computing@computingonline.net
www.computingonline.net

ISSN 1727-6209
International Journal of Computing

DISTRIBUTED REAL-TIME MEASUREMENT SYSTEM USING TIME-TRIGGERED NETWORK APPROACH

Piotr Bilski [1)], Wiesław Winiecki [2)]

[1)] Warsaw University of Life Sciences,
ul. Nowoursynowska 159, 02-667, Warsaw, Poland,
piotr_bilski@sggw.pl,
[2)] Institute of Radioelectronics, Warsaw University of Technology,
ul. Nowowiejska 15/19, 00-665, Warsaw, Poland,
w.winiecki@ire.pw.edu.pl

Abstract: *The paper presents results of the examination of the deterministic network used by the distributed virtual instrument. Software technology applied to control measurement data transfer between the real-time components was presented. Configuration of the laboratory test stand, designed to examine deterministic network is described. Results of the research are presented and conclusions, as well as future prospects iterated.*

Keywords: *virtual instrumentation, Real-Time systems, integrated programming environments*

1. INTRODUCTION

Distributed measurement systems (DMS) are widely exploited in both professional and scientific applications. Their importance grows as the new technologies of the communication between the modules of the system emerge. DMS range increases steadily, thanks to the mobile technologies (such as cellular telephony systems) and usage of Internet. On the other hand, one of the required assets of such a system is its reliability. Multiple applications, namely industrial control and monitoring, diagnostics of machinery and production processes often require precise timing in data acquisition and communication between the modules of DMS [1]. Therefore, operations performed by the modules also must have duration defined with high accuracy. The ability to work in the Real-Time (RT) mode is a common example of the deterministic time operations' application.

As the complexity of the DMS tasks increases, RT mode becomes important not only for the operations performed in a single module (for example, personal computer or PXI component), but also refers to the data transferred between the modules. Real-Time Operating Systems (RTOS) play here important role. Unlike General Purpose Operating Systems (GPOS), in RTOS execution time of the program function can be precisely determined. Therefore such solutions become popular in the Virtual Instrumentation (VI) design and applications (where abilities of the instrument depend on both hardware and software part). Exploitation of VI technology in DMS design allows to obtain powerful solutions in short time period [1]. Therefore such environments as Agilent VEE or LabVIEW (from National Instruments company) are invariably popular. Application of the latter to the RT system design is easy thanks to LabVIEW RT Module (added to the environment from version 7.0). One of the most popular communication technologies used in DMS is the Ethernet standard computer network supported by TCP/IP protocol. Although such a solution delivers high reliability of data transmission, it does not assure that data will be transferred deterministically, i.e. in predetermined time instants. Obtaining determinism in the Wide Area Network, including Internet, is not possible, however, numerous attempt are made to obtain deterministic transfer in LAN configurations. One of them is a new approach in LabVIEW 8.0 (and later) environment.

2. PROBLEMS OF THE REAL-TIME WORK REGIME IN DMS

The RT mode, unlike the traditional MS work mode, allows to obtain deterministic time dependencies between the software functions in the VI. This enables taking measurements in the time instant defined with high accuracy. There are two RT modes: Soft Real-Time (SRT) and Hard Real-

Time (HRT). The former assures that the time requirements are fulfilled most of the time, however, sometimes they can not be met. The latter assures that the time requirements are fulfilled all the time. The experiments with the traditional VIs designed using traditional methods and environments revealed

