



**Proceedings of the
2nd International Conference
on Applied Innovations in IT**

Eduard Siemens (editor in chief) et al.

Proceedings of the 2nd International Conference on Applied Innovations in IT

Eduard Siemens (editor in chief) et al.

This volume contains publications of the 2nd International Conference on Applied Innovations in IT (ICAIIIT), which took place in Koethen March 27th 2014. The conference is devoted to problems of applied research in the fields of automation and communications. The research results can be of interest for researchers and development engineers, who deal with theoretical base and the application of the knowledge in the respective areas.

Proceedings of the 2nd International Conference on Applied Innovations in IT, 2014.

editors: Prof. Dr. Eduard Siemens* (editor in chief),
Dr. Bernd Krause*, Dr. Leonid Mylnikov**

(*Anhalt University of Applied Sciences, ** Perm National Research Polytechnic University)

ISBN 978-3-86011-071-3 (Online)

Content

Section 1. Communication Technologies

Dmitrii Dugaev, Eduard Siemens:

A Wireless Mesh Network NS-3 Simulation Model: Implementation and Performance Comparison With a Real Test-Bed 1

Irina Fedotova, Eduard Siemens:

System Time Issues for the ARM Cortex A8 Processor 7

Danijela Efnusheva, Natasha Tagasovska, Aristotel Tentov and Marija Kalendar:

Efficiency Comparison of DFT/IDFT Algorithms by Evaluating Diverse Hardware Implementations, Parallelization Prospects and Possible Improvements 11

Dmitry Kachan, Eduard Siemens:

Comparison of Contemporary Protocols for High-speed Data Transport via 10 Gbps WAN Connections 21

Bojan Gruevski, Aristotel Tentov and Marija Kalendar:

Implementation of Multi-Core Processor Based on PLASMA (Most MIPS I) IP Core 29

Anton Merkulov, Viatcheslav Shuvalov:

The Research of Increase of Channel Efficiency for IP Traffic Transmission over Digital Power Line Carrier Channels 37

Maya Malenko:

Implementation of Reed-Solomon RS(255,239) Code 43

Tatyana Chavdarova:

Parallel Architecture Prototype for 60 GHz High Data Rate Wireless Single Carrier Receiver 49

Anton Merkulov:

Assessment of the VoIP Transmission Quality over Digital Power Line Carrier Channels ... 57

Hao Hu, Dmitry Kachan, Eduard Siemens:

Improving TCP Performance on CSMA/CA Connections 61

Section 2. Automation and Algorithms

Evgeny Yakimov, Alexander Goldshtein, Valery Bulgakov, Sergey Shipilov:

Data Processing at the Flaw Detector with Combined Multisector
Eddy-Current Transducer 67

Leonid Mylnikov:

Application of Fuzzy Variables for Systems of Management Decision Support 73

Anton Petrochenkov:

Regarding to Implementation of Genetic Algorithms in Life Cycle Management
of Electrotechnical Equipment 79

Igor Shmidt:

Storing Data in the Trial of Complex Technical Products 85

A Wireless Mesh Network NS-3 Simulation Model: Implementation and Performance Comparison With a Real Test-Bed

Dmitrii Dugaev, Eduard Siemens

Anhalt University of Applied Sciences - Faculty of Electrical, Mechanical and Industrial Engineering
Bernburger Str. 57, 06366 Koethen, Germany

E-mail: [d.dugaev, e.siemens}@emw.hs-anhalt.de](mailto:{d.dugaev, e.siemens}@emw.hs-anhalt.de)

Abstract—Wireless mesh networks present an attractive communication solution for various research and industrial projects. However, in many cases, the appropriate preliminary calculations which allow predicting the network behavior have to be made before the actual deployment. For such purposes, network simulation environments emulating the real network operation are often used. Within this paper, a behavior comparison of real wireless mesh network (based on 802.11s amendment) and the simulated one has been performed. The main objective of this work is to measure performance parameters of a real 802.11s wireless mesh network (average UDP throughput and average one-way delay) and compare the derived results with characteristics of a simulated wireless mesh network created with the NS-3 network simulation tool. Then, the results from both networks are compared and the corresponding conclusion is made. The corresponding results were derived from simulation model and real-world test-bed, showing that the behavior of both networks is similar. It confirms that the NS-3 simulation model is accurate and can be used in further research studies.

Keywords: wireless mesh networks, 802.11s, HWMP, network simulation, NS-3.

I. INTRODUCTION

A concept of wireless mesh networks (WMN) has become popular among academic researchers and telecommunication industry within the last decade. The most attractive property of such networks is a possibility of rapid and cost-effective deployment of networks with ability to provide high-capacity services for an end-user. Moreover, a diversity of wireless mesh networks applications is very wide, making it a convenient solution to use them in various projects in diverse areas, such as transport (VANET) [1], civil and military communication infrastructures, environment and public safety [2][3].

At Anhalt University of Applied Sciences (HSA), the Future Internet Lab Anhalt (FILA) is currently conducting a couple of projects with extensive usage of wireless mesh technologies. During the implementation stage of these projects, a question of adequate performance evaluation of WMN arises. As a solution of this problem, a networking simulation environment NS-3 is proposed to be used [4]. Based on the simulation, it is possible to imitate, run and to see a network behavior with any topology, size, mobility, wireless medium parameters and traffic profiles and

intensity. To do that, we need to implement a simulation environment, which behavior will be adequate and comparable to a behavior of a real wireless mesh network under real conditions. This is the main objective of this paper.

For a real-world test-bed, a 802.11s wireless mesh networking standard [5] has been used – an open80211s Linux implementation [6], in particular. For the simulation model – an NS-3 networking simulator with implemented 802.11s MAC-layer stack [7] is chosen. As for performance parameters – an average UDP throughput and an average delay will be evaluated in both test-bed and simulation model.

A. Test-bed description

To be able to compare parameters of the simulated network with a real-world scenario, a test-bed network has been deployed. The network consists of 4 wireless nodes placed in 2x2 grid topology with a distance between them equals to 1 meter. The topology of the test network is illustrated on Fig.1.

As the network's node, an ARM-based system-on-chip (SoC) had been used with a wireless 802.11 adapter based on rt2800 mesh-compatible driver [8]. Then, a Linux OS with modified kernel and the open80211s [6] implementation of the mesh standard has been installed on the SoC [9] (as in Fig. 2).

A mesh mode was switched on with HWMP routing protocol with default Airtime Link routing metric [10], equation (1):

$$c_a = \left[O + \frac{B_t}{r} \right] \cdot \frac{1}{1 - e_f}, \quad (1)$$

where

c_a – the airtime cost;

O – a constant, which defines a channel access time depending on the used physical implementation (802.11a, 802.11b);

B_t – test packet size (8192 bits);

r – channel data rate (Mbps);

e_f – packet error probability.

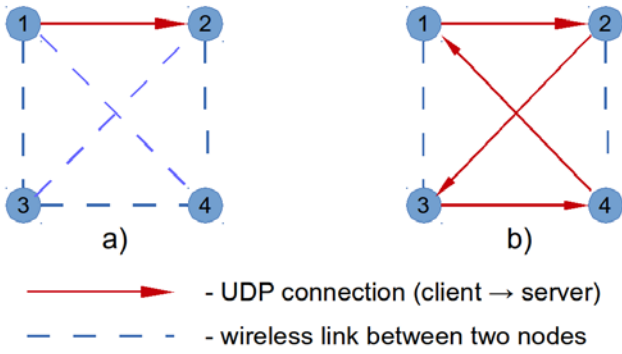


Fig. 1. Test network topology with two experiments:
a – 1 UDP connection;
b – 4 simultaneous UDP connections.

To evaluate r and e_f values, a node sends a block of test packets. The length of the test packet is a default value which is set by the standard – 8192 bits.

In a case of independent packet errors, the packet error probability can be calculated as (2):

$$e_f = 1 - (1 - p_o)^n \approx np_o, \quad (2)$$

where

p_o – transmission bit error probability;
 n – number of bits being transmitted.

B. Simulation description

For the simulation purposes, NS-3 network simulator has been used [4] [11]. NS-3 is a discrete-event simulator with a special focus on Internet-based systems, consisting of different library components (core, simulation, node libraries, physical and channel models, network routing protocols implementations, etc.) written in C++. Such structure allows researchers to modify, adjust and simulate various networking scenarios. The general simulation architecture of NS-3 [11] is depicted on Fig. 3.

In order to simulate the real-world test-bed as accurate as possible, a few main network characteristics must match:

- the same 2x2 grid static topology;
- propagation loss model;
- physical interference model;
- modulation scheme and the frequency range (802.11g, 6Mbps OFDM rate);
- 802.11s peer-link management protocol and HWMP parameters;

The 802.11g modulation standard has been used in both simulation and real test-bed since this is the most recent specification which is supported by the current 802.11s

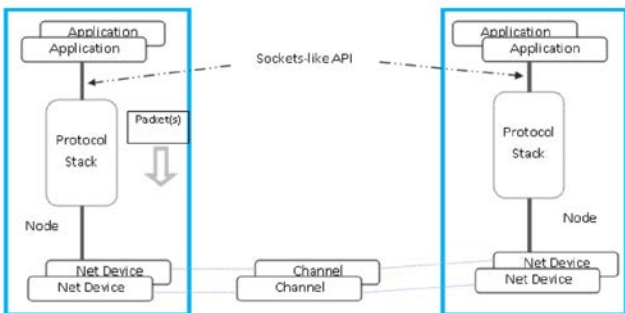


Fig. 3. NS-3 main simulation objects.

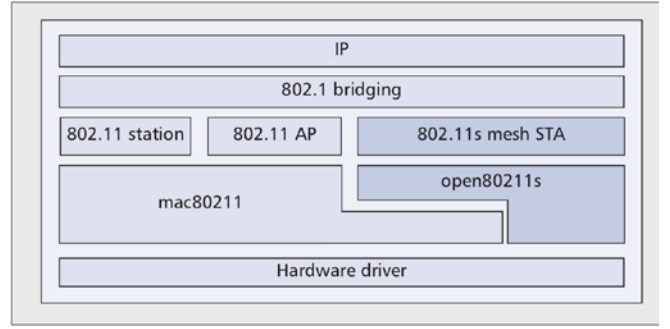


Fig. 2. The open80211s stack integrated into the Linux kernel.

implementation (open80211s). The 802.11n standard is not supported yet by mesh devices.

HWMP stands for Hybrid Wireless Mesh Protocol, which is used as a default multi-hop routing scheme in 802.11s standard.

One of the important parameters used in the simulation is the chosen propagation loss model which should consider the real-world wireless medium irregularities such that various obstacles (walls, doors, people, etc.) and interferences (electrical equipment, other wireless systems, etc.). These factors decrease the signal strength in different ways.

In NS-3, there are several available propagation loss models [11]: fixed RSS loss model, Friis propagation loss model, Jakes propagation loss model, Nakagami propagation loss model, random propagation loss and log distance propagation loss models. They are used in different wired and wireless communication scenarios.

For our case, where the distance between the nodes is short and static, and the measurements are conducted in a building, the Log-Distance Propagation Loss Model [11] was the most suitable and adaptive. This model calculates the reception power (received signal strength) using the following equation (3):

$$L = L_0 + 10 \cdot n \cdot \log_{10} \left(\frac{d}{d_0} \right), \quad (3)$$

where

L_0 – path power loss (signal attenuation) at reference distance (dB);

n – the pass loss distance exponent;

d – distance (m);

d_0 – reference distance (m);

L – path power loss (signal attenuation) (dB).

To adapt this propagation loss model to a real test-bed, correct values of these variables has to be found. A YansWifi [11] physical interference model has been used to simulate the signal's interference; a frequency range and modulation scheme have been taken from 802.11g standard with 6 Mbps bitrate. 802.11s HWMP routing parameters have been set to default values.

II. THE 802.11S ROUTING SCHEME

A 802.11s standard was developed by IEEE 802.11 task group in 2006 with main objective to overcome the limitations of traditional Wi-Fi star topologies (Access Point – Clients) and to enable a deployment of wireless mesh

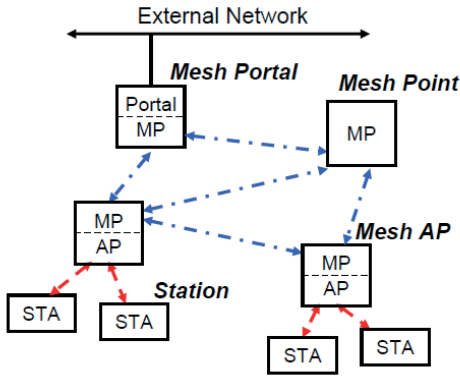


Fig. 4. A general architecture of 802.11s WMN network.

networks based on 802.11 MAC layer with layer-2 (L2) multi-hop routing scheme, provided by HWMP.

A general 802.11s based WMN architecture is shown in Fig. 4. It consists of 4 classes of devices [9]:

- Mesh Point (MP) – wireless mesh node with data routing and forwarding functionality;
- Mesh Access Point (MAP) – wireless mesh node with additional functionality of wireless access point (AP), allowing different wireless clients to connect to WMN;
- Mesh Portal (MPP) – a gateway connecting WMN with another external networks;
- Stations (STA) – wireless 802.11 devices connected to MAP;

Currently, the 802.11s standard categorizes frames as data, control and management frames. Control frames are used for exchanging acknowledgements and path reservation messages, whereas the management frames perform a function of establishing and maintaining the WMN. The 802.11s frame format provides additional mesh control fields for data routing over multiple hops, including source address (initial hop), destination address (final hop) and mesh source address [10].

III. HWMP PROTOCOL

Hybrid Wireless Mesh Network Protocol belongs to the class of so-called hybrid routing schemes, which means that it can work in both proactive and active modes alternatively or simultaneously.

In the reactive mode (Fig. 5a), the node's forwarding table is created right before the beginning of data transmission. Upon sending the data packets, a source node sends a broadcast Path Request (PREQ) message to its neighbors which change the Air-Time Metric value and forward it further. A sink node receives the PREQ and sends back a Path Reply (PREP) message back to the source. When the source node gets the reply (PREP), it obtains the information about the whole path (path metric) and makes a transmission decision [12].

In the proactive mode (as in Fig. 5b), the wireless mesh network has to set a root node, which transmits broadcast root requests (RREQ) in order to form a path tree. In such case, every mesh node has the information about how to deliver data packets to every other node through the root beforehand.

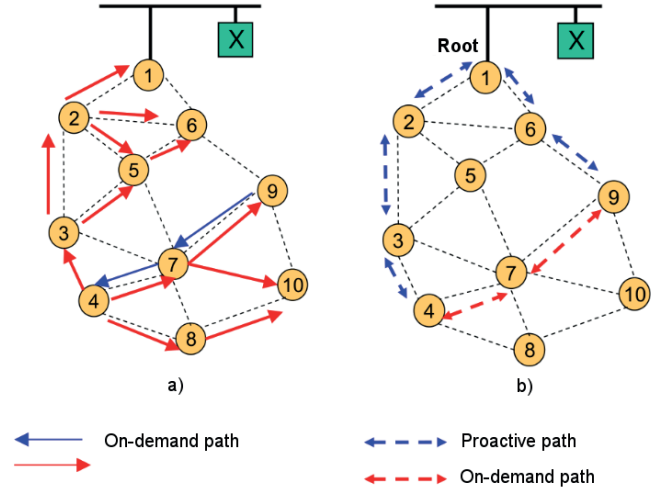


Fig. 5. Connection establishing between nodes 4 and 9: [5]
a – in HWMP reactive mode;
b – in HWMP proactive mode.

IV. EXPERIMENT MAP

As it has been mentioned above, the test topology of the WMN (real-world test-bed and NS-3 simulation script) represent a 2x2 grid with 1 m distance between two nodes, as it is illustrated on Fig.1.

Two main performance parameters are chosen for the evaluation – average UDP throughput and average one-way delay. A network traffic load parameter (in the form of a number of simultaneous UDP connections) is chosen as a variable value.

In the real-world test-bed, the throughput value of the generated UDP traffic is estimated by the client side of the iperf utility [13] after the channel having been loaded according to its established transmission bitrate.

The average one-way delay (OWD) under different traffic loads was measured using LTest tool [14]. The LTest utility generates UDP traffic with fixed data rate and calculates one-way delay from that. However, in contrary to measurements of RTT, on OWD, a high-precision time synchronization of the corresponding CPU clocks is crucial, which is provided by LTest. This provides highly accurate one-way delay values between sender (client) and receiver (server).

The duration of each test in the real-world test-bed as well as in the simulated network has been set to 100 seconds. The UDP packet size and each client's UDP stream data rate have been set to 1024 Bytes and 6 Mbps correspondingly.

In the simulation script, the aforementioned parameters were calculated using formulas (4) and (5):

$$R_{avg} = \frac{N_{rec.bits}}{T_{lastbit} - T_{firstbit}}, \quad (4)$$

where

$N_{rec. bits}$ – number of received bits;

$T_{last bit}$ and $T_{first bit}$ – time of the last and the first bits received correspondingly;

$$OWD_{avg} = \frac{T_{TOTAL}}{N_{rec. packets}}, \quad (5)$$

where

T_{TOTAL} – total transmission time;

$N_{rec. packets}$ – number of received packets.

V. MEASUREMENT AND SIMULATION RESULTS

After having conducted a number of experiments, we derived the dependencies of average throughput and average one-way delay as functions of UDP traffic load (number of UDP connections each with 6 Mbps data rate). It was stated that as the traffic load increases, a common wireless mesh network decreases significantly due to physical characteristics of wireless transmission medium (noisy environment, increasing interference) and current open80211s implementation. This fact is confirmed by the NS-3 simulation, where a similar behavior is detected. In general, the simulated average throughput values correspond to the ones from the test-bed (as in Fig. 6). The absolute average throughput values in the worst case differ in about 34%, and in the best case – 14%.

Furthermore, the average one-way delay increases as the number of UDP connections grow in both real test-bed and simulated networks with a very similar behavior. It can be explained with the fact that the 802.11s standard is mostly based on 802.11 MAC-layer which exploits CSMA/CA multiple access scheme. Therefore, with the increasing network traffic, the number of frame retransmission quickly increases, affecting the overall delay. Moreover, the current 802.11s implementation (open80211s) is still under extensive development and sometimes we were faced with unstable network behavior under very high loads. However, under medium traffic loads, the real-world test-bed network behavior is stable, which is confirmed by the simulation illustrated on Fig. 7.

The average throughput values derived from both scenarios and their percentage difference are presented in Table 1. As it can be seen from the numbers, the relative difference between the real test-bed and the simulation model does not exceed 35% in the worst case (2 UDP connections), and is equal approximately to 14% in the best case (3 UDP connections). The simulation model becomes more accurate when there are two and more simultaneous connections, which can be explained by the fact that the signal interference starts to play the most significant role in the network performance behavior, and the YansWifi signal interference model is well-implemented in NS-3 simulator.

In Table 2, the average one-way delay values and their corresponding percentage difference are presented. In this scenario, the similarity in the real test-bed and the simulation model behaviors is even better. The worst percentage difference is only 24% and the best one is 14%. In this case, it is quite difficult to establish the relation between UDP traffic intensity and the simulation model accuracy. These random fluctuations in accuracy deviations can be explained only by the randomness of wireless channel model used in NS-3 simulator and by unpredictable behavior of the real wireless medium.

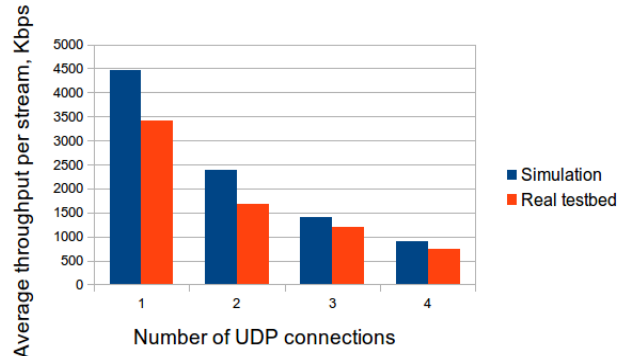


Fig. 6. Average UDP throughput per stream behavior in real test-bed and simulation.

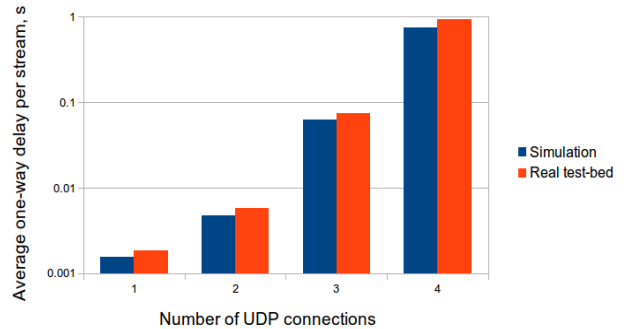


Fig. 7. Average one-way delay per stream in real test-bed and simulation.

TABLE I
AVERAGE THROUGHPUT VALUES OF REAL TEST-BED AND SIMULATION, (KBPS)

Number of UDP connections	1	2	3	4
Simulation, Kbps	4471.6	2387.8	1403.6	908.3
Real test-bed, Kbps	3415.8	1677.3	1209.3	744.7
Percentage difference	26 %	34 %	14 %	19 %

TABLE II
AVERAGE ONE-WAY DELAY VALUES OF REAL TEST-BED AND SIMULATION, (MS)

Number of UDP connections	1	2	3	4
Simulation, ms	1.54	4.81	62.85	743.1
Real test-bed, ms	1.84	5.84	73.86	946.34
Percentage difference	17 %	19 %	16 %	24 %

All in all, a discrepancy in the calculated values is acceptable and can be explained by non-ideal wireless physical medium (additional noise sources, random noise, etc.), occasional unstable behavior of open80211s implementation and wireless driver (rt2800) as well as current 802.11s model characteristics in NS-3. However, the similarity between test-bed and simulated network model is clearly seen.

VI. CONCLUSION

Basing on the results derived from the experiments, we can observe that the behaviors of the real test-bed and the simulated network are quite similar. It means, that the physical, interference and channel models in NS-3 simulator are set with sufficient accuracy and fit to the purposes of evaluation performance parameters of different wireless mesh networks with various topologies. Moreover, the

802.11s protocol stack (Peer Link Management protocol and HWMP) implemented in NS-3 adequately imitates hybrid routing schemes and link establishing algorithms of real wireless mesh network implementation based on open80211s.

Therefore, the experiment results allow us to conclude that the proposed NS-3 simulation model can be used for wireless mesh networks performance evaluation with satisfactory accuracy in ongoing FILA projects (SoCiEr, Smartlight) as well as in any future projects which involve wireless mesh networking communication.

REFERENCES

- [1] S. Zeadally, R. Hunt, Y. Chen, A. Irwin, and A. Hassan, "Vehicular Ad Hoc Networks (VANETS): Status, Results, and Challenges," 2010.
- [2] H. Aiache, V. Conan, G. Guibe, J. Leguay, C. L. Martret, X. Gonzalez, A. Zeini, and J. Garcia, "WIDENS: Advanced Wireless Ad-Hoc Networks for Public Safety," IST summit, 2005.
- [3] A. Yarali, B. Ahsant, and S. Rahman, "Wireless Mesh Networking: A Key Solution for Emergency & Rural Applications," The Second International Conference on Advances in Mesh Networks, 2009.
- [4] NS-3 network simulator official website. [Online]. Available: www.nsnam.org
- [5] "IEEE Draft Standard for Information Technology-Telecommunications and information exchange between systems-Local and metropolitan area networks - Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications-Amendment 10: Mesh Networking," IEEE P802.11s/D10.0, March 2011, pp. 1–379, 29, 2011.
- [6] open80211s - open-source implementation of the recently ratified IEEE 802.11s wireless mesh standard. [Online]. Available: <http://open80211s.org/>
- [7] K. Andreev and P. Boyko, "IEEE 802.11s mesh networking NS-3 model," 2009.
- [8] Existing Linux Wireless drivers. [Online]. Available: <http://wireless.kernel.org/en/users/Drivers>
- [9] G. Hiertz, D. Denteneer, S. Max, R. Taori, J. Cardona, L. Berlemann, and B. Walke, "IEEE 802.11s: The WLAN Mesh Standard," IEEE Wireless Communications, vol. 17, no. 1, pp. 104-111, 2010.
- [10] R. C. Carrano, L. C. S. Magalhaes, D. C. M. Saade, and C. V. N. Albuquerque, "IEEE 802.11s Multihop MAC: A Tutorial," IEEE Communications Surveys and Tutorials Volume 13, Number 1, First Quarter 2011.
- [11] NS-3 doxygen documentation. [Online]. Available: www.nsnam.org/doxygen-release/index.html
- [12] IEEE P802.11s/D1.00. Amendment: Mesh Networking. – IEEE, 2006.
- [13] Iperf - The TCP/UDP Bandwidth Measurement Tool. [Online]. Available: <http://iperf.fr/>
- [14] E. Siemens, S. Piger, C. Grimm, and M. Fromme, "LTest – A Tool for Distributed Network Performance Measurement," Proc. Consumer Communications and Networking Conference, 2004. First IEEE. 2004, pp. 239-244.

System Time Issues for the ARM Cortex A8 Processor

Irina Fedotova, Eduard Siemens

Anhalt University of Applied Sciences - Faculty of Electrical, Mechanical and Industrial Engineering

Bernburger Str. 57, 06366 Koethen, Germany

E-mail: i.fedotova_e.siemens@emw.hs-anhalt.de

Abstract—The use of system-on-chip (SoC) platforms has emerged as an important integrated circuit design trend for communication and industrial control applications. At the same time, requirements of stable, efficient and precise processing time values are growing rapidly. Since the ARM processor doesn't possess known timers of PC-platforms such as TSC counter or HPET timer, the common way to obtain time values on ARM-architecture is still only through Linux system calls which are mapped to ARM specific time counters. Indeed, direct access to hardware can help to reduce costs down to 200 nanoseconds against 2-1 microseconds of the time acquisition via Unix system call interface. However, designing this approach is a challenging task. This paper describes specific issues and features of tracking system time on the ARM Cortex A8 processor under Linux OS.

Keywords: timestamp precision; time-keeping; embedded Linux; ARM architecture.

I. INTRODUCTION

Over the last few years, the ARM architecture has become the most pervasive 32-bit architecture in the world, with wide range of integrated circuits available from various manufacturers. ARM processors are embedded in products ranging from cell/mobile phones to control systems of wind-turbines. Depending on the specific products [1][2][3], requirements of meeting timing constrains vary from reaction in a predefined time frame (as in the hard-real time system) and not critical timely responsiveness (as in the soft-real time system). However, any real-time requirements on ARM embedded projects can't be met without reliable development platform.

The Linux kernel offers interrupt latency less than 100 nanoseconds in most cases on today's fast processors. Although, the non-deterministic nature of task scheduling may make it unsuitable for some hard real-time systems, the wealth of utility programs included with Linux makes it ideal for such tasks as report generation as well as networking and addressing interoperability issues [4][5]. Though the Linux OS, running on diverse ARM architectures becomes increasingly popular in embedded systems, often exactly the lack of high-precision time control pushes system designers to decisions against Linux-based, towards more costly FPGA- or ASIC-based solutions.

Modern PC-platforms come with different hardware timers having different attributes. The most popular in the PC domain is a 64-bit TSC counter which represents relative time values and counts CPU cycles from the power on or

reset of the computer. The HPET device provides multiple timers, each consisting of a timeout register that is compared with its central counter. Meanwhile, the timing mechanism of ARM-architecture appears more complex. The time performance on ARM is limited by a 32-bit register with a relatively low frequency of 41 ns and hence, the overflow occurs in less than every 3 minutes. Additionally, ARM specific restrictions of memory management allow accessing to registers only from kernel space.

The remainder of the paper is organized as follows. In Section II, related work is described. Section III shows the specific details of each initial time source within the ARM Cortex A8 processor. Some experimental results of appropriate timer sources along with their performance characteristics are shown in Section IV. Finally, Section V describes next steps and future work in our effort to develop a tool for efficient high-performance measurements.

II. RELATED WORK

Since the essence and importance of time acquisition in embedded systems has become apparent, several research projects have suggested to design timing mechanism for real-time application [6][7]. However, these researches consider handling of timer on the outdated platforms, such as, for example, AVR microcontrollers. In other proposals, the entire time capturing process is integrated into dedicated hardware devices [8]. For PC-platforms based on x86 and x86-64 processors, the idea of the invention single structure providing access to different hardware timers has already been suggested [9][10]. Nevertheless, our investigations directed to specific timers of ARM-architecture with the potential purpose of designing the new accurate real-time system

III. ARM CORTEX A8 SPECIFIC CLOCKS AND TIMERS

All current investigations are being performed on the *BeagleBone* credit-card-sized Linux computer with single-core AM335x Cortex A8 ARM processor under the Debian GNU/Linux 7.0 with the stable kernel version 3.8.10.

On the given platform, maximum performance and operation timer for user satisfaction is ensured by the special power, reset, and clock management (PRCM) module. This module provides a centralized control for the generation, distribution and gating of most clocks in the device. PRCM gathers external clocks and internally generated clocks for distribution to the model in the device. Moreover, PRCM

manages the system clock generation.

Following to the processor's technical manual [11, Chapter 8], the device has two reference clocks which are generated by on-chip oscillators or externally. These are for the main clock tree and RTC block, respectively. In the case of an external oscillator, the 32-KHz crystal oscillator is controlled and is configurable by RTC. This device also contains an on-chip oscillator. This oscillator is not configurable and is always on. The main oscillator on the device produces the master high frequency clock.

Therefore, we assume that there are oscillators with at least two different frequencies. The first one is widespread 32,768 kHz frequency, which is exactly 2^{15} cycles per second, a convenient rate to use with simple binary counter circuits. In this case, the time resolution is 31,25 μ sec. The second value, so called master frequency, is equaled to 25 MHz with 40 nsec resolution, which allows performing comparatively fast measurements. Table 1 gives more detailed description of available clocks.

TABLE I
CLOCKS RESOLUTION AND MAXIMUM RANGE

Clock	Prescaler	Resolution	Interrupt Period Range	Wraps every
32.768 KHz	1 (min)	31.25 μ s	31.25 μ s to ~36h 35m	37 h
	256 (max)	8 ms	8 ms to ~391d 22h 48m	
25 MHz	1 (min)	40 ns	40 ns to ~171.8s	2, 983 m
	256 (max)	10.24 μ s	~20.5 μ s to ~24h 32m	

A prescaler is an electronic counting circuit, which allows the timer to be clocked at the rate user desires. The prescaler takes the basic timer clock frequency and divides it by some value before feeding it to the timer. For the given clocks, prescaler's divisor varies from 1 to 256 and accordingly provides 8 different values (divisible by two) of clock resolution. So, with every increase of clock range, the resolution is decreased.

According to the processor specification [11, Chapter 20], four possible timers exist on this processor:

- Dual Mode Timer (DMTimer);
- Dual Mode Timer 1 ms (DMTimer 1 ms);
- Real Time Clock Subsystem (RTS_SS);
- WATCHDOG.

The peripheral DMTimer is 32-bit timer and the module contains a free running upward counter with auto reload capability on overflow. The timer counter can be read and written in real-time (while counting) and configured for 32-bit or 16-bit operation. DMTimer can be configured in three modes of operation: timer mode, capture and compare mode. The compare logic allows an interrupt event on a programmable counter matching value. The capture mode allows capturing of the timer value in a separate register based on an input signal. By default, after core reset, the capture and compare modes are disabled.

In fact, DMTimer provides eight multiple timers:

- The DMTimer0 can only be clocked from the internal RC oscillator of 32.768 KHz.

- The DMTimer1 is implemented using the DMTimer_1ms module, which is capable of generating an accurate 1 ms tick using a 32.768 KHz clock. During low power modes, the master oscillator is disabled. Hence, in this scenario for generating the OS 1ms tick and timer based wakeup, it is sourced from the 32K RC oscillator.
- Each functional clock of DMTimer[2-7] is selected using the associated register from 3 possible sources: the 24-MHz system clock, the 32.768 KHz clock (see Table 1) or the external timer input clock. The availability of particular DMTimer[2-7] depends on the platform implementation. The Linux kernel already contains *dmtimer* driver, which allows using system time through accessing to the DMTimer2. Therefore, this timer is considered as preferable.

Additionally, as any other electronic device, the given processor possesses RTC clock, which keeps time of day, and have an alternate source of power, so that it works even with system power off. The RTC supports external 32.768-kHz crystal or external clock source of the same frequency.

The processor also contains a WATCHDOG timer based on an upward 32-bit counter coupled with a prescaler. It causes the system to be reset if it is not poked periodically. After reset generation, the counter is automatically reloaded with the value stored in the watchdog load register, the prescaler is reset and the timer counter begins incrementing again.

IV. TIME FETCHING PERFORMANCE RESULTS

On ARM platforms, all I/O access is memory-mapped. That means that timers' registers have their specific address stored in a known private memory region and access to them potentially possible only from kernel space. The *dmtimer* driver available in Linux also does not provide high-level abstractions. It is merely a small library supplying some functions for the clients of DMTimers to reserve and program the timers. So, the clients have to manage all the low-level programming and interrupt handling themselves. Accordingly, this driver is considered not used by many.

So, the suggested way of avoiding this obstacle for user would be writing a character device driver and implement *mmap()* function and the read-only access to the timer. It will allow mapping of the physical address of the DMTimer2 main counter to a virtual address in the program memory space.

The idea of timers map is as follows: firstly, the user process invokes an appropriate *mmap()* function and the kernel calls the device driver passing all necessary parameters. The driver validates the request and executes a function to map the necessary range of physical addresses into the address space of the user space. The driver returns an exit code to the kernel and the kernel re-dispatches the user process. Therefore, the user process accesses data of timing hardware by accessing the virtual address returned to it from *mmap()* call. Herewith, inaccurate offset in timer register from user space is handled in the open device operation in kernel space and in the case of error, returns user appropriate message.

Table 2 shows the performance results of getting the time values using direct access to hardware versus using a system

call. Initially hardware timer, DMTimer2 with 25 MHz frequency was chosen. For the system call, *clock_gettime()* was issued. For this investigation, time was fetched in a loop of 10 million consecutive runs. The values are not filtered.

TABLE II
THE COSTS OF OBTAINING TIME VALUES ON ARM PROCESSOR

Time source	Mean, μ sec	Standard deviation, μ sec
System Call	1.025	2.201
Direct Access to DMTimer	0.201	0.800

Additionally, during these tests, CPU system time was estimated relatively real execution time of the program. In fact, CPU costs value of system call is more than 10 times greater than direct access to hardware.

TABLE III
CPU COSTS ON ARM CORTEX A8 PROCESSOR

Time source	Real time of test execution	System time	Percentage ratio
System Call	2m 59.203s	0m 20.846s	11.632%
Direct Access to DMTimer	2m 15.723s	0m 1.303s	0.960%

The chart in Fig. 1 demonstrates more detailed results of this experiment. The range of samples was decreased to track the behavior of timers more carefully. It shows the clear similarity of both timers' behaviors and proves that the given system interacts with this particular DMTimer2 hardware.

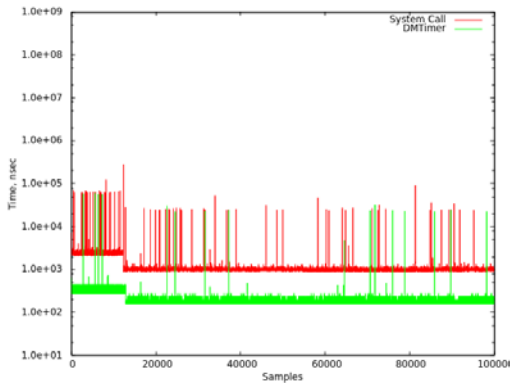


Fig. 1. Measurements of timers costs on the ARM Cortex A8 processor

V. CONCLUSION AND FUTURE WORK

With the rapid development of the field of industrial process control and the wide range of embedded systems, it is necessary to make a higher demand of the time accuracy and reliability of the control system. The embedded ARM Cortex A8 platforms can adapt to strict requirements of the time acquisition and potentially become a basis for building new real-time systems.

Nevertheless, primarily the number of significant issues must be satisfied. Firstly, the reliable interface allowing interacting with system timer must be provided, wherein the benefits of hardware access (such as low cost of obtaining values and low CPU costs) are saved. Secondly, a protection

from timer 2.9 min overflow should guarantee meeting real-time constraints. Therefore, at this stage accomplishment of all above challenges is in progress. Moreover, to the next steps, the better support of the ARM Cortex A9 processor will be addressed. The potential benefits of multi-core processors for real-time embedded systems are enormous, but however have even more dangerous drawbacks.

REFERENCES

- [1] M. Manivannan and N. Kumaresan, "Embedded web server & GPRS based advanced industrial automation using Linux RTOS," *International Journal of Engineering Science and Technology*, vol. 2 (11), no. 8, pp. 6074-6081, 2010.
- [2] D. Wiklund and D. Liu, "SoCBUS: Cwitted network on chip for hard real time embedded systems," *Proc. of the 17th Intl. Symposium on Parallel and Distributed Processing*, Los Alamitos, pp. 78.1, IEEE Computer Society Press, 2003.
- [3] I. Fedotova and E. Siemens, "Usage of high-precision timers in the wind turbines control systems," *Supercomputers Jorنال*, Moscow, Publishing House SCR-Media LTD, Number 16, winter 2013.
- [4] R. Lehrbaum, "Using Linux in Embedded and Real Time Systems," *Linux Journal*, vol. 2000 (75), Specialized Systems Consultants, Inc., 2000.
- [5] B. Japenga, "Why Use Linux for Real-Time Embedded Systems" White paper. [Online]. Available: <http://www.microtoolsinc.com/Articles/Why%20Use%20Embedded%20Linux%20for%20Real%20Time%20Embedded%20Systems%20Rev%20A.pdf>
- [6] A. A. Fröhlich, G. Gracioli, and J. F. Santos, "Periodic timers revisited: The real-time embedded system perspective," *Computers & Electrical Engineering*, vol. 37, no. 3, 365-375, May 2011.
- [7] K. G. Shin, "Real-time dynamic voltage scaling for low-power embedded operating system," *Proc. of the 8th ACM Symposium on Operating Systems Principles*, New York, USA, pp. 89-102, 2001, doi: 10.1145/502034.502044
- [8] A. Pásztor and D. Veitch, "PC based precision timing without GPS," *The 2002 ACM SIGMETRICS international conference on Measurement and modeling of systems*, Marina Del Rey California, USA, vol. 30, no. 1, pp. 1-10, June, 2002, doi: 10.1145/511334.511336.
- [9] A. Aust, J. Brocke, F. Glaeser, R. Koehler, S. Kubsch, and E. Siemens, "Method for processing time values in a computer or programmable machine," *US Patent 8,185,770*, 2012.
- [10] I. Fedotova, E. Siemens, H. Hu. "A high-precision Time Handling Library," *Proc. of the 9th International Conference on Networking and Services*, (ICNS 2013), Lisbon, pp. 193-199, March 2013.
- [11] AM335x ARM Cortex-A8 Microprocessors, *Technical Reference Manual* p. 5.2.5. [Online]. Available: https://s3-us-west-1.amazonaws.com/123d-circuits-datasheets/uploads%2F1378501288286-gibpl1belakmx6r-2561e976ef65a4ecf67b3a3ba2590088_%2FAM335x_ARM_Cortex-A8%28spruh73h%29.pdf

Efficiency Comparison of DFT/IDFT Algorithms by Evaluating Diverse Hardware Implementations, Parallelization Prospects and Possible Improvements

Danijela Efnusheva, Natasha Tagasovska, Aristotel Tentov, Marija Kalendar
SS. Cyril and Methodius University - Faculty of Electrical Engineering and Information Technologies
Rugjer Boshkovik bb, PO Box 574, 1000 Skopje, Macedonia
E-mail: {danijela,ntagasovska,toto,marijaka}@feit.ukim.edu.mk

Abstract—In this paper we investigate various algorithms for performing Fast Fourier Transformation (FFT)/Inverse Fast Fourier Transformation (IFFT), and proper techniques for maximizing the FFT/IFFT execution speed, such as pipelining or parallel processing, and use of memory structures with pre-computed values (look up tables - LUT) or other dedicated hardware components (usually multipliers). Furthermore, we discuss the optimal hardware architectures that best apply to various FFT/IFFT algorithms, along with their abilities to exploit parallel processing with minimal data dependences of the FFT/IFFT calculations. An interesting approach that is also considered in this paper is the application of the integrated processing-in-memory Intelligent RAM (IRAM) chip to high speed FFT/IFFT computing. The results of the assessment study emphasize that the execution speed of the FFT/IFFT algorithms is tightly connected to the capabilities of the FFT/IFFT hardware to support the provided parallelism of the given algorithm. Therefore, we suggest that the basic Discrete Fourier Transform (DFT)/Inverse Discrete Fourier Transform (IDFT) can also provide high performances, by utilizing a specialized FFT/IFFT hardware architecture that can exploit the provided parallelism of the DFT/IDF operations. The proposed improvements include simplified multiplications over symbols given in polar coordinate system, using sine and cosine look up tables, and an approach for performing parallel addition of N input symbols.

Keywords: Cooley-Tukey, DFT/IDFT, FFT/IFFT, Intelligent RAM, look up tables, OFDM, pipeline and parallel processing, polar coordinate system.

I. INTRODUCTION

The Fast Fourier Transform and the Inverse Fast Fourier Transform are widely used efficient and fast techniques for computing the Discrete Fourier Transform and the Inverse Discrete Fourier Transform, respectively. The reduced number of FFT/IFFT calculations required for performing the same set of DFT/IDFT objectives, results in decreasing the execution complexity from $O(N^2)$ to $O(N\log_2 N)$, on algorithmic level [1][2]. Consequently, FFT/IFFT modules are extensively used for analysis and implementations of communication systems with real-time data transmission requirements. Additionally, the recent advances in chip production technologies, as well as in FFT/IFFT algorithms, have resulted with production of several FFT/IFFT chips

that allow significant processing speed up (multi Gb/s) [3]-[6], as well as decreased chip area and reduced energy consumption [7]-[9].

The most important and used FFT/IFFT application is the orthogonal frequency division multiplexing (OFDM), which is the dominant transmission technique used in the 802.11 set of WLAN standards [10]. This advanced modulation technique divides the available spectrum into many overlapping subcarriers, thus allowing more effective channel utilization and reducing inter-symbol interference (ISI) and inter-carrier interference (ICI) caused by multi-path effect [11].

IFFT/FFT modules execute the main functionality in OFDM systems on sending/receiving side, allowing signals to be converted from frequency/time domain to time/frequency domain [2]. Actually, the process of modulation of the subcarriers in the channel with symbol information and making them orthogonal to each other is performed by means of IFFT on the sending side, whereas the FFT is used for efficient demodulation of the received signal. By including the IFFT and FFT modules in OFDM systems, the signal processing complexity is reduced (at both, transmitting and receiving side) and higher transmission rates are achieved.

Efficient FFT/IFFT implementation is a topic of continuous research in the recent years [3][4], and the main goal is reducing the processing complexity of the FFT/IFFT calculations. In general, there are two directions in this area of research. The first one refers to developing algorithms for FFT/IFFT and their optimization. Best known algorithms in this field, worth mentioning in this paper are radix-2, radix-4, radix-8 and split-radix variations of the Cooley-Tukey (C-T) algorithm [12][13] as well as Winograd algorithm [14]. By increasing the base in C-T algorithm, the number of operations is decreased, resulting in FFT/IFFT split-radix implementation becoming superior compared to Winograd algorithm [15]. The second approach is focused on hardware architecture improvements and optimization of the FFT/IFFT module. This includes adequate techniques for parallel processing and pipelining, memory structures for preservation of previously calculated results, as well as

dedicated components (usually multipliers) for more rapid calculations [16][17].

The FFT/IFFT implementation, in order to obtain better performances, can be achieved with various hardware components, including digital signal processors (DSPs), general purpose processors (GPPs), smart memories (IRAM), application specific integrated circuits (ASICs) or specialized circuits implemented on FPGA. The research has shown that GPPs and DSPs [14] are programmable and flexible, but cannot completely satisfy the fast processing requirements. On the other hand, the IRAM architecture is insufficiently explored, but considering its abilities to provide high memory bandwidth and strided memory accesses, it is expected to provide promising results [18]. Most of the FFT/IFFT implementations are realized as specialized logic circuits, characterized with different forms of parallel computing. FPGA components are utterly suitable for implementation of this type of circuits, providing a tradeoff between speed, cost, flexibility and programmability [2][6][15][16][19].

The aim of this paper is to investigate various algorithms for FFT/IFFT computation and to discuss their hardware implementation that mostly satisfies the high speed processing requirements. Actually we talk about several parallelization techniques and methods, emphasizing the problem of data dependences in the FFT/IFFT calculations that limit the processing speed and the maximal utilization of the available hardware resources. Therefore, we suggest that the basic DFT/IDFT calculations are very suitable for parallelization, which can be further improved by implementing several optimizations of the addition and multiplication operations over complex numbers. In order to be effective, the provided modifications should be supported by a specialized processor-in-memory architecture. Initial ideas for implementing such approach, are presented in this paper.

This paper is organized in five sections. Section two discusses variety of FFT/IFFT algorithms for fast calculation of the DFT/IDFT and compares their efficiency. Section three provides an overview of different FFT/IFFT hardware implementations, discussing the achievable speed up by introducing parallelism. Section four presents several techniques for efficient hardware implementation of the basic DFT/IDFT computations, as well as possible DFT/IDFT improvements of the multiplication and addition operations, which are essential for performing the summation of products in the DFT/IDFT. The proposed improvements should involve maximal parallelism, during the execution of the DFT/IDFT computations. The paper ends with a conclusion, stated in section five.

II. SOFTWARE OPTIMIZATION OF DISCRETE FOURIER TRANSFORM

The DFT/IDFT is the most important discrete transform, used to perform Fourier analysis in many practical applications. For example, it has a fundamental function for modulation and demodulation in OFDM systems [2]. This transformation deals with a finite discrete-time signal and a finite or discrete number of frequencies, so it can be implemented in computers by numerical algorithms or even -vector dedicated hardware.

