packet loss rates and control stable stream queue. By-product from this modification it is reduced queue length. The motivations for removing unfairness while maintaining queue length stability is that when the HSBQ scheme decides to discard packets in order to decrease the input traffic before the queues overflow.

We adopt Z parameter from Yellow queue [10] to modify our proposal, the load factor Z will calculate during each averaging interval. The function of Virtual Available Capacity $\hat{c}$ can be updated based on the queue control function, which following [10] Yellow algorithm.

The smoothing parameter (α) and the desired utilization parameter (δ) are adopted from AVQ algorithm [11] that means we get a theorem to modify our proposal.

Also, toward the HSBQ proposal it has been modified based on BLUE active queue management algorithm, then

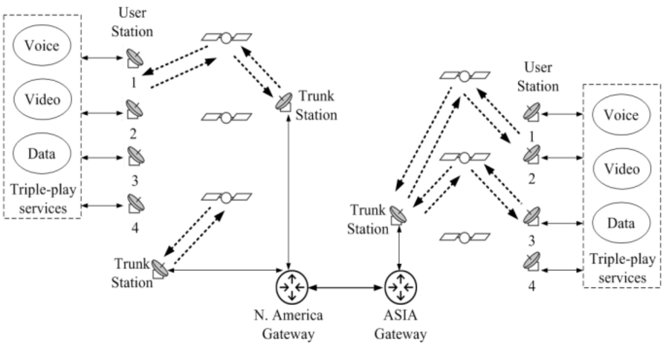


Fig. 1. Broadband hybrid satellite constellation communication system layout.

we have combination with smoothing parameter, desired utilization parameter, and load factor. HSBQ maintains a single probability P_m to mark or drop packets, which the buffer overflows HSBQ algorithm increase P_m to increase the congestion notification and decrease to reduce the congestion notification rate in case of buffer emptiness. Hence, a pseudocode of HSBQ is

```
Upon packet loss ($Q_{len} > L$) event:
  if ( ( ( ( $currently - recent\_update$ ) $\times \beta \times \delta \times \alpha$ ) / C ) > $hold\_time$ )
      $P_m = P_m + \delta_1 \times Z$
      $recent\_update = currently$

Upon link idle event:
  if ( ( ( ( $currently - recent\_update$ ) $\times \beta \times \delta \times \alpha$ ) / C ) > $hold\_time$ )
      $P_m = P_m - \delta_2 / Z$
      $recent\_update = currently$
```

Constant:
hold_time : minimum time period between two consecutive update of p_m
P_m : to either mark or drop the packets
δ_1 : incremented
δ_2 : decremented
β : buffer size
δ : desired utilization of the link
α : smoothing parameter
C : capacity of link
Z : load factor

IV. Implement with Broadband Hybrid Satellite Networks

In hybrid satellite communication networks, different restrictions and requirements on different links have to be taken into account. Thus, we have selected a typical representative of LEO walker-type satellite constellations with global area network, the space section is COMMStellation^{TM} satellite constellation [1], [2], [3], [5] composed of 72 operational satellites plus spares divided into 6 polar orbital planes at an altitude of 1000 km and inclination angle of 90°. The ground section of the link has two types of stations: i) Trunk Stations are 35 based on ground for connected to the high-speed Internet backbone and ii) User Stations. A given User Station is planned to be current or new Internet Service Provider (ISP), which allows individual clients to connect to its satellite link. The satellite constellation parameters are shown in [3], [5]. The satellite system can be linked with the high-speed Internet and other networks by using suitable gateway.

Figure 1 shows the network architecture of the broadband hybrid satellite constellation communication system (BHSCCS) [5] network with two segments: space, ground. We have two gateways connected through a North America and Asia link the bottleneck link. Each User Station pair sends triple-play services packets to destinations. In the Figure 1 illustrates the BHSCCS topology which uses in this work. The RED, BLUE, DropTail, and proposed HSBQ queuing are applied on the queuing system.

In our experiments, the source four User Station links in different areas around South East Asia are chosen to observe traffic network. The source nodes are located in Bangkok-Thailand, Yangon-Myanmar, Vientiane-Laos, and Oil platform-Indian Ocean. The received four nodes are in North America. The destination nodes are at Boston-Massachusetts, Tallahassee-Florida, Saint Paul-Minnesota, and Lincoln-Nebraska respectively. Moreover, we assume to decrease link in satellite links bandwidth to be 90.112 Mbps and 11.264 Mbps. For link capacity of Trunk Station is 6.22 Mbps with 1 msec. Inside of internet backbone which is used in this work to be the bottleneck link, we assume that each continent has fiber link to connect together and with low latency delay between them. We adapted to 1.55 Mbps with 5 msec. The bottleneck link queue operates in bytes mode with buffer size of 400 KB. In addition, we assign error model at 2%. We use bottleneck queue size as an indication of queuing delay in bottleneck.

In this research, new and efficient of active queue management for constellation satellite system mechanisms are proposed into BHSCCS over LEO orbital in network simulation.

For traffic loads, we evaluate the scheme with more realistic traffic. To generate traffic loads in this simulation is [12] triple-play encompasses the provisioning of three services; high-speed Internet, television (Video on Demand or broadcasting video), and Voice over IP through the use of a single broadband connection. Key factors in the triple-play services delivery constitute the technological advances deriving from the development of appropriate equipment such as:

VOICE: [13], [14] Voice over Internet Protocol (VoIP) is a general term for describing transmission technologies for voice delivery over IP networks. Thus, we choose [15] a perceptual quality model for adaptive VoIP technique to run under our scenario. G.711 PCM codec is suitable for transported voice via the broadband Internet. Therefore we adopt voice traffic techniques for realistic form [15].

VIDEO: [16] Internet Protocol Television (IPTV) is a technology for delivering digital television service over IP networks. We collect up [17] video coding which H.264 scalable video coding (H.264/SVC) for video streams produced IPTV traffic model. This codec is more efficient bandwidth to enhance the quality of video delivered on the Internet.

For video traffic, we adopt the video file foreman with Siemens logo. It is contains 30 frames per second. The length of video is about 30 seconds, so we repeat this video three times into one video file and convert to high

TABLE I
Simulations Traffic of Triple-play Services over IPv6

Traffic	Packet size
VoIP with G.711 codec	~220 bytes
IPTV with H.264/SVC	~1540 bytes
FTP with TCP Westwood	~540 bytes

definition on 1920×1080. The summarized traffic of triple-play services over IPv6 parameters are shown in Table 1.

DATA: [18] Generally, File Transfer Protocol (FTP) is used for handling bulk data transfers. It is used over TCP protocol to communicate, so we adopt the TCP Westwood [19] protocol to run under our scenario. TCP Westwood is designed for efficiency for high speed wireless connection.