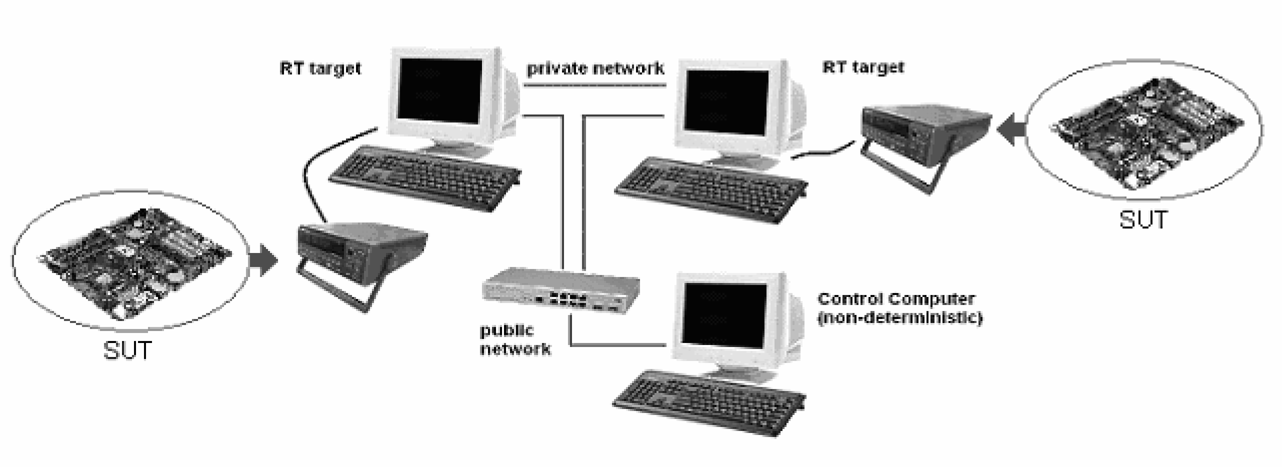

Fig. 1 – Architecture of the time-triggered network [3].

[2] that in the application working under the general purpose operating system (GPOS) only SRT is possible. Moreover, strict design rules of the program functions must be met. To obtain HRT mode, specialized hardware or software solutions are required. In [3] the possibility of using RT operating system (RTOS) to support LabVIEW programming environment was examined. Using the RT module of this environment it was possible to create MS consisting of the deterministic and non-deterministic part. The former was responsible for performing the time-critical tasks, such as data acquisition and signal processing, while the latter was used for presenting results in the front panel. Expanding the characteristics of such a VI on the distributed system, it is apparent that the factors affecting the time determinism are the measurement software, operating system and transmission media. The latter are in most cases implemented as the wired (Ethernet) and wireless (WiFi) computer network. The approaches to obtain the deterministic transmission through the network revealed [4] that it is impossible in the wide range (the Internet) and the only option to assure conditions close to the deterministic transmission is to use the Local Area Network.

2.1. DETERMINISTIC NETWORK ARCHITECTURE

The main drawback of the Ethernet-based LAN networks with respect to the Real-Time measurement systems is the inability to precisely determine time of delivering the data to the target node. DMS packets travel simultaneously with packets from other nodes, collisions and losses occur, as well as traffic jams. Therefore creating reliable system in the network used by another computers and devices (remote printers, etc.) is difficult. One of the solutions proposed by National Instruments to solve the problem is the time-triggered network. Its aim is to ensure determinism in the Ethernet data transfer [3]. The idea is to isolate general-purpose network from the RT network and create two sub-networks with common control node. The former, called public network, can be used freely by both DMS components as well as another computers, performing their own tasks, not related to the measurement process. The latter is reserved only for the modules working in the RT mode (called RT targets). A special role in the system plays the computer controlling RT targets. It uses public network to communicate with RT targets, deploy programs on their hard drives and gather results for presentation on the front panel (therefore programs run on RT targets do not require any particular elements on their front panels). The RT data (for example, when the system is required to perform cross-correlation function calculation between the signal acquired by two different targets) are transferred between RT targets using private, time-triggered network, which is physically realized using separate network interfaces and cables. This architecture is presented in Fig. 1.

Work in the RT mode is possible using RTOS installed on the RT targets with all the required software to control network communication, DAQ operations, etc. The control node runs under GPOS, such as Windows XP. Its task is to present results of the data processing to the user and perform non-deterministic operations. To be able to communicate through both networks, RT targets require two interface cards: one of them is considered “primary” (used for public network), while the second is described as deterministic (aimed at RT communication). Public network interface cards obtain IP address and are accessible using TCP/IP protocol. They are used to communicate (in both directions) with the control node and deploy program on RT target. The deterministic network interface cards are used only after programs were deployed and executed on RT targets. Configuration of the network is possible through Measurement and Automation Explorer, included to LabVIEW environment. RTOS deployed from control node to RT targets works only with particular hardware equipment, requiring Pentium III (or better, or equivalent, such as AMD Athlon) processor, FAT32 hard drive partition, and network interfaces with Intel chipsets: 82559, 82540 or compatible.

