JOURNAL OF TELECOMMUNICATIONS AND INFORMATION TECHNOLOGY

2/2016

(*Contents Continued on Back Cover*)

Editorial Board

Editorial Advisory Board

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

Circulation: 300 copies

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

JOURNAL OF TELECOMMUNICATIONS AND INFORMATION TECHNOLOGY

Preface

This issue of the *Journal of Telecommunications and Information Technology* contains fifteen papers that deal with diverse problems of energy-efficient networks and computing servers, network security, wire and wireless communication, or various issues related to application of high performance computing to neural systems simulation, large finite element problems and social networks analysis. The inspiration for the research presented in most papers published in this issue is the result of collaboration of the authors in the IC1406 COST Action cHiPSet *High-Performance Modeling and Simulation for Big Data Applications*, working group WG1: *Enabling infrastructures and middleware for Big-Data modeling and simulation*.

The first three papers are devoted to energy-efficient networks and data centers. Michał P. Karpowicz and Piotr Arabas in the paper *Server Workload Model Identification: Monitoring and Control Tools for Linux* present a brief overview of performance and power consumption monitoring tools available in the Linux systems. The authors argue that the measurements collected at high sampling rate can be used to develop maximally informative power consumption metrics and accurate dynamical processing models for the purpose of energy-aware design of server controllers. Selected approaches to save power in wired IP networks are discussed in the paper *Energy-saving Algorithms for the Control of Backbone Networks: A Survey*. Mariusz Kamola *et al*., describe proactive solutions employing advanced optimization techniques to squeeze as much energy saving as possible, and reactive ones supplementing the well rooted technologies (OSFP) with an energy-saving extra mechanisms. A model of dynamic power management in energy-aware networks is presented in the paper *New Developments in a Two-criteria Approach to Dynamic Power Management in Energy-Aware Computer Networks*. Andrzej Karbowski and Przemysław Jaskóła propose two-criteria approach to routing table calculation, i.e., energy consumption and the quality of service.

The following two papers deal with the application of GPU accelerators to large scale scientific and engineering computing problems. S. Y. Fialko and F. Zeglen describe in the paper *Preconditioned Conjugate Gradient Method for Solution of Large Finite Element Problems on CPU and GPU* the efficient parallel implementation of the preconditioned conjugate gradient method, and present the efficiency of its application to the large finite element problems. An OpenCL-based software platform for spiking neural networks simulation is described

in the paper *A Novel GPU-Enabled Simulator for Large Scale Spiking Neural Networks*. Paweł Szynkiewicz focuses on computationally efficient implementation of three widely used models of spiking neural networks. The presented results of numerical experiments conducted on AMD and NVIDIA graphical processors confirm the efficiency of the simulator.

The application of wavelet neural networks for the prediction of the stock price is described in the paper *Forecasting Stock Price using Wavelet Neural Network Optimized by Directed Artificial Bee Colony Algorithm*. Thanh Tung Khuat, Quang Chanh Le, Bich Loan Nguyen, My Hanh Le propose to use an artificial bee colony to optimize neural network parameters. The authors describe the results of calculations conducted on real data collected from Yahoo Finance. They claim that their solution can support traders and investors in their decision process.

Next two papers focus on network security. Agnieszka Jakóbik in the paper *A Cloud-aided Group RSA Scheme in Java 8 Environment and OpenStack Software* describes the RSA-enabled cryptosystem that allows a group of users to upload a single masked message to the computing cloud. The implementation employing open source cloud computing software for public and private clouds (OpenStack) is presented. Multiple numerical results are discussed in the final part of the paper. The application of honeypots to threat detection is discussed in the paper *Graph-based Forensic Analysis of Web Honeypot* by Hudan Studiawan, Supeno Djanali and Baskoro Adi Pratomo. The authors propose to employ a graph-based forensic analysis to examine an access log from a Web. Using graphical interface various forensic investigators can collaborate to detect given attacks. Thus, the presented solution can support intrusion detection systems.

Youness Jouihri, Zouhair Guennoun, Youssef Chagh and Driss Zahi in the paper *Network Function Virtualization: Mitigating the Impact of VoLTE on the Policy and the Charging System* consider a problem of the optimization of network resources utilization while providing for adequate transmission quality. The paper addresses issues related to the impact of Diameter signaling generated after massive Voice over LTE (VoLTE) deployment on the operation of a core network. The authors describe the application of network function virtualization technology to create a model that can anticipate the impact induced by massively introducing VoLTE, and support network operators in signaling related to policy management without impacting existing services.

The attention of next two papers is focused on social networks technologies and their application in telecommunication domain. Witold Gruszczyński and Piotr Arabas in the paper *Application of Social Network Inferred Data to Churn Modeling in Telecoms* consider the idea of the usage of social network analysis to divide customers into specific segments, and finally predict churn of telephony network subscribers. The authors describe and evaluate through extensive simulation a hybrid predictor that employs a set of regression models and data describing social links between subscribers. The article *Similarity Index based Link Prediction Algorithms in Social Networks: A Survey* by Pulipati Srilatha and Ramakrishnan Manjula is devoted to the issue of the detection of possible potential links among people. The paper overviews algorithms of links prediction and development lines that can be observed in the literature on the social networks technologies.

The following two papers deal with modern radio systems. *Antenna Arrays Focused on Broadband Signals* by Denis A. Vedenkin, Yuri E. Sedelnikov and Aydar R. Nasybullin addresses issues related to the design and development of broadband antennas. The authors provide models for ultra-wideband signals transmission and antenna arrays, respectively. They draw attention that at wide band signal the difference between function characterizing the spatial selectivity in receiving mode and functions determining the spatial distribution of energy in the transmit mode can be considerable. Oleg G. Morozov, Aydar R. Nasybullin, Denis A. Vedenkin and Timur A. Agliullin in the paper *Radio Photonic Systems for Measurement of Instantaneous Radio Frequency with Amplitude-Phase Modulation of Optical Carrier* focus on application of radio photonics to measure radio signals instantaneous frequency (MSRSIF). The authors claim that a further development of measurement systems considered may be based on the use of amplitude phase modulation transformation (AFMT) of optical carrier, for measuring the instantaneous frequency, and to provide a stable operating mode of conversion devices.

The application of multi-class support vector machine to efficient data decoding of Bose Chaudhuri Hocquenghem code is discussed by V. Sudharsan and B. Yamuna in the paper

Support Vector Machine based Decoding Algorithm for BCH Codes. The presented simulation results confirm that the described algorithm gives better results when compared with the conventional Chase-2 algorithm. The authors highlight the advantages of the SVM based decoding method such as: a more generalized decision model, fast convergence to globally optimal solution and prevention of outliers.

The last paper *An Efficient Early Iteration Termination for Turbo Decoder* is concerned with the development of efficient iteration control techniques for decoding turbo code. P. Salija and B. Yamuna describe a simple method based on absolute value of the mean of extrinsic information at the component decoders of turbo code. The authors present the simulation results that confirm good performance of their method. It allows to reduce the average number of iterations while maintaining performance closed to that of fixed iteration termination.

We wish our Readers an interesting reading time.

Ewa Niewiadomska-Szynkiewicz
Warsaw University of Technology, Poland
Ioan Salomie
Technical University of Cluj-Napoca, Romania
Guest Editors

Paper

Server Workload Model Identification: Monitoring and Control Tools for Linux

Michał Karpowicz and Piotr Arabas

Institute of Control and Computation Engineering, Warsaw University of Technology, Warsaw, Poland
Research and Academic Computer Network (NASK), Warsaw, Poland

Abstract — Server power control in data centers is a coordinated process carefully designed to reach multiple data center management objectives. The main objectives include avoiding power capacity overloads and system overheating, as well as fulfilling service-level agreements (SLAs). In addition to the primary goals, server control process aims to maximize various energy efficiency metrics subject to reliability constraints. Monitoring of data center performance is fundamental for its efficient management. In order to keep track of how well the computing tasks are processed, cluster control systems need to collect accurate measurements of activities of cluster components. This paper presents a brief overview of performance and power consumption monitoring tools available in the Linux systems.

Keywords — cloud computing, energy efficiency, Linux, server performance metering.

1. Introduction

Data centers supporting both cloud services and high performance computing (HPC) applications consume enormous amounts of electrical energy. From 2005 to 2010 the energy consumed by data centers worldwide rose by 56%, which was accounted to be between 1.1% and 1.5% of the total electricity use in 2010. The growth of energy consumption rises operating costs of data centers but also contributes to carbon dioxide (CO_2) production. According to the analysis of current trends (gesi.org/SMARTer2020), the carbon dioxide emissions of the ICT industry are expected to exceed 2% of the global emissions, a level equivalent to the contribution of the aviation [1]. Energy usage in data centers grows rapidly with the climbing demand for cloud and HPC services. However, the growth rate of ICT cannot be sustained unless the power consumption problem is addressed [2]–[4]. In response to the created momentum new computing elements, i.e. CPUs/GPUs, memory units, disks, network interface cards (NICs), have been designed to operate in multiple (performance and idle) modes of differentiated energy-consumption levels (ACPI).

Although energy efficiency (FLOPS/watt) of ICT systems continues to improve, the rate of improvement does not match the growth rate of demand for computing capacity. Based on the projections of technology development it has been argued that continued scaling of available systems will eventually lead to a data center architecture consuming more than a gigawatt of electrical power (at exaflop level), a level that violates economic rationale for providing cloud or HPC services. Unless radically new energy-aware technologies are introduced, both in hardware and software domain, it will not be possible to meet DARPA's 20-megawatt exaflop goal (50 GFLOPS/watt) by year 2020 [3]. Limiting power consumption and related thermal emission has therefore become a key engineering problem.

In order to meet the challenging goals of cloud and high performance computing, advances in hardware layer development require immediate improvements in the design of cluster control software. In Section 2 a general structure and components of a data center monitoring and control system are presented. A brief description of resource allocation and performance control process is also given. In particular, the role of energy-efficient device controllers is indicated. Section 3 presents currently developed concepts of power measurement and control programming interfaces, both for data centers and wired IP networks. Basic performance metrics and benchmarking strategies are presented in Sections 4 and 5. Finally, in Section 6 the results of simple experiments are given to illustrate the discussed profiling and metering techniques.

2. An Overview of Data Center Management

Monitoring of cluster performance is fundamental for its efficient management. In order to keep track of how well the computing tasks are processed, cluster control systems need to collect accurate measurements of activities of cluster components. The collected measurements, including both data processing and power consumption metrics, provide feedback for management operations and serve as a basis for the design of new cluster control systems.

Figure 1 presents an overview of a cluster control system architecture. The racks, supplied with electric power by power distribution units (PDUs), are filled with blade servers. The racks are connected into a data center network with a hierarchy of switches (SW). The management layer is responsible for allocation of resources, job submission, adjustments of the interconnect settings, power budgeting and system monitoring. These tasks are executed by dedicated resource allocation and job management systems (RJMS) and system-wide energy management systems (SEM). The lower control layer, composed of operating sys-

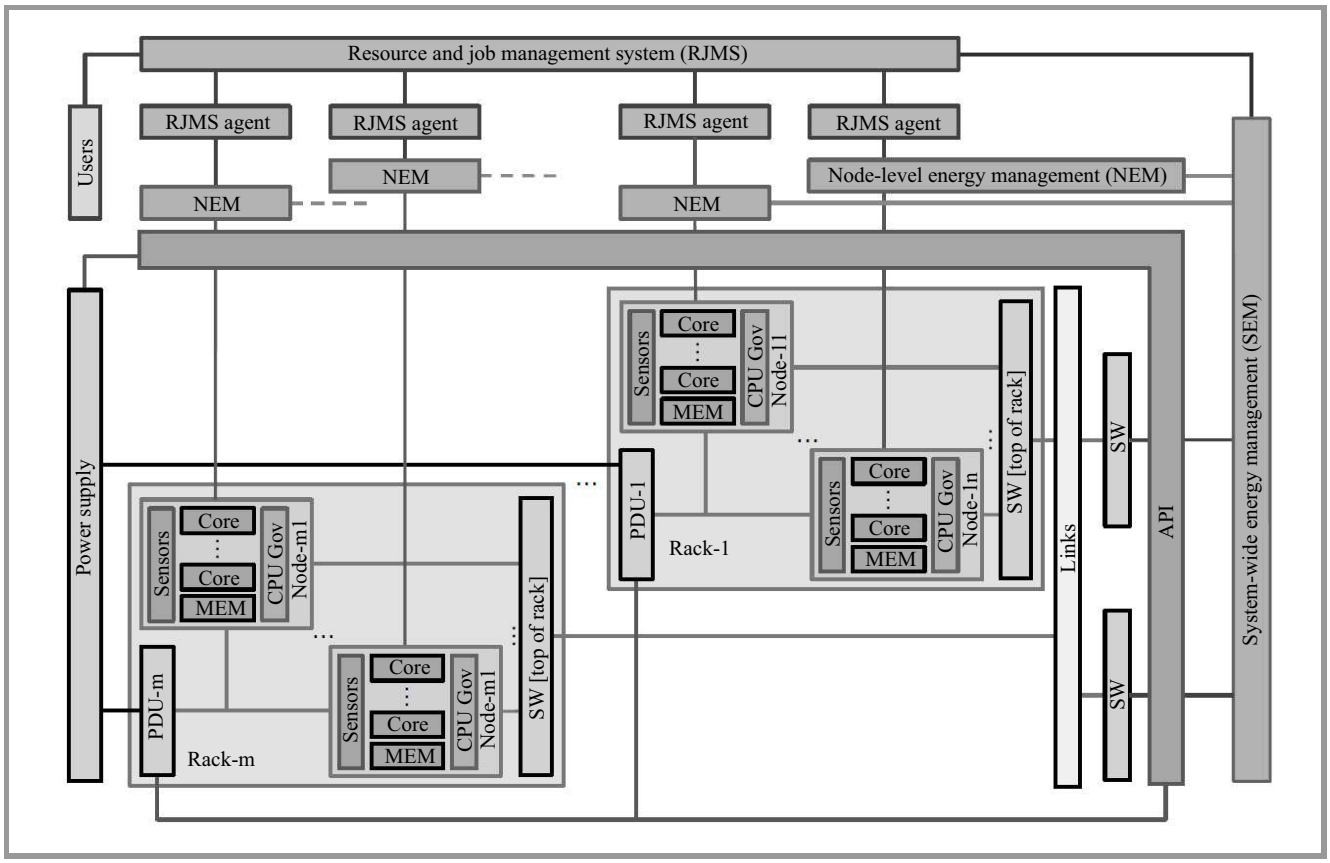


Fig. 1. An overview of cluster control system architecture.

tems controlling servers, is responsible for job execution and enforcement of resource usage constraints in computing nodes [5]–[7]. Available computing resources, including CPUs, memory and sockets, are exposed to the management systems via customized Abstract Programming Interfaces (APIs).

The challenging problem of data center control is to maximize the utilization of the resources subject to energy consumption and quality of service constraints. To solve this problem hardware-specific power saving capabilities of computing elements are used to keep the total power consumption of data center within the required power range. The resources that remain idle can be switched to a power saving (or sleeping) mode for a configurable time and restored to operating mode on demand. The resources performing operations can reduce their power consumption by dynamically adjusting their performance level [8]–[11].

3. Power Control Programming Interfaces

Power management capabilities of hardware layer are exposed in the form of Application Programming Interfaces (APIs). The foundations of power control APIs were built by the Advanced Configuration and Power Interface (ACPI) specification [12]. The specification defines hardware dependent energy saving (idle) and performance (active) states that can be adjusted on demand from the software level. This allows to control power saving and data processing efficiency according to a designed policy.

An attempt to design a vendor-neutral API dedicated for power measurement and control in HPC systems resulted in development of Power API specification [13], [14]. The Power API describes cluster as a collection of objects forming a discoverable hierarchy. Objects in the system are characterized by a set of attributes, which allow for measurement (reading attributes) and control (overwriting attributes) of their power saving capabilities. Functions providing gathering of statistics are provided for objects and groups of objects. Both ACPI and PAPI can be adapted to the Power API abstract model.

A similar approach is proposed by the ETSI Green Abstraction Layer (GAL) standard [15]. It describes a general concept of programming interface for energy state configuration of energy-aware telecommunication fixed network nodes. A hierarchical representation of a network device is proposed, which allows to control the available energy-aware states of its internal components. The innovation is not only in the described unification of control but also in the ability to query energy-aware capabilities of the components.

4. Performance Metrics and Benchmarks

Energy consumption management is a multi-objective optimization problem in which multiple performance and

energy related metrics are considered [16]–[19]. Usually at least the following two objectives are considered:

- minimization of peak power consumption,
- maximization of energy-efficiency.

Limiting peak power consumption is critical to maintaining reliability of data center, avoiding power capacity overloads and system overheating, as well as fulfilling service-level agreements (SLAs). At the same time, since economically feasible power consumption levels are strongly correlated with the costs of electricity and power provisioning, it is important to maximize efficiency of operations performed in data center [2], [3], [20].
Energy-efficiency is defined as a number of operations performed per energy unit, i.e.:

$$\text{Energy efficiency} = \frac{\text{Computing performance}}{\text{Total energy consumed}}. \quad (1)$$

This universal metric has been in the center of research focused on energy-aware control of data processing systems. In order for the metric to be improved it is necessary to increase the number of operations performed per unit of energy consumed or to decrease the amount of energy required per operation. Based on the above observations various strategies of power management have been developed. Consequently, the metric has also been used in many benchmarking methodologies.
Basic industry-standard methodology for power and performance benchmarking of a computing server is SPECpower [21]–[23]. The benchmark measures power consumption of a server running an appropriately designed application (Java application server) at workload ranging from 10 to 100% of peak achievable level. Namely, a steady flow of work requests is submitted to the server under test to determine the number of requests that can be satisfied in a given time. The benchmark drivers request work at intermediate points between zero and the maximum throughput value. The related toolset can be used with other cluster-wide benchmarks.
Energy efficiency has been used as a default benchmarking metric. In the case of HPC systems (or batch processing systems) the performance metric is typically defined by the number of GFLOPS performed on average per watt while executing a selected benchmark [24], [25]. Transaction processing systems, composed of application and Web servers, as well as networks of routers have been evaluated in terms of served requests per watt during throughput-based benchmarks [26]–[28]. Dedicated tests reporting transaction throughput per watt have been developed for storage systems [29], [30] as well. Finally, from the perspective of data center management energy efficiency is viewed as a product of [17]:

- facility efficiency – the ratio of total amount of energy used by a data center facility to the energy delivered to computing equipment (PUE),
- server power conversion efficiency – the ratio of total server input power to its useful power consumed by the electronic components directly involved in the computation (SPUE),
- server's architectural efficiency – the ratio of computing performance metric to total amount of energy used by electronic components.

5. Power Monitoring and Profiling

Monitoring of power and energy consumption in computing clusters is a complex problem [7], [31]–[33]. Two general approaches can be distinguished that allow to perform the required measurements. The first one is based on power metering devices connected to the servers. Basic system-wide measurements are usually provided at rate ranging from 0.01 to several samples per second by power supply units (PSUs) and power distribution units (PDUs) through the Intelligent Platform Management Interface (IPMI) [34]. More accurate and detailed measurements, collected at high sampling rate and covering selected components of servers, may be provided by additional and dedicated metering devices [35], [36].
Whenever IPMI-based monitoring systems directly communicate with the Baseboard Management Controllers (BMC) or metering devices, there is no direct overhead on the observed servers caused by the measurements. Otherwise, perturbations of measurements should be expected. In practice IPMI is often used with server management software running under the local operating system. This allows to access hardware-specific function, exposed by available APIs, and conveniently deal with local measurements, control commands execution, error handling and alerting.
Monitoring, configuration and control of devices that support IPMI on Linux systems can be performed with `ipmitool` utility. A list of sensors visible in the system and their records can be viewed with commands:

```
$ ipmitool sensor
$ ipmitool sdr -v
$ ipmitool sdr elist full
```

Example below presents an outcome of IPMI-based monitoring:

```
# ipmitool sdr elist full
Fan1 RPM         | 30h | ok  |  7.1 | 3840 RPM
Fan2 RPM         | 31h | ok  |  7.1 | 3840 RPM
Fan3 RPM         | 32h | ok  |  7.1 | 3960 RPM
Fan4 RPM         | 33h | ok  |  7.1 | 3960 RPM
Fan5 RPM         | 34h | ok  |  7.1 | 3960 RPM
Fan6 RPM         | 35h | ok  |  7.1 | 3840 RPM
Inlet Temp       | 04h | ok  |  7.1 | 17 degrees C
Exhaust Temp     | 01h | ok  |  7.1 | 28 degrees C
Temp             | 0Eh | ns  |  3.1 | Disabled
Temp             | 0Fh | ns  |  3.2 | Disabled
Current 1        | 6Ah | ok  | 10.1 | 0.60 Amps
Current 2        | 6Bh | ok  | 10.2 | 0 Amps
Voltage 1        | 6Ch | ok  | 10.1 | 230 Volts
Voltage 2        | 6Dh | ok  | 10.2 | 240 Volts
Pwr Consumption  | 77h | ok  |  7.1 | 140 Watts
```

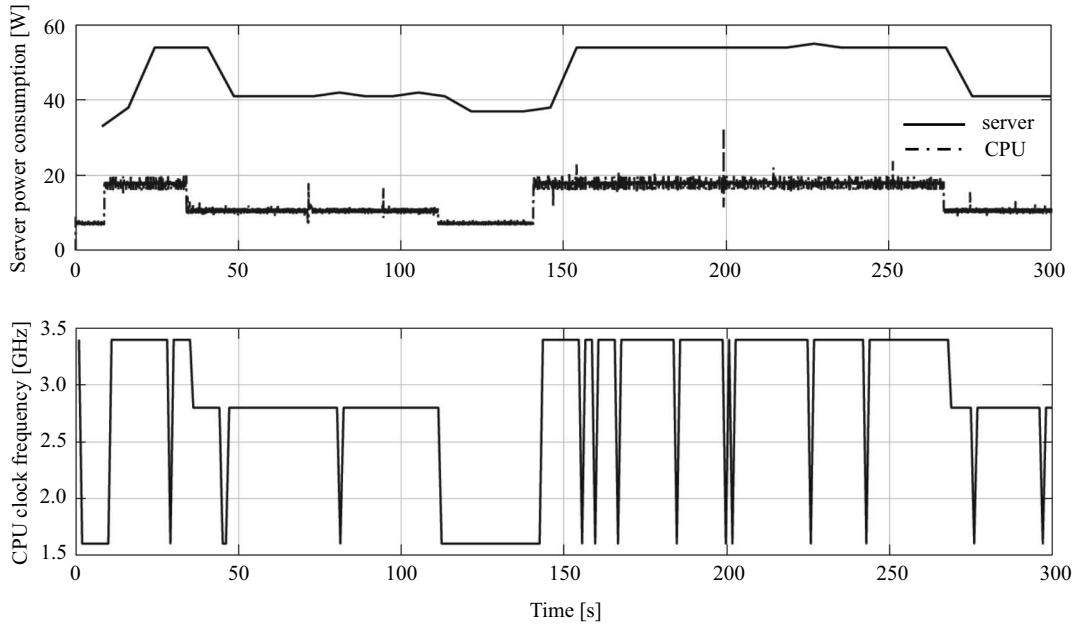


Fig. 2. Correlation of server/CPU power consumption (above) and CPU clock frequency.

The second approach to monitoring of power consumption exploits hardware-level counters provided by selected computing elements, including CPU, GPU and DRAM. The most commonly available counters are exposed through Intel's Running Average Power Limiting (RAPL) interface and NVIDIA's Management Library (NVML) functions [37], [38]. Performance counters allow to collect measurements at rate ranging from 100 to 1000 samples per second with high accuracy [39].

When used together with benchmarking probes of the operating system kernel a runtime estimation of power consumption and performance can be realized on a per-application basis. In the Linux system the required instrumentation is provided by the Performance Application Programming Interface (PAPI) and `perf_events` functions allowing to observe micro-architectural events (such as instructions completed per cycle and cache-misses) [27], [28], [40]–[43]. This paves the way for high resolution identification of data processing dynamics and its energy-efficiency, and potentially for the design of energy-aware application-specific server controllers. Novel auto-tuning systems are also developed that allow to introduce server energy control instructions into application source code [44].

It is important to point out that care must be taken when MSR-based measurements are used for performance benchmarking. Since readouts are taken on the system under test the measurements may be significantly perturbed by the measurement process itself, especially under high sampling rate. Figure 2 shows the results of measurements taken from `ipmi` system and MSRs of CPU.

Both methods of monitoring, briefly discussed above, are conveniently integrated by the resource allocation and job scheduling systems [7]. As a result it is not only possible to perform energy accounting and power profiling per job but also to setup system power-saving configuration for the purpose of job execution. Along with the scheduled batch of jobs appropriately defined control server control policies can be submitted to the computing nodes, thereby optimizing energy efficiency [45].

6. An Experimental Illustration

To demonstrate how benchmarking tools can be combined with performance and power consumption monitoring APIs several simple experiments are presented below.
Listing 1 illustrates how an average power consumption of CPU and DRAM can be calculated in Linux systems. Listing 2 illustrates how the desired RAPL MSRs (`address` variable) can be accessed from application level[1].
In order to get energy consumption information from the Intel's RAPL model specific registers (MSRs)[2] it is necessary to multiply increments of appropriate energy status counters, stored in `MSR_*_ENERGY_STATUS` registers, by scaled energy status unit, stored in `MSR_RAPL_POWER_UNIT` register. Energy status MSRs are updated approximately every 1 ms, with wraparound time of around 60 s when power consumption is high [37], [39].

The following listing shows how Linux `stress` microbenchmark can be analyzed with `perf` tool based on `perf_events` subsystem of the Linux kernel:

[1] Appropriate permissions should be setup to access `/dev/cpu/*/msr` interface.

[2] Intel's Sandy Bridge processors.

```
# perf stat -a\
  stress --cpu 32 --io 32 --vm 32 \
  --vm-bytes 512M --hdd 10 --timeout 30s

task-clock (msec) #   32.006 CPUs utilized
context-switches  #    0.122 K/sec
cpu-migrations    #    0.004 K/sec
page-faults       #    0.125 M/sec
cycles            #    2.154 GHz
instructions      #    0.75 ins per cycle
                  #    0.89 stalled cycles/ins
branches          #  430.045 M/sec
branch-misses     #    0.20% of all branches
```

Finally, in order to retrieve detailed measurements of performance and power consumption of selected parts of application source code, low-level performance and energy counters exposed via PAPI and RAPL MSRs can be used. For illustrative purposes the Linux `stress` micro-benchmark was appropriately adapted. The list of counters introduced into the source code of the benchmark included:

- total number of CPU cycles,
- reference clock cycles,
- number of completed instructions (INS),
- Level 1 instruction cache misses (ICM),
- Level 1 data cache misses (DCM),
- RAPL MSRs counting power consumption of CPU core and DRAM (W).

Listing 1: Simple power consumption monitoring script

```
#!/bin/bash

SAMPLING_RATE=1                 # seconds
MSR_PKG_ENERGY_STATUS="0x611"  # CPU energy
    counter
MSR_DRAM_ENERGY_STATUS="0x619" # DRAM energy
    counter

# Energy Status Units (ESU)
ESU=`echo "ibase=16;\
      1/2^$(rdmsr -X 0x606 -f 12:8)" | bc -l`
# Calculate number of CPU energy status
# counter incremants during sampling period
ESPKG=`a=$(rdmsr -X $MSR_PKG_ENERGY_STATUS);\
       sleep $SAMPLING_RATE; echo "ibase=16;\
        $(rdmsr -X $MSR_PKG_ENERGY_STATUS)-$a"|bc`
# Calculate DRAM energy status
# counter incremants during sampling period
ESDRAM=`a=$(rdmsr -X $MSR_DRAM_ENERGY_STATUS);\
        sleep $SAMPLING_RATE; echo "ibase=16;\
         $(rdmsr -X $MSR_DRAM_ENERGY_STATUS)-$a"|
    bc`
# Calculate power consumption [W]
CPUPOW=`echo "$ESPKG * $ESU" | bc -l`
DRAMPOW=`echo "$ESDRAM * $ESU" | bc -l`
echo CPU: $CPUPOW W
echo DRAM: $DRAMPOW W
```

Listing 2: Example of RAPL MSR read function

```
int read_msr(int cpu, unsigned int address,
    uint64_t *value)
{
  int   err = 0;
  char  msr_path[32];
  FILE *fp;

  sprintf(msr_path, "/dev/cpu/%d/msr", cpu);
  err = ((fp = fopen(msr_path, "r")) == NULL);
  if (!err) err = (fseek(fp, address, SEEK_CUR)
       != 0);
  if (!err) err = (fread(value, sizeof(uint64_t)
      , 1, fp) != 1);
  if (fp != NULL) fclose(fp);
  return err;
}
```

Figures 3 and 4 present the results of experiments in which the server under test[3] was forced to execute CPU and memory intensive benchmarking loops consisting of randomly generated number of iterations. In addition, the experiments were conducted for two different CPU frequency scaling governors, `intel_pstate powersave` and `intel_pstate performance` [46]. The results show a relation between the CPU frequency, power consumption, number of completed instructions and L1 cache misses. It can be seen that energy-efficiency of instructions is high when the CPU frequency is reduced. The same pattern can be observed with L1 cache misses, i.e. probability of a cache miss rises with the number CPU cycles performed. Integration of collected data allows to retrieve aggregated results of experiments.

Based on the obtained results interesting observations can be made regarding operations performed by the operating system. In particular, it is possible to study efficiency of CPU frequency control policy implementation. In order to increase the number of computing operations performed per watt, thereby maximizing Eq. (1), it is necessary to reduce the amount of time the processor spends running idle loops or stall cycles [16]. Therefore, energy-efficiency maximizing CPU controllers should implement a workload following policy dynamically adjusting CPU performance state (ACPI P-state) to the observed short-term CPU utilization or application-related latency metrics. This control concept is indeed implemented in the currently distributed CPU frequency governors of the Linux kernel, namely `intel_pstate` and `cpufreq_ondemand`.

Given a CPU workload estimate the `intel_pstate` governor, used in the presented experiments, applies PID control rule to keep the workload at the default reference level of 97%. In comparison, the `ondemand` governor calculates CPU frequency according to the following policy. If the observed CPU workload is higher than the upper-threshold value then the operating frequency is increased to the maximal one. If the observed workload is below the lower-threshold value, then the frequency is set to the lowest level at which the observed workload can be supported. For a discussion of optimal CPU frequency control policy design problem see e.g. [45], [47]. Identification of server data processing dynamics is discussed e.g. in [27], [28].

[3] DELL PowerEdge R720, 2 × Intel Xeon E5-2670 2.60 GHz, 20 MB cache, 12 × 16 GB RDIMM 1600 MHz, Linux kernel ver. 4.4.

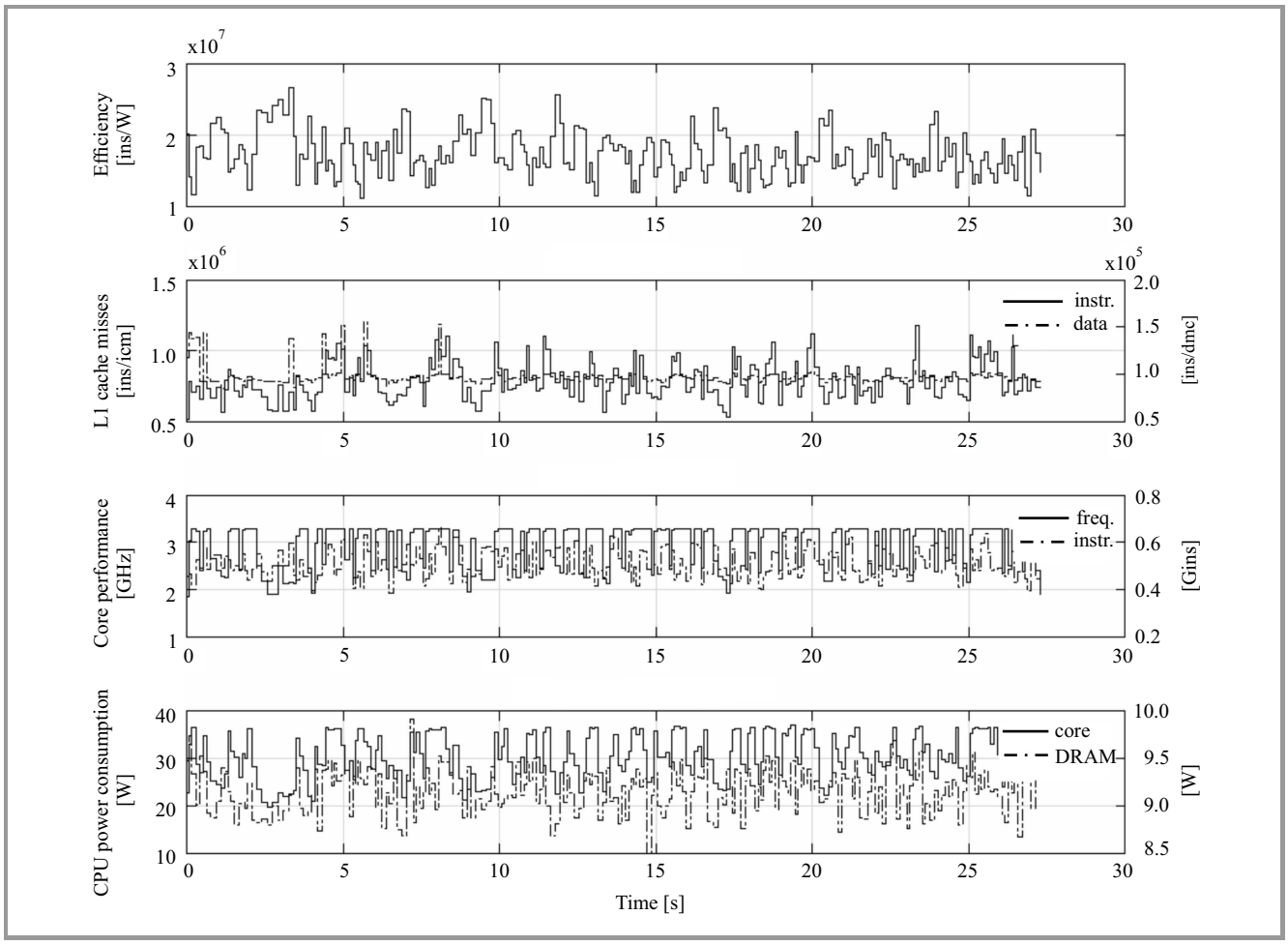


Fig. 3. Server performance and energy-efficiency trace (`intel_pstate powersave`).

7. Summary

Currently developed techniques of server power control exploit increasing possibilities provided by high-resolution sensors of modern computing hardware and software. This paper presents a brief overview of performance and power consumption monitoring tools available in Linux systems. It is argued that the measurements collected at high sampling rate can be used to develop maximally informative power consumption metrics and accurate dynamical processing models for the purpose of energy-aware design of server controllers.

This research was partially supported by the National Science Centre (NCN) under the grant no. 2015/17/B/ ST6/ 01885.

References

[1] J. Koomey, *Growth in Data Center Electricity Use 2005 to 2010*. Oakland, CA: Analytical Press, 2011.

[2] B. Subramaniam and Wu-chun Feng, "Towards energy-proportional computing for enterprise-class server workloads", in *Proc. 4th ACM/SPEC Int. Conf. Perform. Engin. ICPE 2013*, Prague, Czech Republic, 2013, pp. 15–26.

[3] J. Dongarra *et al.*, "The international exascale software project roadmap", *Int. J. High Perform. Comput. Appl.*, vol. 25, no. 1, pp. 3–60, 2011.

[4] S.-Y. Jing, S. Ali, K. She, and Y. Zhong, "State-of-the-art research study for green cloud computing", *The J. of Supercomput.*, vol. 65, no. 1, pp. 445–468, 2013.

[5] V. K. Vavilapalli *et al.*, "Apache hadoop yarn: yet another resource negotiator", in *Proc. 4th Ann. Symp.on Cloud Comput. SOCC'13*, Santa Clara, CA, USA, 2013, pp. 5:1–5:16, 2013.

[6] S. Jha, J. Qiu, A. Luckow, P. Mantha, and G. C. Fox, "A tale of two data-intensive paradigms: applications, abstractions, and architectures", in *Proc. 3rd IEEE Int. Congr. on Big Data BigData 2014*, Anchorage, AK, USA, 2014, pp. 645–652.

[7] Y. Georgiou, T. Cadeau, D. Glesser, D. Auble, M. Jette, and M. Hautreux, "Energy accounting and control with SLURM resource and job management system", in *Distributed Computing and Networking*, M. Chatterjee *et al.*, Eds., *LNCS*, vol. 8314, pp. 96–118. Springer, 2014.

[8] E. Niewiadomska-Szynkiewicz, A. Sikora, P. Arabas, M. Kamola, M. Mincer, and J. Kołodziej, "Dynamic power management in energy-aware computer networks and data intensive computing systems", *Future Gener. Comp. Syst.*, vol. 37, pp. 284–296, 2014 (doi: 10.1016/j.future.2013.10.002).

[9] M. P. Karpowicz, P. Arabas, and E. Niewiadomska-Szynkiewicz, "Energy-aware multilevel control system for a network of Linux software routers: design and implementation", *IEEE Syst. J.*, vol. PP, no. 99, pp. 1–12, 2015
(doi: 10.1109/JSUST.20152489244).

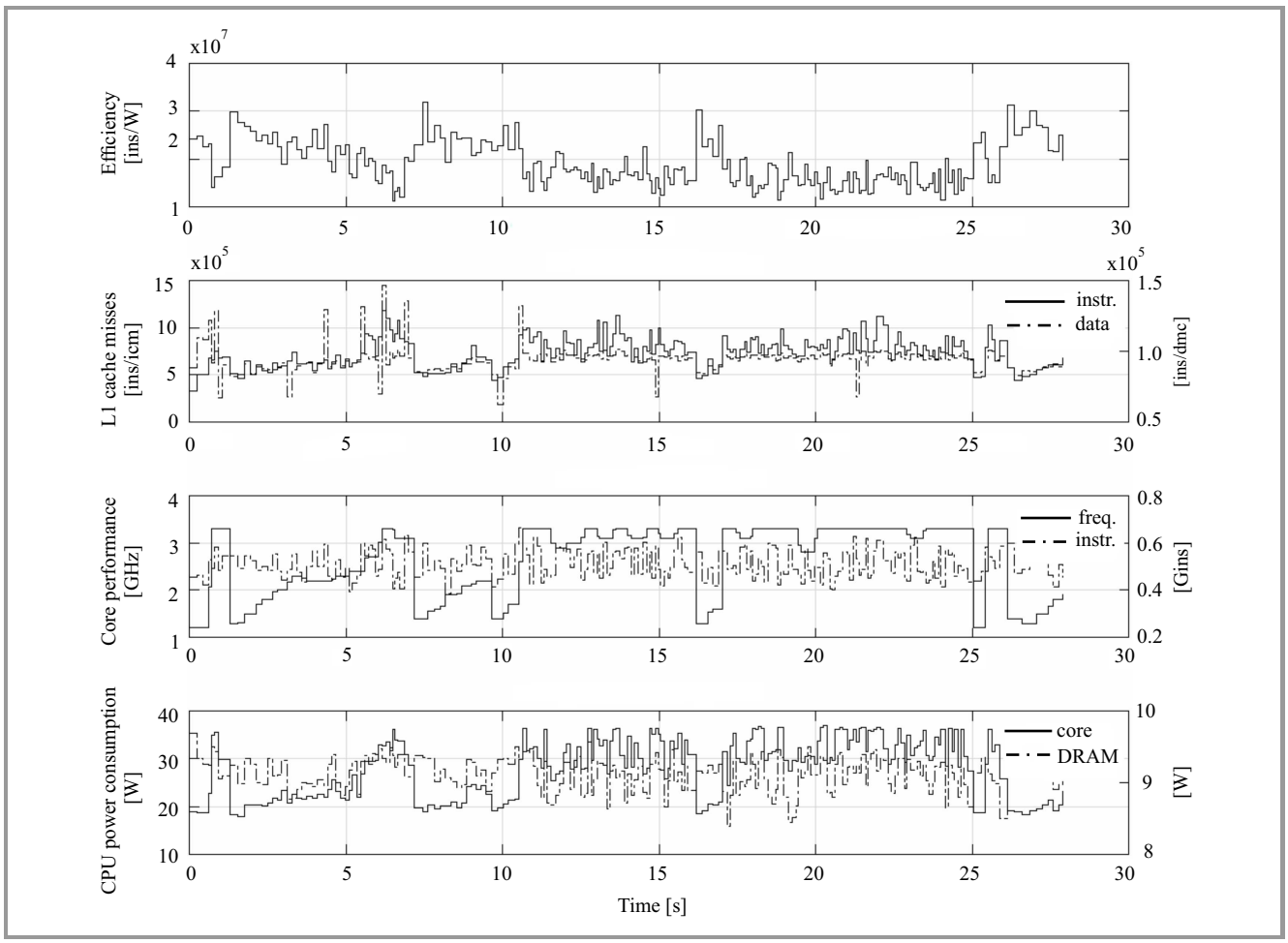


Fig. 4. Server performance and energy-efficiency trace (`intel_pstate performance`).

[10] P. Jaskóła, P. Arabas, and A. Karbowski, "Simultaneous routing and flow rate optimization in energy–aware computer networks", *Int. J. Appl. Mathem. & Comp. Sci.*, vol. 26, no. 1, pp. 231–243, 2016.

[11] A. Karbowski and P. Jaskóła, "Two approaches to dynamic power management in energy-aware computer networks – methodological considerations", in *Proc. of Feder. Conf. Comp. Science and Inform. Syst. FedCSIS 2015*, Łódź, Poland, 2015, vol. 5, pp. 1177–1182.

[12] ACPI Specification Document [Online]. Available: www.acpi.info

[13] J. H. Laros III, D. DeBonis, R. Grant, S. M. Kelly, M. Levenhagen, S. Olivier, and K. Pedretti, "High performance computing-power application programming interface specification", Tech. Rep. SAND2014-17061, Sandia National Laboratories, 2014.

[14] D. DeBonis *et al.*, "A power API for the HPC community", Sandia Report SAND2014-17061, Sandia National Laboratories, 2014.

[15] R. Bolla *et al.*, "Green Abstraction Layer (GAL): power management capabilities of the future energy telecommunication fixed network nodes", Techn. Rep. ES 203 237, ETSI, 2014.

[16] C. Lefurgy, K. Rajamani, F. Rawson, W. Felter, M. Kistler, and T. W. Keller, "Energy management for commercial servers", *Computer*, vol. 36, no. 12, pp. 39–48, 2003.

[17] L. A. Barroso, J. Clidaras, and U. Hölzle, *The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines*, 2nd ed. Morgan & Claypool Publ., 2013.

[18] L. Wang and S. U. Khan, "Review of performance metrics for green data centers: a taxonomy study", *The J. Supercomput.*, vol. 63, no. 3, pp. 639–656, 2013.

[19] T. Mastelic, A. Oleksiak, H. Claussen, I. Brandic, J.-M. Pierson, and A. V. Vasilakos, "Cloud computing: survey on energy efficiency", *ACM Comput. Surveys*, vol. 47, no. 2, pp. 33:1–33:36, 2015.

[20] B. Subramaniam, W. Saunders, T. Scogland, and Wu-chun Feng, "Trends in energy-efficient computing: A perspective from the Green500", in *4th Int. Green Comput. Conf. IGCC 2013*, Arlington, VA, USA, 2013, pp. 1–8.

[21] Standard Performance Evaluation Corporation (SPEC), SPEC Power and Performance Benchmark Methodology [Online]. Available: www.spec.org/power_ssj2008/

[22] K.-D. Lange, "Identifying Shades of Green: The SPECpower Benchmarks", *IEEE Computer*, vol. 42, no. 3, pp. 95–97, 2009.

[23] D. Molka, D. Hackenberg, R. Schöne, T. Minartz, and W. E. Nagel, "Flexible workload generation for HPC cluster efficiency benchmarking", *Comp. Science-Res. & Develop.*, vol. 27, no. 4, pp. 235–243, 2012.

[24] J. J. Dongarra, P. Luszczek, and A. Petitet, "The LINPACK Benchmark: past, present and future", *Concurr. & Comput.: Practice and Experience*, vol. 15, no. 9, pp. 803–820, 2003.

[25] A. Iosup, S. Ostermann, M. N. Yigitbasi, R. Prodan, T. Fahringer, and D. H. J. Epema, "Performance analysis of cloud computing services for many-tasks scientific computing", *IEEE Trans. Parall. & Distrib. Syst.*, vol. 22, no. 6, pp. 931–945, 2011.

[26] S. Bradner and J. McQuaid, "RFC 2544: Benchmarking methodology for network interconnect devices", Mar. 1999.

[27] P. Arabas and M. Karpowicz, "Server power consumption: measurements and modeling with MSRs", in *Challenges in Automation, Robotics and Measurement Techniques*, R. Szewczyk, C. Zieliński, and M. Kaliczyńska, Eds., *Advances in Intelligent Systems and Computing*, vol. 440, pp. 233–244. Springer, 2016.

[28] M. P. Karpowicz and P. Arabas, "Preliminary results on the Linux libpcap model identification", in *Proc. 20th Int. Conf. Methods & Models Autom Robot MMAR 2015*, Międzyzdroje, Poland, 2015, pp. 1056–1061.

[29] SNIA Emerald, SNIA Emerald Power Efficiency Measurement Specification [Online]. Available: www.snia.org

[30] Storage Performance Council (SPC), Storage Performance Council SPC Benchmark 2/Energy Extension [Online]. Available: www.storageperformance.org

[31] D. Hackenberg *et al*., "Power measurement techniques on standard compute nodes: A quantitative comparison", in *Proc. IEEE Int. Symp. on Perform. Anal. Syst. & Software ISPASS 2013*, Austin, TX, USA, 2013, pp. 194–204.

[32] J. Mair, D. Eyers, Z. Huang, and H. Zhang, "Myths in power estimation with performance monitoring counters", *Sustain. Comput.: Inform. & Syst.*, vol. 4, no. 2, pp. 83–93, 2014.

[33] M. E. Mehdi Diouri *et al.*, "Assessing power monitoring approaches for energy and power analysis of computers", *Sustainable Comput.: Informatics and Syst.*, vol. 4, no. 2, pp. 68–82, 2014.

[34] Intel Intelligent Power Node Manager [Online]. Available: www.intel.com

[35] D. Hackenberg *et al*., "HDEEM: high definition energy efficiency monitoring", in *Proc. IEEE Energy Efficient Supercomput. Worksh. SC14*, New Orleans, LA, USA, 2014, pp. 1–10.

[36] M. F. Dolz, M. R. Heidari, M. Kuhn, T. Ludwig, and G. Fabregat, "ARDUPOWER: A low-cost wattmeter to improve energy efficiency of HPC applications", in *Proc. 6th Int. Green Comput. Conf. & Sustain. Comput. Conf. IGSC 2015*, Las Vegas, NV, USA, 2015, pp. 1–8.

[37] Intel IA-64 and IA-32 Architectures Software Developer's Manual [Online]. Available: www.intel.com

[38] NVML API Reference Manual, 2012 [Online]. Available: http://developer.nvidia.com

[39] T. Ilsche, D. Hackenberg, S. Graul, R. Schöne, and J. Schuchart, "Power measurements for compute nodes: improving sampling rates, granularity and accuracy", in *Proc. 6th Int. Green Comput. Conf. and Sustain. Comput. Conf. IGSC 2015*, Las Vegas, NV, USA, 2015.

[40] V. M. Weaver, M. Johnson, K. Kasichayanula, J. Ralph, P. Luszczek, D. Terpstra, and S. Moore, "Measuring energy and power with PAPI", in *Proc. 41st Int. Conf. Parallel Proces. Worksh. ICPPW 2012*, Pittsburgh, PA, USA, 2012, pp. 262–268.

[41] L. Taniça, A. Ilic, P. Tomás, and L. Sousa, "Schedmon: A performance and energy monitoring tool for modern multi-cores", in *Euro-Par 2014: Parallel Processing Workshops*, *LNCS*, vol. 8806, pp. 230–241. Springer, 2014.

[42] Performance Application Programming Interface (PAPI) [Online]. Available: icl.cs.utk.edu/papi

[43] Unofficial Linux Perf. Events Performance Counter Weg Page [Online]. Available: web.eece.maine.edu/∼vweaver/projects/perf_events

[44] M. Gerndt, E. César, and S. Benkner, Eds., *Automatic Tuning of HPC Applications*. Shaker Verlag, 2015.

[45] M. P. Karpowicz, "Energy-efficient CPU frequency control for the Linux system", *Concur. & Computat.: Pract. & Exper.*, vol. 28, no. 2, pp. 420–437, 2016 (cpe.3476).

[46] D. Brandewie, "Intel P-state driver" [Online]. Available: www.kernel.org/doc/

[47] M. P. Karpowicz, P. Arabas, and E. Niewiadomska-Szynkiewicz, "Design and implementation of energy-aware application-specific CPU frequency governors for the heterogeneous distributed computing systems", *Future Generation Computer Systems*, available online, 2016 (doi: 10.1016/j.future.2016.05.011).

Michał Karpowicz received his Ph.D. in 2010. He is an Assistant Professor of Computer Science at the Research and Academic Computer Network (NASK) and the Warsaw University of Technology. His research interests focus on stochastic control theory, control engineering, game theory and network optimization.

E-mail: M.Karpowicz@elka.pw.edu.pl
Research and Academic Computer Network (NASK)
Wąwozowa st 18
02-796 Warsaw, Poland

Piotr Arabas received his Ph.D. in Computer Science from the Warsaw University of Technology, Poland, in 2004. Currently he is an Assistant Professor at the Institute of Control and Computation Engineering at the Warsaw University of Technology. Since 2002 with Research and Academic Computer Network (NASK). His research area focuses on modeling computer networks, predictive control and hierarchical systems.

E-mail: P.Arabas@elka.pw.ewu.pl
Research and Academic Computer Network (NASK)
Wąwozowa st 18
02-796 Warsaw, Poland

Institute of Control and Computation Engineering
Warsaw University of Technology
Nowowiejska st 15/19
00-665 Warsaw, Poland

Paper

Energy-saving Algorithms for the Control of Backbone Networks: A Survey

Mariusz Kamola[1,2], Ewa Niewiadomska-Szynkiewicz[1,2], Piotr Arabas[1,2], and Andrzej Sikora[2]

[1] *Institute of Control and Computation Engineering, Warsaw University of Technology, Warsaw, Poland*
[2] *Research and Academic Computer Network (NASK), Warsaw, Poland*

Abstract—The rapid growth of energy demand by wired IP networks can be mitigated on hardware and software levels. While upgrading to more efficient transmission media still brings biggest savings, we take a look here at power-saving algorithms that combine the capability of setting networking equipment in arbitrary energy states which, combined with profound knowledge of the network traffic matrix, leads to considerable complex optimization problem formulations. Alternatively, lightweighted heuristic approaches are presented, built on much simpler network model but still capable to perform energy-efficient traffic engineering.

Keywords—green routing, mixed integer programming, OSPF heuristics, power-save networks.

1. Introduction

One may point out several reasons for exponential growth traffic observed in contemporary networks. The increase of the number of connected devices, often related to progress in developing countries, as well as the progress of the Internet of Things, is important but still cannot account fully for the nature of the phenomenon. The main reason of its exponential nature is the type of content being transmitted – the multimedia and machine-to-machine communication still have big growth potential. Although the transmission infrastructure keeps up to the pace of demand, the resulting energy consumption, with its economic and environmental consequences, can be frightening.

There are two main technology domains where power consumption savings can be obtained: hardware and software. In wired networks, being the object of interest here, introduction of optical transmission technologies has become the main factor for reducing energy demand. Likewise, reduction of power consumption by on-board devices due to application of modern material technology has played its positive role.

Considered all the above, there is still room for further savings by exploiting the software domain. Any algorithmic framework for traffic admission and control may now utilize the two general power-saving capabilities, offered by majority of components:

- smart standby – automated or controlled deactivation of a hardware component when there is no load,
- dynamic power scaling – adaptation of power used to the actual load on that component.

The two common techniques of power scaling are: adaptive rate (AR) and low power idle (LPI). AR reduces power demand by scaling the data processing capabilities of a device, while LPI puts the device or its component into a "paused", low power mode during short inactivity periods. While utilizing dynamic power scaling often involves deep modifications in the design of software and hardware components of computing and network devices, the smart standby method requires only coordination among these devices to carefully rearrange the data transmission and processing loads that results from switching off selected devices or their components.

Most personal computers implement both AR and LPI techniques. The Advanced Configuration and Power Interface (ACPI) specification described in [1] defines a number of energy-aware states attained via voltage and clock frequency scaling and idle states in which the processor is in the standby mode. Development of APIs and management tools is, without a doubt, essential for optimal utilization of computing resources. On the other hand, system-wide regulation of power consumption needs to be commanded by a centralized management framework, capable of collecting and processing detailed measurements, and taking real-time coordinated actions across the data computing cloud infrastructure. The taxonomy of the energy and power management in computer networks can be found in [2]–[8]. This paper presents a survey of selected approaches described in literature.

2. Layered Architecture of Network Appliances – Basic Power Consumption Profiles

To design mechanisms for energy saving in computer networks, power consumption profiles of the network appliances and their components have to be identified. The architecture of modern network nodes (switches or routers), in many aspects is similar to that of a PC. Each router is a multi-chassis device composed of many entities: processor, chassis, line cards, communication ports, power supply,

fan, etc. Each component must be powered. The only specific element – the switching fabric – may be considered as a kind of a specialized processor connected to communication buses. The main difference with respect to a PC is the number and power consumed by line cards. While a general-purpose PC usually hosts at most several network interfaces consuming only a fraction of its power, in an average switch or router there are usually from tens to hundreds of network ports located on several line cards.
In general, most control mechanisms for energy consumption management take into account such hierarchical internal layout of network devices. The three layers are commonly distinguished, from bottom to the top:

- communication interfaces (ports),
- line cards,
- the whole device (a router or a switch).

In general, each component is powered and can operate in K energy states defined as power settings, enumerated $k = 1, \ldots, K$. When adjusting power state in a higher layer, it is necessary to take into account that layered architecture, e.g. it is not possible to decrease energy state of a line card without affecting (lowering) energy states of the communication ports it hosts.
The legacy network devices can operate only in two energy states: deep sleep, $k = 1$, with negligible power consumption, and active with full power, $k = 2$. The resulting power consumption $P_d(q)$ for total traffic q, served by any component ($d := r$ for a router, $d := c$ for a line card, $d := e$ for a link connecting two interfaces), is as follows [9]:

$$P_d(q) = \begin{cases} P_{d1} & \text{if} \quad q = 0, \\ P_{d2} & \text{if} \quad q > 0, \end{cases} \tag{1}$$

where P_{d1} and P_{d2} denote fixed power levels associated to the device d in deep sleep (state 1) and active (state 2) states, respectively.
Modern network components, equipped with mechanisms for dynamic power management (e.g. InfiniBand cards switching between 1x and 4x mode or bundled WAN links [10], [11]), can operate in a number of energy states ($K > 2$), differing by power usage. Those extra states stand for power scaling and standby techniques. The 802.3az standard [12] defines the implementation of low power idle for Ethernet interfaces.
Numerous classes of analytical models have been proposed describing component power consumption as a function of its load. The simplest approach is to extend the basic on-off model (1) by extra states and assuming a fixed power level P_{dk} in each state [7]. Other authors [9], [13] propose a piecewise linear extension to 1.

3. Power Saving in Whole Network is an Optimization Problem

The common approach to power reduction in a network as a whole is to formulate an optimization problem similar to the traditional network design problem [14] or QoS provisioning task [15], [16], but with the cost function defined as a sum of power consumed by all components of the network. If power profiles are as in Eq. (1), then in contrary to the traditional network design problem, transmission paths should be accomplished through as few devices as possible, instead of balancing traffic in a whole network. The aim is therefore to fill up the active links with traffic. Typically, backbone network infrastructure is to some degree redundant to provide the required level of reliability. Therefore, to reduce power consumption some parts of the network may be switched off or the speed of processors and links may be reduced, if there exist adequate technical capabilities. According to recent studies concerning internet service providers' networks [17], [18], the total energy consumption may be substantially reduced by employing such approaches.

Various formulations of a network energy saving problem are provided and discussed in the literature; starting from mixed integer programming formulation, to its relaxation in order to obtain a continuous problem formulation and employ a simple heuristics. Moreover, various energy models of the devices are used. The common aim is to find the optimal network configuration, i.e. the combination of energy states that would bring maximum power saving without degrading service quality. In general, due to high dimensionality and complexity of such optimization problem, linear energy profiles are preferred. Some authors limit the number of energy states of network equipment (routers and cards) to "enabled" and "disabled" [19], and use energy profile (1). To relax the optimization problem further, they also postulate the use of multi-path routing, typically impractical and avoided in reality. However, the recent trend in green networking, as it has been already mentioned, is to develop devices with greater number of energy states ($K > 2$) [20], [21]. It is obvious that handling such cases implies larger dimensionality of the optimization problem and more sophisticated dependencies among network components. Consequently, large mixed-integer linear problems are formulated [22] to determine the most profitable set of devices to be safely brought down. The common approach to mitigate the complexity problem is to aggregate nodes and flows to decrease the dimensionality [19].

Another branch of research tries to exploit specific properties of optical transport layer to scale link rates by selectively switching off fibers composing them [23] or even to build a two-level model with IP layer set upon optical devices layer [11].

To conclude, the major difficulty caused by layered multi-state architecture is the complexity of the optimization problem, much harder to solve than typical graph optimization problems. Such complexity has its roots in NP-completeness problem formulation, interdependencies between data paths, and the requirement for flow aggregation. Furthermore, energy profiles are often non-convex, making continuous relaxation difficult to work properly, introducing instability and solution suboptimality [24].

3.1. Layered Architecture – Definitions

Let us consider a computer network consisting of the following components:

- R routers: $r = 1,\ldots,R$,
- C line cards $c = 1,\ldots,C$ and
- I communication ports $i = 1,\ldots,I$.

As outlined in Section 2, layered architecture of a router is assumed, i.e., each router is equipped with a number of line cards, and each card contains a number of physical communication ports. E links ($e = 1,\ldots,E$) connect all ports, pair by pair. The ports and the connecting links support K energy states (EASs) labeled $k = 1,\ldots,K$. Assume that two ports connected by the e-th link must be in the same state k. In general, the corresponding power $P_{\text{net}}(q)$ consumed by the whole network can be calculated as a total of power consumed by every network device. Capacity of link e in state k is defined as q_{ek}. V_d denotes the total traffic demand for a link transmitting data from source port s_d to the destination port t_d. The following subsections present selected formulations of network power saving problems.

3.2. Optimization for Stepwise Profiles

The power profile model of each network device can be defined by a stepwise function describing power consumption in a given state (see Fig. 1a):

$$P_{dk}(q,k) = \begin{cases} P_{d1} & \text{if} \quad q = 0, \\ P_{dk} & \text{if} \quad q > 0, \end{cases} \tag{2}$$

where P_{d1} denotes fixed power level consumed by the device d in deep sleep state and P_{dk} – the power consumed in the k-th active energy state, $k = 2,\ldots,K$.
The complete network management problem, stated in terms of binary variables k and assuming that routers and line cards can operate only in two states: $k \in \{1,2\}$, deep sleep or active – cf. (1), and communication interfaces can operate in K states, $k = 1,\ldots,K$ – cf. (2) is formulated as follows:

$$\min_{x_r,x_c,x_{ek},u_{ed}} \left[P_{\text{net}} = \sum_{r=1}^{R} P_r x_r + \sum_{c=1}^{C} P_c x_c + \sum_{e=1}^{E} \sum_{k=1}^{K} P_{ek} x_{ek} \right]. \tag{3}$$

This formulation is completed with binary parameters describing network and device topology: $l_{cp} = 1$ if the port p is located on card c, $a_{ep} = 1$ if the link e is wired to port p, $g_{rc} = 1$ if the card c is located in router r. Otherwise, they take zero values.
The whole problem is subject to the following constraints:

$$\forall_{e\in\{1,\ldots,E\}} \quad \sum_{k=1}^{K} x_{ek} \le 1, \tag{4}$$

$$\forall_{\substack{d\in\{1,\ldots,D\} \\ c\in\{1,\ldots,C\}}} \quad \sum_{i=1}^{I} l_{ci} \sum_{e=1}^{E} a_{ei} u_{ed} \le x_c, \tag{5}$$

$$\forall_{\substack{d\in\{1,\ldots,D\} \\ c\in\{1,\ldots,C\}}} \quad \sum_{i=1}^{I} l_{ci} \sum_{e=1}^{E} b_{ei} u_{ed} \le x_c, \tag{6}$$

$$\forall_{\substack{r\in\{1,\ldots,R\} \\ c\in\{1,\ldots,C\}}} \quad g_{rc} x_c \le x_r, \tag{7}$$

$$\forall_{\substack{d\in\{1,\ldots,D\} \\ r\in\{1,\ldots,R\} \\ i=s_d}} \sum_{c=1}^{C} g_{rc} l_{ci} \sum_{e=1}^{E} a_{ei} u_{ed} - \sum_{c=1}^{C} g_{rc} l_{ci} \sum_{e=1}^{E} b_{ei} u_{ed} = 1, \tag{8}$$

$$\forall_{\substack{d\in\{1,\ldots,D\} \\ r\in\{1,\ldots,R\} \\ i\neq t_d, i\neq s_d}} \sum_{c=1}^{C} g_{rc} \sum_{i=1}^{I} l_{cp} \sum_{e=1}^{E} a_{ep} u_{ed} - \sum_{c=1}^{C} g_{rc} \sum_{i=1}^{I} l_{ci} \sum_{e=1}^{E} b_{ei} u_{ed} = 0, \tag{9}$$

$$\forall_{\substack{d\in\{1,\ldots,D\}, \\ r\in\{1,\ldots,R\}, \\ i=t_d}} \sum_{c=1}^{C} g_{rc} l_{ci} \sum_{e=1}^{E} a_{ei} u_{ed} - \sum_{c=1}^{C} g_{rc} l_{ci} \sum_{e=1}^{E} b_{ei} u_{ed} = -1, \tag{10}$$

$$\forall_{e\in\{1,\ldots,E\}} \quad \sum_{d=1}^{D} V_d u_{ed} \le \sum_{k=1}^{K} q_{ek} x_{ek}. \tag{11}$$

The meaning of binary decision variables is as follows: $x_r = 1$ if the router r is switched on, $x_c = 1$ if the card c is switched on, $x_{ek} = 1$ if the link e is in power state k, $u_{ed} = 1$ if a transmission path d traverses link e (zero – otherwise). P_r, P_c, P_{ek} denote the fixed power consumed by router, card and link, q_{ek} denotes the throughput of the link e in the state k, D stands for a number of assumed flows (demands).

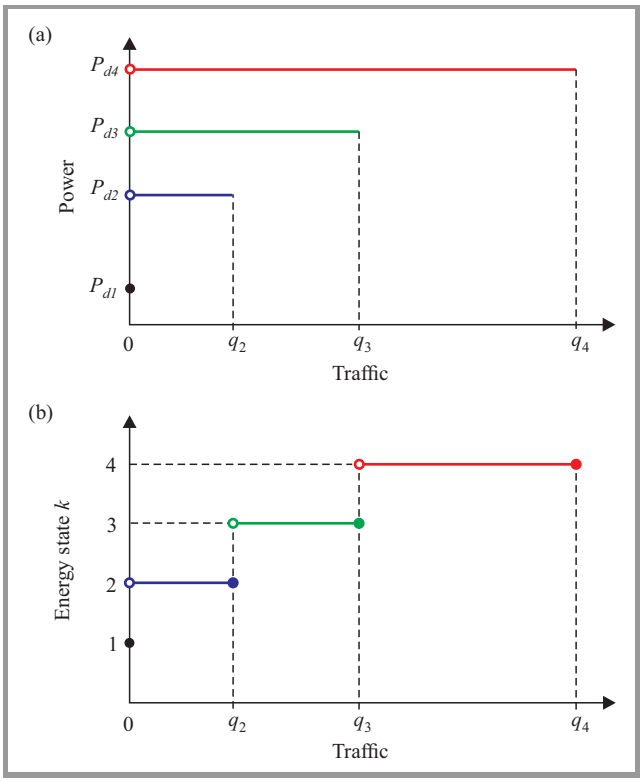


Fig. 1. Power consumption model – K states.

In the problem defined as above the constraints (4) assure that each link can be in one energy-aware state, the constraints (5)–(7) determine the number of routers and cards used for data transmission. The constrains (8)–(10) are formulated according to Kirchhoff's law applied for source, transit, and destination routers, and the constraint (11) assures that the flow will not exceed the capacity of a given link. In the effect of optimization, all components are put in energy states just appropriate to handle the traffic load, without unjustified waste of power. See Fig. 1b where an example power state activation strategy is presented.
To decrease the complexity and size of defined above optimization problem, a formulation of the network energy saving problem in proposed in [25] that is based on subsequent flows aggregations.

3.3. Optimization for Profiles Approximated by Piecewise Linear Functions

Let us assume that power profile of each link is a piecewise linear approximation of the correlation between an amount of transmitted data and energy consumed by a link. Such a model can be defined as follows (see Fig. 2a):

$$P_{ek}(q,k) = \begin{cases} P_{e1} & \text{if} \quad q = 0, \\ \alpha_{ek} + \gamma_{ek} q & \text{if} \quad q > 0, \end{cases} \tag{12}$$

where α_{ek} and γ_{ek} are coefficients of linear empirical approximation in k-th energy state, $k = 2, \ldots, K$.
The problem of energy-efficient network configuration, i.e. optimal routing and choosing energy states, for router and linecard power profiles as in (2), and the link link profiles as in (12), is stated as follows:

$$\min_{x_r, x_c, x_{ek}, u_{ed}} \left[P_{\text{net}} = \sum_{e=1}^{E} \sum_{k=1}^{K} P_{ek} x_{ek} + \right. \tag{13}$$
$$+ \sum_{e=1}^{E} \frac{\psi_{ek}(q_{ek})}{q_{ek}} \left(\sum_{d=1}^{D} V_d u_{ed} - \sum_{k=2}^{K} q_{ek} x_{ek} \right) +$$
$$\left. + \sum_{c=1}^{C} P_c x_c + \sum_{r=1}^{R} P_r x_r \right],$$

subject to the constraints (4)–(11). P_r, P_c, P_{ek} denote the fixed power consumed by router, line card and link, as in (3), while ψ_{ek} is the linear power profile in the energy state k.
In optimal configuration, each link is in a state that is adequate to its traffic load (cf. actual power profile in Fig. 2b), giving power savings wrt. the case with unmanaged ports, configured for best throughput.

3.4. Optimization for Stepwise Profiles, Relaxed to Continuous Formulation

Although the approach presented in Subsection 3.3 is easier to solve thanks to fewer constraints, it turns out to be still too complex for medium-size networks. In [7], the authors propose to apply a widely used technique of problem relaxation to continuous case, to cope with the complexity. The problem presented in Subsection 3.2 gets reformulated, with the decision variables x_r, x_c, x_{ek} and u_{ed} made continuous:

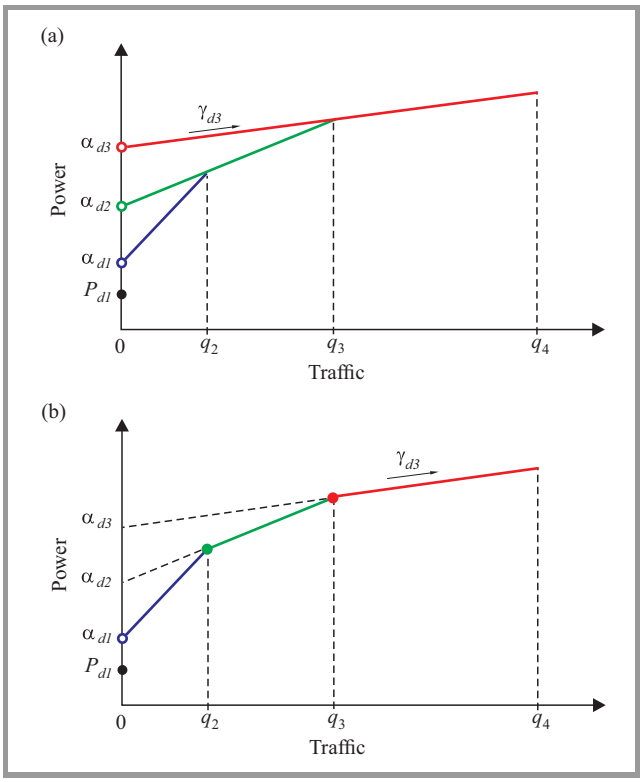


Fig. 2. Power consumption model – K energy states and piecewise linear approximation.

$$\min_{x_c, x_{ek}, x_r, u_{ed}} \left[P_{\text{net}} = \sum_{r=1}^{R} P_r x_r + \sum_{c=1}^{C} P_c x_c + \sum_{e=1}^{E} \sum_{k=1}^{K} P_{ek} x_{ek} \right] \tag{14}$$

subject to the constraints (7)–(11) and additional constraints:

$$\forall_{e \in \{1,\ldots,E\}} \quad x_{e1} \geq x_{ei} \geq \ldots \geq x_{eK}, \tag{15}$$

$$\forall_{\substack{e \in \{1,\ldots,E\} \\ k \in \{1,\ldots,K\} \\ c \in \{1,\ldots,C\}}} \quad \sum_{p=1}^{P} l_{cp} a_{ep} y_{ek} \leq x_c, \tag{16}$$

$$\forall_{\substack{e \in \{1,\ldots,E\} \\ k \in \{1,\ldots,K\} \\ c \in \{1,\ldots,C\}}} \quad \sum_{i=1}^{I} l_{ci} b_{ep} x_{ek} \leq x_c, \tag{17}$$

In the above formulation the power consumption and link load, for link e in state k, are related in the form of an incremental model wrt. the state. The current values of P_{ek} and q_{ek} are calculated recursively: $P_{ek} = pow_e(k) - pow_e(k-1)$ and $q_{ek} = load_e(k) - load_e(k-1)$, where $pow_e(k)$ is the power taken by link e in state k, and $load_e(k)$ is the respective link load. It is also assumed that each link supports more than one energy state. To account for that, the constraint (15) for loads in various states was added. Addi-

tionally, the utilized throughput in subsequent states must be sorted. The constraints (16) and (17) guarantee that x_r, x_c will effectively take binary values only.

Although this approach is much easier to solve thanks to decision variables continuity, it is hard to find a sufficiently efficient optimization routine for real-sized networks. The widely used approach to solve such complex optimization problems is to employ heuristics [23], [26]–[28]. A heuristics that proved capable to solve the relaxed optimization problem (14)–(17) was reported in [7], [29].

3.5. Energy Optimization Problem – Discrete Formulation and Utility Function

All the above optimization problem formulations share the disadvantage of relying heavily on the assumption that the origin-destination traffic, V_d is known. In practice, however, its estimation, let alone the prediction, is very costly and error prone. It happens because of the nature of the traffic: flows are self-similar, long-tailed and correlated, which makes the classical, general and elegant estimation method [30] completely useless here. Contemporary approaches try to employ principal component analysis [31], traffic sampling on selected links [32] and traffic mixture identification [33] to do the job, but without apparent success worth the measures taken.

Therefore, it has been proposed in [34], [35] to redefine the optimization problem as a two-criteria one, and to employ the notion of user utility function (i.e. valuation of transmission service) instead of an inflexible traffic demand vector. The reformulated optimization problem is in fact a mixture of the original goal function from (3), taken with weight α, and the total user utility, calculated over all flows and taken with weight $1-\alpha$. The utility can be perceived as a QoS-related criterion, Q_d for each flow d, which represents a penalty for not meeting the demanded flow rate V_d. $Q_d(v_d)$ is a convex and continuous function, decreasing in interval $[0, V_d]$. It is reaching its minimum (zero) at V_d, i.e. the point where user expectations are fully satisfied. Finally, the two criteria – power and QoS – get scalarized into a mixed integer problem of bandwidth allocation and routing:

$$\min_{\substack{x_r,x_c,x_{ek},u_{ed},v_d \\ r\in\{1,...,R\}, \\ e\in\{1,...,E\}, \\ k\in\{1,...,K\}, \\ d\in\{1,...,D\}}} \left\{ P_{\text{net}} = \right. \tag{18}$$

$$= \alpha \left[\sum_{e=1,3,5,...}^{E-1} \sum_{k=1}^{K} P_{ek}x_{ek} + \sum_{c=1}^{C} P_c x_c + \sum_{r=1}^{R} P_r x_r \right] +$$

$$\left. + (1-\alpha) \sum_{d=1}^{D} Q_d(v_d) \right\},$$

subject to constraints (5)–(8) and the following additional constraints:

$$\sum_{d=1}^{D} v_d u_{ed} \le \sum_{k=1}^{K} q_{ek}x_{ek},\ e = 1,3,...,E-1, \tag{19}$$

$$\sum_{d=1}^{D} v_d u_{ed} \le \sum_{k=1}^{K} q_{\bar{e}k}x_{\bar{e}k},\ e = 2,4,...,E, \tag{20}$$

$$0 \le v_d \le V_d,\ d \in \{1,...,D\}. \tag{21}$$

4. Control Frameworks for Dynamic Power Management

Various control frameworks for dynamic power management of the backbone network through energy-aware routing, traffic engineering and network equipment activity control have been designed and investigated [7], [8], [10], [36], [37]. They utilize smart standby and dynamic power scaling, i.e. the energy consumed by the network is minimized by deactivation of idle devices (routers, line cards, communication ports) and by reduction of the speed of link transfers. Implementation of such framework is described and discussed in [7]. It is assumed there that network devices can operate in energy-aware states differing by power usage (2). Two implementations: of a centralized and of a hierarchical framework operate at two control levels:

- local control level – algorithms executed by networking devices.
- central control level – algorithms executed in a dedicated node that controls the whole infrastructure.