V. Simulation Results

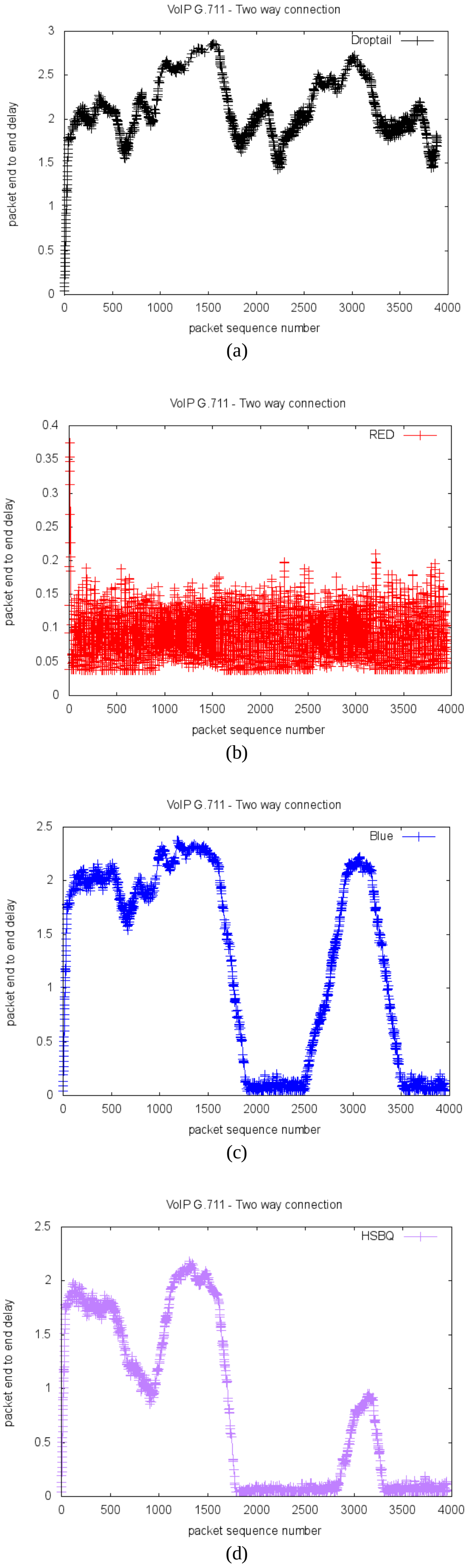
In coherence to other research, we implement queue management technique to study the most usual type of hybrid satellite constellations based on the BHSCCS network topology within the triple-play services application. We tested the performance of new queue algorithm in terms of drop ratio and average queue length per link. Drop ratio is defined as the ratio of dropped packets to the sum of dropped and successfully transmitted packets, and average queue length is the ratio of the sum in the average number of packets as bottleneck gateways.

The simulations are monitored average queue length, queue bandwidth, and packet end-to-end delay over a period of 100 seconds. The load consists of a combination of VoIP G.711 codec, IPTV H.264/SVC, and FTP over TCP Westwood. The simulation results for different AQMs loadings such as BLUE, RED, DropTail and proposed HSBQ, in which the loss rate ranges at 2%.

Voice and video traffic are different packet sizes. But their performances in each AQMs is affected by packet end-to-end delay in transmitted packet rate in same way of a predefined delay threshold. So, Figure 2 (a) DropTail, (b) RED, (c) BLUE, and (d) proposal HSBQ show that packet end-to-end delay of VoIP G.711 with two way connections. We also run simulations for comparison with other AQMs. The graph shows that RED suffers from continual unstable of packet arrival time because RED has random dropped packet in queue due to the deterministic packet marking behavior. Proposed HSBQ is lowest packet end-to-end delay than BLUE and DropTail queues. That means propagation delay of VoIP G.711 with two way connections from source to destination for our proposal over broadband hybrid satellite network has the lowest delays. It has shown the communicated by voice is rather smooth traffic voice because lowest propagation delays in end-to-end communication than DropTail and BLUE.

Figure 3 (a) DropTail, (b) RED, (c) BLUE, and (d) proposed HSBQ shows packet end-to-end delay of IPTV H.264/SVC with 1920×1080 resolutions at 30 frames per second. We also run simulations for comparison with other AQMs. The graph shows that proposed HSBQ is lowest packet end-to-end delay than BLUE and DropTail queues, but it is higher than RED because RED is random dropped packet in queue behavior. That means propagation delay from source to destination for our proposed over broadband hybrid satellite network has the lowest delays. Proposed HSBQ is lowest packet end-to-end delay than BLUE and DropTail queues. That means propagation delay of IPTV H.264/SVC from source to destination for our proposal over broadband hybrid satellite network has the lowest delays. It has shown the communication by IPTV has a rather smoother traffic voice because of lowest propagation delays in end-to-end communication than DropTail and BLUE.

For Figure 4 we have shown the PSNR results to reflect frame number of video in variations of each AQMs. The



(a)

(b)

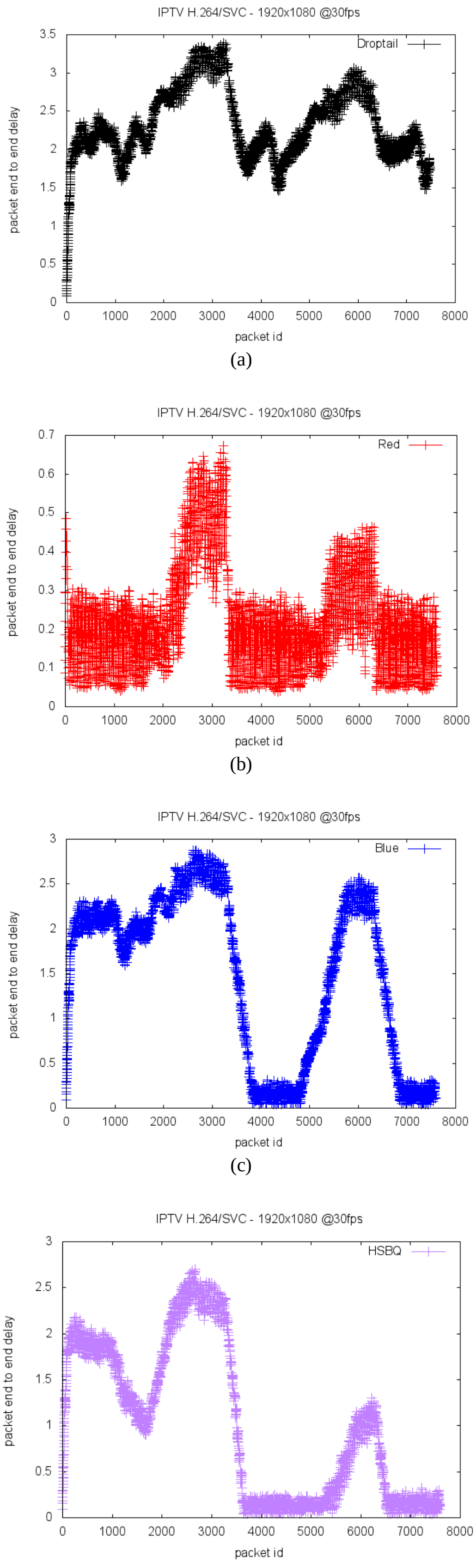
(c)

(d)

Fig. 2. VoIP: packet end-to-end delay in different queue (a) DropTail; (b) RED; (c) BLUE; (d) HSBQ

PSNR is an approximation to human perception of reconstruction image quality, however a higher PSNR generally indicates that the reconstruction is of higher quality but in some cases it may not. Also, the result of PSNR values for IPTV in different AQMs in simulation with foreman Siemens logo are shown in Figure 4 (a), (b), (c), (d) respectively.