2.2. SOFTWARE SUPPORT FOR THE DETERMINISTIC NETWORK

Software part of the RT target is designed in LabVIEW environment, according to the conditions imposed by the rules of the RT program design [6,7]. In the time-triggered technique the most important parts are the functions executed on the RT targets and methods of communication between the RT targets. To obtain efficient measurement application working under RTOS, a specific strategy of the code design must be implemented. For example, it is not advised to put into the code operations requiring writing or reading from hard disk. Efficient method of communication is usage of the shared variables [5]. They exist in two versions: “Network” and “Time triggered”. The difference lies in the method of transporting contents of the variable. The former uses public network, the latter – deterministic network. To obtain RT DMS with deterministic communication technique, only time-triggered variables should be used.

RT programming requires usage of the timed loops, which iterate at predefined time instants. They work similarly as standard loops, but the designer can determine period time, priority in the RTOS and connect timing source [6]. Configuration of the deterministic network depends on the number and types of the RT shared variables used in the code. In every iteration of the main loop during the program execution values of the variables can be transferred through the network. Length of the communication cycle is measured in microseconds. Sending the value of the shared variable requires defined amount of time, depending on the type of the values stored in the variable. The system designer’s task is to assign the shared variables to the time slots in the cycle configuration manager (see Fig. 2).

Besides the shared variables, in the cycle there must be a time slot for the control data (related to the start cycle) reserved [5].

Communication between RT targets is performed in predefined time instants. VIs deployed on RT targets work in cycles, in which they commence calculations and exchange data. The transfer of data is performed in time-windows defined during the design process. Example of the configuration of the time-triggered shared variables generated by the RT targets with 192.168.0.105 and 192.168.0.108 IP addresses respectively, is presented in Fig. 2. Notice that the cycle must be long enough to fit “cycle start” operations and all shared variables communication. Therefore simple DMS with small number of shared variables can iterate faster. Time windows reserved for the particular data transfers can not overlap. When the modules of the DMS are using RT shared variable, during their initialization a synchronization is commenced. This is the process of the determining the delays in the network (which are in most cases equal to tens of microseconds).

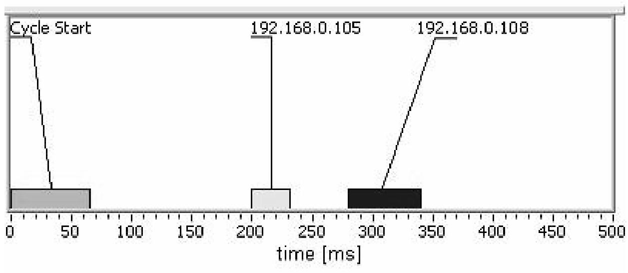

Fig. 2 – Configuration of the operating cycle in the time-triggered network.

3. LABORATORY TEST STAND

To fulfill requirements of the time-triggered network technology, specific three computers were linked into the measurement system. Two, working under RTOS, were performing tasks of the data acquisition and processing tasks. Their parameters were: Pentium III 733 MHz and Pentium III 750 MHz processors, 256 MB of RAM and two network interfaces based on the Intel 82559 chipset. Additionally, one of the computers was equipped with the data acquisition card PCI-6023E from National Instruments company, connected to the signal generator. Both RT targets worked under the RTOS deployed by the LabVIEW RT module.

The third computer, working under GPOS was the control computer, collecting the processed data and presenting results on the front panel. Its configuration included: Intel Celeron 2 GHz processor, 512 MB of RAM and network interface based on Intel 82550 chipset. The operating system that controlled the whole DMS was Microsoft Windows XP Professional. The integrated programming environment used to design and deploy VIs on the RT targets was LabVIEW 8.2 as the newest environment available at the time of conducting experiments.

Configuration of the computers performing deterministic tasks was done using Measurement & Automation Explorer tool. It allowed to assign the IP numbers to the RT targets, and upload the required modules of the RTOS (such as device drivers). The scheme of the laboratory test stand is in Fig. 3. The Ethernet network connecting all the computers through the switch has speed of 100 Mb/s (though LabVIEW RT module is compliant also with 1Gb/s standard).