4.1. Hierarchical Implementation Architecture of the Proposed Optimization Problems

In case of optimization problems formulated in Subsections 3.2 and 3.4, one may balance the scope of decisions made at central and local levels. In the centralized approach, the suggested power states of network devices are broadcast by the central unit, along with the optimal MPLS paths. Then, the activity of a local controller is reduced to effectuate central control signals, taking into account constraints related to current local load and incoming traffic. Meanwhile, in the hierarchical scenario the central unit does not directly force the configuration of the devices, specifying MPLS paths only. Then the local algorithms optimize the configuration of each component autonomously in order to achieve a preset trade off between power consumption and performance.

Usefulness and efficiency of the two abovementioned optimization problem formulations were verified by simulation and by laboratory tests for networks of various sizes and topologies, yielding power saving up to 50% in some cases.

4.2. Distributed Approaches

For practical reasons (scalability, reliability, protocols available) distributed control over the routes as well as individual configurations is an appealing alternative approach.

Distributed energy-aware network control algorithms extend existing mechanisms, as routing protocols (OSPF, BGP, MPLS) – cf. [37], [38], [39]. An important advantage from close cooperation with signaling protocols is that the network state can be observed easily, and may be used further to estimate flows and to reconstruct the traffic matrix [26].

The algorithm proposed in [40] brings selected nodes down or up; leaving the job of optimal routing to standard OSPF protocol. Classification of the nodes and links into "up" and "down" sets can be done by two heuristic algorithms – the more complex finds cliques in network graph in order to determine which parts of the network can be brought down with the least impact on QoS. The other one simply switches off the least loaded link or node. Simulations prove that the earlier approach gives much better performance. The algorithm has many practical advantages. In particular, it can be implemented in one designated router, while the remaining routers in the OSPF area simply implement the shortest path tree calculated and broadcast by it.

GRiDA algorithm [41] is an agent-based approach, where agents, or homogeneous autonomous local control algorithms, decide upon links that can be switched off in order to save power without impacting performance. Their strategy follows machine learning scheme: if shutting down a link causes congestion, the agent's decision is remembered as a wrong one. Agent algorithm incorporates much practical experience and detailed knowledge of energy profile of the device being under its control. Unlike in [40], it does not take into account information from the routing process itself, i.e. network topology and current routing paths.

Another interesting agent-based heuristic approach to energy-efficient traffic routing is proposed in [8]. Like in [40], agents cooperate closely with local OSPF processes, putting down or up local links. For a decision to be made, agents maintain three lists of links:

- that must be permanently switched on in order to maintain network connectivity,
- that have been switched off in order to reduce power consumption,
- that must be switched on in order to prevent congestion ("bypass" links).

Four strategies for link deactivation are proposed, of varying complexity and demand for extra information about the network state:

- LLL – switch off least loaded link in the network,
- LTL – switch off the least traversed link, i.e. with least number of flows using it,
- LDB – switch off the link that would cause the minimum increase of the total of traffic route lengths,
- LDM – switch off the link that would cause the minimal increase of the total of traffic volume over all links.

The proposed strategies for link activation in case of an overload, are as follows:

- RL – roll-back last deactivation,
- RB – activate the link that would cause the maximum decrease of the total of traffic route lengths,
- RM – activate the link that would cause the maximum decrease of the total of traffic volume over all links.

Performance of reasonable pairs of strategies has been examined, showing that perfect knowledge about the origin-destination matrix (LDM, RM) improves energy savings not much more than when simple local heuristics are applied. On the other hand, imperfect or inappropriate information about origin-destination matrix (LDB, RB) can make the things worse than in case when there is no energy-saving algorithm running at all. The proposed approach is viable to implement on any routing device, through command line and basic OSPF protocol.

An interesting alternative approach which also relies on OSPF operation is presented in [42], where non power saving routers coexist in the network with the power saving ones (PSRs). One of PSRs is selected as the coordinator; the remaining ones try to go offline when their local load is below a threshold – provided that network connectivity be maintained. Coordinator role is to schedule PSRs attempts. While in power-save state, a PSR wakes up periodically to observe the transit traffic it receives after having joined the network again. The approach has been verified through simulation, resulting in overall energy savings up to 18%.

5. Conclusion

The authors have shown here selected but representative approaches to save power in wired IP networks. The solutions can be put roughly into two classes: the proactive ones (Section 3) assume deep knowledge of the technology and the network state, put high technological requirements on the networking devices, and employ advanced optimization techniques to squeeze as much energy savings as possible. Those in the other class (Section 4) might be named the reactive ones: they try to supplement the already well rooted technologies (OSPF), which originally had nothing common with energy saving, with an energy-saving extra layer that tries to do its best with neither traffic matrix knowledge nor capability to choose among versatility of power states of each component. The only means of control there is to switch a device on or off, completely.

Both classes are valuable. While the proactive one constitutes the technological avant-garde (and is still lacking efficiency for larger problems), the reactive tries to makes makeshift but practical, "green" heuristic improvements to the networks in their current technological state.

Acknowledgment

This research was partially supported by the Polish National Science Centre (NCN) under the grant no. 2015/17/B/ST6/01885.

References

[1] "Advanced Configuration and Power Interface Specification, Revision 5.0", Hewlett-Packard Corp., Intel Corp., Microsoft Corp., Phoenix Technologies Ltd., and Toshiba Corp., 2011.

[2] A. P. Bianzino, C. Chaudet, D. Rossi, and J-L. Rougier, "A survey of green networking research", *IEEE Commun. Surveys & Tutor.*, vol. 14, no. 1, pp. 3–20, 2012.

[3] R. Bolla and R. Bruschi, "Energy-aware load balancing for parallel packet processing engines", in *Online Conf. Green Commun. GreenCom 2011*, New York, NY, USA, 2011, pp. 105–112.

[4] G. L. Valentini *et al.*, "An overview of energy efficiency techniques in cluster computing systems", *Cluster Computing*, vol. 16, no. 1, pp. 3–15, 2011 (doi: 10.1007/s10586-011-0171-x).

[5] J. R. Lorch and A. J. Smith, "Improving dynamic voltage scaling algorithms with PACE", in *Proc. ACM SIGMETRICS 2001 Int. Conf. Measur. & Model. Comp. Syst.*, Cambridge, MA, USA, 2001, pp. 50–61.

[6] R. Min, T. Furrer, and A. Chandrakasan, "Dynamic voltage scaling techniques for distributed microsensor networks" in *Proc. IEEE Comp. Soc. Worksh. on VLSI 2000*, Orlando, FL, USA, 2000, pp. 43–46.

[7] E. Niewiadomska-Szynkiewicz, A. Sikora, P. Arabas, M. Kamola, M. Mincer, and J. Kołodziej, "Dynamic power management in energy-aware computer networks and data intensive systems", *Future Gener. Comp. Syst.*, vol. 37, pp. 284–296, 2014.

[8] M. Kamola and P. Arabas, "Shortest path green routing and the importance of traffic matrix knowledge", in *24th Tyrrhenian Int. Worksh. Digit. Commun. – Green ICT (TIWDC 2013)*, Genoa, Italy, 2013, pp. 1–6.

[9] J. Restrepo, C. Gruber, and C. Machuca, "Energy profile aware routing", in *Proc. IEEE Int. Conf. Commun. ICC 2009*, Dresden, Germany, 2009, pp. 1–5.

[10] R. Bolla *et al.*, "Large-scale validation and benchmarking of a network of power-conservative systems using ETSI's green abstraction layer", *Trans. Emerging Telecommun. Technol.*, vol. 27, no. 3, pp. 451–468, 2016.

[11] F. Idzikowski, S. Orłowski, C. Raack, H. Rasner, and A. Wolisz, "Saving energy in IP-over-WDM networks by switching off line cards in low-demand scenarios", in *14th Conf. Optical Netw. Design & Model. ONDM 2010*, Kyoto, Japan, 2010.

[12] "IEEE 802.3az Energy Efficient Ethernet Task Force", IEEE, 2012 [Online]. Available: http://grouper.ieee.org/groups/802/3/az/public/index.html

[13] P. Jaskóła, P. Arabas, and A Karbowski, "Combined calculation of optimal routing and bandwidth allocation in energy aware networks", in *26th Int. Teletraff. Congress ITC 2014*, Karlskrona, Sweden, 2014, pp. 1–6.

[14] M. Pióro, M. Mysłek, A. Juttner, J. Harmatos, and A. Szentesi, "Topological design of MPLS networks", in *Proc. IEEE Global Telecommun. Conf. GLOBECOM'2001*, San Antonio, TX, USA, 2001.

[15] P. Jaskóła and K. Malinowski, "Two methods of optimal bandwidth allocation in TCP/IP networks with QoS differentiation", in *Proc. Int. Symp. Perform. Eval. Comp. & Telecommun. Syst. SPECTS'04*, San Jose, CA, USA, 2004, pp. 373–378.

[16] K. Malinowski, E. Niewiadomska-Szynkiewicz, and P. Jaskóła, "Price method and network congestion control", *J. Telecommun. Inform. Technol.*, no. 2, pp. 73–77, 2010.

[17] F. Bianco, G. Cucchietti, and G. Griffa, "Energy consumption trends in the next generation access network – a telco perspective", in *Proc. 29th Int. Telecommun Energy Conf. INTELEC 2007*, Rome, Italy, 2007, pp. 737–742.

[18] S. N. Roy, "Energy logic: a road map to reducing energy consumption in telecom munications networks", in *Proc. 30th Int. Telecommun. Energy Conf. INTELEC 2008*, San Diego, CA, USA, 2008.

[19] L. Chiaraviglio, M. Mellia, and F. Neri, "Minimizing ISP network energy cost: Formulation and solutions", *IEEE/ACM Trans. Netw.*, vol. 20, no. 2, pp. 463–476, 2011.

[20] R. Bolla *et al.*, "Econet deliverable d4.1 definition of energy-aware states", 2011 [Online]. Available: https://www.econet-project.eu/Repository/Document/331

[21] R. Bolla *et al.*, "Econet deliverable d2.1 end-user requirements, technology, specifications and benchmarking methodology", 2011 [Online]. Available: https://www.econet-project.eu/Repository/DownloadFile/291

[22] J. Chabarek *et al.*, "Power awerness in network design and routing", in *Proc. 27th Conf. Comp. Commun. INFOCOM 2008*, Phoenix, AZ, USA, 2008, pp. 457–465.

[23] W. Fisher, M. Suchara, and J. Rexford, "Greening backbone networks: reducing energy consumption by shutting off cables in bundled links", in *Proc. 1st ACM SIGCOMM Workshop on Green Networking Green Networking'10*, New Delhi, India, 2010, pp. 29–34.

[24] N. Vasić and D. Kostić, "Energy-aware traffic engineering", in *Proc. 1st Int. Conf Energy-Efficient Comput & Netw. E-ENERGY 2010*, Passau, Germany, 2010.

[25] P. Arabas, K. Malinowski, and A. Sikora, "On formulation of a network energy saving optimization problem", in *Proc. 4th Int. Conf. Commun. & Electron. ICCE 2012*, Hanoi, Vietnam, 2012, pp. 122–129.

[26] L. Chiaraviglio, M. Mellia, and F. Neri, "Energy-aware backbone networks: A case study", in *Proc. IEEE Int. Conf. on Commun. Worksh. ICC2009*, Dresden, Germany, 2009.

[27] M. Zhang, Ch. Yi, B. Liu, and B. Zhang, "GreenTE: Power-aware traffic engineering", in *Proc. 18th IEEE Int. Conf. Netw. Protoc. ICNP'2010*, Kyoto, Japan, 2010.

[28] G. Shen and R. S. Tucker, "Energy-minimized desig for IP over WDM networks", *J. Optical Commun. & Netw.*, vol. 1, no. 1, pp. 176–186, 2009.

[29] E. Niewiadomska-Szynkiewicz, A. Sikora, P. Arabas, and J. Kołodziej, "Control system for reducing energy consumption in backbone computer network", *Concurr. & Computat.: Pract. and Exper.*, vol. 25, pp. 1738–1754, 2013 (doi: 10.1002/cpe.2964).

[30] Y. Vardi, "Network tomography: Estimating source-destination traffic intensities from link data", *J. of the American Statist. Association*, vol. 91, no. 433, pp. 365–377, 1996.

[31] Z. Wang, K. Hu, K. Xu, B. Yin, and X. Dong, "Structural analysis of network traffic matrix via relaxed principal component pursuit", *Comp. Networks*, vol. 56, no. 7, pp. 2049–2067, 2012.

[32] L. Nie, D. Jiang, and L. Guo, "A power laws-based reconstruction approach to end-to-end network traffic", *Jo. Netw. & Comp. Applic.*, vol. 36, no. 2, pp. 898–907, 2013.

[33] M. Kamola, "Estimation of correlated flows from link measurements", in *Proc. 20th Int. Conf. Methods & Models in Autom. and Robot. MMAR 2015*, Międzyzdroje, Poland, 2015, pp. 272–277.

[34] P. Jaskóła, P. Arabas, and A. Karbowski, "Simultaneous routing and flow rate optimization in energy-aware computer networks", *Int. J. Applied Mathem. & Comp. Sci.*, vol. 26, no. 1, pp. 231–243, 2016.

[35] A. Karbowski and P. Jaskóła, "Two approaches to dynamic power management in energy-aware computer networks - methodological considerations", in *Proc. Federat. Conf. Comp. Sci. & Inform. Syst. FedCSIS 2015*, Łódź, Poland, 2015, pp. 1177–1182.

[36] E. Niewiadomska-Szynkiewicz, A. Sikora, P. Arabas, and J. Kołodziej, "Control framework for high performance energy aware backbone network", in *Proc. Eur. Conf. Model. & Simul. ECMS 2012*, Koblenz, Germany, 2012, pp. 490–496.

[37] A. P. Bianzino, L. Chiaraviglio, and M. Mellia, "GRiDA: A green distributed algorithm for backbone networks", in *Online Conf. Green Commun. GreenCom 2011*, New York, NY, USA, 2011, pp. 113–119.

[38] A. Cianfrani, V. Eramo, M. Listani, M. Marazza, and E. Vittorini, "An Energy Saving Routing Algorithm for a Green OSPF Protocol", in *Proc. IEEE INFOCOM Conf. Comp. Commun.*, San Diego, CA, USA, 2010.

[39] F. Cuomo, A. Abbagnale, A. Cianfrani, and M. Polverini, "Keeping the connectivity and saving the energy in the Internet", in *Proc. IEEE Conf. Comp. Commun. Worksh. INFOCOM WKSHPS 2011*, Shanghai, China, 2011, pp. 319–324 (doi: 10.1109/INFCOMW.2011.5928831).

[40] A. Cianfrani, V. Eramo, M. Listanti, and M. Polverini, "An OSPF enhancement for energy saving in IP networks", in *Proc. IEEE Conf. Comp. Commun. Worksh. INFOCOM WKSHPS 2011*, Shanghai, China, 2011, pp. 325–330 (doi: 10.1109/INFCOMW.2011.5928832).

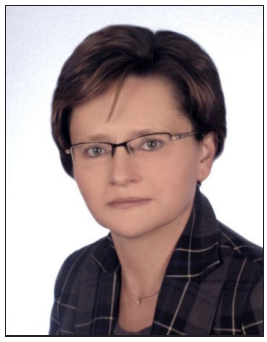
[41] A. P. Bianzino, L. Chiaraviglio, M. Mellia, and J.-L. Rougier, "GRiDA: Green distributed algorithm for energy-efficient IP backbone networks", *Computer Networks*, vol. 56, no. 14, pp. 3219–3232, 2012.

[42] K.-H. Ho and C.-C. Cheung, "Green distributed routing protocol for sleep coordination in wired core networks", in *6th Int. Conf. Networked Comput. INC 2010*, Gyeongju, South Korea, 2010, pp. 1–6.

Mariusz Kamola received his Ph.D. in Computer Science from the Warsaw University of Technology in 2004. Currently he is associate professor at Institute of Control and Computation Engineering at the Warsaw University of Technology. Since 2002 with Research and Academic Computer Network (NASK). His research focuses on big data analytics, social networks and Internet of Things.

E-mail: Mariusz.Kamola@nask.pl
Research and Academic Computer Network (NASK)
Wąwozowa st 18
02-796 Warsaw, Poland

E-mail: M.Kamola@ia.pw.edu.pl
Institute of Control and Computation Engineering
Warsaw University of Technology
Nowowiejska st 15/19
00-665 Warsaw, Poland



Ewa Niewiadomska-Szynkiewicz received her Ph.D. and D.Sc. degrees in 1995 and 2005, respectively. She is currently a professor of control and computer engineering at the Warsaw University of Technology, head of the Complex Systems Group. She is also the Director for Research of Research and Academic Computer Network (NASK), and the author and co-author of three books and over 140 journal and conference papers. Her research interests focus on complex systems modeling, optimization and control, computer simulation, parallel computation, computer networks and ad hoc networks. She was involved in a number of research projects including EU projects, coordinated the groups activities, managed organization of a number of national-level and international conferences.

E-mail: ens@ia.pw.edu.pl
Institute of Control and Computation Engineering
Warsaw University of Technology
Nowowiejska 15/19, 00-665 Warsaw, Poland

E-mail: ewan@nask.pl
Research and Academic Computer Network (NASK)
Wąwozowa st 18
02-796 Warsaw, Poland

Andrzej Sikora received his Ph.D. in Computer Science from the Warsaw University of Technology (WUT), Poland, in 2015. Since 2005 with Research and Academic Computer Network (NASK). His research area focuses on parallel and distributed simulation, computer networks, ad hoc networks and database systems.

E-mail: andrzej.sikora@nask.pl
Research and Academic Computer Network (NASK)
Wąwozowa st 18
02-796 Warsaw, Poland

Piotr Arabas – for biography, see this issue, p. 12.

Paper

New Developments in a Two-criteria Approach to Dynamic Power Management in Energy-aware Computer Networks

Andrzej Karbowski[1,2] and Przemysław Jaskóła[2]

[1] *Institute of Control and Computation Engineering, Warsaw University of Technology, Warsaw, Poland*
[2] *Research and Academic Computer Network (NASK), Warsaw, Poland*

Abstract—In the paper authors continue the development of a model of dynamic power management in energy-aware computer networks, where two criteria: energy consumption and the quality of service are considered. This approach is appropriate when the routing problem with fixed demands is inadmissible. The formulation introducing edge indices is modified and tests on problems of different sizes are performed.

Keywords—data intensive computing, energy-aware network, energy-aware routing, dynamic power management, MIQP problems, traffic engineering.

1. Introduction

The methods for increasing energy efficiency of computer networks gained much attention last years. The reason is, that we are witnessing a rise of energy costs, customer increase, more on-demand services using cloud architectures, mobile Internet, a diffusion of broadband access and a growing number of services offered by Internet service providers. The growth of the energy consumption by network infrastructure may be well illustrated by the overall energy requirements of European Internet operators. In 2005 they needed 14 TWh, in 2010 – 21 TWh, and the forecast for year 2020 is 36 TWh [1].

At the same time the capacity surplus becomes a standard in almost all networks. Consequently, so-called green network technologies are quickly becoming a high-priority issue for the Internet [1], [2].

Efforts to reduce power consumption in telecommunication networks follow in two mutually related directions – design of a more efficient equipment and development of energy-aware network control strategies and protocols. Initial efforts were aimed at assessment of energy characteristics of network equipment and building elementary models [3]. However, it is possible to save even more energy by employing network-wide solutions.

The authors' earlier paper [4] presented a model of energy-aware router, an architecture of a control framework and two-criteria formulation of a network-wide energy saving optimization problem. A broad review of literature is presented there, hence it won't be repeated in this paper.

A two-criteria optimal routing and bandwidth allocation problem, taking into account the energy component, for a completely different network and cost model was recently presented in [5].

The objective was the minimization of the weighted sum of two components: total power utilized by network components and end-to-end Quality of Service (QoS) expressed by a quadratic utility function. In this model some parts of the network can be shifted to low energy mode as a result of the optimization algorithms, where both paths and flow rates are decision variables. It exploits the fact, that Internet traffic used to be elastic in a large part, which means, that a quality of service is little aggravated by small deviations from assumed flow rate.

In this paper the model mentioned above is presented first in a modified version, as regards to the edges. In the authors' opinion the new version is more convenient, because there is no necessity to remember the parities of links labels. Then the results of tests of this optimization model on network problems of different sizes are presented.

2. A Two-criteria Routing Problem

A hierarchical network model proposed in [4] and basically adapted from [6], [7], considers every single communication port $p \in \{1,\ldots,P\}$ of every line card of a router $r \in \{1,\ldots,R\}$ connected to an edge $e \in \{1,\ldots,E\}$. We do not consider individual cards of the router, as it is in [6], [7], because they do not bring anything into the model except additional summations.

Directed links connecting pairs of ports by an edge are denoted by $l \in \{1,\ldots,L\}$. Any network component can operate in $k \in \{1,\ldots,K\}$ energy states, but two ports connected by an edge are in the same state. A demand $d \in \{1,\ldots,D\}$ is characterized by its source s_d, the destination t_d node (router), the maximum volume V_d and the actual flow rate $v_d \in [0,V_d]$.

The topology of the physical network is described by four matrices of binary indicators: g_{pr}, a_{lp}, b_{lp}, which indicate, whether, respectively: port p belongs to the router r, link l is incoming to the port p and link l is outgoing from the

port p. If l is a link outgoing from the port p, the link $\tilde{l}$ denotes its partner link in the edge going in the opposite direction, that is:

$$b_{lp} = 1 \Longleftrightarrow a_{\tilde{l}p} = 1\,. \tag{1}$$

Let us introduce a vector valued function ζ, which for a given edge e determines two links forming it, in the increasing order, that is

$$\zeta(e) = \left[\begin{array}{c} \zeta_1(e) \\ \zeta_2(e) \end{array} \right] = \left[\begin{array}{c} l \\ \tilde{l} \end{array} \right],\ l < \tilde{l}\,. \tag{2}$$

The decision variables are two vectors of binary indicators x_p, z_r – whether the port p or router r is used for data transmission and two incidence matrices with elements: y_{ek} – whether the edge e is in the state k and u_{dl} – whether the demand d uses the link l and flow rates v_d.

A two criteria – i.e. reflecting energy costs F_{LNb} and QoS – mixed integer network problem of simultaneous optimal bandwidth allocation and routing may be formulated in the following way:

$$\min_{\substack{x_p,y_{ek},z_r,u_{dl},v_d,\\ p\in\overline{1,P},e\in\overline{1,E},k\in\overline{1,K}\\ r\in\overline{1,R},d\in\overline{1,D},l\in\overline{1,L}}} \left\{ F_{2C} = \alpha F_{LNb} + (1-\alpha)\sum_{d=1}^{D} Q_d(v_d) = \right.$$

$$= \alpha \left[\sum_{e=1}^{E}\sum_{k=1}^{K} \xi_{ek} y_{ek} + \sum_{p=1}^{P} W_p x_p + \sum_{r=1}^{R} T_r z_r \right] +$$

$$\left. +(1-\alpha)\sum_{d=1}^{D} Q_d(v_d) \right\}, \tag{3}$$

subject to the constraints:

$$\forall_{\substack{d=1,\ldots,D,\\ p=1,\ldots,P}} \quad \sum_{l=1}^{L} a_{lp} u_{dl} \le x_p\,, \tag{4}$$

$$\forall_{\substack{d=1,\ldots,D,\\ p=1,\ldots,P}} \quad \sum_{l=1}^{L} b_{lp} u_{dl} \le x_p\,, \tag{5}$$

$$\forall_{\substack{r=1,\ldots,R,\\ p=1,\ldots,P}} \quad g_{pr} x_p \le z_r\,, \tag{6}$$

$$\forall_{e=1,\ldots,E} \quad \sum_{k=1}^{K} y_{ek} \le 1 \tag{7}$$

$$\forall_{\substack{d=1,\ldots,D,\\ r=1,\ldots,R}} \sum_{p=1}^{P}\sum_{l=1}^{L} g_{pr} a_{lp} u_{dl} - \sum_{p=1}^{P}\sum_{l=1}^{L} g_{pr} b_{lp} u_{dl} = \left\{ \begin{array}{ll} -1 & r = s_d \\ 1 & r = t_d \\ 0 & \text{otherwise} \end{array} \right., \tag{8}$$

$$\sum_{d=1}^{D} v_d u_{d\zeta_1(e)} \le \sum_{k=1}^{K} M_{ek} y_{ek},\ e = 1,\ldots,E\,, \tag{9}$$

$$\sum_{d=1}^{D} v_d u_{d\zeta_2(e)} \le \sum_{k=1}^{K} M_{ek} y_{ek},\ e = 1,\ldots,E\,, \tag{10}$$

$$x_p, z_r \in \{0,1\}\ p \in \overline{1,P}, r \in \overline{1,R}, \tag{11}$$

$$y_{ek}, u_{dl} \in \{0,1\}\ e \in \overline{1,E}, k \in \overline{1,K}, d \in \overline{1,D}, l \in \overline{1,L}, \tag{12}$$

$$0 \le v_d \le V_d,\ d \in \overline{1,D}, \tag{13}$$

where M_{ek} and ξ_{ek} are, respectively, the capacity and the power consumption of the edge e in the state k, and W_p and T_r are power cost coefficients of the port p and the router r. F_{LNb} is the total power consumption by network devices component of the objective function, Q_d is a QoS related component. The latter represents a penalty for not achieving the assumed flow rate V_d by the flow d. $Q_d(v_d)$ is a convex and continuous function, decreasing on interval $[0, V_d]$. It is reaching minimum (zero) at V_d, the point in which user expectations are fully satisfied. The convexity of $Q_d(v_d)$ is associated with the conviction, that small deviations from the nominal throughput $\Delta = V_d - v_d$ are neglected by network users, while large deviations are noticed and should be avoided. Moreover, since $Q_d(v_d)$ is monotonically decreasing, it assures that the slope of the curve becomes steeper, as the rate v_d approaches zero. Constraints (4)–(6) determine the number of ports and routers that are used for data transmission. The conditions (7) assure, that each edge can be in one energy-aware state. The constraints (8) are formulated according to 1st Kirchhoff's law applied to source, destination and transit nodes, and finally, the constraints (9), (10) assure, that the flow will not exceed the capacity of a given edge. They are expressed in a more natural and convenient in implementation way than in the paper [4], where a trick based on parities of link labels was used. Now they concern edges instead of links and it is not necessary for links l and $\tilde{l}$ of the same edge be of different parities.

The above model exploits the fact, that Internet traffic used to be elastic in a large part, which means, that the QoS is only a little aggravated by small deviations from the assumed flow rate. The combined routing and rate control problem has to be solved, which leads to the solution feasible in terms of the formulated model, even when the traffic demand is greater than the capacity offered by the network. Moreover, in some cases a minor reduction of flow rates, which is accepted by the comprehensive model taking into account the elasticity of a demand, may allow to accommodate the traffic in a smaller number of links, thus allowing for further great reduction of power consumption.

The parameter $\alpha \in [0,1]$ is a scalarizing weight coefficient, which can be altered to emphasize any of the objectives.

In general, the formulation (3)–(13) has some drawbacks: it defines a mixed-integer nonlinear programming problem with nonconvex, bilinear link capacity constraints (9),(10). At present the leading solvers – e.g., CPLEX, Gurobi – can solve efficiently convex quadratic mixed-integer quadratically constrained problems MIQCP, with positive semidefinite matrices of constraints quadratic forms, which is not the case of (9), (10) constraints. The general nonlinear, mixed-integer, non-convex solvers are very slow.

Fortunately, the problem (3)–(13) can be quite easily transformed to the form accepted by fast mixed-integer solvers, what is described in Section 3.

3. Elimination of the Nonlinearity from Constraints

From the QoS components of the objective function F_{2C} (3) it is usually expected, that they assure so-called proportional-fairness of the allocations of the bandwidth, when the network is subject to a congestion [8]. Quadratic functions may be used to achieve it [9] (unfortunately, linear – not), so the objective function F_{2C} can be quadratic and convex.

The only problem that still remains to solve is nonconvex nonlinearity of the constraints (9), (10). It can be eliminated by a transformation found in [10].

It consists in the introduction of auxiliary variables $w_{dl} = v_d u_{dl}$, $d \in \overline{1,D}$, $l \in \overline{1,L}$ (denoting the part of a traffic rate in the link l assigned to the flow d) and the substitution of these inequalities with subsequent set of linear inequalities:

$$\forall_{e=1,\ldots,E} \quad \sum_{d=1}^{D} w_{d\zeta_1(e)} \le \sum_{k=1}^{K} M_{ek} y_{ek}\,, \tag{14}$$

$$\forall_{e=1,\ldots,E} \quad \sum_{d=1}^{D} w_{d\zeta_2(e)} \le \sum_{k=1}^{K} M_{ek} y_{ek}\,, \tag{15}$$

$$\forall_{\substack{d=1,\ldots,D,\\ l=1,\ldots,L}} \quad w_{dl} \le V_d u_{dl}\,, \tag{16}$$

$$\forall_{\substack{d=1,\ldots,D,\\ l=1,\ldots,L}} \quad w_{dl} \le v_d\,, \tag{17}$$

$$\forall_{\substack{d=1,\ldots,D,\\ l=1,\ldots,L}} \quad w_{dl} \ge v_d - V_d(1-u_{dl})\,, \tag{18}$$

$$\forall_{\substack{d=1,\ldots,D,\\ l=1,\ldots,L}} \quad w_{dl} \ge 0\,. \tag{19}$$

4. The Final Formulation of the Problem

Summing up, the final formulation of presented two criteria energy-aware integrated routing and flow control problem is as follows:

$$\min_{\substack{x_p,y_{ek},z_r,u_{dl},v_d,\\ p\in\overline{1,P},e\in\overline{1,E},k\in\overline{1,K}\\ r\in\overline{1,R},d\in\overline{1,D},l\in\overline{1,L}}} \left\{ F_{2C} = \alpha F_{LNb} + (1-\alpha)\sum_{d=1}^{D} Q_d(v_d) = \right.$$

$$= \alpha\left[\sum_{e=1}^{E}\sum_{k=1}^{K}\xi_{ek}y_{ek} + \sum_{p=1}^{P} W_p x_p + \sum_{r=1}^{R} T_r z_r\right] + \left. +(1-\alpha)\sum_{d=1}^{D} Q_d(v_d)\right\}, \tag{20}$$

subject to the constraints:

$$\forall_{\substack{d=1,\ldots,D,\\ p=1,\ldots,P}} \quad \sum_{l=1}^{L} a_{lp}u_{dl} \le x_p\,, \tag{21}$$

$$\forall_{\substack{d=1,\ldots,D,\\ p=1,\ldots,P}} \quad \sum_{l=1}^{L} b_{lp}u_{dl} \le x_p\,, \tag{22}$$

$$\forall_{\substack{r=1,\ldots,R,\\ p=1,\ldots,P}} \quad g_{pr}x_p \le z_r\,, \tag{23}$$

$$\forall_{e=1,\ldots,E} \quad \sum_{k=1}^{K} y_{ek} \le 1\,, \tag{24}$$

$$\forall_{\substack{d=1,\ldots,D,\\ r=1,\ldots,R}} \sum_{p=1}^{P}\sum_{l=1}^{L} g_{pr}a_{lp}u_{dl} - \sum_{p=1}^{P}\sum_{l=1}^{L} g_{pr}b_{lp}u_{dl} = \begin{cases} -1 & r = s_d \\ 1 & r = t_d \\ 0 & \text{otherwise} \end{cases}, \tag{25}$$

$$\forall_{e=1,\ldots,E} \quad \sum_{d=1}^{D} w_{d\zeta_1(e)} \le \sum_{k=1}^{K} M_{ek} y_{ek}\,, \tag{26}$$

$$\forall_{e=1,\ldots,E} \quad \sum_{d=1}^{D} w_{d\zeta_2(e)} \le \sum_{k=1}^{K} M_{ek} y_{ek}\,, \tag{27}$$

$$\forall_{\substack{d=1,\ldots,D,\\ l=1,\ldots,L}} \quad w_{dl} \le V_d u_{dl}\,, \tag{28}$$

$$\forall_{\substack{d=1,\ldots,D,\\ l=1,\ldots,L}} \quad w_{dl} \le v_d\,, \tag{29}$$

$$\forall_{\substack{d=1,\ldots,D,\\ l=1,\ldots,L}} \quad w_{dl} \ge v_d - V_d(1-u_{dl})\,, \tag{30}$$

$$\forall_{\substack{d=1,\ldots,D,\\ l=1,\ldots,L}} \quad w_{dl} \ge 0\,, \tag{31}$$

$$x_p, z_r \in \{0,1\} \; p \in \overline{1,P}, r \in \overline{1,R}\,, \tag{32}$$

$$y_{ek}, u_{dl} \in \{0,1\} \; e \in \overline{1,E}, k \in \overline{1,K}, d \in \overline{1,D}, l \in \overline{1,L}\,, \tag{33}$$

$$0 \le v_d \le V_d,\; d \in \overline{1,D}\,. \tag{34}$$

When QoS components $Q_d(v_d)$ are quadratic and convex, the obtained mixed-integer quadratic problem can be solved by effective MILP/MIQP solvers, such as CPLEX, Gurobi.

5. Numerical Evaluation

The problem (20)-(34) was formulated, implemented and solved by means of the CPLEX solver for the test network. The topology of an example network is shown in Fig. 1. It has been inspired by an access/metropolitan segment of a telecom operator network, which was presented in [11]. Its size is however reduced compared with the original.

The access nodes, represented by circles indexed 1–3, are sources and destinations of traffic flows. Transit nodes (5–8) perform traffic switching, and a peering node, labeled T, provides access to the ISP transport network and the Internet.

In the presented example the number of routers $R = 9$, the number of edges $E = 14$ and the number of links and ports $L = P = 28$.

As the (penalty for not achieving) QoS functions we took:

$$Q_d(v_d) = \frac{1}{2}\left(v_d - V_d\right)^2. \tag{35}$$

It was possible for each edge to operate in $K = 5$ energy-aware states. The throughput of a given edge $e \in \overline{1,E}$ and the power consumption in energy-aware state $k \in \overline{1,K}$ were

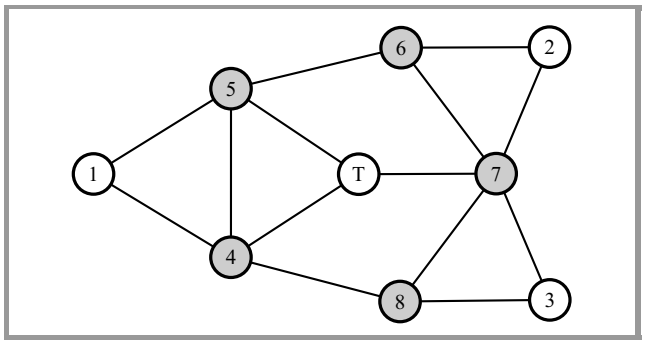


Fig. 1. Example network.

as follows: $M_{e1} = 2$, $\xi_{e1} = 20$; $M_{e2} = 4$, $\xi_{e2} = 40$; $M_{e3} = 6$, $\xi_{e3} = 60$; $M_{e4} = 8$, $\xi_{e4} = 80$; $M_{e5} = 10$, $\xi_{e5} = 100$. As the power cost coefficients we took for ports $W_1 = W_2 = \ldots = W_{28} = 90$ and for routers $T_1 \ldots T_3 = 1000$, $T_4 \ldots T_8 = 10000$, $T_9 = 3000$. The scalarizing coefficient was taken $\alpha = 10^{-3}$. This value was experimentally determined to ensure that from possible Pareto optimal solutions selected are these, which offer satisfactory QoS level in every run of the experiment.

In the first series of experiments a dependency between the number of binary variables and the time of computations have been examined. In subsequent experiments the number of traffic flows was increased, which resulted in a linear increase of the number of binary variables. The QoS function parameter V_d remained the same for every flow: $V_d = 10$. The computation time is presented in Fig. 2. It reveals a sharp increase in function of the problem dimension.

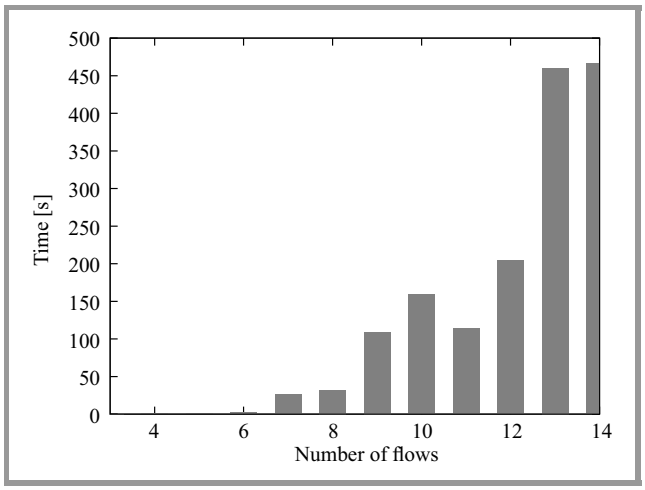


Fig. 2. Computation time as a function of number of traffic flows.

The applicability of this model depends on the chosen control scheme. In an on-line control the maximum acceptable computation time is closely related to the time between reconfigurations, which should be in the order of tens of minutes (see e.g. [11]). The observed trend indicates, that for bigger examples this limit can be exceeded.

The second series of experiments shows that the calculation time depends not only on the number of variables, but also on values of the parameters of the problem. In this case the number of flows was fixed to $D = 10$, while the parameter V_d was altered, which is presented in Fig. 3. Although the dimension of the problem has not changed, the computation time varies to the large extent between experiments.

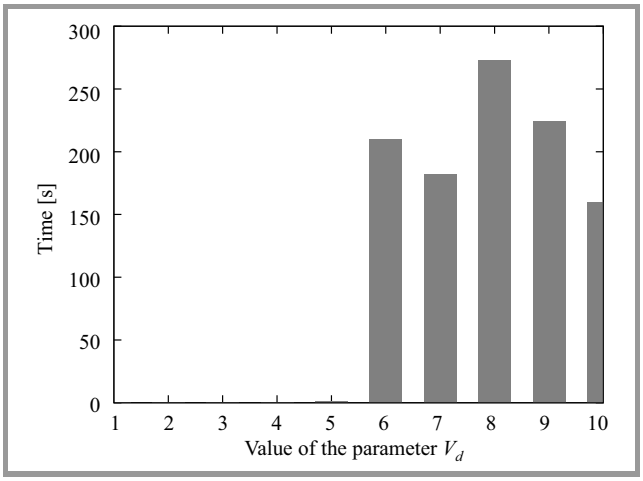


Fig. 3. Computation time as a function of a the parameter V_d.

All the results were obtained on PC with Intel Core i5-2430M CPU running at 2.4 GHz.

6. Conclusions

The authors modified the dynamic power management of energy-aware computer networks model presented in [4] to simplify it and make its implementation easier. It is still suggested to use a two-criteria model with flow rates as decision variables. When the QoS functions are quadratic and convex, it is possible to reformulate the problem in such a way, that the same standard solvers, e.g., CPLEX or Gurobi, can be used to find the solution. The resulting mixed-integer programming problem has more variables, but the additional ones are only real, not binary, what should not influence too much the time of calculations.

The performed numerical tests confirmed the appropriateness of the formulation for practical problems of small dimension. The computation times are however sensitive not only to the dimension of the problem, but also to its parameters, which indicates a need to develop effective heuristic solvers, which are more robust.

References

[1] D. G. Recupero, "Toward a Green Internet", *Science*, vol. 339, no. 6127, pp. 1533–1534, 2013 (doi: 10.1126/science.1235623).

[2] R. Bolla, R. Bruschi, F. Cucchietti, and F. Davoli, "Setting the Course for a Green Internet", *Science*, vol. 342, no. 6164, pp. 1316, 2013 (doi: 10.1126/science.342.6164.1316-a).

[3] R. Bolla, R. Bruschi, F. Davoli, and F. Cucchietti, "Energy efficiency in the Future Internet: A survey of existing approaches and trends in energy-aware fixed network infrastructures", *IEEE Commun. Surv. & Tutor.*, vol. 13, no. 2, pp. 223–244, 2011 (doi: 10.1109/SURV.2011.071410.00073).

[4] A. Karbowski and P. Jaskóła, "Two approaches to dynamic power management in energy-aware computer networks – methodological considerations", *Proc. Federated Conf. Comp. Sci. & Inform. Sys. FedCSIS 2015* (One of the Best Papers Award), Łódź, Poland, 2015, pp. 1177–1182 (doi: 10.15439/2015F228).

[5] P. Jaskóła, P. Arabas, and A. Karbowski, "Simultaneous routing and flow rate optimization in energy-aware computer networks", *Int. J. Appl. Mathem. & Comp. Sci.*, vol. 26, no. 1, pp. 231–243, 2016.

[6] E. Niewiadomska-Szynkiewicz, A. Sikora, P. Arabas, M. Kamola, K. Malinowski, P. Jaskóła, and M. Marks, "Network-wide power management in computer networks", in *Proc. 22nd Int. Teletraff. Congr. Special. Seminar on Energy Efficient & Green Netw. SSEEGN 2013* (Best Paper Award), Christchurch, New Zealand, 2013, pp. 25–30.

[7] E. Niewiadomska-Szynkiewicz, A. Sikora, P. Arabas, M. Kamola, M. Mincer, and J. Kołodziej, "Dynamic power management in energy-aware computer networks and data intensive computing systems", *Future Gener. Comp. Sys.*, vol. 37, pp. 284–296, 2014.

[8] F. Kelly, "Charging and rate control for elastic traffic", *Eur. Trans. Telecommun.*, vol. 8, no. 1, pp. 33–37, 1997 (doi: 10.1002/ett.4460080106).

[9] C. Touati, E. Altman, and J. Galtier, "On fairness in bandwidth allocation", *Tech. Rep. 4269*, Unité de recherche INRIA, Sophia Antipolis, France, 2001.

[10] J. Bisschop, *AIMMS Optimization Modeling.* Paragon Decision Technology B.V., 2007.

[11] A. P. Bianzino, L. Chiaraviglio, M. Mellia, and J. L. Rougier, "GRiDA: GReen Distributed Algorithm for energy-efficient IP backbone networks", *Comp. Netw.*, vol. 56, no. 14, pp. 3219–3232, 2012.

Andrzej Karbowski received his Ph.D. in 1990 and D.Sc. in 2012 in Automatic Control and Robotics from the Warsaw University of Technology, Faculty of Electronics and Information Technology. Currently he is an Associate Professor at the Institute of Control and Computation Engineering of the Warsaw University of Technology and at the Research and Academic Computer Network (NASK). He is the editor and the co-author of two books (on parallel and distributed computing), the author and the co-author of two e-books (on grid computing and the optimal control synthesis) and over 120 journal and conference papers. His research interests concentrate on data networks management, optimal control in risk conditions, water systems management, decomposition and parallel implementation of numerical algorithms.

E-mail: A.Karbowski@elka.pw.edu.pl
Institute of Control and Computation Engineering
Warsaw University of Technology
Nowowiejska st 15/19
00-665 Warsaw, Poland

Research and Academic Computer Network (NASK)
Wąwozowa st 18
02-796 Warsaw, Poland

Przemysław Jaskóła received his M.Sc. in Automatic Control and Robotics from the Warsaw University of Technology, Poland, in 1999. He works as a research associate at the Research and Academic Computer Network (NASK). His current research interests focus on parallel and distributed computing, modeling and multicriteria optimization of computer networks.

E-mail: pjaskola@nask.pl
Research and Academic Computer Network (NASK)
Wąwozowa st 18
02-796 Warsaw, Poland

Paper

Preconditioned Conjugate Gradient Method for Solution of Large Finite Element Problems on CPU and GPU

Sergiy Yu. Fialko and Filip Zeglen

Institute of Computer Science, Faculty of Physics, Mathematics and Computer Science,
Tadeusz Kościuszko Cracow University of Technology, Cracow, Poland

Abstract—In this article the preconditioned conjugate gradient (PCG) method, realized on GPU and intended to solution of large finite element problems of structural mechanics, is considered. The mathematical formulation of problem results in solution of linear equation sets with sparse symmetrical positive definite matrices. The authors use incomplete Cholesky factorization by value approach, based on technique of sparse matrices, for creation of efficient preconditioning, which ensures a stable convergence for weakly conditioned problems mentioned above. The research focuses on realization of PCG solver on GPU with using of CUBLAS and CUSPARSE libraries. Taking into account a restricted amount of GPU core memory, the efficiency and reliability of GPU PCG solver are checked and these factors are compared with data obtained with using of CPU version of this solver, working on large amount of RAM. The real-life large problems, taken from SCAD Soft collection, are considered for such a comparison.

Keywords—conjugate gradient, incomplete Cholesky factorization, iterative solver, NVIDIA CUDA, preconditioned conjugate gradient.

1. Introduction

The computational power of modern PC's becomes enough to solve medium scale complex engineering problems. Intensive development of desktop computers and gaming rigs markets made that for some aspects High Performance Computing (HPC) solutions are no longer necessary for a lot of problems. In the future, this trend will be expanded onto range of issues where ability of PC computers is sufficient for their solution of given problem scale. Improvements in the hardware realizations enhance development capabilities and demands to develop computational methods directly into a specific computer architecture. Processor units for execution of the fast instructions need in efficient memory management. Achievement of peak performance on logical thread must be preceded by elimination of empty cycles on physical core. The ways of efficient memory management for distributed memory architecture Non-Uniform Memory Access (NUMA) of today's HPC systems substantially distinguish from techniques used in Uniform Memory Access (UMA) of PC solutions. On the other hand, computational units of clusters in many cases have the same processor architecture at the level of hardware node as used in desktop solutions. Consequence of these distinctions is the necessity for creation of different algorithms implementing computational methods for desktop systems.

Solving systems of linear algebraic equations, arising from analysis of problem of solid and structural mechanics, by the preconditioned conjugate gradient on the Graphic Processing Unit (GPU) appear in many papers. In example, an article [1] presents the acceleration of matrix-vector product procedure with usage of ELLPACK, BELLPACK, SBELL formats instead of CSR in Compute Unified Device Architecture (CUDA) for packing of sparse symmetrical matrix for 2D elastic problem of solid mechanics. The block compressed sparse row (BCSR) format is applied for acceleration of sparse matrix-vector multiplication (SpMV) procedure in [2] for conjugate gradient (CG) method using CUDA. The application of graphic accelerators in finite element structural analysis is discussed in [3]. Article [4] proposes a level scheduling based on approximate minimum degree reordering algorithm for acceleration the triangular solution procedure.

In presented article, authors limit themselves to solving systems of linear algebraic equations with symmetrical sparse matrices by preconditioned conjugate gradient method. Such matrices arise when finite element method is applied to the problems of structural or solid mechanics. Scientific publications about parallel implementation of conjugate gradient method in architecture Symmetric Multiprocessing (SMP) can be found in [5]. This paper is mainly focused on maximal effective use of RAM memory. In cases when at application of sparse direct method the size of factorized stiffness matrix exceeds the capacity of RAM, it is necessary to use a secondary storage on disk. It leads to drastic increase of the computation time because solver produces lot of IO operations. The proposed iterative method runs in core memory and in the case of fast convergence could be considerably faster. Small number of iterations is achieved primarily using appropriate sparse matrix techniques for constructing of preconditioning based on Cholesky factorization by value method with application of secondary rejection of small entries [6]. Given approach as well as [7], [8] is intended for solution of complex engineering problems and produces all computations on CPU. The leading procedures – matrix-vector multiplication and forward-back substitutions relatively preconditioning – are poorly accelerated due to parallelization on shared-memory

computers when number of threads increases. Usually several right hand parts – load cases – appear in problems of structural mechanics. Therefore, in [5] each right hand part iterates on separate thread, and number of threads is restricted by number of right hand parts. The development of modern graphics cards is driven by the development of PCs. Today, in era of rapid general-purpose GPU development, the calculations are a separated branch and professional computing accelerators are not used as graphics cards although their architecture is made for that and allow it. HPC solutions also are equipped by accelerators, based on GPU.

This paper is devoted to development of preconditioned conjugate gradient method with incomplete Cholesky factorization by value preconditioning [5] using GPU, based on CUDA technology and intended to solution of linear algebraic equation sets with sparse symmetric positive definite matrices. Described implementation involves the use of a single device with general-purpose GPU support. This is a typical situation for PC with one external graphics card or workstation with one computing accelerator.

2. Preconditioned Conjugate Gradient Method

Let us consider the linear equation set

$$\mathbf{Kx} = \mathbf{b}, \tag{1}$$

where $\mathbf{K}$ is a symmetric positive definite sparse matrix arising when the finite element method is applied to problems of structural and solid mechanics. The problems of structural mechanics often are poorly conditioned due to using of thin-walled plates, shells, bars and large scattering of stiffness in structural elements, and the slow convergence of iterative methods occurs in such a situation [9].

Algorithm 1: Incomplete Cholesky factorization

1: $\mathbf{v}_{ip,i} = 0$
2: $ip = 0, 1, \ldots, np-1$
3: $i = 1 \in [1 \ldots N]$
4: **for** $j \in [1 \ldots N]$ **do**
5: $\quad \mathbf{v}_{0,i} = \mathbf{K}_{ij}, i \in L_j$
6: $\quad$ *Parallel* **for** $k \in List_j$ **do**
7: $\quad\quad \mathbf{v}_{ip,i} = \mathbf{v}_{ip,i} - \mathbf{H}_{i,k}\mathbf{H}_{j,k}, i \in L_k$
8: $\quad$ **end for**
9: $\quad$ **for** $ip \in [1 \ldots np-1]$ **do**
10: $\quad\quad \mathbf{v}_{0,i} += \mathbf{v}_{ip,i}, i \in L_j$
11: $\quad$ **end for**
12: $\quad$ **if** $\mathbf{v}_{0i}{}^2 < \psi \mathbf{H}_{ii}\mathbf{H}_{jj}$, $i \in L_j$ **then**
13: $\quad\quad \mathbf{H}_{ii} += \left|\mathbf{H}_{ij}\right| \sqrt{\frac{\mathbf{H}_{ii}}{\mathbf{H}_{jj}}}$, $\mathbf{H}_{jj} += \left|\mathbf{H}_{ij}\right| \sqrt{\frac{\mathbf{H}_{jj}}{\mathbf{H}_{ii}}}, \mathbf{v}_{0,i} = 0$
14: $\quad$ **else**
15: $\quad\quad \mathbf{L}_j \leftarrow \frac{\mathbf{v}_{0,i}}{\sqrt{\mathbf{H}_{jj}}}, List_i = j, \mathbf{v}_{0,i} = 0$
16: $\quad$ **end if**
17: **end for**

The proposed approach allows keeping a small value of rejection parameter ψ and ensures a stable and fast convergence of PCG method even for weakly conditioned problems of structural mechanics. The term "weakly conditioned" means that matrix $\mathbf{K}$ is not singular, but the conditioning number $cond(\mathbf{K})$ is relatively large, and conventional iterative methods demonstrates a slow convergence. The article [5] contains the detail consideration of proposed approach.

The preconditioned problem $\mathbf{B}^{-1}\mathbf{Kx} = \mathbf{B}^{-1}\mathbf{b}$ is solved instead Eq. (1), where $\mathbf{B} = \mathbf{HH}^T$ and $\mathbf{H}$ is the lower triangular matrix. The looking-left column-by-column incomplete factorization procedure is applied [5] as shown in Algorithm 1.

The *ip* and *np* are the thread number and the number of threads, respectively. Next, nonzero entries of current column j of matrix $\mathbf{K}$ are put to the dense vector $\mathbf{v}_0$ ($\mathbf{v}_{0,i} = \mathbf{K}_{ij}$). Expression $i \in L_j$ means that row number i belongs to nonzero structure of current column j. In the loop "parallel for k" columns k located at left from $j (k < j)$ produce the update of column j. Expression $k \in List_j$ means that only such columns k which have nonzero elements $\mathbf{H}_{jk}$ in factorized matrix $\mathbf{H}$ are taken. Each thread *ip* writes results in own vector $\mathbf{v}_{ip}$. Then we sum the results of each thread and obtain updated column j in vector $\mathbf{v}_0$ (loop for $ip = 1$, $np-1$). In the next step, each nonzero entry of $\mathbf{v}_0 (i \in L_j)$ is analyzed and the small entries $\mathbf{v}_{0i}{}^2 < \psi \mathbf{H}_{ii}\mathbf{H}_{jj}$, where $0 < \psi < 1$ (if $\mathbf{v}_{0i}{}^2 < \psi \mathbf{H}_{ii}\mathbf{H}_{jj}$) are rejected. Each rejection results in correction of diagonal entries $\mathbf{H}_{ii}$, $\mathbf{H}_{jj}$ to keep the positive definiteness of $\mathbf{H}$ and preconditioning matrix $\mathbf{B}$ [11]. Only the "large" entries are retained and put it in nonzero structure L_j of matrix $\mathbf{H}$.

Algorithm 2: PCG method

1: $k = 0$, $\mathbf{x}_0 = 0$
2: $\mathbf{r}_0 = \mathbf{b} - \mathbf{Kx}_0$
3: **while** $||r_k||_2 > tol$ **do**
4: $\quad$ Solve $\mathbf{Bz}_k = \mathbf{r}_k$
5: $\quad k = k + 1$
6: $\quad$ **if** $k = 1$ **then**
7: $\quad\quad \mathbf{p}_1 = \mathbf{z}_0$
8: $\quad$ **else**
9: $\quad\quad \beta_k = \frac{\mathbf{r}_{k-1}^T \mathbf{z}_{k-1}^T}{\mathbf{r}_{k-2}^T \mathbf{z}_{k-2}^T}$
10: $\quad\quad \mathbf{p}_k = \mathbf{z}_{k-1} + \beta_k \mathbf{p}_{k-1}$
11: $\quad$ **end if**
12: $\quad \alpha_k = \frac{\mathbf{r}_{k-1}^T \mathbf{z}_{k-1}^T}{\mathbf{p}_k^T \mathbf{K} \mathbf{p}_k^T}$
13: $\quad \mathbf{x}_k = \mathbf{x}_{k-1} + \alpha_k \mathbf{p}_k$
14: $\quad \mathbf{r}_k = \mathbf{r}_{k-1} - \alpha_k \mathbf{K} \mathbf{p}_k$
15: **end while**
16: $\mathbf{x} = \mathbf{x}_k$

Also, the number of current column j is put in $List_i$ of column i, located at right. In addition, after incomplete factoring is finished, secondary rejection of small entries $\mathbf{H}_{ij}{}^2 < \psi_1 \mathbf{H}_{ii}\mathbf{H}_{jj}$, where $0 < \psi < \psi_1 < 1$ is produced. It allows on reduction of nonzero entries in incomplete fac-

tor $\mathbf{H}$ and accelerates triangular solution procedure without significant deterioration of preconditioning properties.
The minimum degree ordering algorithm is applied before incomplete factorization for reducing the number of nonzero entries in factorized matrix. It improves the ability of preconditioning to accelerate a convergence [5].
The Algorithm 2 presents the PCG method.
The residual vector for problem given by Eq. (1) on iteration step k is: $\mathbf{r}_k = \mathbf{b} - \mathbf{K}\mathbf{x}_k$, where $\mathbf{x}_k$ is approximation of exact solution $\mathbf{x}$. For preconditioned problem the residual vector $\mathbf{z}_k$ is evaluated from expression $\mathbf{B}^{-1}\mathbf{r}_k = \mathbf{B}^{-1}(\mathbf{b} - \mathbf{K}\mathbf{x}_k) = \mathbf{z}_k$. Then the set of linear equations relatively preconditioning $\mathbf{B}\mathbf{z}_k = \mathbf{r}_k$, or $\mathbf{H}\mathbf{H}^T\mathbf{z}_k = \mathbf{r}_k$ is solved. The forward substitution $\mathbf{H}\mathbf{y}_k = \mathbf{r}_k \rightarrow \mathbf{y}_k$ and back substitution $\mathbf{H}^T\mathbf{z}_k = \mathbf{y}_k \rightarrow \mathbf{z}_k$ are produced.
The incomplete Cholesky factorization procedure requires a large amount of core memory and authors use the parallel Algorithm 1 implemented on CPU. The both CPU and GPU versions of PCG method exactly correspond to presented Algorithm 2.

3. Conjugate Gradient Method on GPU and Implementation Using CUDA

Preconditioned conjugate gradient method performs the set of linear algebra operations on matrices and vectors. All operations on GPU are produced only with application of procedures from CUBLAS [11] and CUSPARSE [12] libraries: *cusparseDcsrsv-solve()* [13] for triangular solution $\mathbf{H}\mathbf{y}_k = \mathbf{r}_k \rightarrow \mathbf{y}_k$ (forward substitution) and $\mathbf{H}^T\mathbf{z}_k = \mathbf{y}_k \rightarrow \mathbf{z}_k$ (back substitution), *cusparseDcsrmv()* [14] for matrix-vector multiplication $\mathbf{w}_k = \mathbf{K}\mathbf{p}_k$, *cublasDdot()* for evaluation of dot products ${\mathbf{r}_{k-1}}^T\mathbf{z}_{k-1}$, ${\mathbf{p}_k}^T\mathbf{w}_k$, *cublasDaxpy()* for computing of saxpy procedures $\mathbf{x}_k = \mathbf{x}_{k-1} + \alpha_k\mathbf{p}_k$, $\mathbf{r}_k = \mathbf{r}_{k-1} - \alpha_k\mathbf{w}_k$ and cublasDscal() for vector scalar multiplication $\mathbf{p}_k = \beta_k\mathbf{p}_{k-1}$.
The multiplication of sparse symmetric matrix by vector and triangular solutions during forward and back substitutions are the most complex procedures of PCG method. Their duration exceeds 90% of total solution time. The algorithm that performs any operations with sparse matrix must be consistent with the format in which this matrix is stored. CUSPARSE library supports following sparse matrix formats: COO, CSR, CSC, ELL, HYB, BSR, BSRX [15]. Procedures for Symmetric Positive Definite (SPD) matrices operate on matrices stored in Compressed Storage Row (CSR) format. This implementation of PCG method is practically the same as implementation of PCG method from CUDA library [16]. The main difference consists in the construction of preconditioning (see Algorithm 1). Utilization of compressed row format (CSR), which largely focused on low memory requirement, is perfect for the graphics card device, having the restricted memory amount. On the other hand, in the case of sparse matrix with specific structure CSR format greatly reduces possibility of blocking memory for CUDA thread blocks. The consequence of jumps in memory due to specific structure of sparse matrix leads to slowdown of instruction execution by pipelines in block of CUDA threads. Many publications concerning with implementation of sparse matrix-vector multiplication algorithm on the GPU [17] are devoted to achievement of high performance in operations on sparse positive definite matrices, stored in different formats and designed for the GPU. Thus, for GPU computing does no storage format for symmetric positive definite matrices exist, which would always give the most high-performance matrix-vector operations. Therefore, computing performance essentially depends on the structure of sparse matrix and its density.
The triangular solution procedure has a highly sequential nature – its parallelization does not result in considerable acceleration of computations on SMP computers as well as on GPU. Algorithm of triangular solution used in CUSPARSE is presented in [18]. Algorithm 3 contains the pseudocode of version on GPU. The CPU version is presented in [5].

Algorithm 3: Pseudocode of GPU version

1: Aggregate sparse stiffness matrix $\mathbf{K}$ and prepare lower triangular matrix $\mathbf{H}$ using Algorithm 1 (on CPU). Initiate CUDA device
2: Allocate memory on graphic accelerator (device memory) for matrices $\mathbf{K}$, $\mathbf{H}$, packed in CSR format and copy these matrices from host memory to device memory
3: Use *cusparseDcsrsm-analysis()* procedure twice for analysis of structure of the $\mathbf{H}$ and $\mathbf{H}^T$ matrices
4: Allocate device memory for vectors $\mathbf{x}$, $\mathbf{p}$, $\mathbf{r}$, $\mathbf{z}$ and working vector $\mathbf{w}$
5: Run Algorithm 2 until convergence will not be achieved (no transfers of data between host and device occurs)
6: Copy converged vector $\mathbf{x}$ from device to host
7: Deallocate device memory and deinitialize CUDA device

The version of solver on CPU uses in-home algorithms. Authors found in [19] that procedure *mkl-dcsrsymv* (sparse symmetric matrix-vector multiplication) taken from Intel MKL 11.2 accelerates the sparse symmetric matrix - vector multiplication about 2 times in compare with in-home procedure on single thread and about 3 times on multiple threads. In addition, the *mkl-dcsrtrsv* (triangular solution for sparse matrix) procedure from Intel MKL demonstrates the same time on the both: single thread and multiple threads, and is on 20% slower in compare with in-home triangular solution procedure. For proposed class of problems the density of lower triangular matrix $\mathbf{H}$ is in many times more than density of stiffness matrix $\mathbf{K}$, and the duration of triangular solution procedure is in several times longer than the procedure of matrix-vector multiplication (Tables 1, 4, and 7). Therefore, acceleration of matrix-vector multiplication procedure does not produce considerable impact on performance of PCG solver and allows us to use in-home procedures on CPU version.

The authors used Microsoft Visual Studio 2013 IDE and NVIDIA GPU Computing Toolkit v.6.5 with CUBLAS and CUSPARSE libraries. The C++ compiler v. 120 with flags /O2 in realize version and /arch:AVX (Advanced Vector Extension) was applied. Also, the unrolling of loop eight times in the both: in-home matrix-vector multiplication and in triangular solution procedures was used.

4. Test Problems and Hardware Configuration

The three design models with fully different properties of stiffness matrices are considered. Two models of multi-storey buildings with quite different construction schemes and the model of shopping center are analyzed. All these real-life design models are taken from SCAD Soft [20] collection. The research attention is focused on total computation time, on computation time of matrix-vector multiplication procedure and on computation time of triangular solution procedure at the stage of resolution respectively preconditioning (SpTr). These intervals of time encompass all iterations required for achievement of convergence.

Tests were made on following hardware configuration: IBM System x iDataPlex dx360 M4 Server running Windows 7 Ultimate 64 bit, CUDA Toolkit 6.5 with CPU – Intel Xeon E5-2620 2.0 GHz 6C 12T (2.5 GHz Turbo) 6x256 kB L2, 15 MB L3, 32 GB DDR3 (8x4 GB) PC3-10700U (1333 MHz) and GPU – Nvidia Tesla K20m 2496 CUDA cores 705 MHz, 5 GB GDDR5 5200 MHz (1300 MHz).

The following designations are used. Lx is calculation time of forward and backward substitutions for all iterations, Kv – calculation time of procedure SpMV for all iterations, Oth stands for duration of other computing included in PCG iteration, Total – total calculation time for all produced iterations, Itr is the number of iterations required to achieve of convergence, DenK and DenL – densities of matrices **K** and **H** correspondingly (number of nonzero entries/total number of elements in lower triangular part of matrix) expressed as a percentage, ψ – value of rejection parameter $\psi\left(\psi \in \left[10^{-9}, 10^{-20}\right]\right)$, ψ_1 – value of post-rejection parameter.

4.1. Problem 1

Aquamarine is a finite element model of multistory building (Fig. 1) and comprises 176,819 finite elements, equations 881,908 and 149,494 nodes. Authors present the computation times on the both: CPU and GPU with the fixed value of rejection parameter $\psi = 10^{-10}$, and different values of post-rejection parameter ψ_1 (Tables 1 and 2). Table 3 shows comparison of computation times on CPU and GPU.

Higher GPU performance arises only on relatively large values of dropping parameters (Fig. 2). In this case structure of the matrix **H** is more sparser than when using smaller value of dropping parameter ψ_1 (Table 3). In all other cases

Fig. 1. Computational model of Aquamarine.

Table 1
Problem 1 – computation times on CPU at different ψ, ψ_1 parameters

ψ	ψ_1	Kv [s]	Lx [s]	Oth [s]	Total [s]	Itr
10^{-10}	10^{-8}	3.5	44.4	1.1	49	142
10^{-10}	10^{-6}	4.3	31.9	1.8	38	179
10^{-10}	10^{-4}	23.9	102.1	10	136	968
10^{-10}	10^{-3}	169.8	543.7	92.5	806	6550

Table 2
Problem 1 – computation times on GPU at different ψ, ψ_1 parameters

ψ	ψ_1	Kv [s]	Lx [s]	Oth [s]	Total [s]	Itr
10^{-10}	10^{-8}	7.4	309	13.6	330	142
10^{-10}	10^{-6}	9.3	112	9.7	131	179
10^{-10}	10^{-4}	49.9	100	15.1	165	968
10^{-10}	10^{-3}	334.8	185.8	118.4	639	6550

Table 3
Problem 1 – comparison of computation time on CPU and GPU ($\psi = 10^{-10}$, DenK = 0.0033)

ψ_1	CPU [s]	GPU [s]	DenL
10^{-8}	49	330	0.0189
10^{-6}	38	131	0.0099
10^{-4}	136	165	0.0046
10^{-3}	806	639	0.0029

in which the parameter $\psi <= 10^{-4}$, density of the matrix **H** increases, and a considerable advantage of the CPU version (Table 1) over the GPU (Table 2) can be observed.
The matrix-vector multiplication procedure using CPU is less than GPU version for all considered values of dropping parameters (Tables 1 and 2). Duration of matrix-vector multiplication procedure depends on density of stiffness matrix **K**, which strictly depends on considered problem and does not depend on values of dropping parameters. In contrast to matrix-vector multiplication procedure, the duration of triangular solution procedure depends on values of ψ and ψ_1 parameters (Table 3).

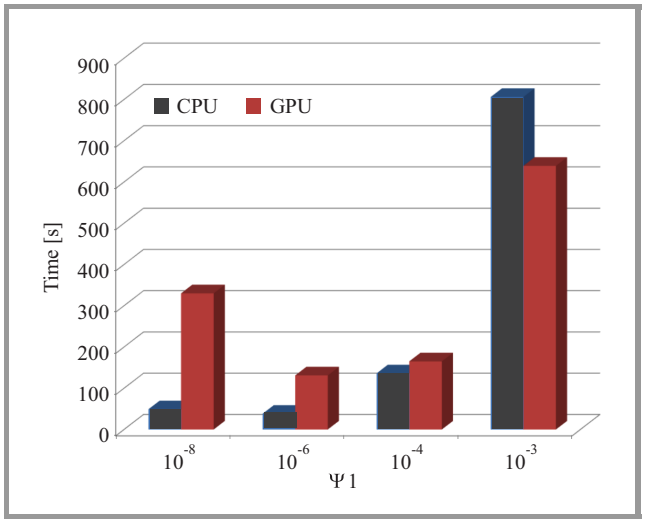


Fig. 2. Problem 1 – comparison of computation times on CPU and GPU for Aquamarine.

The performance of triangular solution procedure by CUDA [13], [17] considerably depends on matrix **H** sparsity. If **H** density is very small, the CUDA realization of triangular solution is faster than CPU version. With decreasing of drop parameter value the density of **H** increases and CPU realization of triangular solution becomes faster (Table 3). For drop parameters values, ensuring acceptably fast solution, CPU realization is more faster than GPU (Fig. 2).

4.2. Problem 2

Schemanew is a finite element model of multistory building (Fig. 3) and comprises 556,905 finite elements, equations 3,198,609 and 534,490 nodes. In this Subsection the CPU and GPU times for different values of rejection parameters ψ, ψ_1 (Tables 4 and 5) and their comparison (Table 6) are presented.
When applying the small post-dropping parameters $\psi_1 <= 10^{-6}$ which satisfy a fast convergence, the size of preconditioning matrix **H** exceeds capacity of GPU memory of Tesla equipped with 5.4 GB. In such cases in Fig. 4 and Tables 5 and 6 only the results obtained on CPU are depicted. In this Subsection authors show on plots two rejec-

Fig. 3. Computational model of Schemanew.

Table 4
Problem 2 – computation times on CPU at different ψ, ψ_1 parameters

ψ	ψ_1	Kv [s]	Lx [s]	Oth [s]	Total [s]	Itr
10^{-11}	10^{-8}	15.5	127.4	8.1	151	196
10^{-9}	10^{-6}	45.6	289.8	23.6	359	497
10^{-9}	10^{-5}	59.8	303	30.2	393	658
10^{-9}	10^{-4}	142	566	73	781	1584

Table 5
Problem 2 – computation times on GPU at different ψ, ψ_1 parameters

ψ	ψ_1	Kv [s]	Lx [s]	Oth [s]	Total [s]	Itr
10^{-11}	10^{-8}	–	–	–	–	–
10^{-9}	10^{-6}	–	–	–	–	–
10^{-9}	10^{-5}	104	139	19	262	659
10^{-9}	10^{-4}	248	148	33	429	1583

Table 6
Problem 2 – comparison of computation time on CPU and GPU (DenK = 0.000667)

ψ	ψ_1	CPU [s]	GPU [s]	DenL
10^{-11}	10^{-8}	151	–	0.00347
10^{-9}	10^{-6}	359	–	0.00196
10^{-9}	10^{-5}	393	262	0.00144
10^{-9}	10^{-4}	781	429	0.00099

tion parameters ψ and ψ_1, separated by semicolon. Higher performance of GPU realization (Table 5) of PCG solver comparing with CPU version (Table 4) is achieved only for respectively large values of drop parameters, which leads to increasing of number of iterations and slowdown of convergence. The acceptable solution time is achieved on CPU version. GPU version for proper values of drop parameter, which ensure the stable and fast convergence, requires essentially more amount of memory than given graphic accelerator possesses.

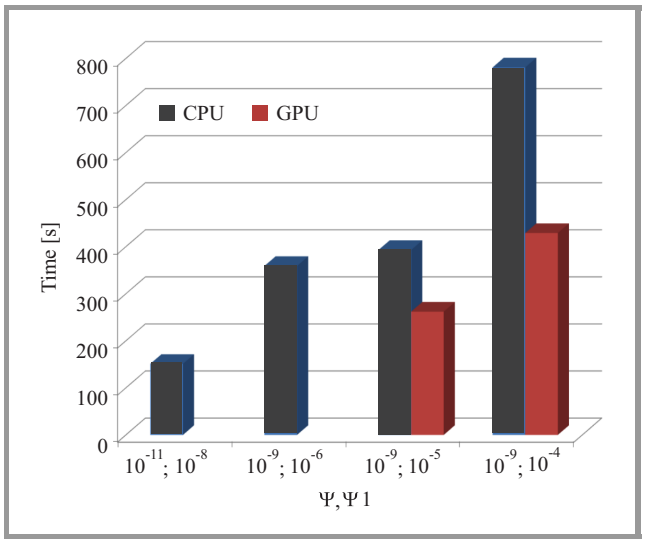


Fig. 4. Problem 2 – comparison of computation times on CPU and GPU for Schemanew.

For $\psi_1 > 10^{-5}$ is possible to put matrix **H** in memory of graphic accelerator, and GPU version is faster than CPU. However, with decreasing of ψ_1 density of matrix **H** increases, and advantage of GPU version becomes smaller (Fig. 4). This tendency suggests, that even if memory of the graphic accelerator would suffice for accommodation of matrix **H** at smaller values ψ_1, with decreasing of ψ_1 CPU becomes faster, than the version on GPU. Probably, it is caused by specifics of triangular solution algorithm [18] (Table 5), developed by NVIDIA, which is very fast for sparse matrices of low density, but with increase of density its performance considerably deteriorates.

4.3. Problem 3

TRK is a finite element model of market building (Fig. 5) and comprises 473,723 finite elements, equations 2,442,846 and 441,300 nodes.

For parameters $\psi = 10^{-8}$ and $\psi_1 = 10^{-7}$ matrix is to large and does not fit into GPU accelerator memory.

The matrix-vector multiplication procedure as well as in the previous problems, on CPU is about two times faster than on GPU (Tables 7 and 8).

The best performance of triangular solution procedure, which mostly affects total time, is achieved on GPU. In each case of preconditioning parameters, authors found considerable advantage of GPU version comparing with CPU (Table 9).

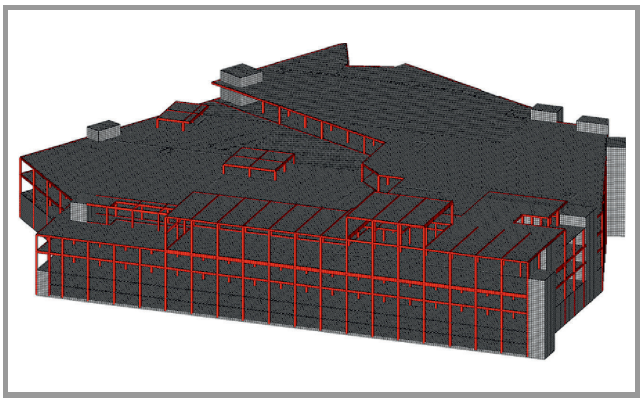


Fig. 5. Computational model of TRK.

Table 7
Problem 3 – computation times on CPU
at different ψ, ψ_1 parameters

ψ	ψ_1	Kv [s]	Lx [s]	Oth [s]	Total [s]	Itr
10^{-10}	10^{-5}	16.1	90.5	8.4	115	231
10^{-8}	10^{-7}	10.4	89.5	6.1	106	177
10^{-8}	10^{-6}	11.2	79.8	7	98	191
10^{-8}	10^{-5}	15	85.8	9.2	110	257
10^{-8}	10^{-4}	41.9	188.9	23.2	254	707

Table 8
Problem 3 – computation times on GPU
at different ψ, ψ_1 parameters

ψ	ψ_1	Kv [s]	Lx [s]	Oth [s]	Total [s]	Itr
10^{-10}	10^{-5}	26.7	36.1	7.2	70	231
10^{-8}	10^{-7}	–	–	–	–	–
10^{-8}	10^{-6}	22.2	41.5	6.3	70	191
10^{-8}	10^{-5}	29.5	28.4	8.1	66	256
10^{-8}	10^{-4}	81.4	30.6	25	137	705

Table 9
Problem 3 – comparison of computation times on CPU
and GPU (DenK = 0.00096)

ψ	ψ_1	CPU [s]	GPU [s]	DenL
10^{-10}	10^{-5}	115	70	0.00233
10^{-8}	10^{-7}	106	–	0.00384
10^{-8}	10^{-6}	98	70	0.00305
10^{-8}	10^{-5}	110	66	0.00224
10^{-8}	10^{-4}	254	137	0.00155

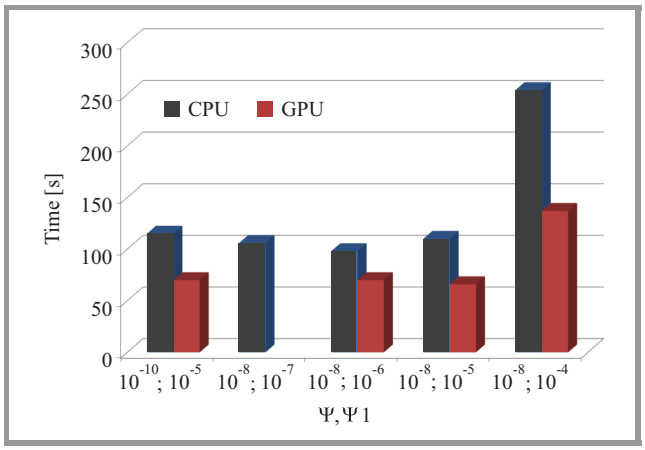


Fig. 6. Problem 3 – comparison of computation times on CPU and GPU for TRK.