From Figure 4 in axis x shows frame number of video that was received in destination node. DropTail is lowest frame number of video than other queues in group of



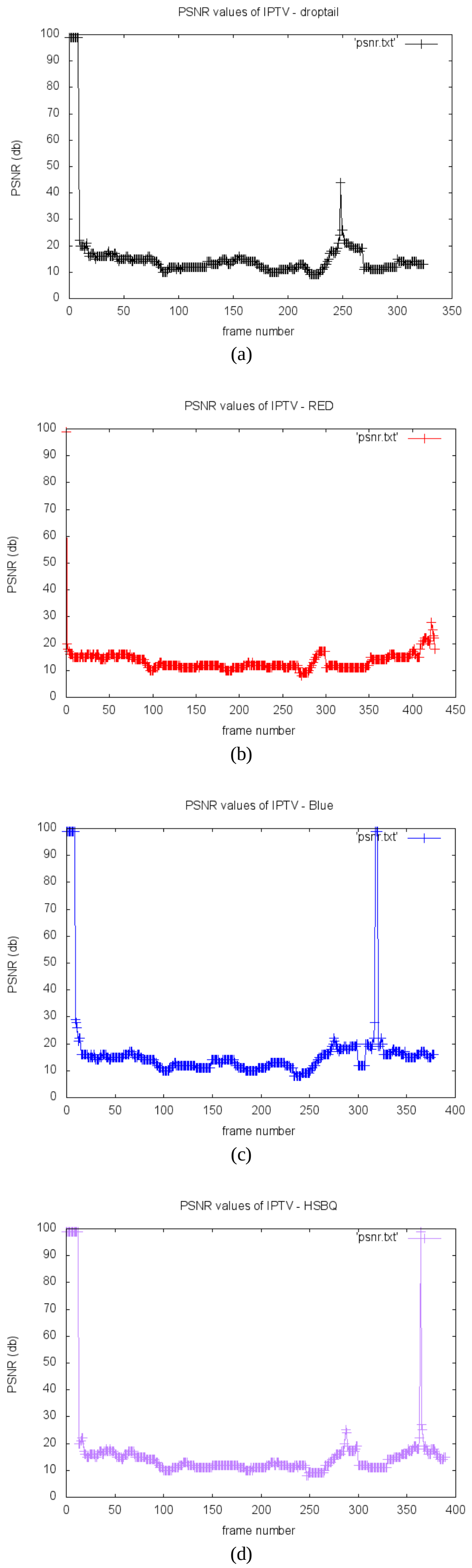
(a)

(b)

(c)

(d)

Fig. 3. IPTV: packet end-to-end delay in different queue (a) DropTail; (b) RED; (c) BLUE; (d) HSBQ



(a)

(b)

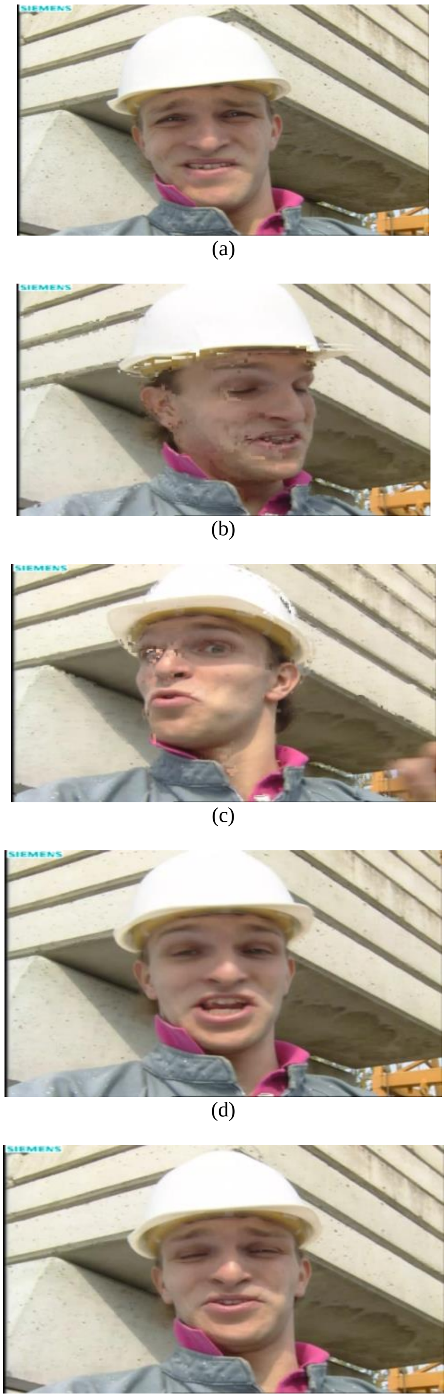
(c)

(d)

Fig. 4. PSNR value of IPTV in different queue (a) DropTail; (b) RED; (c) BLUE; (d) HSBQ

simulation, that mean reconstruction image quality of video is lower. But RED is the highest frame number of videos than other queues in group of simulation, that means highly reconstruction image quality, however a higher of RED that does not prove a smoother video frame that is shown in Figure 5. Next is our proposal HSBQ shown frame number of video is more than DropTail and BLUE. The smoother video frame from reconstruction image quality of HSBQ is show in Figure 5.

As a result, we choose frames number 41 of video traffic to visually compare the performance for each AQMs. Figure 5 (a) is the original frame number 41 of source file at resolutions 1920 × 1080 over codec H.264/SVC. Figure 5 (b), 5 (c), 5 (d), and 5 (e) are DropTail, RED, BLUE, HSBQ respectively. Illustrated shows the smooth motion of the video. In this research, DropTail and RED algorithms are suffering smooth IPTV H.264/SVC over broadband


(a)

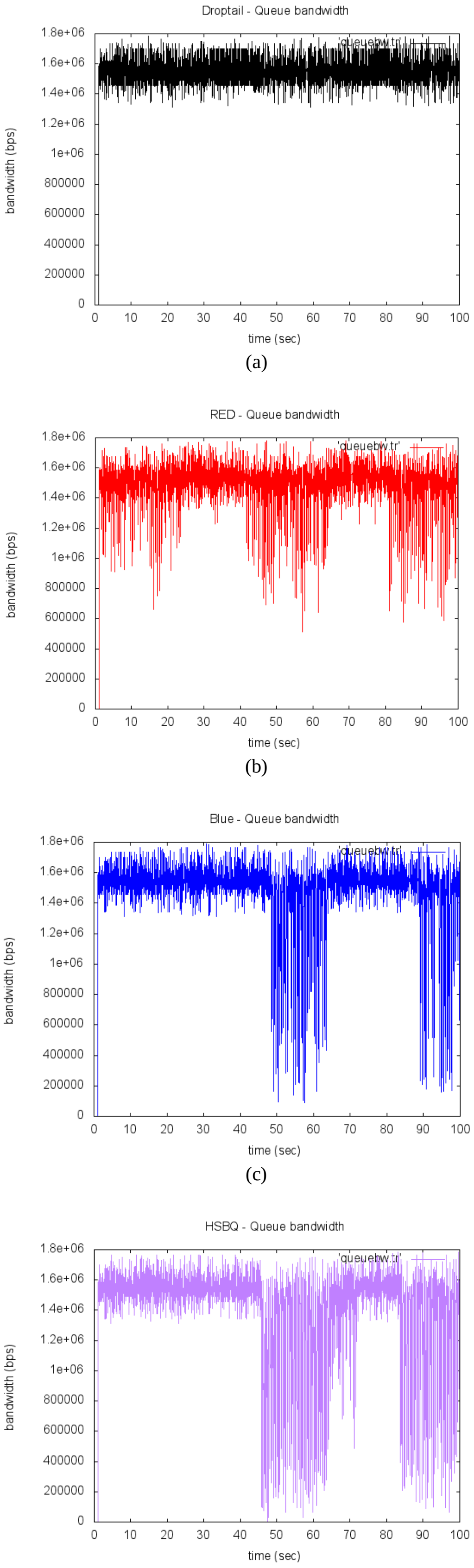
(b)