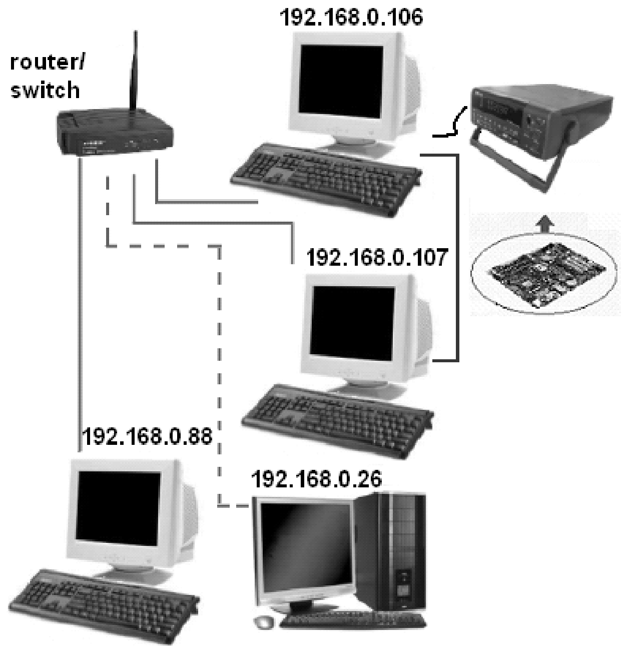

Fig. 3 – Scheme of laboratory test stand for deterministic network examination.

The computers with IP numbers 106 and 107 are RT targets, additionally connected through the private network. The computer with IP number 88 plays role of the system controller. The computer with IP number 26 is the server having purposes not related to the deployed DMS. Its auxiliary function was to generate additional traffic in the public network. The programs deployed on the RT targets were simple applications: one mainly to acquire measurement data (deployed on the computer no 107), another to process them – calculate spectral density and perform harmonic analysis. (deployed on the computer no 106). The acquisition program uses DAQ-MX drivers to control the DAQ card. The data got from the card is put into the shared variable. The processing program obtains the data from the variable and performs signal analysis. Results of the calculations are then send to the control computer, which visualizes them on the front panel.

4. EXAMINATIONS OF THE DETERMINISTIC NETWORK

Experiments considering the abilities of the deterministic network were divided into three groups. In the first, characteristics of the shared variables were determined. The second contains experiments considering analysis of the network congestion for different types and sizes of the data transferred between RT targets. The third was aimed at the analysis of the usage of the CPU and memory during RT program execution in both deterministic machines.

Table 1 – Size of the shared variable and time required to transport it through network in relation to the data type and number of its elements

	Int32 type		Double type	
Number of elements	Variable size [B]	Transmi-ssion time [μs]	Variable size [B]	Transmi-ssion time [μs]
1	4	33	8	33
2	8	33	16	33
4	16	33	32	33
8	32	33	64	35
16	64	35	128	43
32	128	43	256	60
64	256	60	512	95
128	512	95	1024	163
256	1024	163	2048	239
512	2048	239	4096	425
1024	4096	425	8192	762
2048	8192	762	16384	1435

4.1. CONFIGURATION OF THE SHARED VARIABLES

The time required to transfer the data through the shared variable depends on the size of the data. Minimal time required to send a single scalar variable, for example an integer number, is equal to 33 μs. The control data used during the transfer cycle start (Fig. 2) require 66 μs. We determined the amount of time required to transfer scalar and table data of different types and size. The examined types were Int8, Int16, Int32 and Double. Time required to transfer data (set in the configuration tool as in Fig. 2) and size of the variables are in Table I. Number of the matrix elements was set according to the purpose of the instrument, i.e. spectral analysis.

Experiments revealed that the time required to transmit the shared variable does not depend on its type (internal organization), but its size. Matrices requiring identical memory size (in bytes) need identical time to transfer, even if they contain different number of elements (such as 128-element Int32 array and 64-element Double array in Table I, because the latter needs twice memory size as the former). Note, that the time is constant for the smaller matrices (up to 8 elements for Int32 type in Table I) is constant and equal to 33 μs. This is the minimal time required to send a scalar variable. This means that if the designer needs to send through the deterministic network more than one scalar variable, it is more convenient to group them in the table and send using one shared variable, than define one shared variable for every value. In the destination RT target, the data can be broken into the scalar values again and then processed as originally planned.

The only problem is the shared variable that is meant to be used with the data of different types, for example Int16 and Double. Because it is not possible to create shared variable for the Cluster type (destined to store variables of various types), sending such data requires selecting the type of the greatest accuracy, for the above example it would be Double. The dependency between the data transmission time and the number of the matrix size for the selected data types is presented in Fig. 4.