Given n real or complex inputs x_0, \dots, x_{n-1} , the DFT [2] [20], is defined as:

$$y_k = \sum_{0 \leq \ell < n} x_\ell \omega_n^{-k\ell}, \quad 0 \leq k < n, \quad (1)$$

with $\omega_n = \exp(-2\pi i / n)$ and $i = \sqrt{-1}$. Stacking y_k and x_ℓ into vectors $x = (x_0, \dots, x_{n-1})^T$ and $y = (y_0, \dots, y_{n-1})^T$ yields into the equivalent form of a matrix-vector product:

$$y = DFT_n x, \quad DFT_n = [\omega_n^{kl}], \quad 0 \leq k, l < n \quad (2)$$

Straightforward computation of both DFT and convolution is a matrix-vector multiplication, given as $W \bullet \vec{g}$ and takes $O(N^2)$ operations, for N being a transformation size.

The breakthrough of the Cooley-Tukey algorithm's family derives from its capability of significantly cutting down DFT's $O(N^2)$ complexity to an order of $O(N \log_2 N)$, [13]. This advance in computational theory inspired and motivated a stream of researches targeting even further speed up and efficiency of performing DFT, and eventually a whole new class of algorithms was introduced, known as FFT algorithms. A common feature for most FFT algorithms is their order of complexity - $O(N \log_2 N)$.

A. Divide and Conquer Approach as a Basis for FFT Algorithms

The DFT usually arises as an approximation to the continuous Fourier transform, allowing functions sampling at discrete intervals in space or time. In order to make the DFT operation more practical, several FFT algorithms were proposed. The fundamental approach for all of them is to make use of the properties of the DFT operation itself, and thus reduce the computational cost of performing the DFT calculations. This is basically achieved by implementing the divide and conquer approach [21], which is a basis for most of the algorithms for effective computation of DFT.

We already stated that the discrete Fourier transform is a matrix product, whereas x_0, \dots, x_{N-1} is the vector of input samples, $x = (X_0, \dots, X_{N-1})^T$ is the vector of transform values and W_N is the primitive N^{th} root of unity, so that $W_N = \exp(-2\pi i / N), i = \sqrt{-1}$ [20]. This product is given by the following equation:

$$\begin{bmatrix} X_0 \\ X_1 \\ X_2 \\ X_3 \\ \vdots \\ X_{N-1} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & \dots & 1 \\ 1 & W_N & W_N^2 & W_N^3 & \dots & W_N^{N-1} \\ 1 & W_N^2 & W_N^4 & W_N^6 & \dots & W_N^{N(N-1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & W_N^{N-1} & W_N^{2(N-1)} & \dots & \dots & W_N^{(N-1)(N-1)} \end{bmatrix} \times \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{N-1} \end{bmatrix} \quad (3)$$

W_N is also referred to as the *twiddle factor* or *phase factor*. This value being a trigonometric function over discrete points around the 4 quadrants of a two dimensional plane, has symmetric and periodic properties [22]. Using these properties of the twiddle factor, unnecessary computations of the DFT can be easily eliminated.

The direct evaluation of the matrix-vector product requires N^2 complex multiplications and additions. In order to reduce this huge number of operations, a divide and conquer approach is implemented. The general idea of this methodology is to map the original problem into several sub-problems in such a way that the following inequality, [20], is assured:

$$\text{cost}(\text{subproblems}) + \text{cost}(\text{mapping}) < \text{cost}(\text{original problem}) \quad (4)$$

The real influence of this method is that, regularly, the division can be applied recursively to the sub-problems as well, thus leading to a reduction of the order of complexity.

The important point in (4) is that the divide and conquer scheme is consisted of two clear costs: the cost of the mapping (which can be zero when looking at the number of operations only) and the cost of the sub-problems. As a consequence, different types of divide and conquer methods make an effort to find balancing schemes between the mapping and the sub-problem costs [20]. As an example, the Cooley-Tukey radix-2 algorithm can be considered, where the sub-problems results being quite trivial (only sum and differences), although the mapping requires twiddle factors that lead to a large number of multiplications. On the contrary, in the prime factor algorithm, the mapping is done only by means of permutations (no arithmetic operation are required), while the small DFTs that appear as sub-problems indicate substantial costs since their lengths are co-prime numbers.

B. Families of FFT Algorithms

There are two core families of FFT algorithms: the Cooley-Tukey algorithms and the Prime Factor algorithms, [20]. These classes of algorithms differ in the way they translate the full FFT into smaller sub-transforms. Mostly two types of routines for Cooley-Tukey algorithms are used: mixed-radix (general-N) algorithms and radix-2 (power of 2) algorithms. All Radix algorithms are similar in structure, differing only in the core computation of the butterflies, [23]. Each type of algorithm can be further categorized according to additional features, such as whether it operates in-place or requires an additional scrape space, whether its output is in a sorted or scrambled order, and whether it uses decimation-in-time or -frequency iterations.

1) Cooley-Tukey Algorithms

The Cooley-Tukey set of algorithms comprises the most common fast Fourier transform (FFT) algorithms. They re-expresses the discrete Fourier transform (DFT) of an arbitrary composite size $N = N_1 N_2$ in terms of smaller DFTs of sizes N_1 and N_2 , recursively, in order to reduce the computation time to $O(N \log_2 N)$ for highly-composite N (smooth numbers). Usually, either N_1 or N_2 is a small factor (not necessarily prime), called the radix (which can differ between stages of the recursion) [20]-[23]. If N_1 is the radix, the Cooley-Tukey algorithm is called decimation in time (DIT), whereas if N_2 is the radix, it is decimation in frequency (DIF).

A radix-2 DIT FFT is the simplest and most common form of the Cooley-Tukey algorithm, although highly optimized Cooley-Tukey implementations generally use

some other forms of the algorithm. Radix-2 DIT decomposes a DFT of size N into two interleaved DFTs (hence the name "radix-2") of size $N/2$ with each recursive stage, eventually resulting in a combining stage containing only size-2 DFTs called "butterfly" [13], operations (so-called because of the shape of the data-flow diagrams, Fig. 1).

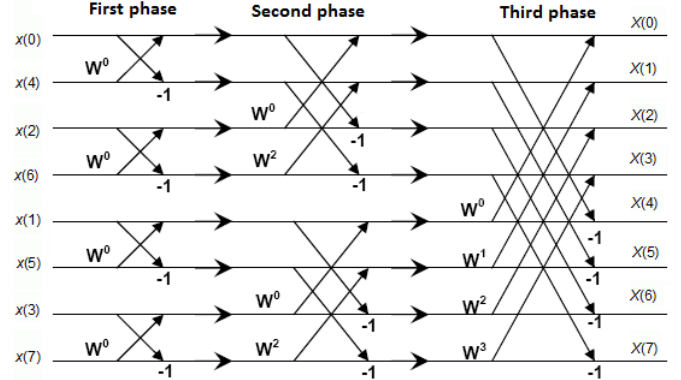


Fig. 1. Flow graph of 8-point DIT FFT radix-2 Cooley-Tukey algorithm.

Besides the radix-2 Cooley-Tukey algorithm, other implementations with radices of 4 and 8 are also used. Actually, the value of the radix (2, 4, 8) indicates that the total number of points used for the transformation can be expressed as 2^x , 4^x or 8^x , accordingly [15]. Therefore, the C-T algorithm can execute parallel and independent butterfly operations with 2, 4 or 8 input/output values, in each of the algorithm phases (for radix x , the number of phases is $\log_x N$).

Mixed-radix (also called split-radix) algorithms work by factorizing the data vector into smaller lengths. These can then be transformed by FFTs with small number of points, noted as small-N FFT [24]. Typical programs include FFTs for small prime factors, such as 2, 3 or 5, which are highly optimized. Actually, the idea of this algorithm is to use many multiplied small-N FFT modules and combine them in order to make longer transforms. If the small-N modules are supplemented by an $O(N^2)$ general-N module then an FFT of any length can be computed. Of course, any lengths which contain large prime factors would perform only as $O(N^2)$.

The well-known radix-2 Cooley-Tukey algorithm is a simplified version of the mixed-radix algorithm, realized by the use of FFT modules whose lengths are only power of two. Radix-2 algorithms have been the subject of much research into optimizing the FFT. Many of the most efficient radix-2 routines are based on the "split-radix" algorithm [15]. This is actually a hybrid which combines the best parts of both radix-2 ("power of 2") and radix-4 ("power of 4") algorithms, for computing distinctive partitions of the Fourier's transformation.

2) Prime Factor Algorithms

The prime-factor algorithm (PFA), (also known as the Good-Thomas algorithm), is a FFT algorithm that re-expresses the DFT of a size $N = N_1 N_2$ as a two-dimensional $N_1 \times N_2$ DFT, but only for the case where N_1 and N_2 are relatively prime. These smaller transforms of size N_1 and N_2 can then be evaluated by applying PFA recursively or by using some other FFT algorithm. Although this algorithm has a very simple indexing scheme, it only works in the case where all factors are co-prime, which makes it suitable as a

specialized algorithm for given lengths, [25]. PFA is also closely related to the nested Winograd FFT algorithm, where the latter performs the decomposed N_1 by N_2 transform via more sophisticated two-dimensional convolution techniques [26]. Winograd algorithm is requiring the least known number of multiplications among practical algorithms for moderate lengths DFTs.

Another less known algorithm is Fast Hartley Transform (FHT) [27]. This effective algorithm cannot be classified in formally presented families, since its core characteristic is what moves them apart. For the DFHT, the kernel is real, unlike the complex exponential kernel of the DFT. For complex data, each complex multiplication in the summation requires four real multiplications and two real additions using the DFT. For the DFHT, this computation involves only two real multiplications and one real addition.

It can be noticed that performing efficient and fast discrete Fourier transform, truly inspired researchers and burst many peoples' creativity. Sequentially, along with many ideas, came many experiments, evaluations and analysis. An effort was made to sum up the general conclusions in terms of complexity (meaning effectiveness) of the different types of FFT algorithms. Table 1 and Table 2 present the results.

From the results gathered in Table 1 and Table 2, an assumption can be made, that in general, split-radix algorithm achieves best performance. However, its irregular structure introduces some difficulties during the implementation [15]. Observing Table 1, another thing can be noticed, and that is the obvious decrement of the number of operations required for FFT, with each increment of algorithms' radix. Regarding the Prime Factor Algorithms, from Table 2 it can be concluded that Winograd is more efficient for smaller FFT sizes while for FFTs with input array sizes greater than 500 points, Prime Factor Algorithms present better results. Thus, for this family of algorithms a general winner or leading algorithm cannot be determined.

At this stage, we conclude FFT algorithms brief overview and efficiency evaluation, considering different manners of influencing DFT performances' only through algorithms, i.e. optimization in software. Nevertheless, these influences' impact depends on the characteristics of the platform the algorithms are executed on. Thus, this paper elaborates the FFT algorithms possible hardware implementations in section 3, as well.

C. FFT Parallelization and Optimization in Software

The emergence of multi-core processors also encouraged further research and optimization of FFT algorithms. The most popular of the previously discussed algorithms were

rewritten introducing multi-threaded FFT programs. However, well known rule of thumb is that, the more the code is optimized, the harder it is to be realized in parallel. Most researches in this direction go with sequentially executing code until a parallel region is reached where multiple threads can be employed, [28]. This practice is truly more a need, since FFT algorithms' separate stages depend on each other's results.

An accent was also put on joint resources, like twiddle factors and bit reversal mapping which are commonly presented as lookup tables (LUT) [28][29]. But then again, memory may be saved, but usually the case is that memory accesses are far more expensive (in terms of speed of execution) than arithmetic operations, due to slower memories. The consequence of this type of implementation may not be the performance one hoped for. This particular fact was the starting point for the FFTW project, [30], where an effort was made for maximum utilization of computers' fast memory (cache and RAM), by introducing self-optimizing FFT algorithms on the behalf of specialized compiler, that adapt themselves in the form most suitable for particular architecture implementation.

III. HARDWARE OPTIMIZATION OF FFT ALGORITHMS IMPLEMENTATIONS

Efficiency comparison on all previously discussed algorithms makes no real sense if they are not associated with their hardware application. By hardware, we mean certain architecture and particular platform. It was proven that the desired performance requirements will only be met if FFT algorithms are applied into suitable hardware set. Architectures vary in number of available processing units as well as in memory size and distribution. These two main features have inevitable impact on FFT algorithms efficiency, since they cover parts of FFT which proved to be the most intense – arithmetic operations and memory accesses.

A. Various Platforms for FFT Algorithms Hardware Implementations

Several architectures have been proposed [31][32], each of them trying to optimize the load of memory accesses and increase overall speed of FFT execution. These architectures effectiveness can be further evaluated by physical implementation. Today's cutting edge technology provides number of hardware platforms fit for deploying FFT algorithms. In this paper GPP, DSP as well as specialized hardware circuits – ASIC and FPGAs implementations are discussed.

TABLE I
COMPARISON OF MULTIPLICATION AND ADDITION COMPLEXITY OF DFT AND COOLEY-TUKEY BASED FFT/IFFT ALGORITHMS

Points	DFT			C-T Radix-2			C-T Radix-4			C-T Radix-8			C-T Split Radix		
	Real Adds	Real Muls	Total	Real Adds	Real Muls	Total	Real Adds	Real Muls	Total	Real Adds	Real Muls	Total	Real Adds	Real Muls	Total
16	992	1024	2016	152	24	176	148	20	168				148	20	168
32	4032	4096	8128	408	88	496							388	68	456
64	16256	16384	32640	1032	264	1296	976	208	1184	972	204	1176			
128	65280	65536	130816	2504	712	3216							2308	516	2824
256	261632	262144	523776	5896	1800	7696	5488	1392	6880				5380	1284	6664
512	1047552	1048576	2096128	13576	4360	17936				12420	3204	15624	12292	3076	15368
1024	4192256	4194304	8386560	30728	10248	40976	28336	7856	36192				27652	7172	34824

TABLE II
COMPARISON OF MULTIPLICATION AND ADDITION COMPLEXITY OF DFT AND PRIME FACTOR BASED FFT/IFFT ALGORITHMS

Points	DFT			Prime Factor			Winograd		
	Real Adds	Real Muls	Total	Real Adds	Real Muls	Total	Real Adds	Real Muls	Total
60	14280	14400	28680	888	200	1088	888	136	1024
240	229920	230400	460320	4812	1100	5912	5016	632	5648
504	1015056	1016064	2031120	13388	2524	15912	14540	1572	16112
1008	4062240	4064256	8126496	29548	5804	35352	34668	3548	38216

Architectural and design issues which are to be considered for diverse FFT hardware implementations are: precision of data, number of points for FFT size, and memory usage. Then performance evaluation is done according to power consumption, required circuit board area and desired circuit frequency. The value of these variables depends on the application which makes use of FFT i.e. employs it, and what's even more is that these same parameters have impact on the whole system (FFT algorithm + platform) performances.

1) General Purpose Processors

Many research efforts have been performed in this area [30][33] being just part of it. What is interesting to notice is that a greater part of the execution time was spent on load and store operations, compared to actual arithmetic computations [20]. This is a straightforward consequence of the GPP architecture, designed to satisfy many diverse applications' needs. Thus, from all implementations, this has proven to be the least effective one. Nevertheless researches utilizing these platforms continue, since many upper layer applications, using everyday technology (consisting GPPs) require FFT. Hence, a great advancement is made with the FFTWs' development, which provides an opportunity for one to choose an optimal algorithm for particular GPP, due to its adaptive features, by benchmarking FFTW library. From the gathered results it is obvious that for different FFT size, different platform is more convenient.

2) Digital Signal Processors

DSPs offer the best flexibility, but limited performance. These processors sturdily service multiply/accumulate based algorithms. Unluckily, this is not the case of any FFT algorithm (where sums of products have been changed to fewer but less regular computations) [20]. Still, DSPs now meet some of the FFT requirements, like modulo counters and bit-reversed addressing. If the modulo counter is general, it will help the implementation of all FFT algorithms, but it is often restricted to the Cooley-Tukey/SRFFT case only (modulo a power of 2), for which proficient timings are provided on nearly all available machines by manufacturers, at slightest for small to medium lengths. DSPs achieve low development cost compared to dedicated hardware solutions, although at the expense of medium performance and high power consumption.

3) Application Specific Integrated Circuit

Application Specific Integrated Circuits are famous for being the fastest platform for processing intense operations. Many hardware product vendors like Xilinx and Altera accepted this appropriateness of using ASIC chips for FFT implementations, and have developed high performance IP cores, from which designers benefit in reducing the time

required for various product development. Compared to GPU and DSP they provide flexibility in terms of algorithms, meaning they are convenient not only for traditional Cooley-Tukey algorithms but also for PFA, [34]. Another reason to use them is their power efficiency requirements which are lowest of all.

4) Field Programmable Gateway Arrays (FPGA)

Fast Fourier transform has been playing an important role in digital signal processing and wireless communication systems, so the choice of FFT sizes is decided by different operation standards. It is anticipated to make the FFT size configurable according to the operation environment, since achieving a successful design means the system should be able to support different operating modes required by diverse applications with low power consumption requirement [35]. This is the reason why reconfigurable hardware has been paid more attention recently. FPGAs combine ASICs desired speed of operation with the flexibility provided from DSP, resulting in a perfect match for FFT implementations. Additionally, the capability of changing the source code according to current specific application makes this type of platform convincing winner for FFT implementation. Despite of all the upsides, a disadvantage must be mentioned, and that its power consumption.

Further, this paper presents another not so explored platform, but from our point of view, suitable for FFT implementation, and that is Intelligent RAM-IRAM.

5) Intelligent RAM

Intelligent RAM [18], is another merged DRAM-logic processor, designed at the Berkeley University of California by a small Patterson's team of students. This chip was designed to serve as multimedia processor on embedded devices. As a result of the design studies of the Berkeley's team research group, it was shown that most applicable architecture to multimedia processing is vector architecture, rather than MIND, VLIW, and other ILP organizations [36]. This is basically because of the vector processors' abilities to deliver high performance for data-parallel tasks execution, and to provide low energy consumption, simplicity of implementation and scalability with the modern CMOS technology.

The resulted IRAM chip is called Vector IRAM (VIRAM). VIRAM is processor that couples on-chip DRAM for high bandwidth with vector processing to express fine-grained data parallelism [18]. The VIRAM architecture (shown on Fig. 2) consists of MIPS GPP core, attached to a vector register file that is connected to an external I/O network and also to a 12MB DRAM memory organized in 8 banks.

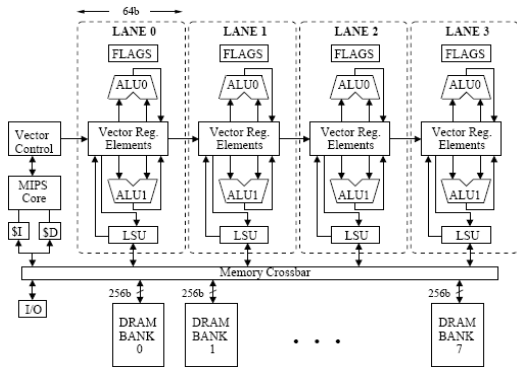


Fig. 2. Architecture of IRAM chip.

FFT computations are difficult to be realized with conventional GPP architectures, given that they require high memory bandwidth and strided memory accesses. Nevertheless, it was shown that the performance of computing floating-point FFTs on a VIRAM processor, [18], can be competitive with that of existing floating point DSPs and specialized fixed-point FFT chips.

From this section it can be concluded that there is no such thing as a universally optimal FFT/IFFT algorithm, architecture and implementation that is appropriate for all systems. Therefore, it is recommended to perform a search across the algorithm, architecture and implementation dimensions for each system.

B. Further FFT Speedup by Introducing Parallelization Techniques in Hardware

Traditional FFT hardware architectures include trade-offs among complexity, power consumption, die size, and other similar parameters. Still, these architectures don't have the scalability to encounter high speed processing requirements on FFT performance, for instance on emerging high data rate technologies in wireless communications, like OFDM. Progress in technology empowers development of architectures optimal for given class of algorithms, including FFT.

In general, there are three main techniques of improving overall FFT speed, all of them based on the Transpose algorithm, described in [37]:

1) Increasing the order of the radix.

This approach improves both latency and throughput, but is expensive in terms of computational resource required for each processing unit.

2) Cascading (pipelining) the processors.

Different processors operate over different stages. Cascading improves throughput, but not latency. Inter stage memory is necessary.

3) Parallelizing the processors.

Parallelizing the processors, so that a single large FFT is divided into N smaller FFTs. Both latency and throughput are improved with this arrangement.

Drawback is the data transfer between stages.

All these improvement methods require architecture modifications. Thus, in literature, commonly two architecture approaches are presented: memory-based FFT and pipeline-based FFT. Memory-based FFT uses only one butterfly and large memories for data storage. On the other hand, pipelined architecture uses many butterflies to improve the speed. There are several FFT hardware modules

that have been proposed to implement both methods (memory-based design and pipelined design) in order to improve speed and minimize area. For example, [38] proposes six memory-based implementations, realized in FPGA. The authors give the name of RX2-B1, RX4-B1, RX2-B2, RX4-B2, RX2-B4, MXRX-B4 for these architectures, whereas the RX presents the used radix and Bi indicates that the architecture operates with i outputs in parallel. The techniques used for each of these architectures are based on memory sharing, conflict free memory addressing, in-place memory processing, continuous flow design, N-word memory size, fixed-point arithmetic and pre-computed twiddle factors stored in ROM. It was found that the fastest processors are the RX2-B4 and MXRX-B4 FFT models which can process four data samples per clock cycle.

Another direction in researching memory based architecture is towards distributed memory and processing units. This technique requires re-expressing the FFT algorithms to adapt them for parallel implementation. According to [21][39], where several algorithms were tested, it was proven that additional complexity is added to the system, as a consequence of the need for communication between different modules. Also, it was noticed that not all algorithms are suitable for parallelization, since only algorithms with regular structure (Cooley-Tukey Radix-2/4, Split Radix) are advisable, while the block complexity of Winograd algorithm makes it difficult and unsuited for parallel implementation.

Regarding the pipelined architectures, many works have presented optimizations to achieve high performance and low area consumption. The most famous architectures of pipeline implementations are multi-path delay commutator, radix-2 single path delay commutator and radix-2 single path delay feedback, [35]. The main difference between these architectures is in the number of inputs, outputs and butterflies used.

A different approach is applied by [40] proposing an array processing architecture for parallel FFT implementation. Key advantage of this architecture is the elimination of inter stage data transfer, so both, the system throughput and latency are improved. This idea is basically similar to IRAM, [18], which also includes vector processing units, capable to process the input data in parallel.

Although there have been several efficient designs, there is an inherent drawback related to the area (and consequently power) overhead. Another essential conclusion is the existing threshold, which delimits FFTs' input size after which this whole additional complication of FFT implementation system is rewarding, so this fact must also be taken into account while developing applications.

IV. HARDWARE OPTIMIZATION OF DISCRETE FOURIER TRANSFORM

The Discrete Fourier Transform is a universal instrument in science and engineering, including digital signal processing, communication, and high-performance computing. Applications employing it belong into the area of spectral analysis, image compression, interpolation, solving partial differential equations, and many other tasks,

[13]. Therefore, as a consequence from its extensive implementation, meeting its computational demands, is the high price applications employing DFT, have to pay.

As stated before, the Discrete Fourier Transform for an N-point sequence $x(n)$ is given by (4), where $X(k)$ and $x(n)$ are complex numbers, n is the time index and k is the frequency index [1]. Actually, the DFT formula is expressed as a summation of complex numbers products.

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-j2\pi nk/N}, \quad k = 0, 1, \dots, N-1 \quad (4)$$

By using this equation the DFT computation requires N^2 complex multiplications and $N(N-1)$ complex additions, leading to a complexity of $O(N^2)$. Each complex multiplication requires four real multiplications and two real additions and each complex addition requires two real additions. Therefore a total of $4N^2$ real multiplications and $N(4N-2)$ real additions are required [1]. Although the DFT employs the highest number of operations, comparing to various FFT implementations, all the operations are independent and can be executed in parallel. This is where we see a chance for maximal parallelization that will allow reaching better and faster results.

Generally, the FFT algorithms decrease the number of operations, but involve significant communication overhead, during the execution. This is a result of the divide and conquer approach which halts the system before proceeding to the next stage until all the output from the previous stage is generated [21]. For example, the execution of the Cooley-Tukey radix-2 algorithm is performed in $\log_2 N$ phases that must execute sequentially, although each phase employs parallel operations. The data dependences that appear between separate stages of the 8-Point Cooley-Tukey FFT radix-2 algorithm and DFT are shown in Fig. 3.

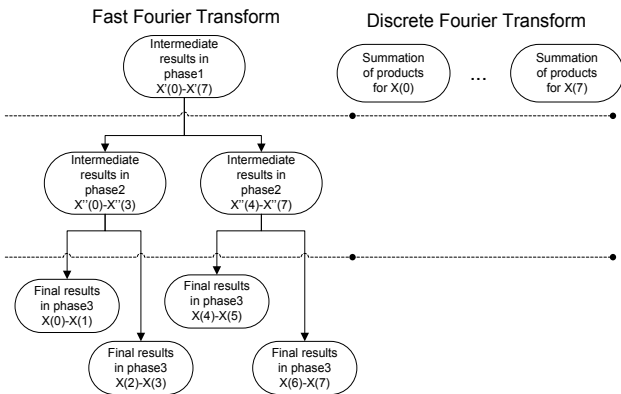


Fig. 3. Data dependence when executing 8-point radix-2 Cooley-Tukey FFT algorithm and DFT.

As can be seen on Fig. 3, the DFT is computed in only one phase. According to the fact that the DFT summations are computed simultaneously, it is of special interest to implement the DFT component on a hardware architecture that can support the provided parallelism (for example IRAM). Furthermore, the computation complexity of the DFT can be significantly reduced by utilizing some techniques that can simplify the multiplications of the inner products and/or parallelize the summations of different outputs.

A. Techniques for efficient hardware implementation of DFT

The strong interaction between the algorithm and its implementation indicates that the algorithm's execution efficiency depends on the hardware architecture and its abilities to support the provided parallelism of the algorithm. Considering the discussion given in the previous section, serious candidates for implementing the DFT are specific application integrated circuits or special purpose processors (similar to IRAM), implemented in FPGA platforms.

Most of the research done so far refers to efficient hardware implementations of FFT, by implementing various parallelization prospects and specialized components. In this section we are discussing several techniques (some of them already used for FFT hardware implementation) that can be applied when implementing the basic DFT computations in hardware. We suggest that the hardware architecture of the DFT module should be specially purposed to exploit the provided parallelism, when executing summation of products.

Assuming that each multiplication is consisted of several additions, it is obvious that the execution of multiplication operations is more complex than additions. Therefore, when implementing the products of the DFT summations many researchers have used specialized multipliers, purposed to improve the execution efficiency of multiply operations. Such approach is presented in [17], where the authors suggest the use of constant co-efficient multiplier. This multiplier utilizes small blocks of ROM that store 16 partial products relating to the fixed coefficient and then use a simple adder to combine these products. As a result, the co-efficient multiplier is less than one third of the size of full multipliers. Another multiplier that is also very efficient, (without using ROM memories) is the shift-and-add multiplier [8]. This multiplier decomposes the product in such a way that it requires only additions and multiplications or divisions by 2, which are realized in a computer by shifting bits left or right.

A different approach is presented in [41], where low power and area efficient adder and Vedic multiplier for computing FFT are presented. Generally, the speed of addition is limited by the time required for carry propagation through the adder, since the sum for each bit position in an elementary adder is generated after the previous bit position has been summed and a carry has propagated into the next position. Referring to [41] the carry select adder allows fastest execution of arithmetic operations by independently generating multiple carries and then selecting a carry to generate the final sum. Although this adder is not area efficient, its ability to decrease the propagation delay makes it popular for use. The combination of the carry select adder and Vedic multiplier provides fast execution and also reduction of FFT chip area and power [41]. The Vedic multiplier is based on Vedic mathematic principles, which employs a unique technique of calculations based on sixteen principles or word-formulae which are termed as Sutras. This mathematic is very suitable for digital signal processing [42].

Very efficient technique for calculation of DFT inner products is Distributed Arithmetic (DA) [43]. This method is used only if one of the vector operands is known,

allowing bit-serial computation that forms an inner product of a pair of vectors in a single direct step. The distributed arithmetic approach replaces the explicit multipliers by ROM look up tables and adders, thus providing considerable reduction in the power consumption. Although this technique appears to be slow because of its bit serial nature, it turns out that when the number of samples in each vector commensurate with the number of bits in each vector element, the DA is very fast. Another similar approach that utilizes look-up tables for pre-computing the products of the DFT summations is presented in [5]. It was shown that such DTF module can be applied to OFDM transmitter for optical communication, which is capable to achieve 101.5 Gb/s through fiber with 16QAM modulated sub carriers, by using two Xilinx Virtex 5 FPGA boards that work in parallel.

The speed of the DFT calculations can be also improved by utilizing the CORDIC (coordinate rotation digital computer) algorithm, which dynamically calculates the trigonometric functions, using processing elements to perform vector rotations [44]. The CORDIC component can also be used for polar to rectangular or rectangular to polar conversions, and as well for computing a vector magnitude. This algorithm operates in such a way that it executes vector rotations by arbitrary angles, using only shift and add operations, thus providing advantages in speed, accuracy, simplicity in design and other aspects of performance requirements.

B. Proposed improvement for efficient hardware implementation of DFT

The total processing time required for computing the DFT, depends on the multiplication time, which is generally far greater than the addition time. The multiplications and additions of the DFT are performed over complex numbers [7], which are usually represented in rectangular format:

$$(a+bj) \cdot (c+dj) = (ac-bd) + (ad+bc)j \quad (5)$$

$$(a+bj) + (c+dj) = (a+c) + (b+d)j \quad (6)$$

As given in equations (4) and (5) each multiplication of two complex numbers requires four multiplications of real numbers (imaginary and real component) and two additions of real numbers. On the other hand, each addition of two complex numbers requires two additions of real numbers. From here, it is obvious that multiplications are more complex than the additions.

There is an approach [13], which reduces the multiplication complexity to 3 multiplications and five additions. This is achieved by regrouping the multiplicands, as presented in equation (6):

$$(a+bj) \cdot (c+dj) = [(a+b)c - b(d+c)] + j[(a+b)c + a(d-c)] \quad (7)$$

Representing complex numbers in rectangular format is a common choice, basically because the arithmetic operations, such as: add, sub, mul and div can be easily implemented. On the other hand, having complex numbers represented in circular form (polar coordinates) can cause many difficulties, while performing adds/subs, but allows simpler implementation of muls/divs operations. Taking all this into

account, in continuation we present a method for simplified implementation of the DFT/IDFT calculations.

The purpose of the DFT module is to perform the summation of products, given in equation (4), over input set of complex numbers. In our approach we consider that the DFT/IDFT module inputs and twiddle factor angles are provided in circular format, with polar coordinates (angle ϕ and amplitude r). All the twiddle factors magnitudes are 1s, while the angles are pre-computed and placed in a look up table. This way, the multiplications between complex numbers and twiddle factors are executed with only one add operation. Actually, each product term of the DFT summations is calculated as:

$$r \cdot e^{i\phi} \cdot e^{-\frac{i2\pi mk}{N}} = r \cdot e^{i(\phi - \frac{2\pi mk}{N})}, k = 0, 1, \dots, N-1 \quad (8)$$

The calculation of the products in the DFT is not data dependent, so it can be executed in parallel. In order to compute the appropriate DFT output symbols, subsets of the calculated products, should be summed up. Given that the addition of circular format complex numbers is very complex, a conversion to rectangular format is made:

$$r \cdot e^{i(\phi - \frac{2\pi mk}{N})} = r(\cos(\phi - \frac{2\pi mk}{N}) + i \sin(\phi - \frac{2\pi mk}{N})) \quad (9)$$

The provided conversion involves cosine and sine operations and two multiplications of real numbers. The cos and sin operations can be calculated by utilizing look-up tables with pre-computed cosine and sine values, while the two multiplications can be performed by using only shift and add operations, as the authors of [14], suggested. The conversion is executed in such a way that firstly the *cos* and *sin* tables are looked-up and then the multiplications are performed. The conversions of different products are independent of each other and can be performed simultaneously.

After the real and imaginary components of each product are computed, next is to sum up the results, and produce the output complex numbers. The number of the outputs depends on the number of points, used in the DFT/IDFT module. Each output is generated by performing a sum over N products. In order to parallelize the summation of N inputs, we propose a parallel algorithm for that purpose.

The proposed algorithm performs parallel additions of N inputs with M bits, without having to wait for the carries generated during the additions. In fact, it allows simultaneous additions over each bit position of the numbers. Therefore, the summations of the *i*-th position (*i*=0,...,M-1) bits of each number are performed in parallel and each of the results is placed in a separate register, and shifted by *i* places. If there is a carry, the result register will have one or more 1s on some of the [*i*+1, *i*+M-1] positions. In that case the algorithm is performed again, and the appropriate additions are executed once more. Actually, the algorithm stops only when the generated output of each of the N registers consists only one 1 on the *i*-th position or all

0es. Other terminating condition of the algorithm can be the state when the results stored in all M registers, are completely the same as the results stored in all M registers, but in the previous iteration. In the final step of the algorithm, each of the registers holds the i-th cipher of the result.

The given algorithm was introduced to work with binary numbers, which sometimes can take a long time. In order to reduce the number of algorithms' iterations, one can work with numbers given in hexadecimal format. However this approach will complicate the design, requiring additional adders for performing the intermediate summations of N hexadecimal numbers.

The proposed modifications are made to allow parallel processing during the execution of the operations. However, the given improvements will only be effective if the hardware implementation of the DFT/IDFT module is consisted of many processing engines that can support the provided add and mul operations. The DFT/IDFT module will also include three look-up tables for holding the twiddle factor angles, and for performing the sine and cosine look-ups. Our opinion is that the processor-in-memory architecture can be very suitable solution for implementing the proposed DFT/IDFT improvements.

V. CONCLUSION

The never-ending aspiration for more efficient calculation of DFT, motivated by its truly widespread expansion in applications, could not disregard this opportunity. Although there are different variations of architectures of parallel systems, the number of researches made in this area, as well as the number of diverse tracks and ideas within it, is certainly surprising. Different combinations have been made, starting from FFT algorithms properties or the properties of parallel systems; architectures, further adjusting one to another in the development process.

All the work done in this area proves that by making the right combination of FFT algorithm, architecture and platform, desired performance results can be achieved. In general, all the efficient FFT implementations require specific hardware architecture that is tailored to support the computations involved in the FFT algorithm used.

In this paper we consider that most FFT algorithms include data dependent operations that limit the execution speed of the algorithm. Therefore, we suggest that the execution of the basic DFT/IDFT computations can provide execution speed-up, despite the fact that the DFT/IDFT involves more computations than many FFT algorithms. This is basically because the DFT/IDFT computations are characterized with high level of parallelism, so most of them can be executed simultaneously.

Considering that the basic DFT/IDFT computation is represented as summation of products, we propose possible improvements of the multiply and add operations over complex numbers. In our approach the multiplications involve only one add operation, since the operands are given in polar coordinate system. The additional cost that should be paid for this is the conversion to rectangular form, which involves two multiplications and calculation of sin and cos functions, using look-up tables. Furthermore, we propose an algorithm for performing parallel additions of N inputs that

excludes the timing overhead caused by the addition of carry bits. The proposed algorithm involves high level of parallelism.

The proposed improvements are essential for performing the summation of products in the DFT/IDFT. It is expected that they should involve maximal parallelism, during the calculation of the DFT/IDFT outputs. However, their execution should be supported by a specialized processor architecture that would be able to exploit the provided parallelism. We believe that processor-in-memory architecture is a good solution for efficient implementation of the DFT/IDFT module with the proposed modifications.

Even though many researches dedicated their work on improving DFT calculating performance, we noticed a gap in examining pure DFT prospects for optimization and parallelization, where we recognize a great potential. For future continuing on this work we plan on implementing the proposed design expecting meaningful results.

ACKNOWLEDGEMENT

This work was partially supported by the ERC Starting Independent Researcher Grant VISION (Contract n. 240555).

REFERENCES

- [1] S. W. Smith, "The Scientist and Engineer's Guide to Digital Signal Processing," California: California technical publishing, 1997.
- [2] K. Adzha and B. Kadiran, "Design and implementation of OFDM transmitter and receiver on FPGA hardware," M.S. thesis, Faculty of electrical engineering, Universiti teknologi Malaysia, Malaysia, 2005.
- [3] M. Bernhard and J. Speidel, "Implementation of an IFFT for an optical OFDM transmitter with 12.1 Gbit/s," ITG Symposium on Photonic Networks, Germany, 2010.
- [4] F. Buchali, R. Dischler, A. Klekamp, M. Bernhard, and D. Efinger, "Realisation of a real-time 12.1 Gb/s optical OFDM transmitter and its application in a 10⁹ Gb/s transmission system with coherent reception," 35th European Conference on Optical Communication, Germany, 2009.
- [5] R. Schmogrow and M. Winter, et al, "101.5 Gbit/s real-time OFDM transmitter with 16QAM modulated subcarriers," OSA/OFC/NFOEC 2011, vol. A247, pp. 529-551, April 2011.
- [6] C. Toal and S. Sezer, et al, "A 1Gbps FPGA-based wireless baseband MIMO transceiver," IEEE International SOC Conference (SOCC'12), pp. 202-207, September 2012.
- [7] K. Maharatna, E. Grass, and U. Jagdhold, "A 64-point Fourier transform chip for high-speed wireless LAN application using OFDM," IEEE Journal of Solid-State Circuits, vol. 39, March 2004.
- [8] K. Maharatna, E. Grass, and U. Jagdhold, "A Low-power 64-point FFT/IFFT architecture for wireless broadband communication," in Proc. 7th Int'l Conf. on Mobile Multimedia Communication (MoMuC), Tokyo, Japan, 2000.
- [9] C. Lin, Y. Yu, and L. Van, "A Low-power 64-point FFT/IFFT design for IEEE 802.11a WLAN application," IEEE International Symposium on Circuits and Systems, Greece, 2006.
- [10] M. Bhardwaj, A. Gangwar, and D. Soni, "A review on OFDM: concept, scope & its applications," IOSR Journal of Mechanical and Civil Engineering, Volume 1, Issue 1, pp. 07-11, 2012.
- [11] L. Lit win and M. Pugel, "The principles of OFDM," RF Signal Processing Journal, pp 30-48, 2001.
- [12] T. Fung, "FPGA design and implementation of a memory based mixed-radix 4/2 FFT processor," M.S. thesis, Dept. of Electrical Eng., Tatung University, Tatung, July 2008.
- [13] R. E. Blahut, "Fast Algorithms for Signal Processing," United Kingdom: Cambridge University Press, 2010.
- [14] J. J. Fuster and K. S. Gugel, "Pipelined 64-point fast Fourier transform for programmable logic devices," in Proc. International conference on advances in recent technologies in communication and computing, India, 2010.
- [15] J. Garcia1, J. A. Michell, G. Ruiz, and A.I M. Burón, "FPGA realization of a split radix FFT processor," in Proc. SPIE, Microtechnologies for the New Millennium, vol. 6590, pp. 1-11, 2007.

- [16] D. Ghosh, D. Debnath, and A. Chakrabarti, "FPGA based implementation of FFT processor using different architectures," *International Journal of Advance Innovations, Thoughts & Ideas*, 2012.
- [17] M. Chandan, S. L. Pinjare, and C. Mohan Umaphy, "Optimised FFT design using constant co-efficient multiplier," *International Journal of Emerging Technology and Advanced Engineering*, 2012.
- [18] R. Thomas, "An architectural performance study of the fast Fourier transform on vector IRAM," Berkeley University, California, Tech. Rep, 2000.
- [19] Y. Ouerhani, M. Jridi, and A. Alfalou, "Implementation techniques of high-order FFT into low-cost FPGA", in *Proc. IEEE 54th International Midwest Symposium on Circuits and Systems*, Korea, 2011.
- [20] P. Duhamel and M. Vetterli, "Fast Fourier transforms: a tutorial review and a state of the art," *Signal Processing Journal*, vol. 19, 1990.
- [21] S. Meiyappan, "Implementation and performance evaluation of parallel FFT algorithms," School of Computing. National University of Singapore, Singapore.
- [22] S. K. Palaniappan, "Design and implementation of radix-4 fast Fourier transform in ASIC chip with 0.18 μm standard CMOS technology," M.S. thesis, Malaysia, 2008.
- [23] M. Soni and P. Kunthe, "A General comparison of FFT algorithms," *Pioneer Journal Of IT & Management*, 2011.
- [24] B. Gough, "FFT algorithms," Tutorial paper, 1997
- [25] K. M. Pavan Kumar and P. Jain, et al, "FFT algorithms: a survey," *The International Journal Of Engineering And Science*, India, 2013.
- [26] S. Winograd, "On computing the discrete Fourier transform," *Proceedings of the National Academy of Sciences of the United States of America*, 1975.
- [27] F. Piccinin, "The fast Hartley transform as an alternative to the fast Fourier transform," Australia, Technical Memorandum, SRL-0006-TM, 1988.
- [28] R. Vincke and S. V. Landschoot, et al., "Calculating fast Fourier transform by using parallel software design patterns," Tech. Report, CW 627, October 2012.
- [29] A. Cortés, I. Vélez, M. Turrillas, and J. F. Sevilano, "Fast Fourier transform processors: implementing FFT and IFFT cores for OFDM communication systems," *Fourier Transform - Signal Processing*, 2012.
- [30] M. Frigo and S. G. Johnson, "The fastest Fourier transform in the west," Tech. Report, MIT-LCS-TR-728, 1997.
- [31] Bevan M. Baas, "An approach to low-power, high-performance, fast Fourier transform processor design," PhD Thesis, Stanford University, USA, 1999.
- [32] Y. Zhao and T. Ahmet, et al., "Architectural evaluation of flexible digital signal processing for wireless receivers," *Signals, Systems and Computers*, 2000.
- [33] A. Ganapathiraju, J. Hamaker, J. Picone, and A. Skjellum, "Contemporary view of FFT algorithms," Mississippi State University.
- [34] J. F. Herron, "Design and development of a high-speed Winograd fast Fourier transform processor board," M.S. thesis, Air University, USA, 1992.
- [35] C. Li, "Design and implementation of variable-length fast Fourier transform processor," M.S. thesis, Tatung University, 2006.
- [36] C. Kozyrakis and D. Patterson, "Vector vs. superscalar and VLIW architectures for embedded multimedia benchmarks," In *proc. of the 35th International Symposium on Microarchitecture*, Instabul, Turkey, November 2002.
- [37] T. Lippert, K. Schilling, et al., "Transpose algorithm for FFT on APE/Quadratics," In *Proc. of HPCN Europe*, pp. 439-448, 1998.
- [38] H.G. Yeh and G. Truong, "Speed and area analysis of memory based FFT processors in a FPGA," *Wireless Telecommunications Symposium*, pp. 1-6, 2007.
- [39] E. Brachos, "Parallel FFT libraries," M.S. thesis, Edinburgh University, 2011.
- [40] S. Johnsson and D. Cohen, "Computational arrays for the discrete Fourier transform," In *Proc. 22nd Computer Society International Conference*, 1981.
- [41] C.A. Silambarasan and L. Vanitha, "Design and implementation of low power and area efficient adder and Vedic multiplier for FFT," *International Journal of Communications and Engineering*, Vol. 1, 2012.
- [42] S. S. Kerur and Prakash Narchi, et al., "Implementation of Vedic multiplier for digital signal processing," *International Journal of Computer Applications*, 2011.
- [43] S. A. White, "Applications of distributed arithmetic to digital signal processing: a tutorial review," *IEEE ASSP Magazine*, 1989.
- [44] T. Sung, H. Hsin, and L. Ko, "Reconfigurable VLSI architecture for FFT processor," *WSEAS Transactions on Circuits and Systems*, 2009.

Comparison of Contemporary Protocols for High-speed Data Transport via 10 Gbps WAN Connections

Dmitry Kachan, Eduard Siemens

Anhalt University of Applied Sciences - Faculty of Electrical, Mechanical and Industrial Engineering,
Bernburger Str. 57, 06366 Köthen, Germany

E-mail: [d.kachan, e.siemens}@emw.hs-anhalt.de](mailto:{d.kachan,e.siemens}@emw.hs-anhalt.de)

Abstract—This work is dedicated to comparison of open source as well as proprietary transport protocols for high-speed data transmission via IP networks. The contemporary common TCP needs significant improvement since it was developed as general-purpose transport protocol and firstly introduced four decades ago. In nowadays networks, TCP fits not all communication needs that society has. Caused of it another transport protocols have been developed and successfully used for e.g. *Big Data* movement. In scope of this research the following protocols have been investigated for its efficiency on 10Gbps links: *UDT*, *RBUDP*, *MTP* and *RWTP*. The protocols were tested under different impairments such as Round Trip Time up to 400 ms and packet losses up to 2%. Investigated parameters are the data rate under different conditions of the network, the CPU load by sender and receiver during the experiments, size of feedback data, CPU usage per Gbps and the amount of feedback data per GiByte of effectively transmitted data. The best performance and fair resources consumption was observed by *RWTP*. From the open source projects, the best behavior is showed by *RBUDP*.

Keywords: high-speed data transport, transport protocol, WAN acceleration, big data.

I. INTRODUCTION

High-speed content delivery is a service that will be more and more demanded by society over the time. Especially it corresponds to reliable delivery, because the movement of Big Data is already part of mission critical services.

High-speed data transmission or more precisely, data transmission with maximum link utilization essentially depends on two factors:

1. link quality: presence of packet losses, abnormal delays or delay jitter;
2. transport protocol: the logic that handles all impairments that interfere data transmission.

Sometimes it is not possible to improve the link quality, moreover, some of impairments that affect data transmission are fundamentally unavoidable e.g. packet delay (OWD or RTT).

The second dependence has another situation: transport protocol is software that can be used on end device and by varying of the algorithms or parameters of the protocol user can achieve significant improvement of transmission rate. The applications that use transport protocols for high-speed data transmission are already available for use on the common PCs. The proprietary algorithms are packed in

commercial applications and they are not available for study, investigation and improvement. In contrary, open source approaches give such opportunity, however in most of the cases a commercial application provides better transmission acceleration. There are two reasons of it: proprietary solutions use more effective algorithms or they have better implementation.

In the paper [1] we have compared contemporary proprietary solutions for high-speed data transmission as opaque applications for data transport. As a logical continuation of the work we have presented there, a comparison of the bare protocol performance is presented in this research. Of interest is assertion of pure protocol performance unaffected by accessing to storage, handling data in the application and the processing of that data.