(c)

(d)

(e)

Fig. 5. Frame no 41 of IPTV in different queue (a) Original; (b) DropTail; (c) RED; (d) BLUE; (e) HSBQ



(a)

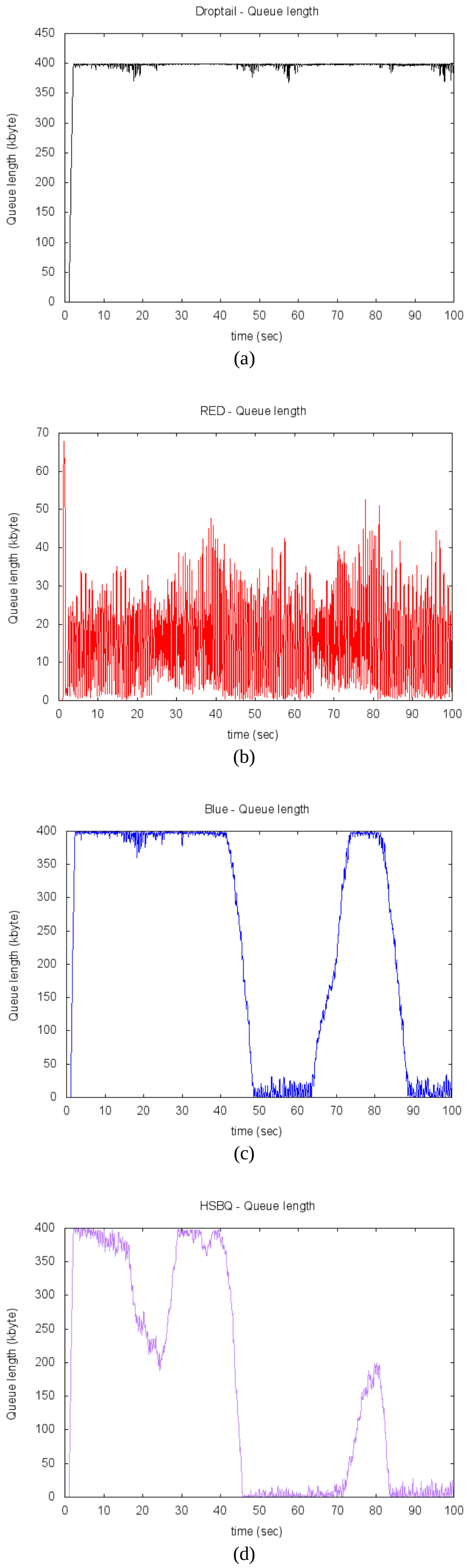
(b)

(c)

(d)

Fig. 6. Queue bandwidth in different queue (a) DropTail; (b) RED; (c) BLUE; (d) HSBQ

hybrid satellite communication networks. On the contrary, BLUE and HSBQ algorithms are very smooth. And proposed HSBQ is the lowest dropped packet of video than BLUE, also HSBQ has a frame number more than BLUE.



(a)

(b)

(c)

(d)

Fig. 7. Queue length in different queue (a) DropTail; (b) RED; (c) BLUE; (d) HSBQ

In Figure 6 we present the queue bandwidth of triple-play services flows. It is related queue length, and it should be matched up with queue length. These figures show that proposed HSBQ has the lowest uses of queue bandwidth in the traffic networks. That means, queuing in the system are lower than conventional queues. Therefore, traffic in hybrid satellite networks will be free more for other applications in the same time.

In Figure 7 we present the queue length with respect to time. Figure 7 illustrates instantaneous queue length of this simulation with 400 KB buffer size in triple-play services traffic loads. In proposed HSBQ, queue length decreases to reduce the congestion and reduces oscillatory behaviors. It is clearly shown that the proposed scheme effectively maintains queue length stability around the target queue length whereas the queue length in BLUE oscillates periodically than the proposed HSBQ. Using RED, the queue length is too low, and it is easy for it to be empty, which causes underutilization. Using DropTail is almost full queue length all the time of simulation and hardly increases over 400 KB. They vary the bottleneck queue length affected when varying the simulation parameters in triple-play services traffic loads.

VI. Conclusions

Broadband hybrid satellite networks are best suited for covering global area network, density traffic networks, providing a variety of mainly triple-play services application, and are expected to play an important role in the next-generation satellite networks. Numerous AQMs algorithm has been proposed to provide for reducing collision voice, video, data traffic in a hybrid Internet network environment, which is highly dynamic.

In this paper, we have proposed a new queue management scheme. It is the HSBQ algorithm classified based on congestion metrics and without the flow information. The performance of active queue management algorithms was evaluated in terms of drop ratio and average queue length per link. The simulation results have shown that the HSBQ algorithm significantly improves bottleneck network performance in comparison with a conventional AQMs algorithm. The queue bandwidth and queue length of HSBQ are steady while maintaining queue in order to decrease triple-play service application traffic before the queues overflow. The results obtained show that the proposed algorithm is capable of handling varying traffic distributions like triple-play services. However, the proposed algorithm has higher performance improvement and lower packet end-to-end delays in BHSCCS. In addition, we will apply the presented analytical of our model in clustering network system which is important from the network operator perspective.

References

[1] G. J. Wells, and D. Cooper, "COMMStellation Implementations for Northern Broadband Communications," in *Proc. 30th AIAA International Communications Satellite System Conference (ICSSC)*, Canada, 2012, pp. 830-839.

[2] G. J. Wells, D. Cooper, P. Sekhavat, S. Engleson, and P. Takats, "COMMStellation^{TM} – A Low Latency Satellite Constellation for Broadband Communications," in *Proc. 30th AIAA International Communications Satellite System Conference (ICSSC)*, Canada, 2012, pp. 858-871.

[3] A. Boriboon, and S. Pongpadpinit, "The COMMStellation^{TM} Satellite Constellation for Broadband Communication System Model in NS-2," *Int. J. Communications, Network and System Sciences*, vol. 7 no. 10, pp. 430-439, Oct. 2014.

[4] Y. Ma, T. Zhang, and J. Zhang, "An AQM Algorithm for LEO Satellite Network," in *Proc. IEEE 4th International Symposium on Microwave, Antenna, Propagation, and EMC Technologies for Wireless Communications (MAPE)*, Beijing, 2011, pp. 615-618.

[5] A. Boriboon, and S. Pongpadpinit, "Optimized routing protocol for broadband hybrid satellite constellation communication IP network system," *EURASIP Journal on Wireless Communications and Networking*, vol. 2016, no. 1, pp. 1-11, May. 2016.

[6] C. Joo, S. Bahk, and S. S. Lumetta, "A Hybrid Active Queue Management for Stability and Fast Adaptation," *Journal of Communications and Networks*, vol. 8, no. 1, pp. 93-105, March. 2006.

[7] K. Fall, and K. Varadhan (ed.), *The ns Manual (formerly ns Notes and Documentaion)*. UC Berkeley, CA: UC Berkeley, LBL, USC/ISI, and Xerox PARC, 2009.