5. Summary and Conclusions

For presented problems of structural mechanics, which we intend to solve by PCG method, the conducted research cannot clearly indicate which one of two comparable devices: CPU or GPU, will demonstrate a better performance. Which device is better, depends on the densities and structures of the matrices **K** and **H**. GPU demonstrates a better performance only for very sparse matrices, the location of nonzero elements of which allows on efficient split on parallel tasks during triangular solution, performed by algorithm [18]. For problem 1, where matrix **H** has highest density among all considered problems, one can observe significant CPU advantage at stage of triangular solution, which has a main impact on total solution time. For problems 2 and 3 matrix **H** is a more sparser, than in problem 1, and GPU version demonstrates a less time of triangular solution algorithm and correspondingly better solution time than CPU, until the amount of graphic accelerator memory is enough large to put matrices **H** and **K**. To improve ability of preconditioning and accelerate convergence, the values of drop parameters ψ, ψ_1 were decreased and density of matrix **H** was increased. When GPU version fails due to insufficient device memory, the CPU version of solver having sufficient amount of RAM, solves these problem faster than GPU version. Only for problem 3 authors found that GPU version occurs faster than CPU. Matrix-vector multiplication procedure is always faster in CPU version, regardless of density of matrix **K**. Decrease of drop parameter value leads to improving of preconditioning properties, reducing the number of iterations for achievement of convergence, but increasing the duration of each iteration due to considerable enlarging of triangular solution time.

Acknowledgements

The authors express their deep gratitude to SCAD Soft IT company for providing a collection of real-life problems from engineering practice.

References

[1] J. Zhang and L. Zhang, "Efficient CUDA polynomial preconditioned conjugate gradient solver for finite element computation of elasticity problems", *Mathem. Problems in Engin.*, article ID 398438, pp. 1–12, 2013.

[2] M. Verschoor and A. C. Jalba, "Analysis and performance estimation of the Conjugate Gradient method on multiple GPUs", *Parallel Comput.*, vol. 38, no. 10–11, pp. 552–575, 2012.

[3] S. Georgescu, P. Chow, and H. Okuda, "GPU Acceleration for FEM-Based Structural Analysis", *Arch. Comput. Methods Engin.*, vol. 20, no. 2, pp. 111–121, 2013.

[4] C. Yao, Z. Yonghua, Z. Wei, Z. Lian, "GPU-accelerated incomplete Cholesky factorization preconditioned conjugate gradient method", *J. of Comp. Res. & Develop.*, vol. 52, no. 4, pp. 843–850, 2015.

[5] S. Y. Fialko, "Iterative methods for solving large-scale problems of structural mechanics using multi-core computers", *Archiv. of Civil and Mechan. Engin.*, vol. 14, no. 1, pp. 190–203, 2014.

[6] M. Suarjana and K. H. Law, "A robust incomplete factorization based on value and space constraints", *Int. J. for Numerical Methods in Engin.*, vol. 38, pp. 1703–1719, 1995.

[7] K. Malkowski, I. Lee, P. Raghavan, and M. J. Irwin, "Conjugate gradient sparse solver: Performance-power characteristics", in *Proc. 20th IEEE Int. Parallel & Distrib. Process. Symp. IPDPS 2006*, Rhodes Island, Greece, 2006.

[8] M. Papadrakakis, "Solving Large-Scale Problems in Mechanics". *Wiley*, 1993.

[9] S. Y. Fialko, "Parallel direct solver for solving systems of linear equations resulting from finite element method on multi-core desktops and workstations", *Comp. & Mathem. with Appl.*, vol. 70, pp. 2968–2987, 2015.

[10] A. Jennings, "Development of an ICCG algorithm for large sparse systems", in *Preconditioned Methods. Theory and Applications*, D. J. Evans, Ed. Gordon and Breach Publishers, 1983, pp. 425–438.

[11] Nvidia Corporation, "cuBLAS API" [Online]. Available: http://docs.nvidia.com/cuda/cublas/

[12] Nvidia Corporation, "cuSPARSE API" [Online]. Available: http://docs.nvidia.com/cuda/cusparse/

[13] Nvidia Corporation, "csrsvsolve()" [Online]. Available: http://docs.nvidia.com/cuda/cusparse/#cusparse-lt-t-gt-csrsvsolve

[14] Nvidia Corporation, "csrmv()" [Online]. Available: http://docs.nvidia.com/cuda/cusparse/#cusparse-lt-t-gt-csrmv

[15] Nvidia Corporation, "cuSPARSE Indexing and Data Formats" [Online]. Available: http://docs.nvidia.com/cuda/cusparse/#cusparse-indexing-and-data-formats

[16] Nvidia Corporation, "CUDA Library – conjugateGradientPrecond – Preconditioned Conjugate Gradient" [Online]. Available: http://docs.nvidia.com/cuda/cuda-samples/#preconditioned-conjugate-gradient/

[17] B.-Y. Su and K. Keutzer, "clSpMV: A Cross-Platform OpenCL SpMV Framework on GPUs" [Online]. Available: http://parlab.eecs.berkeley.edu/sites/all/parlab/files/clspMV-%20Keutzer.pdf

[18] M. Naumov, "Parallel Solution of Sparse Triangular Linear Systems in the Preconditioned Iterative Methods on the GPU" [Online]. Available: https://research.nvidia.com/publication/parallel-solution-sparse-triangular-linear-systems-preconditioned-iterative-methods-gpu

[19] S. Fialko and F. Zeglen, " Block preconditioned conjugate gradient method for extraction of natural vibration frequencies in structural analysis", in *Proc. Feder. Conf. Comp. Sci. & Inform. Syst. FedCSIS 2015*, Łódź, Poland, 2015, vol. 3, pp. 655–66.

[20] V. S. Karpilovskii, E. Z. Kriksunov, A. A. Malyarenko, M. A. Mikitarenko, A. V. Perelmuter, and M. A. Perelmuter, "SCAD computational complex", *ASV*, 2004 (in Russian).

Sergiy Yu. Fialko studied structural mechanics and obtained his degree in 1983 in Institute of Mechanics in Kiev, Ukraine. He worked in Research Institute of Steel Constructions in Kiev, in software company RoboBAT in Cracow, Poland, in National University of Construction and Architecture in Kiev, where obtained his habilitation degree in 2004. Since 2007, he has been working in the Cracow University of Technology. For many years he is the scientific adviser of software company SCAD Soft.
E-mail: sergiy.fialko@gmail.com
Institute of Computer Science
Faculty of Physics, Mathematics and Computer Science
Tadeusz Kościuszko Cracow University of Technology
Warszawska st 24
31-155 Cracow, Poland

Filip Zeglen obtained his B.Sc. graduation in 2012 in Computer Science at Wyższa Szkoła Ekonomii i Informatyki in Cracow and M.Sc. graduation in 2014 at Cracow University of Technology. He worked as .NET, Java developer and freelancing indie games developer. Since 2014 he is Research and Teaching Assistant at Cracow University of Technology and Ph.D. student at Institute of Fundamental Technological Research Polish Academy of Sciences.
E-mail: filipzeglen@hotmail.com
Institute of Computer Science
Faculty of Physics, Mathematics and Computer Science
Tadeusz Kościuszko Cracow University of Technology
Warszawska st 24
31-155 Cracow, Poland

Paper

A Novel GPU-Enabled Simulator for Large Scale Spiking Neural Networks

Paweł Szynkiewicz

System Research Institute, Polish Academy of Science, Warsaw, Poland

Abstract—The understanding of the structural and dynamic complexity of neural networks is greatly facilitated by computer simulations. An ongoing challenge for simulating realistic models is, however, computational speed. In this paper a framework for modeling and parallel simulation of biological-inspired large scale spiking neural networks on high-performance graphics processors is described. This tool is implemented in the OpenCL programming technology. It enables simulation study with three models: Integrate-and-fire, Hodgkin-Huxley and Izhikevich neuron model. The results of extensive simulations are provided to illustrate the operation and performance of the presented software framework. The particular attention is focused on the computational speed-up factor.

Keywords—GPU computing, OpenCL programming technology, parallel simulation, spiking neural networks.

1. Introduction

Simulation of biological-inspired Spiking Neural Networks (SNN) is generally a complex problem that involves cumbersome calculations, especially when processing of large scale networks. The restrictions are caused by demands on computer resources, i.e. processor and memory. As biological neural networks become larger and more complex, the required computational power grows significantly. However, the calculations performed by neural networks simulators can be easily partitioned into large number of independent parts and carried out on many cores or computers. It was observed that parallel implementation based on MapReduce programming model improves the efficiency of the simulator and speeds up a calculation process.

A low-cost, an alternative to supercomputers is the Graphical Procession Unit (GPU) – specialized massively-parallel graphics processor that can be used as a general purpose computational accelerator [1]. GPU-enabled parallel computing is a relatively new area of research that has become extremely popular over the last decade and is rapidly increasing its advance into different areas of technology. Last years a model for parallel computing based on the use of GPUs to perform a general purpose scientific and engineering computing was developed and used to solve complex scientific and engineering problems. Using Compute Unified Device Architecture (CUDA) or Open Computing Language (OpenCL) many real-world applications can be easily implemented and run significantly faster than on multi-processor or multi-core systems. GPU clusters are one of the most progressive branches in a field of parallel computing and data processing nowadays.

The paper addresses issues associated with parallel computing systems and the application of GPU technology to large scale systems simulation. During research a dedicated software framework have been developed and designed that can be used to extensive simulation of spiking neural networks on GPU accelerators. This framework has been implemented in the OpenCL programming technology and can be executed on various computing platforms. The relative benefits and limitations of presented software platform have been evaluated based on results of numerical experiments performed for various less and more complex models of neural networks.

The remainder of this paper is organized as follows. A brief survey of biological-inspired SNN models is presented in Section 2. The overview of parallel SNN simulators is provided in Section 3. The organization, implementation and usage of the software framework for SNN simulation on GPU is described in Section 4. The results of simulations of several complex networks on various hardware platforms are presented and discussed in Section 5. Finally, conclusions are drawn in Section 6.

2. Spiking Neural Networks Modeling

A spiking neural network is composed of a set of N spiking neurons and E synapses – links $< i, j > \in E$, $E \subseteq N \times N$ with weights $w_{ij} \geq 0$, [2]–[5]. Excitatory and inhibitory synapses are distinguished. Excitatory synapses are connections of all excitatory neurons, while inhibitory synapses are connections of all inhibitory neurons. A spike is produced when a condition on the state variables is satisfied, for example when the membrane potential exceeds a threshold value (see Fig. 1). In general, various linear or nonlinear threshold functions $V_{th} : \Re^+ \rightarrow \Re^+$ have been defined for various models of spiking neuron. Thus, a biological-inspired spiking neuron can be described as a hybrid system with one or several continuous state variables (membrane potential V, conductances C, etc.), and spikes received through the synapses that trigger changes in some of the variables. Continuous evolution of a number of state variables is usually modeled by a set of differential equa-

tions with discrete events. The hybrid system formalism is presented in [6].

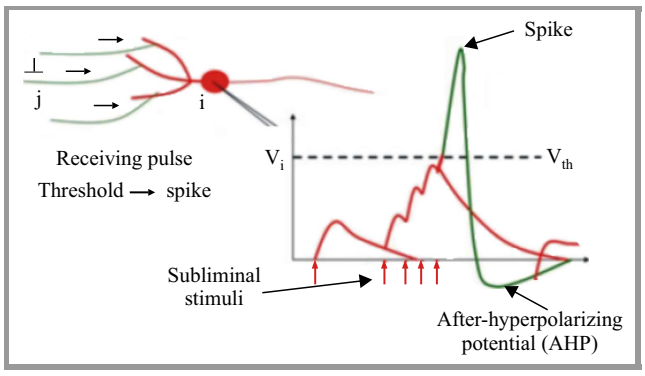


Fig. 1. A model of spiking neuron.

The range of computational problems related to spiking neurons is very large, especially in case when detailed biophysical representations of the neurons have to be used. The example is the reproduction of intracellular electrophysiological measurements. In general, modeling of biologically realistic spiking neural networks requires tuning the enormous number of parameters [7], [8]. In other cases, one does not need to realistically capture the spike generating mechanisms, and simpler models are sufficient.
A survey of models of a spiking neuron is provided in the literature [3], [4], [6], [9]. The research attention is focused on three commonly used models, simple Integrate-and-fire and models developed by Hodgkin and Huxley and Izhikevich.

2.1. Integrate-and-fire Neuron Model

Integrate-and-Fire (I&F) is the simplest spiking neural model described in [3], [6]. Let us refer to V as the neuron's membrane potential (the system state) and to I as the input current. Assuming that I is a sum of excitatory impulses I_E, inhibitory impulses I_I and constant current offset I_{offset} the dynamics of a neural model is described by the following state equations:

$$\frac{dV}{dt} = \frac{1}{\tau}(V_{rest} - V) + \frac{1}{C}\left(I_E + I_I + I_{offset}\right), \tag{1}$$

$$\frac{dI_E}{dt} = -\frac{I_E}{\tau_{synE}}, \tag{2}$$

$$\frac{dI_I}{dt} = -\frac{I_I}{\tau_{synI}}, \tag{3}$$

where τ denotes a model time constant, τ_{synE} and τ_{synI} excitatory and inhibitory synapses time constants, C is membrane capacity, V_{rest} is initial potential of the membrane, V_{reset} is membrane potential after spike (reset potential) and τ_{refrac} stands for relaxation time (time after spike during which neuron is insensitive to stimulation). In I&F model all spikes are generated when $V \geq V_{th}$. Potential after spike is reset to $V \leftarrow V_{reset}$.

2.2. Hodgkin-Huxley Neuron Model

Integrate-and-fire is a simple model that can imitate some of the biological neuron behavior. However, it is unable to reproduce firing patterns like: bursting, chattering, etc. The model developed by Hodgkin and Huxley (H&H) and described in [10] is one of the most successful mathematical model of a complex biological process that has ever been formulated. The idea is that the semipermeable cell membrane separates the interior of the cell from the extracellular liquid and acts as a capacitor. If an input current I is injected into the cell, it may add further charge on the capacitor, or leak through the channels in the cell membrane. In the standard H&H model there are three types of channels: a sodium channel Na, a potassium channel K and an unspecific leakage channel l with resistance R. Assuming that an input current I is a sum of excitatory impulses I_E, inhibitory impulses I_I, constant current offset I_{offset} and externally injected current I_{inj} the model is yield by the state equations:

$$\begin{aligned}\frac{dV}{dt} &= \frac{1}{C}\left[-g_{Na}m^3h(V-E_{Na}) - \right. \\ &- g_K n^4(V-E_K) - g_l(V-E_l) - g_e(V-E_e) - \\ &- \left. g_i(V-E_i) + I_{offset} + I_{inj}\right], \end{aligned} \tag{4}$$

$$\frac{dm}{dt} = \alpha_m(V)(1-m) - \beta_m(V)m, \tag{5}$$

$$\frac{dn}{dt} = \alpha_n(V)(1-n) - \beta_n(V)n, \tag{6}$$

$$\frac{dh}{dt} = \alpha_h(V)(1-h) - \beta_h(V)h, \tag{7}$$

$$\frac{dg_e}{dt} = -\frac{g_e}{\tau_{synE}}, \tag{8}$$

$$\frac{dg_i}{dt} = -\frac{g_i}{\tau_{synI}}, \tag{9}$$

where g_e and g_i denote excitatory and inhibitory synapses conductivity, g_{Na}, g_K, g_l ion channels conductance, E_{Na}, E_K, E_l, E_e, E_i ion channels reverse potentials. The definitions of functions α_m, β_m, α_n, β_n, α_h, β_h are provided in [3]. In order to produce an action potential the membrane potential must be increased quickly enough to cross threshold ($dV/dt \geq V_{th}$).

2.3. Izhikevich Neuron Model

H&H model is computationally expensive and in case of huge networks requires many differential equations solutions. Another kind of formalism which is able to replicate different rich firing patterns achievable with H&H model, using two simple equations with only one super-linear term was proposed by Izhikevich in [11]. Izhikevich model (I) has a computational efficiency similar to I&F model.
Izhikevich reduced biophysically accurate H&H neuronal model to a two-dimensional system of ordinary differential equations of the form

$$\frac{dV}{dt} = 0.04V^2 + 5V + 140 - U + I, \tag{10}$$

$$\frac{dU}{dt} = a(bV - U), \tag{11}$$

$$\frac{dI_E}{dt} = -\frac{I_E}{\tau_{syn}}, \tag{12}$$

$$\frac{dI_I}{dt} = -\frac{I_I}{\tau_{syn}}, \tag{13}$$

$$\text{if} \quad V \geq 30 \text{ mV}, \quad \text{then} \quad V \leftarrow c, \quad U \leftarrow U + d, \tag{14}$$

where I_E and I_I denote presynaptic currents from excitatory synapses and inhibitory synapses respectively, τ_{syn} synapse time constant (for excitatory and inhibitory $\tau_{syn} = 1$ ms), a, b, c, d model parameters, C membrane capacity.
After the spike reaches its apex (+30 mV), the membrane voltage and the recovery variable are reset according to the Eq. (14). The Eq. (10) was obtained by fitting the spike initiation dynamics of a cortical neuron (other choices also feasible) so that the membrane potential V has mV scale and the time t has ms scale. The resting potential in the model is between −70 and −60 mV depending on the value of b.

3. Simulation of Spiking Neural Networks

3.1. Parallel Simulation of SNN

The simulation of spiking neural networks can be naturally decomposed into three main phases:

- integrating the differential equations that describe the neuron models,
- propagating the spikes to target neurons,
- changing states of target neurons.

It is obvious that the bottleneck for large scale networks simulation is the propagation of numerous spikes across the network considered. Recent research has shown that modern simulators of spiking neural networks can be parallelized and executed on both multi-core CPUs and GPUs regardless of the network topology [12]–[15]. Parallel computation can be applied to all listed phases of the network simulation. The parallelization of the first phase, i.e. numerical integration is straightforward. It follows the Single Instruction, Multiple Data (SIMD) paradigm. The number of operations scale with the number of neurons in a network. The total computational cost for large scale networks is dominated by the second phase in which the operations scales with the number of synapses.
Parallel implementations of SNN are reviewed and discussed in [6]. Three parallelization strategies are proposed and discussed:

- N-parallel – spike propagation is parallelized over neurons. Each thread updates the total input of one neuron;
- S-parallel – spike propagation is parallelized over synaptic evens. Each thread implements the effect of a spike arriving at one synapse. The number of executed threads is limited by the total number of synapses in the network executed at each timestep;
- NS-parallel – combination of both aforementioned strategies N-parallel and S-parallel. This approach is recommended for GPU computing.

The software environments for neural networks simulation can implement two simulation modes:

- time-driven – all neurons are updated simultaneously at each timestep (tick of a global clock) – synchronous distributed simulation,
- event-driven – neurons are updated only when the event occur, i.e. they receive or emit a spike – asynchronous parallel simulation.

Time-driven and event-driven algorithms for SNN simulation are described in [6].

3.2. Survey of SNN Simulators

SNN Simulating on CPU. A survey of software environments for spiking neural networks simulation on CPUs is presented in [6], [14]. NEURON [16] is a commonly used, robust and efficient software platform that can support creation and evaluation of various models of biological neurons and neural circuits. It implements both time-driven and event-driven simulation modes. Moreover, NEURON supports parallel processing on multicore and multiprocessor machines employing threads and distributed processing in clusters using MPI standard. It is available on Unix, Linux and MS Windows platforms. It was executed on Cray and IBM Blue Gene supercomputers.
Neural Simulation Tool (NEST) [17] was created as a result of a long term collaborative project to support the development of technology for neural networks simulation. It is designed to large scale neural systems with heterogenity in neuron and synapses types simulation. It supports parallelization by multi-threading and message passing, and can be executed on multiprocessor machine and in a cluster of computers. NEST implements time-driven simulation mode, and is available on Unix, Linux, MS-Windows and Mac OS platforms. The software is provided to the scientific community under an open source license.
Brian [18], [19] is widely used, highly flexible and easily extensible simulator for spiking neural networks available on almost all platforms (Linux, MS Windows, Mac OS). It provides the implementations of I&F and H&H neuron models, and can be easily extended with the others. This software platform is written in the Python programming language. It is easy to learn and use. Various libraries of methods written in the Python can be used, e.g. NumPy and SciPy for numerical calculations, PyLab for results graphical visualization. Parallel Python can be employed to calculation parallelization. The sources, demos, manual

and publications can be downloaded from the project Web page [19]. It is released under the CeCILL license.

Mvaspike [20] is a general purpose tool for modeling and simulating large and complex biological neural networks. It is based on the event-based modeling and simulation strategy. The focus is on spiking neural networks simulation (integrate-and-fire and other spiking point neurons). A good balance between simulation efficiency and modeling freedom is provided. The core of the system is implemented in C++, however, the access from other programming languages is easy. A parallel implementation is available for multiprocessor machines and clusters.

SNN simulating on GPU. Several software environments for SNN simulation on GPU are described in literature.

NeMo [21], [22] is a high-performance environment for large scale spiking neural network simulation. It simulates systems of Integrate-and-fire and Izhikevich neurons on CUDA-enabled GPUs and uses a powerful scatter-gather messaging algorithm to provide further optimization for sparse random connectivities and supports real time simulation. It is a C++ class library. Moreover, NeMo has bindings for C, Matlab, and Python. The software is provided under an open source license.

GPU-enhanced Neuronal Networks (GeNN) [23] is another framework for simulating SNN on GPU. It is an open source library that generates code to accelerate the execution of network simulations on NVIDIA GPUs. It is entirely based on CUDA and C/C++. It is flexible and easily extended software – any neuron model can be simulated. In GeNN users can introduce their own neuron models, synapse models, post-synaptic integration models and synaptic plasticity rules by providing code snippets that are substituted into the network simulation during code generation. GeNN is available for Linux, Mac OS and Windows platforms.

The Myriad [24] CUDA GPU-enabled simulator focuses on realistic biophysical models using H&H neurons and densely integrated network models that scale poorly on clusters of computers. These models have many analogue interactions such as gap junctions and graded synapses that require many model elements to update one another at each timestep. Myriad provides a flexible and extensible interface through a Python module, which is then translated into a C-based implementation layer by code generation.

Table 1
Summary of selected SNN simulators

	Brian	NEST	NEURON	NeMo
Time-driven	Yes	Yes	Yes	Yes
Event-driven	No	No	Yes	No
GPU	Yes	No	No	Yes
Linux	Yes	Yes	Yes	Yes
Windows	Yes	No	Yes	No
Easy installation	Yes	Yes	No	No

SNN simulators – a summary. Selected parallel environments for SNN simulation were installed and tested by the author of this paper. Table 1 presents the summary of their evaluation. The focus is on their implementation and functionality.

4. SNNS – Spiking Neural Network Simulator

4.1. Description of SNNS

SNNS is a GPU-enabled software environment for spiking neural networks simulation using the OpenCL programming model. The aim was to provide a framework which allows performing effective experiments with less and more complicated models of spiking neural networks on various GPUs. It enables simulation study with three models: Integrate-and-fire, Hodgkina-Huxley and Izhikevich neuron model. SNNS implements time-driven simulation mode and NS-parallel parallelization strategy. Two of the system's principal goals are portability and usage in heterogeneous computing environments. SNNS can be executed on GPUs from many vendors.

During simulation experiment performed under SNNS one can distinguish three main stages: preparatory stage, experimental stage and recording test results. At the preparatory stage a neural model, presynaptic and postsynaptic neurons, spiking generator and all initial parameters (total number of neurons and synaps, initial values of state variables, etc.) are provided. The SNNS framework cooperates with the Brian simulator [19]. The neural network to be simulated is generated using tools from Brian. The special programme for recording the generated network into the disc file in the comprehensive SNNS format has been developed.

The experimental stage begins when all decisions regarding the simulated network are made. The corresponding computing modules (kernels) are executed in sequence. The spikes generated at each timestep are collected, neuron states are updated and new spikes are firing and propagating across the network.

Finally, all test results are recorded into a disc file in the defined format. They can be easy visualized in popular graphical programs. The Brian&SNNS system flow diagram is presented in Fig. 2.

4.2. Architecture of SNNS

Programming for a GPU is rather specialised and needs additional effort from the programmer [1]. Due to the fact that the cores found on GPUs are less complex the programs that are executed should be especially tuned for minimizing their limitations and maximizing their potential to provide high level of parallelization. Algorithms that do not take into account the architecture of the GPU will not use it efficiently. In particular, the constraints on memory access patterns have to be considered. Therefore, tuning GPU algorithms to the specific hardware is highly recommended.

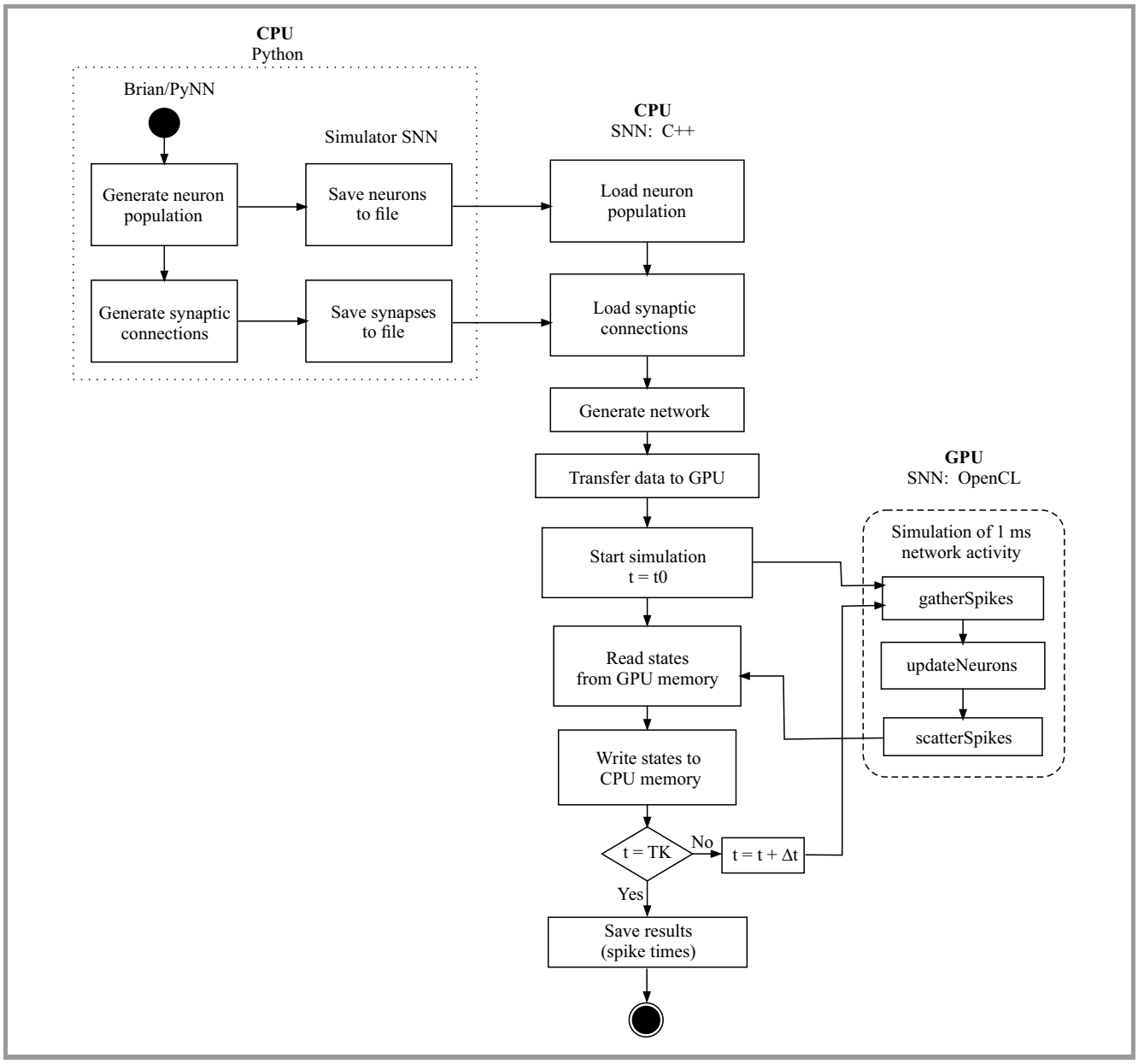


Fig. 2. The Brian&SNNS system flow diagram.

In order to take advantage of GPU accelerators from different vendors OpenCL [25], [26], which is a low level GPU programming toolkit was used. OpenCL is an industry standard computing library developed in 2009 that targets not only GPUs, but also CPUs and potentially other types of accelerator hardware. In OpenCL efficient implementation requires preparation slightly different codes for different devices, however it is much less complicated than writing code in many native toolkits for NVIDIA and AMD devices.

4.3. SNNS Components

SNNS consists of seven components. Its architecture is depicted in Fig. 3. All components have been implemented

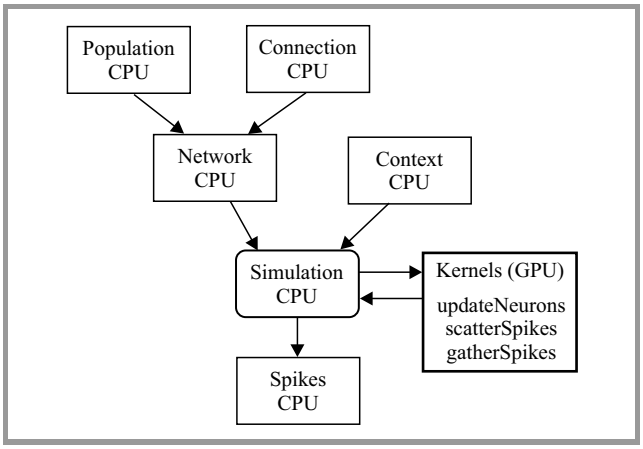


Fig. 3. The SNNS simulator components.

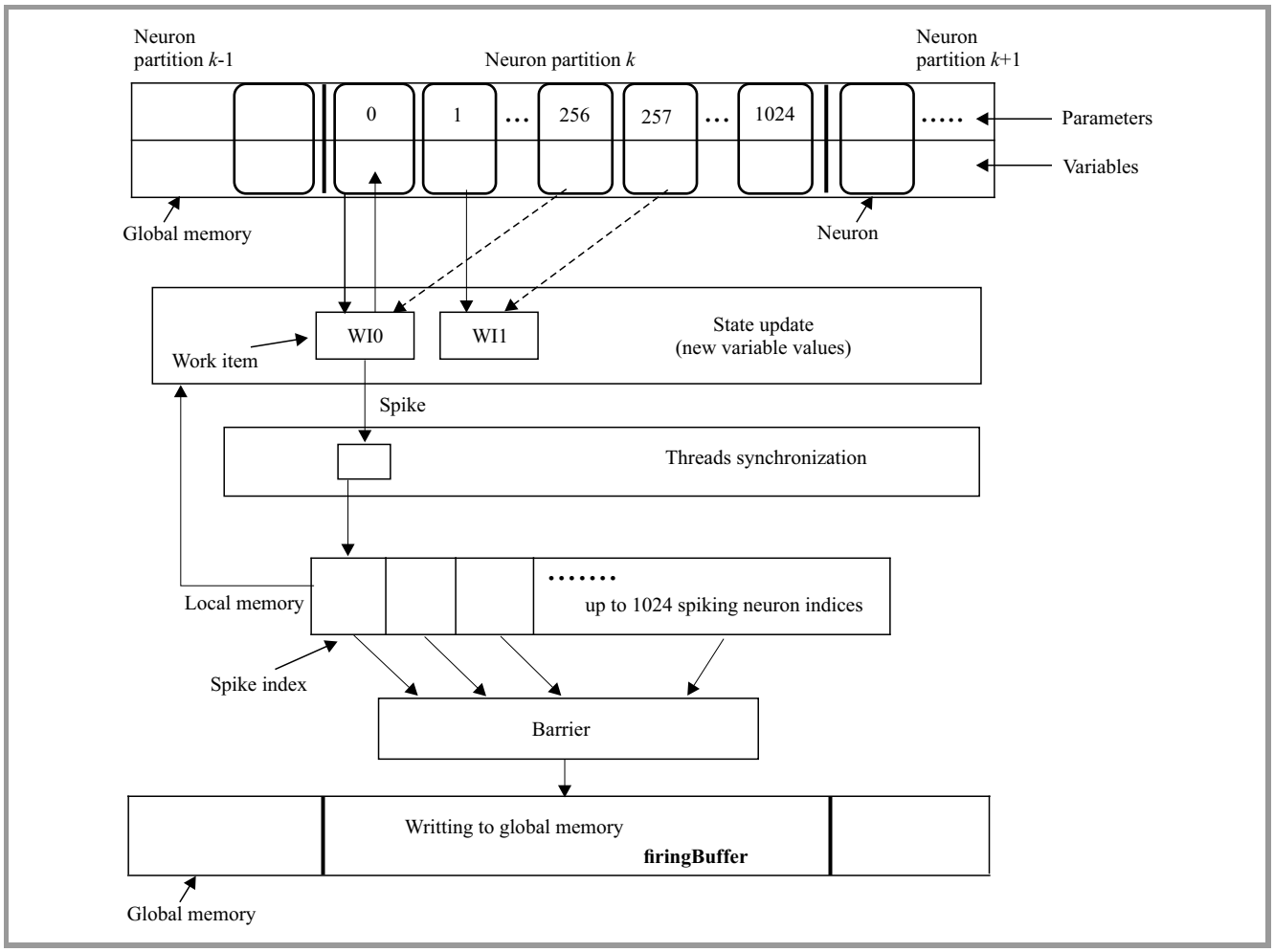


Fig. 4. The performance of the `updateNeurons` kernel.

in OpenCL and C++ and can be executed on CPU and GPU processors. The specification of components is as follows:

- Population – downloading neurons from the disc file (C++),
- Connection – downloading synaptic connections from the disc file (C++),
- Network – generation of network to be simulated (C++),
- Context – CPU – GPU communication mechanisms definition (C++),
- Simulation – a simulation scenario definition (C++),
- Kernels – network simulating (computations) (OpenCL),
- Spikes – test results recording into a disc file (C++).

The goal was to develop an effective, flexible and failure resistance software. Therefore, the main component of the system – *Kernel* – were decomposed into three kernels that perform the following operations:

- `updateNeurons` – neuron states updating,
- `scatterSpikes` – spikes propagating across the network,
- `gatherSpikes` – collecting spikes received by all neurons at each timestep.

The performance of the `updateNeurons` kernel is presented in Fig. 4. It implements the forward and exponential Euler methods for numerical integration [27].

4.4. Memory Issues

It is obvious that mentioned above kernels need to access at each timestep a large amount of memory, since all neurons and synaptic variables corresponding to received spikes have to be accessed. Due to the fact that neuron and synaptic operations are often simple, the speed of memory access limits the efficiency of SNNS. The shared memory of GPU is fast but is very limited. The global memory is very slow. Therefore, the most critical issue is the optimization of read/write memory access to the values of synapses and neural variables at each timestep. The mem-

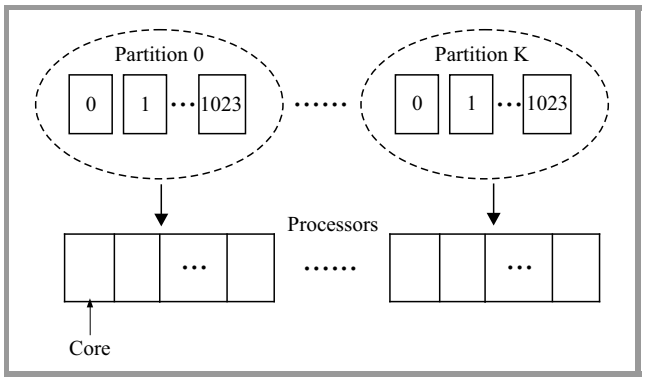


Fig. 5. Allocation of neurons to partitions and processors.

ory transfers on GPU are much faster if variables that are accessed at the same timestep are stored contiguously.
To maximize the speed of SNNS the particular attention was paid on the design of efficient data structures. The implementation is as follows. In each simulation experiment a population of neurons with unique identifiers is divided into partitions. To reduce the competition for memory access each partition is assigned to one GPU processor (see Fig. 5). Moreover, all synapses are divided into separated groups. Synapses with the same presynaptic neuron are collected to one group. Synapses from the same group are aggregated into the packages of fixed size and propagated across the network. Such an implementation allows to reduce memory usage.

5. SNNS Numerical Evaluation

The SNNS framework was used to simulate spiking neural networks with various size (1000 to 30000 neurons). Simulation experiments were conducted for following models:

- Network I&F – Integrate-and-fire model, spike average frequency 7 Hz.
- Network I – Izhikevich model, spike average frequency 15-25 Hz.
- Network H&H – Hodgkin-Huxley model, spike average frequency 15–30 Hz.

The experiments were performed on the following hardware platforms:

- P1: Intel Core2 Quad 2.83 GHz, Radeon HD 6700, 4 GB RAM, Linux x64.
- P2: Intel Core i5-2500K, 3.30 GHz, Radeon HD 6900, Linux x64.
- P3: Intel Core i5-2500K, 3.30 GHz, GeForce GTX 560T, Linux x64.

The results of simulations, i.e. times of calculations performed for various size of networks are presented in Figs. 6–12.

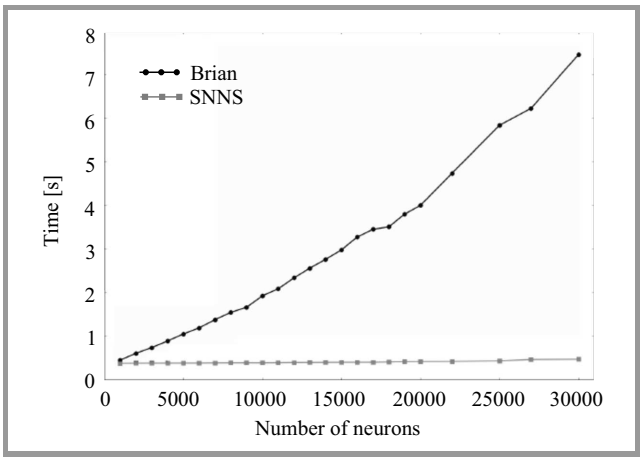


Fig. 6. Simulation time for Brian and SNNS simulators (Network I&F).

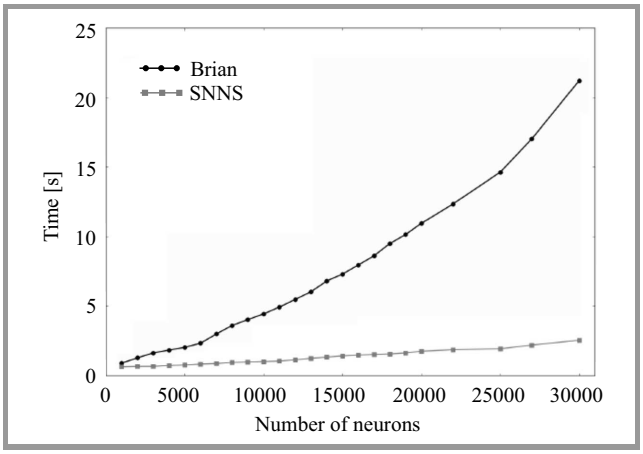


Fig. 7. Simulation time for Brian and SNNS simulators (Network I).

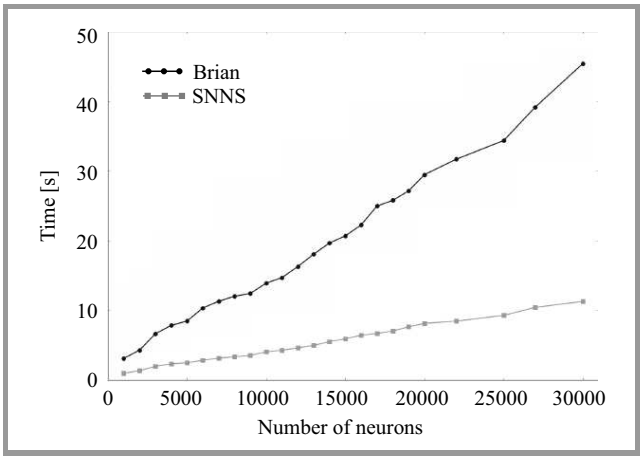


Fig. 8. Simulation time for Brian and SNNS simulators (Network H&H).

5.1. Comparative Study of CPU and GPU Simulators

The aim of the first series of experiments was to compare the efficiency of spiking neural networks simulation on CPU and GPU processors. The performance of the

CPU-enabled Brian simulator and the GPU-enabled SNNS framework was evaluated and compared. The experiments were performed on P1 and P2 hardware platforms. The results obtained for three neuron models are presented in Figs. 6–8.

The presented results show that the speed of simulation strongly depends on the network size and neuron model considered. The usage of GPU enabled speed up the simulation from 4 times for Network H&H, 9 times for Network H&H to 19 times for Network I&F.

5.2. Comparative Study of GPU Simulators

The aim of the next series of experiments was to compare the efficiency of spiking neural network simulation conducted on GPs from different vendors. First, two hardware platforms P1 and P2 with different computing power were tested. The results obtained for I&F and I neuron models are presented in Figs. 9–10.

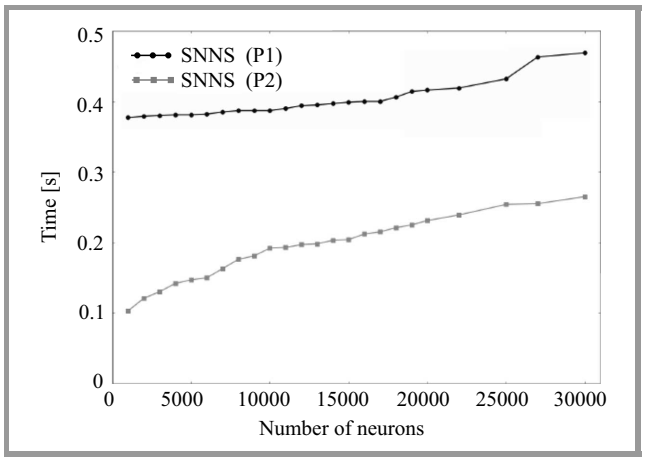


Fig. 9. Simulation time for SNNS simulator, P1 and P2 platforms (Network I&F).

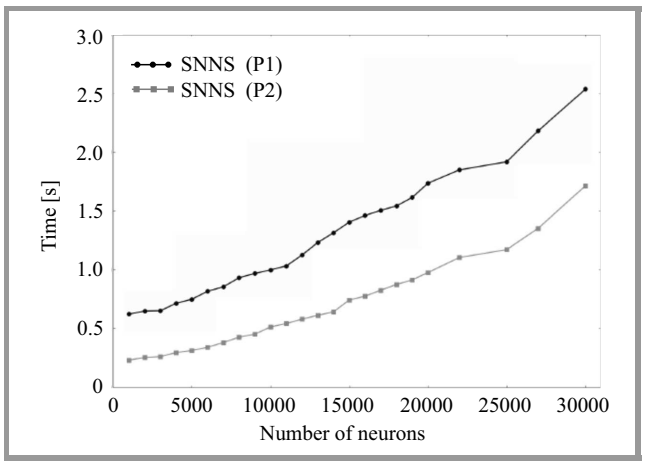


Fig. 10. Simulation time for SNNS simulator, P1 and P2 platforms (Network I).

The usage of more powerful GPU device enabled to speed up the simulation from 2.7 (Network I) to 3.7 (Network I&F) times. The acceleration was decreased with the size of the network.

Finally, the SNNS simulator was compared with the NeMo [21] CUDA-enabled framework for large scale networks simulation. Two series of experiments were performed for Network I&F and Network I. The tests for SNNS simulator were conducted on the P2 platform equipped with the AMD graphical processor Radeon HD 6900. The NeMo was executed on the P3 platform equipped with the NVIDIA graphical processor GeForce GTX 560T. The results of simulation experiments, i.e. times of calculations are depicted in Figs. 11 and 12.

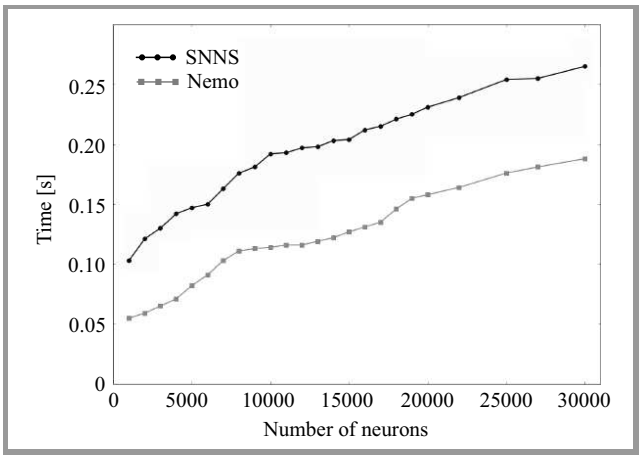


Fig. 11. Simulation time for NeMo and SNNS simulators (Network I&F).

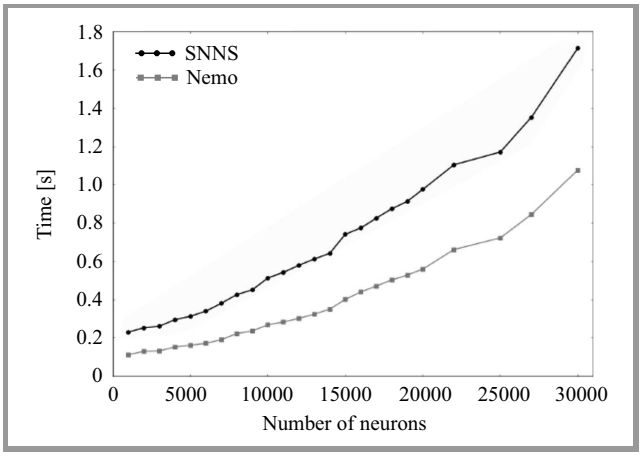


Fig. 12. Simulation time for NeMo and SNNS simulators (Network I).

It is observed that the CUDA-enabled simulator running on NVIDIA hardware gave a better results than the OpenCL one. The simulation time of neural network formed by 1000 neurons performed using the SNNS framework was decreased about 2 times for the NeMo framework and NVIDIA GPU. Such result was expected – CUDA is the technology dedicated to NVIDIA GPU. However, the numerical experiments showed that the acceleration level

with respect to OpenCL and AMD GPU decreases with the bigger size of a network.

6. Summary and Conclusion

The paper provides a short overview of methods and tools for parallel spiking neural networks simulation on GPU accelerators. Spiking neural networks are natural candidates for massively parallel computations. SNN simulation requires complex calculations and parallel processing of large volumes of data, in which a speed of calculation and data decomposition are of essence. The attention of the paper is focused on the OpenCL and GPU-enabled software framework SNNS for simulating large scale networks. SNNS was designed to be powerful, effective, scalable, flexible, and easy to use. The experimental results presented in this paper demonstrate the effectiveness of the SNNS framework, and confirm that the direction to speed up complex systems simulation is to port it to GPU units.
As a final observation one can say that CUDA and OpenCL computing systems offer a new opportunity to increase the performance of parallel HPC applications in clusters, by combining traditional CPU and general purpose GPU devices. However, although much progress has been made in software and hardware for HPC computing simulation of large-scale neurobiologically inspired systems is still a challenging task.

Acknowledgment

The author would like to thank Dr. Paweł Wawrzyński for assistance with this research.

References

[1] W.-M. W. Hwu (Ed.), *GPU Computing Gems Emerald Edition*, 1st ed. Morgan Kaufman, 2011.

[2] N. Carnevale and M. Hines, *The NEURON Book*. Cambridge University Press, 2006.

[3] W. Gerstner and W. Kistler, *Spiking Neuron Models*. Cambridge University Press, 2002.

[4] E. M. Izhikevich, *Dynamical Systems in Neuroscience: The Geometry of Excitability and Bursting*. The MIT Press Cambridge, 2007.

[5] J. Vreeken, "Spiking neural networks: An introduction", Tech. Rep., Artificial Intelligence laboratory, Intelligent Systems Group, University of Utrecht, 2003.

[6] R. Brette *et al.*, "Simulation of networks of spiking neurons: A review of tools and strategies", *J. Computat. Neuroscience*, vol. 23, no. 3, pp. 349–398, 2007.

[7] K. D. Carlson, J. M. Nageswaran, N. Dutt, and J. L. Krichmar, "An efficient automated parameter tuning framework for spiking neural networks", *Front. in Neuroscience*, vol. 8, art. 10, pp. 1–16, 2014 [Online]. Available: http://dx.doi.org/10.3389/fnins.2014.00010

[8] Z. Fountas and M. Shanahan, "GPU-based fast parameter optimization for phenomenological spiking neural models", in *Proc. Int. Joint Conf. Neural Netw. IJCNN 2015*, Killarney, Ireland, 2015, pp. 1–8.

[9] E. M. Izhikevich, "Which model to use for cortical spiking neurons", *IEEE Trans. Neural Netw.*, vol. 15, no. 5, pp. 1063–1070, 2004.

[10] A. L. Hodgkin and A. F. Huxley, "A quantitative description of membrane current and its application to conduction and excitation in nerve", *J. Physiology*, vol. 117, no. 4, pp. 500–544, 1952.

[11] E. M. Izhikevich, "Simple model of spiking neurons", *IEEE Trans. Neural Netw.*, vo. 14, no. 6, pp. 1569–1572, 2003.

[12] R. Brette and D. F. Goodman, "Simulating spiking neural networks on GPU", *Network*, vol. 23, no. 4, pp. 167–182, 2012.

[13] M. Chessa, V. Bianchi, M. Zampetti, S. P. Sabatini, and F. Solari, "Real-time simulation of large-scale neural architectures for visual features computation based on GPU", *Network*, vol. 23, no. 4, pp. 272–291, 2012.

[14] J. M. Nageswaran, N. Dutt, J. L. Krichmar, A. Nicolau, and A. V. Veidenbaum, "A configurable simulation environment for the efficient simulation of large-scale spiking neural networks on graphics processors", *Neural Netw.*, vol. 22, no. 5–6, pp. 79–800, 2009.

[15] R. Ananthanarayanan and D. S. Modha, "Anatomy of a cortical simulator", in *Proc. of ACM/IEEE Conf. Supercomput. SC'07*, Reno, NV, USA, 2007, pp. 1–12 (doi: 10.1145/1362622.1362627).

[16] NEURON Simulator [Online]. Available: http://www.neuron.yale.edu/neuron/

[17] NEST Simulator [Online]. Available: http://www.nest-initiative.org/

[18] D. Goodman and R. Brette, "The Brian simulator", *Front. in Neuroscience*, vol. 3, no. 2, pp. 192–197, 2009.

[19] Brian Simulator [Online]. Available: http://briansimulator.org/

[20] Mvaspike Simulator [Online]. Available: http://mvaspike.gforge.inria.fr/

[21] NeMo Simulator [Online]. Available: http://nemosim.sourceforge.net/

[22] A. Fidjeland, E. Roesch, M. Shanahan, and W. Luk, "NeMo: a platform for neural modelling of spiking neurons using GPUs", in *20th IEEE Int. Conf. Application-specific Syst., Architec. & Processors ASAP 2009*, Boston, MA, USA, 2009.

[23] GeNN Simulator [Online]. Available: http:genn-team.github.io/genn/

[24] Myriad Simulator [Online]. Available: http://cplab.net/myriad/

[25] OpenCL – The open standard for parallel programming of heterogeneous systems [Online]. Available: http://www.khronos.org/opencl/

[26] E. Bainville, "OpenCL multiprecision tutorial", Jan. 2010 [Online]. Available: http:// http://www.bealto.com/mp-opencl.html

[27] R. R. D. Stewart and W. Bair, "Spiking neural network simulation: numerical integration with the Parker-Sochacki method", *J. Computat. Neuroscience*, vol. 27, pp. 115–133, 2009 (doi: 10.1007/S10827-008-0131-5).

Paweł Szynkiewicz received his M.Sc. in Computer Science from the Warsaw University of Technology, Poland, in 2015. Currently he is a Ph.D. student in the Systems Research Institute, Polish Academy of Science. In 2015–2016 employed in Comarch. Since 2016 he is with BrightSolutions IT. His research area focuses on software technologies, HPC computing, neural networks and machine learning, genetic algorithms, computer networks security.
E-mail: pszynk@gmail.com
Systems Research Institute
Polish Academy of Science
Newelska st 6
01-447 Warsaw, Poland

Paper

Forecasting Stock Price using Wavelet Neural Network Optimized by Directed Artificial Bee Colony Algorithm

Thanh Tung Khuat, Quang Chanh Le, Bich Loan Nguyen, and My Hanh Le

University of Danang, University of Science and Technology, Danang, Vietnam

Abstract—Stock prediction with data mining techniques is one of the most important issues in finance. This field has attracted great scientific interest and has become a crucial research area to provide a more precise prediction process. This study proposes an integrated approach where Haar wavelet transform and Artificial Neural Network optimized by Directed Artificial Bee Colony algorithm are combined for the stock price prediction. The proposed approach was tested on the historical price data collected from Yahoo Finance with different companies. Furthermore, the prediction result was found satisfactorily enough as a guide for traders and investors in making qualitative decisions.

Keywords—Artificial Bee Colony algorithm, Artificial Neural Network, back-propagation algorithm, stock price forecasting, wavelet transform.

1. Introduction

The stock price prediction is one of the most important topics in finance and business. An intelligent system would predict the stock price and give a guide to investors to buy a stock before the price rises, or sell it before its value declines. Though it is very hard to replace the role of experts, an accurate prediction algorithm can directly result in high profits for investment companies. The efficient algorithm can also indicate a direct relationship between the accuracy of the prediction algorithm and the profit taken from the use of the algorithm. However, the stock market trends are nonlinear, uncertain, and non-stationary and nowadays it tends to be more risk than before for forecasting the stock price [1]–[3].

Artificial Neural Network (ANN) is one of data mining techniques being widely accepted in the business area due to its ability to learn and detect relationships among nonlinear variables. Several studies have shown that the ANN outperforms statistical regression models and also allows deeper analysis of large data sets, especially those that have the tendency to fluctuate within a short of period of time [4]–[7]. However, in the case of financial forecasting for enormous time series, appropriate data preprocessing techniques and optimization algorithms are required to enhance the accuracy of the predicted results.

In this study, the prediction system is built by the combination of preprocessing techniques including Haar wavelet and a neural network optimized by using Directed Artificial Bee Colony (DABC) algorithm [8]. The Haar wavelet is utilized to decompose the stock price time series and eliminate noise, since the representation of a wavelet can tackle the non-stationary involved in the economic and financial time series [9]. The Artificial Bee Colony (ABC) algorithm is a novel meta-heuristic approach proposed by [10]. Due to the advantages of memory, multi-characters, local search, and a solution improvement mechanism, this algorithm can be used for identifying high-quality optimal solutions and offering the balance between complexity and performance, as well as optimizing predictions effectively. In this study, the DABC which is the improved version of the ABC is used to optimize the weights and biases of ANN before training the network by back-propagation (BP) algorithm. The main goal of this study is to figure out the efficiency of the ANN improved by using DABC for tackling the regression problem on a particular domain such as the stock market.

The remaining of the paper is organized as follows. Section 2 presents some works related to research fields. The proposed approach for the stock price prediction is shown in Section 3. Section 4 describes the experiments and obtained results. Conclusion and future works are presented in Section 5.

2. Related Works

Numerous researches about financial data mining have been done. Kim and Chun [11] implemented a neural network system using the technical analysis variables for listed companies in Shanghai Stock Exchange. They have compared the performance of two learning algorithms and two weight initialization methods. The results indicated that the forecasting of stock market is quite acceptable with both the algorithm and initialization methods, but the performance of the back-propagation can be increased by conjugate gradient learning and multiple linear regression weight initializations. However, when the structure of ANN gets to be complex and there are large training samples, convergent speed in these algorithms will become very slow. This influences the accuracy of the predicted results of ANN. To cope with this problem, a new method using the evolutionary algorithms is proposed in some researches [5], [12]–[14].

Kim and Han [15] used a genetic algorithm to transform continuous input values into discrete ones. The genetic

algorithm was used to reduce the complexity of the feature space. Kishikawa and Tokinaga [16] applied a wavelet transform to extract the short-term feature of stock trends. The past works have used various forecasting techniques in order to predict the stock market trends. Some methods attempted to forecast the daily returns, while some other studies developed forecasting models to predict the rate of returns of individual stocks. In many papers, it was also found that researchers have attempted to compare their results with other statistical tools. Each approach has advantages and disadvantages, so it is able to use one of them to hide the disadvantage of another. These findings provide a strong motivation for modeling forecasting tools for stock market prediction. Besides applying wavelet-based data preprocessing, this study uses the DABC algorithm to optimize the weights and biases of ANN to enhance the accuracy for results of the stock price prediction.

3. Methodology

In general there are two stock prediction methodologies: technical and fundamental analyses. Technical analysis using time-series analysis to deal with the determination of the stock price based on the historical data, while fundamental analysis concentrates on the forces of supply, the past performance of the company and the earnings forecast.

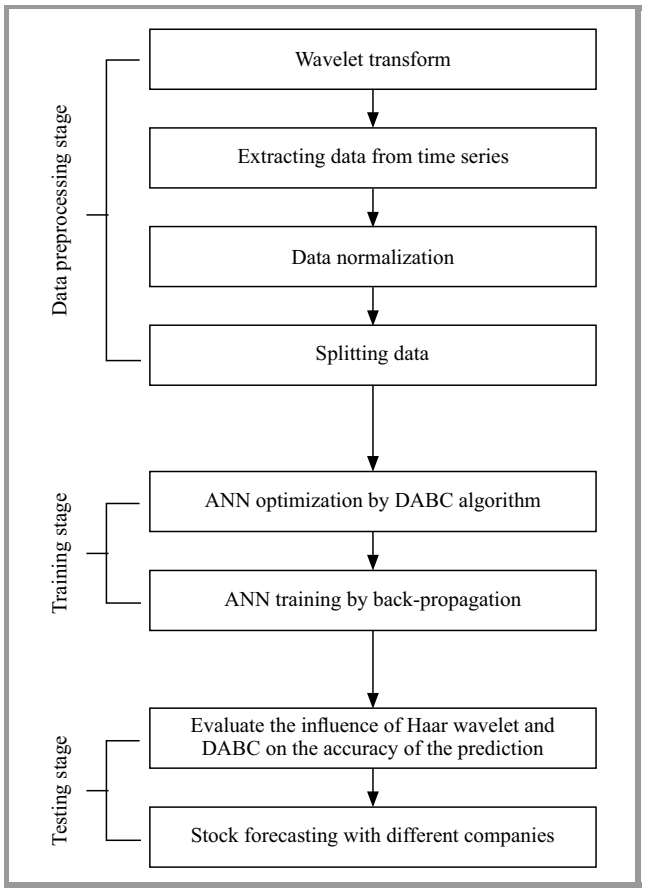


Fig. 1. Proposed approach overview.

To involve both fundamental and technical analyses, this study presents a novel approach that integrates the Haar wavelet transform and the DABC algorithm into the Multilayer Perceptron (MLP) neural network. Figure 1 briefly shows the main process used in this work and it will be explained in more details below.

3.1. Choosing Data Formatting

The factors utilized for training the ANN are chosen based on the experience of trader with regard to the specific stocks. There are many technical indicators and fundamental factors such as: Moving Average (MA), Relative Strength Index (RSI), Boilinger bands, Close/open prices, Volume oscillator being able to be used to analyze the stock market.

This work uses the close price to train the ANN, so the output of the ANN will be the close price as well.

3.2. Data Preprocessing

3.2.1. Noise Filtering Using Haar Wavelet Transform

The first stage of data preprocessing is the use of Haar wavelet to decompose the financial time series and remove noise since the representation of a wavelet can tackle the non-stationarity involved in the economic and financial time series [9]. Wavelets are mathematical functions that break data into various frequency components, and then each component is studied with a resolution matched to its scale. There are a wide variety of popular wavelet algorithms including Daubechies wavelets, Mexican Hat wavelets and Morlet wavelets. These wavelet algorithms have the advantage of better resolution for smoothly changing time series. However, they have the disadvantage of being more expensive to compute than the Haar wavelets. Therefore, this study uses the Haar wavelet which is the simplest algorithm and works well for the stock price time series.

A time series can be viewed in multiple resolutions when using wavelets. Each resolution represents a different frequency. The wavelet method computes averages and differences of a signal, breaking the signal down into spectrums. The Haar wavelet algorithm works on time series whose size is a power of two values (e.g., 32, 64, 128...). Each step of the wavelet transform generates two sets of values: a set of averages and a set of differences (the wavelet coefficients). Each set is half the size of the input data. For example, if the time series has 128 elements, the first step will generate 64 averages and 64 coefficients. The set of averages then becomes the input for the next step (e.g., 64 averages generating a new set of 32 averages and 32 coefficients). This process is repeated until one average and one coefficient are obtained.

The strength of two coefficient spectra generated by a wavelet calculation reflects the change in the time series at different resolutions. The first coefficient band shows the highest frequency changes. This is the noisiest part of the

time series. This noise can be eliminated by using threshold methods. Each later band reflects changes at lower and lower frequencies.

3.2.2. Extracting Data from Time Series

In the stock price prediction, authors have to decide that how many prices of the recent days will be used to predict the price of the next day. That value is called as "windowSize". Traders can use any values for windowSize that they want, commonly in the range of 10 to 180 days.
This work uses 30-to-1 model that means using 30 recent days to forecast the next day. WindowSize is also the number of inputs used in the input layer of the ANN. To train the ANN, we need many 30-to-1 sequences, and each sequence consists of two vectors. The input vector includes 30 prices of 30 recent days while the output vector comprises the price of the next day. In order to obtain n sequences, we have to slide the window back n steps, and then extract one sequence at each step [7].

3.2.3. Data Normalization

Normalization is a process transforming the time series data points into a small pre-specified range generally from 1 to –1 or 0 [17]. In order to facilitate the training process, the data needs to be normalized before training the ANN because the prices are in the different ranges. This study uses Vector Normalization [18] for normalizing data.
Mathematical formula of Vector Normalization is shown as the Eq. 1:

$$N_i = \frac{T_i}{\sqrt{\sum_{j=1}^{k} T_j^2}}, \tag{1}$$

where N_i is the normalized data and T_i is the time-series data, k is the number of values in series, and $i = 1,\ldots,k$.

3.2.4. Splitting Data into Training, Validation and Testing Sets

One of the problems that occur during neural network training is called overfitting. In this case, the error on the training set is very small but when a new data is presented to the network, the error is high. In other words, the ANN performs well on training data and poorly on data it has not seen. This is due to the fact that the network has memorized the training samples but has not learned to generalize to new situations. The ANN will therefore not possess the generalization ability and will give a poor predictive performance.
For the purpose of resolving the overfitting problem, the data will be randomly separated into the training set and validation set. This is one of the simplest and most widely used means for avoiding overfitting [19]. The training set is the data set used to adjust the weights on the neural network. The validation set is used to minimize overfitting, and the weights of the network associated with this data set are not adjusted during the training process. If the accuracy over the training set increases, but the accuracy over then validation set stays the same or decreases, then the ANN is overfitting and should stop training. The accuracy was evaluated by different errors, and in order to speed up the computation, authors ran the validation every 5 training epochs.
Note that the testing set is different with the validation set, because the validation data is independent of the training data. The testing set is used only for testing the final solution in order to confirm the actual predictive power of the network. Therefore, the testing set would be used to evaluate the prediction ability of the proposed approaches.

3.3. MLP Neural Network Setting

In general, the architecture of MLP-ANN can have many hidden layers and each hidden layer can include many neurons. However, theoretical works have shown that a ANN with one hidden layer is good enough to approximate any complex non-linear functions [17], [20]. In addition, many studies and experimental results also indicate that one hidden layer is sufficient for most of the forecasting problems [4], [17], [21]. Therefore, this work uses the architecture of MLP-ANN with one hidden layer.
Other difficult tasks when choosing good parameters for ANN are the number of hidden neurons and activation function. Setting a suitable architecture of the ANN for a particular problem is an important task, because the network topology directly affects to its computational complexity and generalization ability. Too much hidden layers or hidden neurons will drive the ANN to the overfitting. Based on conducted experiments and other researches [7], [22], the ANN with 8 neurons for the hidden layer and Bipolar Sigmoid function (Fig. 2) as activation function for both hidden and output layer is suitable for forecasting the stock price.

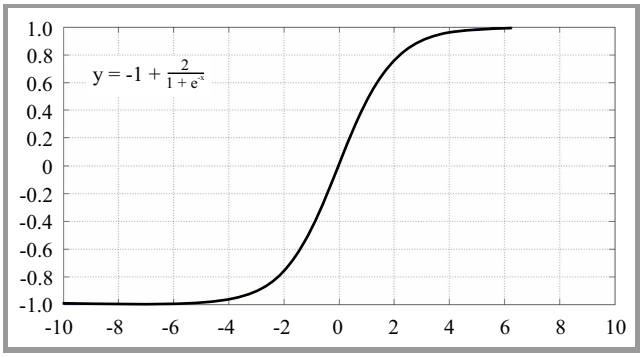


Fig. 2. Bipolar Sigmoid function.

Figure 3 shows the structure of the ANN used for proposed prediction system. The input layer is mapped with the input vector containing 30 (windowSize) latest close prices. The output layer including one neuron denotes the close price of the next day.

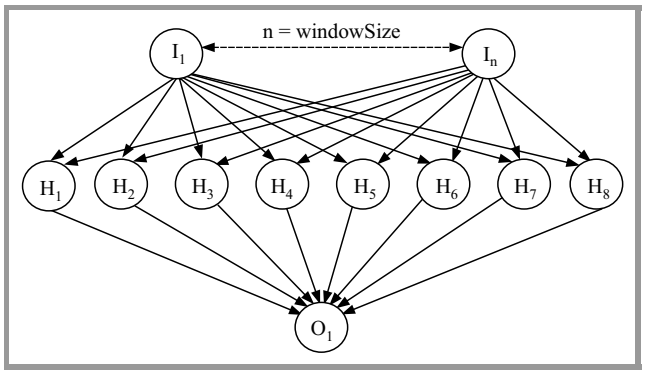


Fig. 3. The architecture of proposed ANN – windowSize-8-1.

It can be also known that the ANN structure also depends on the experience of trader and other factors, so proposed settings for ANN's parameters are just a recommendation for traders.

3.4. MLP Neural Network Training

After the data preprocessing process, the ANN will be trained via two stages. In the first stage, the weights and biases for ANN need to be optimized by using the DABC algorithm to overcome getting stuck in local optima. The DABC algorithm is an effective method for enhancing the convergence to global optimum [8]. In the second stage, the ANN with weights and biases being optimized by the DABC will be trained by back-propagation algorithm.

3.4.1. Optimizing Weights and Biases of the ANN Using the DABC Algorithm

Artificial Bee Colony algorithm being one of the swarm intelligence algorithms is inspired by the foraging behaviors of real honeybee colonies. In ABC algorithm, a colony of artificial bees has three kinds of bees: employed bees, onlookers, and scouts. In this paper, a food source represents a possible set of optimized weights and biases for ANN. Employed bees will exploit a food source and then share information about their food sources with onlooker bees waiting on the dance area within the hive, while scout bees will randomly search for new food sources surrounding the hive to replace the ones abandoned by the employed bees. The details of the DABC algorithm are presented as Algorithm 1.

In this algorithm, fit_i is the fitness value of food source (solution) x_i, v_i is the food source near x_i, p_i is the probability value of x_i, x_{ij} is the j^{th} dimension of x_i, d_{ij} is the direction information of x_{ij}, and *MCN* is the maximum number of cycle in the algorithm. The number of employed bees is equal to the number of food sources. The employed bee whose trial counter is higher than the predefined limit becomes a scout bee.

After the data pre-processing process, the weights and biases of the ANN are initialized by using the directed ABC algorithm. The accuracy of the output of the ANN depends on its initial values of weights and biases. In this work, the

Algorithm 1: The pseudocode for DABC algorithm

```
Load training samples
Generate the initial population x_i, i = 1,...,SN by using Eq. 4
Set the direction information d_ij = 0, i = 1,...,SN; j = 1,...,D
Calculate the fitness fit_i of the population by Eq. 5
cycle = 1
repeat
    for each employed bee do
        Produce new solution v_i by using Eq. 6
        Calculate the value fit_i
        Apply greedy selection process
        Update the direction information
    end for
    Calculate the probability p_i for the solutions x_i by using Eq. 7
    for each onlooker bee do
        Apply Roulette Wheel to select a solution x_i
        Produce a new solution v_i from x_i
        Calculate the value fit_i of v_i
        Apply greedy selection process
        Update the direction information
    end for
    If there is an abandoned solution for the scout then replace it with a new solution, which will be randomly produced by using Eq. 4
    Memorize the best solution so far
    cycle = cycle + 1
until cycle = MCN
```

DABC is used to seek the optimal set of initial weights and biases prior to the training process in order to enhance the convergence speed and the rigor of output values of the ANN. The process of optimizing the biases and weights is conducted by minimizing an objective function such as the mean square error given by Eq. 2:

$$E(\vec{w}_i(t)) = \frac{1}{N} \cdot \sum_{j=1}^{T} (O_j - Y_j)^2, \tag{2}$$

where T is the number of patterns in the training data set, $E(\vec{w}_i(t))$ is the error at the t-th cycle in the algorithm, $\vec{w}_i(t)$ is the vector of the weights of the ANN being included in the i-th individual at the t-th cycle, O_j and Y_j are the desired output and actual value of the j-th training data respectively.

At the beginning of the algorithm, the population is randomly generated and the direction information of all dimensions equals to 0. *SN* is the number of solutions, also the number of employed or onlooker bees. In this work, a food source or an individual represents a possible set of optimized weights and biases for the ANN. Each solution x_i is a D-dimensional vector containing weights and biases of the ANN. As for the proposed ANN architecture (windowSize-8-1), the value of D is the total number of weights and bias of the ANN that needs to be optimized and can be calculated by using Eq. 3 whose details are shown in Table 1:

$$D = IW\{1,1\} + b\{1,1\} + LW\{2,1\} + b\{2,1\}. \tag{3}$$

Table 1
Parameters for proposed ANN (windowSize-8-1)

Value	Symbol	Description
WindowSize · 8	IW{1, 1}	Weights of the connections from input to hidden layers
8	b{1, 1}	Biases of neurons in hidden layer
8 · 1	LW{2, 1}	Weights of the connections between output and hidden layers
1	b{2, 1}	Biases of output neurons

After initialization, the population of the food source positions goes through repeated cycles, $cycle = 1,2,\ldots,MCN$, of three search processes for the employed bees, the onlooker bees, and scout bees phases. An employed bee searches around the current food source for a new food source and recalculates the nectar amount (fitness value) of the new food source. If the nectar amount of the new food source is higher than the previous one, the bee memorizes the new food source position and forgets the old one. Otherwise, it keeps the previous food source position. This is a greedy selection process. After all employed bees finish their neighborhood search, they share the nectar amount of the food sources and their position with the onlooker bees on the dance area. An onlooker bee assesses the nectar information given by all employed bees and selects a food source based on the probability associated with its nectar amount. Similar to the employed bee, the onlooker bee produces a new position and recalculates the nectar amount of the new position. It then applies the same greedy selection process as in the employed bee phase. In the initialization, a food source for employed bee is produced by using Eq. 4:

$$x_i^j = x_{min}^j + r \cdot (x_{max}^j - x_{min}^j), \tag{4}$$

where $i \in \{1,2,\ldots,SN\}, j \in \{1,2,\ldots,D\}$ and r is a random number in the range of $[0\ldots1]$.

The fitness value is computed as Eq. 5:

$$fit_i = \begin{cases} \dfrac{1}{1+f_i}, & \text{if } f_i > 0 \\ 1 + abs(f_i), & \text{otherwise} \end{cases}, \tag{5}$$

where fit_i is the fitness of the i^{th} food source and f_i is the specific objective function value for the optimization problem. In this work, $f_i = E(\vec{w}_i(t))$.

In the DABC algorithm, a new food source position is generated by using Eq. 6:

$$v_{ij} = \begin{cases} x_{ij} + \varphi \cdot (x_{ij} - x_{kj}), & \text{if } d_{ij} = 0 \\ x_{ij} + r \cdot abs(x_{ij} - x_{kj}), & \text{if } d_{ij} = 1 \\ x_{ij} - r \cdot abs(x_{ij} - x_{kj}), & \text{if } d_{ij} = -1 \end{cases}, \tag{6}$$

where $k \in \{1,2,\ldots,SN\}, j \in \{1,2,\ldots,D\}$ are randomly chosen, k must be different from i, abs is the absolute function, d_{ij} is the direction information for j-th dimension of the i-th food source position, φ is a random number in the range of $[-1\ldots+1]$ and r is a random number in the range of $[0\ldots1]$. At the beginning of the algorithm, the direction information for all dimensions is set to 0. If the new solution obtained by using Eq. 4 is better than the old one, the direction information is updated. If the previous value of the dimension is less than the current value, the direction information of this dimension is set to –1, otherwise its direction information of this dimension is set to 1. If the new solution obtained by Eq. 4 is worse than the old one, the direction information of the dimension is set to 0. By this way, the direction information of each dimension of each food source position is used and the local search capability and convergence rate of the algorithm are improved as well [8].