The solutions *RBUDP* and *UDTv4* are protocols that available for free download and currently are provide the fastest transport among open source protocols. *RWTP* is a proprietary protocol for high speed data transport that has been provided as software library by the Tixel [2] company for test proposals. *MTP* – is a protocol of the company Data Expedition [3] and was available for trial downloading from the web site of the company. Other providers of high speed data transport application are not provided access to their protocol.

II. RELATED WORK

The research in [1] shows the performance of applications based on the investigated protocols. The idea was to transfer 30GiBytes via WAN network in presence of different kinds of impairments and compare the achieved data rates and times of transportation for all solutions. Impressive result within single transport socket has been achieved by transport *TIXStream* [2]: in presence of 1% of packet loss and 200ms RTT the solution has almost 40% of 10 Gbps link utilization. The solution is based on Reliable WAN Transfer Protocol (*RWTP*) [3].

Another solution, where an access to its transport protocol was obtained – is *ExpeDat*, however, with this solution, under the same impairments only 4% of link utilization – 400Mbps were achieved. In this research these two proprietary protocols are compared with open source protocols *UDT* and *RBUDP*.

In research [4] Grossman et al. shows that throughput of UDT [5] via 10 Gbps link in presence of 116ms of RTT within a single stream is about 4.5 Gbps. That work also shows that within 8 parallel streams the performance could be increased up to 6.6Gbps. This research shows how real-world data is transported via production network. However of interest is to evaluate whether UDT would be able to handle packet losses in the link as well.

Performance of RBUDP via 10 Gbps connection is presented at the CineGrid 3rd Annual International Workshop [6]. The participants claimed that storage throughput limitation in that particular case was 3.5 Gbps, however the maximal data rate that has been achieved during the workshop did not exceed 1.2 Gbps. The RTT of that link was about 100 ms.

Last example shows implicitly that even if RBUDP was able to achieve better performance – it was limited by 3.5 Gbps, although the link capacity was 10Gbps. However it is not the bottleneck of protocol logics or of protocol implementation or even of the application implementation. It is bottleneck of the hardware. To avoid all issues related to implementation of applications or of the hardware, the evaluation of protocols performance in this work has been performed by transferring of dummy data without accessing to any potentially slow storage.

The packet loss in IP networks is a rather complicated phenomenon that depends on many factors and there is no some universal value that will characterize all the networks in the world. The best way to assess the approximate values of packet losses is through empirical measurements. In [7] Paxson discusses the heterogeneity of packet losses and shows that in 1994, the packet loss ratio between 35 sites in 9 countries was about 2.7 %. These measurements are not really up to date; however the principle knowledge, the author pointed to, is that the distribution of packet losses is not uniform. An assessment of nowadays packet loss dependencies is presented by Wang et al. in [8]. This research describes the tests, that were made across 6 continents between 147 countries, involving about 10 000 different paths. The authors show that across all continents the packet loss rate is less than 2 % for approximately 90 % of continental connections.

For all experiments in the current research, a hardware-based network impairment emulator has been used. In [9] Settlemyer et al. use a hardware emulator to emulate 10 Gbps links, and they compare throughput measurement results of the emulated links with real ones. The maximal RTT of a real link used in the research is 171 ms. The research shows that differences between profiles of both kinds of paths - emulated and real ones – are not significant, and the conclusion is that using emulated infrastructure is a much less expensive way to generate robust and real physical throughput.

III. TESTBED TOPOLOGY DESCRIPTION

The topology for this research replicates the topology in [1] to have a possibility to compare application results with protocol ones, where it is possible.

The scheme of test topology is kept quite simple. In

Fig. 2 the logical connection is presented. The core of the test environment setup is WAN emulator Apposite Netropy

10G [10] that allows an emulation of WAN links with different impairments such as packet loss ratio of up to 100 %, delays of up to 100 000 ms and delay jitter with an accuracy of about 20 ns. By comparison, software-based emulators, such as NetEm, provide an accuracy of about tens of milliseconds and this value is dependent on the hardware and operating system [11]. The Emulator allows a transmission of Ethernet traffic with an overall throughput of up to 21 Gbps on both, copper and fiber optic links.

The experimental setup contains two PC servers, connected via an Extreme Networks Summit x650 10 Gbps Ethernet switch and the WAN Emulator. The topology has been implemented by means of fiber optics with 10 Gbps bandwidth, see

Fig. 2. On the picture, the highlighted connection is the one before network emulator and the connection after it is not highlighted.

There is no background traffic used for experiments, since in the focus of this investigation is the pure protocol performance; the fairness aspects of the protocols are topics for a separate research.

Each server is equipped as follows:

- CPU: Intel Xeon X5690 @3.47GHz;
- RAM: 42 GiBytes (speed 3466 MHz);
- OS: Linux CentOS 6.3;
- NIC: Chelsio Communications Inc T420-CR, 10Gbps.

Operating system socket buffers were extended up to:

- `/proc/sys/core/net/wmem_max` – 64MiBytes
- `/proc/sys/core/net/rmem_max` – 64MiBytes

The MTU size of all network devices along the path has been set to 9000 Bytes.

IV. EXPERIMENTAL RESULTS

The experiment on each data transport solution under consideration consists of 42 consecutive tests. Each test comprises the transfer of dummy data from one server to another via the network emulator with a transmission duration of 180 seconds.

The emulated RTT range is varied from 0 ms up to 400 ms; packet loss ratio reaches up to 2%. One km of optical fiber delays a signal approximately by 5 μ s, therefore the RTT 400ms corresponds to 40 000km of fiber optics. The RTT is configured equally across forward and

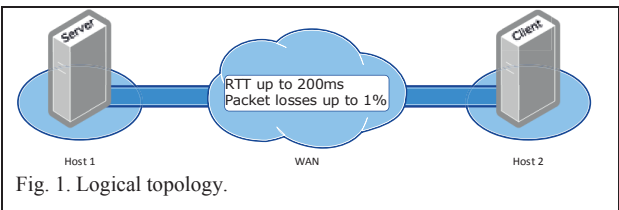


Fig. 1. Logical topology.

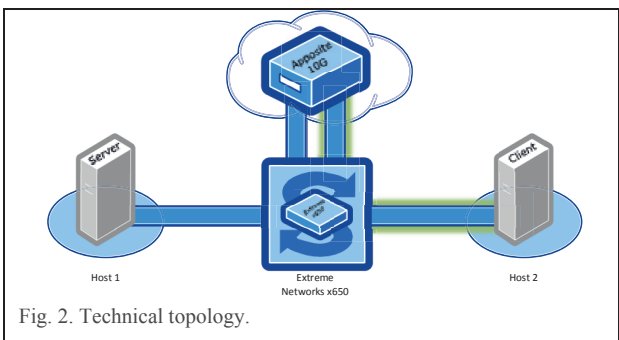


Fig. 2. Technical topology.

backward paths. Thereby 300 ms of RTT will delay signal by 150 ms in one direction and by another 150 in opposite one. The packet losses have a random behaviour with the normal distribution on forward and backward direction. However, it does not correspond to a real packet loss behaviour - in [7] Paxson shows that such behaviour of losses is more complex to handle by protocols than real-world burst packet losses. All the tests were repeated 4 times to avoid inaccuracies, and the best result of each series is presented on the plots.

The result of each test for each set of impairments includes average data rate, average CPU usage by sending host and amount of bytes that have been sent from receiver to sender. From obtained result the following metrics are calculated:

- CPU usage Per Gbps, see (1);

$$U_{per\ Gbps}^i = \frac{U^i}{R^i}, \quad (1)$$

where

i – is the number of test;

$U_{per\ Gbps}^i$ – CPU usage per 1 Gbps;

U^i – average CPU usage for i test;

R^i – average of data rate for the test.

- MiBytes of feedback per 1GiByte of data, see (2).

$$S_{fpGiB}^i = \frac{S_{fbk}^i}{R^i \cdot D/8}, \quad (2)$$

where

i – is a number of test;

S_{fpGiB}^i – size of feedback per GiByte of sent data;

S_{fbk}^i – size of feedback of whole test;

R^i – average of data rate for the test;

D – duration of test.

We consider that test is successful, if the average data rate is higher than 100 Mbps. The metrics have been calculated only for successful tests and average values also have been calculated between successful results.

It was observed that CPU usage behavior of receiver almost in all the cases repeats the usage behavior of sender. The metric CPU usage per Gbps is calculated only for sender. The ratio of average CPU usage on sender side to average CPU usage on reception side is also presented to get a tendency of the CPU consumption differences between the sender and receiver.

CPU consumption has been collected using top application in Linux with period 1s; the size of feedback has been collected using tool dstat with period 1s.

A. UDTv4

UDTv4 is a reliable transport protocol based on UDP. It was developed in 2002 at UCI by Yunhong Gu [12]. The protocol is available as a library for Linux with sample applications for file transmission. For the presented tests, the applications have been rewritten in order to avoid data gathering from file system. Only one data stream is used for both data streaming and control communication.

The following settings have been used to achieve maximal performance:

- $MSS - 8800\ Bytes$

In Fig. 3 the dependency of data rate from sets of impairments is presented. Decisive factor for the data transmission by UDTv4 is a value of packet loss. Already by 0.1 % of packet losses the data rate is no higher than 200 Mbps. In case of packet losses more than 0.1 %, the values of data rate are less than 100 Mbps.

In Fig. 4 CPU usage per Gbps by sender is shown. It is worth to mention that UDTv4 consumes CPU resources significantly even if level of data transmission is quite low. In the best case CPU usage per Gbps is about 28.1 %; in the worst case – 97.8 %. CPU consumption per Gbps at the highest data rate is 28.1 %. The mean value across all experiments is 63.4 % per Gbps.

The CPU consumption at the receiver has the same behaviour; the average of receiver's consumption is 1.1 times less than sender's one.

In Fig. 5 the receiver's feedback statistics is shown. It is easy to observe that the size of feedback not depends on average data rate, however in presences of packet losses its value is decreasing.

The size of feedback per GiByte of data lies in the range from 0.7 MiBytes per GiByte in case of RTT=400 ms and PL=0.1 % and up to 1.5 MiBytes per GiByte in case of RTT=0 ms and PL=0.1 %; in case of best data rate the size of feedback per GiByte is 1.4 MiBytes. The average value across the whole experiment is 1.2 MiBytes per GiByte of data.

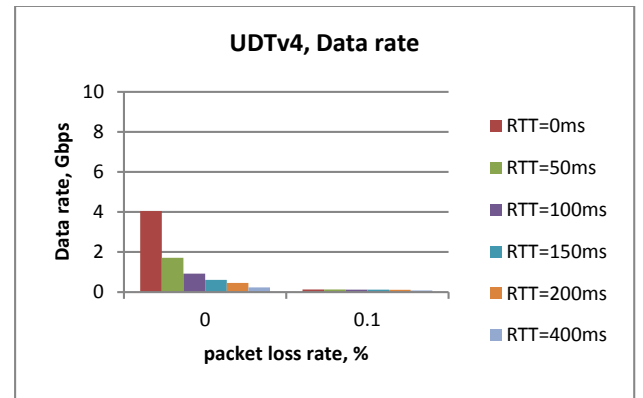


Fig. 3. UDTv4: Data Rate.

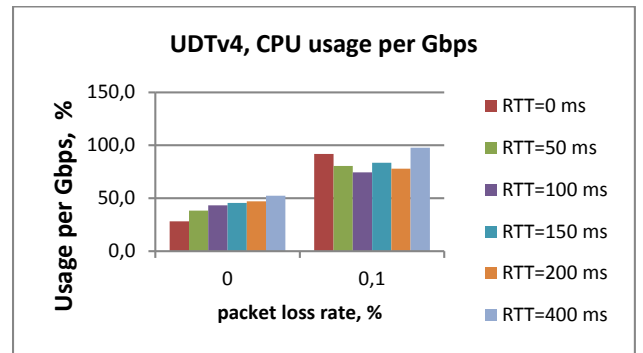


Fig. 4. UDTv4: CPU usage by sender per Gbps.

B. RBUDP

RBUDP [13] is another approach based on *UDP* for reliable data transport. The protocol as well as application was developed also at UCI 2002. In the opinion of the inventors *RBUDP* is a simple replacement of *FTP* for high-speed WAN connections also called Long Fat Pipes (LFP). Both, the application and protocol are open source projects.

The following settings have been used to achieve maximal performance:

- *MSS* – 8800 Bytes

In Fig. 6 the average data rate is shown. The decreasing factor of data rate for *RBUDP* is *RTT*. Even without packet losses, with injection of 50 ms of *RTT* the data rate decreases more than by factor 2. Such a trend is correct also for high values of packet losses e.g. 2%. However with further increase of *RTT* the behavior becomes smoother. The protocol is able to transmit data with data rate up to 3 Gbps on connections with *RTT* about 50 ms. Packet losses also affect the transmission rate, however not that significantly.

In Fig. 7 the distribution of CPU usage per Gbps for each test is shown. The behavior of CPU consumption copies the behavior of average data rate. It is noteworthy that the ratio of CPU consumption on the sender side to CPU consumption by receiver in all cases is approximately equal to 1.2. The values of CPU usage per Gbps lies in the range from 10.4%, for all the cases with *RTT* up to 400 ms, and up to 12.7% by *RTT*=400 ms and *PL*=2%. CPU consumption per Gbps by the highest data rate is 10.4%. The average value is about 10.6% per Gbps.

In Fig. 8 the size of feedback per each GiByte of data is shown. The behaviour mostly depends on losses in the

network. The values lie in range from 30 KiBytes, in the case without impairments, to 50 KiBytes, in the case with the heaviest impairments. The average value is about 37 KiBytes per GiByte of data.

Each transmission is performed using one *UDP* and one *TCP* socket on each side. *UDP* connection is used for data streaming; *TCP* – for control communication.

C. MTP

Multipurpose Transaction Protocol (*MTP*) [14] is a proprietary protocol that is used as a core of commercial application *ExpeDat*. The protocol logic is based on *UDP*, and uses a single *UDP* socket on each side of the connection for both data transmission and control information. To get maximal performance from the protocol the following parameters have been used:

- *MSS* – 8192 Bytes
- Environment variable *MTP_NOCHECKSUM=1*

The data rate behavior of *MTP* protocol is shown in Fig. 9. Although the protocol can resist against packet losses, the growing latency decreases the data rate significantly. Even with injection of at least 50 ms of *RTT* without packet losses the result is 4 times worse than without latency. The interesting behavior of the protocol is presented under condition *RTT*=0 ms, *PL*=0.1% – the data rate exceeds the one without any impairments in the link. Probably the reason for this is increasing of transmission rate by sender when significant delays are detected. In that case decreasing of performance with further increasing of delay is due to buffer limitations.

The CPU usage by sender mostly repeats the behaviour of the data rate. The values of CPU usage per Gbps lie in the range from 8.4% (*RTT*= 50 ms; *PL*= 0.1%) till 17.6%

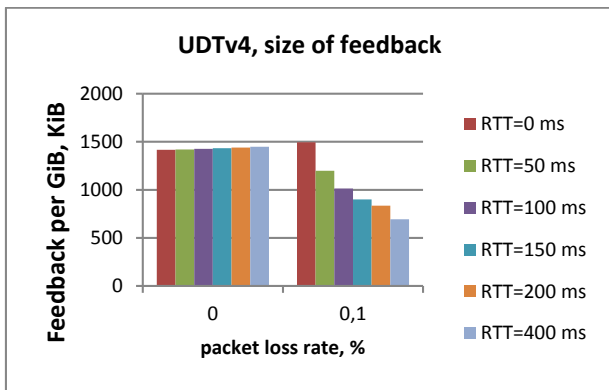


Fig. 5. UDTV4: Size of feedback per GiB of data.

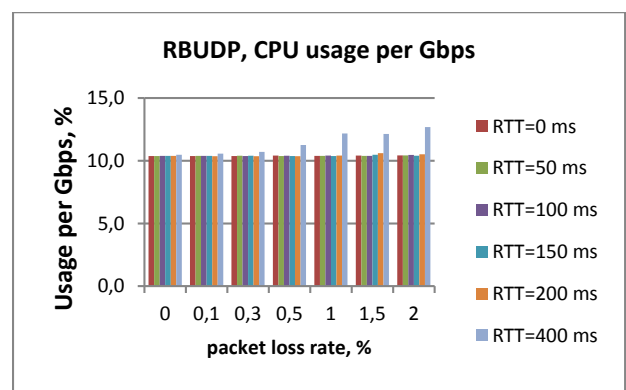


Fig. 7. RBUDP: CPU usage by sender per Gbps.

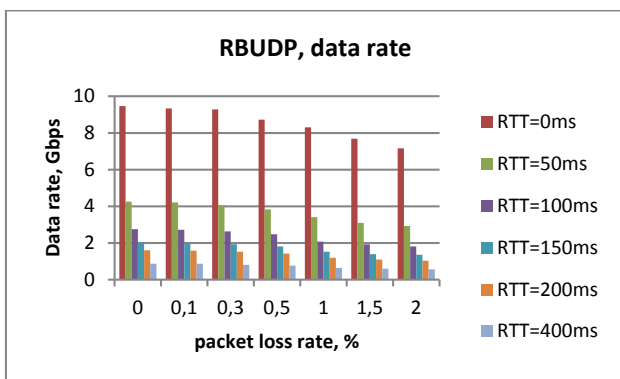


Fig. 6. RBUDP: Data Rate.

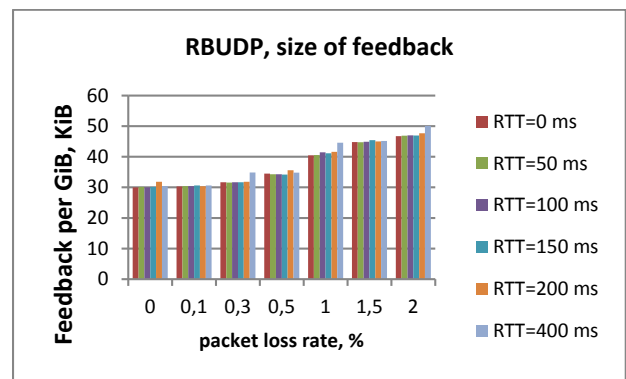


Fig. 8. RBUDP: Size of feedback per GiB of data.

(RTT=50 ms; PL=0 %). The average value is about 11.4 % per Gbps. The matter of interest is CPU consumption of MTP receiver side Fig. 10. MTP is the only one from investigated protocols which CPU consumption on receiver side exceeds the consumption on the sender side. For example by Packet loss 0% the values are approximately in two times bigger. CPU consumption per Gbps by the highest data rate is 9.2%. The average

ratio of sender CPU usage to receiver one is about 0.7.

The feedback size per GiByte of data for MTP is shown in Fig. 11. The biggest size of service information that has been sent back by MTP corresponds to the case when the impairments are absent on the link. The average amount of MiBytes per GiByte of data is about 3.2 MiBytes.

The protocol uses a single UDP connection for both data streaming and control communication.

D. RWTP

Reliable WAN Transfer Protocol [15] is a core technology underneath the *TIXStream* data transfer application [16]. The Protocol is based on *UDP* and uses one *UDP* socket on each side for control communication and data transmission. The following parameters have been tuned to achieve maximal performance by *RWTP*:

- *MSS* = 8800 Bytes
- *Receiver buffer size (on both sides)* = 1073741824 Bytes (1GiByte)
- *Sender buffer size (on both sides)* = 1073741824 Bytes (1GiByte)

In Fig. 12 the data rate of *RWTP* is shown. The protocol shows smooth degradation of data rate as impairments grow. The data rate across all the tests have not decreased lower than 1.5 Gbps. Both factors of latency and packet

losses are affecting the transmission, however only when they are significant: beginning from 100 ms of RTT and 1 % of packet losses.

The CPU consumption per Gbps by sender is shown in Fig. 13. The behaviour of receiver's CPU consumption repeats the sender's one. The ratio of CPU usage by sender to CPU consumption by receiver is about 1.2.

The values of CPU usage per Gbps lie in the range from 12.4 % (RTT=50ms, PL=0.1%) up to 29.9 % (RTT=400ms, PL=2%). CPU consumption per Gbps by the highest data rate is 12.6 %. The average value is about 15.4 % per Gbps.

The amount of bytes sent back to sender per GiByte of data is shown in Fig. 14. The amount of feedback data grows with growing of both packet losses and RTT. The values of feedback per GiByte are in the range from 5.6 KiBytes (RTT=0ms, PL=0%) up to 6 215 KiBytes (RTT=400 ms, PL=2 %). The average value is about 1 MiByte per GiByte of data.

V. COMPARISON OF THE PROTOCOLS

Only two protocols: *RBUDP* and *RWTP* have passed all the tests successfully – the data rate behaviour has not decreased less than 100 Mbps within the investigated impairments space.

The machines used in experiments are quite powerful, see section 0, to prevent issues with flow control due to a slow receiver.

Analysing the data rates of protocols (Fig. 3 Fig. 6, Fig. 9 and Fig. 12) by the packet loss rate 0 %, the decrease of data rate with the grows of RTT is most likely due to buffering issues. Since without latency all solutions show impressive results, it means that the protocol logic is able to handle the data on such rate appropriately. In *UDTv4* the authors claim that the protocol resizes the buffer automatically based on the value of Bandwidth Delay Product (BDP) [17] while in *RWTP* the buffer have been tuned manually to 1GiByte (the maximal possible value). For *RBUDP* and *MTP* only the system *UDP* buffer adjustment is recommended. The data rate is not decreased significantly on high impairments only by *RWTP* protocol.

The strong resistance against packet losses show *RBUDP* protocol and *RWTP*. *MTP* shows reasonable data rate in absence of latencies, however only till 0.5% of packet losses. Packet losses affect behaviour of *UDTv4* dramatically - with injection of at least 0.1 % of packet losses the protocol decreases its data rate down to 130 Mbps which is 1.3 % of its possible link capacity.

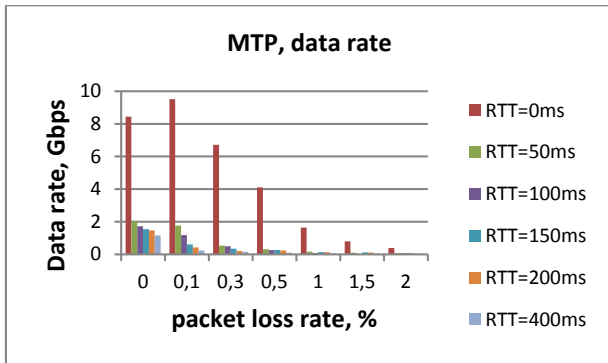


Fig. 9. MTP: Data Rate.

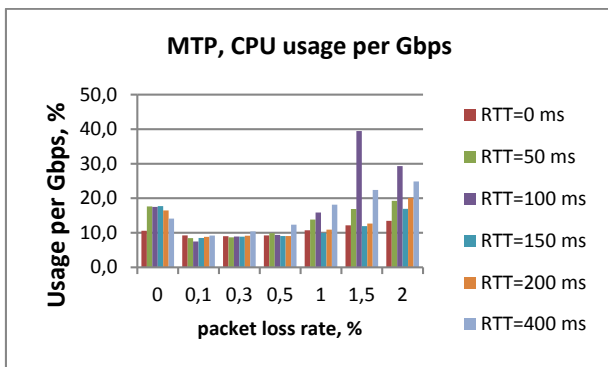


Fig. 10. MTP; CPU usage by receiver per Gbps.

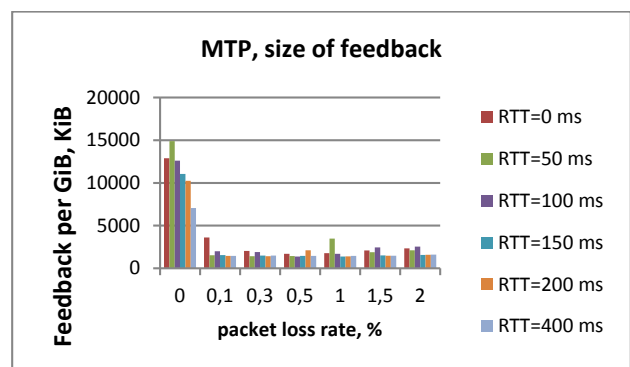


Fig. 11. MTP: Size of feedback per GiB of data.

It is possible to compare the behaviour of the pure protocol with behaviour of the solution based on that protocol. The performance of application based on *RWTP* and on *MTP* are presented in [1]. In case of *TIXStream/RWTP*, the experiment with *RWTP* showed better performance than *MTP*. The difference is most likely caused by application overhead due to the storage access and another system resources that are not responsible for data transmission. In case of *ExpeDat/MTP* the result is better by the application. Possible reason of it is not using of flag “-N 25” that shows that there are significant losses on the link by protocol performance test. However, this flag would not affect the case without packet losses, although even without the losses, the data rate by the application is higher than in the pure protocol case.

CPU consumption of every protocol except *MTP* is higher on the sender side than on the receiver side. The overhead of CPU usage for all solutions is not that significant – about 1.2 times. The only *RWTP* and *UDTv4*

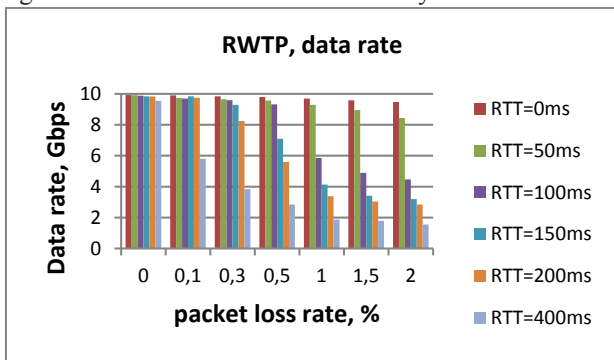


Fig. 12. RWTP: Data Rate.

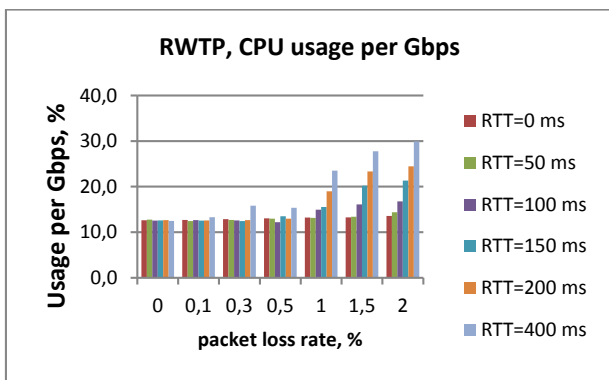


Fig. 13. RWTP: CPU usage by sender per Gbps.

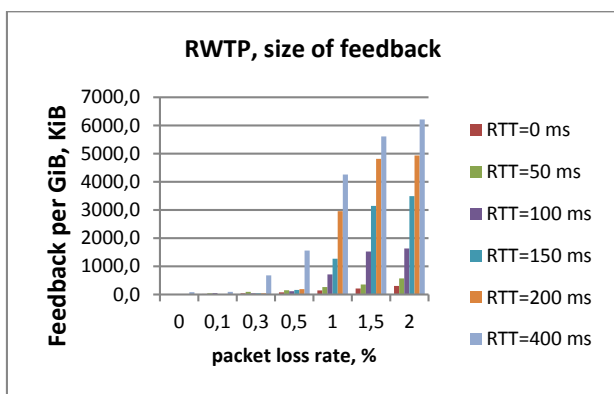


Fig. 14. RWTP: Size of feedback per GiB of data.

showed CPU usage higher than 100% on the sender side – it means that the protocol use more than one core for data transmission. This fact allows to avoid CPU bottlenecks for high-speed data transmission. In case of *RBUDP*, it is observable that under set of impairments where $RTT=0$ ms; $PL=0$ % or $PL=0.1$ % (see Fig. 6) the average CPU consumption is close to 100 % but does not exceed that value. Probably the data rate in these cases is not higher because of CPU bottleneck. This observation corresponds also for *MTP* behaviour.

The lowest average CPU consumption per Gbps is shown by *RBUDP*. However at the maximal data rate, the lowest consumption is achieved by *MTP* – 9.2 %. The highest – by *UDTv4*: 28.1 %. In all protocols, except *MTP*, with growing impairments the relative CPU consumption is increasing. It is most likely due to necessity in heavy retransmissions. *MTP* is the only protocol that has decreasing trend of CPU consumption on raising transmission rate, however the data rate of *MTP* in presence of impairments is quite low.

The lowest values of feedback data rate have been obtained within the experiment with *RBUDP* – across all experiments the values are about tens of KiBytes per GiByte of transmitted data. The reason of it is that, the retransmissions are done in “blast”, at the end of entire data transmission. The biggest value has been obtained by *MTP*. Probably this explains poor behaviour of the protocol. It is considered that packet losses affect feedback channel significantly less than traffic in forward direction [7]. In our experiments the packet losses have somewhat unrealistic random behavior with the normal distribution across both forward and backward link, what corresponds to extreme network conditions. The fair behavior in feedback has *RWTP* – without losses the feedback is measured in KiBytes and with growing impairments the value increases up to few MiBytes per GiByte. *UDTv4* has less feedback in presence of packet losses than without them. Within entire experiment under *UDTv4* the size of feedback is always in the range of few MiBytes.

Besides performance characteristics, usability of the transports are also of interest. The main disadvantage among all tested solutions is that some parameters should be predefined to reach higher performance. So the MSS size adjustment makes significant improvement of performance for all transports. The disadvantage of *RWTP* and *RBUDP* is that the user should also pre-define desired send rate and if the data rate will be higher than available throughput – the congestion occurs from which the software doesn’t recover. So the user should use some third party software like *LTest* [18] to investigate the link before data transmission. In case of *RWTP*, user should manually define the buffer size, if the default one is not big enough.

The only one protocol that uses more than one socket per session on each side is *RBUDP*. It is disadvantage since the protocol is not all-sufficient and needs support of one more reliable transport protocol - TCP.

VI. CONCLUSION

Reasonable performance has been achieved by *RBUDP* and *RWTP* protocols. *RBUDP* acts sometimes better than some proprietary solutions for high-speed data transport in [1]. *RWTP* shows the best results within this research: it

has resistance against packet losses and latency and it consumes reasonable amount of resources, however, the buffers should be predefined. It consumes more than one core which helps to achieve better performance than other solutions. The disadvantage of *RWTP* is that it is proprietary protocol. The *RBUDP* shows worse performance than *RWTP*, however, it is open source, which makes it more attractive to use it in some projects, or even to improve the algorithms of the protocol. The specific of *RBUDP* – requesting of all lost packets after receiving of all the data – allows minimizing the size of feedback, however such specific make impossible use of the protocol for streaming applications.

The size of feedback is not the important parameter as long as it comprises permilles of the net forward data amount, however if it is tens of MiBytes in case of transmission without impairments, as it showed by *MTP* in Fig. 11, the efficiency of that feedback is questionable.

The investigation of CPU usage showed that most likely the transports *RBUDP* and *MTP* would provide higher performance if they would use more than one core for sending of a data, or even if the sending of a data and receiving of a feedback by sender will be distributed between two separate cores.

VII. FUTURE WORK

The logical continuation of that research is a comparison of the protocols overhead in the forward direction. It is of interest to assess the relation of a throughput on the application layer – pure user data – to throughput on the network layer – user data including service information.

REFERENCES

- [1] D. Kachan, E. Siemens, V. Shuvalov, "Comparison of Contemporary Solutions for High Speed Data Transport on WAN 10 Gbit/s Connections," *Journal of Communication and Computer*, vol. 10, Nr. 6, pp. 783-795, 06 2013.
- [2] Tixel GmbH, Tixel, [Online]. Available: http://www.tixeltec.com/ps_tixstream_en.html. [Accessed 20 10 2012].
- [3] S. Höhlig, "Optimierter Dateitransfer über 100 Gigabit/s," In *Proceeding of the 100 Gigabit/s Workshop in Mannheim*, September 2011.
- [4] R. L. Grossman, Y. Gu, X. Hong, A. Antony, J. Blom, F. Dijkstra, C. de Laat, "Teraflows over Gigabit WANs with UDT," *Journal of Future Computer Systems*, pp. 501-513, 2005.
- [5] R. L. G. Y. Gu, "UDP-based datatransfer for high-speed wide area networks," *Computer Networks*, vol. 51, no. issue 7, pp. 1465-1480, May 2007.
- [6] L. Herr, M. Kresek, "Building a New User Community for Very High Quality Media Applications On Very High Speed Networks," *GLIF 2008*, 1 10 2008. [Online]. Available: <http://czechlight.cesnet.cz/documents/publications/network-architecture/2008/krsek-cinegrid.pdf>. [Accessed 05 02 2013].
- [7] V. Paxson, "End-to-End Internet Packet Dynamics," *Networking, IEEE/ACM Transactions*, vol. 7, no. Issue 3, pp. 277-292, 1999.
- [8] Y. A. Wang, C. Huang, J. Li, K. W. Ross, "Queen: Estimating Packet Loss Rate between Arbitrary Internet Hosts," *Proceedings of the 10th International Conference on Passive and Active Network Measurement*, pp. 57-66, 2009.
- [9] B. W. Settlemyer, N. S. V. Rao, S. W. Poole, S. W. Hodson, S. E. Hick, P. M. Newman, "Experimental analysis of 10Gbps transfers over physical and emulated dedicated connections," *proceeding of Computing, Networking and Communications (ICNC)*, pp. 845-850, 2012.
- [10] Apposite Technologies, "Apposite," [Online]. Available: <http://www.apposite-tech.com/index.html>. [Accessed 20 10 2012].
- [11] A. Jurgelionis, J.-P. Laulajainen, M. i. Hirvonen, and A. I. Wang, "An Empirical Study of NetEm Network Emulation Functionalities," 20. *ICCCN*, no. ISBN 978-1-4577-0637-0, pp. 1-6, 2011.
- [12] R. G. Y. Gu, "Udt: a high performance data transport protocol," *Doctoral Dissertation, University of Illinois, Chicago*, 2005.
- [13] E. He, J. Leigh, O. Yu, and T. A. DeFanti, "Reliable Blast UDP : Predictable High Performance Bulk Data Transfer," *Proceedings of IEEE Cluster Computing*, pp. 317-324, Sept. 2002.
- [14] Data Expedition, Inc, "Data Expedition," [Online]. Available: <http://www.dataexpedition.com/downloads/DEI-WP.pdf>. [Accessed 25 10 2012].
- [15] Tixel GmbH, "White Papers and Reports," [Online]. Available: http://www.tixeltec.com/papers_en.html. [Accessed 15 11 2012].
- [16] Tixel GmbH., "Tixel news," [Online]. Available: http://www.tixeltec.com/news_en.html. [Accessed 20 10 2012].
- [17] R. L. G. Y. Gu, "UDTv4: Improvements in Performance and Usability," in *Networks for Grid Applications, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, 2 ed., Berlin Heidelberg, Springer, 2009, p. 9.
- [18] E. Siemens, S. Piger, C. Grimm, M. Fromme., "LTest – A Tool for Distributed Network Performance Measurement," *Proc. Consumer Communications and Networking Conference*, 2004. First IEEE, pp. 239-244, 2004.
- [19] E. S. D. Kachan, "AvBandTest - a Testing Tool for Implementations of Available Band width Estimation Algorithms," in *Proceedings of 1st International Conference on Applied Innovations in IT*, Kothen, 2013.

Implementation of Multi-Core Processor Based on PLASMA (most MIPS I) IP Core

Bojan Gruevski, Aristotel Tentov, Marija Kalendar

SS. Cyril and Methodius University - Faculty of Electrical Engineering and Information Technologies

Karpos II bb, PO Box 574, 1000 Skopje, Macedonia

E-mail: gruevski.bojan@gmail.com, {toto, marijaka}@feit.ukim.edu.mk

Abstract—Multi-core processors is a design philosophy that has become mainstream in scientific and engineering applications. Increasing performance and gate capacity of recent FPGA devices has permitted complex logic systems to be implemented on a single programmable device. By using VHDL here we present an implementation of one multi-core processor by using the PLASMA IP core based on the (most) MIPS I ISA and give an overview of the processor architecture and share the execution results.

Keywords: FPGA, multi-core processor, MIPS, PLASMA, core, implementation, architecture, design.

I. INTRODUCTION

The concept of parallelism plays important role in the computer industry and the scientific community because it defines a way for the computer to realize two or more operations simultaneously and as a result the applications or calculations that are performed usually will finish and give the desired results more quickly. The improvement of the parallelism is one of the most important methods for faster task executions and usually the most applicable results of these parallelism techniques can be found in the processors. For example of a parallelism let's say we transfer two equally large files on a disk [1]. The transfer speed will depend on the bandwidth of the disk. If they simultaneously were transferred on two disks separately then we would have double transfer bandwidth so the transfer will complete twice as fast as in the first case.

There are different types of parallelism and different forms to execute it. One example of usage of the parallelism concept on system level is the usage of multiple disks and processors. The workload to meet customer requests can be distributed between the processors and disks so the result will be a better responsiveness of the system. When a system can expand its memory and its number of processors we call this system a scalable system and this is a highly valued property of the server systems.

Other type of parallelism concept is the concept of instruction level parallelism (ILP) [2]. This concept refers the possibility of execution of more than one instruction in one program at the same time. These instruction may be consecutive or not. One of the simpler forms for achieving this is the use of pipelining which is a technique that was used in the first processors for achieving the effect of

parallelism. The basic idea behind this technique is to overlap the execution of the instructions so it can reduce the execution time needed for a sequence of instructions. A key factor that leads to pipelining is the fact that not all instructions depend on the immediately preceding instruction so therefore they are partially or completely executed in parallel as much as possible.

II. MULTI-CORE PROCESSORS

Regardless of whether the processor core takes advantage of the instruction or thread level parallelism, multi-core processor have the advantage to exploit parallelism at task level found in the applications. A key factor that gives this advantage lies in the architecture of the task itself. Namely, each task has its own program counter, its own address space, its own registers and its own context so different tasks can be executed on different cores in the multi-core processor at virtually the same time [3]. This fact gives tremendous performance boosts on the execution of the applications but it is only achievable by good programming principles and very smart compilers that can order the code so it can be executed on a multi-core processor. If these requirements are not met then the performance of the execution of the application can be same or worse from the performance of the execution of the same application on a single-core processor.

III. PLASMA (MOST MIPS I) IP CORE

The PLASMA microprocessor is a 32 bit RISC processor core designed in VHDL¹ that is fully synthesizable. It executes all the instructions in user mode specified in the MIPS ISA with the exception of the instructions for misaligned memory access² [5]. On [4], the source code of the latest version of the PLASMA core can be downloaded. The current implementation incorporates and some peripherals like UART and Ethernet. It is tested on several FPGAs from [6] and [7].

¹ VHDL presents an acronym that consists from two acronyms: VHSIC and HDL. VHSIC stands for Very High Speed Integrated Circuit and HDL stands for Hardware Description Language. VHDL is a language defined by the IEEE (ANSI/IEEE 1076-1993).

² The implementation of these instructions was not possible due to the fact that those instructions were patented during the design of the processor.

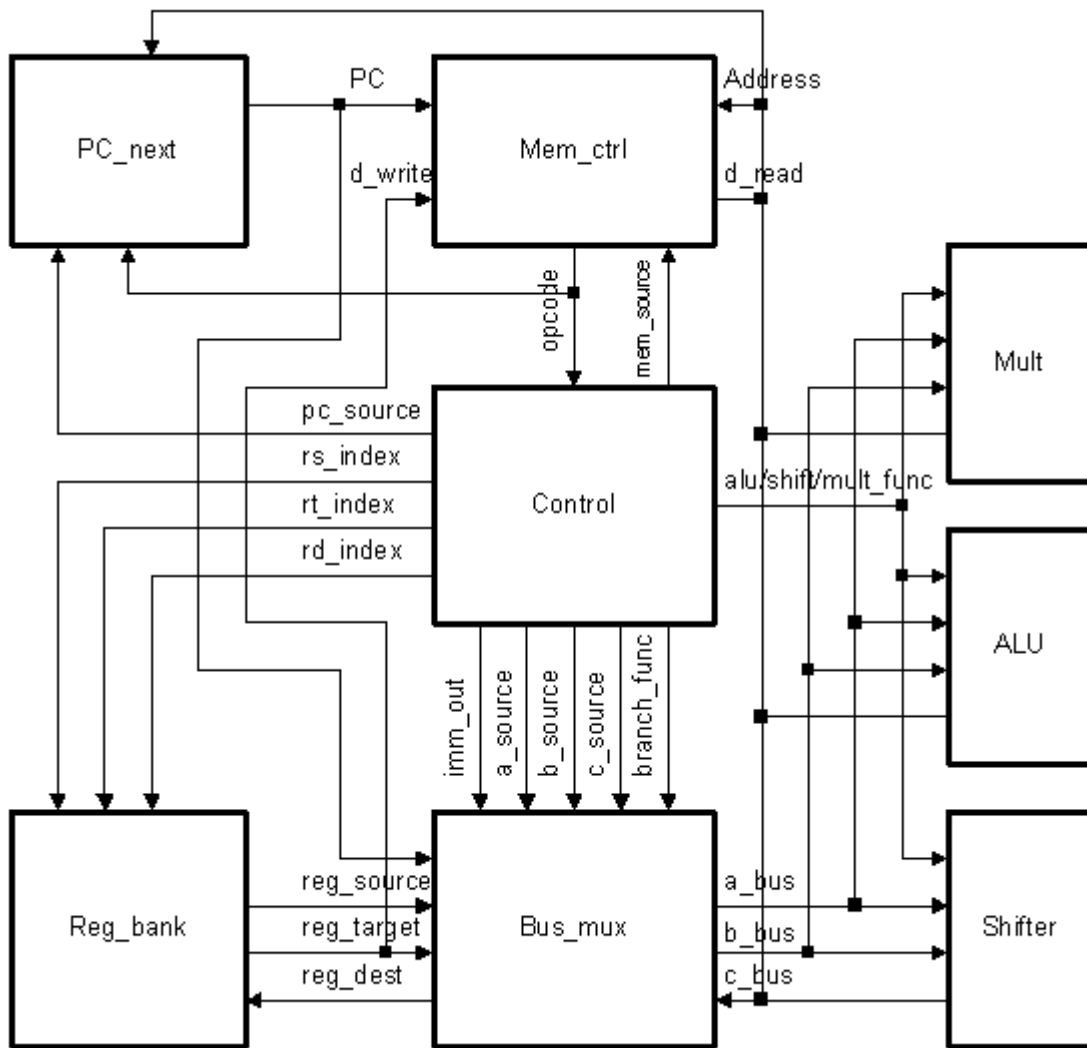


Fig. 1. PLASMA core architecture [1].

The processing unit can be implemented two or three stage pipeline and optional 4kB cache memory. The working frequency when implemented in [6] or [7] FPGAs has reached 25 MHz and 50 MHz depending of the FPGA model. The core is not implemented on a silicon level or at least it was not reported in the consulted bibliography.

Fig. 1 shows the architecture of the PLASMA core. For example an ADD instruction will perform the following steps (stages): the “pc-next” entity will pass the value of the program counter (PC) to the “mem_ctrl” entity which in return will fetch the opcode from memory (end of stage 1); the memory will return the opcode (end of stage 2); the entity “mem_ctrl” will pass the opcode to the “control entity”; this entity will convert the 32 bit opcode to a 60 bit VLIW³ code and will send control signals to the other entities; based on the “rs_index” and “rt_index” control signals, the entity “reg_bank” will send the 32 bit “reg_source” and “reg_target” values to the entity “bus_mux” (end of stage 3); based on the “a_source” and “b_source” control signals, the entity “bus_mux” multiplexes “reg_source” onto “a_bus” and “reg_target” onto “b_bus”; based on the “alu_func” control signals, “alu” adds the values from “a_bus” and “b_bus” and places the result on “c_bus”; based on the “c_source” control signals, “bus_mux” multiplexes “c_bus” onto “reg_dest”; based on the “rd_index” control signal,

³ VLIW is an acronym that stands for Very Large Instruction Word

“reg_bank” saves “reg_dest” into the correct register (end of stage 4, stage 3 if using two stage pipeline); read or write memory if needed (end of stage 5, stage 4 if using two stage pipeline).

The result of the simulation of the operation of the PLASMA core is shown on Fig. 2.

IV. MULTI-CORE PROCESSOR BASED ON PLASMA IP CORE

The multi-core processor showed on Fig. 9 and 10 that was implemented is a four (or two) core processor that is based on the PLASMA IP core mentioned before. It consists of:

Four cores based on the PLASMA IP core: There was very little core modification so that the architecture of the multi-processor remains simple. The only modification was with the addition of an arbiter sub module that will generate requests to the main bus arbiter in order for the processor to start read/write memory cycle. Bus arbiter (Fig 3): this is the module that controls the bus. It assigns the bus to one core at a time so it eliminates bus conflicts between the cores [8]. The main job of this entity is to make sure that the cores will access the bus fairly. For example core zero has bigger priority than one, one has bigger priority than three, three than four and so forth. When the bus will be freed by a core, the arbiter assigns the bus to the next core from the FIFO list that has requested it. Fig. 4 shows the usage of the arbiter.

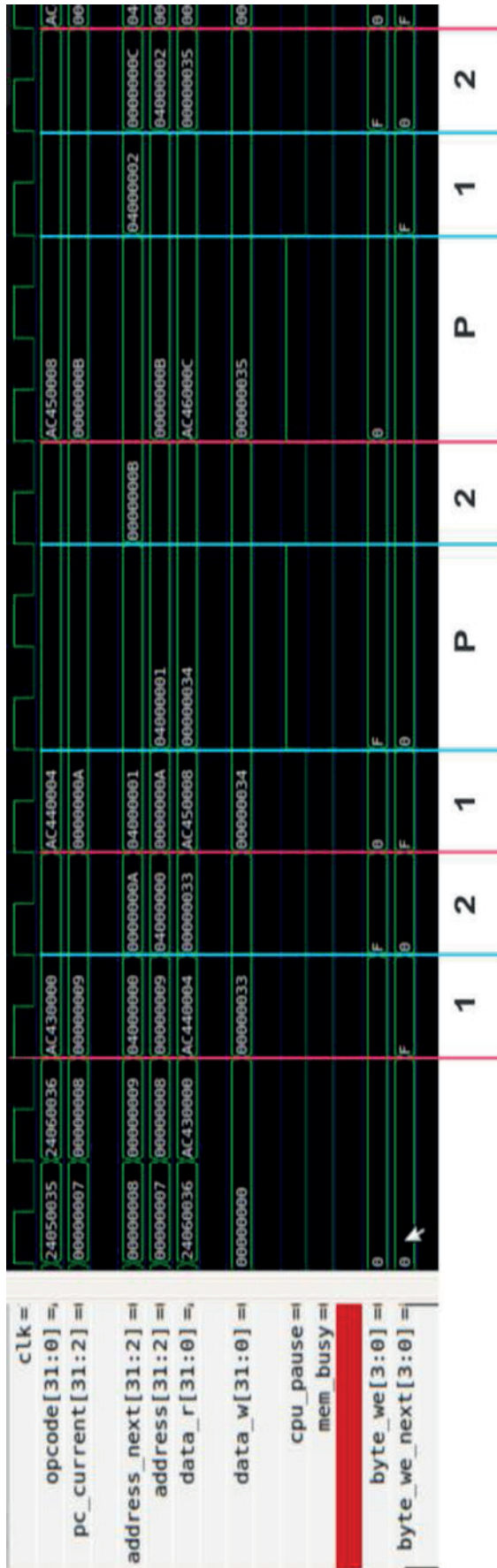


Fig. 2. Operation of the PLASMA core.