[8] K. Chitra, and G. Padamavathi, "Classification and Performance of AQM-Based Schemes for Congestion Avoidance," *Int. J. of Computer Science and Information Security*, vol. 8, no. 1, pp. 331-340, April. 2010.

[9] J. Kim, H. Yoon, and I. Yeom, "Active Queue Management for Flow Fairness and Stable Queue Length," *IEEE Trans. on Parallel and Distributed Systems*, vol. 22 no. 4, pp. 571-579, April. 2011.

[10] C. Long, B. Zhao, X. Guan, and J. Yang, "The Yellow active queue management algorithm," *J. Computer Networks and ISDN Systems*, vol. 47, no. 4, pp. 525-550, March. 2005.

[11] S. S. Kunniyur, and R Srikant, "An Adaptive Virtual Queue (AVQ) Algorithm for Active Queue Management," *IEEE/ACM Transaction on Networking*, vol. 12, no. 2, pp. 286-299, April. 2004. doi:10.1109/TNET.2004.826291

[12] G. P. Sotiropoulos, D. K. Styliaras, E. A. Kosmatos, C. A. Papagianni, N. D. Tselikas, and I. S. Venieris, "Triple Play Service Simulation and Packet Scheduling Performance Evaluation," in *Proc. the International Conference on Digital Telecommunications (ICDT '06)*, France, 2006, pp. 54-59.

[13] M. Ali, L. Liang, Z. Sun, and H. Cruickshank, "Evaluation of Transport Protocols for SIP Signaling over IPv6 DVB-RCS Satellite Networks," in *Proc. 7th International Symposium on Wireless Communication Systems (ISWCS)*, 2010, UK, pp. 800-804.

[14] M. Ali, L. Liang, Z. Sun, and H. Cruickshank, "SIP Signaling and QoS for VoIP over IPv6 DVB-RCS Satellite Networks," in *Proc. the International Workshop on Satellite and Space Communications (IWSSC 2009)*, 2009, Italy, pp. 419-423.

[15] A. Bacioccola, C. Cicconetti, and G. Stea, "User-level performance evaluation of VoIP using ns-2," in *Proc. 2nd International Conference on Performance Evaluation Methodologies and Tools (ValueTools '07)*, 2007, Belgium, pp. 1-10.

[16] S. Pongpadpinit, "Motion vector recovery based colour information," in *Proc. 17th International Conference on Digital Signal Processing (DSP)*, 2011, Greece, pp. 1-4.

[17] C. Ke, "myEvalSVC: an Integrated Simulation Framework for Evaluation of H.264/SVC Transmission," *J. KSII Transactions on Internet and Information Systems*, vol. 6, no. 1, pp. 379-394, Jan. 2012.

[18] P. Lai, H. Subramoni, S. Narravula, A. Mamidala, and D. K. Panda, "Designing Efficient FTP Mechanisms for High Performance Data-Transfer over InfiniBand," in *Proc. the International Conference on Parallel Processing (ICPP '09)*, Vienna, 2009, pp. 156-163.

[19] C. Casetti, M. Gerla, S. Mascolo, M. Y. Sanadidi, and R. Wang, "TCP Westwood: End-to-End Congestion Control for Wired/Wireless Networks," *J. Wireless Networks*, vol. 8, no. 5, pp. 467-479, Sept. 2002.

Performance Analysis of MTD64, our Tiny Multi-Threaded DNS64 Server Implementation: Proof of Concept

Gábor Lencse

***Abstract*—In the current stage of IPv6 deployment, the combination of DNS64 and NAT64 is an important IPv6 transition technology, which can be used to enable IPv6 only clients to communicate with IPv4 only servers. In addition to the existing free software DNS64 implementations, we proposed a tiny multi-threaded one, MTD64.**

In this paper, the performance of MTD64 is measured and compared to that of the industry standard BIND in order to check the correctness of the design concepts of MTD64, especially of the one that we use a new thread for each request. For the performance measurements, our earlier proposed `dns64perf` program is enhanced as `dns64perf2`, which one is also documented in this paper. We found that MTD64 seriously outperformed BIND and hence our design principles may be useful for the design of a high performance production class DNS64 server.

As an additional test, we have also examined the effect of dynamic CPU frequency scaling to the performance of the implementations.

***Keywords*—IPv6 transition, DNS64, BIND, MTD64, Performance analysis.**

I. Introduction

The combination of DNS64 [1] and NAT64 [2] is an appropriate solution for the problem that an IPv6 only client must communicate with an IPv4 only server. This situation occurs when an ISP (Internet Service Provider) decides to introduce IPv6 in a way that it distributes only IPv6 addresses to its subscribers.[1] However, the majority of the servers on the Internet still uses only the IPv4 protocol. Several free software [3] (also called open source [4]) DNS64 implementations exist, e.g. BIND, TOTD, Unbound, PowerDNS, and we have also contributed with a tiny multi-treaded DNS64 implementation named MTD64 [5], [6]. In our earlier papers, we have compared the performances of BIND and TOTD [7], and later BIND, TOTD, Unbound and PowerDNS [8]. The aim of our current work is to compare the performance of MTD64, our new DNS64 implementation to that of BIND, the most well-known and industry standard DNS64 implementation and to thus check if our design concepts were right.

The remainder of this paper is organized as follows. In section 2, we recall the most important performance relevant design concepts of MTD64. In section 3, we disclose our performance measurement method including the documentation of the second version of our `dns64perf` test program. In section 4, we present and discuss our results.

We note that we have given an introduction to the operation of DNS64 in our paper [6], which may be worth consulting for those not familiar with DNS64.

II. Performance Relevant Design Decisions of MTD64

All our design considerations can be found in both [5] and [6] including e.g. to write free software. Now, we focus on those of them, which ones we consider performance relevant. The most important ones required that MTD64 should:

- be simple and therefore short (in source code)
- be fast (written in C, at most some parts in C++)
- not store the AAAA record (IPv6 address) requests in a common data structure, but start a new thread for each of them.

The first two ones are self explanatory, but the third one requires some explanation. A DNS64 server may receive many requests and because of the nature of the DNS64 service (the server must ask information from external source and thus it has to wait for the reply) it is deliberate that if we want a high performance DNS64 server then the processing of the consecutive AAAA record requests must overlap. A natural solution would have been to use e.g. three data structures:

- one for storing the new, unprocessed requests
- one for storing the requests, for which AAAA record requests were sent to the DNS system
- one for storing the requests, for which A record requests were sent to the DNS system.

The first one of them could have been a simple queue, as the incoming requests can be processed in the receiving order. However, the two other ones should support a searching method to find the matching request when a reply is received from the DNS system. Concerning the appropriate data structure, we considered in [5] that e.g. linked list, balanced or unbalanced trees could be used: “Their operations (insert, find, delete) involve programming complexity and the operations may involve significant time complexity if the data structure has high number of elements. Unfortunately there is a trade-off between the programming complexity and the speed. E.g. the operations of the linked list are simple but their time complexity is *O(n)*, where *n* denotes the number of elements

Manuscript received April 21, 2016, revised May 23.

G. Lencse is with the Department of Networked Systems and Services, Budapest University of Technology and Economics, Magyar tudósok körútja 2, Budapest, H-1117, e-mail: lencse@hit.bme.hu.