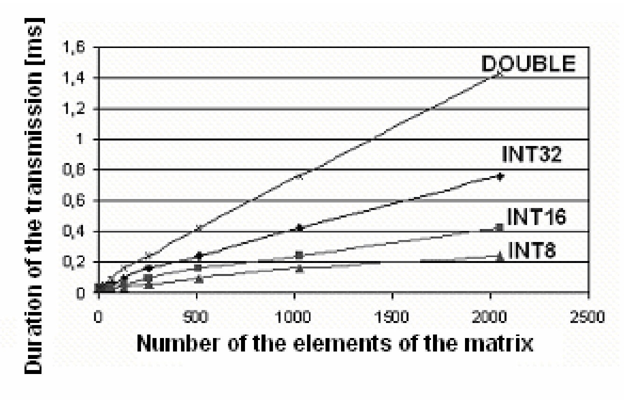

Fig. 4 – Relation between the number of the matrix elements and time required to transfer it through the time-triggered network for the main data types.

4.2. EFFICIENCY OF THE DETERMINISTIC NETWORK

To examine the load of the deterministic network during the data transfer, an Ethernet network analyzer Ethereal [9] was used. Its task was to calculate the packets traveling through the network interface. Because the deterministic network is not accessible from the control computer, the measurements were taken on its interface during usage of the deterministic shared variable, as well as the public shared variable. Comparing these two quantities allowed to measure congestion of transferring the data of the particular size and type. To be able to count the packets that travel between the RT targets, a simpler device – hub had to be installed instead of switch. Its purpose was to send all the packets to all the computers connected to it. Note, that this way we get only the approximated results rather than the exact information about the size of the data transferred through the deterministic network, as the latter uses RT network protocol of different packet structure that Ethernet. However, when there is no specialized analyzer of the deterministic network, the applied method gives a good general overview of the size of the data transferred through the network. Experiment results for the experiment of the 1024-element shared variable are in Fig. 5. The analysis required observing the network traffic for the predefined time (here one minute) after generating the data transfer with the frequency of 1Hz. The packet number sent through the deterministic network is a difference between the number of packets sent through the public network when a public variable is used (the whole traffic goes through the public network), and when a deterministic variable is used (only the control packets go through the public network).

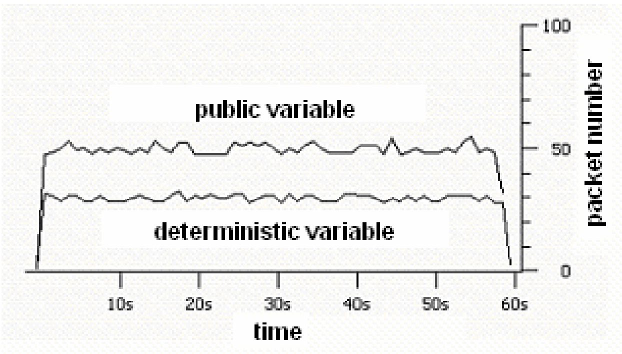

Fig. 5 – Comparison between traffic intensity for the non-deterministic and real-time shared variable.

Table 2 – comparison of network traffic for two types of shared variable carrying 1024-element table for 60 sec. with frequency of 1 Hz

Parameter	Public variable	Deterministic variable
Sum of the packets	2830 packets	1759 packets
Frequency of the packets	47,2 packets/s	29,2 packets/s
Size of the sent data	1,65 MB	1,15 MB

The network traffic caused by the travel of the data packets is significant and is almost equal to 50 percent of the packets transferred between the control computer and the RT targets. The actual value of the packet number sent through the deterministic network is equal to the difference between the higher and lower value in Fig. 5, as the higher values cover the data packets as well as the packets existing in the public network and used to maintain communication between the control computer and RT targets. The dependency between the number of the generated packets and the size of the transferred variable is illustrated in Fig. 6. The number of the packets (already calculated as the difference between higher and lower value as in Fig. 5) increases almost linearly with the doubling number of the elements in the table sent through the network.

The comparison between the traffic in both networks is in Table II. The increase of the traffic in the network after inserting into it the packets also sent through the deterministic network is significant.

The size of the data sent through the deterministic network is about 0,5 MB and requires almost one thousand packets. This sizes almost increases with increase of the number of the number of the table elements transferred between the RT targets.