An artificial onlooker bee selects a food source based on the probability value associated with that food source, p_i, computed by Eq. 7:

$$p_i = \frac{fit_i}{\sum_{n=1}^{SN} fit_n}, \tag{7}$$

where fit_i is the fitness value of the solution i and SN is the number of food sources.

After onlooker bee phase, if the trial counter is higher than the limit, a new food source position is randomly produced for this bee by using Eq. 4.

3.4.2. Training the ANN by Back-propagation Algorithm

After optimizing the ANN by using the DABC algorithm, the training process is continued with back-propagation algorithm to adjust the weights in the steepest descent direction (the most negative of the gradients). The ANN will be initialized with the optimized weights and biases in the first training phase and then back-propagation algorithm will be used to train the ANN for 50 cycles more.

4. Experiments

4.1. Evaluation Criteria

The proposed approaches were evaluated according to the root mean squared error (RMSE), the mean absolute error

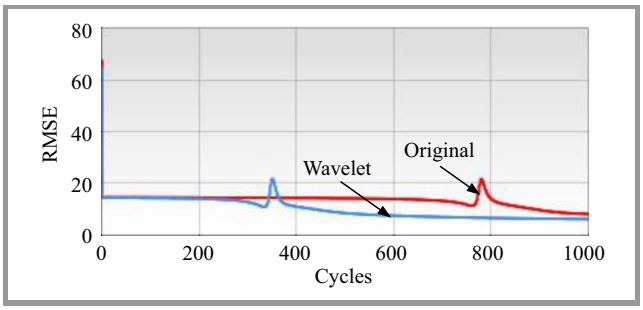


Fig. 4. RMSE of the training process with original data and noise-filtered data (GOOG).

Table 2
RMSE, MAE and MAPE of the training process with original and noise-filtered data

Data	RMSE (USD)		MAE (USD)		MAPE	
Company	Original	Wavelet	Original	Wavelet	Original	Wavelet
AAPL	35.94700656	25.80857226	33.21142773	22.27953928	0.0622156017	0.0452399398
YHOO	5.608004908	4.593439167	4.940398152	3.990453741	0.1038292003	0.0846683761
GOOG	6.561397546	6.058494084	5.155632703	4.477504460	0.0096122608	0.0083394975

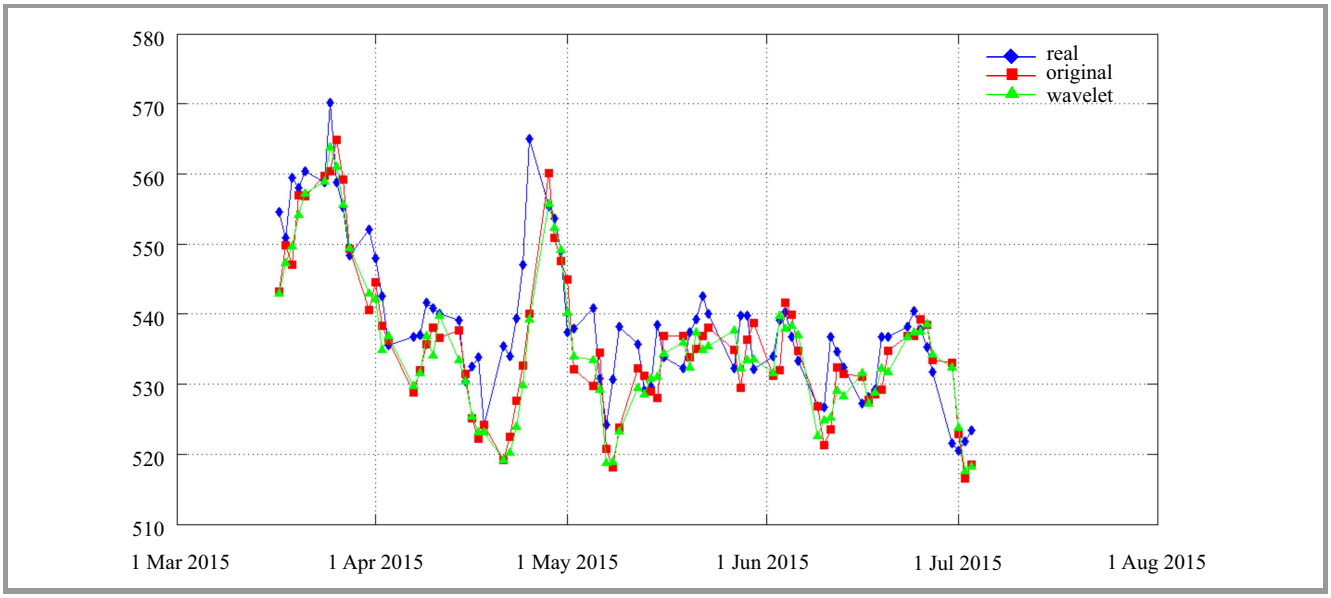


Fig. 5. Real close prices, close price predicted by ANN on original and noise-filtered data (GOOG).

(MAE) and the mean absolute percentage error (MAPE) criteria. These criteria are defined as :

$$RMSE = \sqrt{\frac{1}{N} \cdot \sum_{j=1}^{N} (O_j - Y_j)^2}, \tag{8}$$

$$MAE = \frac{1}{N} \cdot \sum_{j=1}^{N} |O_j - Y_j|, \tag{9}$$

$$MAPE = \frac{1}{N} \cdot \sum_{j=1}^{N} \left| \frac{O_j - Y_j}{Y_j} \right|, \tag{10}$$

where N denotes the size of testing sets.

These criteria measure how the predicted value O is close to the real value Y. The lower these measures are, the better result is. In this study, three these criteria will be used to assess the performance of the following experiments.

4.2. Test Suites

The experiment system is implemented in C# .NET and evaluated on several historical prices data of different companies including the Apple (AAPL) in period 2009–2013, Yahoo! (YHOO) in period 2013–2014 and Google (GOOG) in period 3/2014–7/2015. These data were taken from the datasets in [23].

4.3. The Accuracy of the Stock Price Prediction

As mentioned above, Haar Wavelet transform can eliminate noise. Therefore, it is suitable for handling highly irregular data series.

Using these data to train the network is better than the original data containing a lot of jags. The proof of the benefits of using wavelets is showed in Table 2, Figs. 4 and 5. The use of the Wavelet transform gave the lower error value and the faster convergence of neural network weights.

In this work, the DABC algorithm is used to optimize the network weights and biases by minimizing an objective function such as the root mean square error (RMSE)

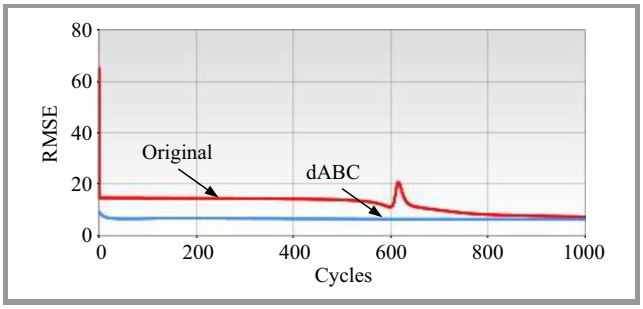


Fig. 6. RMSE of the training process with DABC-ANN and ANN without using DABC (GOOG).

Table 3
RMSE, MAE and MAPE of the training process with DABC-ANN and ANN without using DABC

Data	RMSE (USD)		MAE (USD)		MAPE	
Company	ANN	DABC-ANN	ANN	DABC-ANN	ANN	DABC-ANN
AAPL	36.69341956	28.51912197	33.61652673	25.43316525	0.0692116977	0.0521619069
YHOO	5.559001606	1.272516355	4.644378052	1.047215288	0.1232102173	0.0259282003
GOOG	6.584116070	5.832372776	5.195642497	4.176874924	0.0093781278	0.0077784335

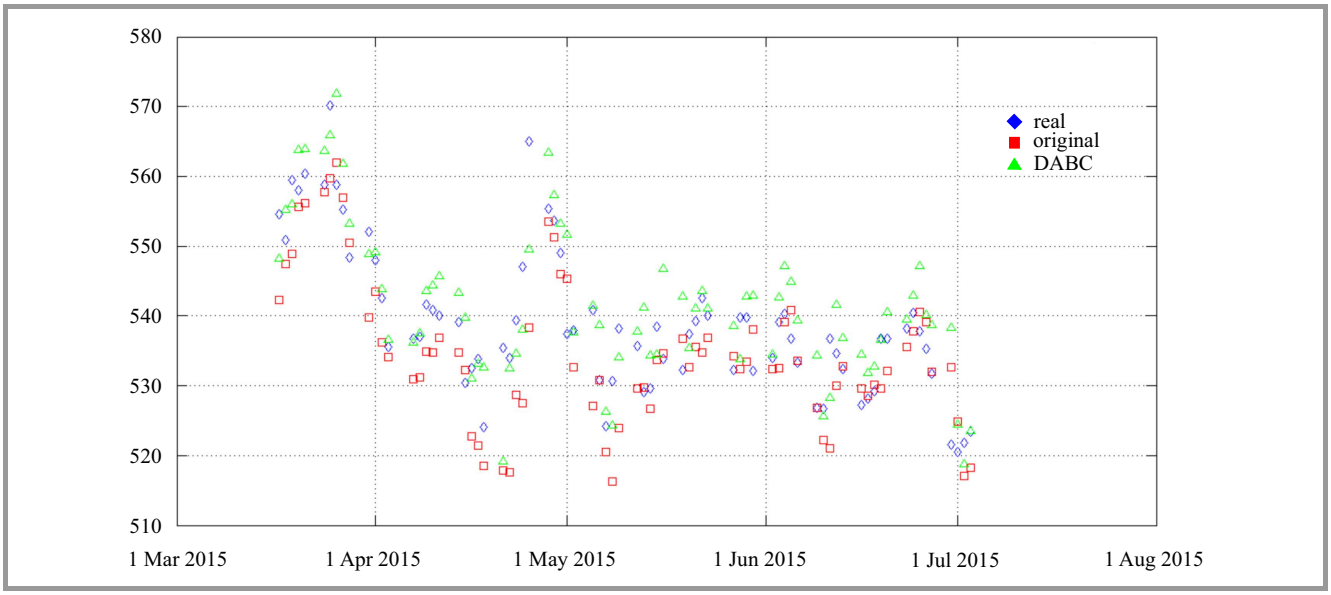


Fig. 7. Real close prices, close price predicted by DABC-ANN and ANN without using DABC (GOOG).

given by Eq. 8. The optimal set of weights and biases found by DABC was used in the next training process of the ANN using back-propagation algorithm. The use of DABC for the optimization of weights and biases gave the lower error value and the faster convergence of neural network weights. Table 3 and Fig. 6 take a proof of the influence of the DABC-ANN model and the ANN (without using the DABC) model. The DABC-ANN gives the lower values through different testing sets. The reduction of values of the criteria given by Eqs. 8–10 is 36.66% on average for three testing data sets. Figure 6 also shows that the RMSE of the DABC-ANN forecasting model converges faster than that of the ANN model. Those figures indicate that the DABC optimization gives the prediction result more accuracy, and it also speeds up the second training stage by reducing the number of training cycles. Figure 7 shows the forecasting results of the DABC-ANN and ANN models and how these prediction values are close to the real values.

4.4. The Prediction Results of the Proposed Approaches

For more accurate in the evaluation of the ANN, each fact possesses a different proportion of training-validating/test set. For the Google in period 2014–2015, the sub-datasets for the first twelve-month period are used for the training-validating process, while those from 3/2015 to 7/2015 are selected for testing. With regard to the Apple and Yahoo, the first ten-month period (87%) is used for the training-validating process and the next two-month period for testing. The statistical meritorious results of testing process are shown in forecasting figures from Fig. 8 to Fig. 10.

4.5. The Execution Time

In this work, the complexity of algorithms depends on the configuration parameters of DABC algorithm and ANN network as well as the size of input values (window sizes). The authors have conducted the experiments and recorded the average time of 20 execution times on each dataset as shown in Table 4. The training data set of Apple is collected in period 2009–2013, Yahoo data set is taken from the period of 2013–2014 and Google data set is in the period from 3/2014 to 7/2015.

Table 4
Execution time of experiments

Data	DABC optimization	ANN training	Total
Company	[s]	[s]	[s]
AAPL	124	3	127
YHOO	30	4	34
GOOG	37	3	40

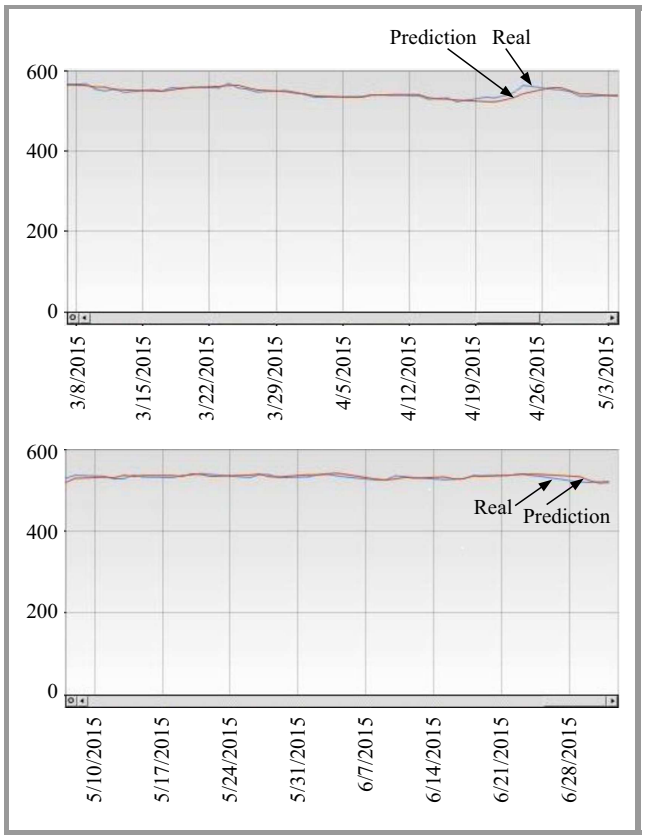


Fig. 8. Testing result of DABC-ANN for Google for period 3/2014–7/2015.

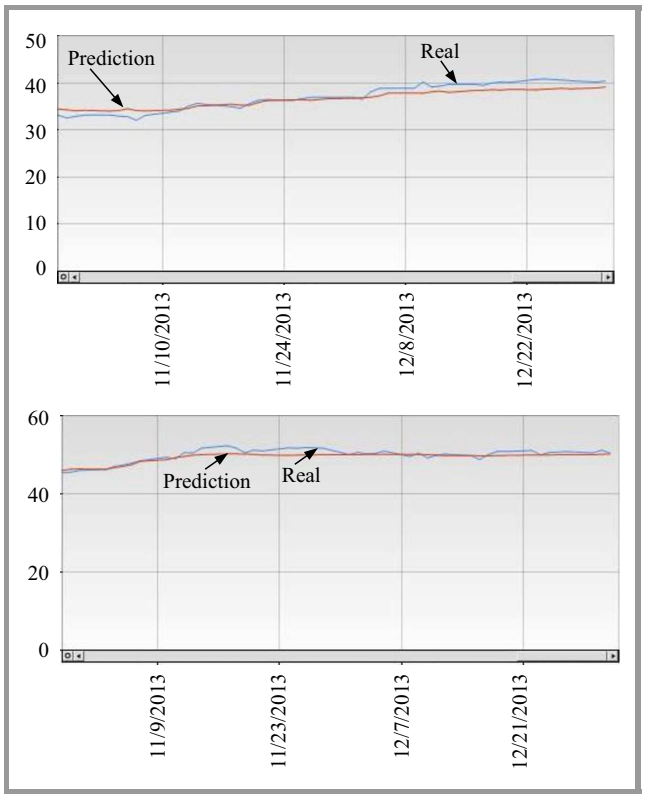


Fig. 9. Testing result of DABC-ANN for Yahoo for period 2013–2014.

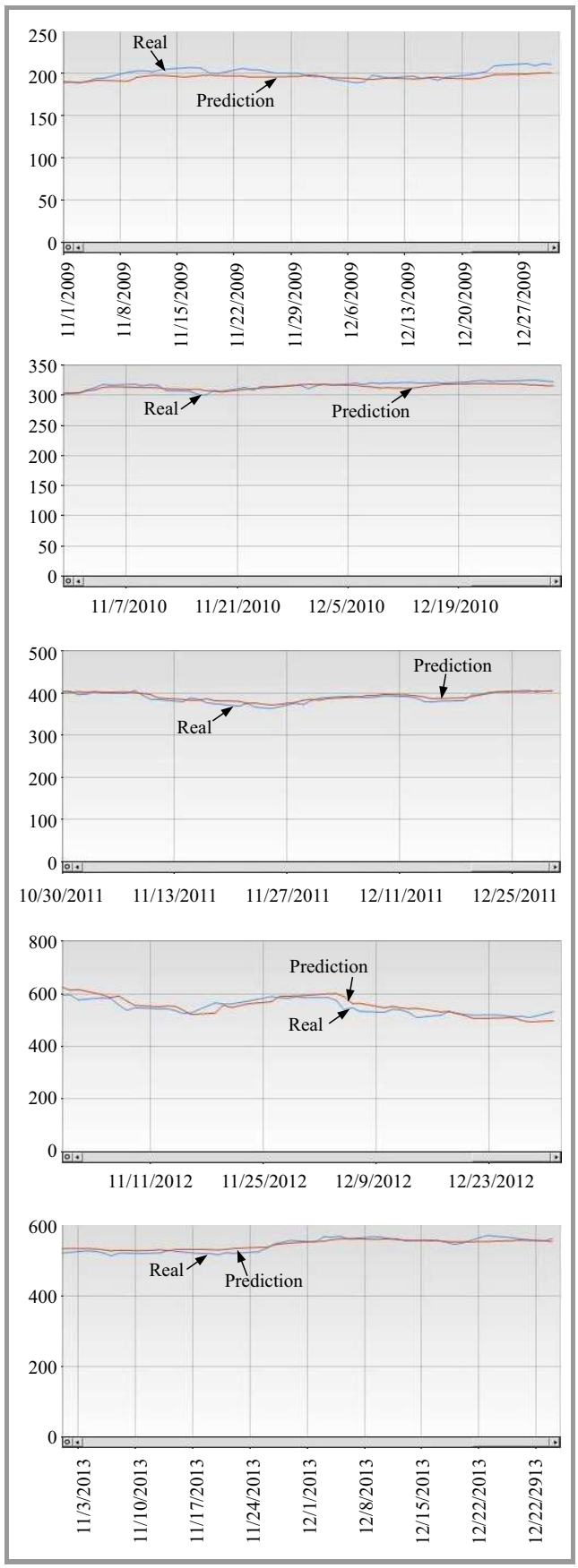


Fig. 10. Testing result of DABC-ANN for Apple for period 2009–2013.

It can be stated that such computation time might meet the requirements of scalping systems in which length of the time window is 15 minutes or even less in the real world when the proposed approach is applied to the value of time windows being less than 15 minutes.

5. Conclusion

Although artificial neural networks have the positive performance in terms of mining non-linear data with self-learning ability, stock forecasting still requires a more reliable method to integrate a precise training process into the neural networks. This study proposed a hybrid approach of the data preprocessing techniques and optimized algorithms with the multilayer feed-forward neural network trained by back-propagation algorithm to create a predictive model for enhancing the accuracy of stock prediction. Haar wavelet transform utilized to decompose the stock price time-series and eliminate noise. Directed Artificial Bee Colony algorithm, which is the improved version of ABC algorithm, was used to optimize the weights and biases for the ANN in the first stage of training process.

Though the proposed integrated system has a satisfactory predictive performance, it still has some insufficiencies. Future work tends to determine the critical impact of specific fundamental analysis variables on the quality of the stock price prediction. In addition, a more advanced pattern selection scheme might be embedded in the system to retrieve significant patterns from the data.

It can be seen that the prediction process of stock prices is usually affected by 4 factors including open, close, high, and low prices, in which the close price was used in this study with the promising results. In reality, open price is not used very often, however low and high price may be crucial, so authors will consider the using these two values in the further studies. This work employed 30-to-1 model that means using 30 recent days to forecast the next day. This value seems to work quite well, but in some different technical analysis indicators the value of 14 is commonly used for the window size. Therefore, we plan to carry out many experiments to assess the influence of window sizes on the accuracy of the stock price prediction problem.

References

[1] G. Marketos, K. Pediaditakis, Y. Theodoridis, and B. Theodoulidis, "Intelligent stock market assistant using temporal data mining", in *Proc. 10th Panhellenics Conf. Inform. PCI05*, Volos, Greece, 2005.

[2] T. H. Roh, "Forecasting the volatility of stock price index", *Expert Syst. with Appl.*, vol. 33, no. 4, pp. 916–922, 2007.

[3] Q. Yang and Y. Wu, "10 challenging problems in data mining research", *Int. J. Inform. Technol. & Decision Making*, vol. 5, no. 4, pp. 597–604, 2006.

[4] A. Adebiyi, C. Ayo, M. O. Adebiyi, and S. Otokiti, "Stock Price Prediction using Neural Network with Hybridized Market Indicators", *J. Emerg. Trends in Comput. & Inform. Sci.*, vol. 3, no. 1, pp. 1–9, 2012.

[5] I. El-Henawy, A. Kamal, H. Abdelbary, and A. Abas, "Predicting stock index using neural network combined with evolutionary computation methods", in *Proc. 7th Int. Conf. Inform. & Syst INFOS 2010*, Cairo, Egypt, 2010.

[6] T. J. Hsieh, H. F. Hsiao, and W. C. Yeh, "Forecasting stock markets using wavelet transforms and recurrent neural networks: An integrated system based on artificial bee colony algorithm", *Appli. Soft Comput.*, vol. 11, no. 2, pp. 2510–2525, 2011.

[7] E. Nourani, A. M. Rahmani, and A. H. Navin, "Forecasting stock prices using a hybrid artificial bee colony based neural network", in *Proc. Int. Conf. Innov. Manag. & Technol. Res. ICIMTR 2012*, Malacca, Malaysia, 2012, pp. 486–490.

[8] M. S. Kiran and O. Findik, "A directed artificial bee colony algorithm", *Appl. Soft Comput.*, vol. 26, pp. 454–462, 2015.

[9] J. B. Ramsey, "The contribution of wavelets to the analysis of economic and financial data", *Philosoph. Trans. Royal Soc. of London A: Mathem., Phys. & Engin. Sci.*, vol. 357, no. 1760, pp. 2593–2606, 1999.

[10] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm", *J. Global Optimiz.*, vol. 39, no. 3, pp. 459–471, 2007.

[11] S. H. Kim and S. H. Chun, "Graded forecasting using an array of bipolar predictions: application of probabilistic neural networks to a stock market index", *Int. J. Forecast.*, vol. 14, no. 3, pp. 323–337, 1998.

[12] H. N. Hao, "Notice of Retraction Short-term forecasting of stock price based on genetic-neural network", in *Proc. 6th Int. Conf. Nat. Comput. ICNC 2010*, Yantai, China, 2010, pp. 1838–1841.

[13] H. Huang, M. Pasquier, and C. Quek, "Financial market trading system with a hierarchical coevolutionary fuzzy predictive model", *IEEE Trans. Evolut. Comput.*, vol. 13, no. 1, pp. 56–70, 2009.

[14] M. E. Abdual-Salam, H. M. Abdul-Kader, and W. F. Abdel-Wahed, "Comparative study between Differential Evolution and Particle Swarm Optimization algorithms in training of feed-forward neural network for stock price prediction", in *Proc. 7th Int. Conf. Inform. & Syst. INFOS 2010*, Cairo, Egypt, 2010.

[15] K. J. Kim and I. Han, "Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index", *Expert Syst. with Appl.*, vol. 19, no. 2, pp. 125–132, 2000.

[16] J. Kishikawa and S. Tokinaga, "Realization of feature descriptive systems for clusters by using rule generations based on the genetic programming and its applications", *IEICE Trans. Fundament. of Electron., Commun. & Comp. Sci.*, vol. 89, no. 12, pp. 2627–2635, 2000.

[17] G. Cybenko, "Approximation by superpositions of a sigmoidal function", *Mathem. of Control, Sig. & Syst.*, vol. 2, no. 4, pp. 303–314, 1989.

[18] S. Shen, H. Jiang, and T. Zhang, "Stock market forecasting using machine learning algorithms", Tech. Rep., Department of Electrical Engineering Stanford University, Stanford, CA, USA, 2012.

[19] G. Orr, N. Schraudolph, and F. Cummins, "Overfitting and regularization" [Online]. Available: http://www.willamette.edu/∼gorr/classes/cs449/overfitting.html (accessed Oct. 11, 2015).

[20] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feed-forward networks are universal approximators", *Neur. Netw.*, vol. 2, no. 5, pp. 359–366, 1989.

[21] H. Takaho, T. Arai, T. Otake, and M. Tanaka, "Prediction of the next stock price using neural network for data mining", in *Proc. Int. Symp. Non-Linear Theory & its Appl. NOLTA 2002*, Xi'an, China, 2002, pp. 411–414.

[22] A. Omidi, E. Nourani, and M. Jalili, "Forecasting stock prices using financial data mining and Neural Network", in *Proc. 3rd Int. Conf. Comp. Res. & Develop. ICCRD 2011*, Shanghai, China, 2011, pp. 242–246.

[23] Yahoo Finance [Online]. Available: http://finance.yahoo.com/ (accessed Oct. 20, 2015).

Thanh Tung Khuat completed the B.Sc. degree in Software Engineering from University of Science and Technology, Danang, Vietnam, in 2014. Currently, he is participating in the research team at DATIC Laboratory, University of Science and Technology, Danang. His research interests focus on software engineering, software testing, evolutionary computation, intelligent optimization techniques and applications in software engineering.

E-mail: thanhtung09t2@gmail.com
The University of Danang
University of Science and Technology
54 Nguyen Luong Bang, Lien Chieu
Danang, Vietnam

Quang Chanh Le is a final-year student at the University of Danang, University of Science and Technology. He is doing the final-year project with the topic "Applying Artificial Neural Networks and nature-inspired algorithms for predicting the stock price".

E-mail: quangchanh.11tclc.dut@gmail.com
The University of Danang
University of Science and Technology
54 Nguyen Luong Bang, Lien Chieu
Danang, Vietnam

Bich Loan Nguyen is a final-year student at the University of Danang, University of Science and Technology. She is doing the final-year project with the topic "Applying Fuzzy Logic and nature-inspired algorithms for predicting the stock price".

E-mail: s.viva13@gmail.com
The University of Danang
University of Science and Technology
54 Nguyen Luong Bang, Lien Chieu
Danang, Vietnam

My Hanh Le is currently a lecturer of the Information Technology Faculty, University of Science and Technology, Danang, Vietnam. She gained the M.Sc. degree in 2004 and the Ph.D. degree in Computer Science at the University of Danang in 2016. Her research interests are about software testing and more generally application of heuristic techniques to problems in software engineering.

E-mail: ltmhanh@dut.udn.vn
The University of Danang
University of Science and Technology
54 Nguyen Luong Bang, Lien Chieu
Danang, Vietnam

Paper

A Cloud-aided Group RSA Scheme in Java 8 Environment and OpenStack Software

Agnieszka Jakóbik

Faculty of Physics, Mathematics and Computer Science, Tadeusz Kościuszko Cracow University of Technology, Cracow, Poland

Abstract—In this paper a RSA based security system enabling the group of users to upload the single masked message to the cloud environment is proposed. Data stored are encrypted using RSA algorithm. The data receiver is able to encrypt the message retrieved from the cloud environment using private key. Two different separate RSA systems are used. The presented approach is divided into three parts. First, an RSA key is generated for each sender. Then masking the message by newly chosen mask proposed individually by every member, additionally encrypted by individual RSA private key of each member is proceed. Next, encrypting the gathered message inside the cloud environment, using the public key of the receiver is executed. In the third step, the message is decrypted by the receiver using his private RSA key. The scheme reduces the computational load on users side and transfers calculations and storage effort to the cloud environment. The proposed algorithm was developed for storing and sending the data that originally are produced by a group of users, but the receiver of the data is single. It was implemented using Java 8 and OpenStack software. Numerical test of different key length for RSA are presented.

Keywords—cloud computing, confidentiality, RSA cryptosystem.

1. Cloud Computing

Cloud computing is based on a business model in which resources are shared among at the network, host, and application level. It provides massive scalability and the ability to store large amount of data with efficient computational power. The cloud computing offers resources such as virtual-machine disks, image libraries, file storages, firewalls, mailing systems, load balancers, IP addresses, virtual local area networks, software bundles, operating systems, programming languages execution environments, databases, and Web servers [1].

Users may access to cloud environment using client devices, such as desktop computers, laptops, tablets and smart-phones. They are thin clients because cloud services do not require dedicated software on the client side (Fig. 1). Clients' software transfers calculation effort to the machine in the cloud. The most widely used examples of cloud computing are Google Cloud, Microsoft Cloud, Amazon Cloud and Adobe Creative Cloud [2]–[5].

The following features of cloud based IT systems distinguish them from traditional services and resources:

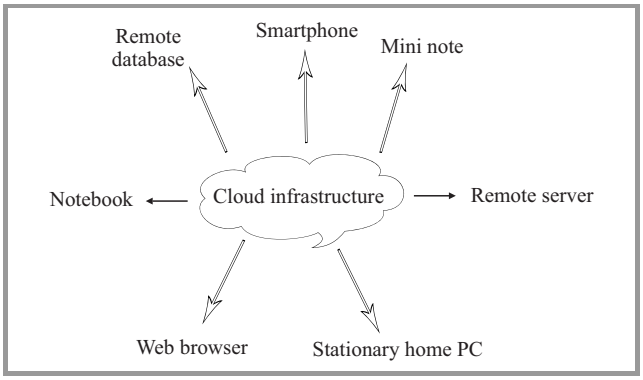


Fig. 1. Features of Cloud Systems.

- multitenancy – unlike previous computing models, which assumed single user resources in cloud model resources they are shared;
- massive scalability – traditional models might have hundreds or thousands of systems, cloud computing provides the ability to scale to tens of thousands of systems, and the ability to massively scale bandwidth and storage space;
- flexibility – users can increase and decrease their computing resources on demand;
- pay as you go – users pay only for using services and for the time they require them;
- self-provisioning of resources – users may introduce additional systems, i.e. processing capability, software, storage space and network resources [6];
- lower cost – cloud models have low upfront IT investment, and modular IT architecture. No infrastructure on the client side is required.

2. Security Issues in Cloud Environment

Cloud environment may be an easy target for hackers and organizations due to system availability and third-parity data usage. Moreover, user must protect the infrastructure by using safe connection to the cloud. Therefore, not every aspect of system security is fully controlled by cloud environment. The service provider have to deliver the

authentication and authorization techniques. Individuals may be assigned to different levels of privileges or may act together sharing the resources. Data and services in the clouds may be affected by numbers of attacks from distributed denial of service, phishing, data loss or leakage, unauthorized sharing of the technology and the abuse of the resources [6]. The security objectives are a key factor for decisions about choosing the cloud vendor. Nowadays, a lot of outsourcing information techniques, services and applications, and other resources has to be located inside a cloud computing environment, due to data size, speed and diversification. The user of the data benefits from procedures implemented inside the cloud. These may include given tools for authorization, authentication or data security support technique.

Cloud platform offers Infrastructure as a Service (IaaS), Platform as a service (PaaS) and Software as a service (SaaS). All three service models may be a target for the threats [7].
Infrastructure Security at the network level requires:

- ensuring the confidentiality and integrity of user data in transit to and from public cloud provider,
- ensuring proper access control (authentication, authorization, and auditing) to resources,
- ensuring the availability of the Internet resources in a public cloud that are being used by user or have been assigned to user by cloud provider [6].

The proposed scheme may be used in two first above levels.
Infrastructure security at the host level requires:

- for PaaS and SaaS models – hiding the host operating system from end users,
- the abstraction layer is not visible to users and is available only to the developers,
- host security responsibilities in SaaS and PaaS services are transferred to the cloud model [6].

The proposed scheme fulfills above requirements.
The data in cloud models should be treated deferentially according their status: data in transit, data at rest, data being processed. Additionally, multitenancy, data lineage, data provenance, data remanence have to be considered. Data in transit should be encrypted during transfer to and from a cloud provider. Encrypting data at rest is strongly suggested. Data lineage is important for an auditor's assurance. Integrity of data ensures that data has not been changed in an unauthorized manner or by an unauthorized person. Provenance means that the data is computationally accurate. Data remanence refers to the policy of treating data that are not used, not actual or has been erased by user from his applications [6]. In the proposed scheme data in transit are treated differently from data in rest, data lineage is strictly defined, processing of the data and data provenance are supported only by cloud software. Data remanence depends on both: user and cloud infrastructure. Data at rest stored for a long time inside cloud infrastructure should be encrypted using strong cryptography.

3. OpenStack Software

OpenStack is a cloud operating system that enables computing, storing, and networking resources [8]. It is open-source software for IaaS. Managing the IT infrastructure for the proposed algorithm was possible by using communication interface. Virtualized resources were used for calculating the results. They were available by prepared GUI environment. It consisted Compute (Nova service) module responsible for arranging, managing and providing virtual machines. Prepared working environment provided massively scalable, on demand access to computing resources. Object Store (Swift service) was used for managing storage system and Image Service (Glance service) was responsible for uploading and discovering data. Services were manageable from Horizon dashboard.

4. RSA Cryptosystem

RSA (proposed by Ron Rivest, Adi Shamir and Leonard Adleman [9]) is the public key algorithm for encrypting and decrypting the data. Encryption enables the communication to be private. Each message is encrypted before transmitting it to the receiver. The receiver knows the proper function to obtain the original message. Only the authorized user can have an access to the data stored. RSA algorithm may be used also for digital signatures evaluating [10]. Every plain text written in natural language is mapped into integer number and encrypted using arithmetic algorithm that proceeds on big numbers. RSA algorithm involves three stages:

- Key generation starts from choosing two different large random prime numbers p and q. Then, calculating $n = *pq$ the modulus for the public key and the private keys. Next step is to calculate the totient: $\phi(n) = (p-1)(q-1)$. After that one have to choose an integer e such that $1 < e < \phi(n)$ and e, is co-prime to e, i.e.: e, and $\phi(n)$ share no factors other than 1; $gcd(e, \phi(n)) = 1$. The public key is made of the modulus n and the public exponent e. The private key is made of the modulus n and the private exponent d, which must be kept secret;
- To encrypt the message M for particular receiver the sender is using public key of receiver. First the sender turns M into a number m (smaller than n) by using a reversible protocol known as a padding scheme [11]. He then computes the cipher text c:
$$c = m^e \mod n; \quad (1)$$
- The receiver can recover m, from c, by using his private key d:
$$m = c^d \mod n. \quad (2)$$
Given m one can recover the original message M by applying the reverse padding scheme.

It is currently recommended that n should be 1024 or 2048 bits long [12]. The RSA algorithm is used for example as a part of security system inside Google Cloud [13], [14], and in many handshaking protocols in the Transport Layer Security (TLS) [15].

5. Proposed Cloud-aided Group RSA Scheme

Proposed cloud-aided group RSA scheme is based on the procedure [16]. Additional stage was added to secure the first step of the algorithm before data are send to the cloud computing center. Moreover, proposed algorithm enables sending not only the single message (as in [16]) but the message may be composed from different parts coming from different senders. Proposed algorithm was designed for two separate data centers. Separating responsibilities increased the security of the cloud infrastructure itself. Furthermore, it made controlling the users privileges much easier.

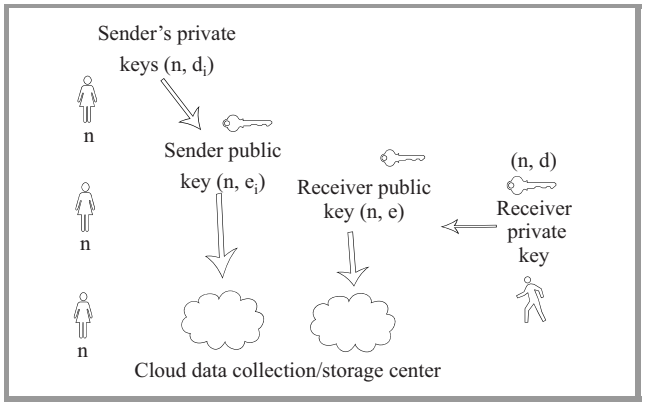


Fig. 2. Key preparation during stage 0.

5.1. Stage 0 – Organizing Infrastructure

Let's assume that a group of users (called senders) $S_1, S_2, \ldots, S_T$ is sharing the same file or data in the cloud infrastructure. They may modify and send it to receiver chosen from outside this group. All parts are gathered to portion of data called a message m in the cloud environment so that the receiver R may retrieve it on demand. The message is stored in the cloud storage center (CSC) in the encrypted form. Four actors are considered: group of senders, cloud data collection center (CDCC), cloud data storage center (CDSC), receiver of the data. The group of senders contacts only with CDCC, receiver only with CDSC. CDCC and CDSC are contacting each other (Fig. 2).

5.2. Stage 1 – Key Preparation and Data Gathering

The receiver generates his RSA private key (d,n) and public RSA key (e,n) and sends public RSA key to the CDSC. The CDSC sends to CDCC parameters n and e. CDCC sends parameter n to each sender. Each sender $i = 1, 2, \ldots, T$ generates the own private RSA key (d_i,n), and public RSA key (e,n). Then he sends public RSA key to the CDCC. Only the public keys are transferred. Then, sending decrypted data to CDCC is made. To do so, each sender is passing his part m_i of the message $m = m_1 * m_2 * \cdots * m_T$. He generates new random number a_i such that $a < n$ and computes $b_i = (m_i * a_i)^{d_i} \mod n$. Then he sends b_i to CDCC. Additionally $a_0 = a_1 * a_2 * \cdots * a_T \mod n$ is added. In case only one member is sending his message, all the a_i but his are set to 1 (Fig. 3).

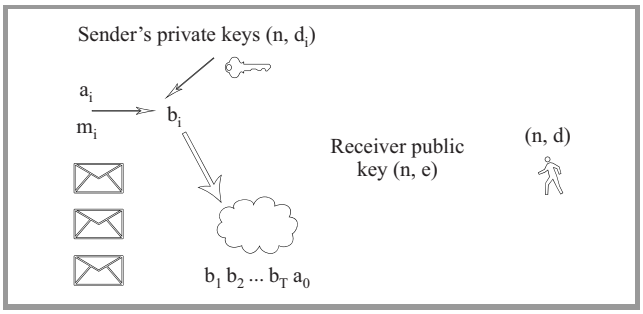


Fig. 3. Gathering messages from recipients inside CDCC during stage 1.

5.3. Stage 2 – Processing Data in CDCC

CDCC applies public key of each sender e_i to proper part of data and then applies public key of the receiver e:

$$v_0 = {a_0}^e, \tag{3}$$

$$v_i = (b_i^{e_i})^e \mod n =, \tag{4}$$

$$\left((m_i * a_i)^{d_i * e_i}\right)^e \mod n = (m_i * a_i)^e \mod n. \tag{5}$$

After that stage, data are composed into $V = (v_0, v_1, v_2, \ldots, v_T)$ vector. Then this vector is permuted by using random permutation P: $V_{perm} = (v'_0, v'_1, v'_2, \ldots, v'_T)$. Vector V_{perm} is sent to CDSC. The data from the group of senders is located in CDSC (Fig. 4).

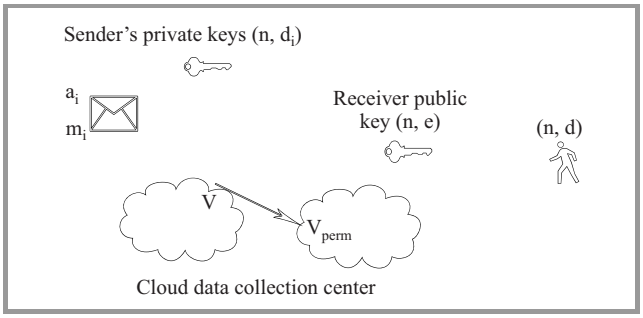


Fig. 4. Mixing messages from recipients inside CDCC into single message during stage 2.

5.4. Stage 3 – Processing Data in CDSC

CDSC applies inverse permutation P^{-1} to V_{perm}, obtaining $V = (v_0, v_2, \ldots, v_T)$ vector. Then computes:

$$c = (v_1 * \cdots * v_T) * v_0{}^{-1} \mod n = , \quad (6)$$

$$(m_1 * m_2 * \cdots * m_T)^e \mod n = , \quad (7)$$

$$m^e \mod n . \quad (8)$$

This value is stored until the receiver would like to retrieve it (Fig. 5).

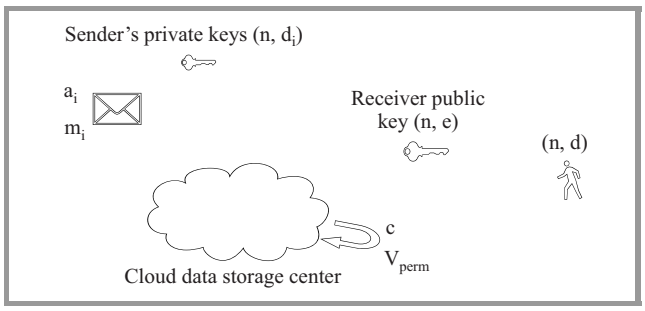


Fig. 5. Storing data inside CDSC in the form of single message during stage 3.

5.5. Stage 4 – Downloading Decrypted Data to CDSC and Encryption by Receiver

Receiver uses his own private key d to calculate:

$$m = c^d \mod n = (m^e)^d \mod n = m \mod n . \quad (9)$$

Given m one can recover the original message M by using reverse padding scheme.

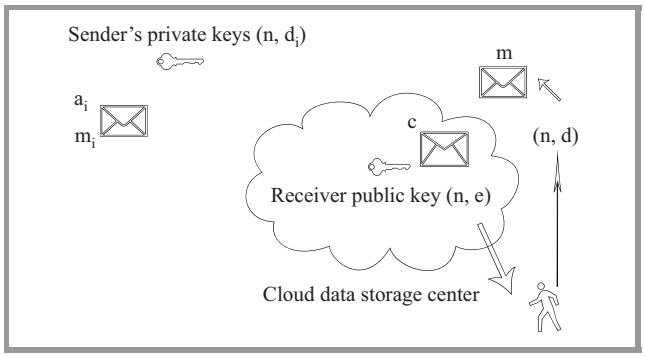


Fig. 6. Retrieving message from CDCC and encryption during stage 4.

Data is more vulnerable to unauthorized modification or appropriation when it is in storage than it is being processed. This is the reason, why the strongest RSA keys should be applied during stages 2 and 3. It also moves storage afford the data into cloud environment. Masks $a_1, a_2, \ldots, a_T$ are different for each message. There is no need for generating new RSA key for every single message. Using the masks, the true message m_i is indistinguishably from empty message m_i during sending stage 1. If particular sender is not generating his part computational afford is low, because masks may be much smaller then messages. The group is invisible to the receiver. The receiver does not know who from the group had sent the particular part of the message or even that the message was composed from parts (Fig. 6).

6. Experimental Results in Java 8 Environment and OpenStack Software

CDCC operation were calculated using several units of PCs connected into one network. Such a solution enables gathering the data from users by independent units. Additionally, the units were used as direct way of the data uploading. The uploading the data for lean clients was made by Web browser application running on each unit. Cloud data storage operations were performed using single OpenStack instance [17].

Java 8 environment was chosen for three main reasons. Firstly, for the convenience of the security package. Secondly, for the possibility of usage the BigInteger library that was necessary for computing on such large numbers. Finally, Java 8 provided secure pseudo-random number generator for calculating e, p and q values.

The RSA part of the scheme was implemented using the following Java 8 classes, methods and interfaces from Java security package:

- Key – interface models the base characteristics for the keys,
- KeyFactory – key factories are used to convert keys into key specifications (transparent representations of the underlying key,
- KeyPair – serves as a container for public and private keys,
- KeyPairGenerator – a class used to generate key-pairs for a security algorithm,
- GeneratePrivate – method generating a private key from the provided key specification,
- GeneratePublic – method generating a public key from the provided key specification,
- Interface KeySpec – transparent specification of the key that sets a cryptographic key. If the key is stored on a hardware device, its specification may contain information that helps identify the key on the device [18].

To generate the keys from the Key Factory the following Java 8 code was implemented:

```
KeyPairGenerator key_par_gen =
  KeyPairGenerator.getInstance("RSA");
key_par_gen.initialize(2048);
KeyPair kp =
  key_par_gen.genKeyPair();
PublicKey publicKey =
  key_par.getPublic();
PrivateKey privateKey =
  key_par.getPrivate();
KeyFactory factory_rsa =
  KeyFactory.getInstance("RSA");
RSAPublicKeySpec pub =
  factory_rsa.getKeySpec
   (publicKey, RSAPublicKeySpec.class);
RSAPrivateKeySpec priv = factory_rsa.
  getKeySpec
   (privateKey,RSAPrivateKeySpec.class);
```

To retrieve the values of the key the following Java 8 code was proposed:

```
BigInteger
n = (BigInteger)
object_input_stream.readObject();
BigInteger
e = (BigInteger)
object_input_stream.readObject();
KeyFactory factory_rsa =
KeyFactory.getInstance(``RSA");
if (keyFileName.startsWith(``public"))
 return factory_rsa.
 generatePublic
 (new RSAPublicKeySpec(n, e));
else
 return factory_rsa.
 generatePrivate
 (new RSAPrivateKeySpec(n, d));
```

Public class called *Cipher* was used to encrypt and decrypt the messages during stage 1 [19].

- javax.crypto.Cipher – class provides encryption and decryption;
- javax.crypto.CipherInputStream – constructs a CipherInputStream from an InputStream and a cipher;
- javax.crypto.CipherOutputStream – constructs a CipherOutputStream from an OutputStream and a cipher.

Keys were stored using secured disk space. There were retrieved on demand when new instance of RSA algorithm was created:

```
Key public_Key =
readKeyFromFile("public.key");
    Cipher cipher =
    Cipher.getInstance("RSA");
cipher.init
  (Cipher.ENCRYPT_MODE, public_Key);

 Key private_Key =
 readKeyFromFile("private.key");
   Cipher cipher =
   Cipher.getInstance("RSA");
cipher.init
(Cipher.DECRYPT_MODE, private_Key);
```

7. Numerical Tests

Two physically separated data centers were used. Cloud data collection center was located in Institute of Computer Science at Cracow University of Technology in Poland, and Cloud data storage center was located in Cloud Competency Center at National College of Ireland [20].

Cloud data collection center consisted of 28 units of the same characteristic: AMD FX-6300 6-cores processor, 2 GHz, cache size 2048 KB, Windows 8.1 Enterprise or Linux Fedora rel. 22. This solution enables uploading the data remotely using Web application and mobile phones or tablets. Additionally 28 users may upload their data in the same time personally while sitting in front of the units. The CDCC used Java 8 Web application, described in Section 5. The CDSC was working under OpenStack software. Single instance was used. The OpenStack instance m1.large was configured as follows: RAM 8 GB, 4 VCPU, 80 GB HDD [21], with Ubuntu operating system and OpenSSL library [22]. The experimental server specified in [23] was used. The characteristics of the virtual machine used for calculation were: OpenStack Nova system, memory 96 KB BIOS, processor Intel Xeon E312xx (Sandy Bridge), 8 GB system memory. The decryption was made by data storage center on users demand. The data was sent to the end user via secure channel.

Table 1
Mean encryption time in CDCC, data file size 10 MB

Key length	1024 bit	2048 bit
Real time	1 m 56.112 s	2 m 25.334 s
User time	1 m 17.357 s	2 m 21.434 s
System time	1 m 14.711 s	2 m 20.724 s

Table 2
Mean decryption time in CDSC OpenStack instance, data file size 10 MB

Key length	1024 bit	2048 bit
Real time	0 m 31.484 s	0 m 45.572 s
User time	0 m 25.984 s	0 m 37.245 s
System time	0 m 26.226 s	0 m 40.378 s

According to the recommendation for key management [24], secure 1024 or 2048 bit keys were used. The chosen results of experiments are presented in Tables 1 and 2.

It was found profitable to proceed long messages in chunks. Chunks were generated by dividing the message into equal parts. Specified number of bits was considered. Chunks was coded and decoded in parallel mode. The time of calculation is highly dependable on the busyness of the computational unit. That is why, mean encryption time in CDCC unit is presented.

It was stated that enlarging key length from 1024 to 2048 bits resulted in computational time increasing about:

- 34% for Data Collection Unit real time,
- 30% for OpenStack instance real time.

Instead of multiplication of fragmentary messages other operations may be performed, then equivalent methods for generating a_0 have to be adopted.

8. Conclusions and Future Development

Proposed RSA based security system enables the group of users sending the single message to the cloud environment. Inside cloud environment data are stored encrypted by RSA algorithm. Receiver of the data is able to encrypt the message retrieved from the cloud environment on demand, regardless when message was send. Two different separate RSA systems are used according to the computing capabilities. Simple, computationally non-demanding procedure was proposed for a thin client in the form of masking. Parallel computations using strong secure keys were incorporated for cloud environment. The RSA key generating and masking the message was proposed for each sender. Such a procedure enabled treating data at rest and data being processed in a different way, according to the security requirements. It also allowed minimizing computational afford on the user's side and using the benefits of cloud computing security infrastructure.
Separating senders from receivers allowed dividing thin and not secured clients from data store center. Grouping senders and receivers in different location simplified the management of users' rights and privileges.
Additionally, encrypting the gathered message inside the cloud environment enables to hide the group from the receiver. Such a scheme may be used during electronic voting, electronic reviewing, and team working on the same document. Decryption the message by the receiver using his private RSA key ensures that if the private key was kept secret no one but receiver could read the massage stored inside cloud environment.
Future development of research is planned. The investigation on the influence of differed OpenStack instances types is considered. Also, exploring more advanced methods for lean clients. The masking method is planned to be replaced by fragmenting and mixing the message. Further investigation on increasing computational efficiency is necessary.

Acknowledgement

The author would like to express great gratitude for Horacio González-Véle, the Head of NCI Cloud Competency Center in Dublin for enabling tests on OpenStack software.

References

[1] J. Rittinghouse and J. Ransome, *Cloud Computing: Implementation, Management, and Security*. CRC Press, 2009.

[2] Google Cloud Platform [Online]. Available: https://cloud.google.com/

[3] Amazon Drive [Online]. Available: https://www.amazon.com/clouddrive/home

[4] Microsoft Cloud [Online]. Available: http://www.microsoft.com/enterprise/microsoftcloud

[5] Adobe Creative Cloud [Online]. Available: http://www.adobe.com/pl/creativecloud.html

[6] T. Mather, S. Kumaraswamy, and S. Latif, *Cloud Security and Privacy An Enterprise Perspective on Risks and Compliance*. O'Reilly Media, 2009.

[7] W. Jansen and T. Grance, "Guidelines on security and privacy in public cloud computing", 2011 [Online]. Available: http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-144.pdf

[8] OpenStack website, https://www.openstack.org

[9] R. L. Rivest, "Cryptography", in *Handbook of Theoretical Computer Science*, J. Van Leeuwen, Ed. MIT Press, 1990, vol. A, pp.717–755.

[10] R. L. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining digital signatures and public-key cryptosystems", *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[11] PKCS #1 RSA ver. 2.2 Cryptography standard, RSA Laboratories, Oct. 2012 [Online]. Available: http://www.emc.com/collateral/white-papers/h11300-pkcs-1v2-2-rsa-cryptography-standard-wp.pdf

[12] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*. CRC Press, 1996.

[13] Google Cloud Documentation [Online]. Available: https://cloud.google.com/storage/docs/authentication

[14] Amazon Elastic Compute Cloud User Guide for Linux [Online]. Available: http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html

[15] T. Dierks and E. Rescorla, "The Transport Layer Security (TLS) Protocol, Version 1.2", Network Working Group, Aug. 2008 [Online]. Available: http://tools.ietf.org/html/rfc5246

[16] C.-H. Lin, C.-Y. Lee, and T.-W. Wu, "A cloud-aided RSA signature scheme for sealing and storing the digital evidences in computer forensics", *Int. J. Secur. & and Its Appl.*, vol. 6, no. 2, 2012.

[17] OpenStack Website, https://www.openstack.org/

[18] Java Platform Standard Edition 8 Documentation, Oracle, 2015 [Online]. Available: https://docs.oracle.com/javase/8/docs

[19] Java Platform, Standard Edition 8 API Specification, Oracle, 2015 [Online]. Available: https://docs.oracle.com/javase/8/docs/api/javax/crypto/Cipher.html

[20] National College of Ireland Website, https://www.ncirl.ie/

[21] D. Grzonka, "The analysis of OpenStack cloud platform: features and performance", *J. Telecommun. Inform. Technol.*, no. 3, pp. 52–57, 2015.

[22] OpenSSL, Cryptography and SSL/TLS Toolkit [Online]. Available: https://www.openssl.org

[23] D. Grzonka, M. Szczygieł, A. Bernasiewicz, A. Wilczyński, and M. Liszka, "Short analysis of implementation and recources utilization for the OpenStack cloud computing platform", in *Proc. 29th Eur. Conf. Modelling & Simul. ECMS 2015*, Albena (Varna), Bulgaria, 2015.

[24] E. B. Barker *et al.*, "Recommendation for Key Management, Part 1: General (Revised)", NIST Special Publication 800-57, National Institute of Standards & Technology, Mar. 2007 [Online]. Available: http://csrc.nist.gov/publications/nistpubs/800-57/sp800-57-Part1-revised2_Mar08-2007.pdf

Agnieszka Jakóbik (Krok) received her M.Sc. in the field of stochastic processes at the Jagiellonian University, Cracow, Poland and Ph.D. degree in the field of neural networks at Tadeusz Kosciuszko Cracow University of Technology, Poland, in 2003 and 2007, respectively. From 2009 she is an Assistant Professor at Faculty of Physics, Mathematics and Computer Science, Tadeusz Kościuszko Cracow University of Technology. Her main scientific and didactic interests are focused mainly on artificial intelligence: artificial neural networks, genetic algorithms, and additionally on parallel processing and cryptography.

E-mail: agneskrok@gmail.com
Faculty of Physics, Mathematics and Computer Science
Tadeusz Kościuszko Cracow University of Technology
Warszawska st 24
31-155 Cracow, Poland

Paper

Graph-based Forensic Analysis of Web Honeypot

Hudan Studiawan, Supeno Djanali, and Baskoro Adi Pratomo

Department of Informatics, Institut Teknologi Sepuluh Nopember (ITS), Surabaya, Indonesia

Abstract—Honeypot still plays an important role in network security, especially in analyzing attack type and defining attacker patterns. Previous research has mainly focused on detecting attack pattern while categorization of type has not yet been-comprehensively discussed. Nowadays, the web application is the most common and popular way for users to gather information, but it also invites attackers to assault the system. Therefore, deployment of a web honeypot is important and its forensic analysis is urgently required. In this paper, authors propose attack type analysis from web honeypot log for forensic purposes. Every log is represented as a vertex in a graph. Then a custom agglomerative clustering to categorize attack type based on PHP-IDS rules is deployed. A visualization of large graphs is also provided since the actual logs contain tens of thousands of rows of records. The experimental results show that the proposed model can help forensic investigators examine a web honeypot log more precisely.

Keywords—access log, attack type, graph agglomerative clustering, visualization of large graphs.

1. Introduction

A honeypot is a system that lets attackers hack into it and records all activities so that their behavior can be observed [1]. There are several types of honeypots based on various services for example web, SSH, database, and network traffic or using the level of interaction or the responses given to the attacker, i.e. low, medium, and high [1]. The use of honeypots for forensic analysis was first developed in 2002 [2]. The researchers proposed a new forensic model and two architectures for honeypots, serial and parallel. In [3] the researcher tried to find the root cause of attack in a honeypot and the association rules were deployed to detect suspicious activity.

Forensic analysis can be conducted of both network traffic [4] and the compromised host [5] to get more detailed information from honeypot logs. More advanced techniques to review honeypot capabilities were introduced by using file system journaling [6]. It provided deeper analysis through file system abstraction to achieve meta data archiving. Forensic investigators are also able to generate attack statistics reports from honeypot deployment as observed in [7], [8]. Riebach *et al.* provided a case study of honeypot deployment to support forensic analysis and gave more attention to worm activities and classical multi-phase attack [9].

The investigation can be conducted in real time, usually called live forensic, as designed in [10] and the author focused on HTTP, FTP, POP3, and telnet protocols. Virtualization technology, cheaper than physical infrastructure could be used to deploy honeypots and supply real time forensics [11]. Attacker pattern was deeply investigated in [12] by clustering technique and time series analysis. The honeypot could be used for army of zombies detection [13]. They supplied an accurate observation of a botnet attack from honeypot trace. The other study of network forensics based on honeypot has been comprehensively described in [14]. They provided a survey, comparison, and future directions for this research area. Another work to provide honeypot for production mode of web application has been proposed by Pohl *et al.* since most honeypots only provide service in a non-productive environment [15]. The implementation of many honeypot platforms is also presented in [16]. However, these previous works still have not provided any interactive, collaborative, or real tools to model and visualize a web honeypot log.

Valli presented a visualization of honeypot data based on graph theory using Graphviz library and exploited AfterGlow for generating link graphs [17]. However, it could not provide interactive display for forensic investigators although could give large graph visualization. A recent study by Cabaj gives a graph visualization to assist in data analysis generated by honeypots [18]. The paper described visualization of the Honeypot Management System (HPMS) deployed at Institute of Computer Science, Warsaw University of Technology, but unfortunately this work only handled an attack to phpMyAdmin.

In this paper, a model for web honeypot logs based on graph theory is proposed. The authors will adopt the graph model to represent a log, and clustering technique will be deployed to analyze attacks for forensic purposes. The forensic investigation will be covered, which needs to know the origin and the type of penetration, e.g., cross site scripting or remote file inclusion. The visualization of proposed method will relax the one proposed by Cabaj [18].

The rest of this paper is organized as follows. Section 2 briefly describes the proposed log model based on graph theory, rule-based attack detection technique, and visualization of generated large graph. Section 3 explains experimental results and its analysis. Finally, conclusion and future works are discussed in Section 4.

2. Proposed Method

2.1. Graph-based Agglomerative Cluster

The main evidence for forensic analysis is the raw access log containing every request to the web honeypot. This

artifact was provided by Honeynet Project [19]. Authors then define graph $G = (V, E)$ to represent all access logs, propose custom agglomerative clustering on the set of vertices V, and generate edges E during the clustering process. The similarity measurement for clustering is used based on the node's attributes value, i.e., attack type, attacker's IP address, origin city and country. This gives an exact similarity since these properties are standardized for all vertices. Therefore, there are six levels of hierarchy in the resulting cluster, i.e., one level for leaf vertex representing the raw request and five upper-levels of intermediate nodes, which represent cluster root of attack type, IP address, city, country, and top-level cluster of the main graph used for overall forensic analysis. Cluster C as a subgraph of G is defined as:

$$C = \{C_{l_i} \mid C \subset G, l = 0, 1, \ldots 5 \text{ and } i = 0, 1, \ldots, p\}, \quad (1)$$

where l is level of hierarchy and p is a set of the total number of clusters in specific level l as shown in Fig. 1 and every level is represented by a different color. Meanwhile, the node size also depicts level in graph – the deeper the level, the smaller its size.

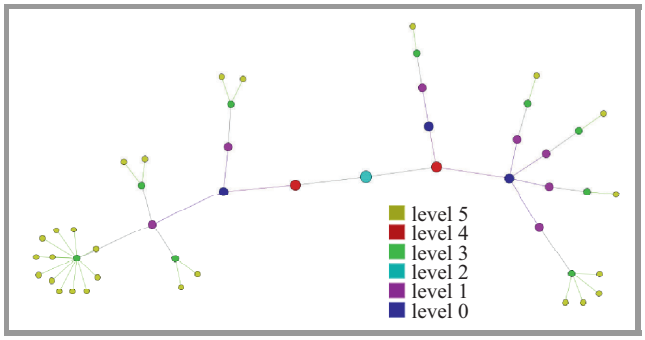


Fig. 1. Illustration of hierarchy level in proposed method. (See color pictures online at www.nit.eu/publications/journal-jtit)

In addition, by V_{l_j} the set of vertices of G in cluster level l is denoted:

$$V = \{v_{0_j}, v_{1_j}, \ldots, v_{l_m}\}, \quad (2)$$

where $j = 0, 1, \ldots, m$ and m is the total number of logs plus intermediate nodes created during clustering in level l. These vertices store every record and its attributes including id, attacker's remote address, time stamp, raw request, referrer, user agent, origin city and country, attack type and its description. Thus, some flags are also maintained as node's attributes in order to handle the clustering process, i.e., is_attackroot, is_iproot, is_cityroot, is_countryroot, and is_mainroot. These flags also distinguish whether or not a vertex is a root of a particular cluster.

Furthermore, there are six types of edges based on cluster level, E_l. These edges connect vertices to the intermediate node in each cluster and are defined as follows:

$$E = \{E_{0_k}, E_{1_k}, \ldots, E_{l_n}\} \quad (3)$$

where $k = 0, 1, \ldots, n$ and n is total number of cluster C_{l_i}. For each E_{l_k}, one can see an edge as a tuple consisting of a vertex and its connected cluster root:

$$E_{l_k} = \{(v_{0_j}, R_{0_i}), (v_{1_j}, R_{1_i}), \ldots, (v_{l_j}, R_{l_i})\}, \quad (4)$$

where variable i, j, and l have been described in previous equations.

Before creating edges E, all of the vertices based on IP address and attack type attributes are clustered. For each cluster, the procedure creates one vertex as an intermediate node acting as cluster root R. In other words, all vertices except v_m are an intermediate node. Formally,

$$R = V \setminus \{v_{x_j}\}, \; x = 0, 1, \ldots, m-1. \quad (5)$$

After that, every vertex in cluster C_{l_i} will have an edge to the root $R_{(l-1)_i}$ so that E_{l_k} is a set of edges in C_{l_i}. The first created R will act as R_{4_i}, root of cluster by attack type, and every R_4 has an edge to R_3, root of cluster by attacker's IP address. This step will provide E_4 and generates all C_{4_i} and its R_{4_i}. In this way, investigators can easily examine how many and what type of attacks were attempted from one IP address. The illustration of how to cluster vertices based on attack type and IP address are given in Figs. 2 and 3, respectively, where the vertices with the same attribute have the same color. In these figures, only 26 records consisting of two detected attacks (cross-site scripting and local file inclusion), seven IP addresses, three cities (Budapest, Seoul, and Osan), and two countries (Hungary and Korea) from the first file of the Honeynet Project

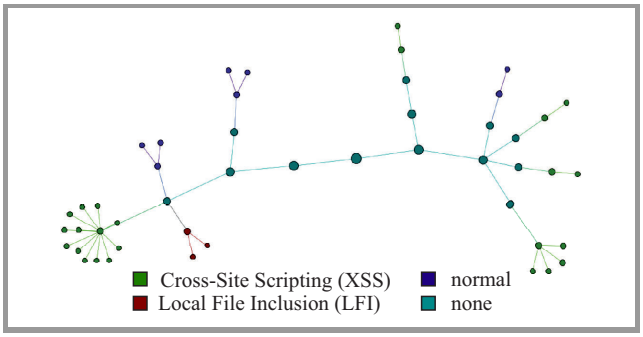


Fig. 2. Cluster by attack type.

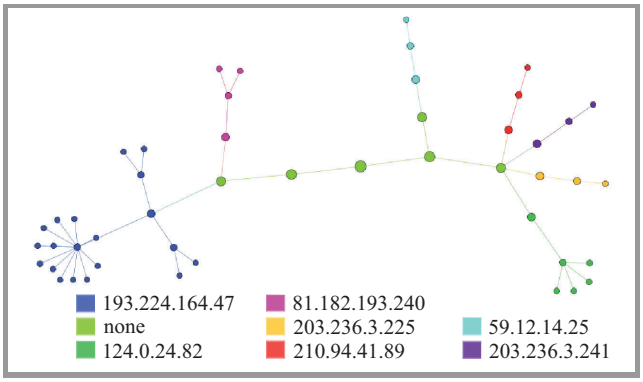


Fig. 3. Cluster by IP address.

dataset [19] in order to reduce the complexity of visualization are used.

The clustering continues to agglomerate existing group C_3 based on the origin city and then country of attacker. This geographic information is based on IP address and can be retrieved using the open GeoIP API module provided by MaxMind [20] and its Python bindings [21] for easy implementation. This procedure will produce cluster C_2, cluster by city, and C_1, cluster by country. The edges are drawn from every vertex in C_2 and C_1 to their respective intermediate root R so that we now have E_1 and E_2. Figure 4 provides clustering based on city data and the illustration of cluster by country can be easily inferred from Fig. 4. All illustrations in this section use the Yifan Hu graph layout [22] implemented in Gephi graph editor [23]. The more complex graph drawing will be explained in the next subsection.

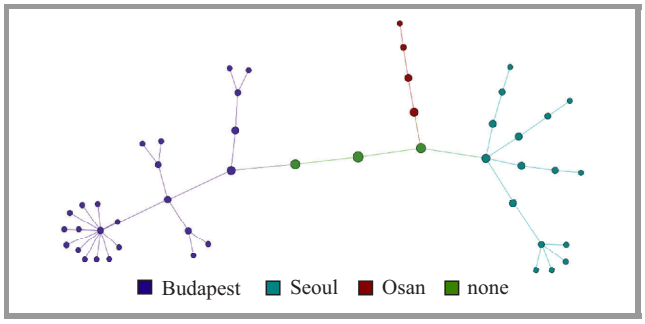


Fig. 4. Cluster by city.

In the last iteration, all separate clusters C_1 are connected to the main root R_0, resulting in both sets of edges E_0 and the cluster C_0. Through the clustering, each root flag is set to True according to processed level. This bottom-up approach provides a complete graph for forensic analysis and understanding the origin of attacks. The proposed method also provides a natural hierarchical structure to assist forensic investigators to understand attacker's behavior.

2.2. Attack Type Detection Using PHP-IDS Rules

The authors match every request log with the PHP-Intrusion Detection System (PHP-IDS) [24]. PHP-IDS rules currently contain 78 filters categorized to nine attack types: cross-site scripting (XSS), SQL injection (SQLi), cross-site request forgery (CSRF), denial of service (DoS), directory traversal (DT), spam, information disclosure (ID), remote file execution (RFE), and local file inclusion (LFI). Originally, these filters are utilized to check whether or not a request is suspicious in a PHP-based web application and it is installed and preconfigured inside the application.

However, instead of using PHP-IDS to detect attack in *ante mortem* conditions, the rules to assist forensic investigators are adopted, when examining the type of attack in *post mortem* fashion. PHP-IDS filters contain a sequence of regular expression (regex), which is developed and maintained periodically by the community. The examples of PHP-IDS filters are given in Table 1 [24].

Table 1
Example of PHP-IDS filters

No.	Filter and description	Tag
1	`(?:%u(?:ff|00|e\d)\w\w)|(?:(?:%(?:e\w|c[^3\W]|))(?:%\w\w)(?:%\w\w)?)` Detects halfwidth/fullwidth encoded unicode HTML breaking attempts	XSS
2	`(?:(?:[;]+|(<[?%](?:php)?)).*[^\w](?:echo|print|print_r|var_dump|[fp]open))|(?:;\s*rm\s+-\w+\s+)|(?:;.*{.*\$\w+\s*=)|(?:\$\w+\s*\[\]\s*=\s*)` Detects code injection attempts	RFE
3	`(?:%c0%ae\/)|(?:(?:\/|\\)(home|conf|usr|etc|proc|opt|s?bin|local|dev|tmp|kern|[br]oot|sys|system|windows|winnt|program|%[a-z_-]{3,}%)(?:\/|\\))|(?:(?:\/|\\)inetpub|localstart\.asp|boot\.ini)` Detects specific directory and path traversal	LFI

To implement the filters in test environment, an Apache Scalp, a Python implementation of PHP-IDS [25] updated using the newest PHP-IDS rules is deployed. Every filter can be attached to one or more attack types (tag) but only the first-found one using non-exhaustive mode in Apache Scalp is included. The procedure first parses the raw request, detects the HTTP method used, and then compares the request line to every regular expression in predefined filters. It returns the attack type for each attacker request to the web honeypot server.

2.3. Large Graph Visualization for Attack Type Analysis

The authors use Gephi [23] as graph editor and OpenOrd [26] as large graph layout since there are tens of thousands of logs to process and the existing typical layout can not well visualize large graph generated. Previous visualization by Cabaj [18] only displays a small portion of the attack where the vertices represent attacker, malware filename, and malware server. The display only showed a small part of the graph so the investigator may not see the entire attack attempts.

Presented work improves upon the one from [18] where the visualization is created as a tool to help forensic investigators inspect the attack type and the attacker location comprehensively using a large graph layout. An example of overall graph is shown in Fig. 5 where each color represents an attacks type: cross-site scripting (light green), remote file execution (purple), and local file inclusion (dark green) [19]. Normal means the request is not malicious and None is for an intermediate node for clustering. One can see in Fig. 5 that this visualization model will easily help the investigator to view, check, and analyze both normal requests and attempted attacks to the web honeypot as a whole. One can not clearly see the local file inclusion attack since it has only a very small number compared to overall records.

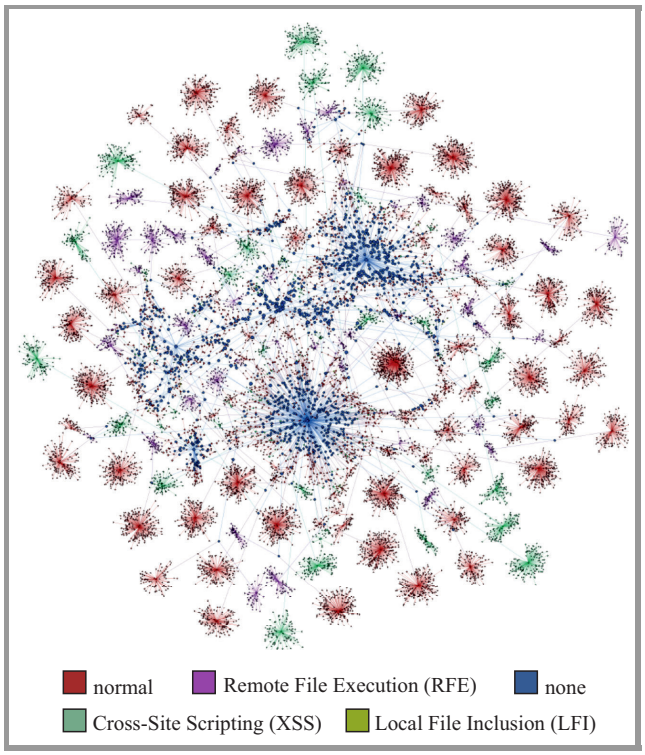


Fig. 5. Generated graph *G* from Honeynet Project dataset.

To enable the collaboration and interactive graph, the network graph from Gephi to Sigma.js was exported and hosted it a in Sigma.js server so that the investigator can access this visualization using a web interface. Sigma.js is a JavaScript library based on Node.js that is specially designed for interactive graph drawing [27]. The interactive mode means that the user is provided with a clickable vertex and cluster of vertices. This technique completes the analysis provided by large graph visualization with detailed examination of each request by displaying more detailed information when a node is clicked.

3. Experimental Results

The dataset used in this experiment is taken from the Honeynet Project [19], which is a real-life log from honeypot deployed in 2006. The complete dataset contains all logs in the /var/log directory from Linux Fedora operating system. As the authors focused on web honeypot, only the access_log file from /var/log/httpd has been taken into account. There are 32 access_log files consisting of 31 archives and one recent log containing 14,398 lines of raw requests, and we examined only the last one. Every line in the log file will be parsed and each entity becomes vertex's attributes as described in Subsection 2.1.

To manage and implement the graph, the Python-igraph [28] was used since it is fast, community-supported, and open source. It is also designed for efficiency, portability, and deployment-friendly. Apache Scalp [25] is employed to detect attack type in every request log.

The output from graph implementation using Python-igraph is a GraphML file, a XML-based format for graphs. This file is processed using Gephi to add color and provide clear and precise layout. The processed graph is then exported to Sigma.js code using Sigma.js exporter plugin in Gephi [29] and the resulting network is configured in Sigma.js server. This action will enable collaborative and interactive visualization between forensic investigators using web-based interface.

The graph *G* produced from Subsection 2.1 (Fig. 5), which is exported to Sigma.js server, is depicted in Fig. 6. When an investigator clicks a vertex, this tool will show detailed information about all attributes. It can be zoomed and slid smoothly to view the node accurately as shown in Fig. 7.

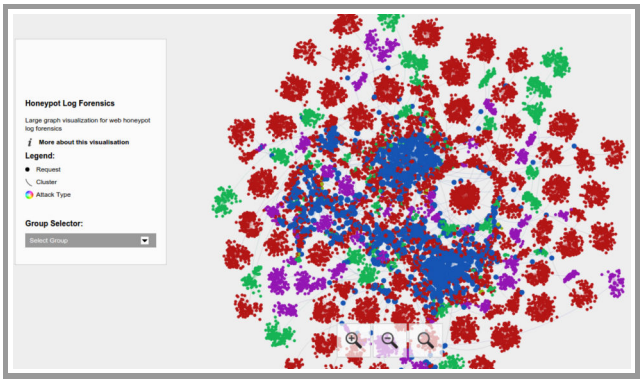


Fig. 6. Large graph visualization in Sigma.js web interface.