[1] This one can be a forward looking solution for the problem of the depletion of the public IPv4 address pool. Of course, there are several other possible solutions, e.g. the distribution of private IPv4 addresses to the clients together with the use of CGN (Carrier-grade NAT).

doi: 10.11601/ijates.v5i2.166

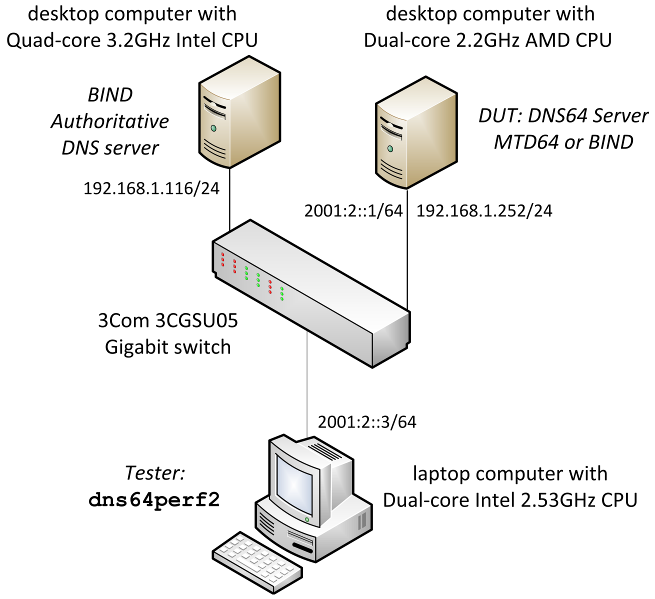


Fig. 1. DNS64 test network (See hardware and software details in subsection III-D.)

in the data structure. The time complexity of the operations of the balanced trees is *O(log n)*, but their operations require more programming work. For more information see [9] and its references." [5]

We decided not to store the requests explicitly but to start a new thread for each request. This was a risky decision and we new its potential positive and negative consequences. The positive consequences are:

- there is no need for implementing a data structure and its operations (programming simplicity, shorter code)
- all cores of the CPU can be utilized without the need of explicit programming efforts for it (may result in a good speed-up if executed by multi-core servers).

The negative consequences are:

- a multi-threaded code is much harder to debug
- starting a new thread for each request might result in too much computation costs (especially, because we used C++ for an easy thread handling)
- starting a new thread for each request makes MTD64 vulnerable to DoS attacks (attackers may exhaust the memory of the server by fake requests).

The aim of our current research is to examine the performance consequences of our design decisions.

III. Performance Measurement Method

A. Overview

We decided to use the DNS64 server performance measurement method that we had developed for our earlier papers [7], [8] for DNS64 server performance testing. That algorithm was later implemented in C/C++ by `dns64perf`, our DNS64 performance test program documented in [10]. Keeping the original algorithm made our current results comparable with our previous ones.

The logical topology of the test setup is shown in Fig. 1. As for *Tester*, we used the modified version[2] of `dns64perf`. The `dns64perf2` program sent a high number of AAAA record requests to the *DUT* (Device Under Test) for different domain names and received the replies. The *DUT* executed the MTD64 or the BIND[3] DNS64 server programs (the latter one served as a performance reference). As for authoritative DNS server, always BIND was used and it was executed by a desktop computer with significantly higher computing power to avoid being a bottleneck.

B. Operation of `dns64perf2`

The details of the testing algorithm that was implemented in `dns64perf` can be found in [10]. Whereas the basics of the method were kept in `dns64perf2`, we had to do some technical modifications for having long enough test runs. This subsection contains the documentation of the operation of version 2.

1) Testing Algorithm: The core of the testing algorithm is very simple: AAAA record queries are sent for domain names, which ones do not have AAAA records but only A records; hence the DNS64 server needs to synthesize IPv4-embedded IPv6 addresses. An independent name space is used to be able to resolve the domain names without delay. This name space is the following: `n1-n2-n3-n4.dns64perf.test`, where n1, n2, n3 and n4 are integers from the [0, 255] interval. This name space can be easily mapped to IPv4: the corresponding IPv4 address is: n1.n2.n3.n4. However, the roles of the four numbers are very different. The first one, n1 is used as a fixed number during an execution of `dns64perf2`. Its task is to define an independent name space for each execution[4]. Each execution of `dns64perf2` contains several experiments (at least 256 and at most 65536), where an experiment contains the resolution of 256 domain names and their time is measured together. The next two numbers, n2 and n3 are used as counters (n2 is the high order one and n3 is the low order one, that is n2*256+n3 equals with the sequence number of the given experiment). Finally the last number, n4 is used within an experiment. An experiment is executed in *n* number of threads, where *n* must be a power of 2. Each thread is responsible for 256/*n* domain names, which ones are requested sequentially with no overlapping: the next one can be asked only after receiving the result of the current request. Nanosecond precision time stamps are taken at the beginning of each experiment (before starting the *n* threads) and at the end of the experiment (after all threads were joined). Their difference is calculated, converted to milliseconds, and printed by `dns64perf2`.

[2] The modification is only technical, essentially the original testing method was used, see details in subsection III-B.

[3] BIND was chosen because it is the most well-known and widespread used free software DNS server implementation. Its DNS64 performance was compared to that of three other DNS64 implementations in our former paper [8]. Using those results and the current performance comparison of MTD64 and BIND, one can also compare the performance of MTD64 to that of the DNS64 implementations included in [8].

[4] It can be useful for different purposes. In our earlier works [7], [8], we used multiple clients to generate high enough load, thus we needed independent name space for each client. In this paper, we use it in a different way.

2) Parameters: The program takes five or six command line arguments. In their presentation, we use the variable names from the source code [11] and the C language notation of the program arguments. The operation of the `dns64perf2` program is controlled by the following parameters:

`a` = `argv[1]` specifies an independent name space for each execution of `dns64perf2`, where `a` is an integer from the [0, 255] interval.

`b` = `argv[2]`. During an execution of the program, 256*`b` experiments are performed, where `b` is an integer from the [0, 255] interval.

`n` = `argv[3]`. In each experiment, `n` threads are used, where `n` must be a power of 2, e.g. 1, 2, 4, 8, etc. and each of the threads sends 256/`n` number of AAAA record requests.

`timeout` = `argv[4]`. The timeout value is given as a positive integer (interpreted as seconds) and it specifies the time while the program waits for a reply (typical values are 1 or 5).

`server` = `argv[5]` is the IPv6 address of the DNS64 server.

`port` = `argv[6]` is the port number on which the DNS64 server program listens (if not supplied then the default value of 53 is used).

In `dns64perf2`, parameter `b` was introduced because `dns64perf` performed only 256 experiments and their execution time proved to be too short for our measurements. The execution of the program could have been repeated multiple times, but we intended to use its continuous operation.

What is the relationship between the command line arguments of the program and the n1, n2, n3 and n4 numbers in the first label of the domain names in the AAAA record requests?

- In all requests, n1 takes the value of `a`.
- The two bytes long counter built up by n2 and n3 takes its values from the [0, 256*`b`-1] interval. (The whole measurement contains 256*`b` number of experiments, and the two bytes long counter identifies the experiments.)
- An experiment contains 256 AAAA record requests, which ones can be distinguished by the value of n4. Each one of the `n` threads sends 256/`n` number of requests.

The load generated by the program may be tuned by the value of `n`, the number of threads. It may be worth increasing `n` over the number of CPU cores of the computer used for the execution of `dns64perf2`, because the threads may be waiting for the replies.