If the bus is free, this module will provide access on it on the first core that will make request. Subsequent request (while the bus is in use) are put in a FIFO list in the order

they happen. If there are simultaneous requests the priority is given by the core number.

The arbiter's buffer has to be with size equal or larger than the number of cores to accommodate the worst case scenario - when all cores have requests for bus utilization. His work is showed on Fig. 4.

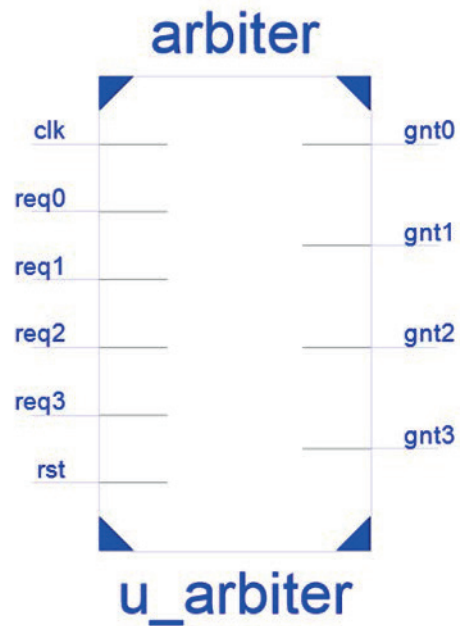


Fig. 3. Architecture of the arbiter.

Bus multiplexer: depending on which core the bus is allocated, the multiplexer connects the selected core bus with the main bus [9]. Its architecture is shown on Fig. 5.

There are several peripherals on the processor:

1. UART - standard UART controller which can be accessed by two registers, one for received data, and the other for data transmission. The data is consisted from 8 bits and it is without any parity bit.
2. Registers for interrupt requests - two registers, one for reading the status of the pending interrupts and the other for interrupt masking.
3. Counter register - 32 bit register used for counting clock cycles. It increases its value on every cycle
4. GPIO ports - two registers, one used to write values to the GPIO pins and the other used to read values from the GPIO pins.

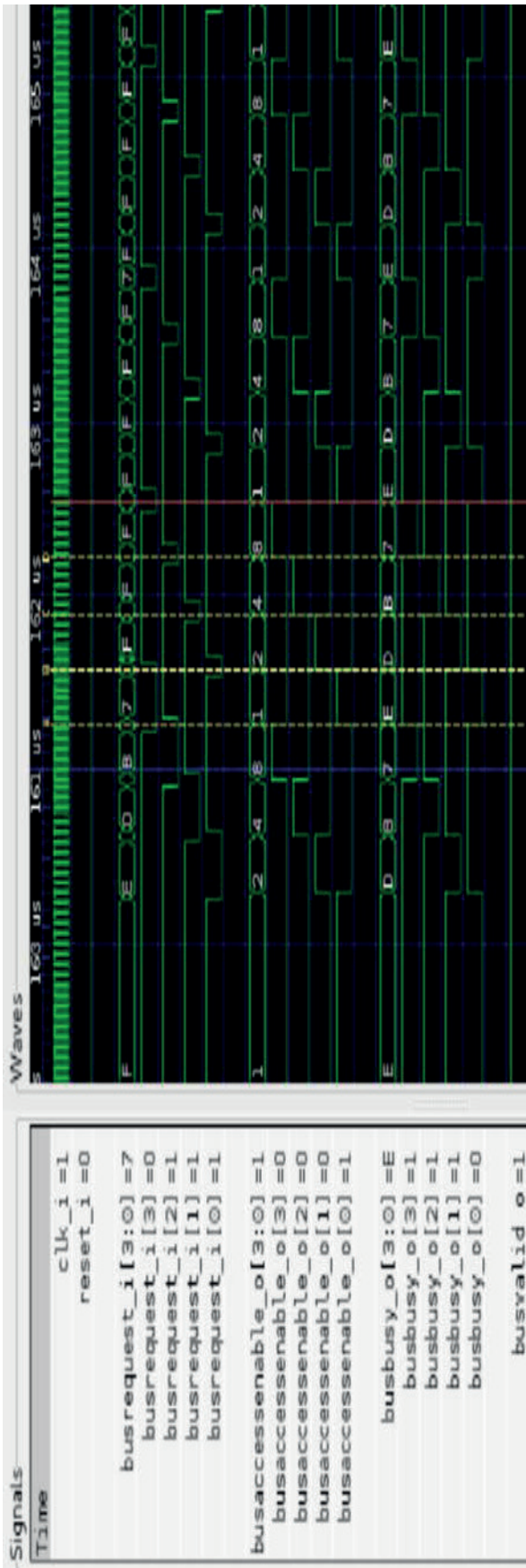


Fig. 4. Operation of the arbiter.

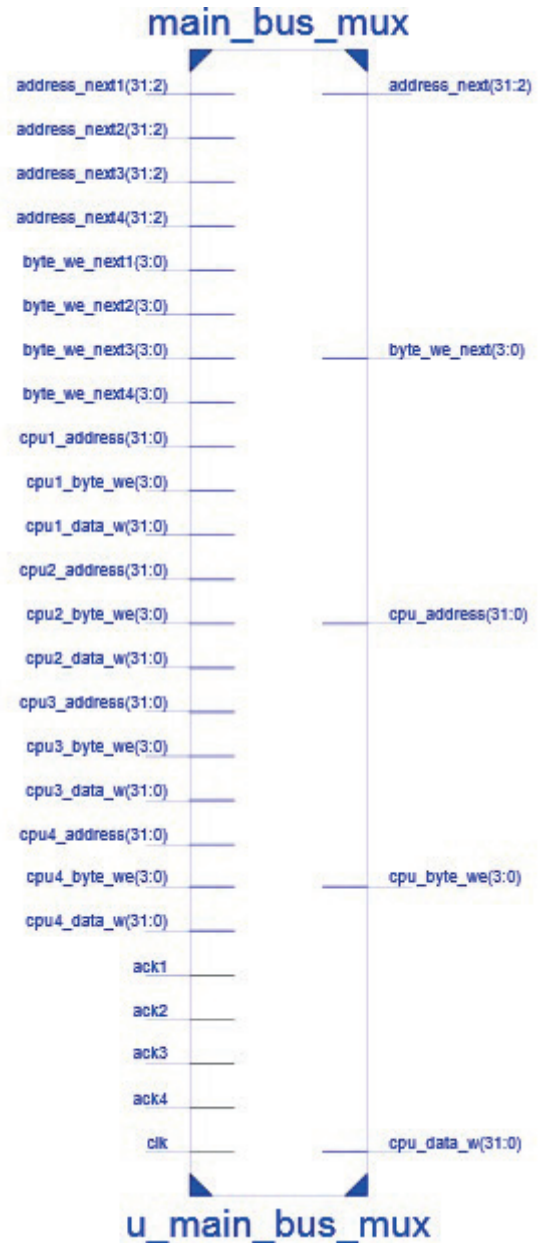


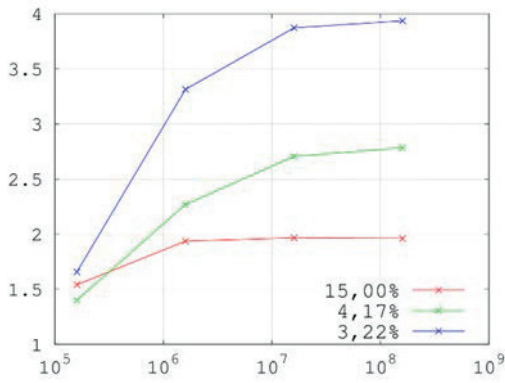
Fig. 5. Bus multiplexer architecture.

V. RESULTS

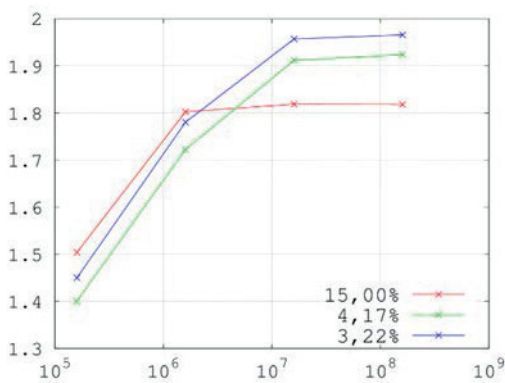
The following findings are based on test runs of different programs and in every succeeding program the number of instructions was increased [10]. Fig. 6 shows how the performance improvement in processing performs with increasing amount of work for the tasks and it is measured by the number of instructions that the tasks are running. The result shown are for three different tasks where the percentage of instructions accessing memory changes. In the plot on Fig. 6 tasks that have 15%, 4.17% and 3.22 % access to the processor bus are shown. The results from Fig. 6a and 6b are comparison of a processor with four and two cores respectively. In all cases it is seen that the performance rises with increasing the work that the task are doing. This is because the percentage of time it takes the operating system to reschedule tasks is declining. Beyond a certain value this time becomes negligible compared to the time of actual processing. It is also noticeable that we always get a limit on the improvement which depends on the percentage of instructions that access the bus. In the best

case on Fig. 6a this value is about 4 whereas on Fig. 6b is 2 which are the respective values that theoretically we can achieve in ideal cases. The ideal case is when we have no collisions on the bus and the only time we can assure that this will be possible is when no core will access the bus at any moment which is impossible. The increase in collisions causes degraded performance.

The next findings on Fig. 7 are based on testing the processor, running programs in which the utilization is increased progressively on the bus [11]. Collisions on the bus increase with increasing the number of cores and with increasing the bus utilization every time we add a core. Evaluation is done on how the efficiency of the core increases as the collisions decrease. On Fig. 7 it is shown that the improvement in the processing time of the tasks decreases along with the increase in the percentage of instructions that access the bus (for unwanted results, which is very low percentage, the improvement decreases again). This is because for these small values, effective processing time of tasks begins to decrease and the time to reschedule the tasks is no longer negligible.



(a) Execution on four cores



(b) Execution on two cores

Fig. 6. Improvement in execution time when going from one to four cores. Shows the relationship of times of execution in function of the amount of processing tasks are run. The amount of processing measuring instructions that execute tasks.

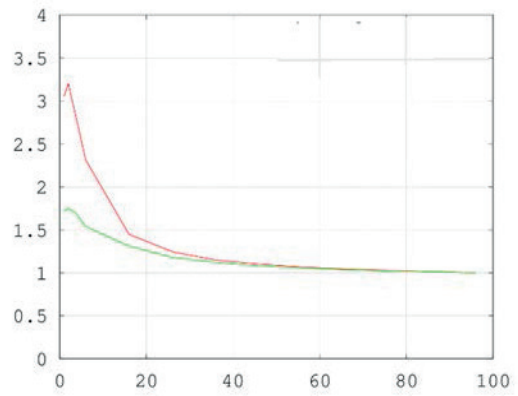


Fig. 7. Improvement in execution time for four and two cores, compared to a core function of the percentage of instructions that access to the bus (red - 4 cores, green - 2 cores).

VI. CONCLUSION

In this paper we have presented an implementation of a multi-core system on which were performed several tests. The presented result shows dependency of performance with the number of cores, the type of application (task) and whether we have memory cache or not. The factors that determinate the main efficiency of a processor is the bus usage on behalf the processing that needs to be done.

There are other factors to be evaluated in the future including other type of arbiter policies and memory cache that is not shared between data and instructions. Other future work may involve implementation of a larger multi-core processor with more than four cores, development of an operating system that will control the proper distribution of the tasks that need to be executed. Here we have showed author's first implementation of a multi-core processor and there are a lot of possibilities for further development of the design.

VII. APPENDIX

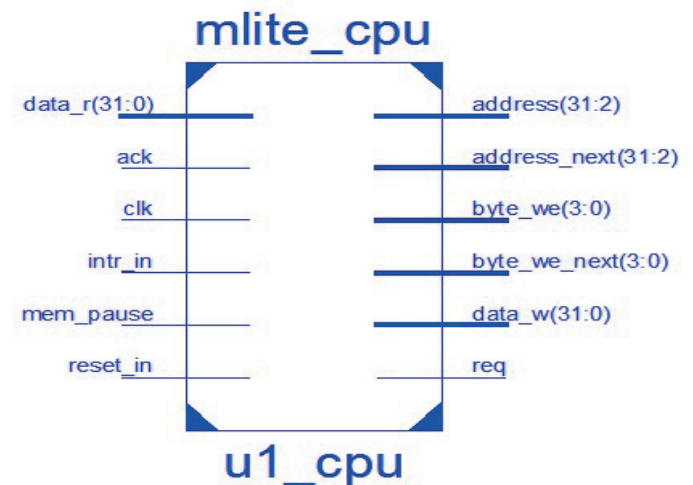


Fig. 8. PLASMA core connections.

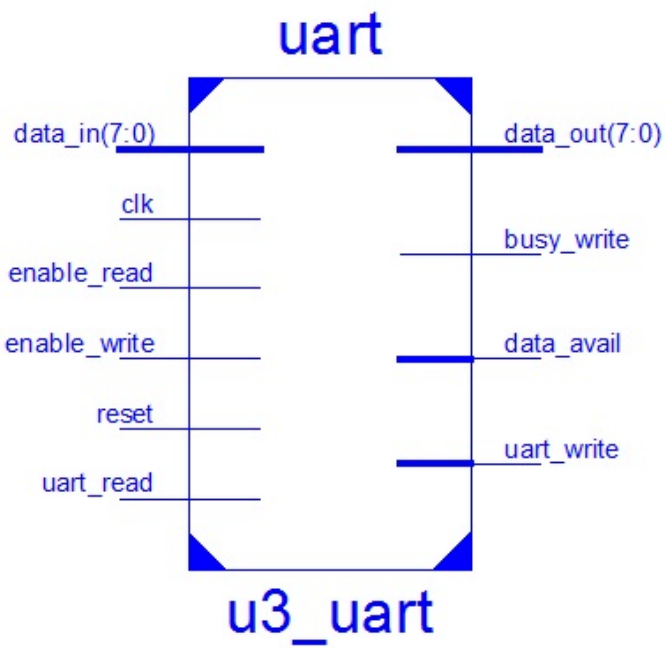


Fig. 9. Multi-core processor UART interface.

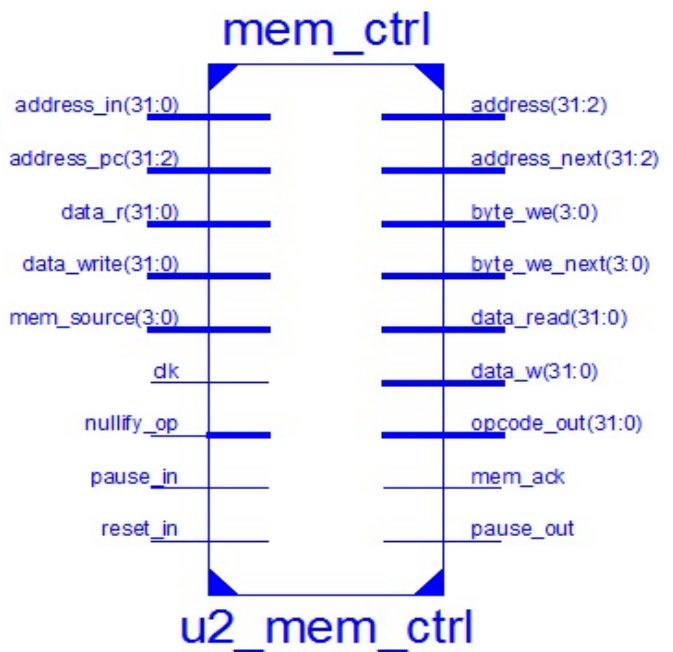


Fig. 10. Memory controller of the multi-core processor.

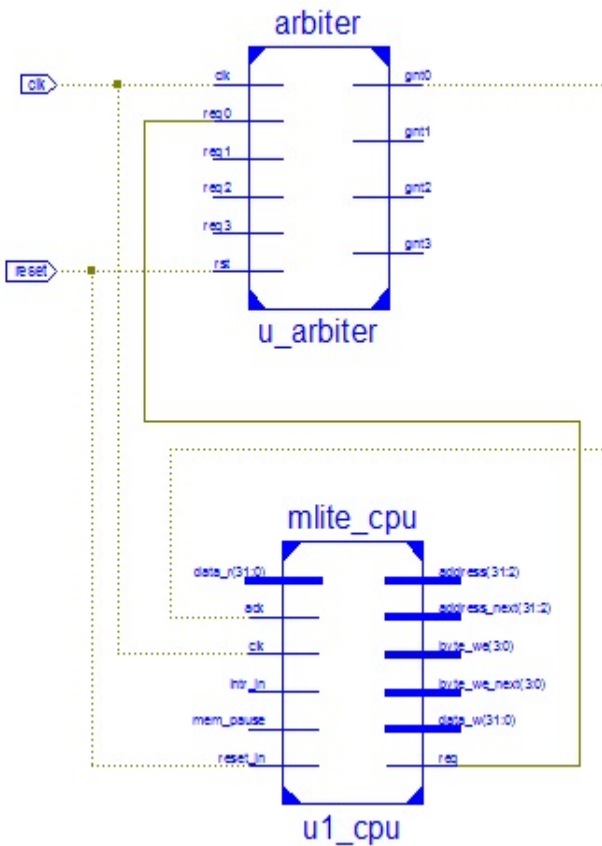


Fig. 11. Connection of the bus arbiter with one of the cores.

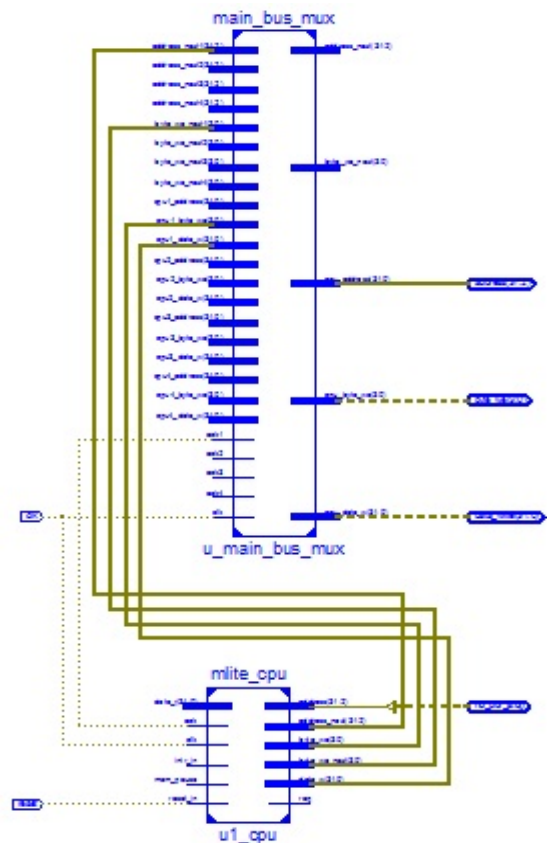


Fig. 12. Connection of the bus multiplexer with one of the cores.

REFERENCES

- [1] M. J. Flynn, "Computer Architecture: Pipelined and Parallel Processor Design," Jones & Bartlett Learning, Sudbury, Massachusetts, 1995.
- [2] J. L. Hennessy, D. A. Patterson, "Computer Architecture - A Quantitative Approach," 4th Edition. Morgan Kaufman Publishers, Burlington, Massachusetts, 2007.
- [3] A. S. Tanenbaum, "Modern Operating Systems," 2nd edition. Prentice Hall, Upper Saddle River, New Jersey, 2002.
- [4] Plasma – most MIPS I (TM) opcodes: Overview [Online]. Available: <http://opencores.org/project.plasma,overview>, (October, 2013)
- [5] MIPS M51xx Warrior-M class CPU Core [Online]. Available: <http://www.mips.com/>, (November, 2013)
- [6] Xilinx [Online]. Available: <http://www.xilinx.com/>, (October, 2013)
- [7] Altera [Online]. Available: <http://www.altera.com/>, (October, 2013)
- [8] David A. Patterson, John L. Hennessy, "Computer Organization and Design - The Hardware/Software Interface", 3rd Edition. Morgan Kaufman Publishers, Burlington, Massachusetts, 2005.
- [9] J. M. Rabaey, Anantha Chandrakasan, and Borivoje Nikolic "Digital Integrated Circuits - A Design Perspective", 2nd Edition, Prentice-Hall, Upper Saddle River, New Jersey, 2002.
- [10] D. Silva, K. Stangherlin, L. Bolzani, F. Vargas, "A Hardware-Based Approach to Improve the Reliability of RTOS-Based Embedded Systems," IEEE Computer Society, 2011, p. 209.
- [11] J. Tarrillo, L. Bolzani, and F. Vargas, "A Hardware-Scheduler for Fault Detection in RTOS-Based Embedded Systems," Digital System Design, Architectures, Methods and Tools, 2009, pp. 341 – 347.

The Research of Increase of Channel Efficiency for IP Traffic Transmission over Digital Power Line Carrier Channels

Anton Merkulov, Viatcheslav Shuvalov
Siberian State University of Telecommunications and Informatic Sciences –
Faculty of Automatic Electrocommunication
Kyrova Str. 86, 630102 Novosibirsk, Russia
E-mail: anton-merkulov@list.ru, shwp@sibsutis.ru

Abstract—This article is devoted to the research of channel efficiency for IP-traffic transmission over Digital Power Line Carrier channels. The application of serial WAN connections and header compression as methods to increase channel efficiency is considered. According to the results of the research an effective solution for network traffic transmission in DPLC networks was proposed.

Keywords: header compression, high voltage Power Line Carrier communication, IP-networks.

I. INTRODUCTION

In the power utilities Digital Power Line Carrier (dPLC) channels over High Voltage Lines (HVL) are widely used for voice and data transmission. It should be noted that PLC have not lost its significance even with the advent of OPGW technology. PLC equipment is widely used for 35/110/220 kV power lines.

Talking about convergent solutions with dPLC the networks built by SIEMENS AG with the use of Frame Relay Access Devices (FRAD) as node equipment is the prime example of successful technical solution. DPLC Frame Relay (FR) networks were organized in the period of 2002-2009 in the countries of Latin America, Asia, Africa and CIS. In many projects, the author was personally involved.

Outdated Frame Relay, removal of Frame Relay multiplexer production, and IP traffic transmission requirements updates led to the need to shift from Frame Relay technology to IP technology. Let's mark that discussing questions related to IP-traffic transmission, we will repeatedly refer back to the experience of creation of FR networks. It is necessary to explain what will be the cost of technology transition and what way is the most efficient.

II. THE VARIANTS OF IP-TRAFFIC TRANSMISSION VIA DPLC LINKS

Let's start from analysing the functionality of modern dPLC equipment for IP traffic transmission. It should be mentioned that in most modern devices function of Ethernet Bridge is integrated. At first glance it is the easiest way of IP networks organization. The use of widely spread Ethernet interface allows to connect any network equipment directly

to the PLC terminals. In this case the PLC terminal participates in packet processing, and it defines PLC terminal as network element.

It can make traffic filtering, control, QoS procedures and etc. Depending on producer decision the functions might be absolutely different. Structural scheme of IP-PLC link with application of Ethernet Bridge is shown in Fig.1.

A significant disadvantage of Ethernet Bridge is a big size of the header S_{L2} of the data link layer protocol Ethernet, equal to 26 bytes. DPLC channels are inherently low speed. In most cases the speed of data transmission does not exceed few tens of kilobits per second. Therefore, a large volume of overhead information leads to a reduction of channel efficiency Eff in link with the initially small data rate.

Turning back to the rich heritage of the FR networks, we can see that the size of the header S_{L2} was equal to only 6-8 bytes that is several times smaller in comparison to Ethernet Bridges. We ask the question whether it is possible to minimize the size of the S_{L2} in IP-based networks. The answer is: in order to reduce the size of the header of the data link layer protocol it is necessary to use PPP-encapsulation of Ethernet frames. But in this case, the IP-router should be connected to the dPLC over serial interface, such as X.21 or V.35.

It should be admitted that the choice of network equipment in this case is limited because many manufacturers do not use serial interface modules in their routers. But this solution has two distinct advantages over Ethernet Bridges. The use of serial WAN connection and PPP-encapsulation reduces the size of the link layer header up to 8 bytes. DPLC equipment is not involved in the packet processing, it just converts electrical signal of X.21 interface to high frequency (HF) signal and vice versa. It simplifies the architecture of dPLC devices that in turn reduces its cost. It also reduces time of channel recovery in case of loss of synchronization between modems. Using Ethernet Bridges it is necessary to consider the time of data exchange between network elements of dPLC devices following after recovering of synchronization between modems. From the author's practical experience the time of channel recovery for Ethernet Bridge links exceeds the time of link with serial WAN connection more than 3 times.

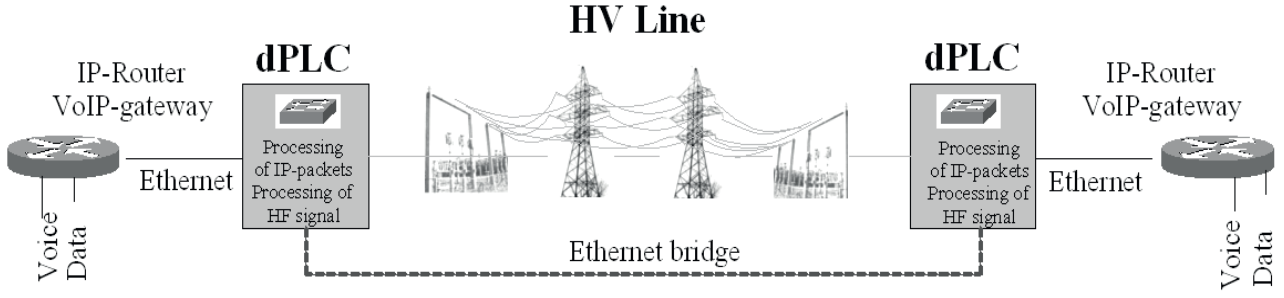


Fig. 1. Structural scheme of IP-PLC link with Ethernet –Bridge.

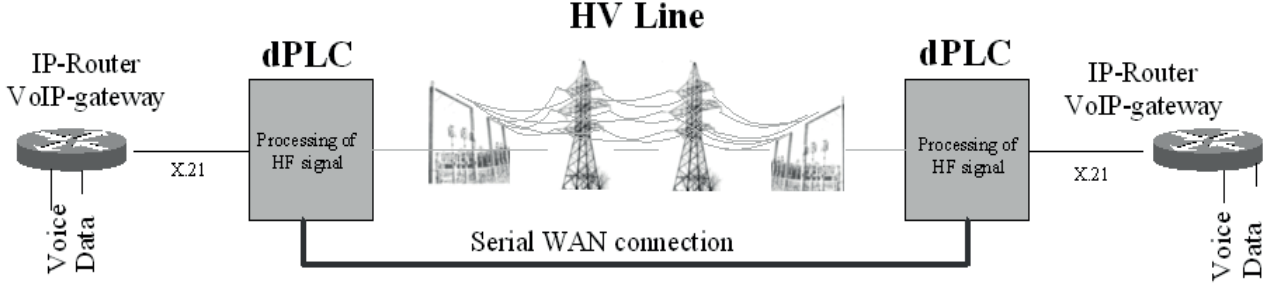


Fig.2. Structural scheme of IP-PLC link with serial WAN connection.

Another advantage of serial interfaces relates to the existing dPLC FR networks. The use of interface X.21 allows replacing FRAD multiplexers on IP routers and upgrading FR network without necessity to substitute the dPLC equipment. The structural scheme of IP-PLC link with serial WAN connection is shown in Fig.2.

At the network and transport layers in IP-networks its own headers are formed, which lead to further reduction of the channel efficiency. It should be noted that henceforth the IP protocol will be considered only in 4th version as widely used in the corporate communications networks.

III. COMPARISON OF ETHERNET BRIDGES AND SERIAL WAN CONNECTIONS

So, we need to compare the channel efficiency for Ethernet Bridge and serial WAN connection. For this we need to calculate the part of the payload in VoIP and in data frames.

The size of VoIP packet S_{VoIP} will depend on type of voice codec. Using a low bit-rate codecs ACELP or MP-MLQ the size of IP-datagram S_{Load} is equal to 20 or 30 bytes. The total size of IPv4/UDP/RTP header $S_{IP/UDP/RTP}$ in VoIP packet is equal to 40 bytes. For calculation of frame size and channel efficiency Eff_{VoIP} for voice frames (1) and (2) formulas are used. The results of calculation are shown in Table 1

$$Eff_{VoIP} = \left(1 - \frac{S_{L2} + S_{IP/UDP/RTP}}{S_{VoIP}} \right) \cdot 100\% \quad (1)$$

$$S_{VoIP} = S_{IP/UDP/RTP} + S_{Load} \quad (2)$$

TABLE I
RESULTS OF CALCULATION OF S_{VoIP} AND Eff_{VoIP}

L2 Protocol	S_{L2} , bytes	$S_{IP/UDP/RTP}$, bytes	S_{Load} , bytes	S_{VoIP} , bytes	Eff_{VoIP} , %
PPP	8	40	20	68	29,41
			30	78	38,46
			20	86	23,26
Ethernet	26	40	30	96	31,25
			20	86	23,26

Calculation shows that in best case Eff_{VoIP} does not exceed 40%, while the use of Ethernet Bridge is the worst variant ($Eff_{VoIP} = 23,26\%$).

We return to the dPLC Frame Relay networks. For the VoFR technology channel efficiency Eff_{VoFR} is 71.4% and 78.9% for voice packets with size of 20 or 30 bytes, respectively. Due to the size of headers Eff_{VoIP} is much lower than Eff_{VoFR} . It means, ceteris paribus, that it is necessary to expand bandwidth of dPLC channels. In most cases it is not possible because of deficit in available frequencies in band of high voltage Power Line Carrier Systems.

It will be more complicated to analyse data packets, because initially we do not know the exact size of maximum transport unit (MTU) which can vary from 46 to 1500 bytes. Since we consider convergent channels, it is necessary to take into account an important requirement - the time of data packets processing should be minimal, in order to provide speech transmission with minimal delay. But the size of the frame should not be too small to be in vain not to increase the share of the overheads. Serialization delay T_{serial} is the time that is necessary for processing of layer 2 frame from the first to the last bit and defines by formula (3). This is a factor that limits the acceptable size of frame.

T_{serial} depends on the size of frame S_{Data} and data rate C :

$$T_{serial} = \frac{8 \cdot S_{Data}}{C}, \quad (3)$$

where

$$S_{Data} = S_{L2} + S_{IP/TCP} + S_{MTU} \quad (4)$$

Recommended value of serialization delay should not exceed 10-15 ms [1]. For dPLC channels to meet this requirement is extremely difficult, explain why. It is clear that the greater capacity of the communication channel, the greater may be the size of data frame at a predetermined T_{serial} . The minimum size of the payload (MTU) for Ethernet is limited by 46 bytes. For guaranteed data transmission TCP protocol is used. The total size of IPv4/TCP headers $S_{IP/TCP}$ is 40 bytes. Consequently, for minimum MTU, the size of Ethernet frame will be equal to 112 bytes and PPP frame – 94 bytes.

The results of T_{serial} calculation for Ethernet and PPP protocols on dependence of data rate and the calculation of the maximum frame size of data for a given serialization delay of 10 ms are shown in Table 2.

Required serialization delay of 10 ms is performed for PPP protocol with data rate not less than 76 kbps and for Ethernet not less than 92 kbps. For dPLC links with 4 and 8 kHz bandwidth such data rate cannot be reached at all. Therefore minimum MTU size needs to be used regardless on data rate. Calculation of the channel efficiency Eff_{Data} for data transmission with the minimum size of MTU gives the following results: PPP-50%, Ethernet – 42%. At the maximum possible speed for dPLC channels in the 320 kbps, S_{data} can be increased up to 400 bytes, Eff_{Data} will be equal to 88-83% accordingly.

IV. APPLICATION OF HEADER COMPRESSION

Reduction of the amount of overhead for transmission of IP packets can be achieved by compression of the headers. The packet consists of two parts - a header and payload. In IP networks, after session establishment the most part of fields of header are static during the session time, and some fields could be inferred. Fig. 3 and Fig. 4 shows the structure of headers IPv4/UDP/RTP and IPv4/TCP and characteristics of its fields [2].

TABLE II
RESULTS OF CALCULATION OF T_{SERIAL} AND S_{DATA}

C , kbps	T_{Serial} (PPP), ms	T_{Serial} (Ethernet), ms	S_{Data} , byte
22000	34,2	40,7	
36000	20,9	24,9	
50000	15,0	17,9	with minimum MTU size
64000	11,8	14,0	
76000	9,9	11,8	
92000	8,2	9,7	115
114000	6,6	7,9	143
128000	5,9	7,0	160
144000	5,2	6,2	180
192000	3,9	4,7	240
190000	4,0	4,7	238
256000	2,9	3,5	320
320000	2,2	2,8	400

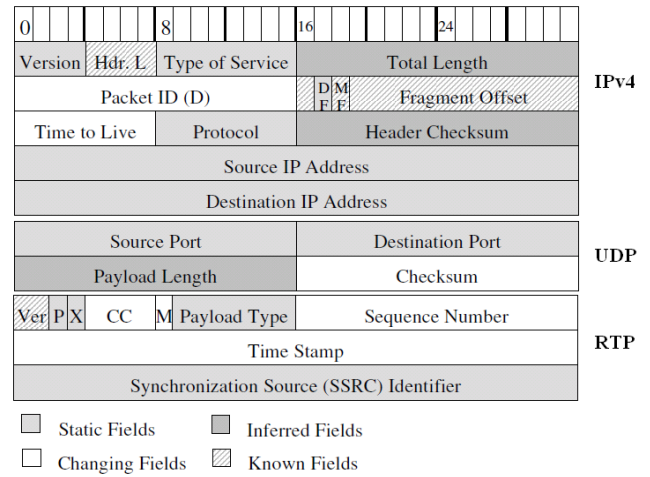


Fig.3. Characteristics of fields of IPv4/UDP/RTP header.

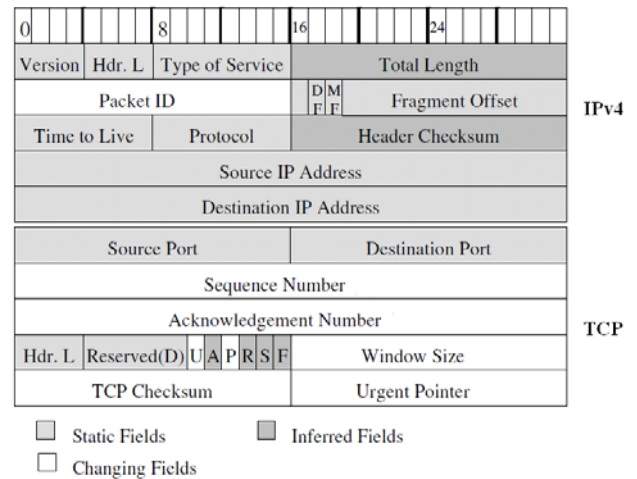


Fig.4. Characteristics of fields of IPv4/TCP header.

In 1990, CISCO software engineer Van Jacobson suggested to use IP header compression on the basis of statement that during session the most part of field in packet headers is static or rarely changing. In the same year the document RFC 1144 was published, describing header compression fundamentals. Today this technique is known

as Van Jacobson Header Compression. Nowadays existing compression techniques can be divided into two groups. The first group includes methods where transmission of delta values of varying fields is used. It includes methods RFC 2507 IPHC, RFC 2508 cRTP, RFC 3545 ECRTP. Its use for PPP connection is described in the document RFC 3544 "IP Header Compression over PPP". All these methods are implemented in practice in CISCO routers and use for the serial WAN connections with PPP-encapsulation. The second group includes methods ROHC (Robust Header Compression) RFC 3095 ROHC, RFC 4996 ROHC-TCP profile, RFC 5225 ROHC v2. In these methods information about changing fields is transferred with the use of Window Based Least Significant Bit Encoding. In most cases methods of ROHC are used in wireless communication. Table 3 shows the characteristics of the existing header compression methods.

The basic characteristics of header compression methods are: compression ratio showing the decrease of the header size after compression, scheme of compressed headers transmission and immunity.

Compression ratio is calculated by the formula (5) [6]:

$$G = 1 - \frac{S_c}{S_U} \quad (5)$$

where

S_c – compressed header size, bytes;

S_U - uncompressed header size, bytes.

In most cases transmission of compressed headers is performed with application of frames. In the beginning of the session N packets with uncompressed headers are transmitted.

Refreshing of the session context is performed by transmission of uncompressed header in the beginning of each following frame with number of packets F . This scheme is particularly effective when there is no feedback between the decompressor and the compressor, since in this case the context update request is impossible. According to the results of the research represented in [7] frame size must be equal to 10 packets to provide high compression ratio even for high values of bit error rate (BER). It should be noted that in practice the size of frame and scheme of compressed headers transmission depends on compression algorithm realized by equipment developers.

Considering F :

$$G = 1 - \frac{\left(\frac{N \cdot S_U + (F - N) \cdot S_c}{F \cdot S_U} + \frac{(Q - F) \cdot (S_U + (F - 1) \cdot S_c)}{F \cdot S_U} \right)}{\frac{Q}{F}} \quad (6)$$

where

Q – total number of packets transmitted during the session.

TABLE III
Characteristics of Header Compression Methods

RFC	Method	Characteristics	Implementation
1144	Van Jacobson Header Compression «VJHC»	IPv4/TCP from 40 to 4-7 bytes	CISCO routers [3] dPLC equipment PowerLink 50/100 (SIEMENS) [4] ETL 600 (ABB) [5]
2507	IP Header Compression «IPHC»	IPv4/TCP from 40 to 4-7 bytes IP/UDP from 28 to 2-5 bytes	CISCO routers
2508	Compressed Real Time Protocol «cRTP»	IPv4/UDP/RTP from 40 to 2-4 bytes	CISCO routers
3095	Robust Header Compression	IPv4/UDP/RTP from 40 to 1-4 bytes	EFFNET controller[6] dPLC equipment PowerLink 50/100 (SIEMENS) [4] ETL 600 (ABB) [5]
3545	Enhanced Compressed RTP (ECRTP)	IPv4/UDP/RTP from 40 to 8-12 bytes	CISCO routers
4996	Robust Header Compression TCP Profile	IPv4/TCP from 40 to 4 bytes	EFFNET controller [6]

The formula for calculation of average compression ratio G^* during the session can be represented as (7):

for $Q \rightarrow \infty$:

$$G^* = 1 - \frac{(F - 1) \cdot S_c + S_U}{F \cdot S_U} \quad (7)$$

Using (7) we find the value of the average size of compressed header during the session S_c^* :

$$S_c^* = S_U \cdot \left(\frac{(F - 1) \cdot S_c + S_U}{F \cdot S_U} \right) \quad (8)$$

Immunity is determined by the compression method resiliency in case of errors in the packets and their losses. According to EFFNET company researches the best immunity have ROHC methods and ECRTP. Compression ratio of ECRTP is significantly lower than ROHC one, because transmission of absolute values of changing fields instead of delta values is used.

Using (5), (7) and (8) we make the calculation of G , G^* , S_c^* . The size of compressed header S_c is assumed as maximum possible according to data shown in Table 3. Results of calculations are represented in Table 4. We can see that the size of compressed header is significantly reduced in comparison with uncompressed header.

TABLE IV
Results of G , G^* , S_C^* , Calculations

Method of compression	S_C , bytes	G	S_C^* , bytes	G^*
VJHC/ cRTP (IPv4/UDP/RTP 40 bytes)	4,0	0,90	≈ 8	0,81
ECRTP (IPv4/UDP/RTP 40 bytes)	12,0	0,70	≈ 15	0,63
ROHC (IPv4/UDP/RTP 40 bytes)	4,0	0,90	≈ 8	0,81
IPHC (IPv4/TCP 40 bytes)	7,0	0,83	≈ 10	0,74
ROHC-TCP (IPv4/TCP 40 bytes)	4,0	0,90	≈ 8	0,81

Finally we need to perform the calculation of the Eff_{VoIP} with the use of header compression. The results of calculations are shown in Table 5. For comparison the data for the VoFR technology is shown. The greatest benefit of compression can be achieved using PPP-encapsulation techniques and ROHC or cRTP compression. In this case channel efficiency increases almost twice and differs from the dPLC Frame Relay networks values not more than 13-16%. For Ethernet Bridge the benefit is not so significant.

TABLE V
Results of Eff_{VoIP} Calculation Given the Use of Header Compression

Payload, byte	Protocol L2	Without header compression	Eff_{VoIP} , %			
			cRTP	ECRTP	ROHC	VoFR (L2=8 bytes)
20	PPP	29,41	55,56	46,51	55,56	71,43
	Ethernet	23,26	37,04	32,79	37,04	
30	PPP	38,46	65,22	56,60	65,22	78,95
	Ethernet	31,25	46,88	42,25	37,04	

For data packets the gain from use of IP header compression can be expressed in opportunity to:

1) Reduce the serialization delay on size of ΔT_{serial} and become closer to the recommended value of T_{serial} :

$$\Delta T_{serial} = \frac{8}{C} \cdot (S_{Data} - (S_{Data} + S_C^* - S_U)) = \frac{8}{C} (S_U - S_C^*) \quad (9)$$

2) Increase the size of MTU on ΔS_{MTU} for the same serialization delay:

$$\Delta S_{MTU} = S_U - S_C^* \quad (10)$$

In the first case the delay is reduced, in the second one speed of information transmission is increased. For applied tasks the use of header compression allows to increase the size of the transport block by 30-34 bytes without increase

of the serialization delay. For example, instead of 46-byte MTU size 80 bytes can be used. It means that in one packet almost twice more information will be transmitted, that in turn will increase the channel efficiency Eff .

V. APPLIED TASKS OF IP-NETWORKS CREATION VIA dPLC

We consider applied tasks of IP networks building over dPLC links. Available on the market dPLC equipment allows realizing IP networks scenarios with the use of Ethernet Bridges and PPP connections. As was shown above, the use of header compression is necessary to increase the channel efficiency.

Compression methods cRTP, ECRTP, VJHC and IPHC are implemented in CISCO routers, but in this case connection to dPLC equipment must be carried out via the serial interface X.21.

Practical application of the most advanced ROHC methods is complicated by the necessity to integrate them into dPLC equipment. The leading companies that deal with the development, adaptation and implementation of ROHC are Swedish «EFFNET» and German «Acticom Mobile Networks». But in order to use their technical solutions the license is needed. Two biggest manufacturers of dPLC equipment SIEMENS AG and ABB already integrated compression ROHC (RFC 3095) and VJHC (RFC1144) into functions of their devices PowerLink 50/100 and ETL 600. In both cases, technical solution from EFFNET was used. Since in both devices Ethernet Bridge is used, Eff does not increase too much. For VoIP packets Eff_{VoIP} increases by no more than 4-6% in comparison to the uncompressed header transmission. For data packets everything depends on the size of MTU. For the minimum size of MTU Eff_{Data} increases on 15%.

Using IP routers with serial interfaces and PPP encapsulation eliminates the problem of large size of the data link layer protocol header. For VoIP packets, even for compression technique ECRTP, increase of Eff_{VoIP} will be more than 17%. Using IPHC compression technique for headers of data packets Eff_{Data} will increase more than 22%.

As result of performed research we can propose the development of converter in which PPP encapsulation is used, and for header compression ROHC techniques (RFC 3095) ROHC-TCP (RFC 4996) are used. In case of external rendition such converter can be used with dPLC equipment and network equipment of any manufacturers. Interconnection with dPLC is performed via serial interface, with network equipment – via Ethernet. This solution simplifies architecture of dPLC devices and reduces its cost. Structural scheme of IP-PLC link for suggested solution is shown in Fig.5.

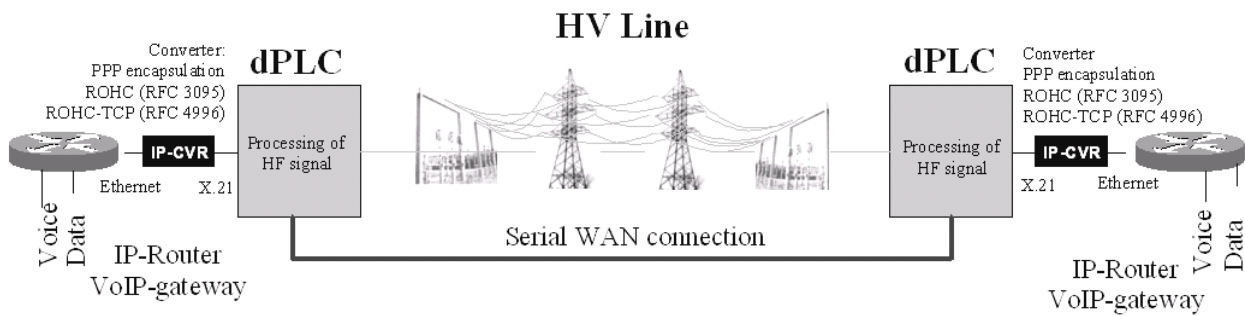


Fig.5. Structural scheme of IP-PLC link with using of external converter for header compression.

VI. CONCLUSION

1) Transmission of IP packets in dPLC networks with uncompressed headers leads to inefficient use of channel, because most of the resources are spent on transmission of overheads.

2) The use of header compression reduces the size of the header for VoIP packets up to 1-12 bytes and for data packets up to 4-7 bytes depending on type of compression techniques.

3) ROHC techniques provide the best compression ratio and high immunity, but its practical use is associated with the necessity to integrate compression algorithm into dPLC device. The simplest practical solution: the use of IP routers which perform header compression with the use of cRTP, ECRTP and IPHC techniques.