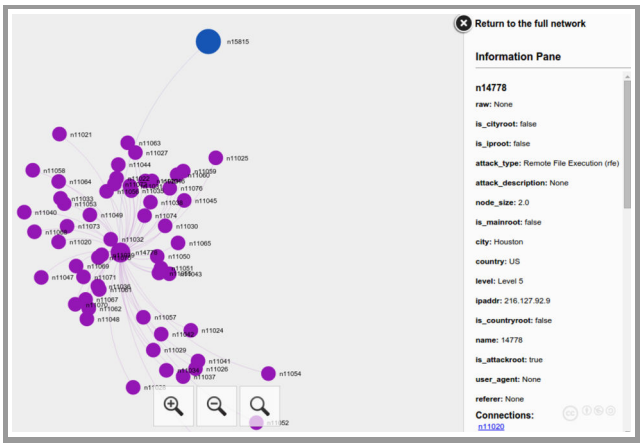


Fig. 7. The display when investigator clicks the root of attack type cluster.

There are three attack types detected in this dataset: 14.66% cross-site scripting (XSS), remote file execution (RFE) with 11.64%, local file inclusion (LFI) with 0.13%, and the normal request account of 65.83%. Table 2 depicts the examples of detected requests based on every PHP-IDS filter shown in Table 1. In the first example, the attacker tried to run a script while in the second one he executed a remote script that was previously downloaded using wget command. The last example shows that the attacker attempts to run an unauthorized file in local directory.

Table 2
Example of detected malicious requests

No.	Raw request	Attack
1	193.224.164.47 - - [27/Feb/2006:03:01:23 -0500] "GET /scripts/..%%35c../winnt/ system32/cmd.exe?/c+dir HTTP/1.0" 400 297 "-" "-"	XSS
2	62.175.253.180 - - [22/Jan/2006:08:28:17 -0500] "GET /awstats/awstats.pl?configdir=| echo;echo%20YYY;cd%20%2ftmp%3bwget%20209 %2e136%2e48%2e69%2fmirela%3bchmod%20%2bx %20mirela%3b%2e%2fmirela;echo%20YYY;echo| HTTP/1.1" 404 296 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;)"	RFE
3	218.26.222.13 - - [07/Feb/2006:01:36:28 -0500] "GET /c/winnt/system32/cmd.exe?/ c+dir HTTP/1.0" 404 293 "-" "-"	LFI

Table 3
Top five countries and their cities for each attack type

No.	Attack type	Country	City	Count
1	Cross-Site Scripting (XSS)	n/a	n/a	1497
		US	Amana	138
		US	Livingston	120
		DE	Berlin	108
		US	Pleasanton	84
2	Remote File Execution (RFE)	n/a	n/a	753
		CN	Beijing	245
		US	Dallas	99
		FR	Bischheim	74
		US	Saint Louis	60
3	Local File Inclusion (LFI)	n/a	n/a	4
		CN	Harbin	2
		CN	Jinan	2
		HU	Budapest	2
		US	Schaumburg	2
List of countries: US – United States, DE – Germany, CN – China, FR – France, HU – Hungary.				

Table 4
Top ten attacker's IP addresses

No.	IP address	Attack type	Count
1	64.6.73.199	Cross-Site Scripting	138
2	81.114.87.11	Cross-Site Scripting	138
3	80.55.248.206	Cross-Site Scripting	132
4	200.99.135.130	Cross-Site Scripting	120
5	209.137.246.36	Cross-Site Scripting	120
6	211.99.203.228	Remote File Execution	120
7	64.214.80.6	Cross-Site Scripting	108
8	82.127.23.55	Cross-Site Scripting	108
9	85.214.20.161	Cross-Site Scripting	108
10	82.177.96.6	Remote File Execution	107

Table 3 shows the top five countries and their respective cities for each attack type while Table 4 lists the top 10 attacker's IP addresses. One can see that there are very high "n/a" values in Table 3 since the free and open source version of GeoIP API is used [20] and the provided data are not as complete as in the paid one. As stated in Table 3, the most frequently detected IP address (64.6.73.199) attempted an XSS attack as shown in Table 4. The remote file inclusion attacks are also included in the top ten IP addresses while there is no local file inclusion since it only has a very small number of attempts from the whole dataset.

4. Conclusions and Future Works

In this paper, a graph-based forensic analysis have been implemented to examine an access log from a web honeypot. The proposed method employs an agglomerative clustering to group every record and to model them as an undirected graph. The clustering has some levels based on node's attributes, i.e., attack type, attacker IP address, and attacker's origin city and country. Every log is checked for its maliciousness, then visualized using a large graph layout, and then accessed using a web browser. These procedures will help forensic investigators to examine logs to work interactively and collaboratively with others.

In the future, authors plan to include more access logs archived in /var/log/httpd. This strategy will enable the investigator to comprehensively analyze through the last period of log rotation (the most common time frame is one month). The Sigma.js web interface will be improved to view a specific range of time, although it still displays a large graph generated from thousands of logs. In addition, the proposed method can be extended to become live forensic analysis of a web honeypot or typical web application by reading and parsing the access log periodically. This approach provides administrators real-time monitoring and reports if there are any attack attempts to their system. In relation to the types of offensive activity, the system will be enhanced with OWASP ModSecurity Rules [30] which contains more complete rules since it is actually a web application firewall but that can be utilized in a forensic manner. To increase the reliability of the proposed technique, authors also plan to implement a graph database such as Neo4j or Titan (natively distributed one) to make analyzed logs become persistent and able to be queried any time.

References

[1] L. Spitzner, Honeypots: *Tracking Hackers*. Boston: Addison-Wesley Longman Publish., 2002.

[2] A. Yasinsac and Y. Manzano, "Honeytraps: A network forensic tool", in *Proc. World Multiconf. System., Cybernet. & Informat. SCI 2002*, Orlando, FL, USA, 2002.

[3] F. Pouget and M. Dacier, "Honeypot-based forensics", in *Proc. AusCERT Asia Pacific Inform. Technol. Secur. Conf. AusCERT2004*, Gold Coast, Australia, 2004.

[4] F. Raynal, Y. Berthier, P. Biondi, and D. Kaminsky, "Honeypot forensics, part I: Analyzing the network", *IEEE Secur. Privacy*, vol. 2, no. 4, pp. 72–78, 2004.

[5] F. Raynal, Y. Berthier, P. Biondi, and D. Kaminsky, "Honeypot forensics, part II: Analyzing the compromised host", *IEEE Secur. Privacy*, vol. 2, no. 5, pp. 77–80, 2004.

[6] K. D. Fairbanks, C. P. Lee, Y. H. Xia, and H. L. Owen, "Timekeeper: a metadata archiving method for honeypot forensics", in *Proc. Inform. Assurance & Secur. Worksh. IAW'07*, West Point, NY, USA pp. 114–118, 2007.

[7] P. T. Chen, C. S. Laih, F. Pouget, and M. Dacier, "Comparative survey of local honeypot sensors to assist network forensics", in *Proc. 1st Int. Worksh. System. Approach. to Digit. Forensic Engin. SADFE'05*, Taipei, Taiwan, 2005, pp. 120–132.

[8] V. Maheswari and P. E. Sankaranarayanan, "Honeypots: Deployment and data forensic analysis", in *Proc. Int. Conf. Computational Intellig. & Multimed. Appl.*, Sivakasi, Tamil Nadu, 2007, vol. 4, pp. 129–131.

[9] S. Riebach, E. P. Rathgeb, and B. Toedtmann, "Efficient deployment of honeynets for statistical and forensic analysis of attacks from the Internet", in *NETWORKING 2005. Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications Systems*, R. Boutaba, K. Almeroth, R. Puigjaner, S. Shen, and J. P. Black, Eds., *LNCS*, vol. 3462, pp. 756–767. Springer, 2005.

[10] W. Ren and H. Jin, "Honeynet based distributed adaptive network forensics and active real time investigation", in *Proc. 20th ACM Symp. Appl. Comput. SAC 2005*, Santa Fe, NM, USA, 2005, pp. 302–303.

[11] A. Capalik, "Next-generation honeynet technology with real-time forensics for U.S. defense", in *Proc. IEEE Milit. Commun. Conf. MILCOM 2007*, Orlando, FL, USA, 2007, pp. 1–7.

[12] O. Thonnard and M. Dacier, "A framework for attack patterns' discovery in honeynet data", *Digit. Investig.*, vol. 5 (Supplement), pp. S128–S139, 2008.

[13] V. H. Pham and M. Dacier, "Honeypot trace forensics: The observation viewpoint matters", *Future Gener. Comp. Syst.*, vol. 27, no. 5, pp. 539–546, 2011.

[14] Q. Nasir and Z. A. Al-Mousa, "Honeypots aiding network forensics: Challenges and notions", *J. Commun.*, vol. 8, no. 11, pp. 700–707, 2013.

[15] C. Pohl, A. Zugenmaier, M. Meier, and H. Hof, "B.Hive: A zero configuration forms honeypot for productive web applications", in *ICT Systems Security and Privacy Protection*, *IFIP Advances in Information and Communication Technology*, vol. 455, pp. 267–280. Springer, 2015.

[16] K. Cabaj, and P. Gawkowski, "Systemy HoneyPot w praktyce (HoneyPot systems in practice)", *Przegląd Elektrotechniczny*, vol. 91, no. 2, pp. 63–67, 2015 (in Polish) (doi: 10.15199/48.2015.OZ.16).

[17] C. Valli, "Visualisation of honeypot data using Graphviz and Afterglow", *J. Digit. Forensic, Secur. & Law*, vol. 4, no. 2, pp. 27–38, 2011.

[18] K. Cabaj, "Visualization as support for web honeypot data analysis", *Inform. Systems in Manag.*, vol. 4, no. 1, pp. 14–25, 2015.

[19] A. Chuvakin, "Free honeynet log data for research", The Honeynet Project [Online]. Available: https://www.honeynet.org/node/456

[20] MaxMind, "GeoIP Legacy C API" [Online]. Available: https://github.com/maxmind/geoip-api-c

[21] Applied Security, "Pure Python API for Maxmind's binary GeoIP databases" [Online]. Available: https://github.com/appliedsec/pygeoip

[22] Y. Hu, "Efficient, high-quality force-directed graph drawing", *Mathematica J.*, vol. 10, no. 1, pp. 37–71, 2005.

[23] M. Bastian, S. Heymann, and M. Jacomy, "Gephi: An open source software for exploring and manipulating networks", in *Proc. 3rd Int. AAAI Conf. on Weblogs & Soc. Media*, San Jose, CA, USA, 2009, pp. 361–362.

[24] PHP-IDS, "PHP-Intrusion Detection System" [Online]. Available: https://github.com/PHPIDS/PHPIDS

[25] R. Gaucher, "Apache Scalp: Apache log analyzer for security" [Online]. Available: https://code.google.com/p/apache-scalp/

[26] S. Martin, W. M. Brown, R. Klavans, and K. W. Boyack, "OpenOrd: An open-source toolbox for large graph layout", *Proc. of SPIE, Visualization and Data Analysis*, San Francisco, CA, USA, 2011, vol. 7868, pp. 786806–786806, 2011.

[27] Sigma.js, "JavaScript library dedicated to graph drawing" [Online]. Available: http://sigmajs.org/

[28] G. Csardi and T. Nepusz, "The igraph software package for complex network research", *InterJournal Complex Systems*, vol. 1695(5), pp. 1–9, 2006.

[29] S. Hale, "Sigmajs exporter – Gephi marketplace" [Online]. Available: https://marketplace.gephi.org/plugin/sigmajs-exporter/

[30] SpiderLabs, OWASP ModSecurity Core Rule Set (CRS) [Online]. Available: https://github.com/SpiderLabs/owasp-modsecurity-crs

Hudan Studiawan earned both a B.Sc. and M.Sc. degree from Department of Informatics, Institut Teknologi Sepuluh Nopember, Indonesia. He is now working as a lecturer at his alma mater university. His research areas are digital forensics, network security, and cluster computing.

E-mail: hudan@if.its.ac.id
Department of Informatics
Institut Teknologi Sepuluh Nopember (ITS)
Surabaya, Indonesia

Supeno Djanali is a Professor of Network Architecture and Design in Department of Informatics, Institut Teknologi Sepuluh Nopember, Indonesia. He studied at University of Wisconsin-Madison, USA, for both his master's and Ph.D. degrees. His research areas are primarily network security and mobile computing.

E-mail: supeno@its.ac.id
Department of Informatics
Institut Teknologi Sepuluh Nopember (ITS)
Surabaya, Indonesia

Baskoro Adi Pratomo is a lecturer in Department of Informatics, Institut Teknologi Sepuluh Nopember, Indonesia. He finished his B.Sc. and M.Sc. degree from the same university as he works at the moment. His main research interest is network security.

E-mail: baskoro@if.its.ac.id
Department of Informatics
Institut Teknologi Sepuluh Nopember (ITS)
Surabaya, Indonesia

Paper

Network Function Virtualization: Mitigating the Impact of VoLTE on the Policy and the Charging System

Youness Jouihri[1], Zouhair Guennoun[1], Youssef Chagh[1], and Driss Zahi[2]

[1] *Electronics and Communications Laboratory (LEC), Mohammadia School of Engineering (EMI), Rabat Mohammed V University (UM5R), Rabat, Morocco*
[2] *Gregor Research Lab, Pantheon-Sorbonne University, Paris, France*

Abstract—Needless to say that telecommunications' operators are showing increasing interests toward solving the dilemma of optimizing network resources while offering state-of-the-art quality of service. Recently, operators have shown an increasing interest to investigate solutions for better control on network utilization, service usage and monetization. They also noticed a significant growth in Diameter signaling and more specifically in signaling related to policy management. A massive introduction of Voice over LTE (VoLTE) service will have a significant impact on the systems handling policy signaling, as VoLTE will reshape the landscape of Long Term Evolution (LTE) policies and completely change the way policy management works. However, this massive approach is meant to provide significant competitive advantages for operators offering LTE services and still require circuit-switched network to provide voice service. The biggest challenge for those operators is to find an appropriate solution, scalable enough to handle the unpredictable growth of Diameter signaling. In this paper, a model, based on Network Function Virtualization (NFV) technology is proposed, able to address the challenges of massively introducing VoLTE, without impacting existing services and without jeopardizing current revenues. In presented approach, the standard VoLTE call flows, referenced user's behavior and latest experiments' results on NFV technology are used.

Keywords—NFV, policy diameter signaling, policy management, VoLTE.

1. Introduction

Maintaining the profitability of services and keeping a high level of customer satisfaction is becoming a challenging situation after the drastic growth in data services. The intensive deployments of LTE networks and the high penetration of smart phones are the major drivers for such growth.

Many research studies have shown that telcos will soon reach an "end-of-profit" point if they do not find appropriate solutions to sustain their profit margin [1]. As the cost of the required network expansions to handle this growth increases faster than the generated profit. Consequently, the focus of operators is put on policy management signaling to sustain the profitability of their business and efficiently control the infrastructure.

A wide range of research studies treated the topic of handling the growth of data services [2], [3]. However, literature addressing the scalability issue of the systems handling the signaling associated with those services is difficult to find. In contrast to existing literature, the research aims to investigate the challenges faced by LTE operators while managing the LTE Diameter signaling, more specifically those related to the policy management signaling after introducing VoLTE.

This paper is structured as follows. Section 2 highlights the growth in Diameter signaling policies versus current policy management systems' limitations. In Section 3, the benefits and impacts of massive VoLTE introduction are shown. In Section 4, the challenges of using Network Function Virtualization (NFV) technology to mitigate the VoLTE introduction impact are studied. Finally, Section 5 concludes the paper.

2. Growth in Diameter Signaling Policies Versus Current Policy Management System Limitation

2.1. Growth of Diameter Signaling

Over the past few years, mobile operators reported a demand shift from voice to data services, and started investing massively to support the impressive growth in mobile data traffic. However, most of them have given less interest to the signaling growth. When comparing the global LTE Diameter signaling growth and the global IP traffic (including mobile data traffic) growth, authors find that the Diameter signaling grows at a Compound Annual Growth Rate (CAGR) of 68%, and will surpass the total global IP traffic by 2019 growing at a CAGR of 23% [4], [5] (Fig. 1).

The Diameter signaling has considerably increased. The high LTE network deployment and smartphone penetration, along with the changing subscribers' behavior were the key players in this drastic increase.

Special care should be given to the Diameter signaling volume growth, as it may overwhelm the mobile operators existing infrastructures, and jeopardize both the cost forecasted for future expansions and the end users' Quality of Service (QoS). Mobile operators should study the Diame-

ter signaling and make their infrastructure ready to deliver future mobile services, with similar or better quality than the current one.

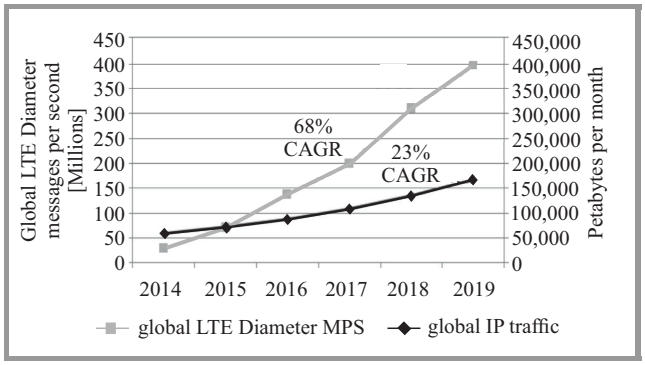


Fig. 1. LTE Diameter signaling and mobile data growth.

In an LTE network, the Diameter signaling traffic is needed to handle Mobility Management, LTE Broadcast and other signaling activities such policy management signaling, required to provide and to handle the LTE services. The policy management signaling is becoming the dominant Diameter traffic and will continue its growth (Fig. 2) [4]. As it is gaining more importance and will significantly influence future services. Operators usually rely on the use of policies for fair usage and traffic management cases. Meanwhile, they started implementing complex use cases to generate more revenues, and providing a wider range of service packages to their end users.

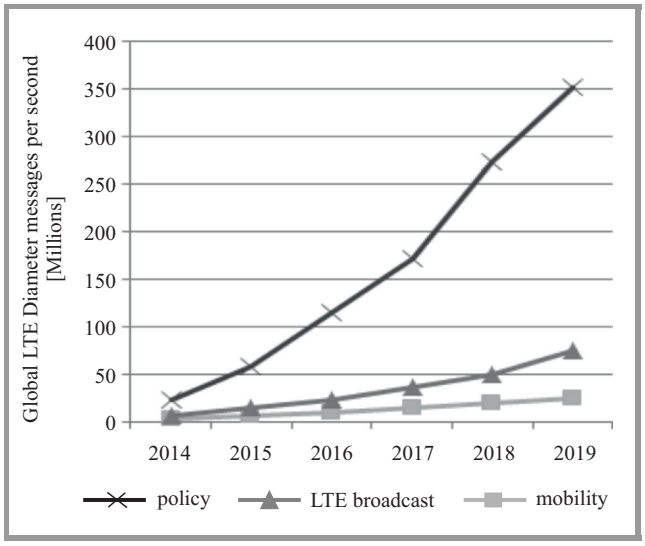


Fig. 2. Global LTE Diameter signaling by use case.

The policy management signaling growth is impacted by the mobile plans concerning service policies. Such plans are the consequence of a significant migration to smartphones, which generate approximately 20 times higher signaling and traffic [6]. Nevertheless, authors still believe that the launch of Voice over LTE service (VoLTE) will have a crucial impact on the policy management processes and growth.

2.2. Reasons and Management Challenges of Policy Signaling Growth

As highlighted earlier, mobile operators are shifting their investment from a voice-dominant towards a data-dominant infrastructure. During the voice-dominant period, the revenue growth was proportional to the traffic volume [7]. However, after moving to the data-dominant network, the data traffic is growing faster than the associated revenues (Fig. 3) [7]. This results in a gap between traffic and revenue, mainly caused by inadequate pricing for some services (e.g. the unlimited data packages), as well as the fierce competition between operators. Moreover, if operators continue to equally treat the data packet's content and enable all the services by default, Over The Top (OTT) players will take the lion's share of the revenue generated by OTT services, which causes a loss of potential revenues. If no remedy action is taken by operators, they might reach an "end of profit" point, as highlighted in many studies [1], [8].

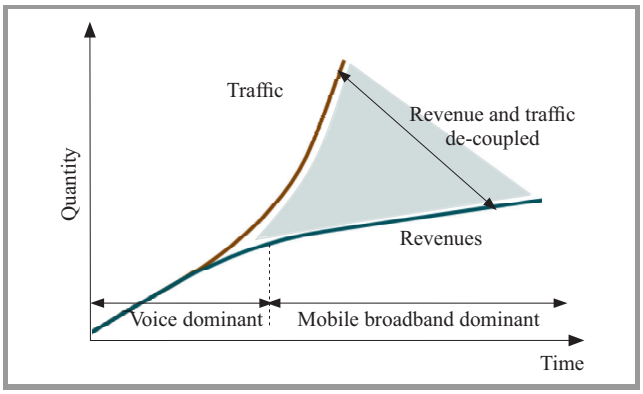


Fig. 3. Coupling of traffic and revenue in mobile networks.

This is to say that operators need techniques to decouple network costs from traffic growth, by monitoring, managing and charging accurately different types of data services. These techniques should also include the control of user's usage, and should have the ability to tackle abuses in consuming network resources. In parallel, these techniques could provide a big support to operators, if they offer the possibility to limit the congestion and enhance the QoS, without any need for capacity expansion. Operators should also investigate ways of increasing subscribers' loyalty, service uptake, and practices that can maximize the revenue even from casual consumers.

The 3rd Generation Partnership Project (3GPP), had the foresight to address these challenges and introduced the network Policy and Charging Rules Function (PCRF) and Policy and Charging Enforcement Function (PCEF), responsible for the management of network QoS, and policy and charging management solutions [9].

Most of the operators start taking control of their network by applying fair usage and traffic management policies, enabled by the PCRF and PCEF functions. However, to achieve the enhanced services' control, charging and qual-

ity, operators ended up implementing huge number of complex policies [4]. Proper policies and QoS strategies result in taking back the control over the traffic consumption and charging. Nevertheless, those strategies have the adverse effect of significantly increasing the Diameter signaling of policies, which raises new challenges in managing and executing such policies. Operators look for a centralized policy decision architecture, which enables easy provisioning and high integrity of data. However, many operators start reporting signaling overload after deploying centralized architecture as stated in [10].

Undesirably, even after investing and making a robust Policy Control and Charging (PCC) architecture with a centralized policy decision, operators would not be able to fully utilize this mechanism to efficiently manage traffic growth and services monetization. The current telecom software solutions can run only on a proprietary hardware. Scaling the policy architecture might require multiple racks, full of servers [10]. After exceeding the maximum limit of one PCRF node capacity, the introduction of a new PCRF node will need the addition of a Diameter Routing Agent (DRA), capable of performing binding. To make sure that all messages associated with a user's particular IP Connectivity Access Network (IPCAN) session are processed by the same PCRF node. This increases latency, and adds new nodes to the infrastructure, resulting in an increase of the capital and operational expenditures [11].

Operators will absurdly reach a situation, where the implemented solution generates the same problems as the ones it was designed to solve by impacting the service quality (caused by latency) and increasing the network cost.

3. Introduction of VoLTE Service: Benefits and Impacts

3.1. Advantages of Massive Migration to VoLTE Service

Most of the operators are now aware of the challenging growth of signaling, and start looking for innovative strategies to be implemented in their policy management system to prevent uncontrolled growth of Diameter signaling. But few of them took into consideration the impact of introducing VoLTE service on policy signaling considering the launch of such a service equivalent to the launch of any LTE data service.

Around 80 operators, in 47 countries, are investing in VoLTE deployments [4]. Most of them are still in a small scale deployment, as they are not yet ready to go for a massive introduction of VoLTE, and get rid of their legacy network built over several years. Moreover, currently, only a limited number of handsets are supporting this service. This adds more constraints on the massive move scenario. That is why, the impact of VoLTE service on policy system will not be noticeable at this stage of small migration.

The spectrum bandwidth consumption for a voice call in LTE network is less than the one in circuit-switched network [6]. Operators can use the spectrum resources in a more efficient manner, and will have the ability to nearly double the spectrum capacity with the use of VoLTE service [6]. Furthermore, the migration of a fair percentage of voice load, from legacy networks into LTE, frees up existing bandwidth. The freed bandwidth can be reused to increase the capacity of LTE network and paves the way for technical innovations, focusing in providing very high data speeds to the end users, such as carrier aggregation techniques. The suggestion of "mind commerce" in [6] for a technical case to be made for a massive introduction of VoLTE might attract many operators, especially those having limitations in spectrum resources. To achieve that, operators are making their VoLTE core network, mainly their IP Multimedia Subsystem (IMS), ready to support this massive move. The IMS is an all IP control platform for both fixed and mobile networks, consisting of several logical functions such as Call Session Control Function (CSCF) that controls voice sessions and Application Server (AS) that controls subscribers' services. Detailed VoLTE architecture, including nodes and interfaces, is presented in Fig. 5.

In parallel, handset suppliers are supporting this massive migration scenario, by driving a rapid erosion of average selling prices of smartphones. This is likely to accelerate user migration from basic and feature phones to smartphones [12].

3.2. Impact of VoLTE Introduction on the Policy Management System

Introducing VoLTE in an LTE network will completely change the way policy management works. Unlike the LTE data services, where the policy management handles only one session with PCEF function, the VoLTE needs to correlate two sessions, one with the PCEF functions, and another one with the application functions [11]. For an appropriate network dimensioning, operators planning to launch VoLTE service need to ensure that their infrastructure can properly handle this additional service, by supporting the associated signaling load and generated traffic. Those signaling and traffic loads are impacted by not only the number of subscribers, but also by the subscribers' behavior, their devices, and the services or applications they invoke.

Recall that the Busy Hour Call Attempts (BHCA) value is one of the key values used to design and dimension circuit-switched network [13]. During conducted analysis of the impact of VoLTE introduction in a LTE network, it was assumed that the behavior of subscribers using voice in a circuit-switched network and their BHCA values will remain the same once they move to VoLTE service.

An LTE subscriber provisioned to use VoLTE service needs two default bearers, one for voice and another for data. These default bearers will remain established as long as the subscriber is attached to LTE network. Besides this, a dedicated bearer will be established in the case of mobile originated call and mobile terminated call. This bearer will be released after the end of the call [14].

Handling these additional bearers will be the real challenge that VoLTE service will add to LTE network. Doubling the use of default bearers will increase the number of Diameter signaling messages required to establish or terminate these bearers. Furthermore, the control of dedicated bearers requires a higher number of Diameter signaling messages comparing to the one used to handle default bearers. Consequently, operators should carefully study the impact of VoLTE service introduction, and should not consider it as a normal data service launch.

3.3. A Case Study to Highlight the Impact of VoLTE on Diameter Signaling

To emphasize the severity of Diameter signaling increase, a use case was simulated, with a proposed model for VoLTE subscribers increase (N_v), and another to estimate the forecast of subscribers using LTE data-only (N_d). The proposed subscribers' forecast is close to a massive VoLTE migration scenario.

The average number of Diameter signaling transactions per second during the busy hour at year x, $S_{total}(x)$, generated by LTE subscribers (VoLTE and LTE data-only) can be calculated as:

$$S_{total}(x) = aN_v(x) + bN_d(x), \tag{1}$$

where $N_v(x)$ is the proposed forecast of VoLTE subscribers' growth at year x, $N_d(x)$ is the forecast of LTE data-only subscribers growth at year x.

The factors a and b are defined as:

$$\begin{aligned} a &= \frac{1}{3600}\big[p_2(t_2+t_1+t_5+t_6)+p_5t_8+ \\ &\quad +p_3(t_4+t_9)+p_1(t_3+p_4t_{10})\big] \\ b &= \frac{1}{3600}\big[p_2(t_1+t_5+t_6+t_7)+p_3t_4\big]. \end{aligned}$$

p_1, p_2, p_3, p_4 and p_5 are parameters used to model the users' behavior and LTE coverage. Their descriptions, values and references used to extract those values can be found in Table 1.

$t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8, t_9$ and t_{10} are parameters representing the transactions required to handle the default and dedicated bearers' activities (i.e. establishment, modification, termination), between the different LTE core elements (Table 1). The standard 3GPP call flows is used [15]–[19] to calculate these values.

The $N_v(x)$ was simulated using an "S" (sigmoid) curve shape:

$$N_v(x) = \frac{S_{\max}}{1+e^{-\lambda_1(x-x_0)}},$$

where $S_{\max}, x_0$ and λ_1 are respectively the maximum number of subscribers, the inflection year (convexity change in subscriber forecast), and the slope parameter for subscriber forecast. For our case, we use $S_{\max} = 20$ millions, $x_0 = 3$ years and $\lambda_1 = 2.5$, to represent VoLTE subscribers increase following an "S" curve going from 0 to 20 millions over 6 years (Table 2). The $S_{\max}$ value is an estimation of number of subscribers, excluding LTE data-only subscribers, in large operators in Middle East and North Africa (MENA) by first quarter of year 2021 [20].

The $N_d(x)$ was simulated using the sum of two normal distributions, $N(\mu_1, \sigma_1)$ and $N(\mu_2, \sigma_2)$, to respectively model the impact of VoLTE introduction on LTE data-only subscribers, and their forecasted growth:

$$\begin{aligned} N_d(x) &= n_1 N(\mu_1, \sigma_1) + n_2 N(\mu_2, \sigma_2), \\ N_d(x) &= \frac{n_1}{\sigma_1\sqrt{2\pi}}\, e^{\frac{-(x-\mu_1)^2}{2\sigma_1^2}} + \frac{n_2}{\sigma_2\sqrt{2\pi}}\, e^{\frac{-(x-\mu_2)^2}{2\sigma_2^2}}, \end{aligned}$$

where n_1 and n_2 are calculated based on the initial and target values of LTE data-only subscriber's growth. In this case: $n_1 = 25$, $n_2 = 8$, $\mu_1 = 6$, $\mu_2 = -0.1$, $\sigma_1 = 2$ and $\sigma_2 = 1.6$. These values lead to an evolution of the number of LTE data-only subscribers', starting from an initial value of around 2.05 million subscribers [21], where VoLTE was not yet launched in large operators in MENA. This initial value includes subscribers using voice service in circuit-switched network. Such a number will then decrease until it reaches the value of 1.5 million after almost three years. This is justified by the fact that many of those subscribers will start using VoLTE service instead of circuit-switched voice. The authors also believe that during the first three years, the number of LTE data-only migrated to VoLTE will be higher than the forecasted growth of LTE data-only subscribers. In the subsequent years, the number of LTE data-only subscribers will increase as per the forecasted growth until it reaches 4.99 million in year 2021 as the growth in VoLTE subscribers will slow down (Table 2).

The Eq. (1) was calculated based on the sum of signaling Diameter transactions generated by the activities associated to the below functions, where the first element of the function represents the number of transactions related to the Internet default bearer, the second corresponds to the number of transactions pertaining to VoLTE default bearer. The third is association with the number of VoLTE dedicated bearer transactions:

- $F_1(x)$ – number of Diameter transactions to establish default bearers (Internet and VoLTE):

$$\begin{aligned} F_1(x) &= \frac{1}{3600}\begin{bmatrix} F_{11} & F_{12} & F_{13} \end{bmatrix}, \\ F_1(x) &= \frac{p_2}{3600}\begin{bmatrix} t_1(N_v(x)+N_d(x)) & t_2N_v(x) & 0 \end{bmatrix}; \end{aligned}$$

- $F_2(x)$ – number of Diameter transactions to authenticate an LTE subscriber (VoLTE and LTE data-only):

$$\begin{aligned} F_2(x) &= \frac{1}{3600}\begin{bmatrix} F_{21} & F_{22} & F_{23} \end{bmatrix}, \\ F_2(x) &= \frac{p_2}{3600}\begin{bmatrix} t_5N_d(x) & t_5N_v(x) & 0 \end{bmatrix}; \end{aligned}$$

Table 1
Signaling Diameter activities – parameter values and traffic quantification

Network attach parameters	Symbol	Value	References
Number of network attach requests per subscriber during busy hour (BH)	p_2	0.75	[23]
Number of requests to refresh IMS registration per VoLTE subscriber (BH)	p_5	1	[24]
Number of Diameter transactions to establish Internet default bearer (Gx interface) – all listed interfaces are presented in Fig. 5	t_1	1	[15]
Number of Diameter transactions to establish VoLTE default bearer (Gx interface)	t_2	1	[15]
Number of Diameter transactions to authenticate an LTE subscriber (S6a interface)	t_5	2	[16]
Number of Diameter transactions to download spending limit for an LTE subscriber (Sy interface)	t_6	1	[15]
Number of Diameter transactions to charge LTE data service (Gy interface)	t_7	2	[17]
Number of Diameter transactions to register VoLTE subscriber in the IMS (Cx, Sh interfaces)	t_8	6	[17] [19]
VoLTE call parameters description			
Busy hour call attempts	p_1	2	Estimated[1]
Percentage of prepaid VoLTE subscribers	p_4	0.8	[22]
Number of Diameter transactions to establish and release dedicated bearer (Gx and Rx interfaces)	t_3	7	[15]
Number of Diameter transactions to charge a prepaid VoLTE call (Ro interface)	t_{10}	2	[18]
Network detach parameters description			
Number of network detach requests per subscriber (BH)	p_3	0.75	[23]
Number of Diameter transactions to release a default bearer (Gx interface)	t_4	1	[15]
Number of Diameter transactions to deregister a VoLTE subscriber from IMS (Cx interface)	t_9	1	[19]

[1] Estimated similarly to many dimensioning use cases, where the initial value was set to 2 and increased yearly. Authors prefer to keep it constant (conservative forecast) as this value influences a lot the Diameter growth.

- $F_3(x)$ – number of transactions to download the spending limit report from on-line charging system:

$$F_3(x) = \frac{1}{3600}\begin{bmatrix} F_{31} & F_{32} & F_{33} \end{bmatrix},$$

$$F_3(x) = \frac{p_2}{3600}\begin{bmatrix} t_6 N_d(x) & t_6 N_v(x) & 0 \end{bmatrix};$$

- $F_4(x)$ – number of Diameter transactions to charge LTE data services:

$$F_4(x) = \frac{1}{3600}\begin{bmatrix} F_{41} & F_{42} & F_{43} \end{bmatrix},$$

$$F_4(x) = \frac{p_2}{3600}\begin{bmatrix} t_7 N_d(x) & t_7 N_v(x) & 0 \end{bmatrix};$$

- $F_5(x)$ – number of Diameter transactions to register a VoLTE subscriber in IMS:

$$F_5(x) = \frac{1}{3600}\begin{bmatrix} F_{51} & F_{52} & F_{53} \end{bmatrix}$$

$$F_5(x) = \frac{p_5}{3600}\begin{bmatrix} 0 & t_8 N_v(x) & 0 \end{bmatrix}$$

- $F_6(x)$ – number of Diameter transactions to release a default bearer (VoLTE and Internet):

$$F_6(x) = \frac{1}{3600}\begin{bmatrix} F_{61} & F_{62} & F_{63} \end{bmatrix},$$

$$F_6(x) = \frac{p_3}{3600}\begin{bmatrix} t_4 N_d(x) & t_4 N_v(x) & 0 \end{bmatrix};$$

- $F_7(x)$ – number of Diameter transactions to deregister a VoLTE subscriber from IMS:

$$F_7(x) = \frac{1}{3600}\begin{bmatrix} F_{71} & F_{72} & F_{73} \end{bmatrix},$$

$$F_7(x) = \frac{p_3}{3600}\begin{bmatrix} 0 & t_9 N_v(x) & 0 \end{bmatrix};$$

- $F_8(x)$ – number of Diameter transactions to establish and release dedicated bearer:

$$F_8(x) = \frac{1}{3600}\begin{bmatrix} F_{81} & F_{82} & F_{83} \end{bmatrix},$$

$$F_8(x) = \frac{p_1}{3600}\begin{bmatrix} 0 & 0 & t_3 N_v(x) \end{bmatrix};$$

- $F_9(x)$ – number of Diameter transactions to charge a prepaid VoLTE call:

Table 2
Increase of VoLTE subscribers in years 2015–2021 with used parameters description

Symbol	Description	2015	2016	2017	2018	2019	2020	2021
N_d	Number of subscribers using LTE data-only [million]	2.05	1.79	1.52	1.92	3.10	4.41	4.99
N_v	Number of VoLTE subscribers [million]	0.00	0.30	3.96	16.04	19.70	19.98	20.00
	Average (av.) number (no.) of network attach Diameter TPS (BH)	2558	3166	14261	52542	65451	67960	68732
	Av. no. of VoLTE calls Diameter TPS (BH)	0	1412	18902	76653	94144	95468	95550
	Av. no. of Network detach Diameter TPS (BH)	426	497	1965	7086	8856	9245	9372
	Av. no. of policy Diameter TPS (BH)	1279	2648	21390	84095	103744	105997	106445
S_{total}	Av. no. of LTE subscribers Diameter TPS (BH)	2984	5074	35128	136281	168451	172673	173654
S_{NFV}	Av. no. of Diameter TPS (BH) moved to NFV architecture (to be used in Subsection 4.2)	0	1773	23738	96262	118227	119891	119993

$$F_9(x) = \frac{1}{3600}\begin{bmatrix} F_{91} & F_{92} & F_{93} \end{bmatrix},$$

$$F_9(x) = \frac{p_1}{3600}\begin{bmatrix} 0 & 0 & p_4 t_{10} N_v(x) \end{bmatrix}.$$

$S_{total}(x)$ calculated in Eq. (1) can also be written as:

$$S_{total}(x) = \frac{1}{3600}\sum_{j=1}^{j=3}\sum_{i=1}^{i=9} F_{ij}.$$

In Table 1, the values and references of all the parameters used are listed and the $S_{total}(x)$ calculated values from year 2015 to year 2021 are shown.

On the simulated use case, the transactions generated by managing the dedicated bearer's associated to the establishment and the termination of voice call activities are calculated. Also the number of transactions required to handle default bearers are computed, mainly related to attach and detach activities to or from network and influenced by the LTE coverage or subscribers' behaviors. Finally, it is estimated that 80% of the subscribers are prepaid, extracted from the GSMA worldwide prepaid percentage report in [22], and the transactions required to charge them are computed.

The average number of LTE subscribers Diameter transactions per second during busy hour, presented in Fig. 4, will go beyond hundred thousand transactions per second after almost three years from the launch of VoLTE. Around 62% of those transactions are related to Diameter signaling policy. Consequently, despite excluding the impact of the policy techniques, introducing VoLTE in a massive migration scenario will significantly influence the Diameter signaling growth in general and policy signaling management in particular.

By analyzing the signaling growth, for an operator expecting to serve around 25 million subscribers in year 2021, and using basic policy techniques to manage data services, one can conclude that operators having similar number of subscribers, and using complex policies or willing to use them, will not be able to go for a VoLTE massive migration, unless their policy management is scalable enough to accommodate the heavy increase in signaling. Ideally, one single PCRF node should be used to manage the complete signaling policies, in order to avoid problems, listed earlier in Subsection 2.2, of introducing another PCRF node. The current PCRF nodes use proprietary hardware, and cannot be scaled to accommodate both the expected signaling growth that is calculated in presented case, and the flexibility in implementing complex policies. The existing solutions to accommodate the policy signaling growth consist of adding new nodes, either new PCRF node or front-end servers if the PCRF solution is a front-end back-end based solution. However for all those solutions, DRA will be needed to perform binding and to ensure proper communication between network elements (authors do not suggest the addition of DRA as explained earlier in Subsection 2.2).

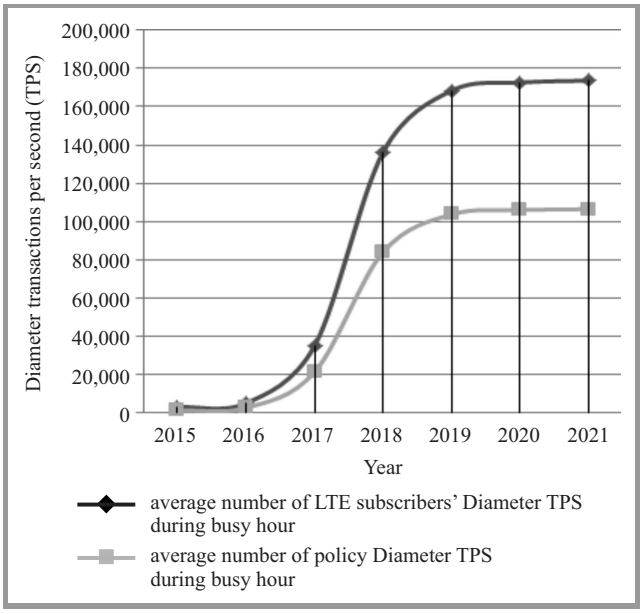


Fig. 4. Prognosed impact of VoLTE on Diameter signaling growth.

4. Challenges of Using NFV Technology to Mitigate the VoLTE Introduction Impact

4.1. Designing an Architecture Model to Overcome VoLTE Challenges

In the current telecom environment, it is not always possible to scale a single point controlling all policies' decision [10]. To have centralized policy decision architecture, able to handle the massive introduction of VoLTE service, it is necessary to find an appropriate environment to build a strong policy system. Ideally, such environment should give the possibility of unlimited scaling, to overcome the unpredictable policy Diameter signaling growth.

Achieving a flexible scaling and giving the possibility to decouple software from hardware in network nodes with a reasonable cost, will help operators to protect their profit margin by preventing expensive network expansions. Adding the possibility of "paying only for what they need, and only when they need it" [25] will be the optimal solution that operators are eager to procure.

Considering the Information Technology (IT) domain experience to solve the issue of inefficient use of servers, by inventing the concept of hardware virtualization and cloud data center; the European Telecommunications Standards Institute (ETSI) published the NFV approach to virtualize telecom network node functions [25]. The cloud environment using the NFV concept seems to be the goal of this, if we could assume that telecom and IT networks' requirements are similar. In telecom networks, it is all about carrier grade systems, where we should meet or exceed "five nines" (99.999%) availability is expected. While in the IT classic cloud, it is not a real necessity [26]. The maturity of the NFV is the biggest constraint blocking the spread of the telecom nodes' virtualization. Many of those nodes require special hardware and software solutions to run properly; and their functioning on a Commercial Off-The-Shelf (COTS) hardware might degrade their performance in contrast to their functioning on a proprietary hardware [25].

The short term virtualization strategy, should focus on network functions allowing minor performance degradation, and where the scaling of control functions overcomes the degradation that might be caused by the use of COTS hardware. Presented NFV architecture is built by analyzing the results of different researches in NFV technology, and taking into consideration the track records' outcomes of the major players in this technology. This architecture aims to mitigate the impact of massive VoLTE introduction, by the efficient use of the facilities acquired while using NFV technology, and by ensuring that the deployment of this architecture will not have any harmful impact on the existing services and infrastructures. Selecting the network functions to be used in this architecture was influenced by the analysis of previous NFV experiments, and by the study of VoLTE introduction impact on the existing network nodes.

During the Mobile World Congress 2015 (MWC 2015), Telefonica company shared many interesting experiences and findings. They confirmed that the classical cloud management can't provide a carrier grade performance. They also confirmed the strong impact of the underlying hardware on network functions' performance. Therefore, they proposed a staged NFV deployment plan [27]. Combining the analysis results of the use case presented in Subsection 3.3, and Telefonica proposals in MWC 2015, a detailed study for the policy management system virtualization is required to achieve authors' goals. Nevertheless the virtualization of any node running business critical services is not yet recommended. This means that the policy management system handling VoLTE service should be different from the one used for LTE data services. Deploying dedicated PCRF and PCEF functions for VoLTE in a virtualized architecture and adding CSCF function to this architecture, appears to be the straightforward solution to avoid all complications accompanying the introduction of VoLTE service, and uncertainty about its way of adoption. Those three functions handle the majority of the VoLTE policy signaling. Important is to highlight that virtualizing those nodes requires a proper assessment of the impact of such virtualization on their performance.

Referring back to Telefonica and other operators' experiences, the signaling functions PCRF and CSCF seem to be a good candidate for virtualization. The reason behind is that even their performance could be impacted by the use of COTS hardware. It can be overcome by scaling up the number of its control plane components. The Mobility Management Entity (MME) is also a pure signaling node, but authors prefer not to include it now, to ensure that proposed architecture's impact on the existing services, nodes and connections is marginal. Dedicating a separated NFV based Home Subscriber Server (HSS) to serve VoLTE subscribers is another interesting initiative; permitting the move of an important part of VoLTE Diameter signaling to virtualized network. But the NFV technology is not yet mature to serve critical systems, such as those handling subscriber databases, as announced by Telefonica [27]. The model will not include HSS, but still use the standard 3GPP Sh and Cx interfaces to communicate with the existing HSS (Fig. 5) will be used.

Dedicating a separate PCEF for VoLTE service requires the appropriation of the complete PGW functions, which are not limited only to signaling control. The PGW is the gateway terminating the connection towards external packet network, in addition to managing policy enforcement, packet filtration, and other control functions. Virtualizing the control part of the PGW can be treated similarly to CSCF and PCRF functions. However, virtualizing its forwarding part is not advisable, especially after Telefonica's experience, where the performance of throughput was significantly degraded using the Hypervisor of the cloud computing environment [28]. This degradation cannot be tolerated in a node requiring high packet-processing rate such as PGW. The Hypervisor is an intermediate software layer between the underlying hardware and the upper vir-

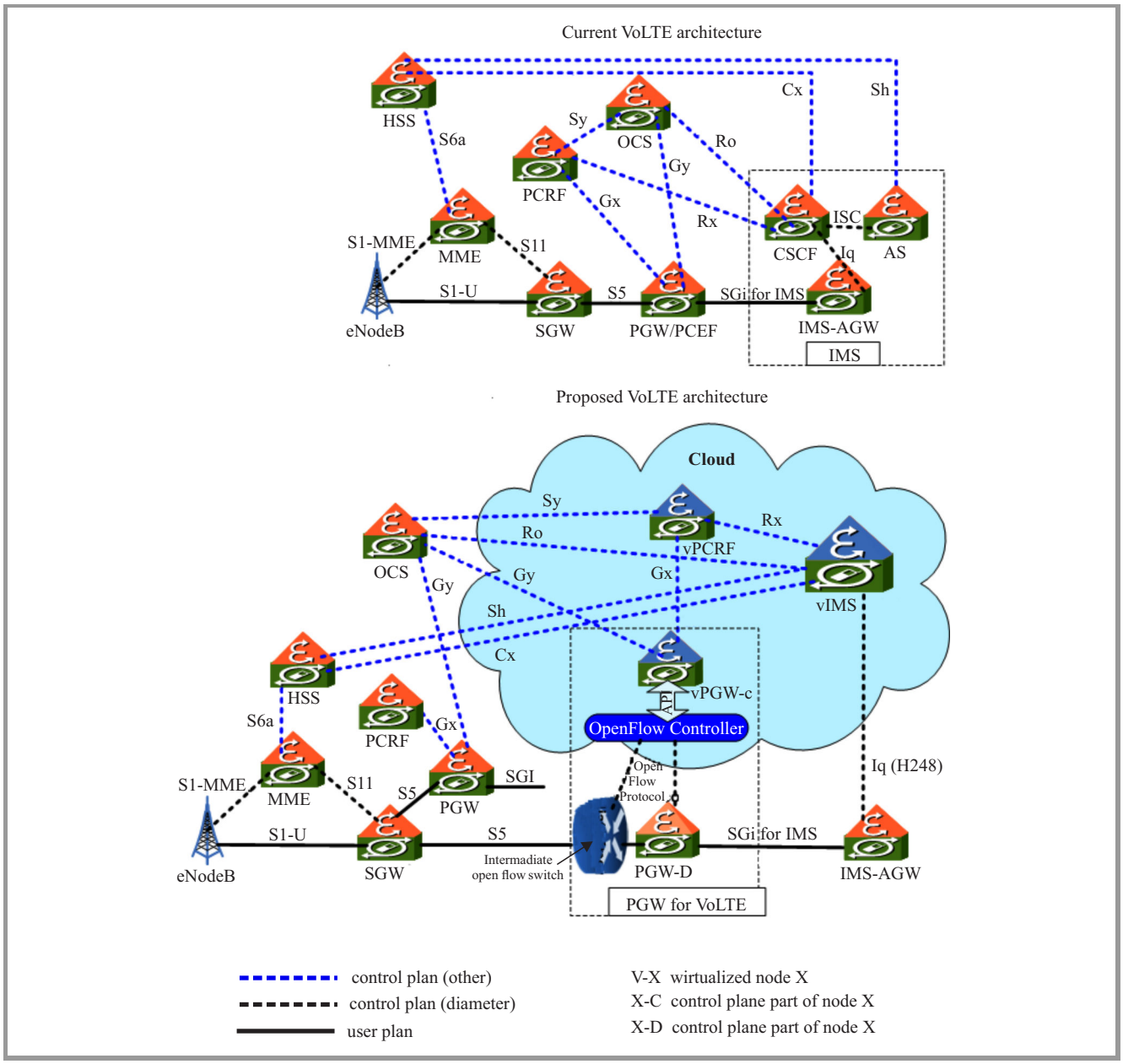


Fig. 5. Proposed network model to mitigate the impact of VoLTE

tual functions, which provides an abstraction of the network hardware, allowing the creation of virtual elements independent from the hardware layer. By bypassing this Hypervisor, Telefonica's NFV Lab has considerably improved the throughput of their virtualized nodes [28]. Such a bypass was performed using the Intel Single Root Input and Output Virtualization (SR-IOV) feature, which allows direct communication between virtual machine and hardware. Nevertheless, this change will have a major impact on the scalability of network functions [29], which is one of the strongest flexibility aspects that NFV technology provides.

Virtualizing the PGW control part allows the move of PCEF functionality to a virtualized domain. But based on Telefonica experience, the forwarding part should be kept out of NFV at this stage of NFV maturity cycle. The authors should look then for a solution to decouple the control and forwarding of the PGW to fulfill proposed requirement, pertaining to virtualizing the PGW control function. The Software Defined Networking (SDN) aims to separate the control and the forwarding in the networking domain. The NFV and SDN are independent approaches [25]. A wide range of research projects consider the SDN as a good enabler for NFV, as it allows the virtualization of the separated control plane software of the network functions [25]. The NFV can achieve the virtualization benefits without requiring SDN implementation, but both technologies complement each other and offer highly flexible network functions [25]. In [30], the authors suggested an SDN architecture to the Evolved Packet Core (EPC), by moving

the control plane of the Serving Gateway (SGW) and PGW (SGW-C, PGW-C), and the MME functions into cloud environment. They replaced the data plane of core gateways by switches using OpenFlow protocol, which is an SDN protocol used to centrally control switches and traffic flows in a network where the control plane is centralized and decoupled from the physical layer. This proposal seems to be the best solution for the demanded control plane decoupling, but only the PGW decoupling part will be adopted. The SGW and the MME decoupling will not be considered, and the communication between PGW in SDN architecture and SGW will still use the standard 3GPP S5 interface (Fig. 5). In order to reduce the number of connections between proposed model and the remaining live network elements, such as connections with radio access nodes a move the PGW only to authors' model is preferred.

The IMS has many components and most of them software-based control functions [25], such as CSCF function. In this architecture, all the IMS control functions will be included. As their migration will not have any impact on the existing services and nodes, their performance degradation can be resolved by scaling them up, and they can benefit from NFV advantages. The AS is one of those IMS control functions; where the facilities that NFV environment can offer to mitigate the complexity of its implementation should be explored. Currently, adding different Application Servers to enable the big list of IMS services did not attract operators. The deployment and testing process of those services is time consuming, and require proprietary hardware. Nevertheless, operators don't have any warranty that those services will attract subscribers and generate profits. The IMS pragmatic approach was the ideal solution to revive the IMS concept, where the focus was first on enabling VoLTE, and then other services can be progressively enabled based on the requirements [31]. This pragmatic approach will work perfectly on a virtualized environment, providing the facility of rapid deployment, and consequently a shorter period to confirm the success or the failure of the deployed service [32]. In addition to reducing the investment cost and time, virtualizing the AS will allow the scaling down or the replacement of unsuccessful services. However, the successful ones will benefit from the immediate scaling up. The IMS has also a forwarding part, IMS-Access-Gateway (IMS-AGW), handling the transcoding and the media services. Similarly to the PGW forwarding function case, the IMS-AGW cannot be virtualized. Due to the lack of works proposing an SDN architecture for IMS control and forwarding functions, authors prefer to use the 3GPP architecture using Iq interface to connect the virtualized IMS control functions to IMS-AGW [33]. Figure 5 summarizes the proposed architecture and depicts the key steps mentioned above.

4.2. Applying the Proposed Model to the Massive VoLTE Migration Use Case

In presented model, the network elements are considered where scaling can overcome the degradation of their performance in NFV, and not bearing any business critical services, to move a significant part of Diameter signaling to virtualized network. The amount of LTE Diameter signaling transactions migrated to proposed model is evaluated by adding an NFV weighting factor $\alpha_i = \begin{bmatrix}\alpha_{i1} & \alpha_{i2} & \alpha_{i3}\end{bmatrix}$ on each element of the F_i functions, described in the case study presented in Subsection 3.3.

α_{ij} is determined by the following rules:

- $\alpha_{ij} = 1$ – means 100% of transactions will be migrated to the virtualized model, if the activities happen between elements in proposed model,
- $\alpha_{ij} = 0.5$ – means 50% of transactions migrated to the virtualized model, if the activities happen between two elements one of them in the model,
- $\alpha_{ij} = 0$ – means 0% of transactions migrated to virtualized model, if the activities happen between elements out of the model.

To calculate $S_{NFV}(x)$, the average number of Diameter transactions per second during busy hour at year x moved to the virtualized architecture, the NFV weighting factor to Eq. (1) will be added:

$$S_{NFV}(x) = \frac{1}{3600}\sum_{j=1}^{j=3}\sum_{i=1}^{i=9}\alpha_{ij}F_{ij},$$

$$S_{NFV}(x) = \frac{1}{3600}\begin{bmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \\ F_{41} & F_{42} & F_{43} \\ F_{51} & F_{52} & F_{53} \\ F_{61} & F_{62} & F_{63} \\ F_{71} & F_{72} & F_{73} \\ F_{81} & F_{82} & F_{83} \\ F_{91} & F_{92} & F_{93} \end{bmatrix}[\alpha],$$

$$S_{NFV}(x) = \frac{1}{3600}\left(N_v(x)\begin{bmatrix} p_2t_1\,p_2t_2 & 0 & \\ 0 & p_2t_5 & 0 \\ 0 & p_2t_6 & 0 \\ 0 & p_2t_7 & 0 \\ 0 & p_5t_8 & 0 \\ 0 & p_3t_4 & 0 \\ 0 & p_3t_9 & 0 \\ 0 & 0 & p_1t_3 \\ 0 & 0 & p_1p_4t_{10} \end{bmatrix} + N_d(x)\begin{bmatrix} p_2t_1 & 0 & 0 \\ p_2t_5 & 0 & 0 \\ p_2t_6 & 0 & 0 \\ p_2t_7 & 0 & 0 \\ 0 & 0 & 0 \\ p_3t_4 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}\right)[\alpha].$$

The matrix $[\alpha]$ is the matrix grouping the entire weighting factors, defined as:

$$[\alpha] = \begin{bmatrix} \alpha_{11} & \alpha_{21} & \alpha_{31} & \alpha_{41} & \alpha_{51} & \alpha_{61} & \alpha_{71} & \alpha_{81} & \alpha_{91} \\ \alpha_{12} & \alpha_{22} & \alpha_{32} & \alpha_{42} & \alpha_{52} & \alpha_{62} & \alpha_{72} & \alpha_{82} & \alpha_{92} \\ \alpha_{13} & \alpha_{23} & \alpha_{33} & \alpha_{43} & \alpha_{53} & \alpha_{63} & \alpha_{73} & \alpha_{83} & \alpha_{93} \end{bmatrix}.$$

For proposed model, the 3GPP interface corresponding to each activity from Table 1, and the graphical representation of each 3GPP interface in Fig. 5 could be extracted. By combining these inputs, the weighting factors are calculated. The obtained $[\alpha]$ matrix is as follows:

$$[\alpha_{Model}] = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0.5 & 0.5 & 0.5 & 1 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0.5 \end{bmatrix}.$$

The values of $S_{NFV}(x)$, number of Diameter transactions per second during busy hour migrated to NFV architecture, from year 2015 till 2021 are calculated and presented in Table 2.

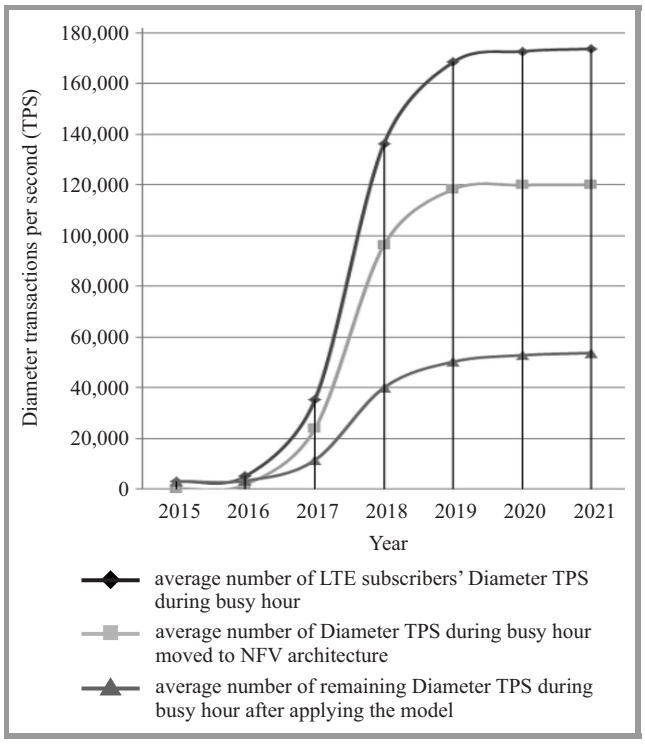


Fig. 6. Mitigating the impact of VoLTE after applying the model.

As shown in Fig. 6, 70% of the generated transactions per second during busy hour will be moved to NFV network. Consequently, operators' existing nodes should deal only with the remaining 30% of the generated transactions. Hence, operators will be relieved from the huge growth of Diameter signaling caused by VoLTE massive introduction.

5. Conclusion

In this paper, the impact of Diameter signaling growth on core network, especially the Diameter signaling traffic generated after massive VoLTE deployment are presented. To anticipate the impact induced by the challenging VoLTE deployment, a model to improve operators' readiness for the adoption of any kind of VoLTE introduction scenarios, either massive or gradual is proposed.

As shown in the model development, up to 70% of the total Diameter signaling transactions will be moved to the proposed architecture. This will relieve operators from the burden of the network cost and scalability, without impacting the running services and without overloading the existing network. Additionally, with this architecture, operators will be able to introduce NFV and SDN technologies and provide valuable contribution in the state-of-the-art of those technologies. NFV and SDN are most likely to redraw the design and implementation guidance for telecom environment in the coming years.

In future work, authors plan to investigate the financial aspects of this NFV based model, and evaluate the possibilities to extend the scope of virtualization to other network elements and services.

References

[1] "Tellabs "End of Profit" study executive summary", Tellabs, Jan. 2011.

[2] A. Aijaz, H. Aghvami, and M. Amani, "A survey on mobile data offloading: technical and business perspectives", *IEEE Wirel. Commun.*, vol. 20, no. 2, pp. 104–112, 2013.

[3] Y. Li, M. Qian, D. Jin, P. Hui, Z. Wang, and S. Chen, "Multiple mobile data offloading through disruption tolerant networks", *IEEE Trans. Mob. Comput.*, vol. 13, no. 7, pp. 1579–1596, 2014.

[4] R. Travis, "Oracle Communications LTE Diameter Signaling Index", 4th ed., Oracle Communications, Oracle, Oct. 2015.

[5] "Cisco Visual Networking Index: Forecast and Methodology, 2014–2019", Cisco, Tech. Rep., May 2015.

[6] J. Rankin, A. Costaiche, J. Zeto, and K. O'Neil, *Validating VoLTE: A Definitive Guide to Succssful Deployments*, 2 ed. IXIA, 2014.

[7] G. Blennerud, "Mobile Broadband: Busting the myth of the scissor effect", Tech. Rep., Ericsson, Dec. 2012.

[8] "Network Policy Control and the Migration to LTE", Sandvine, Tech. Rep., Nov. 2013.

[9] "Quality of Service (QoS) and Policy Management in Mobile Data Networks", IXIA, Tech. Rep. 915-2731-01 Rev. D, Dec. 2013.

[10] "Distributed Decisions in Network Policy Control", Sandvine, Tech. Rep., Nov. 2013.

[11] "New VoLTE Policy Management Architecture: Improve Subscriber Experience and Lower Cost of Growth", Cisco, Tech. Rep. C11-732158-0, Mar. 2015.

[12] J. Gillet and T. Hatt, "Smartphone forecasts and assumptions: 2007-2020", GSMA Intelligence, Tech. Rep., Sept. 2014.

[13] S. Pang, *Successful Service Design for Telecommunications: A Comprehensive Guide to Design and Implementation*. Hoboken, NJ: Wiley, 2009.

[14] "Voice over LTE: Challenges and Opportunities", Sandvine, Tech. Rep. Version 2.0, 2015.

[15] "Policy and charging control signalling flows and Quality of Service (QoS) parameter mapping", TS 29.213 V13.1.0, 3rd Generation Partnership Project (3GPP), Mar. 2015.

[16] "General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access", TS 23.401 V13.0.0, 3rd Generation Partnership Project (3GPP), Sept. 2014.

[17] "Policy and charging control architecture", TS 23.203 V13.3.0, 3rd Generation Partnership Project (3GPP), Mar. 2015.

[18] "Telecommunication management; Charging management; Diameter charging applications", TS 32.299 V13.0.0, 3rd Generation Partnership Project (3GPP), Mar. 2015.

[19] "IP Multimedia (IM) Subsystem Cx and Dx Interfaces; Signalling flows and message contents", TS 29.228 V12.0.0, 3rd Generation Partnership Project (3GPP), Sept. 2013.

[20] "Data for Operators Ranking connections, excluding cellular M2M, Total", GSMA Intelligence, Tech. Rep., Q4 2020.

[21] "Data for Operators Ranking connections, excluding cellular M2M, 4G", GSMA Intelligence, Tech. Rep., Q2 2015.

[22] "% connections, excluding cellular M2M, Prepaid, World", GSMA Intelligence, Tech. Rep., Q4 2015.

[23] Y. Ouyang and M. H. Fallah, "An analysis of traffic and throughput for UMTS packet core networks", in *Research, Practice, and Educational Advancements in Telecommunications and Networking*, M. R. Bartolacci and S. R. Powell, Eds. IGI Global, 2012, pp. 82–107.

[24] "IP multimedia call control protocol based on Session Initiation Protocol (SIP) and Session Description Protocol (SDP); Stage 3", TS 24.229 V13.0.0, 3rd Generation Partnership Project (3GPP), Sept. 2015.

[25] "Bringing Network Function Virtualization to LTE", Tech. Rep., 4G Americas, Nov. 2014.

[26] "MWC 2015 End-to-End NFV Architecture demof", Telefonica, Brocade, Cyad, Intel, and Redhat, Tech. Rep., GSMA Mobile World Congress, Spain, Mar. 2015.

[27] D. Mavrakis, "MWC 2015: NFV gets real", Tech. Rep. TE0006-001035, Ovum, Mar. 2015.

[28] "End to End Network Function Virtualization Architecture Instantiation", Intel, Brocade, Cyan, Red Hat, and Telefonica, Tech. Rep., Feb. 2015.

[29] Z. Huang, R. Ma, J. Li, Z. Chang, and H. Guan, "Adaptive and scalable optimizations for high performance SR-IOV", in *Proc. IEEE Int. Conf. Cluster Comput. CLUSTER 2012*, Beijing, China, 2012, pp. 459–467.

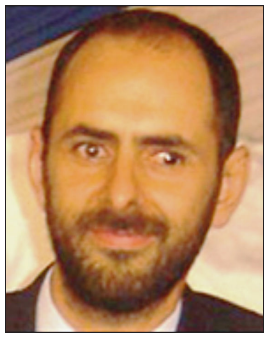
[30] S. B. H. Said, M. R. Sama, K. Guillouard, L. Suciu, G. Simon, X. Lagrange, and J.-M. Bonnin, "New control plane in 3GPP LTE/EPC architecture for on-demand connectivity service", in *Proc. 2nd IEEE Int. Conf. Cloud Netw. CloudNet 2013*, San Francisco, CA, USA, 2013, pp. 205–209.

[31] Y. Jouihri and Z. Guennoun, "Best selection for operators starting LTE deployment towards voice services" in *Proc. Int. Conf. Multimed. Comput. & Syst. ICMCS 2012*, Tangier, Morocco, 2012, pp. 568–573.

[32] G. Brown, "Virtualization of IMS & VoLTE in Mobile Operator Core Networks", Mavenir, Tech. Rep., Mar. 2013.

[33] "IMS Application Level Gateway (IMS-ALG) IMS Access Gateway (IMS-AGW); Iq Interface", TS 29. 334 V10.2.0, 3rd Generation Partnership Project (3GPP), Sept. 2011.

Youness Jouihri received his engineering degree in Electronics and Telecommunications from Mohammadia School of Engineering (EMI), Rabat, Morocco in 2005. His research interests include the LTE limitations, voice service evolution, network function virtualization, and cloud convergence between IT and telecom domains. He has a broad international experience in telecom consultancy services.
E-mail: Jouihri.youness@gmail.com
Electronics and Communications Laboratory (LEC)
Mohammadia School of Engineering (EMI)
Rabat Mohammed V University (UM5R)
Rabat, Morocco



Zouhair Guennoun (IEEE senior member) received his engineering degree in Electronics and Telecommunications from Montefiore Institute, Belgium in 1987. He received his M.Sc. and Ph.D. degrees in 1993 and 1996, in Communication Systems from Mohammadia School of Engineering (EMI). He visited, during 1990-1994 the Centre for Communication Research in Bristol University, UK. From 1996, he is working at EMI as a Professor Lecturer and currently in charge of the research laboratory in LEC. His fields of interest are digital signal, speech and image processing, error control coding, communication systems.
E-mail: zouhair@emi.ac.ma
Electronics and Communications Laboratory (LEC) Mohammadia School of Engineering (EMI)
Rabat Mohammed V University (UM5R)
Rabat, Morocco

Youssef Chagh received his engineering degree in Telecommunications from INPT, Rabat, Morocco, in 2005. His fields of interest are IP multimedia subsystem, network function virtualization, and 5G networks.

E-mail: chagh.youssef@gmail.com
Electronics and Communications Laboratory (LEC)
Mohammadia School of Engineering (EMI)
Rabat Mohammed V University (UM5R)
Rabat, Morocco

Driss Zahi received his M.Sc. degree in Electrical Engineering and Telecommunications from Paris ISEP. He also received a M.Sc. degree in Information and Knowledge Systems and an other in Microeconomics from the Business School of the Sorbonne University. He received the Ph.D. degree from the Sorbonne University in Economy and Management Sciences. he has spent more than 25 years in the telecommunications industry as a consultant for large operators and vendors. He also delivered various training curriculum in Universities.
E-mail: driss.zahi2265@gmail.com
GREGOR Research Lab
Pantheon-Sorbonne University
Paris, France

Paper

Application of Social Network Inferred Data to Churn Modeling in Telecoms

Witold Gruszczyński[1] and Piotr Arabas[2,3]

[1] *Codeflip, Warsaw, Poland*
[2] *Institute of Control and Computation Engineering, Warsaw University of Technology, Warsaw, Poland*
[3] *Research and Academic Computer Network (NASK), Warsaw, Poland*

Abstract—The subject of this work is the use of social network analysis to increase the effectiveness of methods used to predict churn of telephony network subscribers. The social network is created on the basis of operational data (CDR records). The result of the analysis is customer segmentation and additional predictor variables. Proposed hybrid predictor employs set of regression models tuned to specific customer segments. The verification was performed on data obtained from one of the Polish operators.

Keywords—churn reduction, classification, social networks.

1. Introduction

One of the main challenges faced by telecommunications companies is to stop the migration of customers. This process, called *churn*, is in a great extent caused by the operators themselves, who offer attractive conditions for new users to attain dominance in highly competitive market. The issue is so important that, as shown by statistics, client acquisition may be several times more expensive [1], [2] than churn prevention, and nearly 80% of users surveyed admitted they have changed provider at least once [2].

A typical method to reduce churn are advertising campaigns encouraging the purchase of new products, or participation in the loyalty program. To achieve positive effect such an offer should be addressed to carefully selected group of customers to reduce campaign cost, and to avoid bothering loyal clients with too numerous contacts.

The telcos possess abundant data related to the customers and their use of services. The usage information is available mainly in the form of Call Detail Records (CDR) and is believed to be valuable input for users behavior modeling. The classic approach is to use the methods defined broadly as data mining [3] to predict subscribers decisions based on changes in the use of their services. This allows to select the group of customers prone to churn, which then can be addressed adequately prepared offer.

The subject of the work described in this paper is to enrich the standard techniques with data describing social links between subscribers. This information may be obtained by building social network graph basing on CDR data.

The remainder of the article is organized as follows: Section 2 presents a description of the problem and the proposed solution, taking into account previous work on similar issues. In Section 3 the concept of predictive model and its augmentation with social network analysis is outlined. Section 4 briefly presents implemented programs and software packages used to carry out the research. Next Section discusses results of the model validation and finally Section 6 concludes on the research.

2. Problem Description

2.1. Purpose of Work

The aim of the presented work was to create set of models, which could predict if a particular customer is prone to churn. The only input for these models were operating data available in the form of CDRs. In the absence of any other customer data (e.g. information on the place of residence, age or type of work) it was especially important to take full advantage of all its aspects. Authors' previous work [4] demonstrated that applying regression modeling to information derived from CDRs have given promising results. The idea presented here is to introduce segmentation basing on the structure of the social network modeling customers relations to construct a specific and thus more efficient predictive models in each of segments. Additional data obtained during the analysis of the social network and not used for segmentation is included as an input of models. Information inferred through analysis of social graph should augment usage statistics with details describing social linkage between users and may be considered as an attempt to reproduce (or replace) unavailable user related data. Authors believe that so derived variables will increase resolving power of models as they should be unrelated to the base set of variables.

2.2. Similar Solutions and Related Work

A typical approach to the problem of predicting customers behavior is to use different kinds of classifiers, including regression [5], decision trees [6] or genetic algorithms [6]. These techniques have long been known and used in many fields, however, as indicated by a number of

authors, the use of one of them does not guarantee to sufficiently exploit the information hidden in the telecoms operation data [6]–[8]. Application of social networks analysis [9], [10] is proposed by many authors [11], [12], but usually it is employed to analyze the relationship between subscribers only [13]–[15].
In the method proposed in this paper both kinds of data are used: common statistics of traffic data, as well as those resulting from social network analysis. This approach is similar to those presented in [16]. A major difference, however, is that customers are segmented prior to modeling making it possible to identify a number of specific models and thus to increase their accuracy.

2.3. CDR Data

The data were provided by one of the Polish wired telephony operators and contained usage records within three months collected out of a selected part of the operator network. The set consisted of 130 million records generated by 315,000 users. Due to the confidentiality restrictions data have been anonymized by deleting some fields and encoding subscriber and recipient numbers. As a result only date, time, call duration, tariff ID and customer ID were available. The latter information allowed to associate the subscriber with all phone numbers owned.
Although the anonymization is deterministic (i.e. the IDs are coded each time the same) there is some loss of information. Especially it is impossible to distinguish some well known numbers[1], local and long distance calls or interpret tariffs. For similar reasons it was not possible to obtain any additional customer data. The most important problem was lack of information describing user status – a user was considered churner when no CDRs were generated for longer than three weeks. The available information is much more constrained than in the typical analysis carried out by the network operator, who has full access to the data. However, it resembles case of prepaid services, where customer information is generally not available.

3. Construction of the Model

The main idea was to derive new information through the analysis of the social network constructed on the basis of CDR data. This information can provide additional input for classifiers, it can also be used for customer segmentation.

3.1. Social Network Reconstruction

The main problem was the incompleteness of the data, resulting in a significant portion of connections leading outside the network. It may be attributed to the limited customer base, but also to the fact that they were collected only from a part of network. As a result the social network built (internal network) covered less than 281,000 of the approx. 299,000 individual users. The corporate users, i.e. those having more than one phone number were deliberately omitted. It is supposed that the churn mechanism may be different in their case, so other, perhaps individual churn reduction methods should be used.
To increase the number of analyzed users a bipartite network was constructed – a connection between two users was added if they call the same number outside the provider network (for extended discussion of social network reconstruction see e.g. [17]). So constructed network consisted of 13711 subscribers, the remaining 4178 users could not be included in any network.

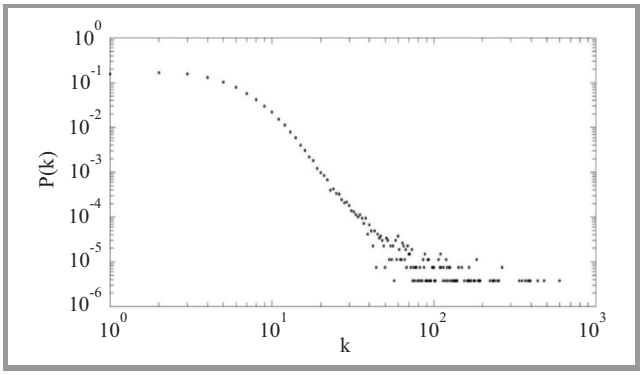


Fig. 1. Inner network node degree distribution.

Analysis of degree distribution in the internal network (see Fig. 1) suggests power-law with characteristic exponent $\alpha = 2.75$ – a value slightly above reported for wired telephone networks [18]. Scale-free nature of the network, allows to expect typical social networks phenomena, especially the nodes with an extremely high degree among majority of low degree nodes. The existence of as many as seven separate connected components (see Table 1) is easily explained by the limited range of the network.

Table 1
Connected components of the inner network

Component	1	2	3	4	5	6	7
Users	80007	67853	38410	26289	25240	22525	20419
%	28	24	14	9	9	8	7

It is probable that in a bigger set of data much more calls can be terminated internally reducing the number of connected components, a more likely outcome would be, however, a greater coverage of the internal network. Connected components can be used for a natural segmentation of users. In the contrast to typical segmentation methods this one does not use any additional user information to explicitly include them into predetermined classes (e.g. division by age, gender, place of residence, etc.). Instead of this unavailable data it uses social contacts characteristic.