We note that in our earlier works [7] and [8], we used different number of client computers (1, 2, 4 and 8) executing the test script to be able to exactly tune the measure of the load. In this way we were able to exactly double the measure of the offered load. This time we will increase the number of threads to tune the measure of the load. We do it for ease of testing now, and we admit that the offered load will not be exactly doubled when doubling the number of threads because all the threads are executed by the same computer with limited resources.

C. Test Script Used for our Measurements

The tests were executed multiple times using different number of threads, to measure and compare the performances of MTD64 and BIND under different load conditions. The following test script was used:

```
#!/bin/bash
#Paramaters:
server=2001:2::1  # IPv6 address of the DNS64 server
dns64=mtd64       # type of the DNS64 server
b=10              # length of the measurement

for (( i=0; i<5; i++ ))
do
  nth=$((2**i));   # number of threads
  ssh $server ./stats $dns64 $nth &  # start dstat
  sleep 1
  ./dns64perf2 $i $b $nth 1 $server > \
    dns64perf2-results-${dns64}-$nth
  ssh $server killall dstat          # stop dstat
  sleep 5
done
```

As it can be seen from the script, the value of parameter `a` took the values 0, 1, 2, 3, 4. This way an independent name space was ensured for each execution of the `dns64perf2` program. The value of `b` was 10 that is 2560 experiments were performed in each run of `dns64perf2`. The number of threads took the values of 1, 2, 4, 8, 16 to increase the load offered by the Tester to the DUT. The timeout value was 1 second.

Though many times the DUT is considered as a black box, now we did not follow this approach, but measured the CPU utilization at the DNS64 server for a deeper understanding of the behavior of the tested DNS64 implementations. We started the `dstat` Linux command from a small `bash` script named `stats` executed by `ssh` (and stopped it by the `killall` command using `ssh`, too). The content of the `stats` script was:

```
#!/bin/bash
nice -n 10 dstat -c --output \
  dns64-stats-$1-$2.dstat > /dev/null
```

To calculate the CPU utilization, we used the idle time percentage from the output of `dstat`, and subtracted it from 100%. We did so because we considered less problem to include also the CPU utilization of some possible other processes, which were not taking part in DNS64 than leaving out the CPU utilization of some processes other than MTD64 or BIND, but doing some work for their interest (e.g. kernel processes sending and receiving packets, writing log files, etc.).

As we tested only two DNS64 implementations, they were started manually (and also their names were set manually in the test script).

D. Hardware and Software Parameters of the Test Environment

For the repeatability of our measurements, we provide the most important hardware and software parameters of our test environment.

Authoritative DNS Server Desktop computer with 3.2GHz Intel Core i5-4570 CPU (4 cores, 6MB cache), 16GB 1600MHz DDR3 SDRAM, 250GB Samsumg 840 EVO SSD, Realtek RTL8111F PCI Express Gigabit Ethernet NIC; Debian GNU/Linux 8.2 operating system, 3.2.0-4-amd64 kernel,

TABLE I
DNS64 PERFORMANCE: MTD64

1	Number of threads used in `dns64perf2`		1	2	4	8	16
2	Exec. time of 256 queries (ms)	average	61.34	33.23	28.53	17.90	15.12
3		std. dev.	0.95	1.13	1.19	0.77	0.82
4	Number of queries per second		4174	7705	8972	14300	16929
5	DNS64 server CPU utilization (%)	average	20.65	37.42	38.96	67.25	84.47
6		std. dev.	1.41	1.77	0.84	0.67	0.49

TABLE II
DNS64 PERFORMANCE: BIND

1	Number of threads used in `dns64perf2`		1	2	4	8	16
2	Exec. time of 256 queries (ms)	average	166.99	101.51	93.09	84.85	88.13
3		std. dev.	3.57	8.29	10.49	9.12	17.79
4	Number of queries per second		1533	2522	2750	3017	2905
5	DNS64 server CPU utilization (%)	average	50.51	72.37	68.87	83.18	86.63
6		std. dev.	1.06	2.26	6.75	4.68	3.93

TABLE III
DNS64 PERFORMANCE: MTD64, DYNAMIC CPU FREQUENCY SCALING ENABLED

1	Number of threads used in `dns64perf2`		1	2	4	8	16
2	Exec. time of 256 queries (ms)	average	93.27	51.23	38.56	23.80	16.87
3		std. dev.	0.93	0.70	1.29	0.66	0.58
4	Number of queries per second		2745	4997	6640	10757	15175
5	DNS64 server CPU utilization (%)	average	11.71	22.41	28.88	50.56	76.60
6		std. dev.	2.93	3.51	1.18	1.63	2.02

TABLE IV
DNS64 PERFORMANCE: BIND, DYNAMIC CPU FREQUENCY SCALING ENABLED

1	Number of threads used in `dns64perf2`		1	2	4	8	16
2	Exec. time of 256 queries (ms)	average	205.71	113.95	95.86	84.30	87.52
3		std. dev.	8.72	7.68	8.97	8.22	17.05
4	Number of queries per second		1244	2247	2670	3037	2925
5	DNS64 server CPU utilization (%)	average	40.38	70.08	65.94	83.99	87.40
6		std. dev.	1.81	1.97	3.45	4.16	3.70

BIND 9.9.5-9+deb8u3-Debian

DNS64 server Desktop computer with 2.2GHz AMD Athlon 64 X2 Dual Core CPU 4200+ (2 cores, 512kB cache), 2GB 667MHz DDR2 SDRAM, 320GB Samsung HD321KJ HDD, nVidia CK804 Gigabit Ethernet NIC; Debian GNU/Linux 8.2 operating system, 3.2.0-4-amd64 kernel, BIND 9.9.5-9+deb8u4-Debian, MTD64 from [12] (Latest commit: January 4, 2015)

Tester Dell Latitude E6400 series laptop with 2.53GHz Intel Core2 Duo T9400 CPU (2 cores, 6MB cache), 4GB 800MHz DDR2 SDRAM, 250GB Samsumg 840 EVO SSD, Intel 82567LM Gigabit Ethernet NIC; Debian GNU/Linux 8.2 operating system, 3.2.0-4-amd64 kernel, `dns64perf2` from [11]

Switch 3CGSU05 5-port 3Com Gigabit Ethernet switch

All three computers are able to use dynamic CPU frequency scaling. First, this feature was disabled on all three computers during our main measurements to eliminate its potential effect to the results. However, as this feature is quite common, we have also examined the case when it was enabled on all three computers.

IV. RESULTS AND DISCUSSION

A. Presentation and Interpretation of the Results

Our DNS64 performance measurement results are presented in Table I and Table II. Both tables follow the same structure. The number of threads used in `dns64perf2` is given in the first row. (We note that the number of threads is a parameter of the Tester and not of the tested DNS64 servers. It is used to tune the intensity of the load provided by the Tester so that the DNS64 server can be tested under different load conditions.) The average and the standard deviation of the execution time of an experiment (256 queries) are shown in row 2 and row 3, respectively. In row 4, we also displayed the number of answered queries per second calculated by using the average execution time from row 2. The average and the standard deviation of DNS64 server CPU utilization can be found in row 5 and row 6, respectively.