4) The use of Ethernet Bridges in dPLC networks is inefficient solution, because of the big size of L2 header. The best solution is application of PPP-encapsulation.

5) Interconnection of IP routers and PLC equipment via serial interface e.g. X.21 allows to avoid involvement of dPLC in packets processing. It reduces the total time of packet processing in the network and reduces the costs of additional functions of dPLC.

6) Creation of proposed converter will allow using the resources of channel capacity with the maximum efficiency.

REFERENCES

- [1] Design Best Practices for Latency Optimization.: CISCO, 2006. 8 p.
- [2] J. A. Ishac, "Survey of Header Compression Techniques," NASA/TM 2001-211154. 2001. Glenn Research Centre, Cleveland, Ohio. 27 p.
- [3] Header Compression.: CISCO, 2012. 6p.
- [4] Power Line Carrier PowerLink 100 CSPi Manual.: Siemens AG, 2011. 620 p.
- [5] Power Line Carrier ETL 600 Manual.: ABB, 2011. 440 p.
- [6] Compressed Time-to-Market for Compressed Headers – With EFFNET ROHC Software on Freescale Devices: Effnet, 2008. 12 p.
- [7] P. Seeling, M. Reisslein, P. T. Madsen, and F. Fitzek, "Performance Analyses of Header Compression Schemes in Heterogeneous Wireless Multi-Hop Networks," *Wireless Personal Communications* (2006) 38: pp. 203–232.

Implementation of Reed-Solomon RS(255,239) Code

Maja Malenko

SS. Cyril and Methodius University - Faculty of Electrical Engineering and Information Technologies
Karpos II bb, PO Box 574, 1000 Skopje, Macedonia
E-mail: majam@feit.ukim.edu.mk

Abstract—In this paper the construction of Reed-Solomon RS(255,239) codeword is described and the process of coding and decoding a message is simulated and verified. RS(255,239), or its shortened version RS(224,208) is used as a coding technique in Low-Power Single Carrier (LPSC) physical layer, as described in IEEE 802.11ad standard. The encoder takes 239 8-bit information symbols, adds 16 parity symbols and constructs 255-byte codeword to be transmitted through wireless communication channel. RS(255,239) codeword is defined over Galois Field $GF(2^8)$ and is used for correcting up to 8 symbol errors. RS(255,239) code construction is fully implemented and Simulink test project is constructed for testing and analyzing purposes.

Keywords: galois field, IEEE 802.11ad, QAM, Reed-Solomon

I. INTRODUCTION

Reed-Solomon codes are powerful technique for error detection and correction. Reed-Solomon codes are representatives of block code family, as they operate on an information by dividing it into blocks of data, consisting of information symbols to which parity symbols are added [1]. Reed-Solomon codes are known as RS(n,k) codes, where n represents the total number of symbols in one code block, while k is the number of information symbols in that block.

IEEE 802.11ad standard is popular standard for wireless local area transmission at 60 GHz with very high throughput. The IEEE 802.11ad standard supports three different modulation methods: spread-spectrum modulation, single carrier (SC) modulation and orthogonal frequency division multiplex (OFDM) modulation [2]. Different coding schemes are defined for each of these modulation methods, each having different error protection coding complexity and performance. One of the modulation methods which are covered by this standard is the Low-Power Single Carrier modulation technique, as subset of

Single Carrier modulation.

The purpose of this paper is to fully describe and implement the coding scheme used in Low-Power Single Carrier, which is RS(255,239) coding. It will be tested and simulated in Simulink.

The block diagram of Low-Power Single Carrier encoding and modulation steps is given in Fig. 1. Our focus will be RS (255,239) encoder block, which is simplified alternative version of LDPC encoding used in Single Carrier. Low-Power Single Carrier uses this encoding mainly because of the minimized power consumption, but still at the expense of less robust error correction.

The paper is being organized in the following chronology. First the Reed-Solomon encoding technique is being discussed, along with the necessary mathematical background needed for construction of RS(255,239) codes. In section III the process of encoding an example 239-byte message is described. In sections IV and V the encoding process is being simulated and verified in Simulink. Lastly, a conclusion is being drawn.

II. REED-SOLOMON ENCODING TECHNIQUE

Reed-Solomon codes are used as an error correction technique which adds redundant information to the data being transferred. The receiver, again using Reed Solomon decoding algorithm, can detect and correct specified number of errors which may have occurred in transmission. Reed-Solomon codes are widely used error correcting codes, mainly because of their efficiency and simplicity, compared to other codes [3].

First, some essential background on Reed-Solomon coding theory is provided, along with Galois field arithmetic which is the base for generating Reed-Solomon codes. Then, the process of encoding SC information, with given parameters is described, and simulated using Simulink.

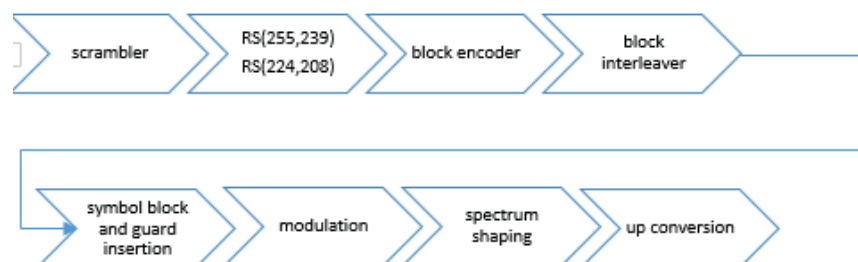


Fig. 1. Block diagram of Low-Power Single Carrier encoding and modulation steps.

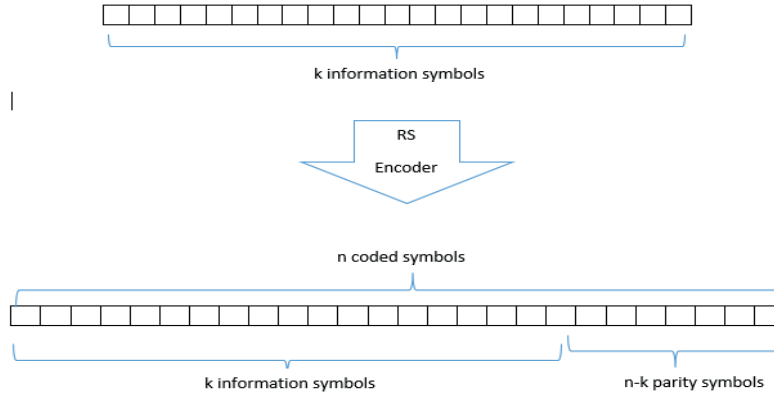


Fig. 2. Reed-Solomon code.

A. Reed-Solomon Codes

Every error detection and correction code has three primary characteristics: code length, dimension and minimum distance. The code length, n , represents the number of symbols per codeword. The dimension of the code, k , represents the number of actual information symbols, transmitted in one codeword. The minimum distance is the minimum number of symbol differences between codewords.

Reed-Solomon codes are block codes, which means they operate on blocks of data. They have linear and cyclic characteristics. Linearity is ability to construct new codeword by adding together already constructed codeword, while the cyclic characteristic is seen as the ability to produce new codeword by cyclically shifting the symbols of a given codeword. Reed-Solomon codes are particularly well suited for detecting and correcting burst data errors. If a symbol has error in more than one bit, that error still counts as one symbol error that can be corrected. This leads to a conclusion that Reed-Solomon codes can correct many bit errors.

The process of encoding a message consists of dividing the message which needs to be transmitted into submessages of specified length. Then, parity protection information is added to the end of each submessage, thus forming one block of specified length (specified number of symbols).

Reed-Solomon codes use abbreviation $RS(n,k)$, where n is the number of symbols in one block of data, and k is the number of information symbols in that block. If the number of bits in one symbol is m , then the number of symbols in a block can't be more than $2^m - 1$. In each block there are $n-k$ parity symbols, which means that the maximum number of corrected errors (in terms of symbols) in a block can be $t = (n-k)/2$, as shown in Fig 2.

Different values of parameters n and k , provide different error correcting capabilities and produce different complexity of the whole error correcting system. In this paper we are focusing on $RS(255,239)$ error correcting code, which can correct up to 8 erroneous symbols in one codeword. In the worst case, 8 bit errors may occur in one codeword, each one in different symbol, so the decoder can detect and correct 16 bit errors. In the best scenario, 8 complete symbol errors may occur, in every bit of 8 symbols, so the decoder can correct 8×8 bit errors.

Reed-Solomon codes perform their function according to

finite field arithmetic, as presented in the next section.

B. Galois Fields

Galois Field is a finite field, consisting of a finite number of primitive elements, usually denoted as α . If a symbol length is m , then the number of elements in a Galois field will be 2^m . The field is abbreviated as $GF(2^m)$ [4].

Each $GF(2^m)$ element can be represented in two m -bit forms, as α coefficient which is an element from the set in (1) or as a polynomial expression, as described as in (2).

$$0, \alpha^0, \alpha^1, \dots, \alpha^{2^m-2} \quad (1)$$

$$a_{m-1}x^{m-1} + \dots + a_1x + a_0 \quad (2)$$

The coefficients $a_{m-1} \dots a_1 a_0$ can take only binary values.

One symbol in $RS(255,239)$ consists of 8 bits, which means that Galois Field has 2^8 , or 256 field elements.

In $GF(255,239)$ there are 256 α coefficients:

$$0, \alpha^0, \alpha^1, \dots, \alpha^{254}$$

and the polynomial representation of a field element in the same Galois Field is:

$$a_7x^7 + a_6x^6 + a_5x^5 + a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0$$

In order to define Reed-Solomon code, two main generator polynomials need to be constructed: field generator polynomial and code generator polynomial [5].

The field generator polynomial, $p(x)$, needs to be irreducible polynomial of degree m . In $GF(2^8)$ there are 18 irreducible polynomials, and we have chosen the primitive polynomial

$$p(x) = x^8 + x^4 + x^3 + x^2 + 1 \quad (3)$$

In order to define the Galois Field, we need to calculate all field elements. Given the primitive polynomial, $p(x)$, and knowing that α must be a root of $p(x)$, we can write:

$$p(\alpha) = 0 \\ \alpha^8 + \alpha^4 + \alpha^3 + \alpha^2 + 1 = 0 \quad (4)$$

$$\alpha^8 = \alpha^4 + \alpha^3 + \alpha^2 + 1$$

This means that α^8 , which is one of 256 α coefficients, represents a Galois Field element written in binary form as 0001101 or in decimal form as 29.

By using Galois arithmetic operations, we can calculate the other 255 Galois Field elements as shown in Table 1.

TABLE I
GF(2⁸) FIELD ELEMENTS, WITH $p(x) = x^8 + x^4 + x^3 + x^2 + 1$

Index form	Polynomial and binary form								Decimal form
	α^7	α^6	α^5	α^4	α^3	α^2	α^1	α^0	
0	0	0	0	0	0	0	0	0	0
α^0	0	0	0	0	0	0	0	1	1
α^1	0	0	0	0	0	0	1	0	2
α^2	0	0	0	0	0	1	0	0	4
α^3	0	0	0	0	1	0	0	0	8
α^4	0	0	0	1	0	0	0	0	16
α^5	0	0	1	0	0	0	0	0	32
α^6	0	1	0	0	0	0	0	0	64
α^7	1	0	0	0	0	0	0	0	128
α^8	0	0	0	1	1	1	0	1	29
α^9	0	0	1	1	1	0	1	0	58
α^{10}	0	1	1	1	0	1	0	0	116
α^{11}	1	1	1	0	1	0	0	0	232
...									
α^{253}	0	1	0	0	0	1	1	1	71
α^{254}	1	0	0	0	1	1	1	0	142

C. Constructing RS(255,239)

In order to represent the information and parity symbols of

Reed-Solomon code as elements of GF(2⁸), the code generator polynomial, $g(x)$, needs to be constructed. The code generator polynomial has $2t=n-k$ factors, and their roots are consecutive elements of previously defined Galois Field. In general form, the code generator polynomial is defined as:

$$g(x) = (x + \alpha^b)(x + \alpha^{b+1}) \dots (x + \alpha^{b+2t-1}) \quad (5)$$

where b can be 0 or 1.

For RS(255,239) code, the number of parity symbols is $n-k=2t=16$, so the polynomial and decimal form of the code generator polynomial for correcting up to 8 errors is:

$$g(x) = (x + \alpha^0)(x + \alpha^1)(x + \alpha^2)(x + \alpha^3)(x + \alpha^4)(x + \alpha^5)(x + \alpha^6)(x + \alpha^7)(x + \alpha^8)(x + \alpha^9)(x + \alpha^{10})(x + \alpha^{11})(x + \alpha^{12})(x + \alpha^{13})(x + \alpha^{14})(x + \alpha^{15}) = (x + 1)(x + 2)(x + 4)(x + 8)(x + 16)(x + 32)(x + 64)(x + 128)(x + 29)(x + 58)(x + 116)(x + 232)(x + 205)(x + 135)(x + 19)(x + 38) \quad (6)$$

Galois Field multiplication and addition operations [6] are performed over $g(x)$, and the resulting field generator polynomial, represented in decimal form, is:

$$g(x) = x^{16} + 118x^{15} + 52x^{14} + 103x^{13} + 31x^{12} + 104x^{11} + 126x^{10} + 187x^9 + 232x^8 + 17x^7 + 56x^6 + 183x^5 + 49x^4 + 100x^3 + 81x^2 + 44x + 79 \quad (7)$$

III. REED-SOLOMON ENCODING PROCESS

In this section the arithmetic Reed-Solomon encoding process will be briefly described, and then applied over an example sending message encoded using RS(255,239) code.

A. Forming The Codeword

The message that needs to be encoded in one block, consists of k information symbols. It can be represented as an information polynomial, $I(x)$, of degree $k-1$:

$$I(x) = I_{k-1}x^{k-1} + \dots + I_1x + I_0 \quad (8)$$

where each of the coefficients I_{k-1}, \dots, I_1, I_0 is an m -bit symbol in GF(2 ^{m}).

The process of encoding the message represented in (8) consists of multiplying the information polynomial, $I(x)$, by x^{n-k} , and then dividing the result by the code generator polynomial, $g(x)$.

The encoded message polynomial, which represents the transmitted codeword, can be formed as follows:

$$T(x) = I(x)x^{n-k} + r(x) \quad (9)$$

where $r(x)$ is the remainder of division operation of $I(x)x^{n-k}$ with $g(x)$.

B. Encoding An Example RS(255,239) Message

In order to produce RS(255,239) codeword, an information message consisting of 239 8-bit symbols was generated in Matlab. The information polynomial is in the following form:

$$x^{238} + 2x^{237} + 3x^{236} + \dots + 237x^2 + 238x + 239 \quad (10)$$

This information polynomial is then multiplied by x^{16} to allow space for 16 parity symbols, and the following polynomial is produced:

$$x^{254} + 2x^{253} + 3x^{251} + \dots + 237x^{14} + 238x^{15} + 239x^{16} \quad (11)$$

This polynomial is divided by the code generator polynomial, given in (6). The remainder represents the 16 parity symbols which are concatenated at the end of the information message, thus producing RS(255,239) codeword.

The encoded message polynomial, $T(x)$, has the following form:

$$T(x) = x^{254} + 2x^{253} + 3x^{252} + \dots + 237x^{18} + 238x^{17} + 239x^{16} + 37x^{15} + 133x^{14} + 255x^{13} + 126x^{12} + 37x^{11} + 59x^{10} + 132x^9 + 133x^8 + 56x^7 + 168x^6 + 179x^5 + 4x^4 + 9x^3 + 99x^2 + 79x + 148 \quad (12)$$

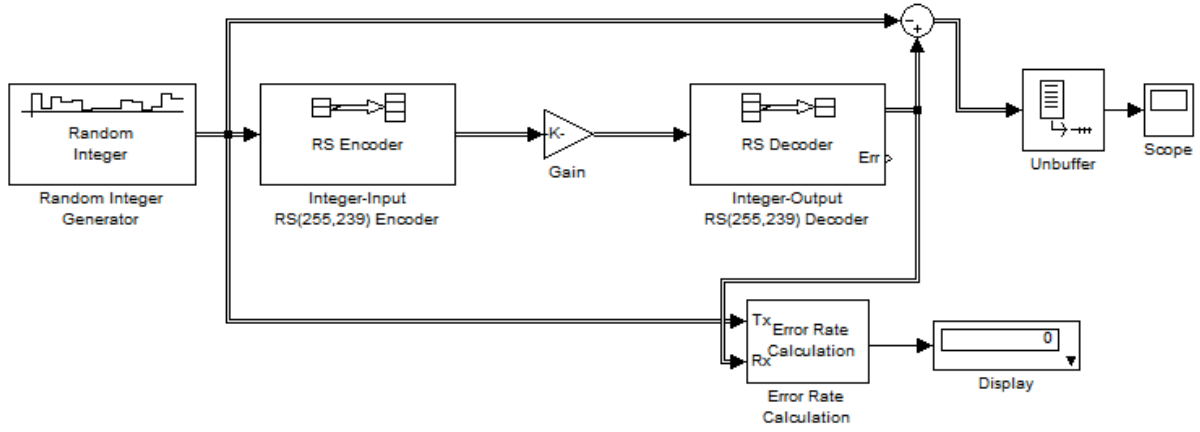


Fig. 3. RS(255,239) elementary system.

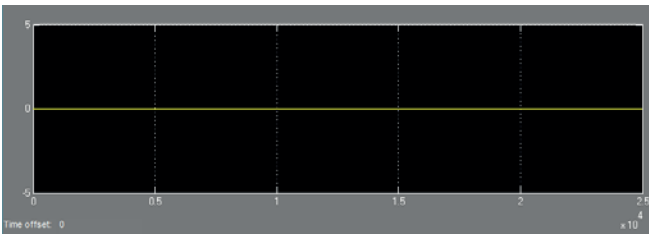


Fig. 4. Scope block plot.

or expressed as $GF(2^8)$ coefficients:

1, 2, 3, ..., 237, 238, 239, 37, 133, 255, 126, 37, 59, 132, 133, 56, 168, 179, 4, 9, 99, 79, 148.

IV. TESTING THE RS(255,239) CODE

RS(255,239) encoding and decoding process was designed and tested in Simulink. Fig. 3 shows the elementary building blocks which are configured in order to perform RS(255,239) coding, without adding noise in the transmitted information. This model will be upgraded in the next section.

Random Integer Generator block produces information symbols which need to be encoded. Mary number value is set to 255 and samples per frame are to 239. The sample time is 1 second for every information symbol. Integer-Input RS Encoder encodes the information symbols, by adding 16 parity symbols. The codeword length value, N , is set to 255, and the message length, K , is set to 239. The primitive polynomial is specified as $[1\ 0\ 0\ 0\ 1\ 1\ 1\ 0\ 1]$, which is binary representation of $p(x) = x^8 + x^4 + x^3 + x^2 + 1$, while the generator polynomial is calculated with specified Matlab function `rsgenpoly(255,239)`. Gain block is set to ones for the first 239 coefficients and the other 16 parity symbols are set to zeros.

Integer-Output RS Decode block has identical settings as the encoder. The Scope block shows the difference between the original message and the recovered message. As can be seen from Fig. 4, the plot is zero, which confirms that the

decoder successfully corrected the errors. The calculated error rate between the information data and decoded data is also zero.

V. MODULATION ADDED

The previously generated Reed-Solomon scheme for coding and decoding is a simple scheme, which does not assume simulation of real physical medium.

In the next implementation of Reed-Solomon coding and decoding process, two main blocks will be added: modulation block, as well as AWGN channel block (Fig. 5). The generated codeword produced by the RS Encoder module, is passed to a digital modulator. The purpose of the modulator is to map group of bits into one symbol to be transmitted, thus increasing the baud rate.

AWGN Channel module is used as a physical medium through which data transmission is conducted. AWGN channel adds white Gaussian noise to the input signal [7]. Fig. 6 shows RS(255,239) code operating with 256-QAM modulation scheme. System simulation parameters are defined as in Table 2.

TABLE II
RS(255,239) SYSTEM SIMULATION PARAMETERS

Property	Value
Sample time in seconds	1
Codeword length (n)	255
Message length (k)	239
QAM order	256
Bits per symbol	8
Uncoded Eb/No	15

The period of each frame in the model is set to 239 seconds, which corresponds to a sample time of 1 second for every information symbol produced at the output of the Random Integer Generator. The symbol time at the output of the Integer-Input RS Encoder is reduces by a factor of the code rate, because 255 symbols are produced over a frame of 239 symbols.

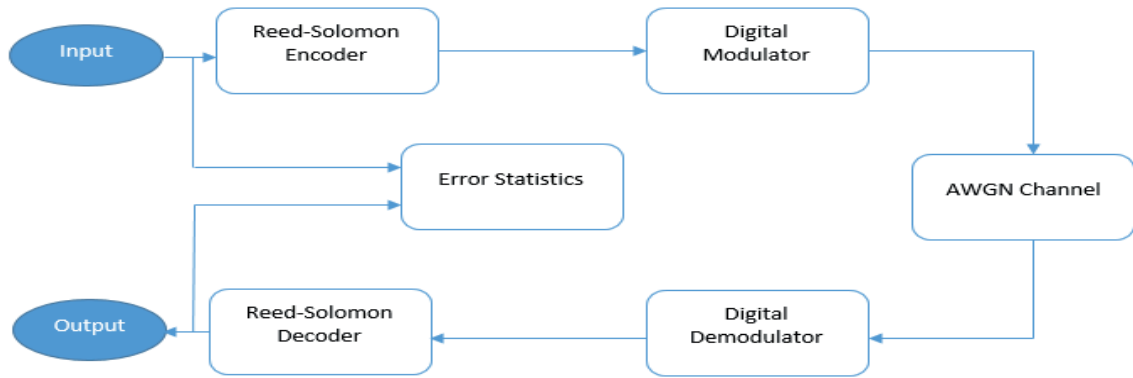


Fig. 5. RS(255,239) main coding and modulation blocks.

The parameters of AWGN channel are defined as in Table 3.

TABLE III
RS(255,239) AWGN CHANNEL PARAMETERS

Field	Description and value
Mode	Signal to noise ratio (Eb/No) [8]
Eb/No (dB)	Ratio of information bit energy to noise power spectral density (without channel coding), in decibels
Number of bits per symbol	8
Input signal power	Power of the symbols at the input of the block. This field value is calculated as: $\text{Uncoded Eb/No} + 10 \cdot \log_{10}(k/n)$
Symbol period	The duration of an information channel symbol (without channel coding), in seconds. This field is set to 1

Two bit error rate calculations are performed. The first one is coded BER, which compares the message prior to entering the RS encoder and the message after being decoded by RS decoder. The second one is channel BER, which compares the coded message, after being encoded by the RS encoder, and the received message, before being decoded by the RS decoder.

In Fig. 6 we can see that the RS decoder successfully corrected certain number of errors, which produces no errors in coded BER block, after the message is being decoded. This is shown by the zero value on the plot produced by the

Scope module (Fig. 3). The channel BER gives the number of errors generated after the data transmission, but before the decoding process. The channel BER is calculated by the Error Rate Calculation Block (Fig. 6). It can be concluded that all errors produced during transmission were successfully corrected by the RS decoder.

IEEE 802.11ad standard suggest using the shortened version of RS(255,239) as coding technique in Low-Power Single Carrier. Shortening means removing symbols from the information message before entering the encoder, not transmitting them and then re-inserting them at the decoder.

For the shortened version of the RS(255, 239) code, which is RS(224,208), the first 31 terms in information polynomial in (8) are set to zero and only 208 information bytes and 16 parity bytes are transmitted.

In Simulink, shortening the code is incorporated in RS Encoder and RS Decoder blocks. To shorten the RS(255,239) code into RS(224,208) code, the codeword length value, n , needs to be set to 224, and the message length value, k , needs to be set to 208.

VI. CONCLUSION

In this paper, the complete process of implementing RS(255,239) code was described. The system was constructed and simulated using Simulink, showing the

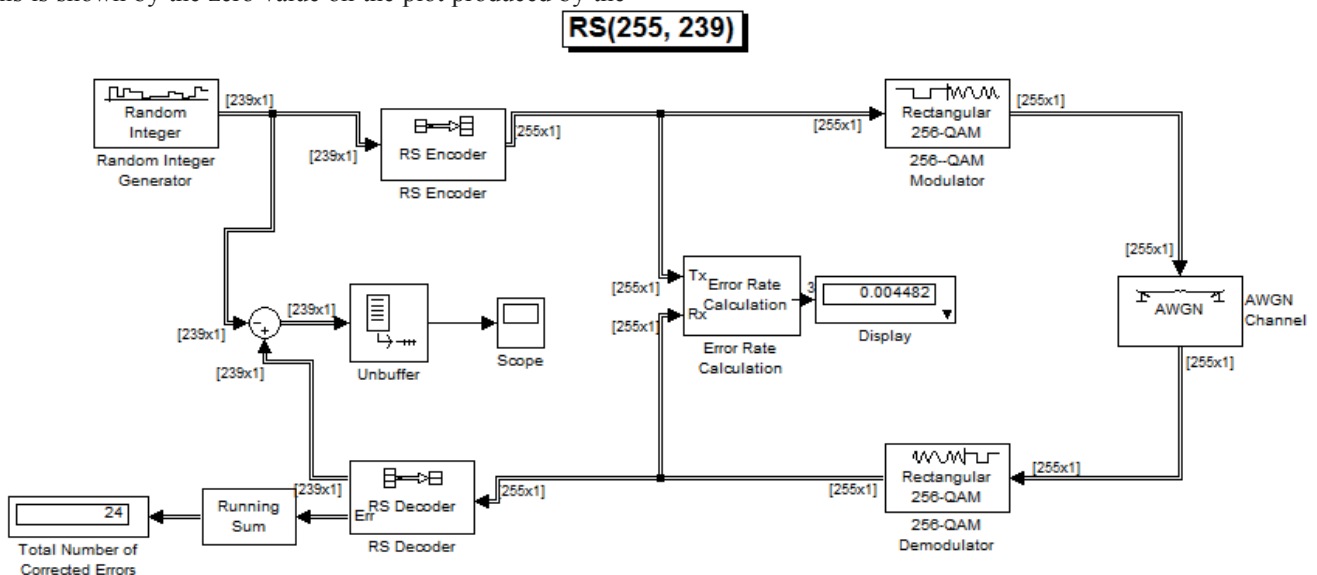


Fig. 6. RS(255,239) coding and modulation process fully implemented in Simulink.

correct implementation of the encoding process. This implementation will play big role in implementing complete IEEE 802.11 ad Low-Power Single Carrier system. Future improvements will be conducted, in terms of research how minimizing the n-k factor will affect on the performance of the code and still maintaining efficient error correcting capabilities.

REFERENCES

- [1] W. W. Pwterson, E. J. Weldon, Jr., "Error Correcting Codes," 2nd ed., MIT Press, Cambridge, Massachusetts, 1972.
- [2] "Wireless LAN at 60 GHz – IEEE 802.11ad Explained," Application Note, Agilent Technologies.
- [3] S. B. Wicker and V. K. Bhg\argava, "Reed-Solomon Codes and Their Application," IEEE Press, 1983 .
- [4] F. J. MacWilliams and N. J. A. Sloane, "The Theory of Error Correcting Codes, Part I," North-Holland Publishing Company, Amsterdam, New York, Oxford, 1977.
- [5] E. A. Berlekamp, "Algebraic Coding Theory," McGraw-Hill, New York, 1968.
- [6] B. Sklar, "Digital Communications: Fundamentals and Application," 2nd ed., Prentice Hall, 2001.
- [7] J. P. Odenwalder, "Error Control Coding Handbook," Linkabit Corporation, San Diego, CA, July 15, 1976.
- [8] J. Hagenauer and E. Lutz, "Forward Error Correction Coding for Fading Compensation in Mobile Satellite Channels," IEEE JSAC, vol . SAC-5, no. 2, February1987, pp. 215-225.

Parallel Architecture Prototype for 60 GHz High Data Rate Wireless Single Carrier Receiver

Tatjana Chavdarova, Aristotel Tentov, Marija Kalendar

SS. Cyril and Methodius University - Faculty of Electrical Engineering and Information Technologies

Karpos II bb, PO Box 574, 1000 Skopje, Macedonia

E-mail: {tatjana.toto,marijaka}@feit.ukim.edu.mk

Abstract—Nowadays a huge attention of the academia and research teams is attracted to the potential of the usage of the 60 GHz frequency band in the wireless communications. The use of the 60GHz frequency band offers great possibilities for wide variety of applications that are yet to be implemented. These applications also imply huge implementation challenges. Such example is building a high data rate transceiver which at the same time would have very low power consumption. In this paper we present a prototype of Single Carrier - SC transceiver system, illustrating a brief overview of the baseband design, emphasizing the most important decisions that need to be done. A brief overview of the possible approaches when implementing the equalizer, as the most complex module in the SC transceiver, is also presented. The main focus of this paper is to suggest a parallel architecture for the receiver in a Single Carrier communication system. This would provide higher data rates that the communication system can achieve, for a price of higher power consumption. The suggested architecture of such receiver is illustrated in this paper, giving the results of its implementation in comparison with its corresponding serial implementation.

Keywords: 60 GHz, millimeter wave, single carrier, VHDL implementation, frequency domain equalization, wireless personal area network.

I. INTRODUCTION

Although the wireless technologies have been significantly improved and widely used in the past few years, the demand for high-definition (HD) video streaming, file transfer, wireless docking station, gaming, short-range backhaul, wireless desktop, and wireless Gigabit Ethernet has also increased.

In order to respond accordingly to the needs of such huge application domain, a complete communication system with very high data rates (above 5 Gb/s) has to be designed. The comparison of the desired data rates, with those being achieved while using the 2.4 GHz and 5 GHz frequency bands, implies that a different approach has to be considered. Particularly attractive and promising candidates for multi Gb/s wireless transceiver systems are the millimeter-wave technologies providing up to 7 GHz unlicensed spectrum, which is available in many countries worldwide. The frequency bandwidth at 60 GHz is one of the largest unlicensed bandwidths ever to be allocated, offering the great potential in terms of capacity and flexibility [1].

When using higher frequencies, as a result of the increased free space path loss (21.6 dB worse than at 5GHz), the propagation of the carrier wave is very short. Therefore, the low-power transmissions will not propagate very far. However, considering that the 60 GHz band still confines the operation within a room in indoor environment, and that this actually reduces the likelihood of co-channel interference (and thus opens the possibility of higher frequency re-use density), this is considered as an advantage [2]. The huge bandwidth available for 60 GHz and UWB systems also simplifies the system design of these technologies. A system with much lower spectral efficiency can be designed to deliver a Gb/s transmission to provide low cost and simple implementation.

The compact size of 60 GHz radio also permits multiple-antenna solutions at the user terminal that are otherwise difficult if not impossible at lower frequencies. The form factor of 60 GHz systems is approximately 140 times smaller and thus can be integrated into consumer electronic products. These advantages led to the development of the 60-GHz millimeter-wave-based wireless standards, that includes the IEEE802.15.3c, IEEE802.11ad, ECMA-387, and ISO/IEC 13156 standards. These standards exploit the 60 GHz frequency band to provide high data rate media streaming, as well as rapid data transfer.

Despite of the various advantages offered, 60 GHz based communications suffer a number of critical problems that must be solved, making the implementation of low cost and low power 60 GHz system a huge challenge.

Two modulations are mainly used within the 60 GHz frequency band: Orthogonal Frequency Division Multiplexing – OFDM and Single Carrier – SC Modulation technique. When using OFDM, conceptually multiple carriers are sent at a specific moment, in parallel, each occupying a narrow frequency band. On the contrary, the SC modulation technique transmits a single carrier to which the symbols are being coded with high symbol rate. These two modulations are preferred choices due to their high bandwidth efficiency, for a cost of higher implementation complexity. Each technique implies important advantages and disadvantages over the other, but in summary the SC modulation is recommended when the system would be mainly deployed in line-of-sight – LOS cases with small delay spread. On the contrary, the OFDM modulation is recommended when the delay spread of the channel is longer. In this paper our focus is on the Single Carrier modulation technique.

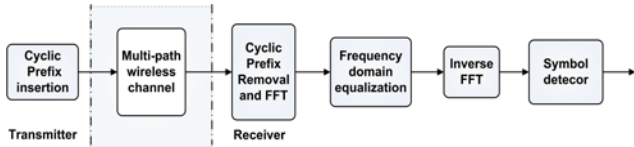


Fig. 1. Main component modules of SC-FDE transceiver.

This paper is organized in several sections, as follows. The second section gives an overview of the main components of a Single Carrier transceiver. Both the transmitter and receiver component modules are being demonstrated, with higher accent to the SC receiver modules as a more complex baseband design, in comparison with the transmitter. In section three, the different types of possible implementations for SC equalizers are listed, emphasizing the advantages and disadvantages of each. In the fourth section we present parallel baseband architecture of the receiver, while choosing a concrete type of the previously illustrated possible implementations of the equalizer. Within this section we present the proportional results between the implemented parallel architecture and the corresponding hardware implementation of the receiver when it is implemented serially. In the following section we conclude the comparison between the two examined architectures. At the end of the paper our future plans for improvement of the presented architecture are listed.

II. SINGLE CARRIER TRANSCIEVER

In this paper we focus on SC baseband modulation with frequency-domain equalization at 60 GHz. The transceiver is designed according to IEEE 802.11ad standards, implemented with VHDL, and so far tested using Xilinx Virtex-5 development board for FPGAs.

In the following sub-sections, each of the conceptual modules of the system is being illustrated, emphasizing its functions in the whole system (illustrated in Fig. 1) and the important implementation aspects.

A. SC transmitter

In SC modulated system, the transmitter has a relatively simple design. Its main functions are to perform symbol modulation, and to form the frame according to the IEEE 802.11ad standard.

1) Symbol modulation

On the physical layer, the stream of bits coming from the upper layers is being grouped in multiple bits that form a single symbol. For this purpose we are using the 64-QAM modulation technique, which transforms 6 incoming bits into one symbol. Each symbol has two components, the in-phase and quadrature component, to which we refer as I/Q components [11]. These components are needed for transmitting the information, since they give the amplitude and the phase changes of the signal that carries the information. Although the standard recommends using less bits per symbol when using SC modulation in combination with QAM (because of the increasing Peak to Average Power Ratio -PAPR of the system), we further explore the 64-QAM modulation with intention to convey more information in one symbol, while suggesting a different

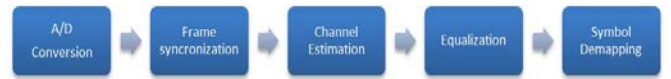


Fig. 2. Main component modules of SC receiver.

approach that would reduce the needed processing during the de-mapping at the receiver (explained in later section).

For this purpose, instead of mapping the bits sequentially, the symbols in the constellation diagram are mapped using Grey codes [12]. This would enable us to de-map each received symbol on bit level, as it would be later illustrated.

2) SC frame format

The format of the frame being composed at this side of the communication, should be followed on the receiving side. Therefore, it is the transmitting side of the communication that, beside the field for frame synchronization, adds in each frame a preamble with additional information, needed on the receiving side. This includes the Golay complementary sequences that comprise the fields of the SC preamble. The main fields included in each SC frame are shown in Fig. 3. Each packet payload consists of 52 symbols. According to the IEEE 802.11ad standard recommendation, the Cyclic Prefix length should be $\frac{1}{4}$ of the packet length. Thus, we choose CP length of 12 symbols, as explained in [6].

B. SC receiver

As illustrated in Figure 6, at the receiving side, first of all, the received carrier analog signal has to be converted into digital signal, using A/D converter with high sampling rate. In order to get valid results, the sampling rate of the A/D converter should be higher than twice the highest frequency that the system under test can pass with significant gain. Additionally, an even higher sampling rate would be recommended, in order to avoid aliasing distortion [9].

Afterwards, using the synchronization fields, prepended at the frame that is being currently received, the receiver determinates the start of a frame. Then, using the separated fields in the preamble of the frame, the receiver is able to do channel estimation. This information is needed when performing the equalization, in order to decrease the BER (bit error rate), which among other factors, increases as a result of the occurred inter-symbol interference in the wireless channel. After equalization, the symbol de-mapping can be performed, forming a stream of bits for the upper layers.

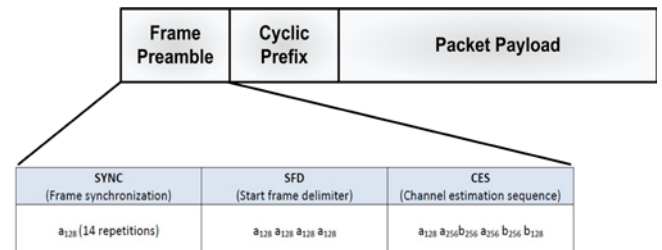


Fig. 3. Structure of a SC frame and preamble according to the IEEE 802.11ad standard [6]. The SYNC field is used for frame synchronization and is consisted of 14 repetitions of the GCS with length 128; The SFD field is used for establishing frame timing and is consisted of 4 GCSs with length of 128; The CES field is used for channel estimation purposes and is consisted of 2 GCSs with length 128 and 4 GCSs with length 256. The exact GCSs are given in [7].

The function of some of the modules illustrated in Fig. 2, as well as their implementation aspects, are explained in more detail in continuance.

1) Golay channel estimator module

More robust system to the channel noise can be implemented if the equalization is performed using information about the channel distortion which is conveyed dynamically, while the system is running. This would make the system more adaptable to the environment, and moreover, such approach has great potential of using it for driving the system in a different operational mode, in order to minimize the power consumption. For example, if the channel impulse response is relatively small in time domain, we could implement a less processing requiring technique that does minimal channel equalization. This leads us to the conclusion that the dynamic determination of the characteristics of the wireless channel is of huge importance, and at the same time, it has great potential for improving the transceiver system.

For these purpose the so called Golay channel estimator has been widely used. The estimator is based on the utilization of the Golay complementary sequences – GCS. GCS are two sequences, whose aperiodic autocorrelations sum to zero in all out-of-phase positions, except in zero [5]. They find huge application mainly because the false peaks cancel each other exactly, and that their autocorrelations can be performed in parallel using a single, hardware efficient and fast correlator [10].

If we have two Golay Complementary sequences, ‘a’ and ‘b’, each of the sequences is convolved with the channel impulse response after sending it through the channel H . When receiving the particular sequence, a Golay correlator is being used, which for known input sequences yields the autocorrelation of one of the sequences which is convolved with the channel impulse response. This is done for both of the sequences. Summing the results of the previous steps would yield the impulse response $h(t)$ of the channel [9].

2) Channel equalization

When using the 60 GHz frequency band, the inter-symbol interference – ISI, which occurs in the wireless channel is much more severe compared with the 2.4 and 5 GHz frequency bands [13], [14]. More specifically, ISI at high frequencies spans over several tens to hundreds of symbol periods. Therefore, the equalization is inevitable part of the baseband processing, and fundamental for reliable data transmission for such channels. The main purpose for performing the equalization is to compensate for the occurred distortion in the channel, which includes ISI. Moreover, the equalization is designed according to the characteristics of the wireless channel, i.e. the channel impulse response.

3) Symbol de-mapping

Since each symbol is mapped with 6-bit binary number that represents a Gray code, we are able to perform the de-mapping of the received symbol into string of bits, on bit

TABLE I
RESULTS OF THE PERFORMED SYNTHESIS OF THE 64-QAM DE-MAPPER

Parameter	Value	Usage
Number of Slice LUTs	24 out of 69120	0%
Number of LUT Flip Flop pairs used:	24	/
Number of IOs	70	10%
Number of bonded IOBs	66 out of 640	/
Maximum combinational path delay	5.403ns (4.094ns logic, 1.309ns route)	/

Summary of the performed synthesis of the 64-QAM de-mapper where the sequence of bits is de-mapped by comparing I and Q values on bit-level. The de-mapper was implemented using VHDL. Utilized FPGA-based platform is Xilinx Virtex-5, with selected device 5vlx110tff1136, and speed grade of -1.

level, as suggested in [12]. This approach is different from the conventional maximum likelihood (ML) algorithm which is usually used as a decision rule. The ML algorithm in this case is impractical since it requires many multiplications, additions, subtractions and comparisons. Therefore, assuming that all bits are equally likely distributed in the constellation plan, and using the Grey codes for each constellation point (as it was afore illustrated), we can implement low complexity demodulator.

This means that with simple comparisons of the bits of I and Q values, we can determine the sequence of bits being sent. Thus, when using 64-QAM, the worst case scenario is performing total 14 comparisons, where several can be performed simultaneously. Therefore, the power consumption of this demodulator is very small, and the de-mapping would require only few cycle periods.

The simplicity of the de-mapping module, when implemented with VHDL, is demonstrated with the results of the synthesis report, summarized in Table 1. The results prove that the device utilization is below 1%.

III. CHANNEL EQUALIZATION

When designing a Single Carrier transceiver system, huge attention should be paid to the type of equalizer that the system would employ. This is a consequence of the fact that the equalizer is the most time requiring and power consuming module of the system. At the same time, the reliability of the system depends on the employed equalizing technique, considering the needed retransmissions that can slow down the system. If the power consumption of the system is of huge importance too, then designing an equalizer with optimal performance could be especially challenging.

There are numerous possibilities when implementing an equalizer for SC modulation. In this section a brief overview of the possible approaches is depicted. The main decision needed to be taken is whether the equalization should be performed in time or in frequency domain. The later implies using Fast Fourier Transform - FFT and Inverse Fast Fourier Transform - IFFT where the equalization is being performed i.e. the receiver.

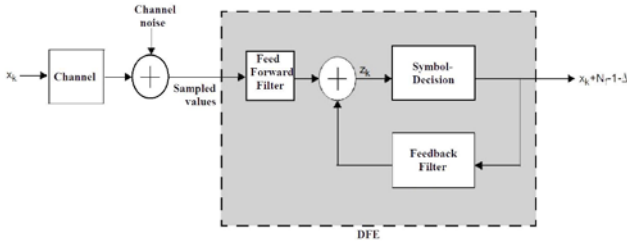


Fig. 4. Block diagram of Decision feedback equalizer – DFE.

A. Time-domain equalization

Since the carrier in an SC transceiver is emitted in time domain, it is intuitive to consider time-domain equalization. The equalizing implemented in time domain is often comprised of two conceptual parts. The first part deals with the precursor ISI, and is referred as linear equalizer -LE. The output of LE is a linear combination (weighted sum) of the received signal and a finite number of previous input values. The cursor is equal to the symbol with the highest amplitude, and its position defines the channel delay [4]. The second part does adaptive equalizing and is conceptually based on the DFE equalizer, explained in later subsection.

Many implementations show that the traditional SC system with adaptive time domain equalization has very high signal processing complexity. This is more emphasized when the length of the channel impulse response exceeds 20 data symbols, which, when using the 60 GHz band, is often the case.

The biggest advantage using this approach is the lack of implementation of the FFT and IFFT required for frequency-domain equalization, which does reduce the power consumption of the system. If this aspect is very important, the time-domain equalization becomes valuable candidate for considering.

B. Frequency-domain equalization

As it is emphasized in [6], when implementing the equalizer in time domain, the complexity of the module is very high. Since the yielded mathematical equations that need to be implemented in hardware are very complex, and thus consume a lot of processing time, often in practice the time-domain approach is avoided for SC systems. Moreover, to get more accurate results the distance between the transmitter and receiver should be known in advance. Since this is very impractical as it conflicts with the practical advantages of using wireless networks from users' perspective, often it is considered worth going into the frequency domain, performing simple one-tap equalization, after which we go back into time domain and perform the symbol de-mapping.

In practice, it is usual to implement the equalizer in frequency domain, in which case, the equalizing operation is reduced to simple multiplication of the incoming data with coefficients that are previously calculated and stored in buffers. When the equalizer is implemented in the frequency domain, its complexity is comparable with the one of an OFDM system. As aforementioned, this implies that an FFT and IFFT are both used at the receiver (Fig. 1), where in between a complex number multiplication is being performed (this multiplication can be optimized as

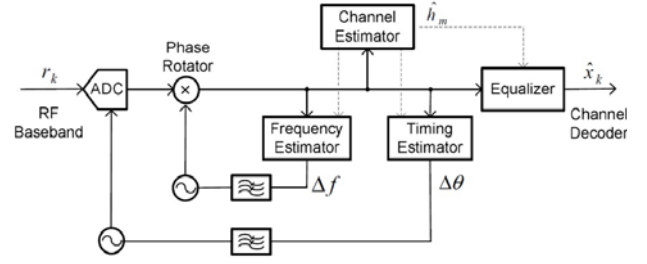


Fig. 5. Diagram of example implementation of a hybrid equalizer, [3].

illustrated in [6]). The values of the coefficients participating in the multiplication are directly connected with the characteristics of the 60 GHz channel.

C. Decision Feedback equalization

To improve the accuracy of an FDE equalizer, very often DFE equalizers are being used. These, so called Decision Feedback Equalizers, use a feedback link which starts from the symbol being detected and affect the values with which we do the multiplication, as illustrated in Fig. 4. The idea of the DFE equalizers is simple: supposing that the detected symbol (using specific demodulation technique) is correct, we can “predict” its effect on the symbols that are yet to be equalized (hopefully corrected) and demodulated.

Since the symbols are being detected after the IFFT operation (thus in time domain) and the equalization is performed in frequency domain, very often in practice, a third FFT module is used in order to improve the precision of the equalizer. The output of the third FFT is used to dynamically change the values of the coefficients that are used to perform the equalization in the frequency domain.

If the power consumption of the system is important too, a different approach is often considered, which is illustrated in [6]. Instead of using a third FFT module, intermediate values are dynamically calculated, and then subtracted from the already equalized (in the first part of the equalizer) values of the received signal. Consequently, the calculation of these values consumes additional energy, but this is proven to be essential, since the part of the equalizer that deals only with the precursor ISI ([13]) does not have good performance in practice, in terms of efficient equalization.

This type of equalizer has lower hardware complexity, comparing it with the TDEs, and thus is often a preferred choice for a SC system. In contrary to the linear equalizer, it does not amplify the channel noise. In other words, it is harder for an error to occur. However, if a symbol is incorrectly detected, the propagation of this error is much bigger, leading to increased number of bit errors.

D. Hybrid equalizers

Considering the various disadvantages and advantages of the aforementioned types of equalizers, very often in practice a hybrid type of equalization is implemented.

Example implementation of such equalizer is demonstrated in Fig. 5, [3]. The equalizer consists of linear equalizer (LE) and two decision-feedback equalizers: a main DFE and a sub-DFE. To cancel the pre-cursor ISI, in this implementation a LE with a limited number of taps is used. The sub-DFE is used to compensate the latency of the loop by limiting the feedback delay to one symbol period.