[1] An example may be customer information office typically considered as one of numbers routinely consulted before deciding to leave the network.

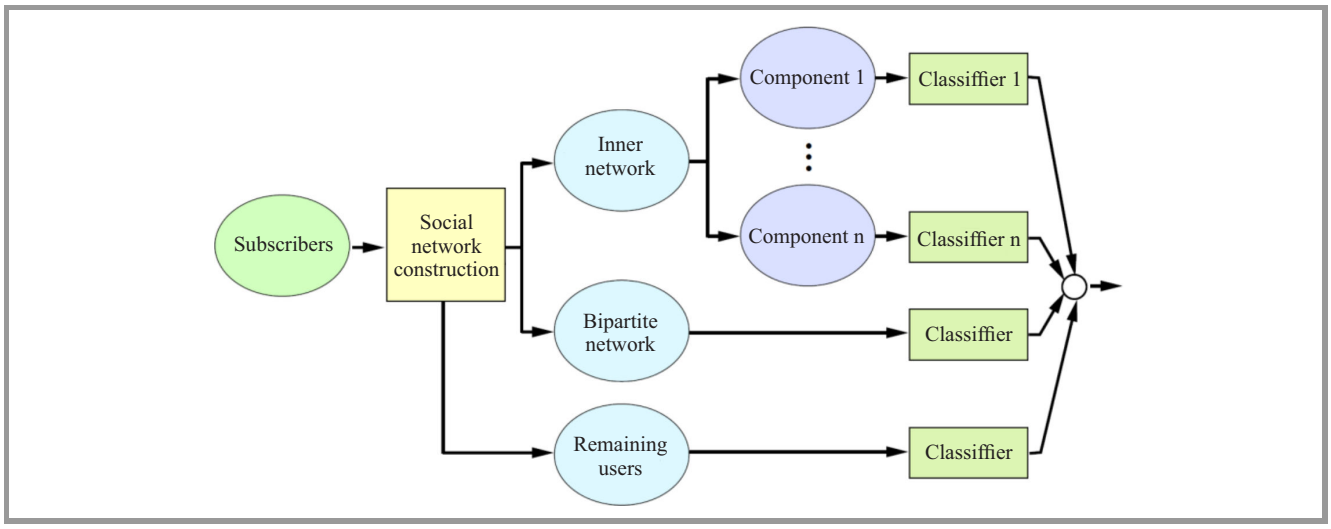


Fig. 2. Hybrid model.

3.2. Regressive Model

Regression provides a reference point for the hybrid model described in the Section 4. The hybrid model is in fact a combination of several specific regression classifiers working for separate segments of customers, hence the principles of their construction and use are the same. In both cases logistic regression was used to create classifier dividing all customers into two classes, with these being prone to churn marked as "positive". A convenient feature of using regression is possibility to evaluate the selection of input variables by examining their significance levels. It should be remembered that as a non-linear classifier logistic regression allows a slightly better distinction of classes than the linear methods, however it still remains quite sensitive to the selection of predictor variables.

The construction of the regression model consists of two phases. First the predictor variables must be selected, then model parameters are identified and classifier quality is assessed. Data used for model identification covered three weeks. To detect changes in the customer behavior three time windows with a width of the week are defined. Predictor variables calculated in these windows, were the statistics corresponding to different types of user activity – e.g. number of failed connections, number of connections of the specified type, cost of calls (i.e. pulses counted). Most of the variables were differences between successive weeks, which allowed to eliminate dependence between subsequent variables. It is crucial for the effectiveness of a classifier to select variables of adequate predictive power – apart from avoiding correlation, it could be done through experiments and evaluation of their significance level. Insignificant variables were removed, and replaced with another variables (if available).

Another problem is the proper choice of samples constituting the training set. It should be noted that the number of users leaving the network was relatively small and amounted to slightly more than 3% in the analyzed period. So uneven distribution of the classes in the training data usually leads to classifiers that work correctly only for the more numerous class. For this reason, the training set was created by sampling of both classes independently, so that the data contained 20–25% of churners.

3.3. Hybrid Modeling

Hybrid model uses seven connected components of the internal network to segment most (namely 280,743) users. In addition social ties of 13711 users may be modeled by bipartite network. In this way, it is possible to build eight customized regression classifiers, which hopefully should better reflect specificity of each segment, and be more effective than basic classifier described in Subsection 3.2. The improvement may be attained not only by preselecting customers and so making input data more homogeneous, but also by using new predictor variables derived by the social network analysis. The variables considered were node degree, closeness or page-rank, as well as the number of node neighbors, who left the network recently. The last variable is a natural, and widely used churn indicator, which calculation has little algorithmic complexity. Relatively small number (4178) of subscribers cannot be connected into any social network and so remains outside 8 segments. In their case it is only possible to build a classifier using basic predictor variables like the one described in Subsection 3.2. The complete diagram of the hybrid model is presented in Fig. 2.

4. Software

To facilitate the prepossessing of data, and the identification and verification of classifiers custom programs were prepared. Visualization of graphs was performed with help of LaNet-vi package [19]. MS SQL Server was used to store initial CDR data and results of classification. Pre-

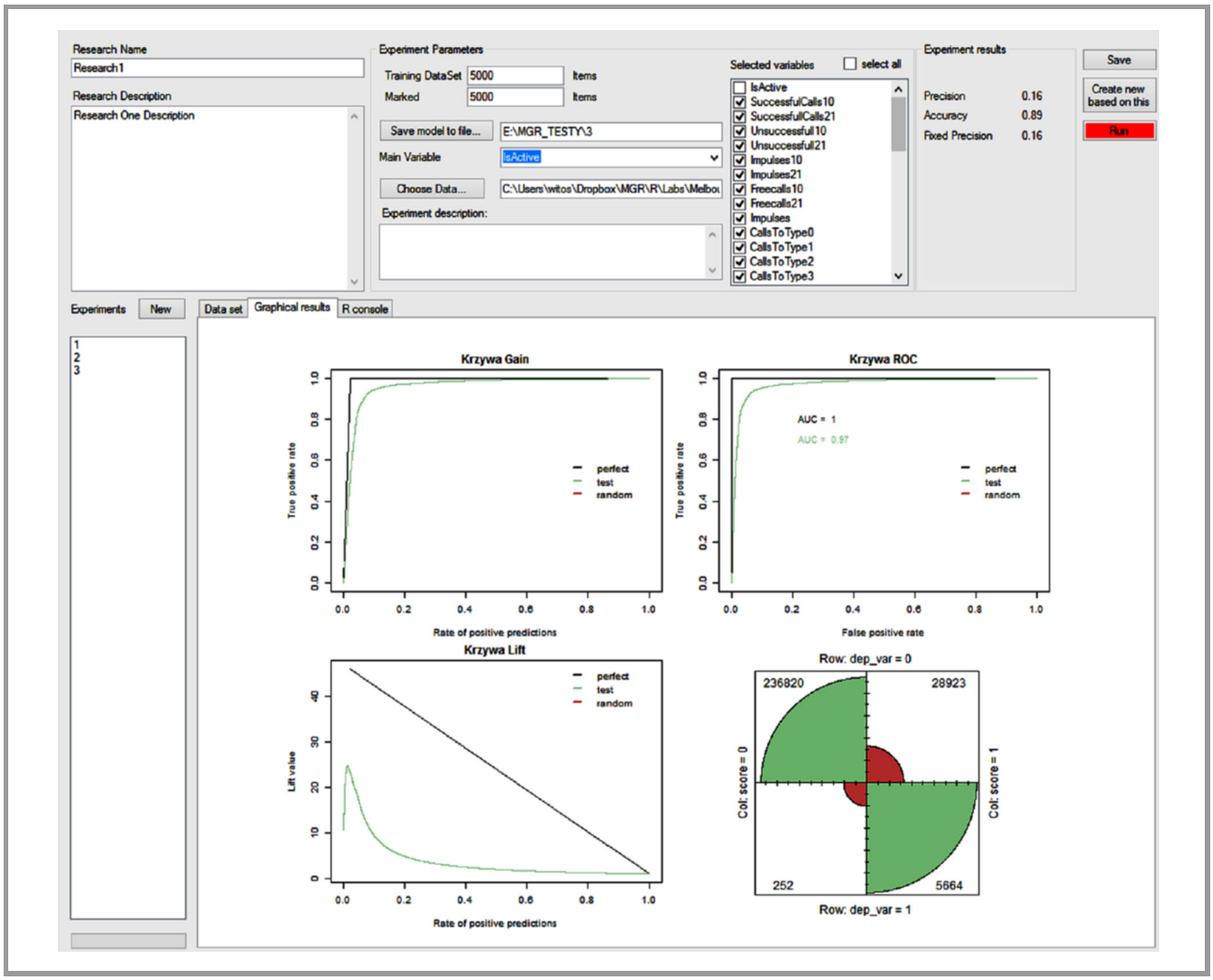


Fig. 3. Model identification and validation GUI.

processing, done mostly using capabilities of the MS SQL Server included CDR compression, separation of corporate users and removal of anomalous entries. Specifically, nodes generating thousands of calls which are supposed to be kind of automatic network testers. Compression of records may be achieved mainly by removing fields that were cleared during the data anonymization. Removal of automatically generated calls was necessary as they could distort user activity statistics. To speed up creation and analysis of social networks appropriate indexes were added. The most important part of the developed software was designed to carry out the identification and validation of models. Its main functionalities are importing data from the database and assisting user in the selection of predictor variables. Identification of classifier (logistic regression) parameters was implemented with help of R package [20] and was performed on a training data set prepared through sampling. After identifying the model it is possible to verify it on a separate data set. Program implements simple to use graphical interface – e.g. Fig. 3 presents verification of classifiers tab.

5. Model Identification

5.1. Verification Methods

In order to verify the results of identification a number of classification experiments on a separate datasets was carried out. The results can be illustrated in the form of a confusion matrix, being a table containing the numbers of correctly and incorrectly classified samples in both classes – see Table 2. The true negative (TN) indicates the number of samples correctly classified as having no investigated characteristic – in this case the users who chose to remain subscribers of the telecoms. False positive (FP) refers to samples incorrectly classified as leaving the network (type

Table 2
Confusion matrix

Classification result		N	P
Original classes	N	TN	FP
	P	FN	TP

1 error). Samples labeled false negative (FN) are users incorrectly classified as non churners (type 2 error), and finally, true positive (TP) are correctly detected users that decided to leave the network.
Confusion matrix contains a full description of results, however, it is not convenient when several models need to be compared. To make it easier F1 measure, defined by the following formula, was used:

$$F1 = \frac{2TP}{2TP + FN + TP}, \quad (1)$$

where symbols TP, FN i FP are defined according to the Table 2. It may be visible, that F1 does not consider true negatives allowing to minimize the overhelming influence of dominant class [21].

5.2. Regressive Model

As mentioned in the Subsection 3.2 basic regressive model is a reference point for the hybrid modeling. The following predictor variables were used:

- difference of the number of successful connections,
- difference of the number of failed connections,
- difference of the number of calls of specific types,
- difference of the number of calls calls on weekends,
- difference of the number of pulses,
- difference of the number of free calls.

Differences are calculated in the three consecutive weekly periods. The call was considered unsuccessful if it lasted less than 2 s. Connection types were read from the appropriate column in CDRs, however unfortunately any interpretation could be associated to them. Using logistic regression allowed to identify relatively efficient classifier – see the verification results in Table 3 and Fig. 4 which is a graphical representation of the confusion matrix (cf. Table 2) – parts of pie chart correspond to the cells of confusion matrix, note that FP radius is scaled by TN and FN is scaled by TP.

Table 3
Identification of basic regression classifier

Training set cardinality	12000
Number of churners in training set	3000
Validation set cardinality	285017
F1 value	0.558

The results of the verification show that the model performs better for the more numerous class, while the most visible shortcoming is the high number of errors of the first type – in fact, there are more false positives then properly detected

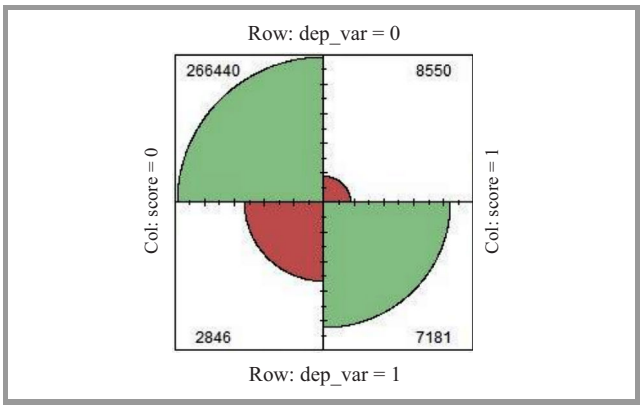


Fig. 4. Graphical representation of basic regression classifier confusion matrix.
(See color pictures online at www.nit.eu/publications/journal-jtit)

churners. This means that taking preventive measures – e.g. advertising campaign would cost twice as much as in the case of perfect information, however would be far more effective than untargeted action. The number of undetected churners – 2846 (which amounts to 28%) seems to be less alarming. However, it also shows that better modeling is necessary.

5.3. Hybrid Model

Hybrid model consisted of nine logistic regression classifiers operating on separate segments of customers built of internal network components (7 segments, see Table 1), bipartite network and set of remaining users. Predictor variables of basic regression model (see Subsections 3.2 and 5.2) have been supplemented with the data resulting from the social network analysis:

- node degree,
- closeness,
- page-rank,
- number of node neighbors who left the network.

5.4. Verification of Hybrid Model Components

The results of the identification and verification of the classifier for the components 1–7 are presented in Table 4 and Fig. 5a-g respectively. Comparing the value of the F1 measure and analyzing confusion matrices for the first component it can be noticed that the classifier is much more effective than simple regression (see Subsection 5.2). This results mainly from limiting the scope to the group of users sharing common features. Augmenting the set of predictor variables is another source of improvement – it allows to introduce new, important information derived through analysis of social ties between customers.
The results of verification for the component 2 are shown in Table 4 and Fig. 5b. Unfortunately, they are not as good as for the component 1, and are even slightly worse than the results obtained using the basic regression classi-

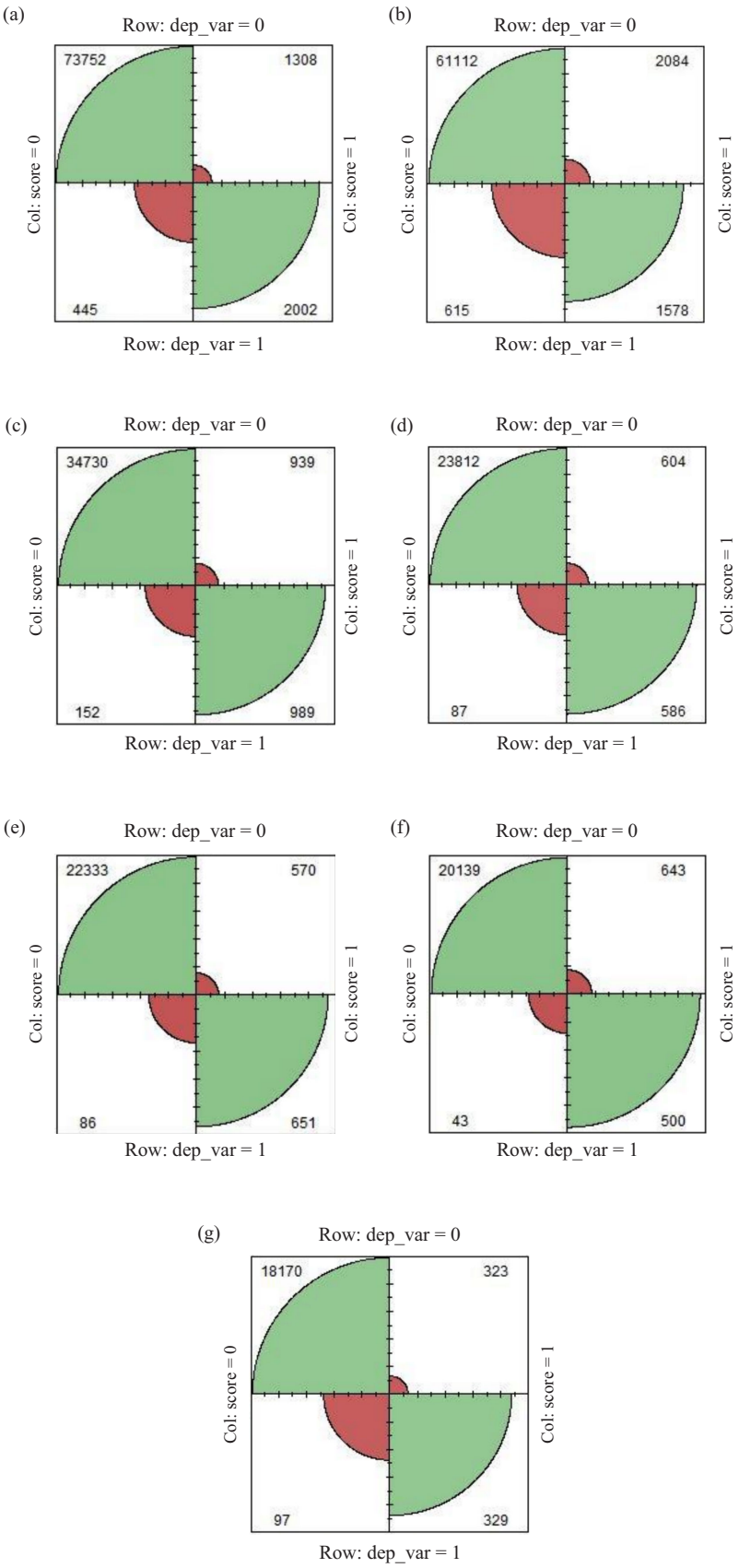


Fig. 5. Graphical representation of the confusion matrix for the components: (a) 1, (b) 2, (c) 3, (d) 4, (e) 5, (f) 6, (g) 7.

Table 4
Identification of the classifier for the components 1–7

Component	1	2	3	4	5	6	7
Training set cardinality	2500	2500	1600	1200	1600	1600	1500
Number of churners in training set	500	500	400	300	400	400	300
Validation set cardinality	77507	65353	36810	25809	23640	36810	18919
F1 value	0.696	0.539	0.645	0.629	0.665	0.593	0.610

fier. Presumably the segment consists of different kinds of users and requires further division or the use of more elastic classifier. Figure 5c-g shows the results for the other components. They are worse than for the component 1, however, smaller component size should be taken into consideration, it must be also noticed that the effectiveness of prediction is greater than the reference method.

Table 5
Identification of the classifier for the bipartite network

Training set cardinality	1500
Number of churners in training set	500
Validation set cardinality	12211
F1 value	0.507

Seven classifiers described above cover 89% of users. For the remaining 17889 users two additional classifiers were built. The first one was based on bipartite network containing 13711 users. Verification results for this model are shown in Table 5 and Fig. 6. They are much worse than in the previous cases. It may be explained by the fact of weaker ties between customers building it and the provider as their calls are terminated outside. For this reason, they are more likely to leave the network. In fact churn rate is significantly higher in this segment – 15.48% compared to 3.68% in the entire network. It should therefore be expected that special methods are required for churn prediction as well as for its limitation in this segment.

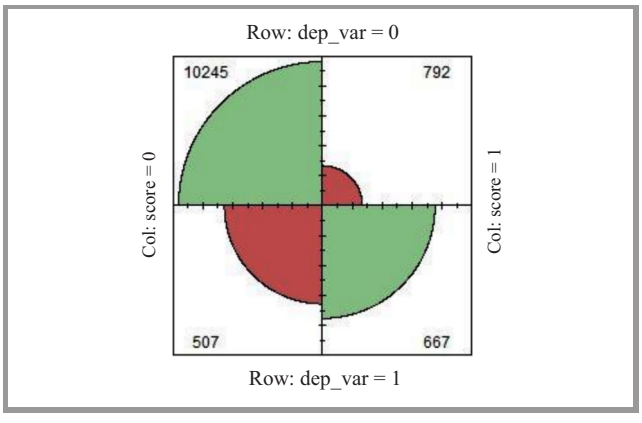


Fig. 6. Graphical representation of the confusion matrix for the bipartite network.

The last segment is formed of users that cannot be connected into any network. It can be assumed that they are also not similar in any other way, explaining relatively week

Table 6
Identification of the classifier for the remaining users

Training set cardinality	1200
Number of churners in training set	300
Validation set cardinality	2978
F1 value	0.466

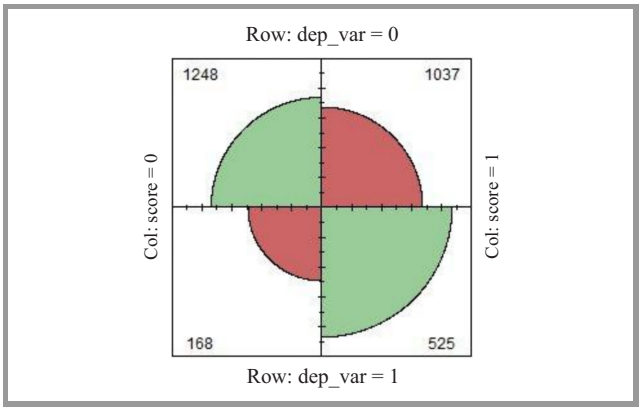


Fig. 7. Graphical representation of the confusion matrix for the remaining users.

results of prediction – see Table 6 and Fig. 7. However, it is the smallest of the segments and its impact on the effectiveness of the hybrid model is negligible.

5.5. Hybrid Model Verification

Classification by the hybrid model is equivalent to proper use of the nine described classifiers, so verification outcome is in fact the sum of the results for individual models (see Tables 7 and 8) giving little, but visible improvement

Table 7
Identification summary for hybrid model

Training set cardinality	14800
Number of churners in training set	3500
Validation set cardinality	283868
F1 value	0.599

Table 8
Hybrid model confusion matrix

Classification result		N	P
Original classes	N	265541	8300
	P	2200	7827

Table 9
Effectiveness of hybrid model components, "B" stands for bipartite network and "R" for users outside any network; bold numbers are predictors performing below average

Components	1	2	3	4	5	6	7	B	R
F1	0.696	**0.539**	0.645	0.629	0.665	**0.593**	0.610	**0.507**	**0.466**

over basic regressive model. The biggest problem is still the excessive number of false positives. It is worth mentioning however, that classifiers for the components 1 and 3–7 perform better than basic classifier.

While investigating sources of imperfections the worst operating classifiers should be examined first (see Table 9). The lowest value of the F1 measure is achieved by the last classifier, acting for users who do not belong to any social network. As noted earlier this segment is related to a small group of users (a little more than 1%) so its impact on the overall classification effectiveness is also low. A slightly larger group are the subscribers forming bipartite network. They are weakly bound to the service provider and therefore it can be difficult to detect their intention of leaving. The case of component 2 seems to be different. It represents more than 20% of all users, so it cannot be easily neglected. A hint can be components visualization by LaNet-vi software using k-cores decomposition – see Fig. 8 for component 1, and Fig. 9 for component 2.

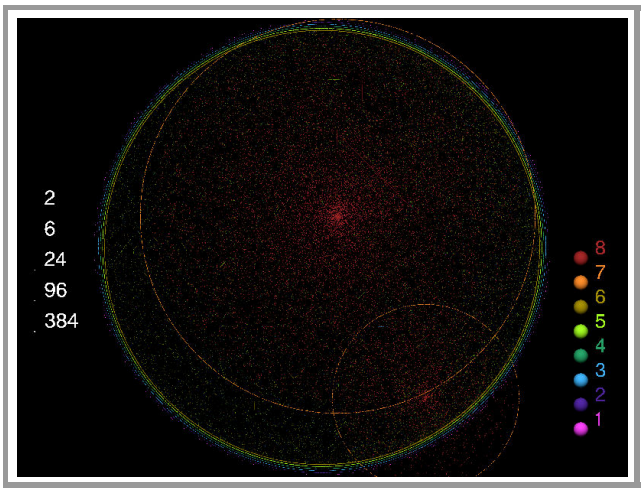


Fig. 8. Component 1 k-cores.

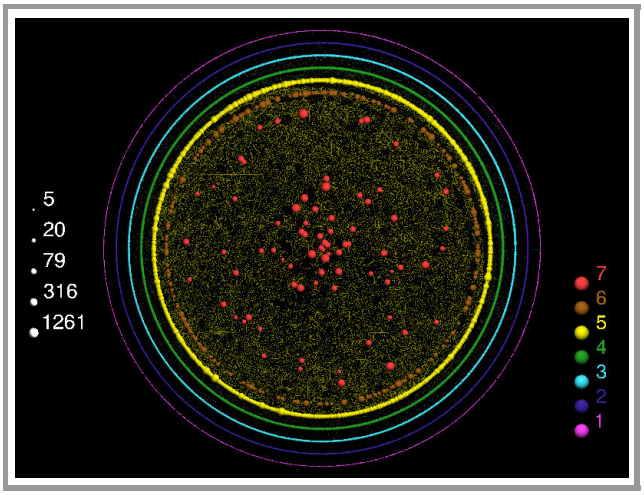


Fig. 9. Component 2 k-cores.

The k-core is a set of vertices of the network, each of which has at least k connections with its neighbors. It is important to note that this is not equivalent to a subset of nodes that had degree k in the original graph. Instead a k-core can be found by recursive rejection of vertices with degree lower than k. The colors in Figs. 8 and 9 represent the k-core number and circle diameter initial degree of the node. The difference between the component 1 and 2 is visible – in the first one a majority of low degree nodes encircles the center consisting little number of higher degree nodes. In component 2 much more nodes belong to the higher k-cores while lower degree ones are concentrated in several groups. It may be suspected that topological differences may indicate distinct characteristics of users. The analysis of node distribution among k-cores (see Fig. 10) shows, that while most of them follows similar, growing line, components 2 raises rapidly for $k = 5$. For the component 6 most of nodes are concentrated in the first three cores. Both components 2 and 6 are most problematic ones with F1 measure of 0.539 and 0.593 suggesting that differences in the degree structure may have important effect on churn.

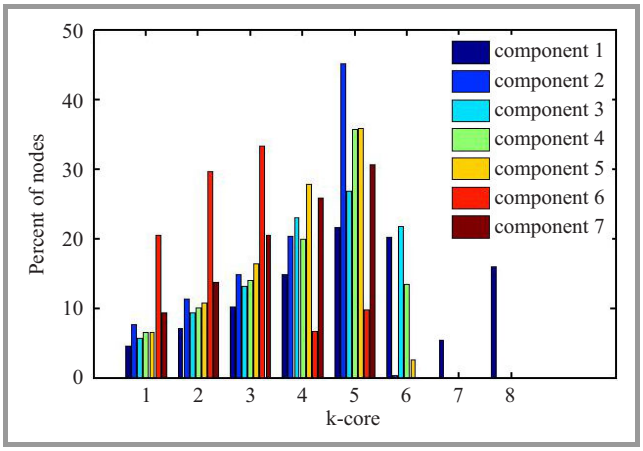


Fig. 10. Cardinality of k-cores for the inner network components.

To investigate it further triadic census[2] and clustering coefficient were calculated for all segments – see Table 10. It may be seen that component 2 differs from the remaining

[2]As the graph analyzed is undirected 4 types of triads are possible.

Table 10
Triadic census of model components

Component	Full triads	Forbidden triads	Clustering coefficient
1	75 550	6 477 654	0.0751
2	34 972	6 239 851	0.0691
3	26 205	3 037 690	0.0769
4	16 484	1 371 366	0.0820
5	15 852	1 208 919	0.0910
6	7 473	742 104	0.0913
7	9 972	1 191 967	0.0780
Bipartite	1 213 494 564	723 940 498	0.7384

inner network components as it has approximately twice as much forbidden triads, i.e. triads with connection between two nodes missing. Such a composition is reflected by the lowest value of clustering coefficient. Different structures of ties between users – lack of triangle closure may suggest that they are using a different kind of communication channel and therefore are more prone to churn. On the other hand the users may have no reason to communicate – e.g. their professional activity may require only contacts with a kind of a central office. In both cases it seems that local (i.e. user) value of the clustering coefficient may convey some information of user willingness to stay and could be considered the additional predictor variable.

6. Conclusion

The results of the presented work show that the social network analysis can be a valuable tool for customer segmentation. The approach is particularly convenient when available information is limited mainly to CDRs making it applicable for prepaid services, or a situation when the access to customer data is restricted due to the confidentiality. An additional benefit from social network analysis is the possibility to augment the set of predictor variables, with ones carrying new information, usually uncorrelated with simple statistics, and thus improving the classification efficiency.

There is no doubt that these results, although promising, are far from ideal. Particularly worrying is the high number of errors of type 1 (FP). It is possible to propose two ways to solve this problem. One of them may be fine tuning classifiers, especially by choosing predictor variables in each segment separately. Confirmation of the validity of this approach is good performance of the classifier for the component 1. The second way is to use more sophisticated classifiers – it must be stressed that the logistic regression is used because of its simplicity and ease of analysis. In this sense, presented work is a demonstration of the applicability of social network analysis to user segmentation and new predictor variables definition rather than a final assessment of its effectiveness.

Finally, to consider the analysis of the social network itself – as it was mentioned the work uses relatively simple indicators. It is possible that a deeper analysis of the characteristics of the network, in particular topological, or related to the connection dynamics [22] may result in new significant predictor variables as well as valuable observations on user behavior.

References

[1] H. Kim and C. Yoon, "Determinants of subscriber churn and customer loyalty in the Korean mobile telephony market", *Telecommun. Policy*, vol. 28, no. 9–10, pp. 751–765, 2004.

[2] C. Borna, "Combating customer churn", *Telecommun. – Americas Edit.*, vol. 34, no. 3, pp. 83–85, 2000.

[3] G. M. Weiss, "Data mining in telecommunications", in *Data Mining and Knowledge Discovery Handbook*. Kluwer Academic, 2005.

[4] W. Gruszczyński and P. Arabas, "Application of social network to improve effectiveness of classifiers in churn modelling", in *Proc. Int. Conf. Computat. Aspects of Soc. Netw. CASoN 2011*, Salamanca, Spain, 2011, pp. 217–222.

[5] T. Mutanen, "Customer churn analysis – a case study", VTT Research Report no. VTT-R-01184-06, 2006 [Online]. Available: http://www.vtt.fi/inf/julkaisut/ muut/2006/ customer_churn_case_study.pdf

[6] V. Yeshwanth, V. Vimal Raj, and M. Saravanam, "Evolutionary churn prediction in mobile networks using hybrid learning", in *Proc. 24th Florida Artif. Intell. Res. Soc. Conf. FLAIRS-24*, Palm Beach, FL, USA, 2011, pp. 471–476.

[7] T. Sato, B. Q. Huang, Y. Huang, and M. T. Kechadi, "Local PCA regression for missing data estimation in telecommunication dataset", in *11th Pacific Rim Int. Conf. Artif. Intell. PRICAI 2010*, Daegu, Korea, 2010, pp. 668–673.

[8] J. Haden, A. Tiwari, R. Roy, and D. Ruta, "Churn prediction using complaints data", in *Proc. World Academy of Science, Engineering and Technology*, vol. 19, 2006.

[9] A. L. Barabasi and R. Albert, "Emergence of scaling in random networks", *Science*, vol. 286, no. 5439, pp. 509–512, 1999.

[10] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks", *Physical Review E*, 69, 026113, 2002.

[11] Y. Richter, E. Yom-Tov, and N. Slonim, "Predicting customer churn in mobile networks through analysis of social groups", in *Proc. SIAM Int. Conf. on Data Mining SDM 2010*, Columbus, OH, USA, 2010, pp. 732–741.

[12] K. Dasgupta, R. Singh, and B. Viswanathan, "Social ties and their relevance to churn in mobile database technoltelecom networks", in *Proc. 11th Int. Conf. on Extending Database Technology: Advances in Database Technology EDBT'08*, Nantes, France, 2008, pp. 668–677.

[13] M. Karnstedt, M. Rowe, J. Chan, H. Alani, and C. Hayes, "The effect of user features on churn in social networks", in *Proc. 3rd Int. Web Science Conf. WebSci'11*, Koblenz, Germany, 2011, pp. 1–8.

[14] M. Zawisza, P. Wojewnik, B. Kamiński, and M. Antosiewicz, "Social-network influence on telecommunication customer attrition", in *Agent and Multi-Agent Systems: Technologies and Applications LNCS*, vol. 6682, pp. 64–73. Springer, 2011.

[15] M. N. Abd-Allah, A. Salah, and S. R. El-Beltagy, "Enhanced customer churn prediction using social network analysis", in *Proc. 3rd Worksh. Data-Driven User Behav. Model. & Mining from Social Media DUBMOD'14*, Shanghai, China, 2014, pp. 11–12.

[16] W. Verbeke, D. Martens, and B. Baesens, "Social network analysis for customer churn prediction", *Appl. Soft Comput.*, vol. 14, pp. 431–446, 2014.

[17] M. Kamola, E. Niewiadomska-Szynkiewicz, and B. Piech, "Reconstruction of a social network graph from incomplete call detail records", in *Proc. Int. Conf. Computat. Aspects of Soc. Netw. CASoN 2011*, Salamanca, Spain, 2011, pp. 136–140.

[18] W. Aiello, F. Chung, and L. Lu, "A random graph model for massive graphs", *Proc. 32nd Annual ACM Symp. Theory of Comput. STOC '00*, Portland, OR, USA, 2000, pp. 171–180.

[19] LaNet-vi website [Online]. Available: http://lanet-vi.fi.uba.ar/

[20] Igraph: Network Analysis and Visualisation [Online]. Available: http://cran.r-project.org/web/packages/igraph/

[21] D. M. W. Powers, "Evaluation: From precision, recall and F-measure to ROC, informedness, markedness & correlation", *J. Machine Learn. Technol.*, vol. 2, no. 1, 37–63, 2011.

[22] R. Kasprzyk and Z. Tarapata, "Graph-based optimization method for information diffusion and attack durability in networks", in *Rough Sets and Current Trends in Computing*, M. Szczuka, M. Kryszkiewicz, S. Ramanna, R. Jensen, and Q. Hu, Eds., *LNCS*, vol. 6086, pp. 698–709. Springer, 2010.

Witold Gruszczyński received his M.Sc. in Computer Science from the Warsaw University of Technology, Poland, in 2015. Currently he is with Codeflip company. His research area focuses on applications of data mining and social networks.

E-mail: wgruszczynski@codeflip.pl
Codeflip
Belgradzka st 4/40
02-793 Warsaw

Piotr Arabas – for biography, see this issue, p. 12.

Paper

Similarity Index based Link Prediction Algorithms in Social Networks: A Survey

Pulipati Srilatha and Ramakrishnan Manjula

School of Computer Science & Engineering (SCOPE), VIT University, Vellore, Tamil Nadu, India

Abstract—Social networking sites have gained much popularity in the recent years. With millions of people connected virtually generate loads of data to be analyzed to infer meaningful associations among links. Link prediction algorithm is one such problem, wherein existing nodes, links and their attributes are analyzed to predict the possibility of potential links, which are likely to happen over a period of time. In this survey, the local structure based link prediction algorithms existing in literature with their features and also the possibility of future research directions is reported and discussed. This survey serves as a starting point for beginners interested in understanding link prediction or similarity index algorithms in general and local structure based link prediction algorithms in particular.

Keywords—link prediction, similarity based link prediction, similarity index, social network analysis.

1. Introduction

Social network analysis [1] has gained lot of importance with millions of users being part of one or other online social networking sites. Link prediction problem in social networks has gained considerable interest from researchers of various domains [2]–[6].

Everyone in this world is connected to each other through an average small length was claimed by Stanley Milgram in 1967 itself in his ”small world” experiments [7]. But predicting the link or possibility of potential link among each other based on social networks is also an interesting task as social networks is also a kind of “small world” [8] where in nodes are connected to each other either through a direct or indirect link through intermediate nodes.

Link prediction as a tool helps in understanding potential links, which may happen among nodes in a network over a period of time. Based on individual preference, either through attributes or link structure, recommending friends (or likeminded people), passing information to individuals having similar interests, recommending products [9]–[13] seems to be an interesting and challenging task as the online social network size is growing with time and some of the conventional algorithms find it difficult to analyze such huge data volume.

In general link prediction refers to possible links among a pair of nodes in a social network [14] but it can also be thought as a possible link prediction among a node and commodity, i.e. how likely that a person is interested in having link with the commodity or in simple purchase the commodity [15]. Recommender systems [16], [17] which recommend products or commodities to individuals can also be thought as a special case of link prediction problem. Apart from identifying potential links and recommending commodities, link prediction problem also help to solve similar problems like spurious link detection [18], mitigating e-mail spams in social networks [19] defending against Sybil attacks [20].

In literature many methods have been proposed for link prediction problem. Broadly they can be classified into similarity based methods, maximum likelihood based methods and probabilistic model based methods [21], [22]. Similarity based methods predict links among nodes based on similarity score or similarity index calculated among a pair of vertices or nodes. Pair of nodes having high similarity value will tend to form a link in future. Maximum likelihood methods predict links based on hierarchical structure model or stochastic models [23], [24]. Probabilistic methods of link prediction are defined over Markov chains or Bayesian learning models in turn using principles of artificial intelligence in broad sense [25]–[30].

In this survey, we are concentrating only on the similarity based methods used in link prediction. The methods exploit the nature or structure of graphs how the nodes are connected and attributes associated with the nodes for link prediction. Similarity based methods can be further classified into three types based on the nature of similarity score or index calculation, as follows:

- local structure based,
- global structure based,
- quasi-local structure based.

Investigations have been carried out to understand the link prediction based on the nature of link over a period of time or evolutionary history of links and clustering based link prediction techniques. All the techniques or methods discussed in this survey are proposed over unweighted edges for the results on link prediction over weighted edges [31]–[33].

This paper is organized as follows. Section 2 introduces the preliminaries of graph theory. Section 3 deals with different types of similarity based prediction techniques and

how they are defined. Section 4 deals with the discussion on different similarity based prediction techniques and their summary. Section 5 contains conclusions.

2. Preliminaries

2.1. Graph

A graph G is defined as G = (V, E), where V is a set of vertices or nodes and E is a set of edges. The notion of vertices and nodes, links and edges are used interchangeably in the literature.

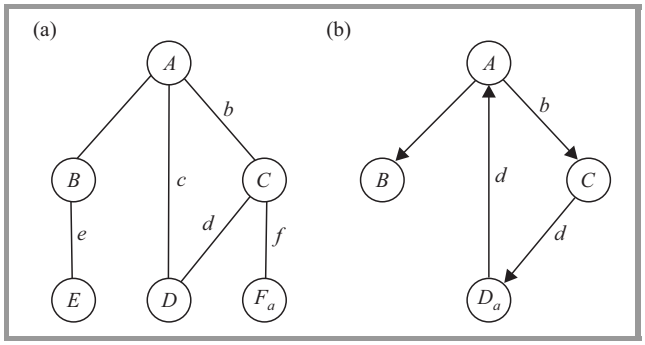


Fig. 1. Graphs: (a) undirected, (b) directed.

Graph shown in Fig. 1a is a simple undirected graph with set {A, B, C, D, E} as vertices and set {a, b, c, d, e, f} as edges. An edge from a node A to node B is called a path and it is of length 1 (or simply path length is 1) where as the path from node A to node D via node C is of length 2. Closed path from node A to node A (via node C and D) is called a circuit or cycle. The girth of a graph is number of edges in shortest cycle. Graph shown in Fig. 1b represents a graph with directed edge, in such graph, traversal is only possible in the direction of edge. Traversal from node A to node D via node C is possible where as traversal is not possible from node B to node C. In an undirected graph, degree d of a node is the number of edges or links connected to that node where as in case of directed graph, outdegree and indegree respectively indicate number of links or edges directed out and directed into a node. In case of Fig. 1a, degree of node A is 3 and of node e is 1. Similarly, for Fig. 1b, indegree for node A is 1 and outdegree is 2.

2.2. Adjacency Matrix

Adjacency matrix is used to understand the existence or non-existence of links or edges between nodes in a graph. Apart from that, other advantages includes that it can be given computed easily, can be given as an input to computer and also to understand the nature of path length by taking powers of adjacency matrix.
Adjacency matrix A_{ij} is defined as:

$$A_{ij} = \begin{cases} 1 & \text{if there is an edge from node } i \text{ to node } j, \\ 0 & \text{otherwise.} \end{cases}$$

where i and j are nodes in the graph. For the example considered in graph shown in Fig. 1a adjacency matrix is given by

$$A = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix},$$

where the matrix entry 1 indicates the existence of link or edge between nodes in the graph and a entry 0 indicates no link or edge.
Path length more than 1 can be computed from adjacency matrix by taking its powers. Adjacency matrices A^2, A^3 gives the nodes connected by path length 2 and 3 respectively. But the entries in table will remain either 0 or 1 for any powers of A except the elements of principal diagonal. However, principal diagonal elements have no role to play with regard to link prediction as these elements indicate links or edges to same nodes.
Similarly, matrix A^2 indicates nodes connected with a path length 2. In matrix A^2, nodes A and E, A and F, B and C etc., are connected with path length 2:

$$A^2 = \begin{bmatrix} 3 & 0 & 1 & 1 & 1 & 1 \\ 0 & 2 & 1 & 1 & 0 & 0 \\ 1 & 1 & 3 & 1 & 0 & 0 \\ 1 & 1 & 1 & 2 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}.$$

2.3. Link Prediction Problem

Link prediction problem is identifying a potential or possible link among vertices or nodes in a network. The link prediction problem can be defined as, given node E, what are the chances that node D and node F would like to connect with node E (Fig. 1). Predicting such links is not an easy task. Many similarity index based link prediction methods have been defined based on nature of computing them, as local structure, global structure and quasi local structure [21], [22].
Path length equal to 1 indicates the existence of direct link between nodes in a network. Local structure based method computes the similarity score using a node and its neighborhood nodes (of path length 2). The aim is to determine the chances of possible links between a pair of vertices where as global structure based method computes similarity score considering over all structure of network (mainly paths of length > 2) by ranking each nodes by similarity score and nodes with highest value of similarity score tends to have more links than the ones having lower value. Quasi-local structure based similarity measure is even though computed similar to the local structure based link prediction method, it uses path of length more than 2. Hence, covering the entire network like global structure based methods.

3. Methods in Similarity Index Based Link Prediction

Link prediction between nodes in a network is given by a score $S(a,b)$ where, a and b are nodes. Higher the value of S between the pair of vertices, higher is the chance of forming a link between them in future.

3.1. Local Structure Based Similarity Index Methods

3.1.1. Common Neighbor

Common Neighbor [34] method is one of the simplest techniques used for link prediction. Two nodes are likely to form a link if they have many common neighbors [35]. This method of link predicition is also called friend-of-a-friend link prediction [36]. Index $S_{CN}(a,b)$ for Common Neighbor method is computed as:

$$S_{CN}(a,b) = |\Gamma(a) \cap \Gamma(b)|, \tag{1}$$

where a and b are two nodes and $\Gamma(a)$ and $\Gamma(b)$ denote the set of neighbors of nodes a and b respectively. Common Neighbor method is being used extensively in various context to predict future links or collaborations [14], [37], [38].

3.1.2. Salton Index

Salton index [39] also called as Salton Cosine index is used to find the similarity index based on cosine angle between rows of adjacency matrix having nodes a and b [40]. Salton index $S_{Salton}(a,b)$ is computed as follows:

$$S_{Salton}(a,b) = \frac{|\Gamma(a) \cap \Gamma(b)|}{\sqrt{k_a \times k_b}}, \tag{2}$$

where k_a is the degree of node a and k_b is degree of node b.

3.1.3. Jaccard Index

Jaccard [41] in 1901 proposed a statistic to compare similarity and diversity of sample sets. It is the ratio of common neighbors of nodes a and b to the all neighbors nodes of a and b. As a result value of Jaccard index prevents higher degree nodes to have high similarity index with other nodes [42]. Jaccard index $S_{Jaccard}(a,b)$ is computed as:

$$S_{Jaccard}(a,b) = \frac{|\Gamma(a) \cap \Gamma(b)|}{|\Gamma(a) \cup \Gamma(b)|}. \tag{3}$$

3.1.4. Sørenson Index

A measure similar to Jaccard [41] proposed by Sørenson [43] in 1948 to measure similarities among species. Sørenson index is calculated as ratio of twice the common neighbors of nodes a and b to the sum of degrees of nodes a and b or total degree of nodes a and b. Sørenson index $S_{Sørenson}(a,b)$ is computed as:

$$S_{Sørenson}(a,b) = \frac{2|\Gamma(a) \cap \Gamma(b)|}{k_a + k_b}. \tag{4}$$

3.1.5. Hub Promoted Index

Hub Promoted index [44] is a measure defined as the ratio of common neighbors of nodes a and b to the minimum of degrees of nodes a and b. Hub Promoted index $S_{HPI}(a,b)$ is computed as:

$$S_{HPI}(a,b) = \frac{|\Gamma(a) \cap \Gamma(b)|}{\min\{k_a, k_b\}}. \tag{5}$$

3.1.6. Hub Depressed Index

Hub Depressed index [21], [22] defined as the ratio of common neighbors of nodes a and b to the maximum of degrees of nodes either a or b. Hub Depressed index gives lower score compared to Hub Promoted index as the similarity measured is computed by taking maximum of degrees of nodes a and b [45]. Hub Depression index $S_{HDI}(a,b)$ is computed as:

$$S_{HDI}(a,b) = \frac{|\Gamma(a) \cap \Gamma(b)|}{\max\{k_a, k_b\}}. \tag{6}$$

3.1.7. Leicht Holme Newman Index-1

Leicht *et al*. in [46] proposed a measure to define local structure based similarity measure. It is the ratio of common neighbors of nodes a and b to the product of degrees of nodes a and b. Leicht Holme Newman index-1 is computed as:

$$S_{LNH1}(a,b) = \frac{|\Gamma(a) \cap \Gamma(b)|}{k_a \times k_b}. \tag{7}$$

Equations (7) and (2) differ only in their denominator. As a result, for a same pair of nodes Salton index always assigns a higher score compared to Leicht Holme Newman index-1 [45].

3.1.8. Preferential Attachment Index

Preferential Attachment index [47] is a similarity score calculated independent of the neighborhood of each node. Social networks expand as and when new nodes joins in and the new nodes joins in with the existing nodes having higher degree compared to lower degree nodes [48]. Barbasi *et al*. showed this result for the evolution co-authorship links [49]. Preferential Attachment index $S_{PA}(a,b)$ between nodes a and b is computed as:

$$S_{PA}(a,b) = k(a) \times k(b). \tag{8}$$

3.1.9. Adamic-Adar Index

Adamic-Adar index [50] proposed by Adamic and Adar is calculated by adding weights to the nodes which are connected to both nodes a and b. Adamic and showed how similar nodes are linked analyzing text, out-links, in-links and mailing lists from homepages of students of Massachusetts

Institute of Technology (MIT) and Stanford University [50]. The index is computed as:

$$S_{AAI}(a,b) = \sum_{z \in \Gamma(a) \cap \Gamma(b)} \frac{1}{\log k_z}. \quad (9)$$

where z is a common neighbor to nodes both a and b and k is the degree of node z.

3.1.10. Resource Allocation Index

Resource Allocation index [22] is calculated based on the intermittent nodes connecting node a and b. The similarity index S_{RAI} is defined as the amount of resource node a receives from node b through indirect links and each intermediate link contributes an unit of resource. Also, $S_{RAI}(a,b)$ is symmetric to $S_{RAI}(b,a)$:

$$S_{RAI}(a,b) = \sum_{z \in \Gamma(a) \cap \Gamma(b)} \frac{1}{k_z}. \quad (10)$$

Equations (9) and (10) differs only in their denominator as the former takes the logarithm of the denominator. As a result, for a same pair of nodes Adamic-Adar index always assigns a higher score compared to Resource Allocation Index.

3.2. Global Structure Based Similarity Index Methods

3.2.1. Katz Index

Katz index [51] is calculated considering all paths instead of individual paths among node pairs. Like all global structure based similarity indices Katz index considers paths of length ≥ 2.

Katz score between two nodes a and b is calculated as a sum of products of path length (of all length) and β a free parameter used to control path weights. It is computed as:

$$S_{Katz}(a,b) = \sum_{l=1}^{\infty} \beta^l \cdot |paths_{ab}^{<l>}| =$$

$$= \beta A_{ab} + \beta^2 (A^2)_{ab} + \beta^3 (A^3)_{ab} + \ldots \quad (11)$$

where A is the adjacency matrix of the graph, $|paths_{ab}^{<l>}|$ is the set of all paths with length l connecting nodes a and b. When β is very small Katz index will be similar to Common Neighbor method of link prediction defined in Subsection 3.1.1. Scalability of Katz index is discussed in [52] implementing it on Hadoop parallel Map/Reduce tool [53], [54].

3.2.2. Average Commute Time

Average Commute Time (ACT) [14] is defined as the average number of steps required for a random walker to reach node b starting from node a. For computation simplicity pseudo-inverse of Laplace matrix is used for representing the commute time between node a and node b. It is computed as:

$$S_{ACT}(a,b) = \frac{1}{l_{aa}^+ + l_{bb}^+ - 2l_{ab}^+}, \quad (12)$$

where a and b are nodes, l_{aa}^+ is corresponding entry in Laplacian Matrix, L^+.

3.2.3. Random Walk with Restart

Random Walk with Restart [55] Index is calculated for similarity between nodes. Other variations to this approach include SimRank [56], Neighborhood Formation and anomaly detection [57], PageRank algorithm [58], Local Random Walk [59]. Random Walk with Restart is computed as:

$$S_{RWR}(a,b) = q_{ab} + q_{ba}, \quad (13)$$

where a and b are nodes, q_{ab} is the probability with which a random walker starting at node a comes back to node a itself visiting a random neighbor, mathematically stated if a random walker visits random neighbor of node a with probability c and returns to node a with probability $1-c$. q_{ab} and q_{ba} are not symmetric. It is similar to Rooted PageRank index [14].

3.2.4. SimRank

SimRank [56] is calculated for node similarity between nodes, intuition behind SimRank similarity index is two nodes are similar if they are referred by similar nodes [60]. SimRank $S_{SimRank}(a,b)$ is recursively computed as:

$$S_{SimRank}(a,b) = C \frac{\sum_{z \in \Gamma(a)} \sum_{z' \in \Gamma(b)} S_{SimRank}(z,z')}{k_a \cdot k_b}, \quad (14)$$

where z is the set of neighbors of node a ($\Gamma(a)$) and z' is the set of neighbors of node b ($\Gamma(b)$). $C \in [0 \ldots 1]$ is the decay factor. $S(a,a) = 1$ is assumed and indicates each node is similar to itself [60].

3.2.5. Escape Probability

Escape Probability [61]–[63] is a global structure based link prediction algorithm computed as:

$$S_{EP}(a,b) = \frac{Q[a,b]}{Q[a,a]Q[b,b] - Q[a,b]Q[b,a]}, \quad (15)$$

where a and b are nodes, $Q = \frac{RPR}{1-\beta_{RPR}}$, and RPR is the rooted page rank algorithm, which is similar to Random Walk with Restart similarity measure [63].

3.2.6. Leicht Holme Newman Index-2

Leicht *et al*. [46] proposed a measure to define global structure based similarity measure and it is a path dependent

method covering entire network. Leicht Holme Newman index-2 S_{LNH2} is defined as:

$$S_{LNH2}(a,b) = \delta_{ab} + \frac{2M}{k_a k_b} \sum_{l=0}^{\infty} \phi l \lambda^{1-l} (A^l)_{ab}, \qquad (16)$$

where δ_{ab} is Kronecker's function, k_a is the degree of node a, k_b is degree of node b, ϕ is a free parameter, A is an adjacency matrix, M is the total number of edges in the network, l is the path length and λ is an Eigen vector. Two measures are proposed to predict links between nodes in a network, one is local structure based link prediction method (shown in Subsection 3.1.7) and other is global structure based link prediction method (Subsection 3.2.6).

3.3. Quasi-Local Structure Based Similarity Index Methods

3.3.1. Local Path Index

Local Path index [22], [64] is a similarity score calculated between nodes similar to local structure based methods but by considering paths of length > 2. Local Path index S_{LPI} is computed as

$$S_{LPI}(a,b) = A^2 + \varepsilon A^3 + \varepsilon^2 A^4 + \ldots + \varepsilon^{n-2} A^n, \qquad (17)$$

where A is the adjacency matrix, ε is a free parameter and $n > 2$. Equation (17) will be equal to common neighbors similarity index if ε is zero. If $\varepsilon = 0$, Eq. (17) reduces to $S_{LPI}(a,b) = A^2$, which is similar to Common Neighbors similarity based index defined in Subsection 3.1.1. Local Path index is computed for path with length > 2 with undirected edges. Recently Wang *et al*. [65] proposed a variation to local path index method to predict links in a directed network.

3.3.2. Local Random Walk

While computing Local Random Walk index [59] S_{LRW}, random walker is put on node a and initial density is given by $\vec{\pi}_a(0) = \vec{e}_a$, which will evolve as $\vec{\pi}_a(t+1)$ for $t \geq 0$. Local Random Walk index S_{LRW} is computed as:

$$S_{LRW}(a,b)(t) = q_a \pi_{ab}(t) + q_b \pi_{ab}(t), \qquad (18)$$

where a and b are nodes, q is the initial configuration t represents the initial density at time $t = 0$ and later for $t \geq 0$.

3.3.3. Superposed Random Walk

Superposed Random Walk proposed by Liu and Lü in [59] is the sum of all Local Random Walk in the network. Superposed Random Walk $S_{SRW}(a,b)(t)$ is computed as:

$$S_{SRW}(a,b)(t) = \sum_{\tau=1}^{t} S_{LRW}(a,b)(\tau) = \sum_{\tau=1}^{t} \left[q_a \pi_{ab}(t) + q_b \pi_{ab}(t) \right], \qquad (19)$$

where a and b are nodes, S_{LRW} is defined in Eq. (18) and t denotes the time steps.

3.3.4. Extended Jaccard Index

The Jaccard index is defined as a similarity index to identify possible link between nodes in common neighborhood. Jaccard index is a local structure based algorithm and considers paths of length 2. With a notion that the nodes tend to connect in path length are more likely to connect in path length ≥ 2, Chartsias [52] proposed extended Jaccard algorithm and implemented on Hadoop parallel Map/Reduce tool for scalability.
Extended Jaccard index is computed as:

$$S_{Jaccard}(a,b) = \frac{|\Gamma_d(a) \cap \Gamma_d(b)|}{|\Gamma_d(a) \cup \Gamma_d(b)|}, \qquad (20)$$

where a and b are two nodes and $\Gamma_d(a)$ and $\Gamma_d(b)$ denote the set of extended neighbors of nodes a and b respectively at hops $1 \ldots d$ for each of the node.

3.3.5. FriendLink

Papadimitriou *et al*. [66] defined FriendLink algorithm for measuring similarity between nodes. FriendLink algorithm is a quasi-local structure based algorithm and uses paths of length ≥ 2. Similarity measure $S_{FriendLink}$ is computed as:

$$S_{FriendLink}(a,b) = \sum_{i=2}^{l} \frac{1}{i-1} \cdot \frac{|paths^i_{a,b}|}{\prod_{j=2}^{i}(n-j)}, \qquad (21)$$

where n is the number of vertices in graph, l is the path length considered $l \geq 2$, $\frac{1}{i-1}$ is the attenuation factor that weights path according to length l. $\prod_{j=2}^{i}(n-j)$ is the number of possible length l-paths from a to b.

4. Summary and Discussion

Local structure based methods of link prediction computes the similarity score based on common neighbors which gives an accurate measure to know link structure arising between nodes. Such measure is computed only between nodes of path length 2 and not beyond that. As a result some interesting and potential links may be missed and also as a matter of fact it will be difficult and time consuming for computing similarity score for all nodes in network. Developing parallel algorithms for such task seems to be interesting but not yet explored.
Global structure based methods of link prediction computes similarity score based on global link structure of graph and computed for nodes having path length > 2. As a result many interesting and potential links can be indentified which are being missed in local structure based link prediction methods. But calculating similarity measure based on global structure is time consuming and difficult in case of large networks such as online social networks where petabytes of data has to be analyzed for predicting links. Developing scalable global structure based algorithms seems to be interesting field but not yet investigated in full pledge except a handful of attempts [52], [66]–[68].

Table 1
Summary of similarity based link prediction algorithms

Link prediction technique	Local	Global	Quasi-local	Scalability
Common Neighbor	✓			Not investigated
Salton index	✓			
Jaccard index	✓			
Sørenson index	✓			
Hub Promoted index	✓			
Hub Depressed index	✓			
Leicht Holme Newman index-1	✓			
Preferential Attachment index	✓			
Adamic-Adar index	✓			
Resource Allocation index	✓			
Katz index		✓		Investigated
Average Commute Time		✓		Not investigated
Random Walk with Restart		✓		
SimRank		✓		
Escape Probability		✓		
Leicht Holme Newman index-2		✓		
Local Path index			✓	
Local Random Walk			✓	
Superposed Random Walk			✓	
Extended Jaccard index			✓	Investigated
FriendLink			✓	

Quasi-local structure based methods of link prediction seems to be more accurate than local link structure based link prediction algorithms. Starting with neighborhood of nodes similar to local structure based methods, quasi-local structure based methods cover entire network by considering all paths between nodes of length more than 2. Some of the methods such as FreeLink, Extended Jaccard are been verified for scalability issues. Further developing algorithms able to compute similarity index for entire network with more accuracy and scalability seems to be interesting problem to work on.

In all the methods discussed in Section 3, information about the local communities are not considered while calculating similarity index. By considering local information about community while calculating similarity index, precision of link prediction increases as shown by Soundarajan and Hopcraft [69] for Common Neighbor and Resource Allocation. But the method proposed by them is not being further explored over other link prediction methods and other bench mark datasets. It will be interesting to investigate similarity based link prediction with local information about community as it can be further extended to Louvain method [70] proposed by Blondel *et al*.

5. Conclusions

In this survey, authors have reported similarity based link prediction algorithms existing in literature and possibility of future research issues. Summary of these methods is given in Table 1. This paper serves as a starting point for researchers and novice readers interested in understanding basics of link prediction in online social networks in general and local structure based link prediction methods in particular.

References

[1] J. Scott, *Social Network Analysis*. Sage, 2012.

[2] Z. Huang and D. K. Lin, "The time-series link prediction problem with applications in communication surveillance", *INFORMS J. Comput.*, vol. 21, no. 2, pp. 286–303, 2009.

[3] K. Jahanbakhsh, V. King, and G. C. Shoja, "Predicting missing contacts in mobile social networks", *Pervas. & Mob. Comput.*, vol. 8, no. 5, pp. 698–716, 2012.

[4] R. Guimerà and M. Sales-Pardo, "Missing and spurious interactions and the reconstruction of complex networks", *Proceedings of the National Academy of Sciences (PNAS)*, vol. 106, no. 52, pp. 22073–22078, 2009.

[5] E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing, "Mixed membership stochastic block models for relational data with application to protein-protein interactions", in *Proc. Int. Biometrics Society Annual Meeting ENAR 2006*, Tampa, FL, USA, 2006.

[6] U. M. Singh-Blom, N. Natarajan, A. Tewari, J. O. Woods, I. S. Dhillon, and E. M. Marcotte, "Prediction and validation of gene-disease associations using methods inspired by social network analyses", *J. PLOS One*, vol. 8, no. 9, 2013.

[7] S. Milgram, "The small world problem", *Psychology Today*, vol. 2, no. 1, pp. 60–67, 1967.

[8] S. Goel, R. Muhamad, and D. Watts, "Social search in small-world experiments", in *Proc. 18th Int. Conf. World Wide Web WWW'09*, Madrid, Spain, 2009, pp. 701–710.

[9] Y. Sun, R. Barber, M. Gupta, C. C. Aggarwal, and J. Han, "Co-author relationship prediction in heterogeneous bibliographic networks", in *Proc. Int. Conf. Adv. Social Netw. Anal. Mining ASONAM 2011*, Kaohsiung, Taiwan, 2011, pp. 121–128.

[10] M. E. Newman, "The structure and function of complex networks", *SIAM Rev.*, vol. 45, no. 2, pp. 167–256, 2003.

[11] Z. Yin, M. Gupta, T. Weninger, and J. Han, "Linkrec: a unified framework for link recommendation with user attributes and graph structure", in *Proc. 19th Int. Conf. World Wide Web WWW'10*, Raleigh, NC, USA, 2010, pp. 1211–1212.

[12] U. Shardanand and P. Maes, "Social information filtering: algorithms for automating «Word of Mouth»", in *Proc. SIGCHI Conf. Human Fact Comput. Syst. CHI'95*, Denver, CO, USA, 1995, pp. 210–217.

[13] S. Hill, F. Provost, and C. Volinsky, "Network-based marketing: identifying likely adopters via consumer networks", *Statistical Sci.*, vol. 21, no. 2, pp. 256–276, 2006.

[14] D. Liben-Nowell and J. Kleinberg, "The link-prediction problem for social networks", *J. American Soc. for Inform. Sci. & Technol.*, vol. 58, no. 7, pp. 1019–1031, 2007.

[15] X. Li and H. Chen, "Recommendation as link prediction in bipartite graphs: a graph kernel-based machine learning approach", *Decision Support Syst.*, vol. 54, no. 2, pp. 880–890, 2013.

[16] F. Ricci, L. Rokach, and B. Shapira, *Introduction to Recommender Systems Handbook*. Springer, 2011.

[17] L. Lü, M. Medo, C. H. Yeung, Y.-C. Zhang, Z.-K. Zhang, and T. Zhou, "Recommender systems", *Phys. Reports*, vol. 519, no. 1, pp. 1–49, 2012.

[18] A. Zeng and G. Cimini, "Removing spurious interactions in complex networks", *Phys. Rev. E*, vol. 85, no. 3, article ID 036101, 2012.

[19] S. Garriss, M. Kaminsky, M. J. Freedman, B. Karp, D. Mazières, and H. Yu, "Re: Reliable email", in *Proc. 3rd Symp. Netw. Syst. Design & Implement. NSDI'06*, San Jose, CA, USA, 2006, vol. 6, pp. 22–22.

[20] H. Yu, M. Kaminsky, P. B. Gibbons, and A. Flaxman, "Sybilguard: defending against sybil attacks via social networks", *ACM SIGCOMM Comp. Commun. Rev.*, vol. 36, no. 4, pp. 267–278, 2006.

[21] L. Lü and T. Zhou, "Link prediction in complex networks: A survey", *Physica A: Statist. Mechan. & its Appl.*, vol. 390, no. 6, pp. 1150–1170, 2011.

[22] T. Zhou, L. Lü, and Y.-C. Zhang, "Predicting missing links via local information", *The Eur. Phys. J. B*, vol. 71, no. 4, pp. 623–630, 2009.

[23] M. Girvan and M. E. Newman, "Community structure in social and biological networks", *Proceedings of the National Academy of Sciences (PNAS)*, vol. 99, no. 12, pp. 7821–7826, 2002.

[24] R. Guimera, M. Sales-Pardo, and L. A. Amaral, "Classes of complex networks defined by role-to-role connectivity profiles", *Nature Phys.*, vol. 3, no. 1, pp. 63–69, 2007.

[25] L. Getoor, N. Friedman, D. Koller, and A. Pfeffer, "Learning probabilistic relational models", in *Relational Data Mining*, S. Džeroski and N. Lavrač, Eds. Springer, 2001, pp. 307–335.

[26] R. R. Sarukkai, "Link prediction and path analysis using Markov chains", *Computer Networks*, vol. 33, no. 1, pp. 377–386, 2000.

[27] P. Sarkar and A. Moore, "A tractable approach to finding closest truncated-commute-time neighbors in large graphs", in *Proc. 23rd Conf. Uncertainty in Artif. Intellig. UAI 2007*, Vancouver, BC, Canada, 2007 (arXiv preprint arXiv:1206.5259, 2012).

[28] D. Heckerman, C. Meek, and D. Koller, "Probabilistic entity-relationship models, PRMs, and plate models", in *Introduction to Statistical Relational Learning*, L. Getoor and B. Taskar, Eds. MIT Press, 2007, pp. 201–239.

[29] H. Kashima and N. Abe, "A parameterized probabilistic model of network evolution for supervised link prediction", in *Proc. 6th Int. Conf. Data Mining ICDM 2006*, Honk Kong, China, 2006, pp. 340–349.

[30] C. Wang, V. Satuluri, and S. Parthasarathy, "Local probabilistic models for link prediction", in *Proc. 7th Int. Conf. Data Mining ICDM 2007*, Omaha, NE, USA, 2007, pp. 322–331.

[31] L. Lü and T. Zhou, "Link prediction in weighted networks: The role of weak ties", *Europhys. Lett. (EPL)*, vol. 89, no. 1, pp. 18001-p1–18001-p6, 2010.

[32] J. Zhao, L. Miao, J. Yang, H. Fang, Q.-M. Zhang, M. Nie, P. Holme, and T. Zhou, "Prediction of links and weights in networks by reliable routes", *Scientific Reports*, vol. 5, article no. 12261, 2015.

[33] N. Sett, S. R. Singh, and S. Nandi, "Influence of edge weight on node proximity based link prediction methods: An empirical analysis", *Neurocomputing*, vol. 172, pp. 71–83, 2016.

[34] F. Lorrain and H. C. White, "Structural equivalence of individuals in social networks", *The J. Mathem. Sociol.*, vol. 1, no. 1, pp. 49–80, 1971.

[35] H. Liao, A. Zeng, and Y.-C. Zhang, "Predicting missing links via correlation between nodes", *Physica A: Statist. Mechan. & its Appl.*, vol. 436, pp. 216–223, 2015.

[36] Z. Zhang, Y. Liu, W. Ding, W. W. Huang, Q. Su, and P. Chen, "Proposing a new friend recommendation method, frutai, to enhance social media providers' performance", *Decision Support Syst.*, vol. 79, pp. 46–54, 2015.

[37] M. E. Newman, "Clustering and preferential attachment in growing networks", *Phys. Rev. E*, vol. 64, no. 2, article ID: 025102, 2001.

[38] G. Kossinets, "Effects of missing data in social networks", *Social Networks*, vol. 28, no. 3, pp. 247–268, 2006.

[39] G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval*. New York: McGraw-Hill, 1986.

[40] A. Rodriguez, B. Kim, M. Turkoz, J.-M. Lee, B.-Y. Coh, and M. K. Jeong, "New multi-stage similarity measure for calculation of pairwise patent similarity in a patent citation network", *Scientometrics*, vol. 103, no. 2, pp. 565–581, 2015.

[41] P. Jaccard, "Étude comparative de la distribution florale dans une portion des Alpes et des Jura", *Bulletin del la Société Vaudoise des Sciences Naturelles*, vol. 37, pp. 547–579, 1901 (in French).

[42] H. Liao and A. Zeng, "Reconstructing propagation networks with temporal similarity", *Scientific Reports*, vol. 5, article no. 11404, 2015.

[43] T. J. Sørensen, *A Method of Establishing Groups of Equal Amplitude in Plant Sociology Based on Similarity of Species Content and its Application to Analyses of the Vegetation on Danish Commons*. Biologiske skrifter, Kongelige Danske videnskabernes selskab. København: I kommission hos E. Munksgaard, 1948.

[44] E. Ravasz, A. L. Somera, D. A. Mongru, Z. N. Oltvai, and A.-L. Barabási, "Hierarchical organization of modularity in metabolic networks", *Science*, vol. 297, no. 5586, pp. 1551–1555, 2002.

[45] Y.-L. He, J. N. Liu, Y.-X. Hu, and X.-Z. Wang, "OWA operator based link prediction ensemble for social network", *Expert Syst. with Appl.*, vol. 42, no. 1, pp. 21–50, 2015.

[46] E. Leicht, P. Holme, and M. E. Newman, "Vertex similarity in networks", *Phys. Rev. E*, vol. 73, no. 2, article ID: 026120, 2006.

[47] Z. Huang, X. Li, and H. Chen, "Link prediction approach to collaborative filtering", in *Proc. 5th ACM/IEEE-CS Joint Conf. Digit. Librar.*, Denver, CO, USA, 2005, pp. 141–142.

[48] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks", *Science*, vol. 286, no. 5439, pp. 509–512, 1999.

[49] A.-L. Barabási, H. Jeong, Z. Néda, E. Ravasz, A. Schubert, and T. Vicsek, "Evolution of the social network of scientific collaborations", *Physica A: Statist. Mechan. & its Appl.*, vol. 311, no. 3, pp. 590–614, 2002.

[50] L. A. Adamic and E. Adar, "Friends and neighbors on the Web", *Social Networks*, vol. 25, no. 3, pp. 211–230, 2003.

[51] L. Katz, "A new status index derived from sociometric analysis", *Psychometrika*, vol. 18, no. 1, pp. 39–43, 1953.

[52] A. Chartsias, "Link prediction in large scale social networks using hadoop", PhD thesis, Technical University of Crete, Greece, 2010.

[53] T. White, *Hadoop: The Definitive Guide*, 3rd ed. O'Reilly Media, 2012.

[54] K.-H. Lee, Y.-J. Lee, H. Choi, Y. D. Chung, and B. Moon, "Parallel data processing with MapReduce: a survey", *ACM SIGMOD Record*, vol. 40, no. 4, pp. 11–20, 2012.

[55] J.-Y. Pan, H.-J. Yang, C. Faloutsos, and P. Duygulu, "Automatic multimedia cross-modal correlation discovery", in *Proc. 10th ACM SIGKDD Int. Conf. Knowl. Discov. & Data Mining KDD'04*, Seattle, WA, USA, 2004, pp. 653–658.

[56] G. Jeh and J. Widom, "Simrank: a measure of structural-context similarity", in *Proc. 8th ACM SIGKDD Int. Conf. Knowl. Discov. & Data Mining KDD'02*, Edmonton, AB, Canada, 2002, pp. 538–543.

[57] J. Sun, H. Qu, D. Chakrabarti, and C. Faloutsos, "Neighborhood formation and anomaly detection in bipartite graphs", in *Proc. 5th IEEE Int. Conf. Data Mining ICDM'05*, Houston, TX, USA, 2005, pp. 418–425.

[58] S. Brin and L. Page, "Reprint of: The anatomy of a large-scale hypertextual web search engine", *Computer Networks*, vol. 56, no. 18, pp. 3825–3833, 2012.

[59] W. Liu and L. Lü, "Link prediction based on local random walk", *Europhys. Lett. (EPL)*, vol. 89, no. 5, article ID: 58007, 2010.

[60] W. Yu, X. Lin, W. Zhang, and J. McCann, "Fast all-pairs simrank assessment on large graphs and bipartite domains", *IEEE Trans. Knowl. & Data Engin.*, vol. 27, no. 7, pp. 1810–1823, 2015.

[61] P. G. Doyle and J. L. Snell, *Random Walks and Electric Networks*, The Carus Mathematical Monographs no. 22. Mathema Association of America, 1984.

[62] H. Tong, C. Faloutsos, and Y. Koren, "Fast direction-aware proximity for graph mining", in *Proc. 13th ACM SIGKDD Int. Conf. Knowl. Discov. & Data Mining KDD'07*, San Jose, CA, USA, 2007, pp. 747–756.

[63] H. H. Song, T. W. Cho, V. Dave, Y. Zhang, and L. Qiu, "Scalable proximity estimation and link prediction in online social networks", in *Proc. 9th ACM SIGCOMM Conf. Internet Measur. Conf. IMC 2009*, Chicago, IL, USA, 2009, pp. 322–335.

[64] L. Lü, C.-H. Jin, and T. Zhou, "Similarity index based on local paths for link prediction of complex networks", *Phys. Rev. E*, vol. 80, no. 4, article ID: 046122, 2009.

[65] X. Wang, X. Zhang, C. Zhao, Z. Xie, S. Zhang, and D. Yi, "Predicting link directions using local directed path", *Physica A: Statist. Mechan. & its Appl.*, vol. 419, pp. 260–267, 2015.

[66] A. Papadimitriou, P. Symeonidis, and Y. Manolopoulos, "Fast and accurate link prediction in social networking systems", *J. Syst. & Softw.*, vol. 85, no. 9, pp. 2119–2132, 2012.

[67] Y. Dong, C. Robinson, and J. Xu, "Hadoop based link prediction performance analysis".

[68] M. Fire, L. Tenenboim, O. Lesser, R. Puzis, L. Rokach, and Y. Elovici, "Link prediction in social networks using computationally efficient topological features", in *Proc. 3rd IEEE Int. Conf. on Priv., Secur., Risk and Trust and 3rd IEEE Int. Conf. on Social Comput. PASSAT/SocialCom 2011*, Boston, MA, USA, 2011, pp. 73–80.

[69] S. Soundarajan and J. Hopcroft, "Using community information to improve the precision of link prediction methods", in *Proc. 21st Int. Conf. Companion on World Wide Web WWW 2012*, Lyon, France, 2012, pp. 607–608.

[70] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks", *J. Statist. Mechanics: Theory and Experim.*, vol. 2008, no. 10, p. P10008, 2008.