As for the performances of the two DNS64 implementations, MTD64 seriously outperformed BIND even when a single thread was used: MTD64 processed 4174 queries per second whereas BIND could do only 1533. This difference was growing further when the number of threads was increased. When 16 threads were used, MTD64 outperformed BIND more than five times by processing 16929 queries per second

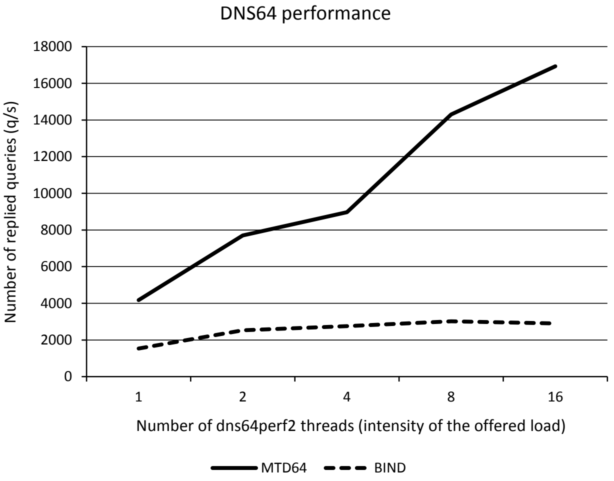


Fig. 2. DNS64 performance of MTD64 and BIND

whereas BIND could do only 2905. The explanation is clear: MTD64 used much less computing power than BIND, e.g. 20.65% vs. 50.51% at one thread or 37.42% vs. 72.37% at two threads, therefore MTD64 had still significant amount of spare CPU capacity at two threads and thus it could cope with the higher loads produced by 4, 8 and 16 threads execution of `dns64perf2`. Whereas the performance of BIND (2905qps) was somewhat less at 16 threads than at 8 threads (3017qps), MDT64 could still significantly increase its performance from 14300qps to 16929qps when the number of threads was increased from 8 to 16.

Fig. 2 provides a graphical comparison of the DNS64 performances of MTD64 and BIND as a function of the number of `dns64perf2` threads, that is, the intensity of the load.

B. Discussion

We can lay down that the results justify the design principles of MTD64. However, we can consider this excellent performance result only as a "proof of concept". We do not recommend MTD64 to be used as a real life DNS64 server for several reasons, including:

- MTD64 was not written to be a productive MTD64 server. It is neither supported, nor maintained.
- Though our tests have shown that it can perform DNS64 server functionality properly [6], it has not undergone extensive testing and may contain bugs.
- In its current state, it is not a real server program: it is not daemonized but runs in the foreground.
- MTD64 was not tested against DoS attacks for which it is vulnerable by design.

Thus our final evaluation is that the experiment of creating MTD64, a tiny multi-threaded DNS64 server was successful, and the our principles seem to be viable for the design and implementation of a production class DNS64 server.

We note that MTD64 has been released as a free software under GPL v2 license, thus anyone can make a fork of its source code available on Github [12] and may further develop it.

We also note that MTD64 is a light-weighted software omitting many real world scenarios (e.g. controls, exceptions, etc.) thus its performance comparison with BIND is not completely fair.

C. The Effect of Dynamic CPU Frequency Scaling

The DNS64 measurements were also performed with dynamic CPU frequency scaling enabled on all three computers. Table III and Table IV show the results.

The results produced under moderate load (using a single thread only) are significantly different from those produced when the dynamic CPU frequency scaling was disabled: MTD64 processed only 2745qps instead of 4174qps and BIND served 1244qps instead of 1533qps. Thus this result shows that disabling such mechanisms is a must when performance measurements are taken.

On the other side of the coin, the heavier the load was, the better the results of measurements with dynamic CPU frequency scaling approximated the results of the measurements without dynamic CPU frequency scaling. (As for BIND, the results for 4-16 threads are very similar; as for MTD64, the result for 16 threads are getting similar.) This observation justifies the application of dynamic CPU frequency scaling in production systems: the computers can still provide their full performance under high load conditions, and energy may be saved under lower load.

V. Conclusion

We conclude that the design principles of MTD64, our tiny multi-threaded DNS64 server can be useful in the design of a high performance production class DNS64 server.

References

[1] M. Bagnulo, A. Sullivan, P. Matthews, and I. Beijnum, "DNS64: DNS extensions for network address translation from IPv6 clients to IPv4 servers", IETF RFC 6147.

[2] M. Bagnulo, P. Matthews, and I. Beijnum, "Stateful NAT64: Network address and protocol translation from IPv6 clients to IPv4 servers", IETF RFC 6146.

[3] Free Software Foundation, "The free software definition", Available: http://www.gnu.org/philosophy/free-sw.en.html

[4] Open Source Initiative, "The open source definition", Available: http://opensource.org/docs/osd

[5] G. Lencse, and A. G. Soós, "Design of a tiny multi-threaded DNS64 server", in *Proc. 38th Internat. Conf. Telecommunications and Signal Processing*, Prague, 2015, pp. 27–32. DOI: 10.1109/TSP.2015.7296218

[6] G. Lencse, and A. G. Soós, "Design, implementation and testing of a tiny multi-threaded DNS64 server", *International Journal of Advances in Telecommunications, Electrotechnics, Signals and Systems*, vol. 5. no. 2, pp. 68–78, Mar. 2016, DOI: 10.11601/ijates.v5i2.129

[7] G. Lencse, and S. Répás, "Performance analysis and comparison of different DNS64 implementations for Linux, OpenBSD and FreeBSD", in *Proc. IEEE 27th Internat. Conf. Advanced Information Networking and Applications*, Barcelona, 2013, pp. 877–884. DOI: 10.1109/AINA.2013.80

[8] G. Lencse and S. Répás, "Performance analysis and comparison of four DNS64 implementations under different free operating systems", *Telecommun. Syst.*, in press, DOI: 10.1007/s11235-016-0142-x

[9] G. Lencse, "Investigation of event-set algorithms", in *Proc. 9th European Simulation Multiconference (ESM'95)*, Prague, 1995, pp. 821–825.

[10] G. Lencse, “Test program for the performance analysis of DNS64 servers”, *International Journal of Advances in Telecommunications, Electrotechnics, Signals and Systems*, vol. 4. no. 3. pp. 60–65, Sept. 2015, DOI: 10.11601/ijates.v4i3.121

[11] G. Lencse, “The `dns64perf2` DNS64 performance tester”, Available: http://www.hit.bme.hu/~lencse/dns64perf2

[12] A. G. Soós, “Multi-threaded DNS64 server” source code and documentation, Available: https://github.com/Yoso89/MTD64

Gábor Lencse received his MSc in electrical engineering and computer systems at the Technical University of Budapest in 1994, and his PhD in 2001.

He has been working for the Department of Telecommunications, Széchenyi István University in Győr since 1997. He teaches Computer networks and the Linux operating system. Now, he is an Associate Professor. He is responsible for the specialization of the information and communication technology of the BSc level electrical engineering education. He is a founding member and also a core member of the Multidisciplinary Doctoral School of Engineering Sciences, Széchenyi István University. The area of his research includes discrete-event simulation methodology, performance analysis of computer networks and IPv6 transition technologies. He has been working part time for the Department of Networked Systems and Services, Budapest University of Technology and Economics (the former Technical University of Budapest) since 2005. There he teaches Computer architectures and Computer networks.

Dr. Lencse is a member of IEEE, IEEE Communications Society, and the Institute of Electronics, Information and Communication Engineers (IEICE).

ISSN 1805-5443

9 771805 544167 02