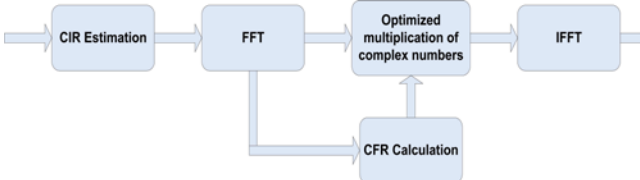


Fig. 7. SC frequency domain equalizer without DFE. The figure illustrates the modules of FDE equalizer, where the coefficients for performing the equalization are dynamically calculated using the CES field of the SC preamble. First, the CIR is calculated, then using FFT the CFR is being calculated, and the results are stored in buffers (which values are re-calculated before storing depending on the equalization algorithm being applied). After the FFT operation of the CP and the Payload of the frame, the complex number multiplication can be performed with the stored FIR values.

IV. PARALLEL ARCHITECTURE OF SC RECEIVER

The results presented in [6] show data rates that are below the goals of our project. Thus, to achieve higher throughput of the SC transceiver, in this paper we propose a different architecture, where the leading idea for performance improvement is hardware parallelization. The aim is to achieve data processing with high frequencies in order to accomplish multi-gigabit data rates. Since we use an FPGA platform, such high frequencies are not achievable even with more advanced FPGA boards. Thus, we propose a design of the receiver in a 60 GHz SC communication system, which has 4-parallel hardware architecture as illustrated in Fig. 6.

Since we present 4-parallel lines of processing, the clock period can be set to $\frac{1}{4}$ of the input symbol rate. In other words, the clocking signal of the ADC (the input symbol sampling rate) is 4 times faster than the signal clocking the rest of the modules. This way, by setting each of the

modules to use the maximal available frequency, we would achieve 4 times bigger throughput of the receiver. This can be implemented if the input symbol sampling rate is not smaller than 4 times the maximal frequency. In the rest of the cases we should use $\frac{1}{4}$ of the input symbol rate as stated afore.

The ADC converted values, which works 4 times faster than the rest of the modules, are stored in reserved buffer. This buffer copies the values of the A/D convertor when there are enough values for computing (4 frames). Then the synchronization is performed according to the sync field in the preamble of each frame, for which purpose we use 4 separate synchronization modules. From this point on, the four frames are being processed in parallel, where at each stage the corresponding field of the preamble is removed, as demonstrated in Fig. 6.

A. Type of equalizer for a parallel SC receiver

As illustrated in the previous section, there are many types of equalizers that can be implemented. Although when the power consumption of the system is important the time-domain equalizers should also be further investigated, in this paper we focus on the implementation of a frequency-domain equalizer, in continuance to our previous work [6].

Since the nature of the DFE approach is serial processing of the data, because the output of the equalizer is fed back to the input of the second part of the equalizer (the part of the equalizer that deals with the post-cursor ISI), this approach contradicts to our intent to increase the data throughput of the system by hardware parallelization. We should also note the disadvantage of this type of equalizer, its long error propagation.

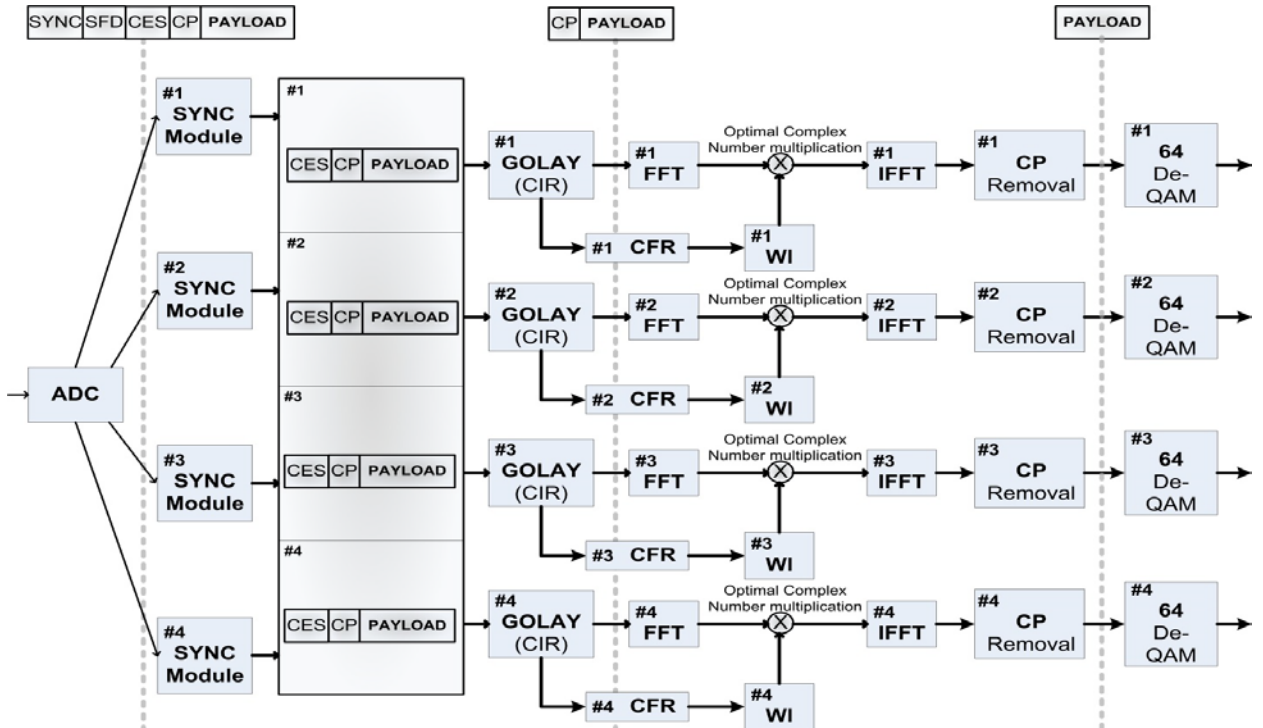


Fig. 6. Quad-parallel SC receiver architecture.

Consequently, in this paper we suggest equalization performed in frequency domain with coefficients that are being dynamically calculated. This topic has already been induced afore, when introducing the Golay complementary sequences, specifically when emphasizing their application in the communication systems. By using the GCS, we can dynamically determine the multipath channel impulse response – CIR information. By knowing the CIR, we would be able to calculate the multipath channel frequency response – CFR, and multiply the output of the FFT with these values (as illustrated in Fig. 7).

$$H(4k + j) = \frac{1}{2}(FC_{ra}(4k + j) + FC_{rb}(4k + j)) \quad (1)$$

The CFR calculation can be performed as demonstrated in [15]. The Golay correlator, based on the GCSs that the preamble of each frame consists of, calculates the cross-correlation values $C'_{ra}(4n+j)$ and $C'_{rb}(4n+j)$, $j=0\sim3$. At each stage in the Golay correlator the output is 1 bit shifted and $C'_{ra}(4n+j)$ and $C'_{rb}(4n+j)$ are sent to the FFT to calculate their frequency domain equivalent values ($FC'_{ra}(4k+j)$ and $FC'_{rb}(4k+j)$). The final CFR values are average of the two at each frequency index (as illustrated in equation 1). Once the CFR is being calculated, the linear equalization algorithm can be performed, yielding the final equalizing W_1 values, where $l=0,1,\dots,63$.

B. Parallel data processing

From the four frames, that are processed in parallel, their CES fields from their preambles are firstly removed. The CES field is used by the Golay correlator in order to determine the channel impulse response. The afore-illustrated equalizer implies that it has two phases. First of all, the FFT should be used for the calculation of the needed W_1 coefficients. If the Minimum Mean Square Error - MMSE equalization algorithm is applied, the FIR values should be normalized and inverted before storing them [18]. Otherwise, if the Zero Forcing – ZF algorithm is being used, the reciprocal of the CFR has to be calculated. Afterwards, the receiver would be able to perform the equalization.

In order to complete the equalization, the FFT first takes the rest of the frame (the CP and the payload) and performs the FFT operation, by which point we have the frequency equivalent of the received frame. As emphasized in [6], the CP must not be removed before the FFT operation is done,

TABLE II
RESULTS OF THE PERFORMED SYNTHESIS OF THE IMPLEMENTED PARALLEL ARCHITECTURE

Parameter	Value
Maximum frequency	229.609 MHz
Maximum combinational path delay	0.55ns

The results are referring to the FFT, the optimized multiplication of complex numbers [6], and the IFFT, of the illustrated parallel architecture.

when working in SC mode. The final step of the calculation of the W_1 coefficients can be performed simultaneously with the FFT calculation of the CP and the payload.

TABLE III
COMPARISON BETWEEN SERIAL AND PARALLEL ARCHITECTURE BASED ON THE SYNTHESIS REPORT

Parameter	Serial architecture [6]	Parallel architecture
Maximum frequency	229.609 MHz	229.609 MHz
Maximum combinational path delay	0.55ns	0.55ns
Minimum input arrival time before clock	2.291ns	3.189ns
Maximum output required time after clock	3.259ns	3.259ns
Total number of paths / destination ports	197005 / 19625	563772 / 70204
Delay	4.355ns	4.355ns
Cell:in->out -> fanout	13	24
Cell:in->out -> Gate	0.818ns	0.818ns
Delay		
Cell:in->out -> Net Delay	0.546ns	0.832ns

Summary comparison between the serial and parallel architectures. For this purpose only the FFT, complex number multiplication and IFFT of both types of architecture are compared. The implementation has been performed by using VHDL, Virtex-5 FPGA board, and 5v1x110tff1136 as selected device with speed -1.

When the FFT operation is done, the multiplication of the W_1 coefficients with the output of the FFT can be performed. Since we have multiplication of complex numbers, this operation can be optimized as illustrated in [6]. The result of the multiplication is again converted into time domain and stored into buffers.

The equalized data can then be de-modulated using four de-modulators that perform the corresponding demodulation technique. By this operation from each complex pair of I/Q values we get the equivalent of the detected symbol. In case of 64-QAM, this would yield a sequence of 6 bits, that is stored in a buffer.

C. Performance analysis of the parallel architecture

As expected, the parallelization improves the performance of the system in aspect of the processed data per second (the throughput of the system), for a price of increasing the power consumption. The results of the performed synthesis of the part of the parallel architecture, which is consisted of the FFT, the optimized multiplication of complex numbers [6] and the IFFT, are presented in Table 2.

In order to compare the parallel architecture with its serial counterpart demonstrated in [6], we only consider the receiver where the equalization is performed. Both architectures represent different types of equalizers, which have in common that they perform the equalization in the frequency domain. Therefore, the comparison only makes sense if we compare only the part of the equalizers consisting of the FFT, the complex number multiplication and the IFFT.

The proposed architecture was implemented using VHDL and the Xilinx Virtex-5 FPGA based platform. The results of the synthesis reports for both architectures are summarized in Table 3. For both types of the architecture the used device was 5v1x110tff1136, and selected speed grade of -1.

The summary of the device utilization, comparing the serial and parallel architecture is shown in Fig. 8:

Device utilization summary:			
Selected Device : 5v1x110tff1136-1			
Slice Logic Utilization:			
Number of Slice Registers:	11749	out of 69120	16%
Number of Slice LUTs:	10688	out of 69120	15%
Number used as Logic:	7471	out of 69120	10%
Number used as Memory:	3217	out of 17920	17%
Number used as RAM:	680		
Number used as SRL:	2537		
Slice Logic Distribution:			
Number of LUT Flip Flop pairs used:	13591		
Number with an unused Flip Flop:	1842	out of 13591	13%
Number with an unused LUT:	2903	out of 13591	21%
Number of fully used LUT-FF pairs:	8846	out of 13591	65%
Number of unique control sets:	293		
IO Utilization:			
Number of IOs:	130		
Number of bonded IOBs:	130	out of 640	20%
Specific Feature Utilization:			
Number of BUFG/BUFGCTRLs:	1	out of 32	3%
Number of DSP48Es:	22	out of 64	34%

Device utilization summary:			
Selected Device : 5v1x110tff1136-1			
Slice Logic Utilization:			
Number of Slice Registers:	25292	out of 69120	36%
Number of Slice LUTs:	21236	out of 69120	30%
Number used as Logic:	17828	out of 69120	25%
Number used as Memory:	3408	out of 17920	19%
Number used as RAM:	1184		
Number used as SRL:	2224		
Slice Logic Distribution:			
Number of LUT Flip Flop pairs used:	22016		
Number with an unused Flip Flop:	-3276	out of 22016	-14%
Number with an unused LUT:	780	out of 22016	3%
Number of fully used LUT-FF pairs:	24512	out of 22016	111% (*)
Number of unique control sets:	416		
IO Utilization:			
Number of IOs:	514		
Number of bonded IOBs:	514	out of 640	80%
Specific Feature Utilization:			
Number of BUFG/BUFGCTRLs:	1	out of 32	3%
Number of DSP48Es:	152	out of 64	237% (*)

WARNING:Xst:1336 - (*) More than 100% of Device resources are used

Fig. 8. Device utilization comparison of the synthesis reports for the serial (on the left side) and the parallel (on the right side) receiver architecture, respectively.

As can be noticed, many of the resources are not increased by a multiple of four after the synthesis. The number of used slice registers and LUTs is about twice higher than the same of the serial architecture, instead of four times higher, as it was primarily expected.

V. CONCLUSION

The 60 GHz band represents a great technology for building multi-Gb/s wireless systems for indoor communications. Along with the millimeter-wave technologies the SC and OFDM modulations are widely used. In this paper, we focus on the SC modulation technique, where the carrier is occupying the whole frequency band, and thus more information is conveyed.

In this paper, we enumerate the modules from which an SC transceiver is consisted of, while focusing on the equalizer, as the most challenging module when designing the SC system. The possible types of implementations of the equalizer are briefly listed.

To achieve high data rates, we propose quad-parallel hardware architecture of the receiver. The results of the performed synthesis of such architecture, which was implemented on an FPGA board, are also presented, in comparison with the corresponding serial implementation.

Important conclusion that can be drawn from the presented results in this paper is that the used resources do not increase by a multiple of 4 as expected. Therefore, we suggest that further improvement of the presented design is definitely worth considering even in cases when the power consumption is important. Furthermore, the parallel architecture is essential for achieving data rates of 5 Gb/s and beyond.

Since the final transceiver would potentially employ both modulations (OFDM and SC), and moreover, since for the OFDM modulation the FFT is essential part of the receiver, we consider that the approach of implementing the equalization in the frequency domain as quite reasonable. This is based on the idea of module reutilization. In other

words, the design can be such that when the transceiver works in SC or in OFDM mode it will use the same modules, whenever possible.

VI. FUTURE PLANS

Although this approach potentially offers great advantages, which are crucial for achieving one of the goals of our project - the high data rates, it also introduces potential disadvantages which implies that it should be further measured, compared with other implementations, and accordingly improved.

Our future plans that refer to the above suggested architecture include using a FFT module that does the transform using parallel architecture within the module itself. This is a promising approach, since there is lot of research done in this area which does improve the FFT performance significantly [16], [17].

The tradeoff between the power consumption and the improvement of the speed of data processing when implementing additional FFT should be also investigated. Since the presented architecture has two phases, one in which the equalizing coefficients are calculated, and another in which the equalization is performed, namely we can improve the performance if additional FFT is used. Meanwhile, during the calculation of the FIR (from the CIR) and its normalized and inversed values, we can calculate the frequency equivalent of the CP and the payload of the frame (with the other FFT). After the two FFT modules finish the conversion, we can perform the second phase, the complex number multiplication, by which we would decrease the processing time.

Since the results presented in this paper are promising, our future plans include increasing the level of parallelization. Instead of having quad-parallel architecture, we can implement higher parallelization achieving higher data rates. Here, an important limitation is the frequency of the ADC.

As mentioned, important consideration is the reutilization of the implemented modules. Each of the two modulations has disadvantages and advantages depending on the scenario under which the system works. The dynamic switching between the two modulation techniques is offering great improvement of the system. Thus, our future plans include implementation of a design able to share the modules between the two supported modulations.

ACKNOWLEDGMENT

This work was partially supported by the ERC Starting Independent Researcher Grant VISION (Contract n. 240555).

REFERENCES

- [1] S. Yong, P. Xia, and A. Valdes-Garcia, "60 GHz technology for Gbps WLAN and WPAN: from theory to practice," ISBN 9780470972939, 2011.
- [2] Agilent Technologies, "Wireless LAN at 60 GHz – IEEE 802.11ad Explained : Application Note," 5990-9697EN, USA, 2013.
- [3] J. Park, B. Richards, and B. Nikolic, "A 2-Gb/s 5.6-mW Digital Equalizer for a LOS/NLOS Receiver in the 60GHz Band," IEEE Asian Solid-State Circuits Conference, China, Nov. 2010.
- [4] S. Binggeli, "Fractionally Spaced Equalizer for NLOS Receiver in the 60 GHz Band," Master Thesis, Berkeley University of California, 2011.
- [5] M. G. Parker, K. G. Patersony, and C. Tellamburaz, "Golay Complementary Sequences," Jan. 2004.
- [6] T. Chavdarova, G. Jakimovski, B. Jovanov, A. Tentov, and M. Malenko, "Analysis and implementation of frequency domain equalizer for single carrier system in the 60 GHz band," presented at 9th Annual International Joint Conferences on Computer, Information, Systems Sciences, and Engineering, CISSE Online E-conference, Dec. 12-14 2013 (to be published).
- [7] IEEE P802.11 Wireless LANs, "Complete Proposal for 802.11ad," doc: IEEE 802.11-10-0499-02-00ad, May 2010.
- [8] IEEE 802.15.3c (2009), "IEEE Standard for information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements. Part 15.3: Wireless medium access control (MAC) and physical layer (PHY) specifications for high rate wireless personal area networks (WPANs) Amendment 2: Millimeter-wave-based alternative physical layer extension," [Online]. Available: <http://www.ieee802.org/15/>
- [9] Scott Foster, "Impulse Response Measurement Using Golay Codes," Trans. ICASSP '86, pp.929-932, 1986. [Online]. Available: http://www.researchgate.net/publication/224737472_Impulse_response_measurement_using_Golay_codes
- [10] A. Hernández, J. Ureña, D. Hernanz, J. J. García, M. Mazo, J. Dértin, J. Serot, and S. E. Palazuelos, "Real-time implementation of an efficient Golay correlator (EGC) applied to ultrasonic sensorial systems," March 27 2003. [Online]. Available: <http://www.geintra-uah.org/system/files/private/a2003.pdf>
- [11] National Instruments, "What is I/Q Data?" Sep 12, 2013. [Online]. Available: <http://www.ni.com/white-paper/4805/en/>
- [12] Fung, Andy W. Y., "Suboptimal soft-bit level demapper for M-QAM-OFDM systems," Hong Kong, Dec. 2010.
- [13] C. Langton, "Inter Symbol Interference (ISI) and raised cosine filtering," 2002.
- [14] H. Xu, V. Kukshya, and T.S Rappaport, "Spatial and temporal characteristics of 60-GHz indoor channels," IEEE J. Selected Areas in Communications, vol. 20, pp.620-620, Mar.2002.
- [15] F. Hsiao, A. Tang, D. Yang, M. Pham, M. F. Chang, "A 7Gb/s SC-FDE/OFDM MMSE Equalizer for 60GHz Wireless Communications," IEEE Asian Solid-State Circuits Conference, Korea, Nov. 2011.
- [16] Miloš Krstić, Maxim Piz, Marcus Ehrig, Eckhard Grass, "OFDM Datapath Baseband Processor for 1 Gbps Datarate," IHP, Im Technologiepark 25, Frankfurt, Germany.
- [17] Somasundaram Meiyappan, "Implementation and performance evaluation of parallel FFT algorithms," Matriculation No.: HT023601A, National University of Singapore, Singapore.
- [18] K. Takeda and F. Adachi, "Frequency-Domain MMSE Channel Estimation for Frequency-Domain Equalization of DS-CDMA Signals," IEICE Trans. Communications, vol. E920-B, pp1746-1753, July 2007.

Assessment of the VoIP Transmission Quality over Digital Power Line Carrier Channels

Anton Merkulov, Viatcheslav Shuvalov
Siberian State University of Telecommunications and Informatic Sciences –
Faculty of Automatic Electrocommunication
Kyrova Str. 86, 630102 Novosibirsk, Russia
E-mail: anton-merkulov@list.ru, shwp@sibsutis.ru

Abstract—This article is devoted to the research of VoIP transmission quality over Digital Power Line Carrier channels. Assessment of quality transmission is performed using E-model. Paper considers the possibility of joint using of Digital Power Line carrier equipment with different architecture in one network. As a result of the research, the rule for constructing of multi-segment Digital Power Line Carrier channels was formulated. This rule allows minimizing the transmission delay and saving frequency resources of high voltage Power Line Carrier range.

Keywords: E-model, high voltage Power Line Carrier communication, IP networks

I. INTRODUCTION

The use of phase conductors of high voltage lines as a transmission medium of information signals between electrical substations began since 30th years of last century. This type of communication is widely known as Power Line Carrier communication (PLC). A little more than ten years ago, with the development of microprocessor technique, in the PLC equipment signal processors started to be used. It allowed to expand functions of devices and to create Digital Power Line Carrier (dPLC) systems. In modern dPLC equipment the methods of modulation such as QAM/TCM, MCM, OFDM, COFDM are used. DPLC channels are widely used on HVL of 35 kV, 110 kV and 220 kV. In spite of the fact that data rate in dPLC does not exceed few tens kilobit per second, low cost, fast deployment and simplicity in maintenance provide it popularity in the modern power utilities.

Talking about dPLC application the networks built by SIEMENS AG with the use of Frame Relay Access Devices (FRAD) as node equipment is the prime example of successful technical solution. DPLC Frame Relay networks were organized in the period of 2002-2009 in the countries of Latin America, Asia, Africa and CIS. In most projects, the author was personally involved. Unfortunately, with transition to IP platform the life cycle of the dPLC Frame Relay networks comes to the end. Outdated Frame Relay, removal of Frame Relay multiplexer production, and IP traffic transmission requirements updates led to the need to shift from Frame Relay technology to IP technology. For creation of convergent IP-PLC channels assessment of voice transmission quality is very important. Let's observe questions related to the assessment of VoIP transmission

quality in dPLC networks taking into account characteristics of applied dPLC equipment and quantity of segments in multi-segment communication channels (MSCC).

II. ASSESSMENT OF VOIP TRANSMISSION QUALITY OVER DPLC CHANNELS WITH APPLICATION OF THE E-MODEL

For transmission quality assessment, we use E-model. This method is described in ITU-T recommendation G.107. It is possible to find examples of using E-model in different sources. Examples of calculations of transmission rating factor R and value of mean opinion score – MOS applicable for dPLC channels are shown in [1].

The main difference of the materials represented in this paper is the accounting of joint use of the dPLC equipment with different architecture in one network. Calculation was made according to the last edition of the document ITU-T: G.107 (12/2011) "The E-model: computational model for use in transmission planning"

The main objective of calculations is search of value of factor R . According to [2] R is calculated by formula (1):

$$R = R_0 - I_S - I_d - I_{E-eff} + A \quad (1)$$

where

R_0 – factor represents basic signal-to-noise ratio;

I_S – factor represents combination of all impairments which occur more or less simultaneously with the voice signal;

I_d – factor represents impairments caused by delay;

I_{E-eff} – factor represents impairments caused by application of low bit-rate codecs;

A – advantage factor for compensation of voice quality degradation for specified system in comparison with conventional systems.

When we calculate R_0 , using default values defined in [2], but at the same time taking into account an increased noise level on the substation, R_0 is equal to 91,2.

In dPLC networks low bit-rate codecs like ACELP and MP-MLQ are used and factor I_S characterizing quantization errors will be equal to 0.

The factor I_d in formula (1) depends on characteristics of dPLC equipment. For dPLC terminals where QAM/TCM modulation is used the delay of information processing T_{PLC} almost does not depend on channel bandwidth. But for dPLC terminals with OFDM modulation the bandwidth is factor of high impact. It occurs due to the internal features of system architecture of OFDM devices. Typical values of T_{PLC} for modern QAM/TCM and OFDM systems taken

from [3][4][5][6] are shown in Table 1. We should note that in practice dPLC links with the bandwidth of 4, 8 and 12 kHz are the most popular.

TABLE I
TYPICAL VALUES OF T_{PLC} FOR QAM/TCM AND OFDM dPLC EQUIPMENT

Bandwidth, kHz	T_{PLC} QAM/TCM, ms	T_{PLC} OFDM, ms
4		160
8	60	120
≥ 12		40

I_d is calculated by formula (2) [2]:

$$I_d = I_{dte} + I_{dle} + I_{dd} \quad (2)$$

where

I_{dte} – factor of impairments due to talker echo;

I_{dle} – factor of impairments due to listener echo;

I_{dd} – factor of impairments due to long absolute delay T_a .

Application of echo cancellers in dPLC networks allows to exclude impairments (appeared because of echo) even for absolute delay more than 400 ms, therefore I_{dle} and I_{dte} are equal to 0.

The last item of I_{dd} is calculated using formulas (3) and (4) [2]:

$$I_{dd} = 25 \left\{ \left(1 + X^6 \right)^{\frac{1}{6}} - 3 \left(1 + \left[\frac{X}{3} \right]^6 \right)^{\frac{1}{6}} + 2 \right\} \quad (3)$$

$$X = \frac{\log \left(\frac{T_a}{100} \right)}{\log 2} \quad (4)$$

In order to calculate I_{dd} firstly we need find the value of T_a . The diagram of time delay sources in dPLC channel is shown in Fig. 1.

$$T_a = T_{voice} + 2 \cdot T_{Router} + T_{PLC(MD)} + T_{(link)} + T_{PLC(DMD)} + T_{serial} + T_{de-JIT} \quad (5)$$

where

T_{voice} – delay of voice processing including algorithmic delay of voice codec and packetization delay;

T_{Router} – internal delay in router;

$T_{PLC(MD)}$ – internal delay in dPLC terminal during signal processing in the transmission path;

$T_{(link)}$ – time of HF signal propagation via HVL;

$T_{PLC(DMD)}$ – internal delay in dPLC terminal during signal processing in the receiving path;

T_{serial} – serialization delay;

T_{de-JIT} – delay of de-jitter buffer.

Delay of voice processing defined as (6) [7]:

$$T_{voice} = [T_{frame} \cdot (N + 1) + T_{look-ahead}] \quad (6)$$

where T_{frame} – voice frame duration;

N – number of voice frames per packet;

$T_{look-ahead}$ – prediction delay.

Popular voice codec G.723.1 uses band-limited input speech signal sampled at 8000 Hz, converts it into a 16-bit linear PCM signal, at the same time operating 240 samples. Therefore, each frame corresponds to 30 milliseconds of speech signal, that is why $T_{frame} = 30$ ms. Also codec has a prediction delay $T_{look-ahead} = 7.5$ ms. As result total delay for $N = 1$ is equal to 67.5 ms.

Basic ITU-T specification G.729 describes the voice codec operating at bit-rate 8 kbit/s. This codec uses the input frames of 10 ms corresponding to 80 samples at a sampling rate of 8000 Hz. G.729 has a prediction delay $T_{look-ahead} = 5$ ms., the resulting algorithmic delay of G.729 is 15 ms. By the default, using G.729 codec one IP-datagram includes two speech frames, and total delay is equal to 35 ms.

The delay of packet processing in routers T_{Router} can arise only in case of appearing of queues. In absence of queues internal delay of routers does not exceed several milliseconds. Assume in calculations T_{Router} equal to 3 ms.

Delay of signal processing in dPLC $T_{PLC(MD)}$ and $T_{PLC(DMD)}$ in sum represent T_{PLC} which values was shown in Table 1.

The high frequency signal propagates in HVL approximately with light velocity, therefore value of $T_{(link)}$ can be neglected.

Serialization delay T_{serial} is the time that is necessary for processing of layer 2 frame from the first to the last bit and defines by formula (7) [8]:

$$T_{serial} = \frac{S}{C} \quad (7)$$

where S – size of L2 frame, bits;

C – data rate, bps.

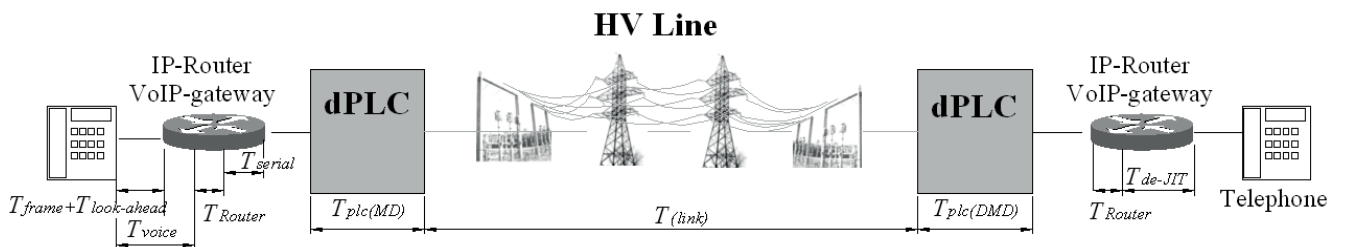


Fig.1. Diagram of time delay sources in dPLC link.

T_{serial} depends on data rate of the communication channel. In T_a calculations we will assume that data rate of the communication channel allows to provide a serialization delay no more than 20 ms while recommended value is 10-15 ms.

In calculations of T_a it is necessary to consider the size of de-jitter buffer T_{de-JIT} . The volume of the buffer shouldn't be too big in vain not to increase a time delay, but also shouldn't be less than required minimum $T_{de-JIT(min)}$. We assume $T_{de-JIT(min)} = 30$ ms.

According to [9], the value of factor I_{E-eff} for codec G.729A+VAD is equal to 11, and for codec G.723.1+VAD is equal to 15.

The last term of (1) is an advantage factor A . It is easy to show that the using data given in [2], the value of the coefficient A can be approximated by the expression (8):

$$A = 0,0444 \cdot T_a - 6,6667 \quad (8)$$

Calculation of MOS is performed with use of (9) [2]:

$$MOS = \begin{cases} 1, & \text{for } R < 0, \\ 1 + 0,035 \cdot R + R \cdot (R - 60) \cdot (100 - R) \cdot 7 \cdot 10^{-6}, & \text{for } 0 < R < 100 \\ 4,5, & \text{for } R > 4,5 \end{cases} \quad (9)$$

Values of R and MOS was derived for G.729A+VAD codec as it has smaller value of I_{E-eff} and algorithmic delay in comparison with G.723.1. The results of calculations are shown in Table 2. The best transmission quality of voice signal is provided using OFDM equipment with bandwidth of 12 kHz and more. For channels with a bandwidth of 4 and 8 kHz for minimization T_{PLC} , it is expedient to use QAM/TCM equipment.

TABLE III
RESULTS OF R AND MOS CALCULATION

dPLC equipment	T_a , ms	T_{PLC} , ms	I_d	A	R	MOS
OFDM (bandwidth 4 kHz)	251	160	9,04	4,48	75,47	3,84
OFDM (bandwidth 8 kHz)	211	120	4,22	2,70	78,51	3,97
OFDM (bandwidth 12 kHz)	131	40	0,01	0,00	80,02	4,02
QAM/TCM (any bandwidth)	151	60	0,18	0,04	79,89	4,02

III. VOICE TRANSMISSION VIA MULTISEGMENT DPLC CHANNELS

For one-segment channels it is possible to provide good quality of speech with dPLC equipment of any configuration and architecture. Difficulties start to arise in case of appearance of transit nodes. In most cases the dPLC

network consists of several multi-segment channels. In practice number of hops in MSCC rarely exceeds 4.

Let's calculate the maximum number of segments M in MSCC depending on used equipment. Calculation will be done considering that fact that VoIP transmission quality for any segment shall not be worse than fair, i.e. $R = 60$ and $MOS > 3$, with T_a not more than 400 ms. Transmission delay for MSCC with number of segment equals M can be found as (10):

$$T_a = T_{voice} + T_{Router} + \sum_{i=1}^M (T_{Router}^{(i)} + T_{plc(MD)}^{(i)} + T_{link}^{(i)} + T_{plc(DMD)}^{(i)} + T_{serial}^{(i)}) + T_{de-JIT} \quad (10)$$

After R and MOS calculation was done with the use of (10) we can make the following conclusion:

1) with the given characteristics, in case of using OFDM equipment with the bandwidth of 4 kHz, it is possible to provide only 1 segment channel;

2) using only OFDM equipment with bandwidth of 8 kHz, it is possible to create 2 segment MSCC with characteristics of the most distant segment $M=2$: $T_a = 354$ ms, $R=68,9$ and $MOS=3,55$;

3) the use of QAM/TCM equipment allows to create 4 segment MSCC with characteristics of the most distant segment $M=4$: $T_a = 400$ ms, $R=67,05$ and $MOS=3,46$;

4) the best results could be obtained with application of OFDM equipment with bandwidth 12 kHz and more. This solution allows to create MSCC with $M=5$;

5) During planning of dPLC network the following question ought to be considered: what type of dPLC architecture should be in MSCC segments in order to reduce voice transmission delay.

IV. RULE OF CREATION OF MULTISEGMENT DPLC CHANNELS

In conclusion we formulate the rule of choice of dPLC equipment for MSCC segments. This rule allows minimizing the transmission delay and saving the frequency resources: "For minimizations of transmission delay if QAM/TCM equipment provides sufficient capacity for defined value of SNR (signal to noise ratio), it needs to be used in segments with bandwidth of $B=4$ and $B=8$ kHz. In other cases it is necessary to use OFDM equipment. For transmission of voice signal through the maximum number of segments OFDM equipment with bandwidth ≥ 12 kHz has to be used".

For applied tasks this rule means: in one dPLC network both OFDM and QAM/TCM equipment should be used. For the most distant segments of MSCC QAM/TCM equipment should be used in channels with bandwidth of 4 and 8 kHz. OFDM equipment should be used in segments with bandwidth of 12 kHz and more. Joint use of the equipment of different architecture provides minimization of transmission delay and higher transmission quality of voice consequently, also saving frequency resource of the high voltage PLC range.

REFERENCES

- [1] S.E. Romanov, "Transmission of Voice Signals over dPLC channels." [Online]. Available: <http://romvchvlcomm.blogspot.com/>
- [2] Recommendation ITU-T G.107 – 2011 The E-model: a computational model for use in transmission planning. 18 p.
- [3] Power Line Carrier PowerLink 100 CSPi Manual. Germany: Siemens AG, 2011. 620 p.
- [4] Power Line Carrier ETL 600 Manual. Switzerland:ABB, 2011. 440 p. Digital Power Line Carrier STE-D technical description. Italy: SELTA, 2011. 38 p.
- [5] Product description of Digital Power Line Carrier Equipment CVK-16. Saints-Petersburg, Russia: NPF MODEM 2012. 83 p.
- [6] Recommendation ITU-T G.108 – 1999 Application of E-model: A planning guide. 130 p
- [7] Recommendation ITU-T G.108 – 1999 Application of E-model: A planning guide. 130 p
- [8] Design Best Practices for Latency Optimization. CISCO, 2006. 8 p.
- [9] Recommendation ITU-T G.109 Amendment 1 – 2007 New Appendix I – The E-model-based quality contours for predicting speech transmission quality and user satisfaction from time-varying transmission impairments. 9 p.

Improving TCP Performance on CSMA/CA Connections

Hao Hu, Dmitry Kachan, Eduard Siemens

Anhalt University of Applied Sciences - Faculty of Electrical, Mechanical and Industrial Engineering

Bernburger Str. 57, 06366 Koethen, Germany

E-mail: [h.hu, d.kachan, e.siemens}@emw.hs-anhalt.de](mailto:{h.hu,d.kachan,e.siemens}@emw.hs-anhalt.de)

Abstract—In IP networks, most of packets, that have been dropped, are recovered after the expiration of retransmission timeouts. These can result in unnecessary retransmissions and needless reduction of congestion window. An inappropriate retransmission timeout has a huge impact on TCP performance. In this paper we have proved that CSMA/CA mechanism can cause TCP retransmissions due to CSMA/CA effects. For this we have observed three wireless connections that use CSMA/CA: with good link quality, poor link quality and in presence of cross traffic. The measurements have been performed using real devices. Through tracking of each transmitted packet it is possible to analyze the relation between one-way delay and packet loss probability and the cumulative distribution of distances between peaks of OWDs. The distribution of OWDs and the distances between peaks of OWDs are the most important parameters of tuning TCP retransmission timeout on CSMA/CA networks. A new perspective through investigating the dynamical relation between one-way delay and packet loss ratio depending on the link quality to enhance the TCP performance has been provided.

Keywords: CSMA/CA, TCP performance, retransmission timeout, wireless.

I. INTRODUCTION

With the popularity of third generation wireless wide-area networks (WWAN), the CDMA technology [1] is fairly mature now. Fourth generation WWAN, basically based on LTE [2][3] and WiMAX [4] technology, are being increasingly deployed throughout the world. Besides 3G and 4G networks, wireless LAN networks are widely used at homes, schools and enterprises. Besides the common telephone speech data, contemporary mobile access networks are increasingly used to convey data traffic to the internet – for private as well as for business users. With this trend, wireless access networks will convey all the burden of personal and business data traffic in the future. Consequently, efficient data transport will become the key for a commercial success of many mission- and business-critical applications. As the most popular transport protocol in internet, TCP performance in such wireless connections is crucial. Media Access Control (MAC) is the core of the data link layer. Although the wireless MAC Layer has defined two types of access methods, Distributed Coordination Function (DCF) [5] and Point Coordination Function (PCF) [6], only the DCF mechanism, which is based on CSMA/CA [7] technology, is used in the most commercial products [8]. So, an increase of TCP

performance on CSMA/CA networks is an important task that will have a broad and significant impact on the exploding mobile based applications.

TCP uses two types of retransmissions: one is based on timeout and the other is based on the receiving of duplicated ACKs. When a timeout based retransmission occurred, TCP sender reduces its Congestion Window Size (CWS) down to one segment and continues transmission within slow-start phase until the CWS reaches the half of the value that was before the timeout. Then the sender enters the congestion avoidance phase. As a reaction on ACKs-duplication based retransmission, the sender only halves its CWS, then immediately enters the congestion avoidance phase. The timeout based retransmissions have bigger impact on TCP performance than the duplicated ACKs. When the value of TCP retransmission timeout is too small, after a timeout, TCP sender assumes that the outgoing packet is lost and the slow-start phase is triggered. However the acknowledgment will be returned back shortly. If the retransmission timeout is too large, it will also affect the TCP performance. This paper is motivated by recent researches [9][10][11]. In [9] and [10], authors have shown that 70% of the dropped packets are recovered after expiration of retransmission timeouts. In addition, the study in [11] has proposed that more than 20% of the retransmission timeouts are caused spuriously by transmission delay of packets which results in unnecessary retransmissions and needless reduction of congestion window size. The idea of setting the retransmission timeout properly is the focus of our further work.

In this work, we have proved that the usage of CSMA/CA mechanism can cause TCP retransmissions and proposed a new perspective to improve the TCP performance on CSMA/CA connections. Within the accomplished experiments the salient part is that tracking each outgoing packet in data link layer with the measurement tool LTest [12] on CSMA/CA connections was realized. The basic assumption hereby is that through analyzing of the relation between one-way delay and packet loss probability, the normal calculation of retransmission timeouts can be adapted to the CSMA/CA characteristic in the future.

II. RELATED WORK

The performance of TCP over wireless networks has been studied intensively in recent decades. Early research [13][14][15][16], showed that high Bit Error Rates (BER) of wireless links affects the TCP performance significantly, since they are designed for wired networks TCP and can't distinguish whether packet losses are caused from congestion or other reasons, and invoke congestion control and congestion avoidance algorithms to respond on packet losses, that is a reason of a degraded performance in wireless connections. The proposal [17] showed the interest in adding SACK mechanism to TCP that provides to the sender sufficient information for quick recovering in presence of multiple packet losses within a single transmission window.

Work in [18][19][20] have reveal, that local retransmissions and error correction have reduced loss ratio on the wireless link. But at the same time the adverse effect of variations in packet transmission delay on upper layer protocol TCP has been brought.

In [21][22][23][24], the impact of data rate and latency variations on TCP performance using wireless links has been evaluated. In [21], an enhancement to TCP's error recovery has been proposed. Hereby, extra information in TCP header is used to eliminate the spurious timeouts, then the load is restored and next unsent packet is retransmitted. In [22] several recommendations like the use of Selective Acknowledgments, Large Window Size and Explicit Congestion Notification and enabling the timestamp option has been presented, to improve the performance of TCP on CSMA/CA connections. In [23], the ACK regulator has been presented as a solution to avoid multiple packet drops at the congested router for long-lived flows. In [24], Window Regulator Algorithm is proposed to maximize long-lived TCP performance in the presence of channel variations for any given buffer size at the congested router.

There are also some studies [25][26][27] proposed analyze of the data link layer to promote the TCP performance on CSMA/CA networks. In [25], authors show that the proper choice of parameters of backoff mechanism has a huge influence on the network performance. The most crucial point is to select minimal Contend Window (CW_{min}) and maximal Contend Window (CW_{max}) depending on the number of stations between the hosts. In [26], a simple self-adaptive Contention Window adjustment algorithm-MIMLD for CSMA/CA has been proposed. It adjusts the initial contention window dynamically to a near optimal point according to the traffic activity. In [27], a novel scheme has been proposed. This scheme is supposed to decrease protocol overhead, that causes the inefficiency of CSMA/CA and further achieve a higher data link rate.

In more recent research [12] and [28], solutions for detecting congestion retransmission timeouts and non-congestion timeouts have been presented. The paper [28] has further differentiated the two different types of non-congestion timeouts, random loss and spurious retransmission timeouts.

Although so many suggestions to improve TCP performance in this field have been proposed, but we have not found researches related to investigation of CSMA/CA

influence on retransmissions and CWS of TCP. This paper is devoted to this topic.

III. TESTBED TOPOLOGY DESCRIPTION

The testbed topology used for the investigations of this paper contains:

- three computers with the CPU Modell Intel Pentium (R) E5500 @2.8GHz, Memory 1890M, under operation System Linux OpenSuSE 12.3, kernel 3.7.10-1.16;
- two CAT.6 SSTP 4X2XAWG27/7 patch cables
- one Linksys wireless broadband router which supports 802.11 b/g, model WRT54GL, firmware v4.30.7;
- two TP-LINK USB wireless adapters supported with 802.11 b/g/n.

The 802.11g supports a maximal link speed of 54 Mbps, however in real tests it is possible to achieve only about 40% of this maximal value. Since we want to track each outgoing packet into the radio link, besides those listed equipment, we need measurement tool LTest and network analysis tool Wireshark installed on these computers. The main goal is to achieve the observation of data link layer. Since LTest runs on application layer and there is partly wired connection, shown in Fig. 1, before performing of the test it is necessary to be sure whether these factors affect the CSMA/CA character or not. Usually the time delay of CSMA/CA is high and changes very often with a high standard deviation. For wired connection the back-off is only applied when the collision occurred, however due to the use of a switched full duplex connection, not collisions can occur at all. The packet delays in the wired Ethernet network is by orders of magnitude less than on the wireless links. LTest measures the one-way delay between two computers. In the wired local network is permanently about 200 μ s. To get rid of the delay that caused not by wireless link, we conducted tests locally on PC and got the latency of 10 μ s. It's obvious that these factors can be omitted when to study the characteristic of CSMA/CA.

The chosen wireless mode is the most common infrastructure mode, which, in comparison with ad-hoc mode offers advantage of stability, scalability, improved

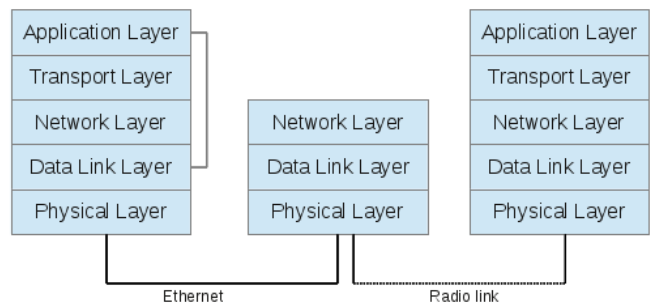


Fig. 1. Network layer topology.

reach and centralized security. The experiment consists of three scenarios:

- connection with good link quality and signal level;
- connection with a bad link quality and signal level;
- connection in presence of cross traffic

These three cases allow emulating the most common types of wireless environment. Fig. 2 shows the topology for scenario 1 and scenario 2. This topology allows observing how CSMA/CA works on each packet, because

here CSMA/CA is only used in the downlink direction. Otherwise CSMA/CA would affect the results on both, uplink and downlink.

Computer 1 is connected via wire to the access point. The second and third workstations are connected to AP via wireless links. LTest-client on computer 1 sends packets to LTest-servers on computer 2 or 3. In scenario 1 computer 2 and access point are about 1 meter away from each other. For scenario 2, the computer 3 is moved 8 meters away from the access point and separated by a concrete wall in another room to get worse wireless conditions. The link parameters of both scenarios are shown in Table 1.

TABLE I
LINK INDICATORS

Scenario	Link Quality	Signal Level
1	65/70	-45 dBm
2	51/70	-59 dBm

The topology of scenario 2 is used to prove the assumption that the CSMA/CA mechanism can cause TCP retransmissions, though no packet loss is occurred. Fig. 3 shows the logic connections in Scenario 2. Two LTest servers on computer 3 are started on the same interface on different ports. After Wireshark is started on computer 1 to monitor interface, one LTest client uses TCP to connect to LTest Server 1 and the other LTest client uses UDP to connect to LTest Server 2. Both clients send data with a constant bit rate of 5 Mbps with MTU 1024 bytes. The duration of both measurement sessions is 5 minutes.

In scenario 3, additionally cross traffic is emulated. For this purpose the connection with third machine is used. In this scenario cross traffic is generated between machines 1 and 3 and the probe traffic is generated between machines 1 and 2. Worth to mention that cross traffic is acting during performing of test on scenario 3. Fig. 4 shows the described connections

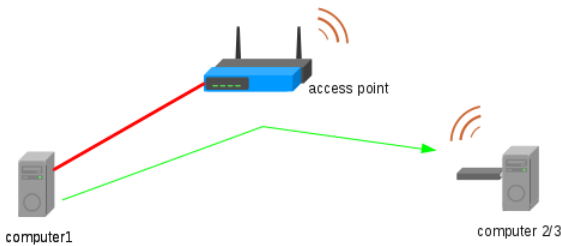


Fig. 2. Technical view of topology.