Pulipati Srilatha received her B. Tech. and M. Tech. from Jawaharlal Nehru Technological University, Hdyerabad, India in 2002 and 2006, respectively. Presently she works as Assistant Professor at Anurag Group of Institutions (Formerly CVSR College of Engineering), Hyderabad and pursuing Ph.D. at the School of Computing Science and Engineering, VIT University, Vellore, India. Her interest include data mining, social network analysis and link prediction algorithms.
Email: sreelatha.pulipati@gmail.com
School of Computer Science & Engineering (SCOPE)
VIT University
Vellore, Tamil Nadu, India

Ramakrishnan Manjula received her B.E. in Computer Science & Engineering from University Visvesvaraya College of Engineering (UVCE), Bangalore, Karnataka State, India in 1992 and M.E. in Software Engineering from Anna University, Tamil Nadu, India in 2001 and Ph.D. in Computer Science and Engineering from VIT University, Vellore, India. Presently she is working as Associate Professor at the School of Computing Science and Engineering at VIT University, Vellore, India. Her area of specialization includes software process modeling, software testing and metrics, service oriented architecture, data mining and social network analysis.
Email: rmanjula@vit.ac.in
School of Computer Science & Engineering (SCOPE)
VIT University
Vellore, Tamil Nadu, India

Paper

Antenna Arrays Focused on Broadband Signals

Denis A. Vedenkin[1,2], Yuri E. Sedelnikov[1], and Aydar R. Nasybullin[1,2]

[1] *Institute of Radio Electronic and Telecommunications, Kazan National Research Technical University named after A. N. Tupolew, Kazan, Russian Federation*

[2] *Faculty of Radio Engineering, Volga State University of Technology, Kazan, Russian Federation*

Abstract—Broadband and ultra-wideband signals are increasingly used in modern radio systems. Traditional performance of evaluation antennas operating with narrowband signals are not always adequately reflect the characteristics of broadband antennas, at least in view of the frequency dependence of the antenna pattern. Accounting for broadband signals the antennas becomes important in the low-frequency range of the spectrum. Systems using these types of signals may include control of the atmosphere and measuring its frequency-selective properties in the range meter and decameter wavelengths. Possibility of spatial selection based on focusing of broadband signals in this case promises to implement a number of additional features. Therefore, it is important to evaluate the properties of antennas based on the spectral content of the signal, as well as taking into account the ways of its processing in the receiving equipment. Consideration features of functioning the antenna array, focused on broadband signal is devoted to this article.

Keywords—antenna array, broadband antenna, broadband signal, focusing on the broadband signals.

1. Introduction

The narrow-band antennas, often used to form the fields of radiation (or reception) in the far field, performing the required functions in a relatively small band $\Delta f/f_0 \ll 1$. These frequency boundaries of directional characteristics do not undergo significant change and are considered in analysis and synthesis of antennas for monochromatic signal with a frequency f_0. The finiteness of frequency band in these cases taking into account only when considering matching antenna feeder.

The broadband antenna is considered as antenna designed to operate in the relative frequency band $\Delta f/f_0$, amounts to no more than several tens of percent. In these cases, antennas are used in radio systems using relatively narrow-band signals. These antennas should provide radiation and reception at f_0 rearrangements carrier frequency within a desired frequency range. The behavior of antennas within the frequency band Δf is also insignificant and is not considered in estimates of the directional properties. The required broadbandness of such antennas is determined mainly as ensuring harmonization within a predetermined frequency range.

An ultra-wideband (UWB) antenna is composed of two fundamentally different groups. The first is represented by antennas with constant directivity and matching characteristics in a wide frequency band. The fundamental principles of the independence of the frequency characteristics of antennas installed in the 1950s, is well known [1]. In most practical applications, such UWB antennas are used in radio communication, sending or receiving radio signals with low or average values of the relative width of the spectrum, but with possibilities of changing its center frequency in a wide range. Special group of UWB antennas are for of radio facilities, mainly radars, with high value of the absolute width of the signal spectrum, for their part of the order of 0.5 GHz and more. These signals can be created with carrier frequencies equal centimeter or millimeter wave. The antennas providing transmitting and (or) receiving may be in the class of relatively narrow band. In those cases, where the carrier frequencies are shifted to a lower range, the antenna must be capable of operating with radio signals fractional bandwidth which

$$\frac{f_{\max} - f_{\min}}{0.5(f_{\max} + f_{\min})} \gg 1 .$$

Antennas used in such radio facilities, have a number of significant differences from the traditional. There are currently substantial number research studies of such antennas, for example [2]–[4].

Basic fundamental difference of UWB antennas properties as part of radio facilities is that every antenna is characterized by changes in the frequency characteristics of the transmission and reception. In radio devices with a relatively narrow bandwidth such changes in the frequency band corresponding to the signal spectrum may be disregarded. For this reason, such indicators as the antenna pattern, antenna directivity are assumed constant in the frequency band signal and introduced to a monochromatic signal. For UWB radio antennas, these changes may be so significant that they cannot be neglected and it is necessary initially to reckon with the presence of a significant frequency dependence of the directional characteristics $F(\theta, \varphi, f)$. Under these conditions, it is possible to determine the directional properties of the antenna apart from the spectral composition of the emitted radio signal and a method of treatment in the receiving equipment. Therefore, the definition of the antenna pattern composed of UWB radios is ambiguous. Moreover, it is considered that in these circumstances, there may be a mismatch patterns in transmit and receive modes. The authors will illustrate the situation in a simplified presentation.

2. UWB Signals Transmitting and Receiving by Antennas

2.1. Transmit Mode

When the antenna radiates UWB signal with a spectrum $G(f)$, the intensity of the electric field in the direction (θ, φ) can be written as:

$$E(\theta, \varphi, f) = F(\theta, \varphi, f)G(f). \tag{1}$$

To evaluate the integral effect is necessary to know exactly how to use the energy of the electromagnetic field in Eq. (1), which provides not enough information about the antenna already. Depending on the purpose of applying, the concept of the radiation pattern is filled with a different meaning. In some cases, in microwave technology, resulting effect may be evaluated by energy radiated signal in the direction (θ, φ):

$$\left|F_{UWB}^{TR.en}(\theta, \varphi, f)\right|^2 = \int_{f_0-\Delta f}^{f_0+\Delta f} \left|F(\theta, \varphi, f)G(f)\right|^2 \mathrm{d}f, \tag{2}$$

where f_0 and $2\Delta f$ are the average frequency and the bandwidth of the transmitted signal.

The power of the received UWB signal can be viewed as the result of the linear filter with frequency response receiving device $K(f)$. In this case the antenna pattern in the transmission mode is defined as:

$$\left|F_{UWB}^{TR.sign}\right|^2 = \left| \int_{f_0-\Delta f}^{f_0+\Delta f} F(\theta, \varphi, f)G(f)K(f)\mathrm{d}f \right|^2. \tag{3}$$

Difference in antenna pattern given by Eqs. (2) and (3) is evident. Also well visible is another fact – the directional properties of the antenna in the transmit mode to a certain extent depend on the properties of the receiver.

2.2. Receive Mode

The EMF is induced in the UWB antenna with a spectrum $G(f)$, and thus the radiation pattern in the receive mode, can be represented by the value $F(\theta, \varphi, f)G(f)$. The linear receiver performs the function of a weighted summation of the amplitudes of the oscillations in the band $[f_0 - \Delta f, f_0 + \Delta f]$ with frequency characteristics $K_{rec}(f)$. The radiation pattern in the receive mode may be determined by the resulting output of the received signal:

$$\left|F_{UWB}^{REC.sign}\right|^2 = \left| \int_{f_0-\Delta f}^{f_0+\Delta f} F(\theta, \varphi, f)G(f)K(f)K_{rec}(f)\mathrm{d}f \right|^2. \tag{4}$$

Comparison of patterns of representations given by Eqs. (2)–(4) illustrates the well-known fact, which is treated as a contrast, in general, antenna pattern for UWB signals in the receive and transmit modes. To consider more stringent formulations the categories of antenna pattern, gain, as well as additional properties of UWB antennas as part of radio in the case of far field one can refer to article [2]–[4]. Behavior and properties of antennas in the area near the radiated field are not considered to date. In this paper some results for antenna arrays, focused on UWB signals are presented.

3. Antenna Arrays with UWB Radio Equipment

The antenna array in the near radiated field zone can be represented using a matrix model. As a partial result of these representations, it is possible to completely correct use of the current model as a linear combination of the partial patterns of array elements with weights with the physical meaning of radiating currents. For antenna arrays composed of UWB resources necessary to apply to a complete model of the array [5], [6].

Let us consider antenna array as a union of a finite number of emitters and switchgear (Fig. 1, U_f – incoming wave). For antenna arrays a matrix model with element wise account the effects of mutual coupling of emitters is sufficient, visual and convenient for practical use.

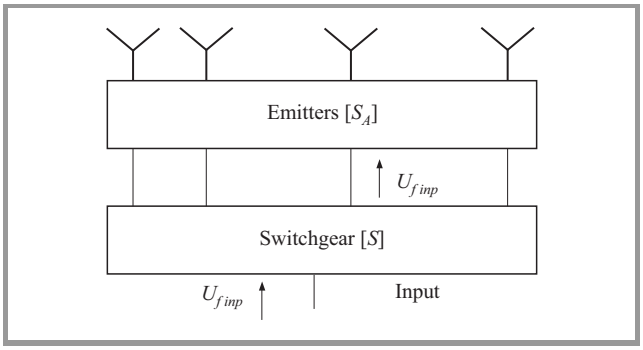


Fig. 1. Antenna array model.

According to the matrix model the distribution of the electric field generated by the emitters system in the near radiated field zone is determined by the scattering matrix of the inputs $[S_A(f)]$. The partial distribution of fields is corresponding to the antenna array element radiation pattern $e_i(x, y, z, f)$ when excited input unit incident wave in the presence of matched loads connected to the other input, and the scattering matrix switchgear $[S(f)]$.

$$[S(f)] = \begin{bmatrix} S_{11}^b(f) & S_{12}^b(f) \\ S_{21}^b(f) & S_{22}^b(f) \end{bmatrix}, \tag{5}$$

where $S_{11}^b(f)$ is an input reflection coefficient of the switchgear, $S_{21}^b(f)$ is the transmission coefficients from input to output and $S_{22}^b(f)$ is the transmission coefficients between the outputs. In addition $S_{12}^b(f) = S_{21}^b(f)$.

For a monochromatic signal with frequency f the spatial field distribution of the antenna array is represented as:

$$E(x,y,z,f) = \langle e(x,y,z,f) | U_f(f) \rangle , \tag{6}$$

where $|U_f(f)\rangle$ is the column vector of complex amplitudes of the incident waves at emitters entrance. Those values are determined by the properties of the emitters and the switchgear:

$$|U_f(f)\rangle = \frac{1}{E - [S^b_{22}(f)][[S_A(f)]]} |S^b_{21}(f)\rangle . \tag{7}$$

Equations (6) and (7) give an accuracy sufficient for most practical purposes, to determine the strength of the electric field in the excitation input switchgear unit amplitude incident wave with frequency f. The representation for non monochromatic signal can be carried out based on the relations (2)–(4). Equations (6)–(7), together with (2)–(4) highlight the fundamental properties of the antenna consisting UWB radio: spatial distribution of the fields generated in the transmission mode and the corresponding parameters in the receiving mode depend (including the frequency dependency) of the scattering matrix dispenser. This means that these parameters cannot be identified apart from properties of the feeder devices in the array.

Accordingly, the current model of the antenna can be used only with certain assumptions. Formally, the spatial distributions of the fields produced by the antenna array in the zone near the radiated field, for each value of the frequency f. In the far field they may be determined as:

$$E(x,y,z,f) = \langle e_I(x,y,z,f) | I(f) \rangle , \tag{8}$$

where $\langle e_I(x,y,z,f)|$ is partial distribution of the fields of array elements defined with respect to single radiating currents in the emitter. The therm "formally" used in this case means that the properties of the array are not determined by its structural elements, but also of the declared frequency characteristics of the radiating currents $|I(f)\rangle$.

Thus Eqs. (6) and (7) together with (2)–(4) allow to determine the electromagnetic field in the near radiated field zone with used properties emitters of antenna array $\langle e(x,y,z,f)|$, $[S_A(f)]$ and the switchgear $[S(f)]$.

Subsequently, the analysis of the main differences between electromagnetic field focused using wideband signals and a monochromatic signal, use the concepts of current model taking into account the reservations made earlier in this paper.

4. Antenna Arrays

The ability to focus the radiation of the electromagnetic field in a certain region of space opens up a number of antenna technology capabilities. The task of building apertures focused, the purpose of operation and properties evaluation was in aim of many papers, for example [7], [8]. Generally, these studies examined organizations focused on the field of a monochromatic signal for coherent radiation of electromagnetic energy. In [9] the following properties of focused apertures are introduced and quantified: energy gain, stealth factor, and the dimension of the focus. Some of them, for distances up to the point of focus of 10 to 90 km, are represented in the Figs. 2 and 3.

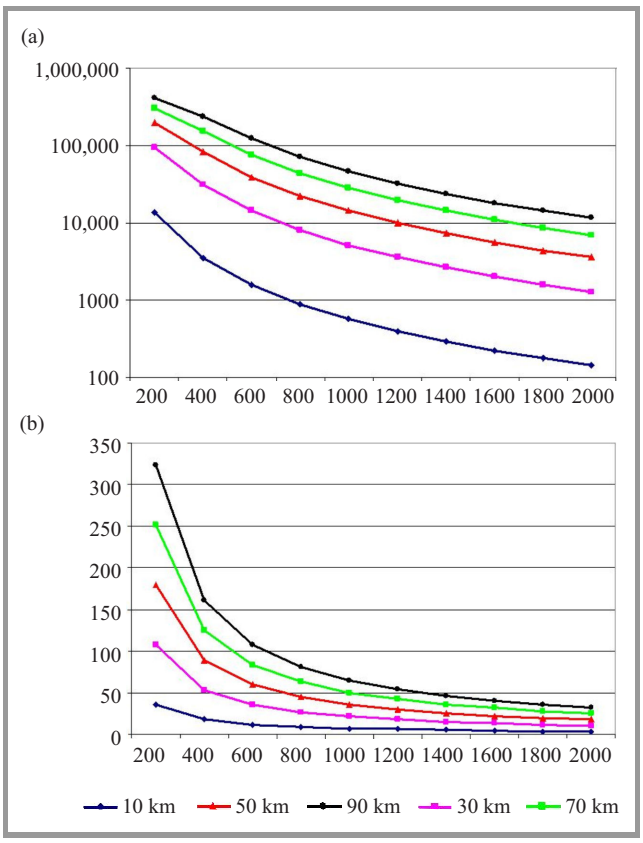


Fig. 2. The size of the area of focus: (a) lengthwise, (b) transverse. (See color pictures online at www.nit.eu/publications/journal-jtit)

The possibility of using wideband signals to solve focus problem opens up new possibilities in the construction of these systems. However, their technical implementation is determined by the properties of transmitters and receivers that work with broadband signals. Articles devoted to focusing on the UWB signals are not known to the general public. Hence, more details would be presented here.

4.1. Transmit Mode

For a discussion of the most significant properties of the focused UWB antenna arrays radio equipment, differentiating them from the case of monochromatic radiation signal, in the case of linear equidistant antenna array (Fig. 4) is considered. In such an area the representation of the electric field in the form is:

$$E_x(x,z,f) = \sum_{-N}^{N} I_n(f) e_{xn}(x,z,nd,f) , \tag{9}$$

$$E_x(x,z,f) = \sum_{-N}^{N} I_n(f) e_{zn}(x,z,nd,f) , \tag{10}$$

$$E(x,z,f)^2 = |e_x(x,z,f|^2 + |e_z(x,z,f)|^2 , \tag{11}$$

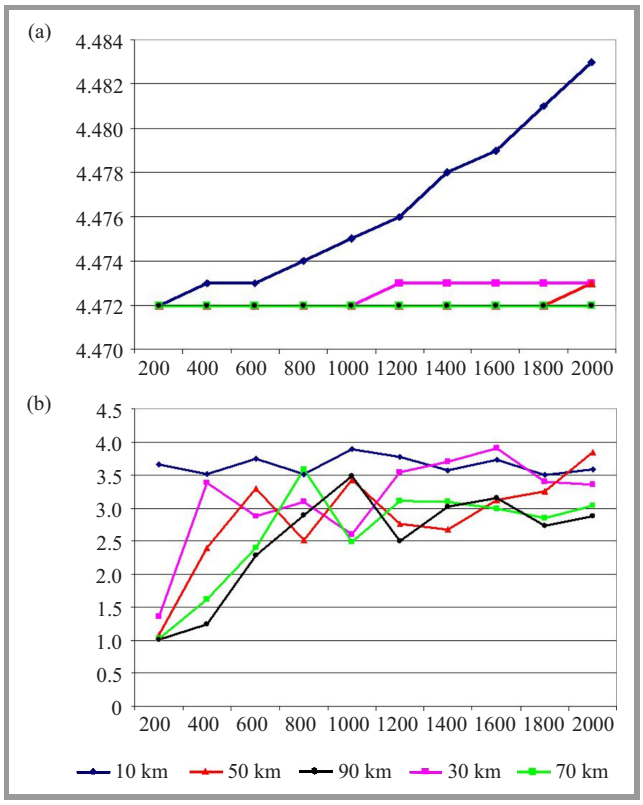


Fig. 3. Properties focused aperture coefficients: (a) efficiency, (b) secrecy.

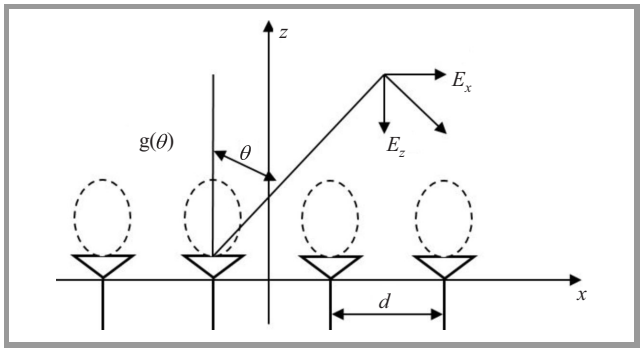


Fig. 4. Linear antenna array.

where:

$$e_{xn}(x,z,nd,f) = \\ = g\big(\theta(x,z,nd,f)\big)\cos\theta(x,z,nd)\frac{e^{jk(f)r}}{k(f)r(x)}, \qquad (12)$$

$$e_{zn}(x,z,nd,f) = \\ = g\big(\theta(x,z,nd,f)\big)\sin\theta(x,z,nd)\frac{e^{jk(f)r}}{k(f)r(x)}, \qquad (13)$$

$g(\theta)$ – the radiation pattern of the emitter

$$\arg\big(I_n(f)\big) = -k(f)r(x_0,z_0,nd). \qquad (14)$$

In many applications the most appropriate indicator of the properties of focused antenna array is the spatial distribution of the electromagnetic energy in the form given by Eq. (2). The picture of the spatial distribution of $|F_{UWB}^{TR.en}(\theta,\varphi,\Delta f)|^2$ to signal spectrum that is symmetric about the center frequency, in essential details similar repeats $E(x,z,f)^2$ at the central frequency signal spectrum is presented in Figs. 5 and 6. The most notable difference is specified for the gratings with a pitch of the order of a wavelength or more, for which there is some reduction in the far side lobes similar diffraction lobe in the antenna array pattern in the far field. In Figs. 5–6 the antenna array of 11 emitters with a pitch equal to the wavelength at the center frequency f_0 is used, radiation pattern of the element – $\cos\theta$, polarization – along the aperture, and the range of power-triangular. Focus point (0.5λ), (a) the relative bandwidth $\Delta f/f_0 = 0.01$, (b) $2\Delta f/f_0 = 0.5$.

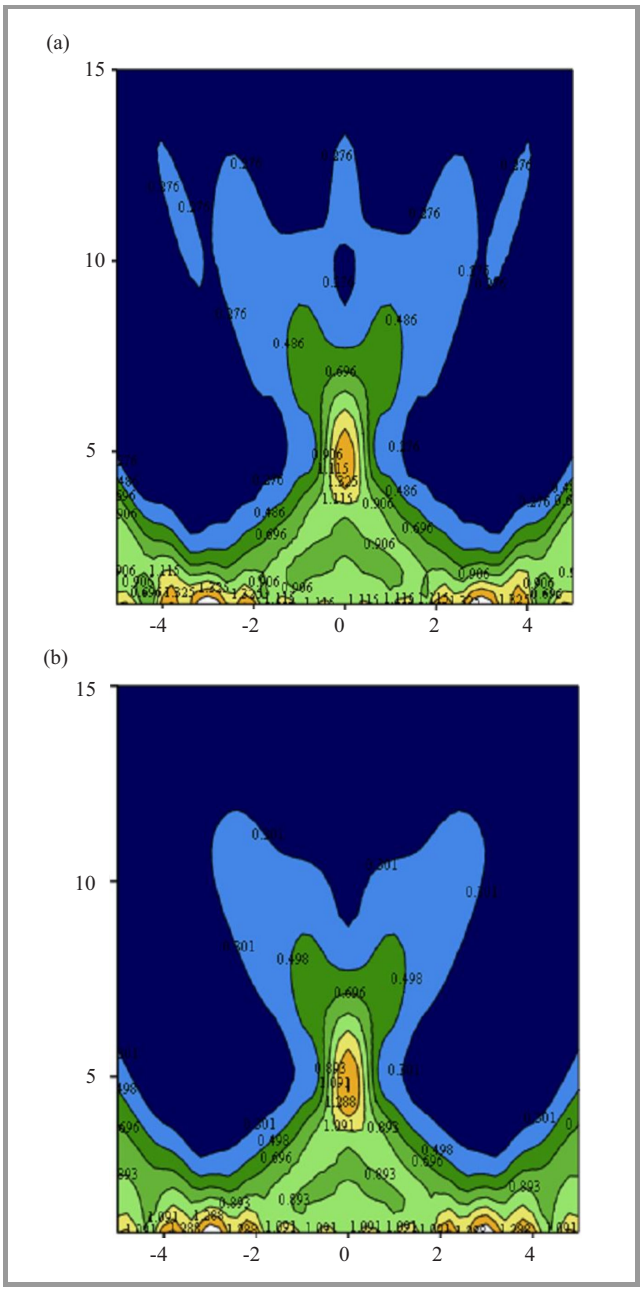


Fig. 5. Spatial distributions $|F_{UWB}^{TR.en}(\theta,\varphi,\Delta f)|^2$.

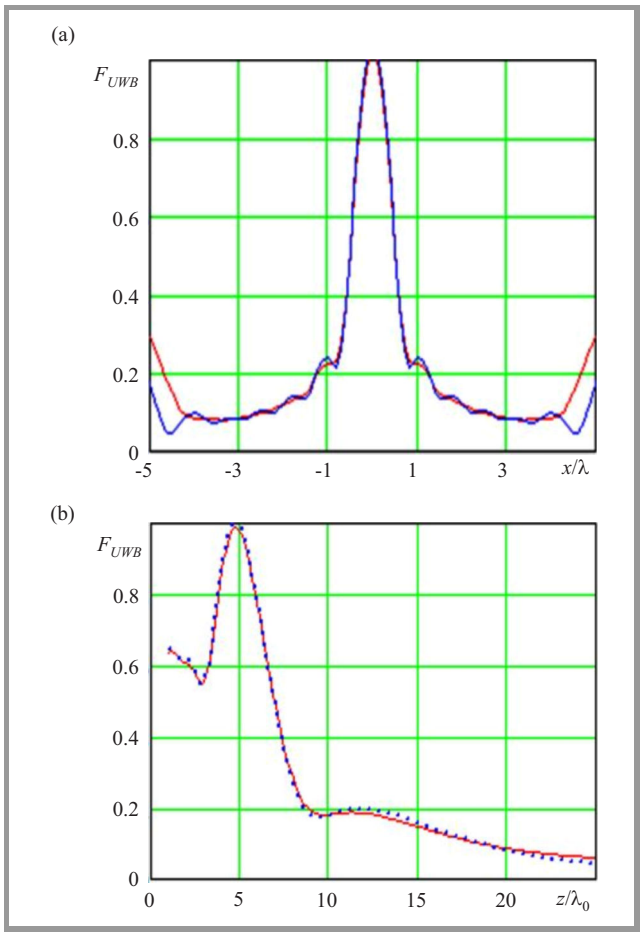


Fig. 6. Spatial distributions $|F_{UWB}^{TR.en}(\theta,\varphi,\Delta f)|^2$: (a) in a direction parallel to the aperture, (b) perpendicular to the aperture.

Thus, in a focused antenna array for broadband signals symmetric about the center frequency f_0 spectrum holds important feature: the spatial distribution of energy near the focus point substantially coincides in shape with the spatial distribution of the squared modulus of the electric field at f_0.

4.2. Receive Mode

Receiving UWB signals using multielement systems (MIMO technology) [10] can be carried out in various embodiments. In this section, the case when the antenna array is made the traditional way, and operates as part of an UWB receiver that linear filtering signals from its output (Fig. 7) is considered.

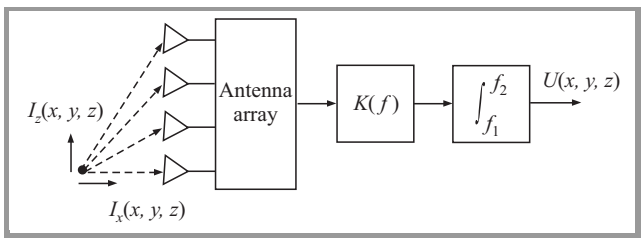


Fig. 7. Antenna array in receive mode.

For linear antenna the amplitude of the output signal $U_x(x,z)$ when receiving the radiation source with unit amplitude spectrum $G(f)$ (polarized parallel to the aperture and located at the point (x,z)) is:

$$U_x(x,z) = \int_{f_1}^{f_2} K(f)\,G(f) \sum_{-N}^{N} I_n(f)\,e_{xn}(x,z,nd,f)\,. \qquad (15)$$

Similarly, for a source of polarized perpendicular to the aperture:

$$U_z(x,z) = \int_{f_1}^{f_2} K(f)\,G(f) \sum_{-N}^{N} I_n(f)\,e_{zn}(x,z,nd,f)\,. \qquad (16)$$

Equations (15) and (16) characterize the spatial selectivity of the system "array plus receiver". As can be seen from (15) and (16) directional properties depend not only on the parameters of the antenna array, i.e. array step and amplitude-phase distribution $I_n(f)$, $-N \le n \le N$, but also on the range signal $G(f)$ in the frequency band $f_1 \le f \le f_2$ and the weighting function $K(f)$.

The function defined spatial selectivity of the system in receiving mode should be different from the corresponding values for monochromatic signal at frequency f within a given band, characterized by values $E_x(x,z,f)$ and $E_z(x,z,f)$ – Eqs. (9)–(13).

Let us consider qualitatively essence of these differences, assuming that the amplitude distribution $I_n(f)$ does not depend on the frequency and phase – corresponds to the focus point $(0,z_0)$, located on the axis of symmetry of the antenna array in the direction normal to the aperture. In this case, antenna array receive only from the source of radiation polarized parallel to the aperture. The spatial characteristics of the electoral system in the receiving mode is the value of $U_x(x,z)$.

4.3. Characteristic Properties

In this subsection consideration of the characteristic properties and difference from the case radio reception focused grating monochromatic signal to conduct linear equidistant grating with a uniform frequency independent amplitude distribution is provided.

The character of magnitude changes $U_x(x,z)$ when receiving a wideband signal is characterized by a significant improvement in the focusing direction perpendicular to the aperture. Change in the plane parallel to the aperture is not so significant. The width of the main peak is located at a symmetric spectrum of the emitted signal. The side lobe levels are reduced and are almost unchanged. The noticeable decrease is in the side lobes arising from the lattice spacing a $0.7\ldots1$ wavelength at the average frequency spectrum of the signal – see Fig. 8.

The signal spectral composition is symmetric about the center frequency, as would be expected, and there has been a noticeable change in the peak width. The most significant changes are related to the side lobes levels for signals with

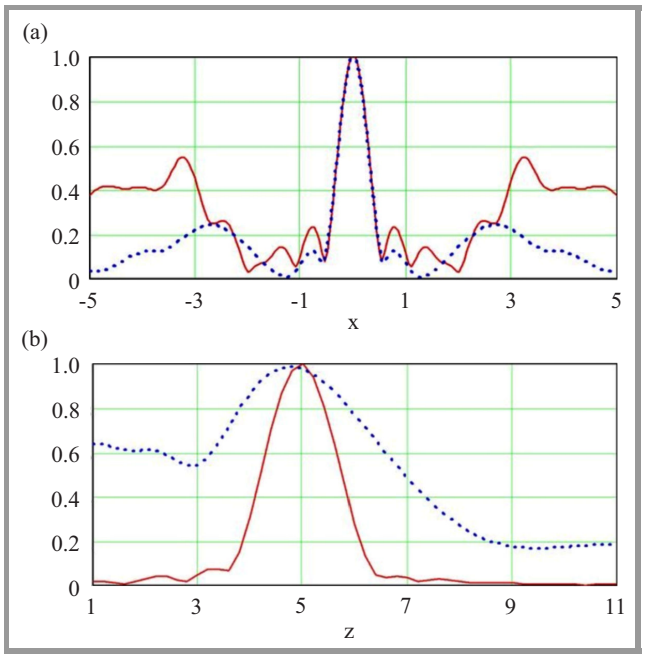


Fig. 8. The spatial dependence of the $U_x(x,z)$ amplitude for the monochromatic signal (solid line) and a wideband signal (dotted line) in the direction of: (a) a parallel aperture, (b) perpendicular to the aperture.

a predominance of high frequency components the side-lobe levels increase significantly with a slight decrease in the direction normal to the antenna – Fig. 9.

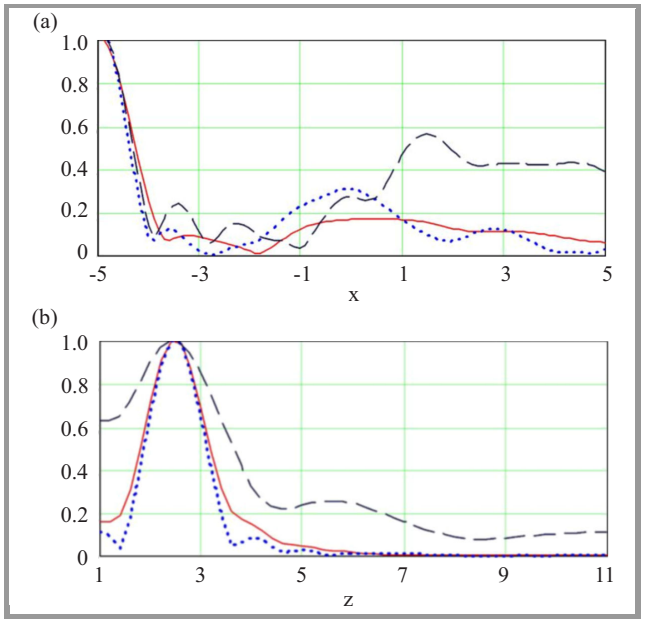


Fig. 9. Functions: (a) $U_x(x,z_0)$, and (b) $U_x(0,z)$ in a monochromatic signal (dashed line) and the signal with the spectrum of type $1-f/f_0$ (solid line) and f/f_0 (dotted line).

The type of used weight function also significantly affects the characteristic spatial selectivity. In assessing this effect it makes sense to compare the two versions of the weighting functions: $K(f)=1$ (the receiver with a uni-

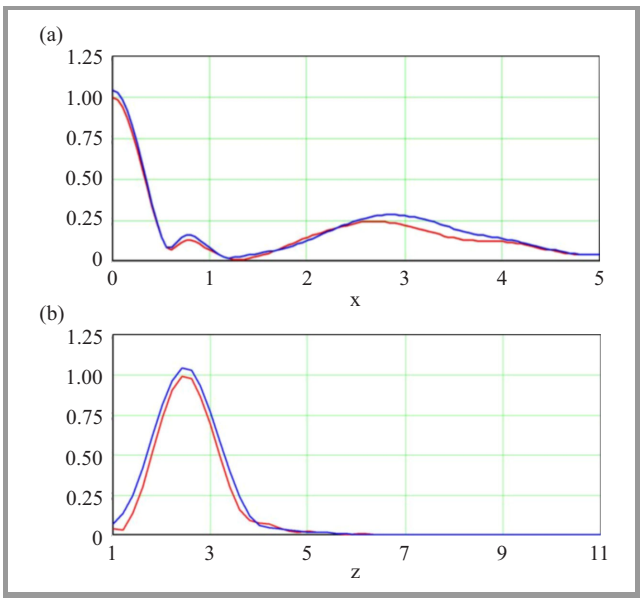


Fig. 10. Curves of: (a) $U_x(x,z_0)$ and (b) $U_x(0,z)$ for the processing of the received signal with a uniform and matched filtering.

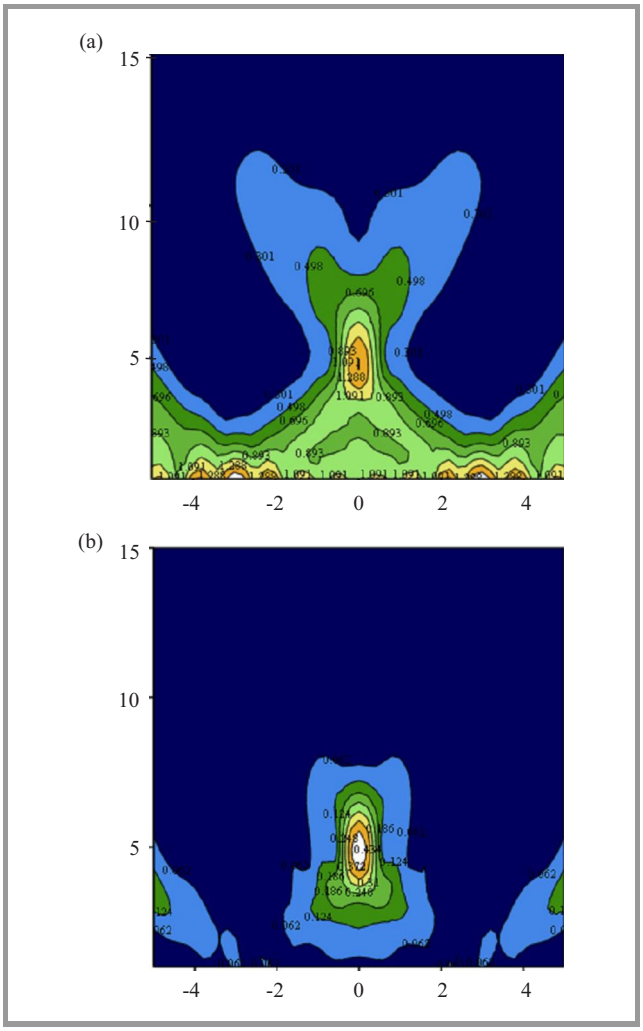


Fig. 11. Curves of: (a) $U_x(x,z_0)$ and (b) $U_x(0,z)$ for the processing of the received signal with a uniform and matched filtering.

form frequency response) and $K(f) = |G(f)|$, corresponding to the matched filtering case. As the assessment conducted under symmetric spectral function $G(f)$, the difference from the functions of spatial selectivity $U_x(x,z)$ at the uniform and consistent filtration is not observed. However, as one would expect the level of $U_x(x,z)$ is at the focal point above – Fig. 10.

5. Conclusions

To sum up, once again draw attention to the difference function, which characterizes the spatial selectivity in receiving mode $|F_{UWB}^{REC}|^2 = |U_x(x,z)|^2$ and functions $|F_{UWB}^{TR.en}(\theta,\varphi,f)|^2$, determining the spatial distribution of energy in the transmit mode. One can see, that at wide band signal, this difference can be considerable (Fig. 11).

In all cases, the antenna array consisted of 11 emitters with a uniform amplitude distribution and orientation diagrammatic form $\cos\theta$ increments of the wavelength is focused on the axis of symmetry at a distance of five wavelengths. The signal spectrum is triangular in the band of $\pm 5\%$ f_0 (narrowband) and $\pm 25\%$ f_0 (broadband). In the transmission mode field distribution in space – energy EMF. In receive mode, and in the same conditions, but in the assumption of the matched filter with a frequency response spectrum of the signal, and when receiving basic components of the field, i.e. antenna is oriented to receive the main components.

Acknowledgment

This work was supported by the Russian Science Foundation (grant no. 15-19-10053).

References

[1] L. S. Benenson, *Ultra-wideband Antennas*. Moscow: Mir, 1964 (in Russian).

[2] M. Bulatov, A. P. Ovcharov, and Yu. E. Sedelnikov, "Features ultra-wideband radio antennas", *Herald MarSTU*, no. 1, pp. 13–24, 2011 (in Russian).

[3] G. I. Sagadeev and Yu. E. Sedelnikov, "Synthesizing the directional properties of antennas for the aeronautical mobile radio using broadband signals", *Phys. of Wave Process. & Radio Syst.*, no. 2, pp. 21–26, 2007 (in Russian).

[4] A. P. Ovcharov, "Direction and energy characteristics of antennas for UWB short-range radar", M.Sc. thesis, Kazan National Research Technical University, 2007, Kazan, Russia (in Russian).

[5] G. A. Morozov and Yu. E. Sedelnikov, *Low-Intensity Microwave Technology (Issues and Implementation)*. Moscow: Radiotechnika, 2003 (in Russian).

[6] A. P. Ovcharov and Yu. E. Sedelnikov, Antenna arrays for ultra-wideband radio", *Radio*, no. 1, pp. 22–28, 2014 (in Russian).

[7] D. A. Vedenkin, O. V. Potapova, Yu. E. Sedelnikov, "Antennas, focused in the near radiated field zone. Features and technical application" in *Proc. 9th Int. Conf. Antenna Theory & Techniq. ICATT 2013*, Odessa, Ukraine, 2013, pp. 560–565.

[8] D. A. Vedenkin, "Focused antenna array as part of small groups radio-electronic means of unmanned aerial vehicles", M.Sc. thesis, Kazan State Technical University A. N. Tupolev, Kazan, 2012 (in Russian).

[9] D. A. Vedenkin and Yu. E. Sedelnikov, "Active focused antenna arrays for radio technique equipment of small unmanned aerial vehicles", *Phys. of Wave Process. & Radio Syst.*, vol. 11, no. 4, pp. 40–46, 2008 (in Russian).

[10] M. G. Bakulin, L. A. Varukina, and V. B. Kreyndelin, *Technology MIMO: the Principles and Algorithms*. Moscow: Hotline-Telecom, 2014 (in Russian).

Denis Andreevich Vedenkin holds Ph.D. degree and currently is an Assistant Professor in Radio Photonics and Microwave Technologies Department of Kazan National Research Technical University A. N. Tupolew (KNRTU-KAI). He is also senior researcher in Department of Radio Engineering and Communications of Volga State University of Technology. His research interests includes: antenna, microwave devices, focusing aperture, microwave technology, radio photonics, telecommunication technologies.

E-mail: denis_ved@mail.ru
Kazan National Research Technical University
named after A. N. Tupolew
Institute of Radio Electronic and Telecommunication
Karl Marks st 10
Kazan, Tatarstan
420111 Russian Federation

Faculty of Radio Engineering
Volga State University of Technology
Ploszczad Lenina 3
Yoshkar-Ola, Mari El Republic
424000 Russian Federation

Yuri Evgenevich Sedelnikov is Ph.D. and Professor, Honored Worker of Science and Technology of the Republic of Tatarstan. He is a professor in Department of Electronic and Telecommunications Systems of Kazan National Research Technical University A. N. Tupolew (KNRTU-KAI). His research interests includes: antenna, microwave devices, focusing aperture, microwave technology, radio photonics, etc.

E-mail: sedhome2013@yandex.ru
Kazan National Research Technical University
named after A. N. Tupolew
Institute of Radio Electronic and Telecommunication
Karl Marks st 10
Kazan, Tatarstan
420111 Russian Federation

Aydar Revkatovich Nasybullin holds Ph.D. degree and currently is an Assistant Professor in Radio Photonics and Microwave Technologies Department of Kazan National Research Technical University A. N. Tupolew (KNRTU-KAI). He is also senior researcher in Department of Radio Engineering and Communications of Volga State University of Technology. His research interests includes: antenna, microwave devices, focusing aperture, microwave technology, radio photonics, telecommunication technologies.

E-mail: aydar.nasybullin@mail.ru
Kazan National Research Technical University
named after A. N. Tupolew
Institute of Radio Electronic and Telecommunication
Karl Marks st 10
Kazan, Tatarstan, 420111 Russian Federation

Faculty of Radio Engineering
Volga State University of Technology
Ploszczad Lenina 3
Yoshkar-Ola, Mari El Republic
424000 Russian Federation

Paper

Radio Photonic Systems for Measurement of Instantaneous Radio Frequency with Amplitude-phase Modulation of Optical Carrier

Oleg G. Morozov[1], Aydar R. Nasybullin[1,2], Denis A. Vedenkin[1,2], and Timur A. Agliullin[1]

[1] *Institute of Radio Electronic and Telecommunications, Kazan National Research Technical University named after A. N. Tupolew, Kazan, Russian Federation*

[2] *Faculty of Radio Engineering, Volga State University of Technology, Kazan, Russian Federation*

Abstract—In this article questions related to the development instantaneous radio frequency measurement system based on application in them original ways of the amplitude-phase modulation transformation of single-frequency optical carrier by a radio signal in symmetric two-frequency and measuring "frequency-amplitude" transformation in fiber Bragg grating with special profile are considered. Such systems have broad prospects for use in telecommunications, military systems and for environmental monitoring.

Keywords—amplitude-phase modulation, transformation, fiber Bragg grating, instantaneous frequency, radio photonics.

1. Introduction

With development of radio photonics the measurement systems of radio signals instantaneous frequency (MSRSIF) become one of perspective tools applied in various military and civil systems, designed on the principles of complex processing of radio signals in the optical range of electromagnetic waves.

In comparison with classical methods [1], which use radioelectronic multichannel technologies, radio photonic MSRSIF have essential advantages on the wide frequency and amplitude range of measurements, small losses, a high electromagnetic noise stability, and also simplicity, compactness and small weight [2]. Traditional electronic technologies of MSRSIF [1] estimating phase change rate, by selecting radio signal carrier (continuous or pulsed) with highest amplitude. The system receives signal measurements for a fixed period time in a multi frequency device, where multi-channel feature is designed by narrowband phase discriminator. Radio photonic MSRSIF technology [2] includes modulation transformation of optical carrier by radio signal. With the "time-frequency", "frequency-space" or "amplitude-frequency" measurement conversion types, opto-electronic transformation in the photodetector and calculating temporal, spatial or amplitude ratio function is uniquely dependent on the measured signal frequency (compared to reference) to eliminate the influence of instabilities.

The modulation of intensity or phase in Mach-Zehnder Modulator (MZM), parallel modulation of the intensity and phase in the polarization modulator (PolM) are used for modulation transformation [3]–[5]. There is conducted research method on the using only amplitude modulation with suppressed optical carrier in MZM and PolM circuits, during their operation in the "zero" point of the modulation characteristic.

MSRSIF is seen as the most forward-coding technologies of determining the frequency with "amplitude-frequency" transformation type in optical mediums, including those in Fiber Bragg Grating (FBG) [6]–[8]. Benefits of using FBG are in a unique transformation of the measured frequency to the amplitude of the reflected or transmitted radiation of optical carrier, modulated by RF signal, and the possibility of a simple fabrication. The main disadvantages of the FBG are monotone measuring conversion to the central wavelength and the high level of gratings response on temperature and deformation. In typical implementation the amplitude and phase modulation the bandwidth defined sidebands are usually two or four times the measured frequency. This leads to the need for "wide-band" FBG, with a low reflectance and monotonic profile in the resonance region.

In recent years, a significant attention is paid to determining the spectral characteristics of the optical fiber selective structures based on the use as probing signal a symmetrical two-frequency or polyharmonic continuous wave with suppressed carrier. The signal is synthetized from the carrier using its sequential amplitude phase modulation transformation (AFMT) by the Ilyin-Morozov method. It is based on 100% amplitude modulation of a single frequency coherent radiation with sequential switching on the π phase, while the envelope of amplitude modulated radiation passing the minimum [9]–[12]. Its features include high spectral purity of the output radiation and the conversion factor, as well as the possibility of obtaining the difference frequency equal to the modulation frequency. The latter feature was never before used in applications of optoelectronic systems, and is the basis for this work.

The conducted analysis allowed to formulate the design requirements for the radio photonic MSRSIF, which are the need to use in its structure only one laser, a minimum of one modulator, one FBG and possibly only one narrowband photodetector [13]–[17]. General requirements for the monitoring channel of operating modes are determined by the need to use a symmetric dual-frequency radiation to control the position of the FBG central wavelength and control the amplitude modulators operating point [18], [19].

2. The Amplitude-phase Modulation Conversion of Optical Carrier

In this section authors briefly discuss features of AFMT and forming signal a symmetrical dual-frequency radiation with suppressed carrier by radio. In the analysis a classic generalized scheme of single-port radio photonic conversion unit for modulating the optical carrier is considered [20], which has been transformed from parallel to serial circuit type (Fig. 1) in order to implement the AFMT method of Ilyin-Morozov.

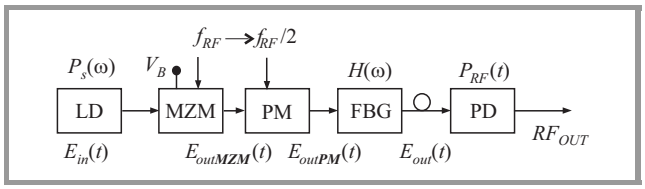


Fig. 1. A generalized block diagram of serial type single-port radio photonic unit of modulation conversion: LD – laser diode, MZM – Mach-Zehnder modulator, PM – phase modulator, FBG – fiber Bragg grating, PD – photodetector.

The investigations showed the possibility of AFMT implementing based on sequentially arranged amplitude and phase MZM. The obtained equations were used to calculate the spectrum of the radiation at the output of modulators. It was shown that the AFMT implementation to obtain the difference frequency equal to the measured frequency of the radio signal is possible. The optimal modulation factor $m = 5/9$ in the case of modulating type oscillations as $S_1(t) = S\cos(2\Omega t + \pi)$ and $m = 1$ in the case of oscillations of the form as $S_2(t) = S|\sin\Omega t|$, where $\Omega = f_{RF}/2$ and f_{RF} – radio frequency. The radiation spectrum for the two components of the modulator output in the case $S_1(t)$ is described by:

$$E_{APM}(t) = 0.49E_0\{\sin(\omega_0+\Omega)t - \sin(\omega_0-\Omega)t\} - \\ -0.007E_0\{\sin(\omega_0+3\Omega)t - \sin(\omega_0-3\Omega)t\} + \dots, \quad (1)$$

and in case of the modulation by oscillation $S_2(t)$:

$$E_{APM}(t) = 0.56E_0\{\sin(\omega_0+\Omega) - \sin(\omega_0-\Omega)t\} + \\ +0.05E_0\{\sin(\omega_0+3\Omega)t - \sin(\omega_0-3\Omega)t\} + \dots, \quad (2)$$

As can be seen from Eqs. (1) and (2) the difference frequency between the components of the two-frequency radiation 2Ω equal to modulating signal frequency f_{RF}. The higher harmonics can be ignored due to their low amplitudes. The narrowing of the difference frequency is twice more as compared with the classical scheme of its doubling applied in known radio photonic MSRSIF, using a single amplitude MZM working at zero point of the modulation characteristics to suppress the carrier.

3. Improving the Metrological Characteristics of the Radio Photonic MSRSIF

This section describes three methods of improving the characteristics of existing radio photonic MSRSIF: widening the range, increasing the measurements resolution at low frequencies and sensitivity at high measured frequencies.

Let us consider a short theoretical substantiation of MSRSIF method with the widening conversion measurements in twice frequency range. The block diagram for its implementation is shown in Fig. 2a, an explanation of the operating principle is shown in Fig. 2b, and the measuring characteristics is presented in Fig. 2c. The optical carrier from a laser diode LD at the frequency f_0 is supplied to the M2M and PM modulators block MB where is modulated by unknown frequency f_{RF} and amplitude A_{RF} microwave signal, and then divided into two channels in the optical splitter OS.

On first channel the radiation through the circulator C entrance is fed to FBG, and then reflected from it and from the output C is applied to the first photodetector PD1. The second channel with PD2 is used as a reference. The microcontroller MCU detects the microwave signal and computes the parameters according to the ratio of the signals amplitudes from PD1 and PD2 output using the amplitude comparison function, that uniquely depends on the frequency of the microwave signal and does not depends on the power of the laser radiation. According to the Eqs. (1) or (2) the output radiation from MB is a dual-frequency with $f_0 - f_{RF}/2$ and $f_0 + f_{RF}/2$ sidebands and suppressed carrier f_0. Amplitude of the components A_{-1} and A_{+1} are determined by the value of the Bessel function.

The output current of PD2 from the reference channel is proportional to

$$i(t) \propto A_{-1}^2 + A_{+1}^2 + A_{-1}A_{+1}\cos 2\pi f_{RF}t\,, \quad (3)$$

and DC detecting power at $A_{-1} = A_{+1} = A_1$:

$$P_{RF} \propto A_{-1}^2 + A_{+1}^2 = 2A_1^2. \quad (4)$$

For amplitude-frequency measurement conversion, a classical FBG with a Gaussian profile of the envelope with center frequency f_B is chosen, which is characterized by dependence

$$R(f_{RF}) = R_B\,\mathrm{e}^{-4\ln 2\left(\frac{f_{RF}-f_B}{\Delta f_B}\right)^2}, \quad (5)$$

where: R_B – reflectance at f_0, Δf_B – full width at half maximum of FBG.

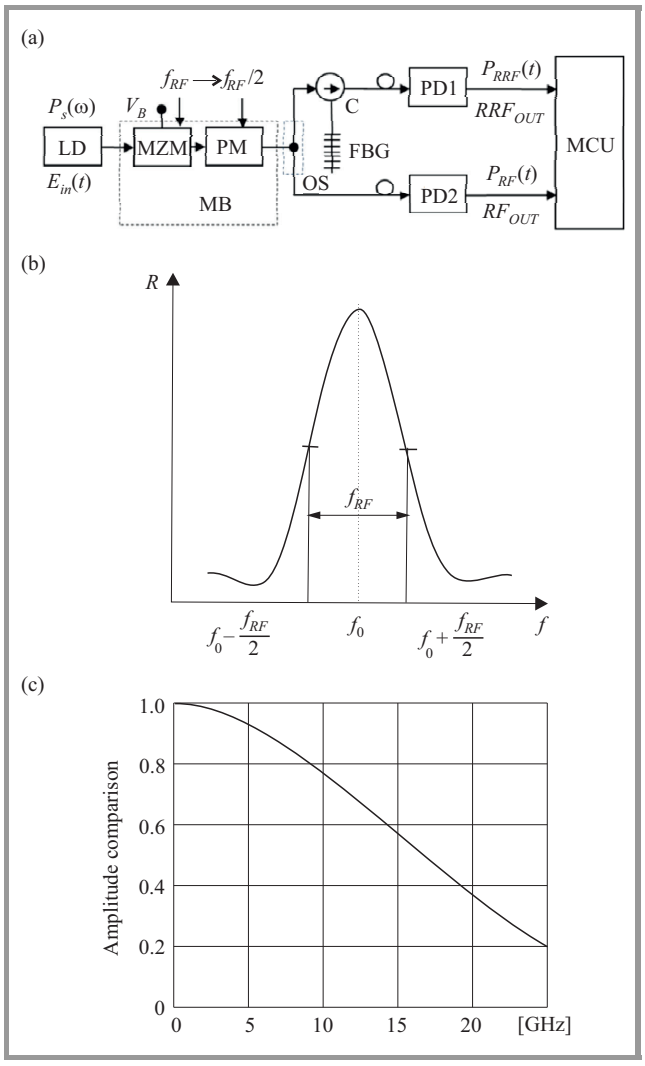


Fig. 2. To theoretical justification of the way of expand the range of measured frequencies: (a) block diagram, (b) spectral representation, (c) measuring characteristics.

Thus, when the FBG center frequency is tuned to the frequency of the optical carrier, the two-frequency components of radiation reflected from the FBG (depending on the frequency) have amplitude:

$$A_{-1}^{R}RF = A_{+1}^{R}RF = A_1 R(f_{RF}), \tag{6}$$

and the PD1 output power in the measuring channel:

$$P_{RRF} \propto = 2A_1^2 R^2(f_R F). \tag{7}$$

The frequency dependent function of the power ratio, defined as $\rho = P_{RRF}/P_{RF}$, equal to:

$$\rho = R^2(f_{RF}). \tag{8}$$

Figure 2c shows a unique dependence between ρ and f_{RF}, calculated in Matlab, which is independent of the laser power and the power of the microwave signal.
Thus, knowing only the power ratio ρ, the instantaneous frequency of microwave signal f_{RF} could be determined.

The amplitude of the unknown microwave signal is determined by the output signal of PD2 at the calibrated power of DFB-laser and the known characteristics of the modulation conversion in the modulator block.
The difference frequency between the components of the two-frequency signal is equal to the measured frequency, allowing for the twice extension the range of measured frequencies in comparison with the classical methods of MSRSIF at a predetermined full width at half maximum of FBG.

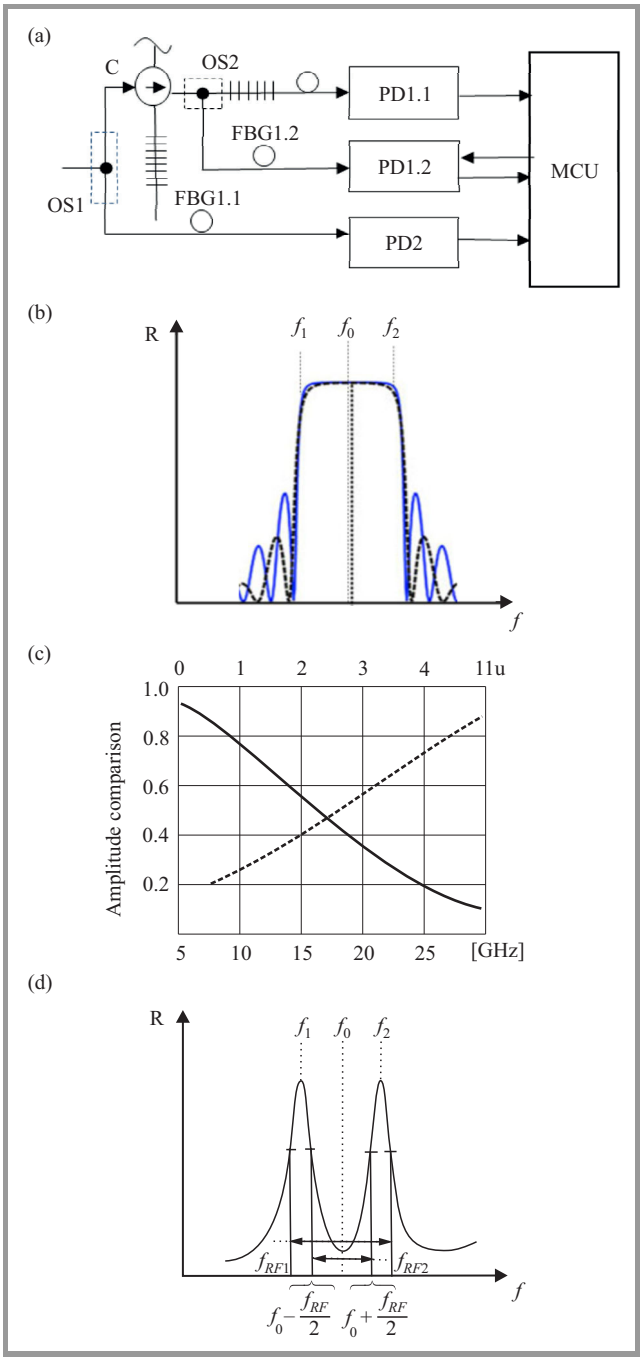


Fig. 3. MSRSIF test circuit for increasing the resolution of the measurements in the area of low measured frequencies: (a) block diagram, (b) spectral representation, (c) measuring characteristics, (d) contour of band stop FBG1.2.

In the next step let us consider a method to improve the measurements resolution on the low measured frequency. When using classical FBG in this area the monotonous envelope is observed (Fig. 2b), resulting in a reduction of measurement resolution. To improve FBG with π-phase shift and frequency response with a special form has been used, which has a transparency window in the low frequencies. It allows to create dual-band setup. The block diagram of the setup is shown in Fig. 3a, an explanation to the principle of the work is presented in Fig. 3b, and measuring characteristics is shown in Fig. 3c.

To separate signal ranges FBG1.2 was used with rectangular form (Fig. 3d) in the frequency range from f_1 to f_2 ($\pm$4 GHz) to provide lower and higher channels and a special switching algorithm in the MCU. Above 4 GHz PD1.1 and PD1.2 channel blocked by MCU. In frequencies up to 4 GHz the PD1.2 channel is activated by MCU, because the signal level in the channel with PD1.1 lies below a predetermined threshold.

The results of numerical simulations in Matlab, confirming the possibility of increasing the resolution in the low frequencies to the level of resolution in the medium and high frequencies of 0.8–1 GHz/dB are shown in Fig. 3c.

Based on the analysis of noise characteristics of radio photonic systems, implementing a variety of receiving variants of two-frequency radiation (in the band of the measured frequency and constant component) a technique of splitting components of two-frequency radiation at a fixed difference frequency $f_{DF} = 100$ MHz is proposed, which is in the region of photodetector minimal noise (Fig. 4a). This provided a narrow-band receiving and registration of the measured components amplitude according to the envelope of split components (Fig. 4b) on f_{DF} when filtering in units of frequency selection UFS (narrow-band filter with a center frequency of 100 MHz and a minimum bandwidth defined by the width of the laser).

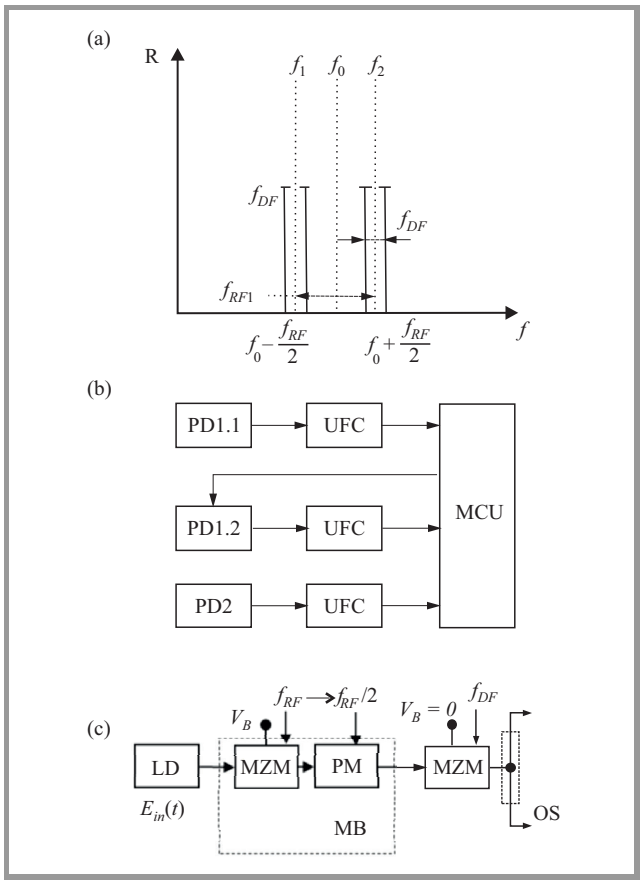


Fig. 4. MSRSIP test circuit for increasing the sensitivity of measurements in the field of high measured frequency: (a) spectral representation, (b) block diagram of a receiver part, (c) block diagram of a transmitter part.

The measurement sensitivity was 3-6 times improved as compared to direct detection method in the frequency band. The impact of photodetectors flicker noise on the accuracy of amplitude measurements was reduced.

Block diagram of developed radio photonic MSRSIF at a level of modulation conversion is shown in Fig. 4c. It differs from circuit shown in Fig. 2a, having the second amplitude MZM operating in a mode of suppression of the carrier frequency in the zero point. A imitating modeling of proposed methods is provided in the application package OptiSystem simulated in OptiGrating 4.2.

The improvements of functional parameters were confirmed by the results of theoretical research and by developed radio photonic MSRSIF, which takes into account benefits of the all developed methods.

Measurement of the instantaneous frequency in a lower band of radio frequencies (VHF and UHF) requires the use of FBG with of ultra-narrow bandwidth, for example with a phase shift and providing stable working conditions in high temperatures.

4. Conclusions

The analysis of information on existing and prospective radio photonic MSRSIF with a measurement conversion amplitude-frequency in FBG, allowed to improve their metrological and technical and economic characteristics. The paper show that a further development of these systems may be based on the use therein AFMT of optical carrier, for measuring the instantaneous frequency, and to provide a stable operating mode of conversion devices.

The used AFMT of optical carrier to symmetric dual-frequency radiation, by splitting its components to a fixed difference frequency, located in the region of minimal noise of photodetector, with dual-band frequency-amplitude measurement conversion in FBG and a special form of optoelectronic conversion in narrowband photodetector, allowed to:

- twice extension of MSRSIF band,
- increasing the resolution of the measurements in the field of low frequency to the level of 0.8–1 GHz/dB,
- increasing measurement sensitivity by 3–6 times compared with the broadband photodetection,
- reducing the impact of low-frequency fluctuations in the amplitude accuracy.

Acknowledgment

This work was supported by the Russian Science Foundation (grant no. 15-19-10053).

References

[1] P. W. East, "Fifty years of instantaneous frequency measurement", *IET Radar Sonar Navig.*, vol. 6, no. 2, pp. 112–122, 2012.

[2] P. Ghelfi *et al.*, "A fully photonics-based coherent radar system", *Nature*, vol. 507, pp. 341–345, 2014.

[3] X. H. Zou, S. L. Pan, and J. P. Yao, "Instantaneous microwave frequency measurement with improved measurement range and resolution based on simultaneous phase modulation and intensity modulation", *J. Lightw. Technol.*, vol. 27, no. 23., 2009, pp. 5314–5320.

[4] J. Q. Zhou *et al.*, "Photonic measurement of microwave frequency based on phase modulation", *Opt. Express*, vol. 17, no. 9, pp. 7217–7221, 2009 (doi: 10.1364/OE.17.007217).

[5] W. Li, N. H. Zhu, and L. X. Wang, "Reconfigurable instantaneous frequency measurement system based on dual-parallel Mach-Zehnder modulator", *J. of Photon.*, vol. 4, no. 2, pp. 427–436, 2012.

[6] Z. Li *et al*., "Photonic instantaneous measurement of microwave frequency using fiber Bragg grating", *Opt. Commun.*, vol. 283, no. 3, pp. 396–399, 2010.

[7] X. Zhang *et al*., "Microwave frequency measurement using fiber Bragg grating as V-shape filter", in *Proc. Int. Conf. Digit. Manufac. & Autom. ICDMA 2010*, Changcha, China, 2010, pp. 921–924.

[8] Z. Li *et al*., "Instantaneous microwave frequency measurement using a special fiber Bragg grating", *IEEE Microw. & Wirel. Compon. Lett.*, vol. 21, no. 1, pp. 1346–1348, 2011.

[9] O. Morozov *et al*., "Synthesis of two-frequency symmetrical radiation and its application in fiber optical structures monitoring", in *Fiber Optic Sensors*, M. Yasin, Ed., InTech, 2012, Ch. 6, pp. 137–165 [Online]. Available: http://www.intechopen.com/books/fiber-optic-sensors/synthesis-of-two-frequency-symmetrical-radiation-and-its-application-in-fiber-optical-structures-mon

[10] O. G. Morozov, D. L. Aibatov, and T. S. Sadeev, "Synthesis of two-frequency radiation and its use in fiber-optic systems, distributed and multiplexed measurements", *Physics of Wave Process. & Radio Syst.*, vol. 13, no. 3, pp. 84–91, 2010 (in Russian).

[11] O. G. Morozov and G. I. Ilyin, "The amplitude-phase modulation radiophotonic systems", *Bull. Volga State Technol. Univ.. Series: Radio Engin. and Inform. and Commun. Syst.*, no. 1, pp. 3–42, 2014 (in Russian).

[12] A. A. Sevastyanov *et al*., "Formation of multi-frequency radiation in the dual-port Mach-Zehnder modulator", *Scientific & Technical Gazette Volga*, no. 4, pp. 232–236, 2013 (in Russian).

[13] O. G. Morozov *et al*., "Measurement of instantaneous frequency using a dual frequency probing", *Scientific & Technical Gazette Volga*, no. 4, pp. 146–149, 2012 (in Russian).

[14] O. G. Morozov *et al*., "Measurement of the instantaneous frequency of the microwave radio signals in the optical domain to transform the "frequency-amplitude" in the fiber Bragg grating with a phase shift", *Bull. Volga State Technol. Univ.. Series: Radio Engin. and Inform. and Commun. Syst.*, no. 3, pp. 30–41, 2013 (in Russian).

[15] O. G. Morozov *et al*., "Instantaneous frequency measurement using double-frequency probing", *Proc. of SPIE*, vol. 8787, Optical Technologies for Telecommunications 2012, 878708, Mar. 2013 (doi: 10.1117/12.2017834).

[16] O. G. Morozov, M. R. Nurgazizov, and A. A. Talipov, "Double-frequency method for the instantaneous frequency and amplitude measurement", in *Proc. IX Int. Conf. Antenna Theory & Techniq. ICATT'13*, Odessa, Ukraine, 2013, pp. 381–383.

[17] O. G. Morozov *et al*., "Instantaneous frequency measurement of microwave signals in optical range using "frequency-amplitude" conversion in the π-phase-shifted fiber Bragg grating", *Proc. of SPIE*, vol. 9136, p. 91361B (doi: 10.1117/12.2051126).

[18] O. G. Morozov *et al.*, "Instantaneous microwave frequency measurement with monitoring of system temperature", *Proc. of SPIE*, vol. 9156, 2014, p. 91560N (doi: 10.1117/12.2054256).

[19] M. R. Nurgazizov, "Two-frequency measuring system instantaneous frequency microwave radio with temperature stabilization", *Modern Problems of Science and Education*, no. 4 (54), 2014 [Online]. Available: www.science-education.ru/118-14134 (in Russian).

[20] I. Gasulla and J. Capmany, "Analytical model and figures of merit for filtered microwave photonic links", *Opt. Express*, vol. 19, no. 20, pp. 19758–19774, 2011 (doi: 1.1364/OE.19.019758).

Oleg Gennadyevich Morozov is a Ph.D. and Professor, and Head of Radio Photonics and Microwave Technologies Department at the Kazan National Research Technical University named after A. N. Tupolew (KNRTU-KAI). His research interests includes: fiber optic sensor systems, radio photonic systems, microwave technologies, antennas, laser monitoring systems, etc.
E-mail: microoil@mail.ru
Kazan National Research Technical University
named after A. N. Tupolew
Institute of Radio Electronic and Telecommunication
Karl Marks st 10
Kazan, Tatarstan
420111 Russian Federation

Timur Arturovich Agliullin is a bachelor student in the Radio Photonics and Microwave Technologies Department at the Kazan National Research Technical University named after A. N. Tupolew (KNRTU-KAI). His research interests includes: antennas, radio photonics.

E-mail: taagliullin@mail.ru
Kazan National Research Technical University
named after A. N. Tupolew
Institute of Radio Electronic and Telecommunication
Karl Marks st 10
Kazan, Tatarstan
420111 Russian Federation

Denis Andreevich Vedenkin – for biography, see this issue, p. 101.

Aydar Revkatovich Nasybullin – for biography, see this issue, p. 102.

Paper

Support Vector Machine based Decoding Algorithm for BCH Codes

V. Sudharsan and B. Yamuna

Department of Electronics and Communication Engineering, Amrita School of Engineering, Coimbatore, Amrita Vishwa Vidyapeetham, Amrita University, India

Abstract—Modern communication systems require robust, adaptable and high performance decoders for efficient data transmission. Support Vector Machine (SVM) is a margin based classification and regression technique. In this paper, decoding of Bose Chaudhuri Hocquenghem codes has been approached as a multi-class classification problem using SVM. In conventional decoding algorithms, the procedure for decoding is usually fixed irrespective of the SNR environment in which the transmission takes place, but SVM being a machine-learning algorithm is adaptable to the communication environment. Since the construction of SVM decoder model uses the training data set, application specific decoders can be designed by choosing the training size efficiently. With the soft margin width in SVM being controlled by an equation, which has been formulated as a quadratic programming problem, there are no local minima issues in SVM and is robust to outliers.

Keywords—BCH codes, Chase-2 algorithm, coding gain, kernel, multi-class classification, Soft Decision Decoding, Support Vector Machine.

1. Introduction

In communication systems, there is an increasing demand for reliable and efficient transmission of data. When data is transmitted over a noisy communication channel, errors are bound to occur. Error control coding techniques are used to detect and correct these errors. The two main types of error correcting codes are block and convolutional codes. Bose Chaudhuri Hocquenghem (BCH) codes are a type of cyclic error correcting block codes with applications in digital, space and satellite communications. Conventional Hard Decision Decoding (HDD) algorithms like Peterson-Gorenstein-Zierler algorithm and Berlekamp-Massey (BM) algorithm have a standard error correcting capability of $t = \left\lfloor \frac{d_{\min}}{2} \right\rfloor$ errors. Though these algorithms have a decent error correcting performance for BCH codes, much research has been on the Soft Decision Decoding (SDD) algorithms to increase the error correction capability. SDD algorithms make use of the channel statistic information, which associates a reliability value to each of the received bit and helps in estimating a more accurate codeword at the receiver.

In the past decade, there has been consistent work on the application of heuristic, evolutionary and artificial intelligence techniques to the decoding problem. These techniques were more robust and had a faster convergence rate. On similar lines, Kao and Berber used SVM, a maximum margin classification technique, for decoding convolutional codes [1]. The same has been extended to discuss the effect of channel and modulation techniques for basic error control coding schemes in wireless applications [2]. In this paper, a SVM based decoding algorithm has been proposed for BCH codes. The proposed algorithm can be used for any (n,k,d) BCH code. The decoding problem has been approached as a multi-class classification problem. The SVM decoder has been programmed and performance comparison has been established against conventional Chase-2 algorithm.

The paper is organized as follows: In Section 2 the existing decoding algorithms for BCH codes are reviewed. Section 3 gives an overview of Support Vector Machines. The proposed decoding algorithm for BCH codes has been explained systematically in Section 4. Section 5 discusses the simulation results of the proposed algorithm followed by conclusion in Section 6.

2. Decoding Algorithms for BCH Codes

BCH codes form a class of powerful error correcting cyclic codes constructed using finite fields. They are known for their multiple error-correcting capabilities and the ease of encoding and decoding [3]. Peterson, gave a decoding algorithm for binary BCH codes based on syndrome decoding. Based on his observation on the linear recurrence in BCH codes, he came up with a set of linear equations, solving which the error locations can be obtained [4]. This algorithm was further generalized to GF(p^m) by Gorenstein and Zierler [5]. Later, Chein devised a fast decoding algorithm for determining the roots of error locating polynomials over finite fields [6].

The Berlekamp-Massey-Forney algorithm is the most commonly used HDD algorithm for BCH codes. The Berlekamp-Massey algorithm is an alternative method to solve Peterson's linear equations to obtain the error location polynomial in a simplified manner [7], [8]. Forney proposed an algorithm for determining the roots of error correcting polynomial [9]. Chase, put forth a class of decoding algorithms that make use of the channel measurement information and claimed a two fold increase in error correcting capability over traditional HDD algorithms [10]. The Least Reliable Positions (LRPs) were identified based on the magnitude of each element in the received vector

and the decoded codeword was estimated from a candidate set of probable codewords generated using LRPs. This additional improvement in performance comes with an additional complexity. Reeve and Amarasinghe proposed a parallel Viterbi decoder for cyclic BCH codes since the usual algebraic decoding methods are not readily adaptable for soft decoding [11]. Yingquan formulated a list decoding algorithm for BCH codes to correct upto $1-\sqrt{1-D}$ errors based on Guruswami-Sudan algorithm [12]. A Reliability Level List based decoding algorithm for binary BCH codes – which uses the exact reliability values to arrive at the most probable codeword - has been proposed by Yamuna and Padmanabhan [13]. In the past decade, much effort has been put on the application of heuristic evolutionary algorithms like Genetic Algorithm (GA), Particle Swarm Optimization (PSO) and Neural Networks (NNs) to the decoding problem [14], [15]. They were more robust and had a faster decoding convergence. Azouaoui and Belkasmi applied the heuristic GA to BCH decoding to increase the robustness and efficiency [16]. These algorithms facilitate easier implementation of decoders for Software Defined Radio (SDR) applications, where adaptability is an important factor. To decrease the hardware complexity, an interpolation based one pass Chase decoder was proposed [17] and it was 2.2 times higher in hardware efficiency than Berlekamp in terms of throughput over area ratio. Torres *et al*. attempted a radial basis NN as error correction technique to decode BCH codes [18].

The different hard decision and soft decision schemes proposed in literature have different degrees of performance enhancement and complexity. Attempts on performance enhancement or complexity reduction, trading-off one for the other has been an open problem for researchers.

The procedure of traditional decoding algorithms has the same computational complexity even at a lower noise level. However, modern communication systems need adaptive decoders that cater to changes in channel characteristics. Given the dynamic requirements of emerging trends in channel decoding, in this paper – SVM – a multi-class classification technique has been applied to the decoding problem. The SVM model which is constructed according to the training data is channel adaptive and hence results in a much better performance than conventional methods.

3. SVMs for Data Classification

Support Vector Machines are a class of supervised learning algorithms based on Statistical Risk Minimization (SRM) principle. SVMs analyze the training data, recognize pattern and construct a model. The model is then used for the classification of unknown data. SVMs are generally used for classification and regression [19], [20]. Though SVM was traditionally used for binary classification problems, gradually it was used for multi-class classification problems as well. Cortes and Vapnik formulated a one against all SVM where a multi-class classification problem was converted into N binary classification problems, where N denotes the number of classes [21]. Krebel later came up with a pairwise one versus one approach, involving NC_2 binary classifiers and reduced the unclassifiable regions that occur in one versus all SVM [22]. Studies by Abe, Kao, Hsu and Lin [23]–[25] show that one versus one algorithm is best suited for multi-class classification problem. So this approach has been used in the proposed decoding algorithm.

In a binary classification problem, given a set of labeled, linearly separable training data that belong to two different classes, SVM finds an Optimal Separation Canonical Hyperplane (OSCH), i.e. to achieve the largest minimum distance that separates the data points of one class from the other class and constructs a decision function that defines the margin. Each classifier has a subset of training data – decision variables, x_i – called the support vectors, which are the data points that lie closer to the margin and they characterize the margin. Now, any unknown data can be classified to one of the two classes by evaluating the decision function for that unknown data. Each support vector (SV) has an associated coefficient vector w that defines its role in the classifier. In order to obtain an optimal classifier with minimum number of misclassified data, there is necessity to have maximum margin. Here $\frac{2}{||w||}$ is taken to be the classifier margin.