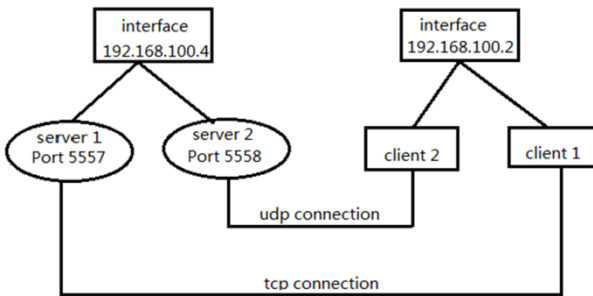


Fig. 3. Logic connections.

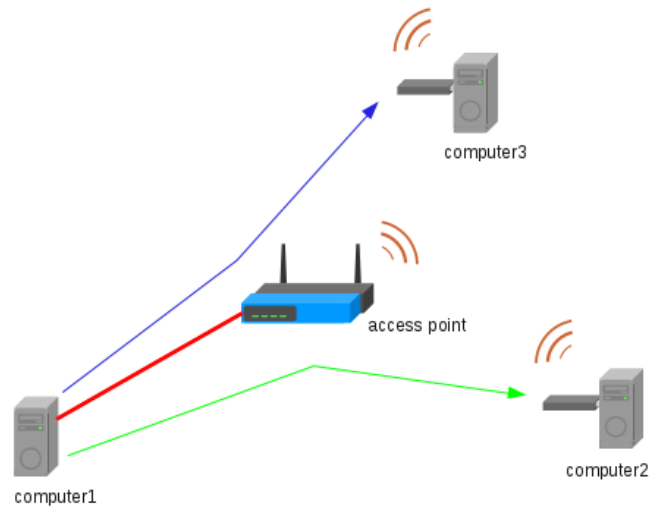


Fig. 4. cross traffic topology.

IV. EXPERIMENTAL RESULTS

During the experiment according to Fig. 3 6 retransmissions have occurred in the TCP session. At the same time there were only 2 packet losses in the UDP-connection. Sending time of TCP retransmissions and UDP packet losses are shown in **Fehler! Verweisquelle konnte nicht gefunden werden.** UDP has no timeout mechanism, so the packets were in fact lost. It means that at least two retransmissions over TCP-connection could be caused due to poor link quality. At least the other 4 retransmissions are then caused by TCP timeout. TCP timeout is caused by the long delay of CSMA/CA mechanism.

TABLE II
TCP RETRANSMISSION AND UDP PACKET LOSSES

Sending time of TCP retransmissions[s]	Sending time of UDP packet losses[s]
10.812	
26.885	26.822
28.999	
70.622	
92.924	
121.087	121.032

The TCP retransmissions that happened almost at the same time with UDP losses are not that valuable, since they happened due to real losses in the media. The behavior of TCP and UDP OWDs of around of the TCP retransmission at 92.924 s are plotted in Fig. 5. From the plot it is clear that TCP retransmission, depicted as a vertical line in the figure, occurred near the peak of UDP OWDs, where TCP OWD is almost twice bigger than UDP OWD. It is obvious that this TCP retransmission is caused by CSMA/CA mechanism.

The experiment for each scenario consists of 20 tests that have been made. Each test for all experiments is a transmission of data traffic with a constant bit rate of 5 Mbps using UDP with MTU 1024 bytes via AP from client machine to server machine. The duration of each test is 5 minutes. As the result of experiments the behavior of one-way delay (OWD) for each packet is collected for all the tests. These data will be used to show the delay profiles of CSMA/CA. The plot in Fig. 6 shows the behavior of OWD for one test of Scenario 1.

In Fig. 6 is worth to mention that one packet is delayed by about 10.9 ms after 6.44 s from the beginning of measurement, caused by the timeout in data link layer, then the other 7 packets will be buffered in buffer, after 10.9 ms these packets are sent successively, it can be easily proved on (1).

$$\text{Interpacket time} = \frac{1000 \text{ ms}}{5000000 \text{ bit}} * \frac{1024*8 \text{ bit}}{1 \text{ packet}} = 1.638 \frac{\text{ms}}{\text{packet}} \quad (1)$$

The amount of buffered packets (n) can be got through (2):

$$n = \frac{10.9 \text{ ms}}{1.638 \text{ ms/packet}} = 7 \quad (2)$$

In a wireless environment in presence of impairments, these peaks, caused by collisions and bit errors, can appear very often and even cause successive packet losses. Each retransmission in data link layer leads to increasing of contention window, naturally the time delay between transport layers will be also increased. If the retransmission timeout in TCP is not properly set, it could also cause packet losses. It highlights the importance of this work to study profile of traffic that passes CSMA/CA connection, then tune TCP on CSMA/CA.

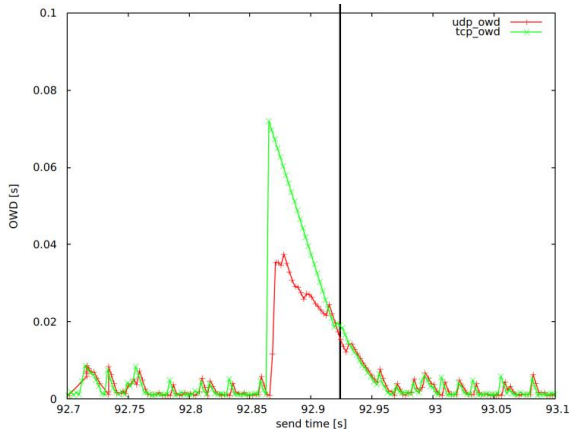


Fig. 5. OWDs over TCP-and UDP-connection.

There are no packet losses for first scenario, during the test. The result shows that the mean value of OWDs is about 900 μ s. In comparison with scenario 1, in presence of packet loss ratio between 0 to 0.005 %, the mean value of OWDs increases up to approximately 1800 μ s in scenario 2. In scenario 3 packet losses are between 0.0001 and 0.21 % and the mean value of OWDs between 2000 μ s and 8500 μ s are retrieved.

In Fig. 7 the cumulative distribution of OWDs is displayed in 3 scenarios. The OWDs in scenarios 1, 2 and 3 are distributed in the interval from 0 s to 0.12 s, from 0 s to 0.18 s and from 0 s to 0.67 s accordingly. The curves of scenarios 1 and 2 are much steeper than the of scenario 3. It means that the packet loss probability of scenario 1 and 2 decreases very rapidly with the increasing OWD. With a probability of 10^{-5} , when a CSMA/CA off period occurs, it will cause in scenario 1 and 2 a delay of at least 0.101 s and 0.152 s respectively, however in scenario 3 it would cause an off-time of 0.65 s with the same probability. To suppress the packet loss rate, occurred on CSMA/CA peaks below 10^{-6} , in scenario 3 the actual retransmission timeout must be

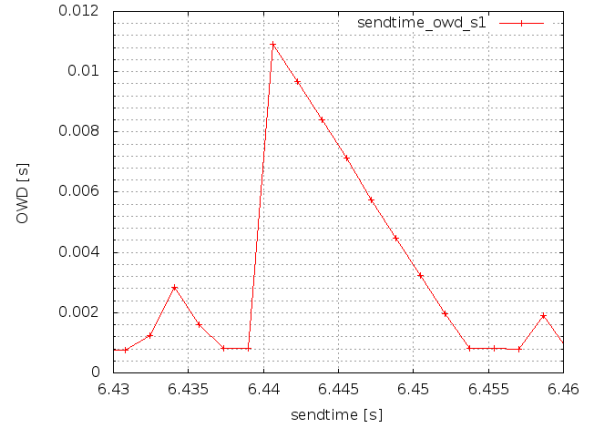


Fig. 6. OWD

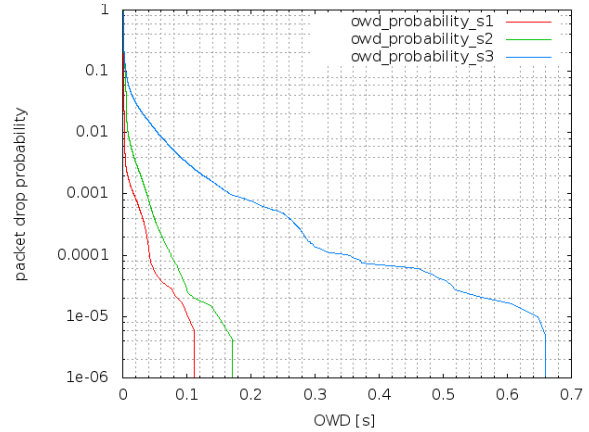


Fig. 7. Cumulative distribution of OWDs in 3 scenarios.

set to at least 0.67 s, whereby in scenario 1 and 2 a timeout of 0.12 s and 0.18 s respectively would be sufficient.

Through analysis of the distance between the OWDs peaks in 3 scenarios, the frequency of the appearance of the peak of OWDs in each scenario could be figured out. In other words, it will be known how often the peaks of OWDs occur. In Fig. 8 the cumulative distribution of distances between the peaks of OWDs is displayed for 3 scenarios. All peak values in scenario 2 occur under the distance of 0.03 s. Respectively 99.9855% and 99.963% of the total peaks in scenario 3 and 1 occur under the distance of 1.1 s. Peaks less than 0.15 s in scenario 3 and 1 take almost equal 99.9928% of the total. It is also recognized that the peaks occur most frequently in scenario 2. Then the peaks in scenario 3 appear more often than in scenario 1.

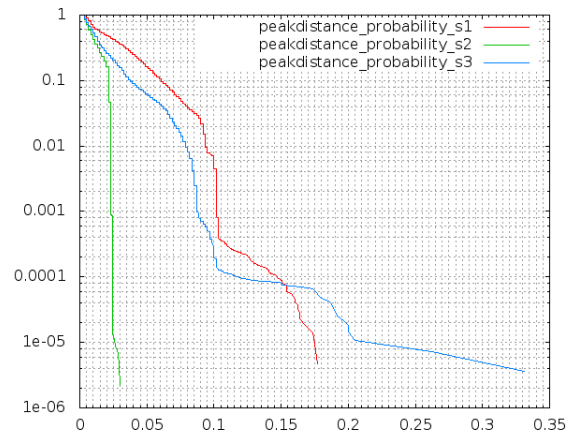


Fig. 8. Cumulative distribution of distance between peaks in 3 scenarios.

V. CONCLUSION AND FUTURE WORK

In this work we have proved that the CSMA/CA mechanism can cause TCP retransmissions. Besides, this work emulates the 3 most popular occasions in CSMA/CA connections: with good link quality, with bad link quality and in presence of cross traffic. Using LTest each packet has been traced, the influence of CSMA/CA on packet was extendedly observed. In summary, it can be determined that the CSMA/CA actually has a negative impact on TCP performance. The cumulative distribution of OWDs shows the relation between packet loss probability and OWDs.

The experiment shows that scenario 1, 2 and 3 need respectively 0.12s, 0.18s and 0.67 s to suppress the packet loss rate under 10^{-6} . In addition, the cumulative distribution of the distances between the peaks describes how often the peaks of OWDs appear. In the experiment it has been also retrieved that the peaks occur most frequently in scenario 2 and in scenario 3 the peaks appear more often than in scenario 1. It affects the performance of TCP, however if only the retransmission timeout is properly set, TCP performance will be increased. These two important informations could be used to modify the TCP-RTO. The way to improve the performance of TCP is proposed, however this work is still undergoing. In the future more measurements will be carried out in more complex test networks. We will test the relation between OWDs and packet losses probability with different link quality and contention levels, trying to get the dynamical relation CSMA/CA profiles. It would be very helpful to adjust the transmission timeout of TCP to CSMA/CA connections. The ultimate goal is to get a new algorithm for timeout in wireless network. We consist on, this will be a huge step towards tuning TCP performance over wireless network.

REFERENCES

- [1] TIA/EIA/cdma2000, "Mobile Station - Base Station Compatibility standard for Dual-Mode wideband Spread Spectrum Cellular Systems", Washington: TIA, 1999.
- [2] E. Dahlman, S. Parkvall, J. Skoeld, P. Beming, "HSPA and LTE for Mobile Broadband", Elsevier, 2007.
- [3] 3GPP TS36.300, "Evolved Universal Terrestrial Radio Access (EUTRA) and Evolved Universal Terrestrial Radio Access Network (EUTRAN); Overall description".
- [4] WiMAX Forum. "Mobile WiMAX-Part I: A Technical Overview and Performance Evaluation", 08 2006. [Online]. Available: http://www.wimaxforum.org/news/downloads/Mobile_WiMAX_Part_1_Overview_and_Performance.pdf
- [5] G. Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Coordination Function". Selected Areas in Communications, IEEE Journal on 18th Volume, 03.2000.
- [6] M. A. Youssef and R. E. Miller. "Analyzing the Point Coordination Function of the IEEE 802.11 WLAN Protocol using a System of Communicating Machines Specification". 05 2002.
- [7] IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, ISO/IEC 8802-11:1999E. 08 19999.
- [8] A. S. Tanenbaum, D. J. Wetherall. "Computer Networks Fifth Edition", Pearson Education, 2011
- [9] NJ Kothari, BM Gambhava, KS Dasgupta. "RTT utilization by detecting avoidable timeouts". 14th IEEE International Conference on Networks, Singapore. 09 2006.
- [10] D Ciullo, M Mellia and M Meo. "Two schemes to reduce latency in short lives TCP flows". IEEE Communications Letters 13. 10 2009.
- [11] P Mi-Young and C Sang-Hwa. "Distinguishing the cause of TCP retransmission timeouts in multi-hop wireless networks". 12th IEEE International Conference on High Performance Computing and Communications. 09 2010.

- [12] E. Siemens, S. Piger, C. Grimm, and M. Fromme, "LTest - a tool for distributed network performance measurement," Consumer Communications and Networking Conference, 2004. CCNC 2004, first IEEE, 01 2004.
- [13] A. Bakre and B. R. Badrinath, "Handoff and System Support for Indirect TCP/IP". in proceedings of Second Usenix Symposium on Mobile and Location-Independent Computing, 04 1995.
- [14] H. Balakrishnan et al. "Improving TCP/IP Performance over Wireless Networks". in proceedings of ACM Mobicom, 11 1995.
- [15] R. Ludwig et al. "Multi-layer Tracing of TCP over a Reliable Wireless Link". in proceedings of ACM SIGMETRICS, 1999.
- [16] S. Paul et al. "An Asymmetric Link-Layer Protocol for Digital Cellular Communications", in proceedings of INFOCOM, 1995.
- [17] M. Mathis, J. Mahdavi, S. Floyd and A. Romanow. "TCP Selective Acknowledgments Options". 07 1996.
- [18] E. Aynoglu, S. Paul, T. F. LaPorta, K. K. Sabnani and R. D. Gitlin. "AIRMAIL: A Link-Layer Protocol for Wireless Networks". 02 1995.
- [19] Third Generation Partnership Project. "RLC Protocol Specification (3G TS 25.322)". 1999.
- [20] TIA/EIA/IS-707-A-2.10. "Data Service Options for Spread Spectrum Systems: Radio Link Protocol Type 3". 01 2000.
- [21] R. Ludwig and R. H. Katz. "The Eifel Algorithm: Making TCP Robust Against Spurious Retransmissions". In ACM Computer Communications Review, Vol. 30, No. 1. 01 2000.
- [22] H. Inamura et al.. "TCP over 2.5G and 3G Wireless Networks". draft-ietf-pilc-2.5g3g-07. 08 2002.
- [23] M. C. Chan and R. Ramjee. "TCP/IP Performance over 3G wireless links with rate and variation". In Proceedings of ACM Mobicom'02. 2002.
- [24] Mun Choon Chan and Ramachandran Ramjee. "Improving TCP/IP Performance over Third Generation Wireless Networks". Mobile Computing, IEEE Transaction on Volume 7, Issue 4. 04 2008.
- [25] M. Natkaniec and A. R. Pach, "An Analysis of the Backoff Mechanism used in IEEE 802.11 Networks". Computers and Communications. 07 2000.
- [26] Q. Pang, S. C. Liew, J. Y. B. Lee, and S.-H. G. Chan, "A TCP-like Adaptive Contention Window Scheme for WLAN". Communications, IEEE international Conference on Volume 6. 06 2004.
- [27] T. Li. "Improving Performance for CSMA/CA Based Wireless Networks". A dissertation submitted for the degree of Doctor of Philosophy. 12 2007.
- [28] P. Sreekumari and M. Lee, "TCP NRT: a new TCP algorithm for differentiating non-congestion retransmission timeouts over multihop wireless networks". EURASIP Journal on Wireless Communications and Networking. 06 2013.

Data Processing at the Flaw Detector with Combined Multisector Eddy-Current Transducer

Evgeny Yakimov, Alexander Goldshtein, Valery Bulgakov, Sergey Shipilov
National Research Tomsk Polytechnic University- Institute of Nondestructive Testing
Lenina Ave. 30, Tomsk, 634050, Russia

E-mail: shishkovka@mail.ru, algo154@yandex.ru, bvf49@sibmail.com, shipilov@webmail.tsu.ru

Abstract—There is described data processing at the flaw detector with combined multisectional eddy-current transducer and heterofrequency magnetic field. The application of this method for detecting flaws in rods and pipes under the conditions of significant transverse displacements is described.

Keywords: combined multisector eddy-current transducer, eddy-current testing, nondestructive testing of bars and tubes.

I. INTRODUCTION

At flaw detecting of extended metal articles (rods and pipes) in a process of production there are problems caused by heavy testing conditions. Firstly, it is necessary to provide independence from the large transverse displacements of controlled product. Secondly, it is necessary to detect defects with different physical natures and geometrical parameters: faulty fusions, cracks, hair cracks, rolling laps, nonmetallic inclusions. Thirdly, electromagnetic properties of controlled article material could vary from sample to sample. It makes difficult calibration of the flaw detector. Cause of the difference in electromagnetic properties could be variation of temperature and different chemical composition of material.

Above mentioned problems could be solved using combination attachable multisector ECT (eddy-current transducer) and transmission multisector ECT with excitation of circularly and longitudinally directed ECs (eddy-currents) at different frequencies [1][2]. Advantages of such technical solution are high sensitivity to both local defects and extended defects in the presence of large transverse displacements between surface of tested object and windings of ECT, small dependence of signal from the position of defect in the control zone and transverse displacement of the object.

Effective detuning from the influence of transverse displacements can be provided even by the large change of electromagnetic properties [3].

II. FLAW DETECTOR WITH TRANSMISSION MULTISECTOR AND ATTACHABLE MULTISECTOR EDDY-CURRENT TRANSDUCER

Fig. 1a shows the structural scheme of the FD (flaw detector) in which the proposed testing method is implemented and Fig. 1b shows the design of the ECT of the

FD. For clearness, the ECT windings are conditionally separated along the longitudinal Z axis, but in reality, all the winding have the same longitudinal location.

The FD principle of operation is as follows. Generators $1-3$ generate harmonic voltages at frequencies ω_1 , ω_2 , and ω_3 . All generators are synchronized by synchronization circuit 4; due to this, the frequency differences $\omega_2-\omega_1=\Delta\omega$ and $\omega_3-\omega_1=k\Delta\omega$ (here, k is an integer) are maintained at stable levels. The frequencies ω_1 , ω_2 , and ω_3 differ from $\Delta\omega$ by an integer number of times. In this case, different variants of synchronizing the generator frequencies are possible, for example, a variant where the synchronization circuit contains a generator of rectangular pulses of the reference frequency ω_0 , and generators $1-3$ contain frequency dividers by factors of $m(n-1)$, mn , and $n(n-1)$, respectively, and the selective circuits for selecting the first harmonics of rectangular signals (m and n are integers). In this case, the frequencies of the output voltages of the generators are, respectively, $\omega_1=\omega_0/mn$, $\omega_2=\omega_0/n(n-1)$, and $\omega_3=\omega_0/n(n-1)$.

The frequency difference is $\Delta\omega=\omega_0/mn(n-1)$ and $k=m-n+1$. The signal of the difference frequency $\Delta\omega$ can be obtained by the consecutive division of the frequency ω_0 by m , n , and $(n-1)$. This signal is used for processing ECT signals.

The output signals of generators $1-3$ are fed to the excitation windings 5-7 of the ECT. The currents in these windings create a magnetic field with three harmonic orthogonal spatial frequency components ω_1 , ω_2 , and ω_3 in the ECT testing zone. The three frequencies exciting magnetic field induces eddy currents at three frequencies in a tested article. The two-sectional attachable-type windings 5 and 6 are used to excite longitudinal eddy currents at frequencies ω_1 and ω_2 , which have close values, in the surface layer of a tested article. Transmission-type winding 7 serves for exciting eddy currents of circular direction at the frequency ω_3 . Note that, in this case, in contrast to the use of classical multifrequency ECTs, the frequency difference $\Delta\omega$ is negligibly small from the standpoint of the results of electrodynamic interaction of the magnetic field with a tested object.

In order to reduce signals from structural inhomogeneities that usually correlate with local changes in the magnetic properties, magnetic biasing of a tested object is performed in the FD by the magnetic field of a direct current in magnetization winding 17, which is connected to stabilized magnetizing-current source 16.

To measure the components of the field of eddy currents at the frequencies ω_1 and ω_2 , four-sectional and two-sectional measuring windings 8 and 9 and two-sectional and four-sectional measuring windings 10 and 11 are used, respectively. The components of the field of eddy currents at the frequency ω_3 are measured with one-sectional measuring windings 12–15. Owing to the appropriate directions of coiling the sections of the exciting and measuring windings and their mutual arrangement (Fig. 1b), when an article is absent in the ECT testing zone and the axis of an article, which is placed in the testing zone, coincides with the longitudinal axis of the ECT, the initial and introduced EMF (electromotive force) of the measuring windings at the frequency of the measured magnetic field are absent. This property of the ECT that is used in the FD makes the latter poorly sensitive to such interfering factors as changes in the diameter and the electromagnetic properties within the

acceptable range for a suitable article. An EMF at the frequency of the measured magnetic field in the measuring windings arises upon disturbance of the symmetry of eddy currents that are induced in an article in the presence of a flaw, a radial displacement (for windings 8–11), or a skew (for windings 8–15) of a tested article relative to the longitudinal axis of ECT.

Amplitude–phase processing of signals is used to separate these effects. For this purpose, the FD has ten identical measuring channels 18–25, each of which consists of an amplitude–phase detector and an integrating digitizer that connected in series. Channels 18–21 are intended for selecting signals from extended flaws, channels 18 and 20 are intended, in addition, for selecting signals from transverse displacements of a tested article along the Y and X axes, respectively, and channels 22–25 are intended for selecting signals from short flaws.

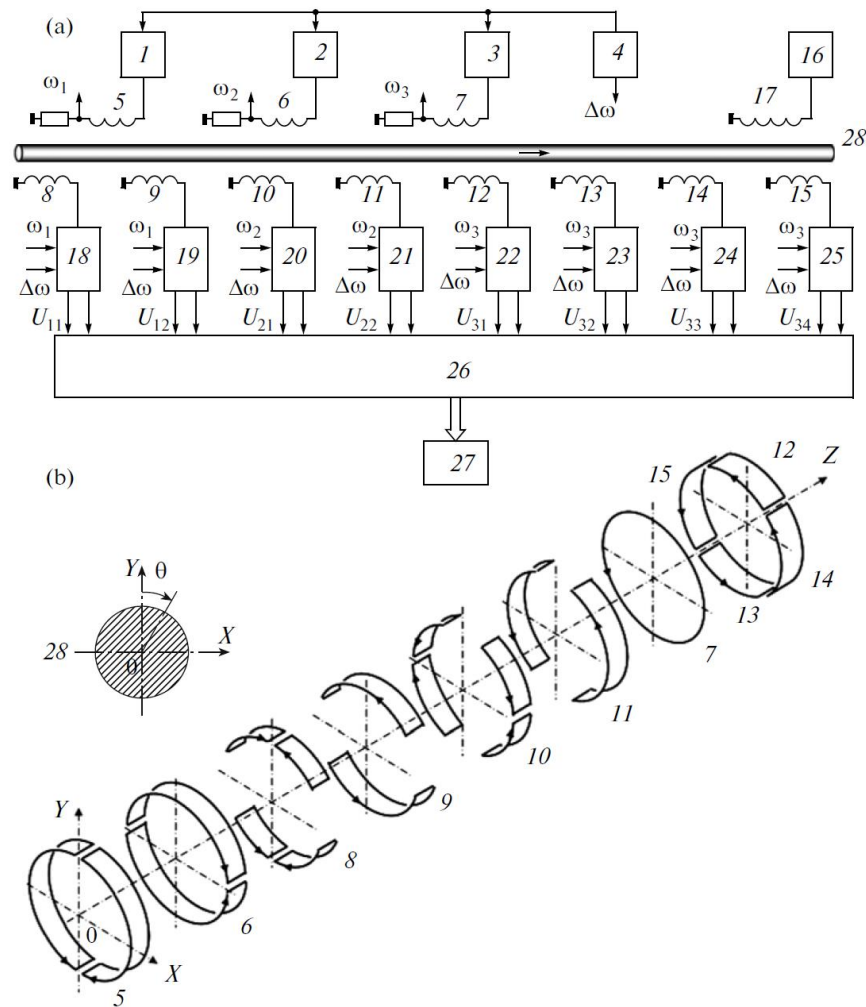


Fig. 1. (a) Structural diagram and (b) the design of the ECT of a flaw detector: (1–3) generators, (4) synchronization scheme, (5–7) excitation windings, (8–15) measuring windings, (16) magnetizing current unit, (17) magnetizing winding, (18–25) measuring channels, (26) calculation unit, (27) actuating devices, and (28) tested object

The amplitude–phase detector performs synchronous detection of the voltage across the measuring ECT winding with the corresponding control frequency ω_i and the integrating digitizer averages the output signal of the amplitude–phase detector over the time $T = 2\pi / \Delta\omega$, which is set by the output signal of synchronization circuit 4. The integrating digitizer circuit includes the circuits of an analog integrator and a storage unit, which are connected in series and over which a negative feedback loop is placed, and is a pulsed low-pass filter with a finite pulsed characteristic [4]. The transfer coefficient of such measuring channel has the following dependence on the input signal frequency ω :

$$K(\omega) = \frac{T}{2} \int_{-\frac{T}{2}}^{\frac{T}{2}} \sin \omega t \cdot \sin \omega_i t \cdot dt = \frac{2}{\pi} \cdot \frac{\omega_i \cdot \Delta\omega}{\omega^2 - \omega_i^2} \cdot \sin \pi \frac{\omega}{\Delta\omega} \quad (1)$$

The analysis of the amplitude–frequency characteristic (AFC) of the measuring channel (Fig. 2) with a reference frequency ω_i (curve 1) shows that the dependence $K(\omega)$ has zeros at frequencies that differ from ω_i by values multiple of $\Delta\omega$. A modulated signal at a frequency ω_i from a flaw with the spectrum shown in Fig. 2 is transmitted by the measuring channel almost without distortions and a modulated signal at a nearby frequency $\omega_i + \Delta\omega$ (its spectrum is shown by curve 3) is attenuated by more than one order of magnitude. It is also important that slowly changing signals at the frequency $\omega_i + \Delta\omega$ (drift of the unbalance voltage, displacement-caused signals) are completely suppressed by the measuring channel even when the values of ω_i and $\omega_i + \Delta\omega$ are very close to each other.

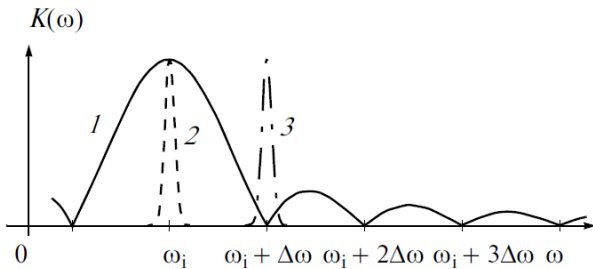


Fig. 2. The amplitude–frequency characteristic of the measuring channel with a reference frequency ω_i .

As a result of such signal processing, signals U_{11} and U_{12} are extracted at the outputs of measuring channels 18 and 19. These signals are proportional to the amplitudes of the complex components of the introduced voltages of four-sectional and two-sectional measuring windings 8 and 9, respectively, at the frequency ω_1 . Signals U_{21} and U_{22} are extracted at the outputs of measuring channels 20 and 21. These signals are proportional to the amplitudes of the complex components of the introduced voltages of four-sectional and two-sectional measuring windings 10 and 11, respectively, at the frequency ω_2 . Signals U_{31} , U_{32} , U_{33} , and U_{34} , which are proportional to the amplitudes of the complex components of the introduced voltages of one-sectional measuring windings 12–15 at the frequency ω_3 , are extracted at the outputs of measuring channels 22–25.

The high quality separation of signals that are caused by each separate magnetic field component allows efficient use

of the amplitude–phase tuning out from the influence of radial displacements and skews. This kind of tuning out in each channel is performed by regulating the phase shifts of the reference voltages of the amplitude–phase detectors. The voltages across resistors, which are connected in series with the excitation windings, which coincide in phase with the ECT excitation currents, are used as the reference voltages in the FD circuit. For the measuring channels with the frequencies ω_1 and ω_2 , the direction of the tuning out from displacements, which is orthogonal to the displacement line, virtually coincides with the line of flaws for the used ECTs and testing conditions; for the measuring channels with the frequency ω_3 , the direction of the tuning out makes an angle of 30–45° with the lines of flaws. As a result of the amplitude–phase tuning out, the imaginary components of the output voltages of the FD’s measuring channels are determined mainly by the presence or absence of a defective zone of the article in the tested zone and the real components are determined by the transverse displacement of the article relative to the longitudinal axis of the ECT.

Calculation unit 26 determines the resulting voltages U_1 of the measuring channels of the longitudinal components of eddy currents at the frequencies ω_1 and ω_2 and U_2 of the measuring channels of the circular components of eddy currents at the frequency ω_3 . In addition, unit 26 compares these voltages to specified threshold values.

The signal amplitudes from a flaw in each measuring channel depend not only on the flaw geometry (depth, opening, and orientation), but also on the azimuth (angle θ of the location on the article surface (Fig. 1b).

Fig. 3 shows the dependences of the flaw-introduced voltages of the channels for extracting signals from extended flaws on the flaw azimuth. For the amplitude of the total signal to be independent of the azimuth of an extended flaw, the pairwise algebraic summation is performed and signals from flaws of measuring channels 18–21 are then vector summed:

$$U_1 = \sqrt{(\text{Im} \dot{U}_{11} + \text{Im} \dot{U}_{22})^2 + (\text{Im} \dot{U}_{12} + \text{Im} \dot{U}_{21})^2} \quad (2)$$

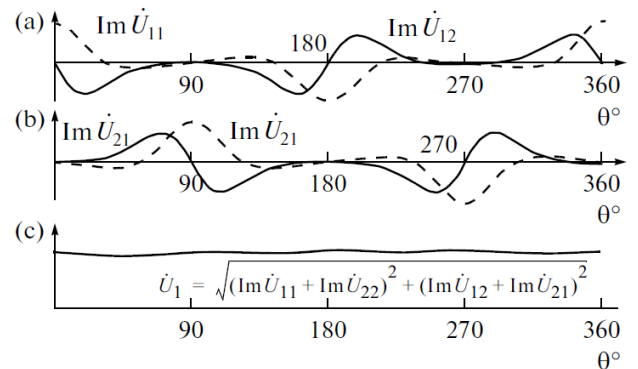


Fig. 3. The dependences (a, b) of the flaw-introduced voltages of individual FD channels and (c) the resulting signal on the flaw azimuth.

The resulting signal U_1 is virtually independent of the flaw azimuth. Another positive feature of this algorithm for processing ECT signals is that, in this case, a noise signal that is caused by the possible violation of the optimal condition of tuning out from a displacement because of changes in the electrophysical properties of a tested article

and a disturbance of the perpendicularity of the tuning out direction and the displacement line, is substantially attenuated. Physically, this weakening of the influence of the displacement is explained by the fact that when the article shifts along the OY and OX axes signals arise in measuring windings 8–11 and 9–10, respectively; these signals are in antiphase in both cases. As a result, after the detected voltages are summed, signals from the displacement in different channels are mutually compensated.

For the amplitude of the total signal U_2 to be independent of the azimuth of a short flaw and in order to weaken the dependence of U_2 on transverse displacements of a tested article, the output signals from flaws of measuring channels 22–25 are algebraically summed in calculation unit 26 by analogy with [5] with coefficients s_1 , s_2 , s_3 , and s_4 , respectively, which are calculated as functions of the values of transverse displacement of the article.

The necessity of using correcting coefficients during summation of signals from short flaws is determined by the much stronger dependence on the transverse displacements of a tested article than in the case of extended flaws.

Fig. 4 shows the behavior of the dependences of the output signals from a flaw of measuring channels 22–25 $\text{Im}U_{31}$, $\text{Im}U_{32}$, $\text{Im}U_{33}$, and $\text{Im}U_{34}$ on the flaw azimuth θ for a transverse displacement of a tested article along the OY axis ($y=-2\text{mm}$). All the dependences are normalized to the maximum signal from the flaw in the measurement channels without a displacement.

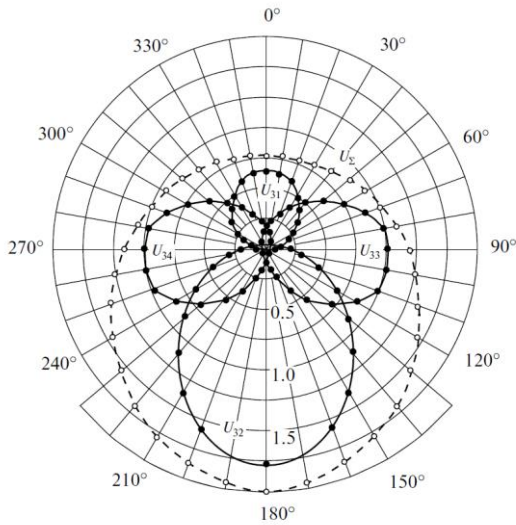


Fig. 4. Dependences of signals of the measuring channels on the flaw azimuth during a transverse displacement of a tested article by 2 mm to winding 14.

The analysis of the curves in Fig. 4 shows that the flaw-introduced signals in measuring channels 23 and 25, which are connected to windings 13 and 15, are virtually independent of a transverse displacement of the mentioned direction. During a transverse displacement in the given direction, the flaw-introduced signal in measuring channel 22, which is connected to winding 12, decreases by a factor of 1.54; the flaw-introduced signal in measuring channel 24, which is connected to winding 14, increases by a factor of 1.76.

A dotted line shows the total signal U_Σ of all four measuring channels without using correcting coefficients.

In this case, the degree of nonuniformity of the sensitivity to a flaw, which is caused by transverse displacements, reaches 160%.

The initial data for determining the correcting coefficients s_1 , s_2 , s_3 , and s_4 are the dependences of signals from flaws of measuring channels 22–25 $\text{Im}U_{31}$, $\text{Im}U_{32}$, $\text{Im}U_{33}$, and $\text{Im}U_{34}$ and signals from displacements of measuring channels 18 and 20 $\text{Re}U_{11}$ and $\text{Re}U_{21}$ on transverse displacements x and y along the OX and OY axes, respectively.

Fig. 5 shows the behavior of the dependences of the signals $\text{Im}U_{31}(y)$ and $\text{Im}U_{32}(y)$ from flaws on a transverse displacement y along the OY axis and the signal $\text{Re}U_{11}(y)$ on the displacement. These dependences can be represented by the following functions:

$$\begin{aligned} \text{Im}U_{31}(y) &= F_1(y)U_{N1}; \quad \text{Im}U_{32}(y) = F_2(y)U_{N2}; \\ \text{Re}U_{11}(y) &= F_3(y) \end{aligned} \quad (3)$$

Here, U_{N1} and U_{N2} are signals from a flaw in the absence of a displacement ($y=0$), $F_1(y)$ and $F_2(y)$ are functions that describe the dependences of signals from the flaw on the displacement, and $F_3(y)$ is a function that describes the dependences of a displacement-caused signal on the displacement.

The function $F_3(y)$ can be considered linear with a sufficient degree of accuracy:

$$\text{Re}U_{11}(y) = ty \quad (4)$$

where

t - a constant factor.

From here, the displacement value can be determined:

$$y = \frac{\text{Re}U_{11}}{t} \quad (5)$$

Using the calculated value of y , the displacement independent values of signals from a flaw can be determined:

$$U_{N1} = \frac{\text{Im}U_{31}}{F_1(y)} \quad \text{and} \quad U_{N2} = \frac{\text{Im}U_{32}}{F_2(y)} \quad (6)$$

Thus, the correcting coefficients are

$$s_1 = \frac{1}{F_1(y)} \quad \text{and} \quad s_2 = \frac{1}{F_2(y)} \quad (7)$$

In view of the same character of the dependences of signals on displacements x and y , the displacements x and the correcting coefficients are found in a similar manner:

$$s_3 = \frac{1}{F_1(x)} \quad \text{and} \quad s_4 = \frac{1}{F_2(x)} \quad (8)$$

The functional dependences $F_1(y)$ and $F_2(y)$ and the value of the factor t are usually determined experimentally. The functions $F_1(y)$ and $F_2(y)$ are specified in calculation unit 26

in a tabulated form or are approximated by analytical expressions. The resulting signal U_2 of channels for measuring the circular components of eddy currents at the frequency ω_3 is equal to

$$U_2 = s_1 \operatorname{Im}U_{31} + s_2 \operatorname{Im}U_{32} + s_3 \operatorname{Im}U_{33} + s_4 \operatorname{Im}U_{34} \quad (9)$$

and is virtually independent of the flaw azimuth; it depends weakly on the transverse displacement of a tested article. The degree of nonuniformity of the sensitivity of a flaw, which is caused by transverse displacements, decreases owing to the above described transformation to an acceptable value of ~20%.

When the voltages U_1 and U_2 exceed the threshold values, the calculation unit forms a control signal for actuating devices 27 (rejection-bin control relay, paint marker, etc.).

III. CONCLUSION

The proposed flaw detection method was implemented in a FD prototype. Tests confirmed that such method has high sensitivity to both local defects and extended defects in the presence of large transverse displacements of tested object, small dependence of signal from the position of defect in the control zone and transverse displacement of the object.

REFERENCES

- [1] A. E. Goldshtein, V. F. Bulgakov, and H. M. V. A. Kroning, "A Method of Eddy-Current Flaw Detection of Bars and Tubes Based on the Use of a Combined Eddy-Current Transducer with Excitation of Spatial Magnetic-Field Components at Different Frequencies," *Russian Journal of Nondestructive Testing*, Vol. 47, No. 11, pp. 747-753, 2011.
- [2] A. A. Polevoda and I. Yu. Fedosenko, "On Eddy-Current Flaw Detection with Transmission Transducers for Flow-Line Testing of Pipes and Rolled Stocked, *Zavod. Lab.*, Vol. 64, No. 1, pp. 34-37, 1998.
- [3] V. K. Zhukov and P. A. Ovsyannikov, "ED-3.02 Electromagnetic Flaw Detector for Testing Long Cylindrical Articles," *Defektoskopiya*, No. 4, pp. 30-36, 1983.
- [4] A. E. Goldshtein and S. A. Kalganov, "Eddy-Current Nondestructive Testing of Long Cylindrical Articles Using Spatial Magnetic-Field Components of Different Frequencies," *Defektoskopiya*, No. 5, pp. 65-71, 2000.
- [5] V. F. Bulgakov, "Decrease in the Nonuniformity of the Sensitivity of Eddy-Current Transducers under Radial Displacements of Tested Articles," *Defektoskopiya*, No. 9, pp. 3-8, 1999.

Application of Fuzzy Variables for Systems of Management Decision Support

Leonid Mylnikov

Perm National Research Polytechnic University - Electrotechnical Department

Komsomolsky Ave, 29, 614990, Perm, Russia

E-mail: leonid@pstu.ru

Abstract—The following article describes an approach covering the variety of opinions and uncertainties of estimates within the chosen technique of decision support. Mathematical operations used for assessment of options are traced to operations of working with functions that are used for assessment of possible options of decision-making. Approach proposed could be used within any technique of decision support based on elementary mathematical operations. In this article the above-mentioned approach is described under analytical hierarchy process.

Keywords: fuzzy set, management, decision-making, algorithm, analysis, prediction, membership function, set, analytical hierarchy process.

I. INTRODUCTION

Throughout the whole life man, social community or organizational system have to face numerous choices and to make a variety of decisions. Decision-making based on the given choices is inseparable from their activities. As any product of human activity, decision-making mechanism has a dual nature: subjective and objective [3]. It was noticed that the subjective component by no means always leads to the best result. Thus, in order to enlarge the objective component of choice of this or that scenario a variety of decision support methods were developed.

All decision support methods could be divided into 2 big groups: 1) methods handling accurate practical data (OLAP, Data Mining etc) and presenting it in a convenient form for decision-making person; 2) methods handling expert analysis data or Q-data (SWOT-analysis, analytical hierarchy process, economic techniques etc).

Methods of the second type are commonly used in economics and organizational system management. In these particular systems the objective expert estimate is the most challenging part. So the problem of accuracy improvement of such methods' estimates is crucial.

The only way to improve methods' accuracy is to improve estimates. As experts are not able to give different estimates, it makes sense whether to use the theory of probability. to assess the level of certainty of experts in their estimates and thus neutralize experts' errors or to use additional experts for estimates' precision. However methods in an explicit form are unable to cover several expert opinions. For those purposes the approaches based on average values given by an expert group or interval value

estimates are used [4][11]. They reduce accuracy and bring additional simplifications to methods which are already not always precise enough.

II. TECHNIQUE OF FUZZY VARIABLES APPLICATION

Description of complex estimates (estimates including the level of certainty or estimates of experts' group) is possible through functions (even in the format of a table or matrix). These functions could be considered as membership functions $\mu_A(x)$ from fuzzy-set theory (fuzzy logic). Application of fuzzy-set theory in comparison with other theories has lots of advantages as here all elementary mathematical operations could be defined. Moreover, other theories are based on the probabilities and could be narrowed to fuzzy-set theory [1][5][10].

Another advantage of fuzzy-set theory is that lots of data collected through opinions on this or that problem could not be presented as one figure but as some verbal estimates (e.g. bad, good, adult, child, prospective etc). These estimates in practice are considered as linguistic variables that are as well described by membership functions [7]. Though, if considering several opinions, even such estimates could vary making the general estimate quite blurry.

Accordingly, the general scheme of decision-making methods within fuzzy-set theory could be presented as following: (Fig. 1 [9]).

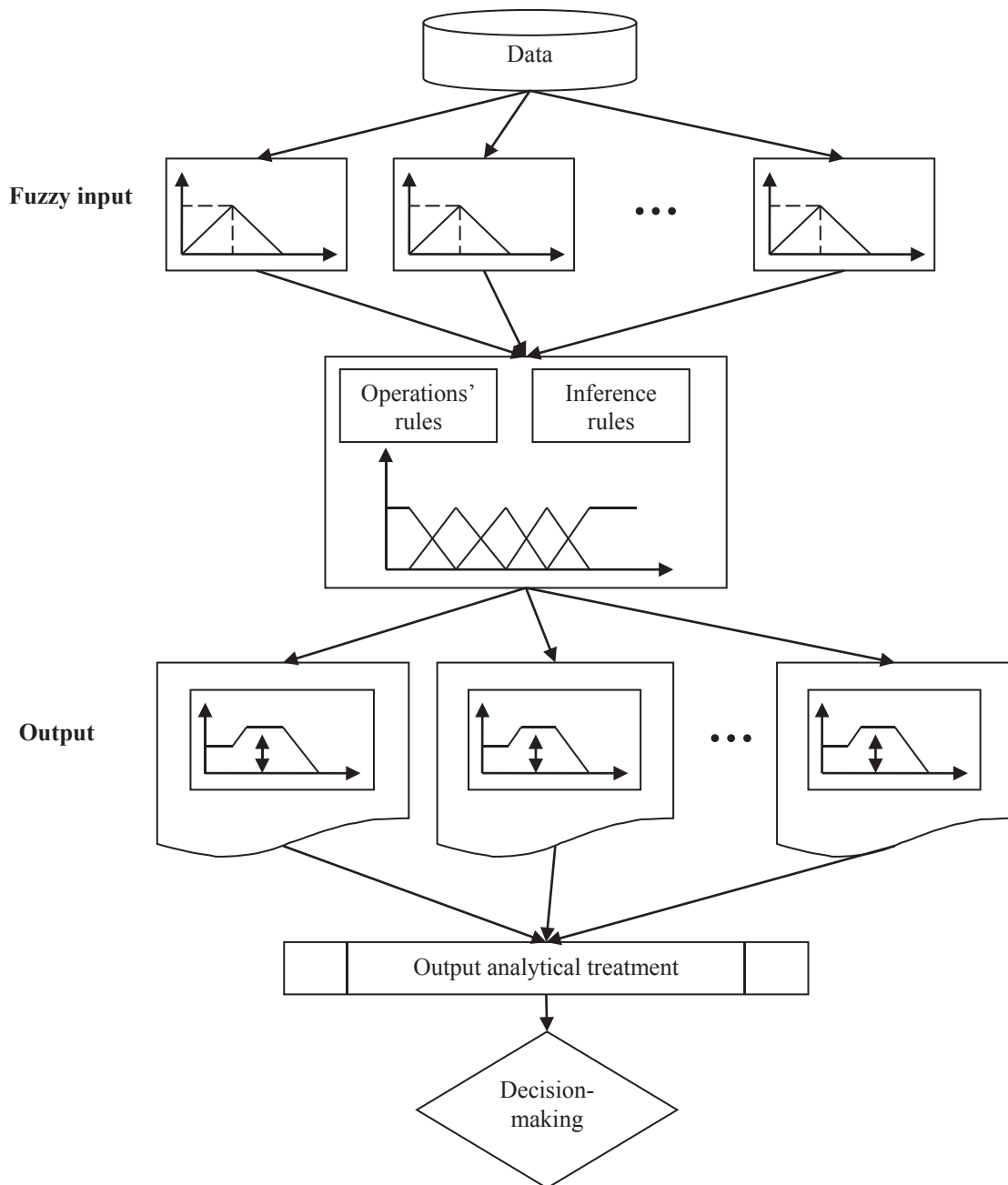


Fig. 1. Method application scheme.