When the training data are linearly inseparable, we go for a soft margin SVM. To allow inseparability and compensate for the misclassifications, i.e. to accommodate the data that do not have the maximum margin, the non-negative slack variable (ξ) is introduced. Thus for a maximum margin classifier, the SV parameters should be minimized. This has been formulated as a quadratic programming function in SVM. To determine the optimal SV parameters namely coefficient vector w and bias term b, we need to minimize Eq. (1) given below:

$$\frac{1}{2}w^T w + C\sum_{i=1}^{N}\xi_i, \tag{1}$$

where: w – coefficient vector, C – margin parameter, ξ – slack variable, with respect to the constraint in Eq. (2):

$$y_i(w^T\phi(x_i)+b) \geq 1-\xi_i \quad \text{and} \quad \xi_i \geq 0, \quad i=1,2,\ldots,N, \tag{2}$$

where: $\phi(x)$ – non-linear kernel function, b – bias term, y_i – class label.

In order to maximize the generalization ability and to enhance the classification of non-linear data, the input training data is mapped into a higher dimensional space called feature space using a kernel function. This is called kernel trick. Since the application of SVM to decoding problem comes under the non-linear category, Radial Bias Function (RBF) kernel as given in Eq. (3) has been incorporated to map the input training data into higher dimensional space. Further RBF kernel, which uses Euclidean distance prevents the effect of outliers in performance.

$$K(x_i,x_j) = \mathrm{e}^{-\gamma\left\|x_i-x_j\right\|^2}, \quad \gamma \geq 0. \tag{3}$$

4. SVM Based Decoding Algorithm for BCH Codes

Consider a BCH (n,k,d) code, consisting of 2^k message words where k denotes the number of bits in each message word, n denotes the codeword length and d denotes the minimum distance between codewords. With each message word considered as a class, there are $N = 2^k$ classes. These message words are encoded into a codeword of length n. Each bit in the codeword is a feature that defines the class to which the received codeword belongs. Each codeword in the $N = 2^k$ codeword set has a one to one correspondence to a unique message word and the codeword is associated with a class label y_i, where $1 \leq i \leq N$.

SVM based decoding involves two major phases: the training phase and the decoding phase. The decoder model trained and constructed in the training phase is used to classify the received sequence in the decoding phase.

4.1. Training Phase

In training phase, an appropriate model has to be constructed by generating suitable training data. Each codeword of class i is modulated, repeated M number of times and then corrupted by an Additive White Gaussian Noise (AWGN) of SNR, 0 dB.

Now, we have $N \times M$ number of codewords belonging to N different classes, which form the training data for the model. The training is done at a high level of noise at 0 dB, to represent the worst-case scenario and to maximize the generalization characteristic of the decoder. These training data are sent to the pairwise one versus one SVM decoder, where each class of data is compared with another class and NC_2 classifiers (decision functions) are constructed and support vectors (decision variables) which is usually a subset of the training data are obtained.

To develop an optimal model, optimal training parameters should be selected namely, the margin parameter C and kernel parameter γ. This is done using a v-fold cross-validation method. In a v-fold cross validation, the training data is divided into v equal sized subsets. The model is constructed using v-1 subsets as training data and tested with the one remaining set. For each combination of (C, γ), this process is repeated v times. The contour plots of a 10-fold cross validation for BCH $(15,7,5)$ code are shown in Fig. 1. The value of (C, γ) with highest cross validation accuracy is taken as the optimal training parameter. Thus at the end of training phase, we have an optimal decoder model with m (where $m < N \times M$) support vectors.

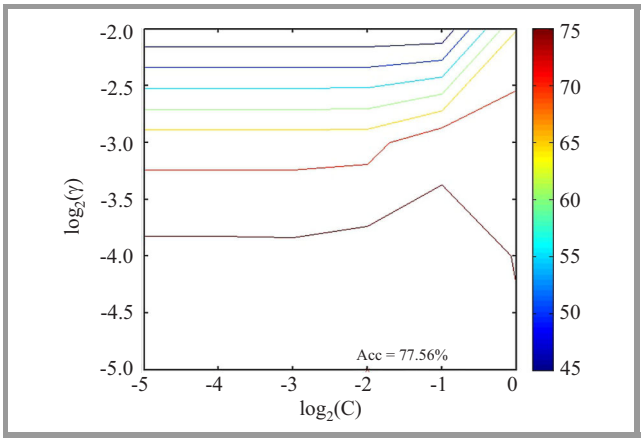


Fig. 1. A 10-fold cross validation done using LIBSVM. (See color pictures online at www.nit.eu/publications/journal-jtit)

4.2. Decoding Phase

In decoding phase, each of the n bit received soft decision sequence is the unknown data that has to be classified into one of the N different classes, i.e. valid codewords. The decoding phase is thus a simple multi-class classification problem and each classifier is a binary classifier. The noisy received codeword is passed through NC_2 classifiers, where each classifier has a set of support vectors generated during the training phase. The received codeword is transferred into the higher dimensional space using the same RBF kernel function and evaluated using the decision function constructed during the training phase. The output of the decision function determines the class to which the received codeword belongs. This is repeated for all the NC_2 classifiers.

Now, each classifier would have given a vote to one of the N different classes. The received codeword gets decoded to the class which gets maximum number of votes. The output here refers to the maximum value of the decision function, which is directly related to the soft value associated with each received bit. This is known as winner – takes – all (WTA) principle [22]. Since there is a one to one correspondence between the codeword and the message word, the message word can be directly estimated by observing the class value. The proposed SVM based decoding algorithm is given in six steps in Algorithm 1.

Algorithm 1: SVM decoding proposal

1: For a (n, k, d) binary BCH code, each message word in the 2^k set is associated with a class label $y_i (N = 2^k)$.
2: Each message word is encoded into a n-bit codeword to obtain N unique codewords.
3: Each codeword is then transmitted M times through an AWGN channel with SNR= 0 dB.
4: These $N \times M$ codewords along with their associated class label form the training data. The training data set of one class is compared against training data of another class and hence NC_2 classifiers are constructed.
5: Each classifier has an associated set of Support Vectors (decision variables). Thus, the SVM model is constructed.
6: The unknown codeword is now passed through the decoder model and based on the WTA principle, it gets classified to one of the N classes and the corresponding message is obtained.

5. Simulation Results and Discussions

LIBSVM, a software for multi-class SVM classification and regression, has been used for the construction of SVM model and testing of received codeword [26]. The AWGN channel has been considered and Binary Phase Shift Keying (BPSK) is used for modulation. All simulations have been performed using Matlab.

The proposed SVM decoding algorithm has been applied to BCH (15,7,5) code and the performance of SVM based decoding algorithm has been compared against Chase-2 and HDD algorithm as shown in Fig. 2. At a BER of 10^{-3}, the SVM decoder is found to have a coding gain of 0.8 dB over Chase-2 algorithm and a coding gain of 2 dB over HDD algorithm.

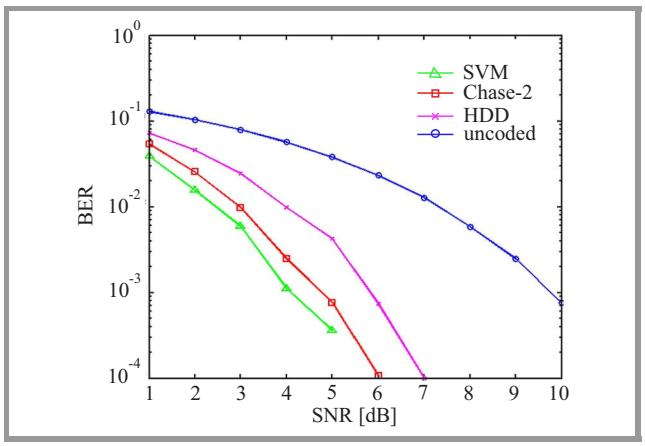


Fig. 2. BER versus SNR plot of BCH (15,7,5) code.

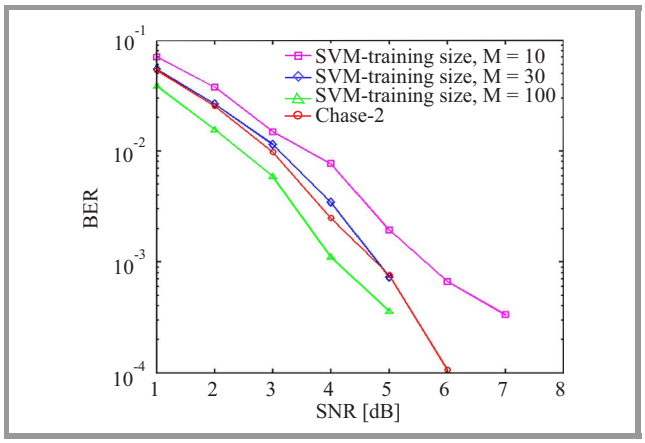


Fig. 3. Performance of SVM decoder for BCH (15,7,5) code at different training size.

Figure 3 shows that the performance of the SVM decoder improves when the training data size is increased. However, the increase in training size in turn increases the number of SVs. This complexity due to increase in SVs can be compensated by puncturing the classifiers, which consistently misclassifies the test data set during cross-validation. Though the increase in training size improves the performance, due to over fitting of data, improvement saturates as shown in Fig. 4.

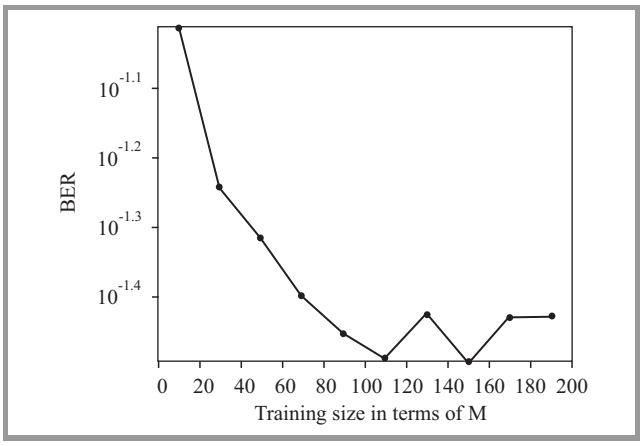


Fig. 4. BER of SVM decoder under different training size with SNR fixed at 1 dB.

Unlike in soft decision decoding algorithms like Chase-2 decoder, SVM based decoder does not involve hard decision error correction, thus eliminating HDD complexity completely. However, when the value of N increases, more classifiers have to be constructed and this results in additional complexity at the testing phase. This can be overcome by cascading SVM decoders. For a N-class problem, initially one decoder can be modeled to classify them into two classes and then two more decoders can be modeled to further classify them into $\frac{N}{2}$ sub-classes, thus reducing the complexity at each decoder. Thus, the proposed SVM algorithm can be combined with the cascading technique and applied to higher block length codes. The additional complexity due to this process is negligible because training is done only once during the initial setup of the communication system. The complexity at the decoding stage depends directly on the number of support vectors generated, which can be controlled according to the application thus striking a trade-off between complexity and performance.

6. Conclusion

This paper presents a SVM based decoding technique for BCH codes, where the decoding problem has been approached as a multi-class classification problem. This algorithm makes maximum use of the channel measurement information combined with the margin based classification feature of the SVM to give an optimal decoder estimate. From the simulation results, it can be seen that the proposed decoding algorithm has a better performance than the conventional Chase-2 algorithm at higher training size. The technique can be applied to higher block length codes by using cascaded SVM. A more generalized decision model, convergence to global optimal solution and prevention of outliers are major leads in this algorithm and thus proves to be efficient for the decoding of BCH codes. The proposed SVM based decoding algorithm can be extended to decoding of high performance robust turbo codes as well.

References

[1] J. Kao and S. Berber, "Error control coding based on support vector machine", in *Proc. 1st IAPR Worksh. Cogn. Inform. Process.*, Santorini, Greece, 2008, pp. 182–187.

[2] R. Ramanathan, N. Valliappan, S. Pon Mathavan, M. Gayathri, R. Priya, and K. Soman, "Generalised and channel independent SVM based robust decoders for wireless applications", in *Proc. IEEE Int. Conf. Adv. Recent Technol. in Commun. Comput. ARTCom'09*, Kottayam, Kerala, Indie, 2009, pp. 756–760.

[3] R. Bose and D. Ray-Chaudhuri, "On a class of error correcting binary group codes", *Inf. Control.*, vol. 3, no. 1, pp. 68–79, 1960.

[4] W. Peterson, "Encoding and error-correction procedures for the Bose-Chaudhuri codes", *IEEE. Trans. Inform. Theory*, vol. 6, no. 5, pp. 459–470, 1960.

[5] D. Gorenstein and N. Zierler, "A class of error-correcting codes in p^m symbols", *J. Soc. Ind. Appl. Math.*, vol. 9, no. 2, pp. 207–214, 1961.

[6] R. Chien, "Cyclic decoding procedures for Bose-Chaudhuri-Hocquenghem codes", *IEEE Trans. Inform. Theory*, vol. 10, no. 4, pp. 357–363, 1964.

[7] E. Berlekamp, "On decoding binary Bose-Chadhuri-Hocquenghem codes", *IEEE Trans. Inform. Theory*, vol. 11, no. 4, pp. 577–579, 1965.

[8] J. Massey, "Step-by-step decoding of the Bose-Chaudhuri-Hocquenghem codes", *IEEE Trans. Inform. Theory*, vol. 11, no. 4, pp. 580–585, 1965.

[9] G. Forney, "On decoding BCH codes", *IEEE Trans. Inform. Theory*, vol. 11, no. 4, pp. 549-557, 1965.

[10] D. Chase, "A Class of algorithms for decoding block codes with channel measurement information", *IEEE Trans. Inform. Theory*, vol. 18, no. 1, pp. 170–182, 1972.

[11] J. Reeve and K. Amarasinghe, "A parallel Viterbi decoder for block cyclic and convolution codes", *Signal Process.*, vol. 86, no. 2, pp. 273–278, 2006.

[12] Y. Wu, "New List Decoding Algorithms for Reed-Solomon and BCH Codes", *IEEE Trans. Inform. Theory*, vol. 54, no. 8, pp. 3611–3630, 2008.

[13] B. Yamuna and T. R. Padmanabhan, "A reliability level list based SDD algorithm for binary cyclic block codes", *Int. J. Comput. Commun. Control*, vol. 7, no. 2, pp. 388–395, 2012.

[14] J. Yuan, L. Wang, Q. He, H. Li, and Y. Wang, "A novel genetic probability decoding (GPD) algorithm for the FEC code in optical communications", *Int. J. Light Elec. Opt.*, vol. 124, no. 15, pp. 1986–1989, 2013.

[15] J. Yuan, C. He, W. Gao, J. Lin, and Y. Pang, "A novel hard decision decoding scheme based on genetic algorithm and neural network", *Int. J. Light Electron Opt.*, vol. 125, no. 14, pp. 3457–3461, 2014.

[16] A. Azouaoui and M. Belkasmi, "A soft decoding of linear block codes by genetic algorithms", in *Proc. Int. Conf. Intell. Comput. Syst. ICICS'2012*, Dubai, United Arab Emirates, 2012.

[17] X. Zhang, "An efficient interpolation-based chase BCH decoder", *IEEE Trans. Circuits Syst. II: Express Briefs*, vol. 60, no. 4, pp. 212–216, 2013.

[18] H. Torres, M. Jamett, C. Urrea, and J. Kern, "Design of a fault tolerant digital communication system, by means of RBF networks. Comparison simulations with the encoding and decoding algorithms BCH (7,4,1)", *IEEE Latin America Trans.*, vol. 12, no. 8, pp. 1365–1374, 2014.

[19] J. Gokulachandran and K. Mohandas, "Comparitive study of two soft computing techniques for the prediction of remaining useful life of cutting tools", *Int. J. Intell. Manuf.*, vol. 26, no. 2, pp. 255–268, 2013.

[20] J. Gokulachandran and K. Mohandas, "Prediction of cutting tool life based on Taguchi approach with fuzzy logic and support vector regression techniques", *Int. J. Qual. Reliab. Manage.*, vol. 32, no. 3, pp. 270–290, 2015.

[21] C. Cortes and V. Vapnik, "Support-vector networks", *Machine Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[22] U. Krebel, "Pairwise classification and Support Vector Machines", in *Advances in Kernel Methods: Support Vector Learning*, B. Schölkopf, C. J. C. Burges, and A. J. Smola, Eds. Cambridge, USA: MIT Press, 1999, pp. 255–270.

[23] S. Abe, *Support Vector Machines for Pattern Classification*. Springer, 2005.

[24] J. Kao, "Methods of artificial intelligence for error control coding and multi-user detection", Ph.D. Thesis, The University of Auckland, New Zealand, 2010.

[25] C. W. Hsu and C. J. Lin, "A comparison of methods for multiclass support vector machines", *IEEE Trans. Neural Netw.*, vol. 13, no. 2, pp. 415–425, 2002.

[26] C.-C. Chang and C.J. Lin, "LIBSVM: A library for support vector machines", *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 2, pp. 27:1–27:27, 2011. Software available at http://www.csie.ntu.edu.tw/∼cjlin/libsvm

V. Sudharsan is pursuing his senior year of undergraduate B.Tech. program at the Department of Electronics and Communication Engineering, Amrita School of Engineering, Amrita Vishwa Vidyapeetham. His research interests include error control coding, networking, soft computation and artificial intelligence.

E-mail: sud6har2san@gmail.com
Dept. of Electronics and Communication Engineering
Amrita School of Engineering, Coimbatore
Amrita Vishwa Vidyapeetham
Amrita University
India

B. Yamuna received her M.E. in VLSI design and Ph.D. in Error Control coding from Amrita School of Engineering, Amrita Vishwa Vidyapeetham, is working as an Associate Professor in the Dept. of ECE at Amrita School of Engineering. She has published and presented papers in international journals and conferences. She has reviewed papers for international and national conferences, and chaired sessions. Her current research interests include coding for reliable communication, spread spectrum communication, VLSI architectures for communication systems, genomic coding and cryptosystems based on error control codes. She is a member of ISTE and IETE.

E-mail: b_yamuna@cb.amrita.edu
Dept. of Electronics and Communication Engineering
Amrita School of Engineering, Coimbatore
Amrita Vishwa Vidyapeetham
Amrita University
India

Paper

An Efficient Early Iteration Termination for Turbo Decoder

P. Salija and B. Yamuna

Department of Electronics and Communication Engineering, Amrita School of Engineering, Coimbatore, Amrita Vishwa Vidyapeetham, Amrita University, India

Abstract—Turbo code finds wide applications in mobile communication, deep space communication, satellite communication and short-range communication despite its high computational complexity and iterative nature. Realizing capacity approaching turbo code is a great achievement in the field of communication systems due to its efficient error correction capability. The high computational complexity associated with the iterative process of decoding turbo code consumes large power, introducing decoding delay, and reducing the throughput. Hence, efficient iteration control techniques are required to make the turbo code more power efficient. In this paper, a simple and efficient early iteration termination technique is introduced based on absolute value of the mean of extrinsic information at the component decoders of turbo code. The simulation results presented clearly show that the proposed method is capable of reducing the average number of iterations while maintaining performance close to that of fixed iteration termination. The significant reduction in iteration achieved by the method reduces decoding delay and complexity while maintaining Bit Error Rate performance close to standard fixed iteration turbo decoder.

Keywords—early termination, complexity reduction, mean of extrinsic information, turbo decoder.

1. Introduction

One of the major challenges in wireless communication is to establish energy efficient and performance optimized error correction over dynamic channel conditions and to prolong the network/hardware lifetime. The near Shannon's performance and excellent Bit Error Rate (BER) performance of turbo code has resulted in its wide use in the entire range of wireless communication regardless of its high computational complexity [1], [2]. Key to the outstanding performance of turbo code is the parallel concatenation, interleaved recursive systematic operation, and iterative exchange of extrinsic information [3], [4]. Turbo decoding being an iterative process, its decoding delay, computational effort and energy consumption increase linearly with iterations [5]. The number of iterations required to obtain an acceptable performance depends on the channel characteristics [6]. The fact that sometimes an acceptable performance is not achieved even with infinite number of iterations has motivated researchers to find low complexity performance optimized turbo decoding techniques.

The objective of the early termination is to reduce unnecessary computation complexity, power consumption and decoding delay. If the channel conditions are bad, performance improvement is negligible even after infinite iterations whereas under good channel conditions desired performance may be reached after just a few iterations. Therefore, finding the appropriate condition to terminate the turbo decoding iteration while maintaining a tradeoff between acceptable performance and complexity is a major challenge [7]. Early iteration termination techniques based on soft and hard information have been reported. Sign Change Ratio (SCR) [8], an early termination technique compares the sign differences between two consecutive iterations. Mean Estimate (ME) [9] technique is based on mean of the absolute values of Log Likelihood Ratios (LLRs). In Mean Reliability [10], [11] based stopping technique, the iteration is stopped when the means of the absolute value of extrinsic information at two consecutive iterations are same. Hard Decision Aided (HDA) [8] stopping technique based on the hard values at the output of component decoder, stops iteration when the hard decision at two consecutive iterations is the same. Sign Difference Ratio (SDR) [12], [13], uses the number of sign differences between a priori and a posteriori information at output of component decoder 2.

In this paper, an iteration termination technique for turbo decoding at low and high SNR conditions is proposed. This technique is based on monitoring absolute value of mean of the extrinsic information over a frame at the output of two component decoders.

Paper organization is as follows: Section 2 summaries turbo code principles and iterative turbo decoding relevant to the proposed technique. Section 3 deals with analysis and challenges of iteration control in turbo code. Mean based new stopping technique is presented in Section 4. Section 5 shows the simulation results followed by conclusion in Section 6.

2. Turbo Code Principles and Iterative Decoding

Practical codes failed to reach Shannon's capacity limit until the discovery of turbo code in 1990s by Claude

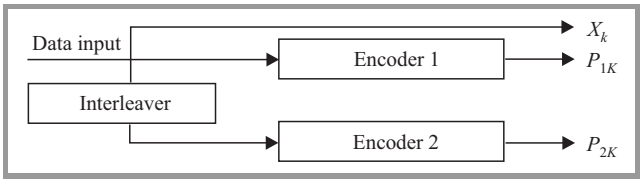


Fig. 1. Generic turbo encoder.

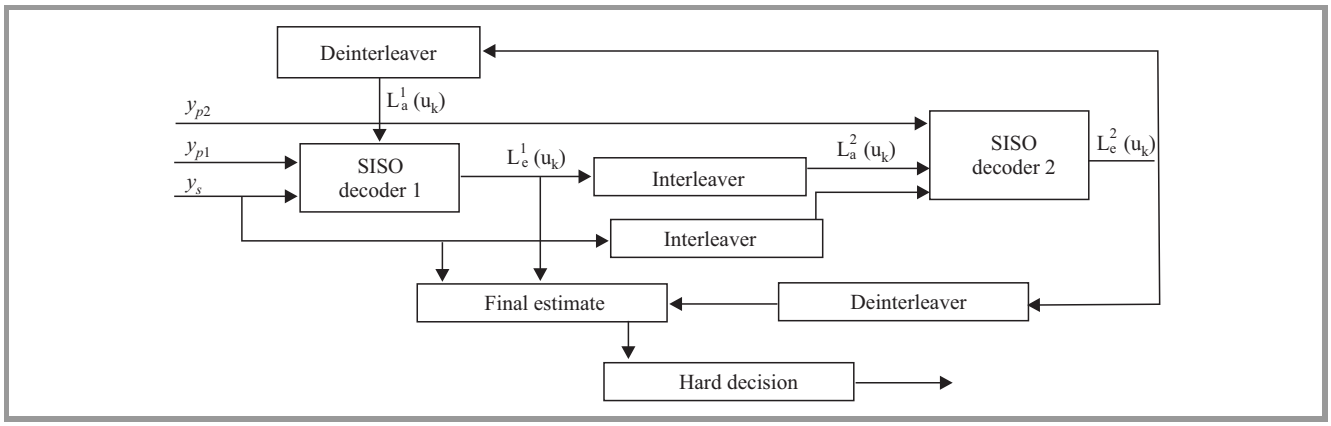


Fig. 2. Turbo decoder.

Berrou *et al*. [14], [15]. Due to its excellent error correction capability, turbo code finds wide applications in mobile communication, short-range communication, deep space communication, and wireless networks [3], [16], [17]. Turbo encoder consists of two Recursive Systematic Convolutional (RSC) encoders separated by an interleaver [12] as shown in Fig. 1. X_k, P_{1k} and P_{2k} are the systematic output, parity check bits 1 of the encoder 1 output, and parity check bits 2 of the encoder 2 output respectively.

RSC encoder 1 has two outputs namely systematic output and parity output. RSC encoder 2 consists of only parity outputs. The presence of interleaver increases codeword weight. If the RSC encoder 1 produces low weight codeword then the probability of producing low weight at the output of RSC encoder 2 is low [18]. The concatenated outcome from the turbo encoder would be a strong one. The data rate can be changed by applying puncturing technique at the output of the encoder.

Turbo decoding takes place by iterative exchange of extrinsic information [19]. The heart of the turbo decoder is the SISO decoder. Maximum A Posteriori (MAP) algorithm is the most commonly used, which outperforms other SISO algorithms under low SNR conditions [18]. The effectiveness of performance depends on the iteration of MAP algorithm on each received code. MAP algorithm calculates A Posteriori Probability (APP) values for each information bit. However, the computational complexity associated with the MAP algorithm is extremely high [20]. Log MAP and Max Log MAP algorithms are two variants of MAP algorithm with reduced computational complexity [21].

The basic iterative turbo decoding architecture is shown in Fig. 2. The input to the component decoders: SISO decoder 1 and SISO decoder 2 are the received systematic information, parity received information and a priori information. Based on the information available at the input of component decoders, the extrinsic a posteriori L values are generated. In the first iteration the input to the SISO decoder 1 – the received systematic and parity information-generates extrinsic a posteriori information $L_e^1(u_k)$ for the received sequence. Extrinsic information generated by the first decoder serves as the a priori information to the second decoder. A priori information, $L_a^1(u_k)$ is zero for the first iteration of SISO decoder 1.

The inputs to the SISO decoder 2 are the received parity information, interleaved systematic bits and a priori information. Decoder 2 generates the extrinsic a posteriori L values, $L_e^2(u_k)$ corresponding to the inputs received. In the subsequent iterations, SISO decoder 1 receives the a priori information corresponding to the de-interleaved extrinsic a posteriori L information from decoder 2. Log-MAP algorithm has been considered in this work for implementing turbo codes, since it is less complex than MAP algorithm and has better performance than Max Log MAP algorithm [20]. Turbo decoder calculates Log Likelihood Ratios (LLRs) for each data bit and after a certain number of iterations, hard decisions are estimated based on the LLR values. The process for calculating LLR values is as follows [21]. The branch metric values are calculated from the received information and extrinsic information of the previous component decoder according to Eq. (1). Let $u = [u_1, u_2, \ldots, u_n]$ be the data sequence, which is encoded and transmitted through a channel. The corresponding received data sequence is $y = [y_1, y_2, \ldots, y_n]$. Then the branch metric values are given by:

$$\gamma_k(s', s) = \frac{u_k(L_a^i(u_k))}{2} + \frac{L_c \sum_{i=1}^{n} y_{kl} u_{kl}}{2}, \tag{1}$$

where L_c is the channel reliability factor, s' and s represent the previous state and present state respectively. The forward metric α and backward metric β are calculated as follows:

$$\alpha_{k+1}(s) = \ln \sum e^{\gamma_k(s', s) + \alpha_k(s')}, \tag{2}$$

$$\beta_k(s') = \ln \sum e^{\gamma_k(s', s) + \beta_{k+1}(s)}. \tag{3}$$

Extrinsic information at the output of component decoders can be calculated as in Eq. (4),

$$L_e^i(u_k) = \ln \left(\frac{\sum_{(s', s) \in \Sigma_k^+} e^{\frac{L_c}{2} y_{pk} p_{lk} + \alpha_k(s') + \beta_{k+1}(s)}}{\sum_{(s', s) \in \Sigma_k^-} e^{\frac{L_c}{2} y_{pk} p_{lk} + \alpha_k(s') + \beta_{k+1}(s)}} \right). \tag{4}$$

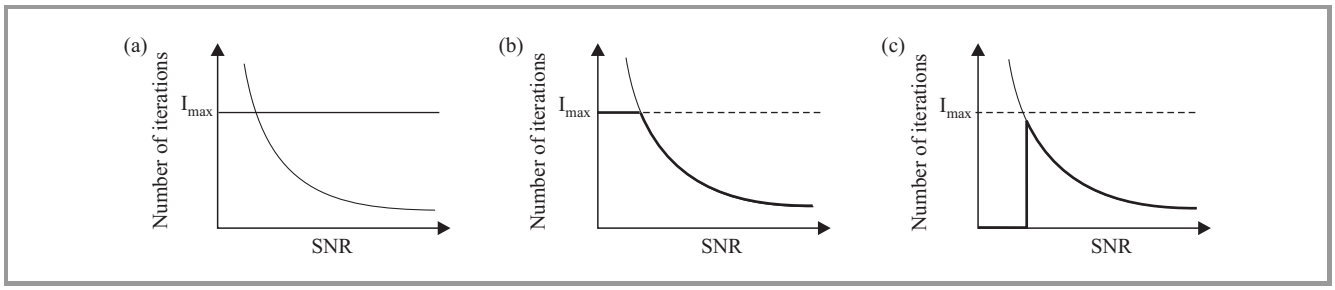


Fig. 3. Average number of iteration required with: (a) fixed iteration termination, (b) termination at high SNR, (c) terminating at low and high SNR.

The LLR at the output of decoder can be computed from the extrinsic information of the component decoders and the received systematic information. LLR values $L_1^i(u_k)$ and $L_2^i(u_k)$ from SISO decoder 1 and decoder 2 can be calculated as in Eqs. (5) and (6) respectively:

$$L_1^i(u_k) = L_{e2}^{i-1}(u_k) + L_{e1}^i(u_k) + L_c y_{uk}\,, \tag{5}$$

$$L_2^i(u_k) = L_{e1}^i(u_k) + L_{e2}^i(u_k) + L_c y_{uk}\,. \tag{6}$$

Each component decoder generates extrinsic information for the next decoder until the maximum number of iteration is reached. After completing the final iteration, hard decisions are made from the output of decoder 2 after de-interleaving.

3. Iteration Control in Turbo Code – Analysis and Challenges

Turbo code achieves excellent performance at the cost of high computational complexity and power. Each iteration consumes a large amount of energy for computations and memory requirements. Computational complexity in the circuit is directly related to the power consumption [22] and functionality is limited by the available power in resource-constrained networks. Computational complexity of the turbo code can be reduced by reducing the number of iterations. However, reducing the number of iterations affects the performance of the turbo code resulting in a tradeoff between performance and complexity [6]. Depending upon the channel conditions, the performance and the required number of iterations also change [23]. The average number of iterations required for successful decoding versus SNR is shown in the Fig. 3 for different iteration termination schemes. Figure 3a shows the fixed iteration scheme, which uses maximum number of iterations at any SNR. Figure 3b shows the iteration termination at high SNR conditions. Here only a few iterations are required to obtain successful decoding at high SNR conditions. Figure 3c indicates the optimized termination technique that considers termination at low and high SNR conditions.

Figure 4 shows the performance of turbo decoding with Log Map algorithm over Additive White Gaussian Noise (AWGN) channel with Binary Phase Shift Keying (BPSK) modulation. From the figure, it is seen that though turbo code performance improves with iteration, at low SNR conditions the improvement is not significant even after the 15-th iteration. It is also observed that at high SNR conditions there is no significant improvement in performance after few iterations.

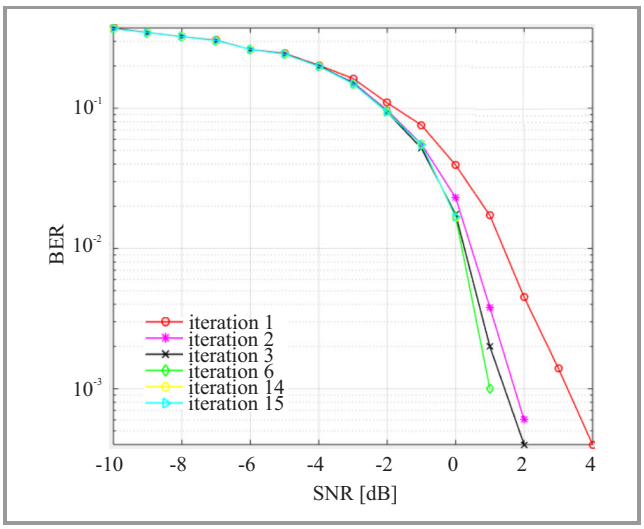


Fig. 4. SNR versus BER performance of turbo code. (See color pictures online at www.nit.eu/publications/journal-jtit)

The challenge is to determine the proper time when further iteration provides no or little improvement in performance. The technique used for this is called the early termination technique. While most of the cases require a few iteration to achieve acceptable performance, undesirable channel conditions require more iteration to achieve an acceptable performance [24]. Sometimes it is observed that although the errors are reduced to acceptable level after a few iterations, errors appear to increase again after some additional iteration. This is because of the numerical errors accumulating in the algorithm with iterations [9]. Figure 5 shows the performance of turbo decoder that accumulates errors with iterations.

From these observations, it is clear that the best performance and computation complexity reduction can be obtained if the iteration is stopped at the proper time. The objective of the early termination technique is to stop the iterations to avoid additional computations that contribute little or no improvement.

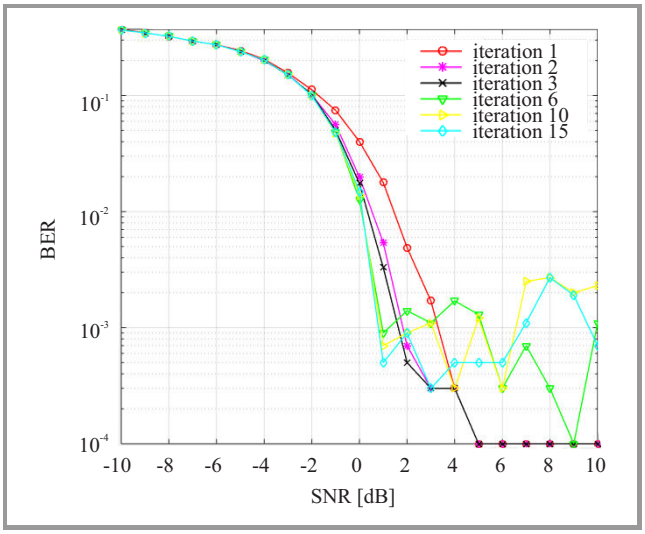


Fig. 5. Turbo decoding performance with accumulation of errors.

Several methods have been introduced by researchers to stop the termination early by considering some objectives like maintaining performance close to that of fixed iteration technique, and reducing power consumption and computational complexity [25]. Efficient iteration control has many advantages such as [6]:

- decoding more number of data blocks and hence improving the throughput,
- reducing the decoding delay,
- reducing the computational complexity,
- improving the performance by reducing accumulation of errors,
- reducing power/energy consumption.

Real time applications and multimedia transmission require systems with low latency and reduced complexity without performance degradation. In resource constrained networks functionality is limited by the stringency in available power [26]. Turbo decoder consumes large part of energy due to the computations associated with the iterations.

Stopping rules are mainly based on hard decision, soft decision, and extra checking categories [22], [27]. In the Cross Entropy (CE) based termination technique, the CE between log likelihood ratios at the output of component decoders at each iteration is calculated [28], [29]. Iteration is terminated if the CE $T(i)$ satisfies the condition: $T(i) < (10^{-2} - 10^{-4})T(1)$. SCR [8] approach is based on the number of sign changes in the extrinsic information between two consecutive iterations. Here the iteration is terminated if the number of sign changes $C(i)$ satisfies the condition: $C(i) \leq (0.05 - 0.3)N$. In SDR [12], [13] technique the iteration is terminated if the number of sign differences $D(i)$ between a priori and a posteriori information satisfies the condition: $D(i) \leq (0.001 - 0.01)N$. ME [9] technique is based on calculating the mean of absolute values of LLRs and stopping the iteration when the mean value is greater than a predefined threshold. Mean reliability based stopping rule has been proposed in literature [10], [11]. In this approach if the mean of absolute values of log likelihood ratios does not change between two consecutive iterations, then the process is terminated. Early iteration termination of turbo code based on histogram method uses LLR metric [20] where iteration is terminated if the central bin of the histogram is zero. In HDA [8] based stopping technique which uses the hard decision from the LLR values, the iteration is terminated if the hard decision values between two consecutive iterations are same.

Many channel conditions require only a few iterations to decide on the decoded bits, but few channel conditions require more iteration. CE, SDR, ME and histogram techniques are applicable to terminate iteration at high SNR conditions whereas SCR and HDA methods are applicable at both low and high SNR and require minimum of two iterations to make a decision even at high SNR. In SDR and CE techniques the data from both the component decoders is required instantaneously to make a decision, so the half iteration technique is not possible in such cases. In mean reliability technique, mean of the absolute values of LLR between iterations is equal at low SNR conditions. Hence, it is applicable to terminate the iteration at low SNR conditions. Techniques which reduce computations as much as possible while maintaining the acceptable performance, capable of saving half iteration computations and stopping the iteration at low and high SNR conditions at proper time still remain a challenge.

4. Mean Based New Stopping Criterion

The excellent performance of turbo code is due to the iterative exchange of extrinsic information. The magnitude of extrinsic information which depends on the channel characteristics (SNR) influences the calculation of branch metrics, forward metrics, backward metrics and LLR values as given in Eqs. (1)–(6). The final output decision is influenced to a very large extent by the extrinsic information. During the processing at the component decoders, the a priori information changes between iterations while systematic and parity inputs remain the same. Based on the mean value of extrinsic information at the output of component decoders an early iteration technique is proposed in this paper. This technique continuously monitors the extrinsic information at the output of component decoder. After each iteration, mean of extrinsic information is calculated. If the mean value is greater than a predefined threshold or if the difference in mean value between iterations lies within a threshold, the iterative process is terminated. The fact that the magnitude of extrinsic information varies with SNR and iteration number is shown in Figs. 6 to 9 as variations of the extrinsic information for low and high SNR at first iteration and fifteenth iteration. The variations are shown for frame size of 10,000 at first and fifteenth iteration for SNR of –5 dB and 5 dB at the output of the two component decoders.

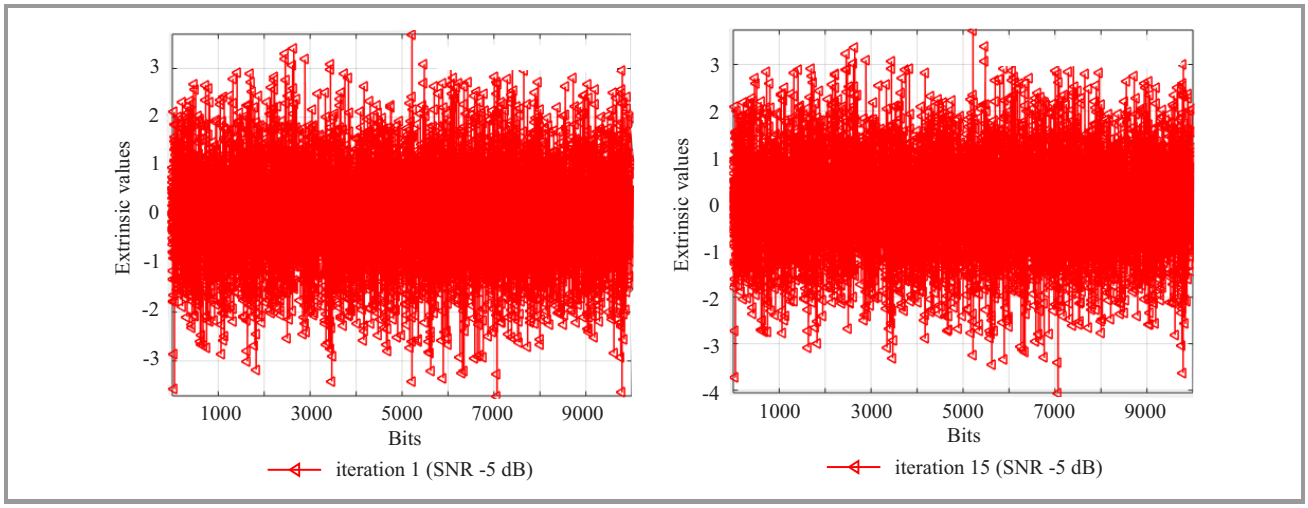


Fig. 6. Extrinsic values at the output of component decoder 1 at –5 dB.

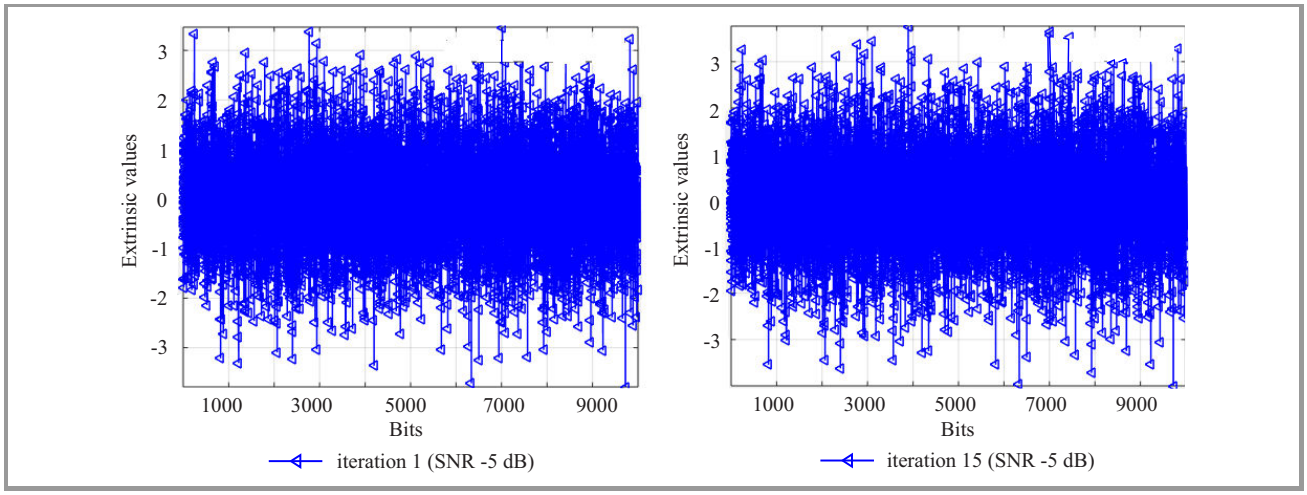


Fig. 7. Extrinsic values at the output of component decoder 2 at –5 dB.

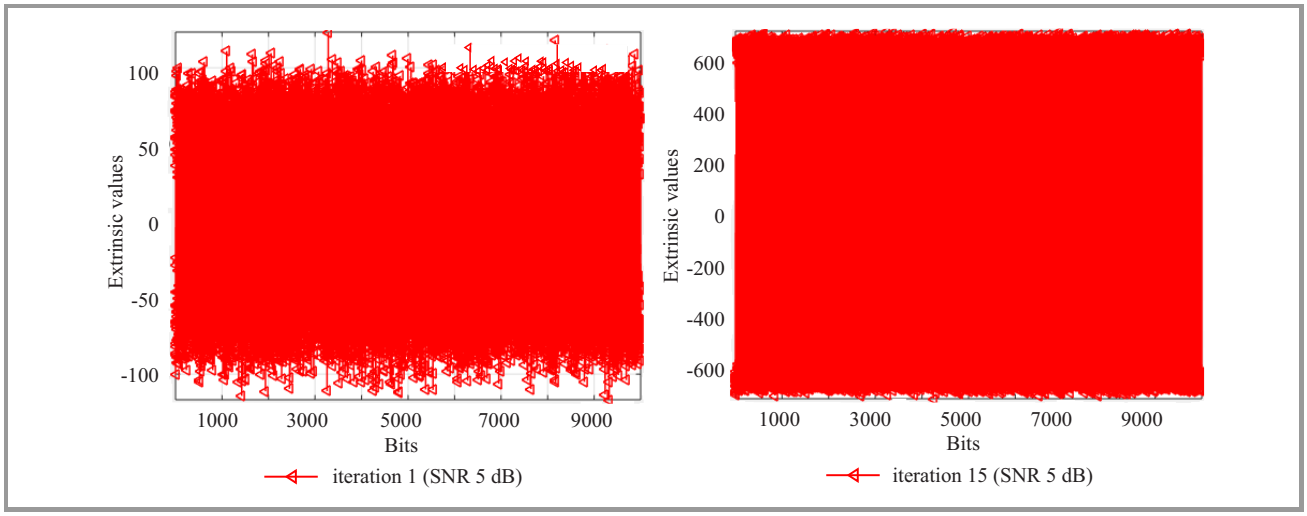


Fig. 8. Extrinsic values at the output of component decoder 1 at 5 dB.

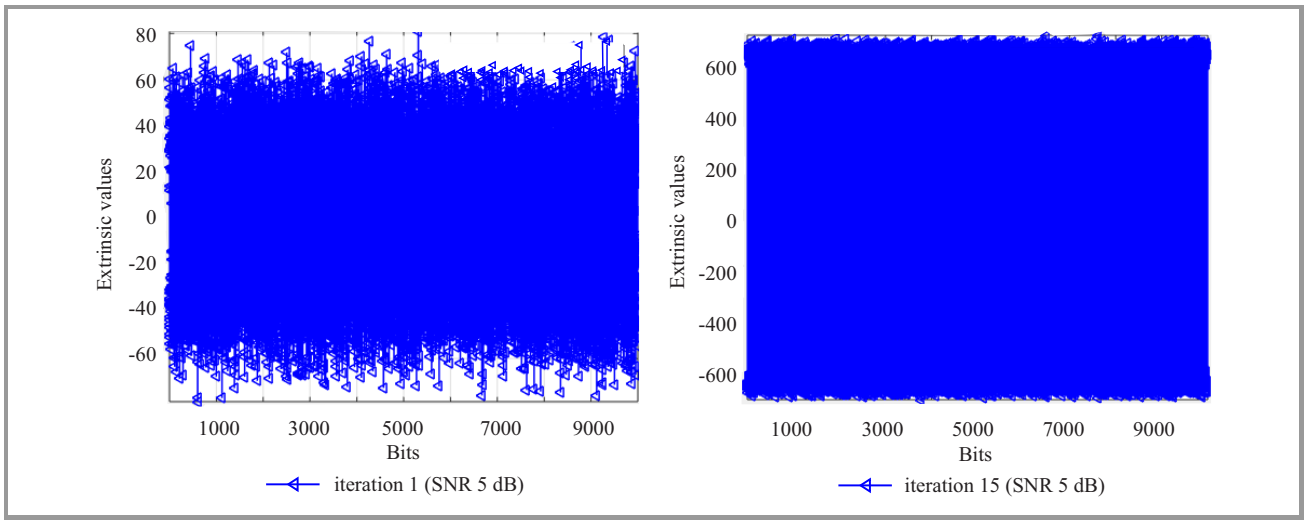


Fig. 9. Extrinsic values at the output of component decoder 2 at 5 dB.

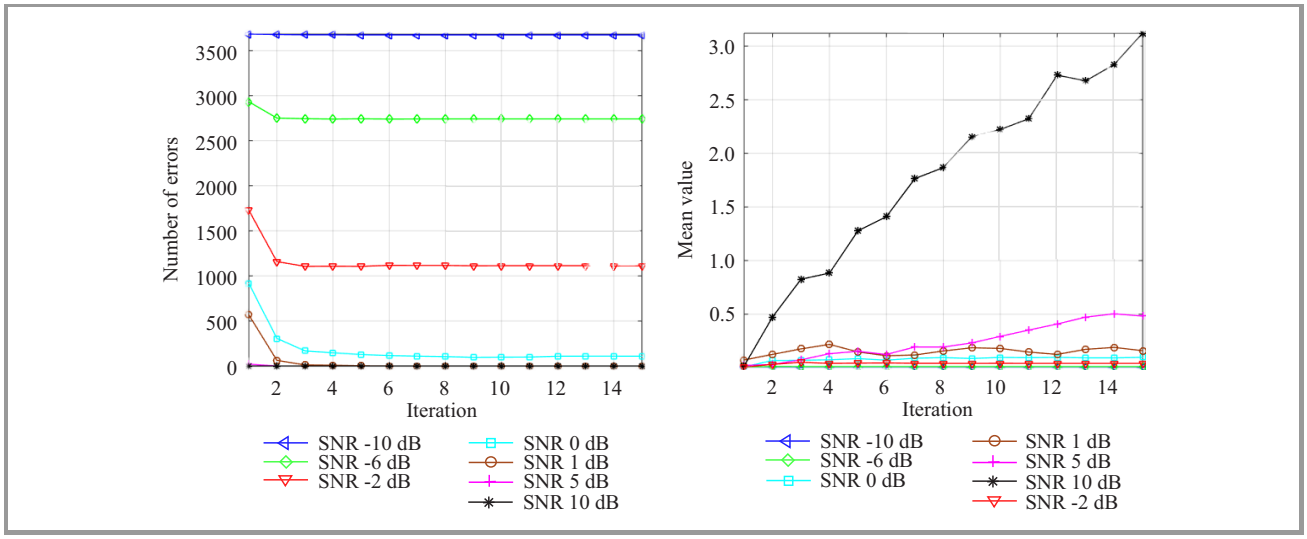


Fig. 10. Analysis of number of errors and absolute value of mean varying with iterations at decoder 1.

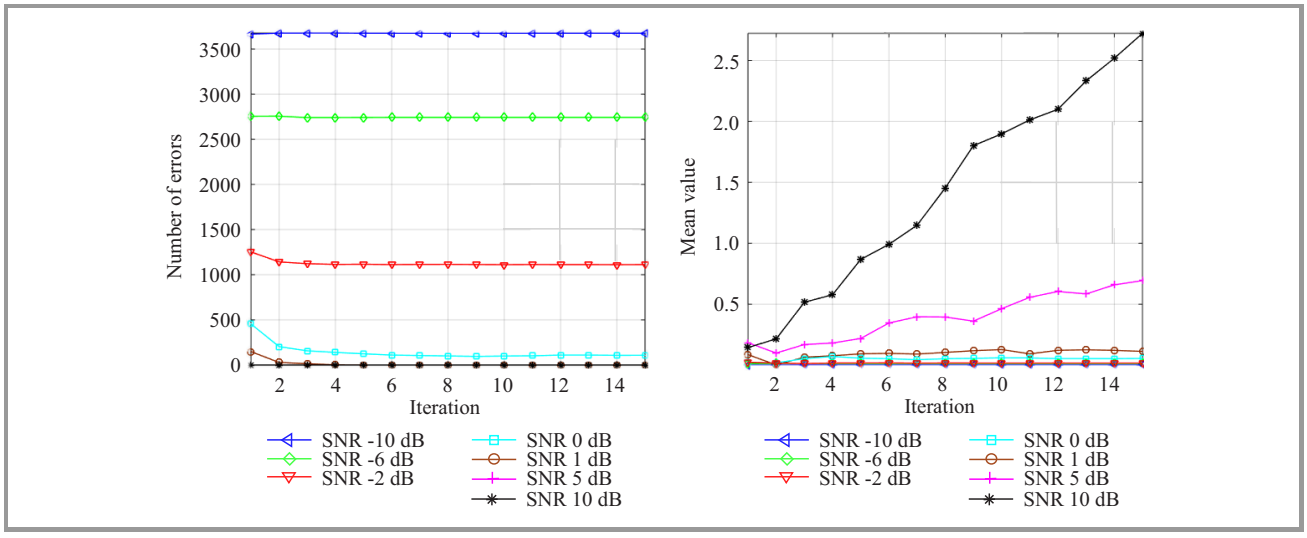


Fig. 11. Analysis of number of errors and absolute value of mean varying with iterations at decoder 2.

A two state turbo code with generator polynomial $[1, \frac{1}{1+D}]$, overall coding rate of $\frac{1}{3}$ and random interleaver is considered. It can be observed from the figures that for low SNR the extrinsic values at the output of the two component decoders are almost constant from first iteration to the maximum iteration number of 15. It can also be seen that the extrinsic values increase for high SNR from first iteration to the maximum iteration number of 15. Extrinsic information calculated at each decoder serves as the a priori information for next component decoder, which in turn uses the extrinsic information for calculation of the branch metric. It can be seen from Eq. 1 that the branch metric is computed by adding the a priori information and channel reliability. Branch metric calculation is dominated more by the higher magnitude extrinsic information. At low SNR the effect of extrinsic values, which serve as a priori information for calculating branch metric values, are negligible. Depending on the channel conditions and errors introduced in the received information, it is either possible to correctly decode before the maximum number of iterations or unable to achieve an acceptable performance even after infinite iterations.

The proposed approach is based on monitoring the absolute value of the mean of the extrinsic information at the output of the component decoders. As the magnitude of extrinsic values changes with iteration and SNR, the mean value also changes correspondingly. Figures 10 and 11 show how the absolute of mean and corresponding errors vary with iteration for a frame size of 10,000.

A close examination of the simulation results presented in Figs. 6 to 11 show that in most of the cases same BER is achieved from the output of the two component decoders after the first few iterations. It can also be seen that many cases require only two iterations to decode correctly since no improvement is observed beyond that. At high SNR conditions, the mean value increases with iterations and requires only a single half iteration or single iteration to correctly decode the sequence. It is also observed that at low SNR conditions, the mean value is almost constant between iterations and the performance improvement is negligible between first and last iteration. In general, the desired output may be reached at the outputs of both the component decoders well before the fixed number of iterations or never reached even after infinite iterations. Based on this analysis, authors propose a new early iteration termination technique for low and high SNR conditions. It utilizes the half iteration termination technique, which reduces the additional computational complexity.

The technique calculates the mean of the extrinsic information at the output of component decoders and compares the absolute value of mean at iteration i with a predefined threshold as in Eq. (7):

$$\text{abs}\big(Mean(i)\big) \geq Th1\,. \tag{7}$$

If the condition in Eq. (7) is satisfied, the iteration is terminated, else it compares the absolute value of mean at iteration i with the mean values of previous iterations as in Eq. (8):

$$\text{abs}\big(Mean(i) - Mean(i-1)\big) \leq Th2\,. \tag{8}$$

If the condition in Eq. (8) is satisfied, the iteration is terminated. The Algorithm 1 shows the proposed technique.

Algorithm 1: Proposed technique

Initialization: $i = 1$; stop $= 1$; set imax
while stop or $i \leq$ imax
  Perform the i-th iteration for component decoder 1
  Calculate *Mean1*
  if abs$(Mean1(i)) \geq$ Th1
    stop $= 0$
  end if
  if $i > 1$ && abs$(Mean1(i) - Mean1(i-1)) \leq$ Th2
    stop $= 0$
  end if
  Perform the i-th iteration for component decoder 2
  Calculate *Mean2*
  if abs$(Mean2(i)) \geq$ Th1
    stop $= 0$
  end if
  if abs$(Mean2(i) - Mean2(i-1)) \leq$ Th2
    stop $= 0$
  end if
  $i = i + 1$
end while
Final output

5. Simulation Results

Simulation results for the proposed extrinsic mean based iteration stopping technique for turbo code are presented here. Simulations were performed on turbo code transmitted over AWGN channel with BPSK modulation with the maximum number of iterations set to 15. A two state turbo code with generator polynomial $\left[1, \frac{1}{1+D}\right]$,

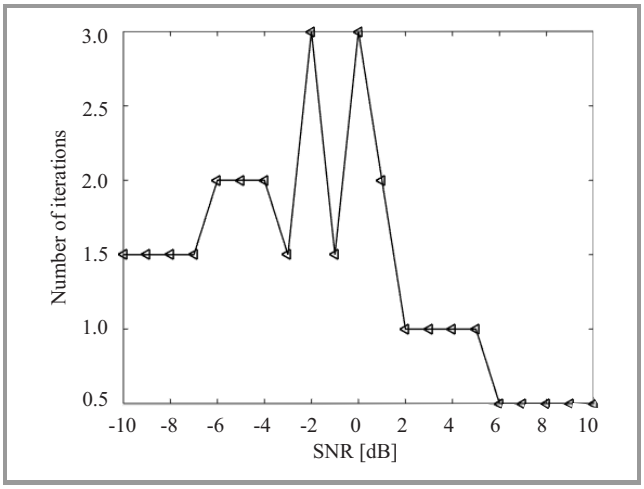


Fig. 12. Number of iterations as a function of SNR.

overall coding rate of $\frac{1}{3}$ and random interleaver is considered. The performance and complexity trade off depend on the threshold values. The threshold values selected are as $0.15 \leq \text{Th1} \leq 0.5$ and $0 \leq \text{Th2} \leq 0.001$. The number of iterations required for different values of SNR are shown in Fig. 12 for a frame size of 10,000 and Th1 = 0.3 and Th2 = 0.0009. As the decoding iterations come down the resulting overall computational reduction leads to a considerable power saving. Thus, the proposed algorithm terminates the unnecessary iterations and reduces both decoding delay and power consumption. The performance analysis of the proposed technique is shown in Fig. 13.

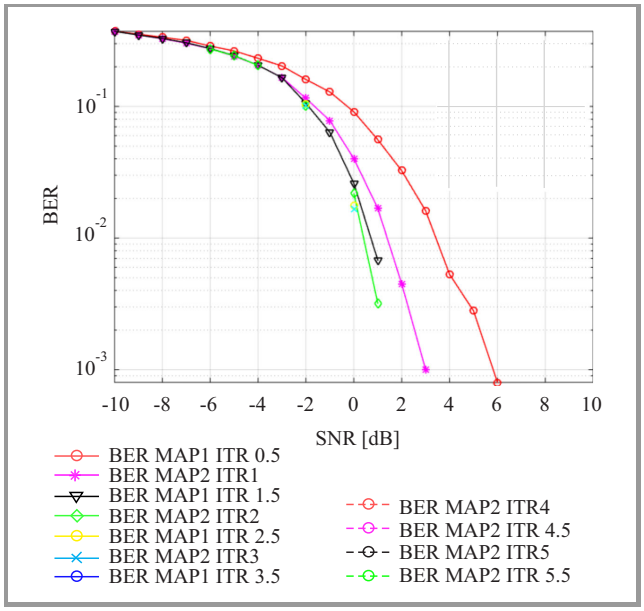


Fig. 13. SNR versus BER performance of the proposed technique.

The performance comparison of the proposed technique with histogram technique, SCR, mean reliability, ME, SDR, fixed iteration and HDA is shown in Fig. 14. Figure 15

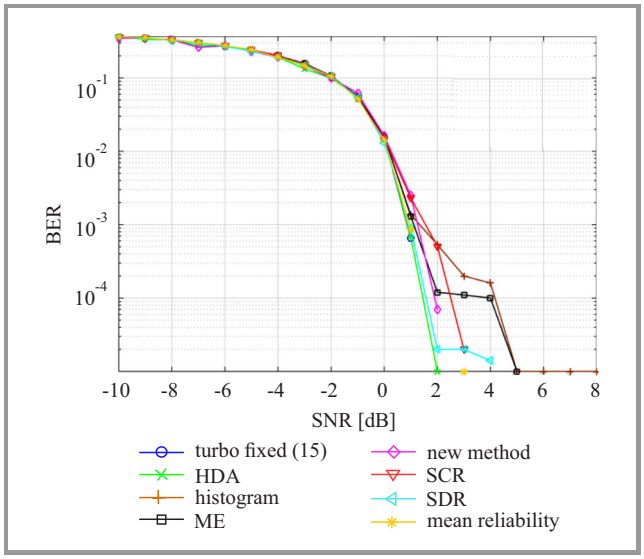


Fig. 14. Comparison of SNR versus BER performance.

shows the corresponding average number of iterations required as a function of SNR. It is observed from the simulation results that the effect of proposed algorithm on turbo decoder helps to reduce decoding delay, computations and thus power consumption. As expected, performance and required number of iterations change with the threshold value. From the simulation results, one can see that proposed technique achieves performance close to that of fixed iteration technique. It is clear that the scheme is able to terminate early for low and high SNR and also able to potentially save half iterations.

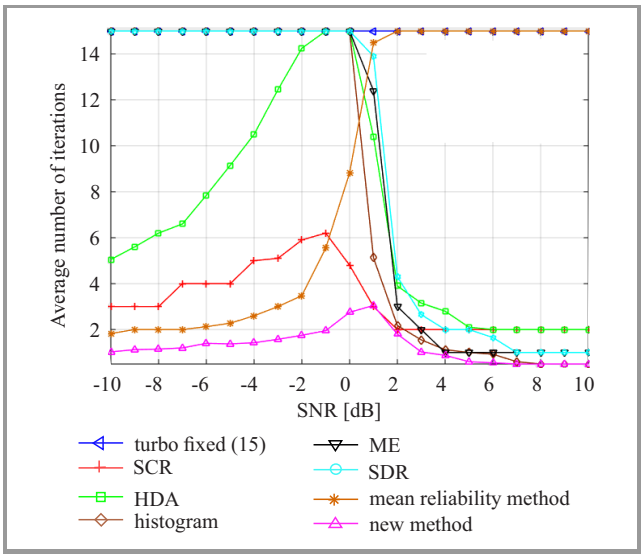


Fig. 15. Comparison of SNR versus average number of iterations.

Computational time and average number of iterations is reduced as compared to the already available techniques. The proposed early termination technique is appropriate for wireless communication systems requiring shorter latency and low power consumption. Table 1 shows the computational time required for the different stopping techniques with maximum iteration kept as 15, SNR from –10 to 10 dB and for a frame size of 10,000 when run on Intel Core i5-4690S @ 3.20 GHz with 64 bit Windows operating system.

Table 1
Computation time comparison

Algorithm	Computation time for 15 iteration and SNR: –10…10 dB
Fixed iteration method	37.22 s
Histogram technique	20.50 s
Sign Difference Ratio	21.93 s
Mean estimate	22.39 s
Mean reliability	24.89 s
Sign Change Ratio	7.811 s
Hard Decision Aided	16.94 s
Mean based new stopping method	**4.57 s**

6. Conclusion

Turbo code achieves significant error correction capability in wireless channel at the cost of computational complexity. A new stopping criteria based on the absolute mean value of extrinsic information is proposed in this paper. Performance is analyzed in terms of BER, average number of iterations and average computation time. The analysis shows that the proposed method can reduce the average number of iterations and decoding delay as compared to other techniques. The method is equally effective at high as well as low SNR conditions in reducing the number of required iterations. It takes advantage of the half iteration but with negligible performance degradation as compared to fixed iteration termination scheme. Average iteration number reduction leads to corresponding power saving, reduction in decoding delay and throughput improvement by decoding more number of data blocks. In the proposed technique, depending on the choice of threshold value, there is a trade-off between performance and complexity of operation. The performance improvement achieved in terms of BER, iteration number, and processing time, makes the proposed method really attractive for systems requiring low latency, low power consumption and better performance in terms of error correction.

References

[1] C. E. Shannon, "A mathematical theory of communication", *ACM SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 5, no. 1, pp. 3–55, 2001.

[2] P. D. Hegu and V. Ghodke, "A survey on designing of turbo encoder & turbo decoder ", *Int. J. Adv. Found. Res. Comput. (JAFRC)*, vol. 2, no. 6, pp. 9–16, 2015.

[3] V. Patil and P. C. Latane, "Implementation of efficient turbo code encoder-decoder with MAX-Log-MAP algorithm", *Int. J. Adv. Technol. Engin. Res.*, no. 2250–3536, pp. 73–77, 2014.

[4] K. M. Bogawar, S. Mungale, and M. Chavan, "Implementation of turbo encoder and decoder", *Int. J. Engin. Trends & Technol. (IJETT)*, vol. 8, no. 2, pp. 73–76, 2014.

[5] A. Imran, "Software implementation and performance of UMTS turbo code", M.Sc. Thesis, Tampere University of Technology, 2013.

[6] F. Gilbert, F. Kienle, and N. Wehn, "Low complexity stopping criteria for UMTS turbo-decoders", in *Proc. 57th IEEE Semiann. Veh. Technol. Conf. VTC 2003 Spring*, Jeju, South Korea, 2003, vol. 4, pp. 2376–2380.

[7] M. Moher, "Decoding via cross-entropy minimization", in *Proc. IEEE Global Telecommun. Conf. GLOBECOM'93*, Houston, TX, USA, 1993, vol. 2, pp. 809–813 (doi: 10.1109/GLOCOM1993.318192).

[8] L. A. Perisoara, R. Stoian, D. I. Sacaleanu, and D. A. Stoichescu, "A new stopping criterion for turbo decoders based on the minimization of error probability", in *Proc. 22nd IEEE Telecommunications Forum TELFOR 2014*, Belgrade, Serbia, 2014, pp. 391–394.

[9] F. Zhai and I. J. Fair, "New error detection techniques and stopping criteria for turbo decoding", in *Proc. IEEE Canadian Conf. Electr. Comput. Engin. CCECE 2000*, Halifax, NS, Canada, 2000, vol. 1, pp. 58–62.

[10] I. Land and P. Hoeher, "Using the mean reliability as a design and stopping criterion for turbo codes", in *Proc. Inform. Theory Worksh.*, Cairns, Australia, 2001, pp. 27–29.

[11] F. Zhai and I. J. Fair, "Techniques for early stopping and error detection in turbo decoding", *IEEE Trans. Commun.*, vol. 51, pp. 1617–1623, 2003.

[12] M. S. Aziz, H. Abdel-Kader, and K. Y. Youssef, "Implementation of a smart and power efficient Turbo Decoder using SDR algorithm", *Int. Conf. Comp. Engin. Syst. ICCES 2011*, Cairo, Egypt, 2011, pp. 173–177.

[13] W. Yufei, B. D. Woerner, and W. J. Ebel, "A simple stopping criterion for turbo decoding", *IEEE Commun. Lett.*, vol. 4, pp. 258–260, 2000.

[14] F. M. Li, C. H. Lin, and A.-Y. Wu, "A new early termination scheme of iterative turbo decoding using decoding threshold", in *Proc. IEEE Worksh. Signal Process. Syst. Design & Implement.*, Banf, Alta, Canada, 2006, pp. 89–94.

[15] K. N. V. Khasim, R. Kumar, K. P. Raju, and V. Glory, "Implementation of a Turbo Encoder and Turbo Decoder on DSP Processor TMS320C6713", *Int. J. Engin. Res. Develop.*, vol. 2, no. 5, pp. 37–41, 2012.

[16] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes", *Int. Conf. Commun. ICC'93*, Geneva, Switzerland, 1993, vol. 2, pp. 1064–1070.

[17] P. Chhabra and V. Nath, "An efficient high performance turbo code implementation in Rayleigh fading channel", in *Proc. IEEE Symp. Wirel. Technol. & Appl. ISWTA 2014*, Kota Kinabalu, Malaysia, 2014, pp. 47–52 (doi: 10.1109/ISWTA.2014.6981193).

[18] R. Bose, *Information Theory, Coding and Cryptography*, 2 ed. New Delhi: Tata McGraw Hill, 2008.

[19] P. Ituero and M. L. Vallejo, "Further specialization of clustered VLIW processors: A MAP decoder for Software Defined Radio", *ETRI J.*, vol. 30, pp. 113–128, 2008.

[20] I. Amamra and N. Derouiche , "A stopping criteria for turbo decoding based on the LLR histogram", *IEEE 16th Mediterranean Electrotech. Conf. MELECON 2012*, Yasmine Hammamet, Tunisia, 2012, pp. 699–702.

[21] S. Lin and J. D. Costello, *Error Control Coding*, 2 ed. Prentice-Hall Inc. USA, 2004.

[22] P. Reddy, F. Clermidy, A. Baghdadi, and M. Jezequel, "A low complexity stopping criterion for reducing power consumption in Turbo Decoders", in *Design, Autom. & Test in Europe DATE 2011*, Grenoble, France, 2011, pp. 1–6.

[23] A. Andreoni, M. Bondani, and M. A. C. Potenza, "Energy efficient turbo decoding for 3G mobile", in *Proc. 2001 Int. Symp. Low Power Electron. & Design ISLPED 2001*, Huntington Beach, CA, USA, 2001, pp. 328–333 (doi: 1.1109/LPE.2001.945427)

[24] J. Geldmacher, K. Hueske, J. Gotze, and M. Kosakowski, "Hard decision based low SNR early termination for LTE Turbo decoding", in *Proc. IEEE 8th Int. Symp. Wirel. Commun. Syst. ISWCS 2011*, Aachen, Germany, 2011, pp. 26–30 (doi: 10.1109/ISWCS.2011.6125303).

[25] R. Mohamad, H. Harun, M. Mokhtar, and W. A. W. Adnan, "Performance Analysis of Stopping Turbo Decoder Iteration Criteria", in *IEEE 10th Int. Colloq. Sig. Process. & Its Appl. CSPA 2014*, Kuala Lumpur, Malaysia, 2014, pp. 5–9.

[26] P. Salija, and B. Yamuna, "Optimum energy efficient error control techniques in wireless systems: A survey", *J. Commun. Technol. Electron.*, vol. 60, no. 11, pp. 1257–1263, 2015.

[27] A. Matache, S. Dolinar, and F. Pollara, "Stopping Rules for Turbo Decoders", in *The Telecommunications and Mission Operations Progress Report 42-142*, Jet Propulsion Laboratory, 2000 [Online]. Available: http://ipnpr.jpl.nasa.gov/progress_report/42-142/142J.pdf

[28] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes", *IEEE. Trans. Inform. Theory*, vol. 42, no. 2, pp. 429–445, 1996.

[29] R. Y. Shao, Shu Lin, and M. P. C. Fossorier, "Two simple stopping criteria for turbo decoding", *IEEE. Trans. Commun.*, vol. 47, no. 8, pp. 1117–1120, 1999.

P. Salija received her B.Tech. degree in Electronics and Communication Engineering in 2011 from Cochin University of Science and Technology and M.Tech. degree in Communication Systems in 2013 from Karunya University. She is currently a Ph.D. student in the Department of ECE at Amrita School of Engineering. Her current research interest is error control coding in wireless systems.

E-mail: p_salija@cb.amrita.edu
Department of Electronics
and Communication Engineering
Amrita School of Engineering, Coimbatore
Amrita Vishwa Vidyapeetham
Amrita University India

B. Yamuna – for biography, see this issue, p. 112.

Information for Authors

Journal of Telecommunications and Information Technology **(JTIT)** is published quarterly. It comprises original contributions, dealing with a wide range of topics related to telecommunications and information technology. **All papers are subject to peer review**. Topics presented in the JTIT report primary and/or experimental research results, which advance the base of scientific and technological knowledge about telecommunications and information technology.

JTIT is dedicated to publishing research results which advance the level of current research or add to the understanding of problems related to modulation and signal design, wireless communications, optical communications and photonic systems, voice communications devices, image and signal processing, transmission systems, network architecture, coding and communication theory, as well as information technology.

Suitable research-related papers should hold the potential to advance the technological base of telecommunications and information technology. Tutorial and review papers are published only by invitation.

Manuscript. TEX and LATEX are preferable, standard Microsoft Word format (.doc) is acceptable. The author's JTIT LATEX style file is available:
http://www.nit.eu/for-authors

Papers published should contain up to 10 printed pages in LATEX author's style (Word processor one printed page corresponds approximately to 6000 characters).

The manuscript should include an abstract about 150–200 words long and the relevant keywords. The abstract should contain statement of the problem, assumptions and methodology, results and conclusion or discussion on the importance of the results. Abstracts must not include mathematical expressions or bibliographic references.

Keywords should not repeat the title of the manuscript. About four keywords or phrases in alphabetical order should be used, separated by commas.

The original files accompanied with pdf file should be submitted by e-mail: redakcja@itl.waw.pl

Figures, tables and photographs. Original figures should be submitted. Drawings in Corel Draw and PostScript formats are preferred. Figure captions should be placed below the figures and can not be included as a part of the figure. Each figure should be submitted as a separated graphic file, in .cdr, .eps, .ps, .png or .tif format. Tables and figures should be numbered consecutively with Arabic numerals.

Each photograph with minimum 300 dpi resolution should be delivered in electronic formats (TIFF, JPG or PNG) as a separated file.

References. All references should be marked in the text by Arabic numerals in square brackets and listed at the end of the paper in order of their appearance in the text, including exclusively publications cited inside. Samples of correct formats for various types of references are presented below:

[1] Y. Namihira, "Relationship between nonlinear effective area and mode field diameter for dispersion shifted fibres", *Electron. Lett*., vol. 30, no. 3, pp. 262–264, 1994.

[2] C. Kittel, *Introduction to Solid State Physics*. New York: Wiley, 1986.

[3] S. Demri and E. Orłowska, "Informational representability: Abstract models versus concrete models", in *Fuzzy Sets, Logics and Knowledge-Based Reasoning*, D. Dubois and H. Prade, Eds. Dordrecht: Kluwer, 1999, pp. 301–314.

Biographies and photographs of authors. A brief professional author's biography of up to 200 words and a photo of each author should be included with the manuscript.

Galley proofs. Authors should return proofs as a list of corrections as soon as possible. In other cases, the article will be proof-read against manuscript by the editor and printed without the author's corrections. Remarks to the errata should be provided within one week after receiving the offprint.

A copy of the JTIT is provided to each author of paper published.

Journal of Telecommunications and Information Technology has entered into an electronic licencing relationship with EBSCO Publishing, the world's most prolific aggregator of full text journals, magazines and other sources. The text of *Journal of Telecommunications and Information Technology* can be found on EBSCO Publishing's databases. For more information on EBSCO Publishing, please visit www.epnet.com.

(*Contents Continued from Front Cover*)

INSTYTUT ŁĄCZNOŚCI
PAŃSTWOWY INSTYTUT BADAWCZY

Editorial Office
National Institute
of Telecommunications
Szachowa st 1
04-894 Warsaw, Poland

tel. +48 22 512 81 83
fax: +48 22 512 84 00
e-mail: redakcja@itl.waw.pl
http://www.nit.eu

JOURNAL OF TELECOMMUNICATIONS AND INFORMATION TECHNOLOGY 2/2016