The first stage is the transfer of numerical estimates into fuzzy form (fuzzification). Here the Gaussian function could be used. The calculation is based on the following formula:

$$\mu(x) = e^{-\left(\frac{x-c}{\sigma}\right)^2}$$

where

- $\mu(x)$ - the level of membership to fuzzy set;
- σ - set as level of expert certainty (from 99 % to 1%, that equals values of σ 0.01 and 0.99 respectively);
- x - range of variation of expert estimates, varies from minimum possible to maximum possible total value of the indicator (on Fig. 2 value variation is from 0 to 2);
- c - value of expert estimate or weighting factor.

Another way of transferring numbers into fuzzy form is to define fuzzy set as a triangular fuzzy number [2].

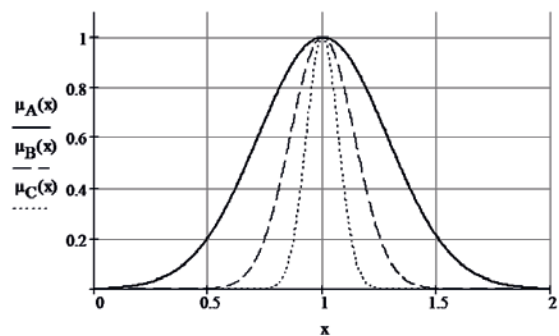


Fig. 2. Graphs of Gaussian function membership with different level of expert certainty.

The majority of techniques and methods needs only 4 basic operations as addition, subtraction, multiplication and division to make a decision. These operations are defined for fuzzy sets and described in literature both for fuzzy sets based on Euler formula and for triangular fuzzy numbers (including operations between variables in fuzzy and explicit form). It is important to notice that performing operations with fuzzy numbers, for instance in triangular shape, gives as a result a number different from a "triangular" one. Therefore some assumptions could be introduced only at the stage of transfer to fuzzy form.

Some arithmetical operations have a number of realization techniques for fuzzy sets. Each technique has its own advantages and should be chosen in accordance with the established task (for example, sum of Zadeh gives the highest accuracy in comparison to other methods). However, despite the difference of realization techniques, the performance of operations does not bring any assumptions or errors into initial data which makes final values true.

Some methods include steps based on various functional connections or conditions. At present such type of theory for fuzzy sets is elaborated and called "Fuzzy systems of logical inference" [2].

III. FUZZINESS OF DECISION SUPPORT METHODS

Let's analyze operations with fuzzy variables with consideration of several opinions by the classic example of selecting a school for a child based on analytical hierarchy process. Let us assume that the parents choose one of 3 available options. First, they need to define criteria for estimation of different options. Criteria should cover, if possible, all areas of the problem. If decomposition received is not sufficient to compare the importance of criteria, further decomposition should be done. Let's assume the received scheme looks as shown on Fig.3:

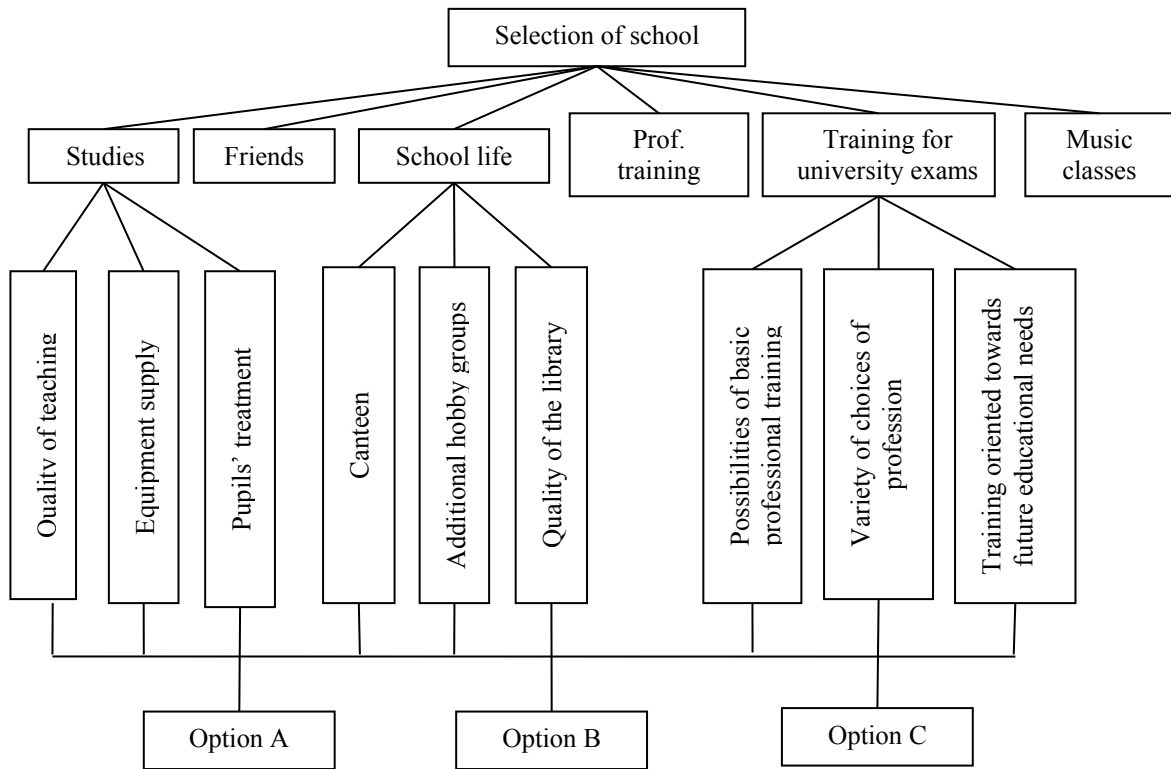


Fig. 3. Hierarchy of Selection of school based on analytical hierarchy process[6].

Decomposition of criteria Friends, Prof. training and Music Classes are skipped for short. After the following scheme is developed it is necessary to do double estimates. If parents have different opinions (the importance of upper level in relation to a target) then double estimates could be done by different parents.

For example, one of the parents compiled the following matrix:

Selection of school	Studies	Friends	School life	Prof. training	Training for university exams	Music classes
Studies	1	5:1	1	6:1	3:1	8:1
Friends	1:5	1	1:5	1:5	1:3	3:1
School life	1	5:1	1	4:1	2:1	6:1
Prof. training	1:6	5:1	1:4	1	4:1	5:1
Training for university exams	1:3	3:1	1:2	1:4	1	4:1
Music classes	1:8	1:3	1:6	1:5	1:4	1

The other parent's matrix looks as following:

Selection of schools	Studies	Friends	School life	Prof. training	Training for university exams	Music classes
Studies	1	4:1	1	6:1	3:1	8:1
Friends	1:4	1	1:5	1:5	1:3	3:1
School life	1	5:1	1	5:1	2:1	4:1
Prof. training	1:6	5:1	1:5	1	6:1	5:1
Training for university exams	1:3	3:1	1:2	1:4	1	4:1
Music classes	1:8	1:3	1:4	1:5	1:4	1

Quality of teaching	School A	School B	School C
School A	1	3:1	5:1
School B	1:3	1	1:2
School C	1:5	2:1	1

Quality of teaching	School A	School B	School C
School A	1	2:1	2:1
School B	1:2	1	1:2
School C	1:2	2:1	1

Based on the criterion *Studies* parents made the following matrixes:

Studies	Quality of teaching	Equipment supply	Pupils' treatment
Quality of teaching	1	2:1	1:3
Equipment supply	1:2	1	1:5
Pupils' treatment	3:1	5:1	1

Studies	Quality of teaching	Equipment supply	Pupils' treatment
Quality of teaching	1	2:1	1:2
Equipment supply	1:2	1	1:3
Pupils' treatment	2:1	3:1	1

Similar matrixes were compiled for all other criteria of second level by both parents.

Next step is dual comparison of options. We assess the advantage of one option over the other in relation to criteria of third level. It is much easier than to do similar assessment in relation to the problem on the whole. For example, for criteria *Quality of teaching* there were generated the following matrixes:

After dual comparison is finished we need to consolidate matrixes so we could apply analytical hierarchy process. In order to do that, numerical estimates (dual comparisons from the matrixes) shall be transferred into fuzzy form and summed up. Consolidation of dual estimates transferred into fuzzy shape could be considered as membership function of fuzzy set. The particularity of the membership curve resulting from summarizing is that depending on the range of expert estimates it would have several peaks in different places. Estimates with big variation range (e.g. issues where expert opinions are very different) should have smaller weight. Thus it is necessary to test how true are the expert estimates and modify membership functions accordingly. The test of truth is defined for two fuzzy sets thus by reference to transitivity rule if task has more than two estimates then it is possible to carry out consequently operations with couples of fuzzy variables with further aggregation of obtained results [9] (Fig. 4).

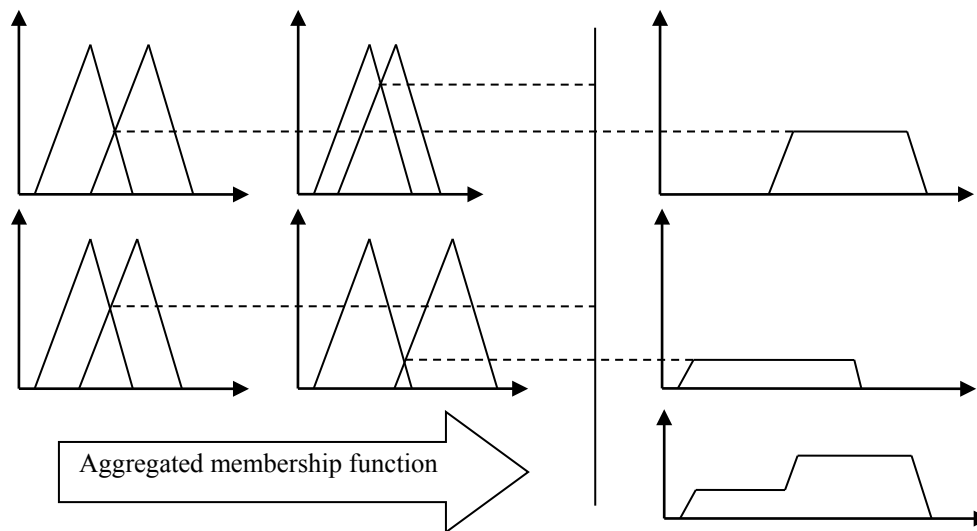


Fig. 4. Test of truth and aggregation of estimates presented as fuzzy variables.

Now it is necessary to calculate eigen vectors (priority vectors) for the obtained matrixes of priority assessment. Calculation could be done by several methods however all of them are restricted by above-mentioned operations and those defined by fuzzy sets. As a result the quantity of vectors will correspond to the quantity of hierarchy levels in the task.

Therefore each criterion will have its vector (priority vector), let us call it V_{ij} where i is a number of hierarchy

level and j is a number of level criterion. Each vector has the dimension equal to the quantity of elements under given criterion. For instance, if there are 3 elements of the next level under criterion, then vector's dimension is 3. In addition to elements connected with given criterion, at the next level there are elements that are not connected with it. We may consider that all similar elements have a priority in relation to the given vector that equals 0. Thus it is possible to expand priority vectors for all criteria by adding 0, for

elements that are not connected with it. For example, the priority vector $V_{11} = (a, b)$ after expansion will look as $V_{11} = (0, a, b, 0)$. As a result all priority vectors of elements of one level will have identical dimension. The following step requires compilation of matrix of priorities for each hierarchy level, let's call it P_i . Matrix is compiled from expanded priority vectors located in different columns. i.e.

$$P_i = (V_{i0}|V_{i1}|\dots|V_{ik}|\dots).$$

After matrixes of priorities are developed it is possible to calculate a priority vector for alternative options. If the quantity hierarchy layers equals N , then vector is calculated under the formula [8]:

$$p = P_{N-1} * P_{N-2} * \dots * P_0.$$

Studies	0.36
Friends	0.05
School life	0.29
Prof. training	0.16
Training for university exams	0.1
Music classes	0.03

a)

Quality of teaching	0.23
Equipment supply	0.12
Pupils' treatment	0.65

b)

School A	0.66
School B	0.15
School C	0.2

c)

Fig. 5. Developed matrixes of
a- first level;
b-second level;
c- third level.

The results after calculation of priorities based on explicit numbers (for descriptive reasons) for the first matrix enable us to draw the following conclusion: the most significant criteria in school selection are *Studies, School life and the Prof. training.*

In a matrix of second level the most important role for parents plays Pupils' treatment, then goes Level of teaching, and the last is Equipment supply.

Matrix of the third level gives the following conclusion: school A, by criterion Studies, considerably surpasses other schools.

In certain cases, when there is no one-valued occurrence of one fuzzy set into the other or when they are equal, the interpreting of the received results is not as simple, as in scalar form. The complexity lies in the necessity to compare membership functions received after the calculation for each option. And these functions might have quite elaborate configuration. In scalar form such comparisons are elementary. Therefore for comparison these numbers could be converted into habitual scalar form by methods based on calculations of the centre of gravity, the centre of maxima etc.

IV. CONCLUSION

Based on the above reasoning we can draw the following conclusion: transfer of expert estimates into fuzzy form allows to reduce errors by the human factor since instead of the average value, the set of expert estimates will be considered and also the level of expert certainty. The absence of assumptions and simplifications during the decision-making allows to say that described approach does not bring additional errors. Therefore, transfer into fuzzy form allows to use well proved methods in new conditions.

REFERENCES

- [1] I. Z. Batyrshin, A.O. Nedosekin, A.A. Stetsko, V. B. Tarasov, A.V. Jazenin, and N.G. Jarushkina, "Theory and practice of fuzzy hybrid systems," Ed. by N.G.Jarushkina. M: Phismatlit, 2007. (rus).
- [2] S. L. Blumin, I.A. Shujkova, P.V. Saraev, and I.V. Cherpakov, "Fuzzy logic: algebraic bases and appendices," Lipetsk: LEGI, 2002.-113 p. (rus)
- [3] V. N. Burkov, N.A. Korgin, and D.A. Novikov, "Introduction into the theory of organizational systems management," Moscow: Printing house "Librokom", 2008.-264 p. (rus)
- [4] K. Sugihara, Y. Maeda, and H. Tanaka, "Interval Evaluation by AHP with Rough Set Concept,". Berlin / Heidelberg : Springer , 1999.
- [5] V. V. Kruglov, M. I. Dli, and R. Y. Golunov "Fuzzy logic and artificial neural networks," Training manual. M: Publishing house of physical and mathematical literature, 2001. - 224 p. (rus)
- [6] L. Mylnikov and A. Maksimov, "Use social networks to make decisions," 3rd International Conference on Applied Social Science (Taipei, Taiwan, 15-16 January 2013), - pp. 137-139.
- [7] F. Novikov, "Discrete mathematics for programmers," S-Pt: Piter, 2004. (rus)
- [8] T. Saaty, "Decision-making: hierarchy analysis method,". Moscow: "Radio and communication", 1993-278 p. (rus)
- [9] T. Arslan, "A hybrid model of fuzzy and AHP for handling public assessments on transportation projects/ Transportation," N36, 2009, p. 97-112.
- [10] L. A. Zadeh, "Fuzzy sets," Information and Control, vol.8, N 3, 1965, pp.338-353.
- [11] Z. Wei-ying, L. Yan, J. Zhuo-Shang, and D. Lin, "Multi-objectives fuzzy optimization model for ship form demonstration based on information entropy," Journal of Marine Science and Application, Harbin Engineering University, 2006.

Regarding to Implementation of Genetic Algorithms in Life Cycle Management of Electrotechnical Equipment

Anton Petrochenkov

Perm National Research Polytechnic University - Electrotechnical Department
Komsomolsky Ave. 29, 614990, Perm, Russia

E-mail: pab@msa.pstu.ru

Abstract—Genetic algorithms regarding to life cycle management of electrotechnical equipment are considered. The concept of “techno-individual” is introduced.

Keywords: life cycle, electrotechnical equipment, genetic algorithms, techno-individual.

I. INTRODUCTION

One of the solution techniques, allowing creating effective algorithms for a wide range of tasks, is the usage of a subclass of the directed random search methods – methods of genetic modeling.

Numerous approbation of use and application of genetic methods in scientific and industrial spheres allow saying with confidence that when using the evolutionary methods, realization of the comprehensible well-founded decision search problems is always reached. In overwhelming majority of cases the use of natural analogues gives positive results. This is explained by the fact that the analogue taken from the nature was improved for many years of evolutions and at the present time has the optimum structure [1].

The idea is to consider these genetic algorithms regarding to life cycle management of electrotechnical equipment.

Let's consider the main advantages of genetic algorithms usage:

- Ability to manipulate with many parameters at the same time;
- Good applicability when solving large-scale problems of optimization;
- Leniency of assumptions at a goal function estimation that expands a class of tasks which can be solved by means of genetic algorithms;
- Simplicity and "transparency" in implementation;
- Application in tasks with changeable place (adapting to changeable place);
- Ability to arrive at enough good solutions fast.

Genetic algorithms use direct analogy to the mechanism of biological development.

Like biological development each technological product develops continuously from a stage of origination (setting up) to a stage of collapse (recycling), going through any environment effects and adapting for them.

Algorithms work with a group of "individuals" – population, each of which represents possible solution of the given problem. Each individual is estimated by a measure of its "suitability" according to how rationally solution of the problem suits to it.

In each generation of the chromosome genetic algorithms (coded solutions) are the result of application of some genetic operations [1].

II. THE KEY GENETIC RELATIONS REGARDING LIFE CYCLE MANAGEMENT OF ELECTROTECHNICAL EQUIPMENT

For description of the genetic algorithms when developing methods of the electrotechnical equipment' life cycle [2]-[6] estimation let's introduce some concepts.

“*Techno-individual*” – it's a type of any equipment (transformer, load etc), life cycling of which is considered with the view to biological evolution.

Initial conditions – datum values of the equipment parameters (for example, values of reliability including the level of the serviceability, revitalization, mean time between failures, probability of non-failure operation etc.).

Principal criteria, which are necessary to be followed, – to control the target level of the parameters values. It means that it is desirable during the resolution output to safe initial conditions with minimum costs.

Characteristics, defining the level of the organism development and its adjustment: possible methods of service, support of the set state, different from each other according to the degree of complexity and quality of execution.

Operations must precede the process of algorithms construction:

- Selection of the initial conditions;
- Criteria selection (criteria functions);
- Analysis and selection of the boundary conditions.

Selection operations of the initial level depend on the type of the current task and goals, which must be reached as a result of algorithm realization.

If it concerns maintenance of the parameters on the certain level and reliability index of the equipment in the course of its operation then as initial conditions it is reasonable to use data values of parameters and the indexes defined on a development stage in the form, installed by manufacturer and presented in the maintenance

documentation (in particular, in maintenance documentation for the specific type of the equipment).

In this case, the main task is conformity «real reliability of the equipment – reliability set as zero conditions» that is support of higher level of reliability.

It is necessary to introduce criterion of an optimality estimation which will characterize level of considered solutions under the control of the technical state of the equipment [5][6].

Realization of the genetic algorithm assumes the fitness function. Input data of this function gets binary chromosome and returns the number, showing how good this chromosome is. The given function will be the criteria for the algorithm forming. In the embodied genetic algorithm it is reasonable at first to define the worse chromosome (having maximum deviations from the target level of parameter) and to measure the index value.

The got number is called the bad one with respect to which the quality of the other chromosomes is valued: fitness of the chromosome is calculated as difference between index value, set by the given chromosome and bad index value.

Let the chromosome is $X = x_1, x_2$ assigned with the value of the chosen index n (for example, some index of the equipment sustainability) and the chromosome $Y = y_1, y_2$ with the index value n' . If $n' > n$, so X – the worst chromosome, and $fitness(Y) = n' - n$.

As a result of these calculations each chromosome gets its fitness- number, which turns to be relatively small for the bad ones' and relatively big for the good ones', optimal reliability index, provided with given technologies.

Let, for example, have a look some time resolution, limited by the first equipment failure. So as a criteria it is reasonable to use the index value of the mean time between failure of the given type of high-voltage electrotechnical equipment. The highest the value of the criterion function, the more optimal is the method of the equipment maintenance, the more optimal is the set of methods, providing this maintenance and their content (the set of the actions, the level and the quality of their realization).

The criteria choice is the main preliminary step when building the algorithm, because only it will define the further algorithm filling, its logical direction and convergence.

One of the important moments is the analysis and the limitation selection of the given system and for the algorithm is its functioning. Limitation can absolutely different indeed:

Economic ones – in the network of the limited founding;

Technological ones – limitation in the instrument base and documentation base, skilled personnel and etc;

Temporal constraints – conducting the actions and etc.

Limitations are the necessary addition, which lets to take into account the particularities of the real technological objects and systems, factors, influencing on its activity. The majority of the real tasks have deterministic character.

Limitations can be also made as some rules and conditions and can be used when selecting and making the possible variants of solutions (heuristics).

The coding of the chromosomes is implemented according to the numbers of the methods, which are on the

stage of “techno-individual” development.

In this case equipment is characterized by particular set of maintenances, with the help of which life cycle following is put into effect [6]. Each application for different methods can be done in different ways. And this is the difference in chromosome genes.

This difference will be: in application of various instrument base (for example, in one case optical resources are used at a visual estimation of a state of pendant basic insulators, and in other – there is nothing), taking into account some external factors, in application to various approaches concerning state estimation (up to distance and points control), and also in application to various methods of results processing.

It is necessary to add that all complex of actions should correspond to the circuit of support of the set technical state of the equipment (and to include not only actions for an estimation of the technical state in the form of monitoring and diagnostics, but also reducing and preventive operations etc.).

Changing of the purposes and the problems solved by system, can lead to revision as maintenances of preliminary operations, and a way of "filling" of chromosomes (for example, concerning economic aspect of a problem, the concepts connected with a given problematic and components can be used).

For the interpretation of genetic concepts concerning “techno-individual”, the following definitions are introduced.

Reproduction. It includes some elements of standardization, as the developed techniques of service and decision-making. As a result of a reproduction we will receive element high voltage electrotechnical equipment and a set of the regulated actions by means of which its support and the control of its technical state will be carried out.

Mutation. In whole, a mutation – the ambiguous phenomenon in most cases calling negative consequences. In this case under mutation we will understand effect of some factors on equipment maintenance. To mutation factors we refer:

- Environment conditions (the external factor, in view of object distribution is important);
- Skills degree of operating staff.

Crossover. Refinement of existing techniques of the control and equipment service, on the basis of expert estimations and other sorts of the analysis of existing systems. Framing of concrete solutions on upgrading, support of any solutions of documentation base is necessary. The completion phase is development of the standard of the enterprise regulating and considering all features of a set of actions under the control and support of the set technical state of the equipment.

III. GENERIC FLOWCHARTS OF LIFE CYCLE OF ELECTROTECHNICAL EQUIPMENT

Generic flowchart of life cycle of electrotechnical equipment with the use of genetic algorithm theory is showed in the Fig. 1. On this Fig. 1 the next blocks are shown:

Block 1 – operation complex, corresponding to the concept of the genetic modeling reproduction (on this stage

population initialization of “techno-individuals” or ”techno-individual” for the private case is happening).

Block 2 – carrying out actions corresponding to the process of “techno-individual” ability to live.

Block 3 – complex of operations according to criterion assessment (fitness) of «techno -individual» (includes calculation of value of the selected goal function (fitness

function) and an estimation of fitness level of every «techno-individual» according to the received values).

Block 5 – chain of operations and decisions, corresponding to the definition “ survival rate” of the “techno-individual” (decision about possibility of its further vital capacity or life cycle end, if it is necessary).

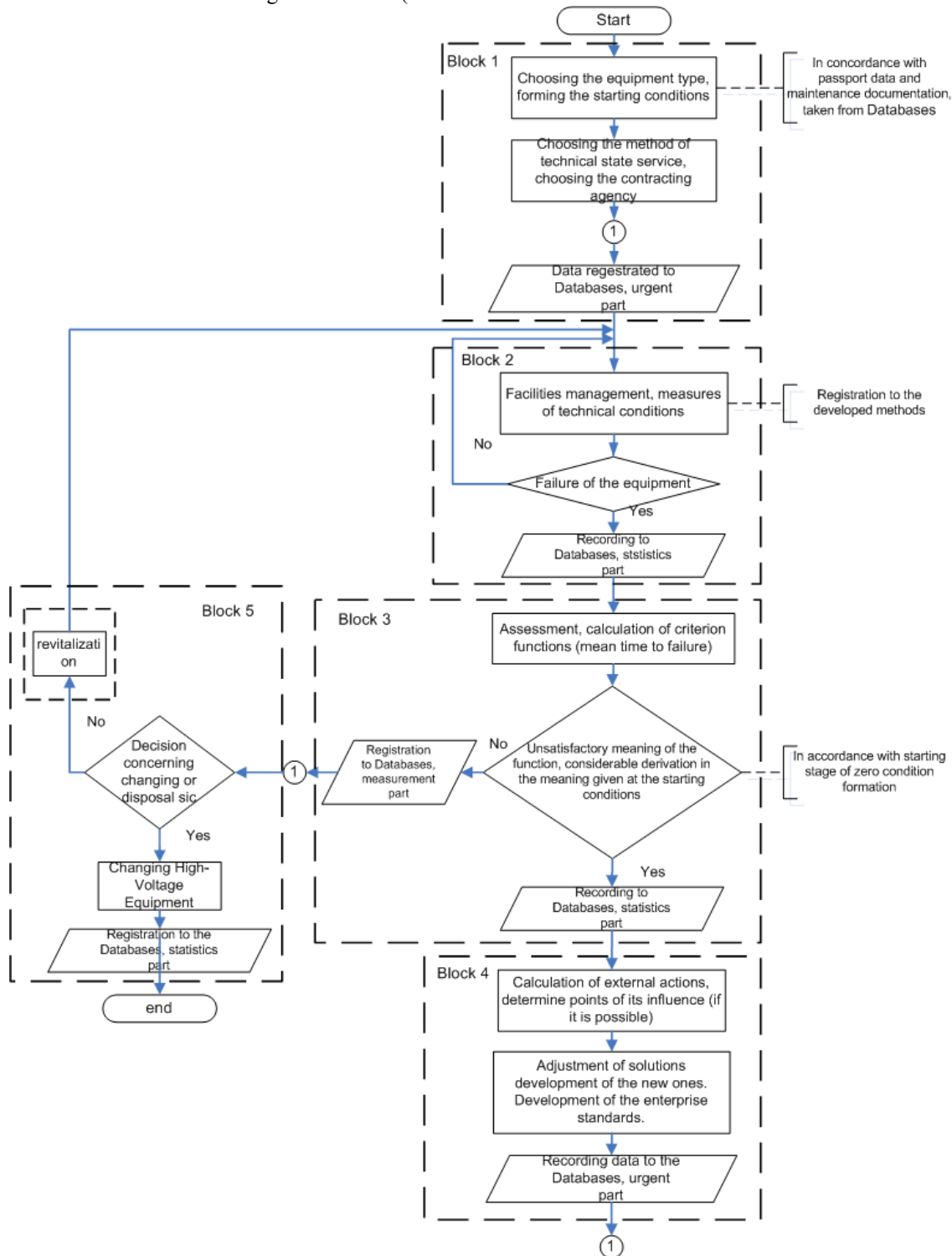


Fig.1. Generic block-schematic life cycle of equipment with the use of genetic algorithm theory.

The set of actions in maintenance and support of equipment life cycle will be the base for the formation of the possible variants (heuristics) for our task.

Let's separate some components of the scheme [7]-[10]:

- A – Monitoring;
- B – Diagnostics;

C – Maintenance, testing, measurements;

D – Failure control works;

In addition, let's add additional components:

E – Expert forecasting system of the technical state.

F – Integral estimate of the technical state.

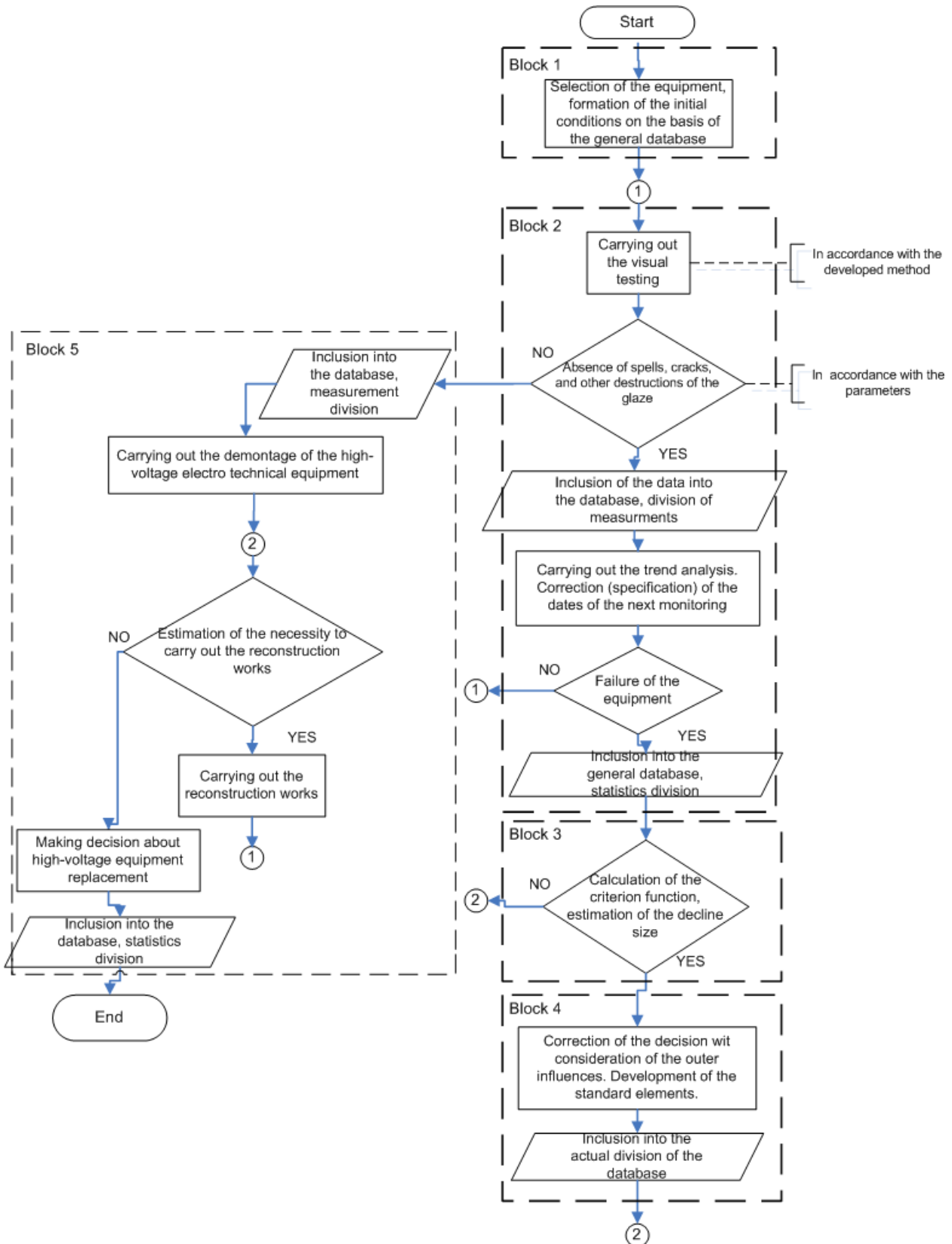


Fig. 2. Generalized insulators life cycle analyses algorithm with application of the second heuristics (the monitoring based on visual survey).

As a result of the primary sanction combinations in technical state control with additional ones will be get possible variants, which are heuristics :

- $H_1 = \{AE\},$
- $H_2 = \{AF\},$
- $H_3 = \{BE\},$
- ...
- $H_8 = \{DF\}.$

In the real task the number of combinations and their width can be much more. They can be added different preliminary set rules and conditions.

For example, let's have in more details the realization of the algorithm for the suspended insulators in the network of the simplify problem (Fig. 2).

- The activities for quality control of the technical state are:
- Monitoring system [3];

– Diagnostics system [11].

Concerning the diagnostics we will decide that this action either take place (A) or not (B).

Regarding the system of monitoring several possible variants were suggested; let's have a look possibility of its implementation:

- 1 – monitoring by «hectic rush»;
- 2 – monitoring, based on the visual examination of the equipment;
- 3 – monitoring with the use of infrared and ultraviolet control;
- 4 – combined monitoring.

By variants combination we get 8 possible heuristics:

$$H_1 = \{1B\},$$

$$H_2 = \{2B\},$$

$$H_3 = \{3B\},$$

$$H_4 = \{4B\},$$

$$H_5 = \{1A\},$$

...

$$H_8 = \{4A\}.$$

The meaning of the blocks on the Fig. 2 fits the structure for the Fig. 1. Decomposition of the blocks 2 and 5 is more detailed, because it corresponds to the occurrence (monitoring high-voltage equipment on the base of visual examination). Moreover, block 2 is the base of the information fulfillment system (getting and analysis of the information, necessary for the following stages).

IV. CONCLUSION

The difficulty and the goodness of action realizations in technical state of the equipment increases, the number of the operations required for the algorithm realization increases. But in spite of the apparent bulking, algorithms are simple in their accomplishment because of their coherency.

Electronic educational resources were developed using an educational process for training students with the specializations “Electrical Power Supply,” “Automation of Technological Processes and Production,” and “Automated Management of Product Life Cycle” of Perm National Research Polytechnic University [12].

Works on this direction are conducted within the Russian Foundation for Basic Research Grant of Russia No 14-07-96000 “Development of an intellectual decision support system to ensure of energy facilities trouble-free operation”.

REFERENCES

- [1] L. A. Gladkov, V. V. Kurejchik, V. M. Kurejchik, “Generic algorithms,” Under the editorship of V.M. Kurejchika. – M: FIZMATLIT, 2006. (rus).
- [2] D. K. Eltyshev, A. B. Petrochenkov, S. V. Bochkarev, “Application of genetic techniques in the implementation of the life cycle support system of electrical equipment. Energy. Innovative trends in the energy sector. CALS technologies in energy,” Proc. of the Third All-Russian Internet conference. Perm, 2-30 November 2009. Perm: Publishing House Perm State Technical University, 2010. P.149-159. (rus)
- [3] A. B. Petrochenkov, A. V. Romodin, “Energy-optimizer complex,” Russian Electrical Engineering, 2010, vol. 81, no. 6, pp. 323-327. doi: 10.3103/S106837121006009X.
- [4] A. B. Petrochenkov, A. V. Romodin, N. I. Khoroshev. About one formalized method of an assessment of administrative decisions (on an example of management of electrotechnical objects). Scientific and technical sheets Saint Petersburg State Polytechnical University, no. 5 (87), pp. 166-171, 2009.
- [5] A. B. Petrochenkov, “Methodical Bases of the Integrated Electrotechnical Complexes Life Cycle Logistic Support,” Proc. of the First International Conference on Applied Innovations in IT, E. Siemens (editor in chief) et al. Dessau, Anhalt University of Applied Sciences, 2013. – P.7-11. doi: 10.13142/kt10001.02.
- [6] A. B. Petrochenkov, “Regarding Life-Cycle Management of Electrotechnical Complexes in Oil Production,” Russian Electrical Engineering, 2012, vol. 83, No.11., pp.621-627. doi: 10.3103/S1068371212110090.
- [7] A. B. Petrochenkov, “On the Problem of Development of Models of Processing Operations Performed during Repair of Electrical Engineering Complex Components,” Russian Electrical Engineering, 2013, Vol. 84, No. 11, pp. 613–616. doi: 10.3103/S1068371213110096.
- [8] N. Beliaeva, A. Petrochenkov, K. Bade, “Data Set Analysis of Electric Power Consumption,” European Researcher, 2013, Vol.(61), № 10-2, P.2482-2487. doi: 10.13187/issn.2219-8229.
- [9] A. B. Petrochenkov, S. V. Bochkarev, A. V. Romodin, D.K. Eltyshev, “The Planning Operation Process of Electrotechnical Equipment Using the Markov Process,” Russian Electrical Engineering, 2011, Vol. 82, No.11., pp.592-595. doi: 10.3103/S1068371211110113.
- [10] V. P. Kazantsev, A. B. Petrochenkov, A. V. Romodin, N. I. Khoroshev, “Some aspects of Technology of Use of Electrical Objects on the Basis of Methods of Short Term Forecasting of Technical Condition,” Russian Electrical Engineering, 2011, Vol. 82, No.11., pp.600-606 doi: 10.3103/S1068371211110058.
- [11] A. B. Petrochenkov, E. M. Solodkii, “On the Methods for Constructing Failure Models of Complex Systems,” Russian Electrical Engineering, 2011, Vol. 82, No.11., pp.623-627. doi: 10.3103/S1068371211110125.
- [12] S. V. Bochkarev, A. B. Petrochenkov, A. V. Romodin, “Automation of Life Cycle Management of Electrotechnical Production: Studies. The grant,” Perm: Publishing House Perm State Technical University, 2008. (rus).

Storing Data in the Trial of Complex Technical Products

Igor Shmidt

Perm National Research Polytechnic University - Electrotechnical Department

Komsomolsky Ave. 29, 614990, Perm, Russia

E-mail: shmidt@msa.pstu.ac.ru

Abstract—The article considers the ways of organization of databases for the storage of the results obtained during testing. A new variant of the organization of the data to ensure the ability to write to the database different sets of parameters in the form of chronological series. The required set of parameters depends on the modification of the tested technical installation.

Keywords: testing of complex technical products, noSQL, chronological database, document-oriented DBMS.

I. INTRODUCTION

Among the various tasks that require high-volume data storage is allocated the task of gathering and storing of data relating to tests of complex technical products [1,2,3]. A feature of such tests is:

- You need to store a large amount of information that describes the trends of measured parameters during the test. Some tests require the speed of data collection to a few KHz and the duration of the test can be up to several days;
- There is no strict structure of the data collected. Various modifications of products can have different sets of parameters. Thus, over the entire life cycle of the product can be tested with various modifications;
- Data retention period should correspond to the total time of the product life cycle;
- In addition to storing the measured during testing the parameters you want to store data pertaining to the operation of the test items during its life cycle.

Much of the information that accumulates when testing are the results of observation over parameters of the process of testing. When observing the values of the parameters are recorded and linked to the moment of observation. The result is an ordered in chronological sequence of measured values is called a time series.

II. WHY RDBMS IS NOT FIT FOR STORAGE OF TEST RESULTS?

Traditional industrial relational DBMS are not adapted for efficient storage and processing of time series. This is because the relational DBMS include the following mechanisms:

- Normalization, which on the one hand ensures the absence of redundancy and logical errors when updating and sample data, but other decreases the capacity of

- Support of the transaction, which ensures the reliability and consistency of data, but requires substantial resources.

These mechanisms are useful when solving business problems, but they are not needed in the problems of collection and storage of information about the tests. As a result of the performance of processing time series is very low, and the complexity of the description of the application logic high.

To solve the problem of storing temporary data, there are special temporal extension RDBMS, but most of the developments in this direction are intended for storage of slowly varying data.

It is proved that the storage time series in a relational form causes some redundancy in the volume of content and information. Storage of temporal data in a relational is the amount, which is calculated by the formula [4,5].

$$T = \sum_{i=1}^K \sum_{t=1}^{Z_i} [\sum_{j=1}^N V_{ij} + V_t], \quad (1)$$

where

T - the total volume of information;

V_{ij} - the amount of information contained in the j attribute tuple i;

K and N - number of records and columns in the table;

V_t - size of the temporary attribute;

Z_i is the number of states tuple i.

While the amount of data in the same table excluding temporality is expressed by the formula:

$$T = \sum_{i=1}^K \sum_{t=1}^{Z_i} V_{ij}, \quad (2)$$

where

T - the total volume of information;

V_{ij} - the amount of information contained in the j attribute tuple i;

K and N - number of records and columns in the table.

Indeed, it is easy to see redundancy even if only saves the current value of the entity in the temporal RDBMS.

Possible decrease in the volume V_t by changing its type and the creation of a «cheap» surrogate primary keys for the identification of a tuple and calculate its temporary values relative to the parent of a tuple. Decrease in volume V_t through the use of surrogate key is based on the records of

relative time values for each tuple.

The total volume of the stored information will be presented to the expression of the form (3). This method will always be much more economic than presented by the expression (1).

$$T = \sum_{i=1}^K [V_t + \sum_{t=1}^{Z_i} \sum_{j=1}^N V_{ij}] \quad (3)$$

On the basis of comparison of expressions (1) and (3) we can construct a mapping of the sets being a description of the database structure. So, for the formula (1) corresponds to the set of (4) and formula (3) corresponds to the set of (5).

$$\{\{T_i, V_j\}\} \quad (4)$$

$$\{T_i, \{V_j\}\} \quad (5)$$

where

T_i - timestamp;

V_j - the value of the tag.

When creating the system of tests of important factors that you need to evaluate the DBMS for storage of large amounts of data, are: the complexity of the structure, the volume occupied by the data and the speed of receiving data.

III. USING NOSQL STORAGE OF TEST DATA

An alternative approach is the use of noSQL [6]. This approach will build a system capable of adapting to the increasing amounts of data and effectively handle them. NoSQL solutions provide much higher data throughput than traditional DBMS.

Storage time series are most adapted store known as «key-value». Such databases are the easiest way to store multiple values associated with the same timestamp tag. Often such databases are included with the software for measuring or checking or integrated into the SCADA system [8]. Examples include the database Citadel, which is part of the graphical programming platform National Instruments - LabView. To improve the characteristics of the storage of historical data in databases is usually possible to buffering and compression.

A database that is organized as a «key-value», allow to implement storage of large amount of historical data, providing high speed write and retrieve data. However, if you change the structure of the stored data will be impossible to make a selection of the parameters corresponding to one physical parameter.

The optimal solution is to use a document-oriented database. Data model such storage allows you to merge multiple key-value pairs in an abstraction called an «document». Documents can be nested structure and form collections. Collections can contain other collections. It is this capability that allows to model the relationship is one-to-many.

Such use document-oriented database unusual, because these data are not the document.

Document-oriented architecture DBMS allows you to use a hierarchy of nested and move to a higher level of General

information about the entity . Such a data model will eventually permit from the base type of the provision of a set of values (5) go to the form (6)

$$\{T_0, \{T_m, \{T_s, \{T_i, V_i\}\}\}\}, \quad (6)$$

where

T_0 - the timestamp of the start of the test;

T_m - timestamp minutes;

T_s timestamp seconds;

T_i is the timestamp of the smallest interval of time;

V_i - the value of the tag.

The use of document-oriented database allows identifying a common data structure in the form of fuzzy structures -- structures that can be dynamically expanded. The data is organized as a tree that allows efficient indexing by navigating to the key elements of the tree. This approach is effective because the majority of requests contain the time as the main parameter of search. Thus, a custom look at the data will coincide with the hierarchical structure of the applied model of data in most of the queries. Structure, which is optimal for indexing time, also represents a structure of nested objects, which fully corresponds to the real logic of the objects of the test.

This approach to the implementation of the database structure, there is virtually no duplication of data, and data integrity is ensured by a hierarchical structure.

The proposed flexible structure of the database allows to expand the horizontal functionality of the database (the add parameter types, event types, etc), and also will allow to store the compressed ranks, and automatically, transparently to carry their processing.

For the implementation of such structure will use the database MongoDB [8][9].

IV. THE STRUCTURE OF THE DATA IN PRACTICE

In the end, the overall structure of the developed document-oriented storage will look like the following:

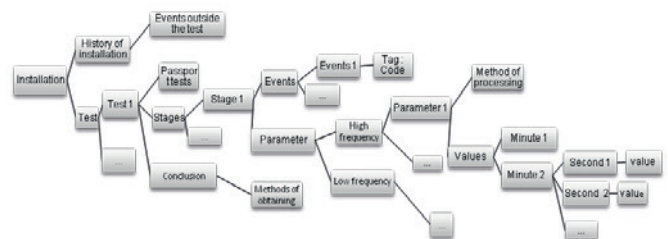


Fig. 1. Example data structures.

Such a structure allows to store:

- the results of observation over parameters of the process of testing;
- meta-information about testing;
- information about the operation of the test items during its life cycle.

Storage is not historical information in a document-oriented database can be organized as a projection of the structure RDBMS containing this information. This

structure will contain all the necessary information, ways of transformation parameters, directly series, information on testing. Storage same time series in the structure of the document requires additional study.

To improve data search and to conserve space is proposed to use the hierarchy of the document and represent time series in the form of a tree. There is a particular task of determining the optimal nesting branches of the tree. In practice, it is necessary to find a compromise between the memory size and speed. For this you must define the target function f and solve optimization problem

$$f(B,C) = k_m f_m(B,C) + k_v f_v(B,C) \quad (6)$$

$$f(B,C) \rightarrow \min,$$

where

B - nesting tree branches;

C - the number of elements in the tree;

f_m - function determining the memory footprint;

f_v function determining the speed of access to data;

k_m and k_v – weights.

REFERENCES

- [1] S. Schubring and I. Munoz, "Field Performance Testing from an Operators Point of View," Gas Turbine User Symposium 2005. -Las Vegas, Nevada, 2005.
- [2] D. Popov and I. Shmidt, "Development of the functional structure of the software complex tests of gas turbine units with a capacity up to 40 MW," Research and innovation, vol. 6, pp. 264–270, 2012.
- [3] B. V. Kavalerov, V. P. Kazantsev, I. A. Shmidt, "Simulator and Semi-Nature Testing of Gas-Turbine Power Units Control Systems, Information and Control Systems," St.-Petersburg, Russia, vol 5, pp. 25-31, 2009.
- [4] S. Navathe, "Temporal Extensions to the Relational Model and SQL," in A. Tansel, J. Cliord, S. Gadia, S. Jajodia, A. Segev, and R. Snodgrass, editors; Temporal Databases: Theory, Design, and Implementation. / S. Navathe, R. Ahmed. – Benjamin/Cummings Publishing Company, 1993. – pp. 92-109.
- [5] R. Snodgrass, "The TSQL2 Temporal Query Language," Kluwer Academic Publishers, 101 Philip Drive, Assinippi Park, Norwell, Massachusetts 02061, USA, 1995
- [6] G. Vaish, "Getting Started with NoSQL," Packt Publishing, 2013.
- [7] D. Bailey and E. Wright, "Practical SCADA for industry," Oxford(GB): Elsevier, 2003. -304 p.
- [8] K. Banker, "MongoDB in Action," Manning Publications, 2011. p. 375.
- [9] K. Chodorow, "Scaling MongoDB," O'Reilly Media, 2011. p. 62.