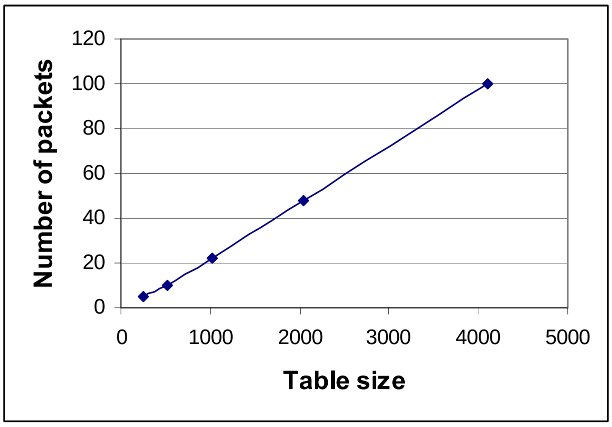

Fig. 6 – Average number of the packets in the deterministic network for the Int32 data table for different number of elements.

4.3. MEMORY AND CPU LOAD ANALYSIS

The last set of experiments was aimed at determining the load of the CPU in the RT targets, depending on the size of the acquired and processed data. The experiments consisted of the series of measurements using Real-Time Monitor [8] for the different sizes of the acquired data. Analysis was performed for both RT targets, so the difference between the machine responsible for the data acquisition (computer number 107) and signal processing (computer number 106) could be determined. The exemplary result for the computer number 106 and acquisition rate of 256 and 2048 samples per second is shown in Fig. 7. The increase of the CPU load after 60 second of monitoring is connected to increasing the number of the acquired samples.

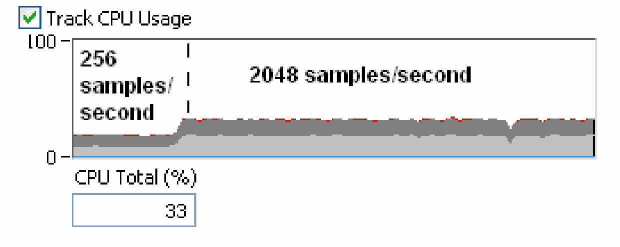

Fig. 7 – Exemplary result of the CPU load monitoring in the signal processing RT target.

The increase of the CPU load for both machines is significant only for larger sampling rate. For example, to detect five percent of the load increase, we needed to increase the size of the acquired samples vector from about ten to twenty thousand. On the other hand, changes below 512 samples per second do not result in visible CPU load increase (which is below one percent). Results for the CPU load change of the RT targets are in Fig. 8. The data acquisition computer performs simpler algorithms as the most of the operation is performed by the DAQ card. The sampling rate was changed through all the range possible for the PCI-6023E card. As can be seen, the power spectrum calculation takes much more of the processor's effort, because for the same number of the processed samples results in almost 15 percent lower load. Both computers have similar processors, so the comparison between the program execution on them is justified. The change of the load again is almost linear. Size of the memory used by both RT targets is similar and equal to 25 percent of the total RAM (which is 256 MB), including the processes of the RTOS.

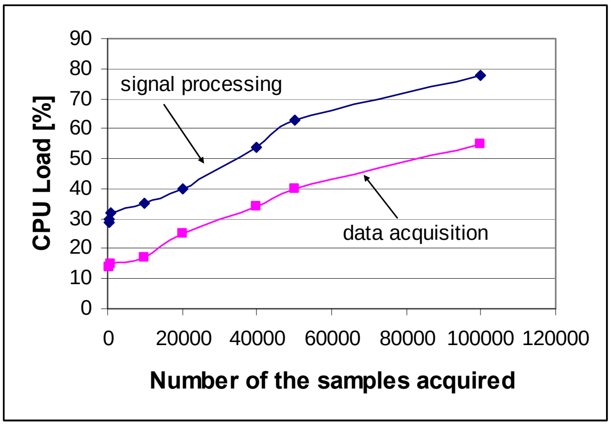

Fig. 8 – CPU loads for both RT targets depending on the size of the processed samples vector.

Analysis of the hardware requirements of the RT DMS using LabVIEW RT module reveals that even old and slow computer can be used for that purpose. The most challenging is the control computer configuration because of the high requirements of the programming environment. For the computers working as RT targets even 450 MHz processors suffice to obtain a reliable system. The minimum operational memory required by the RTOS and a typical application is 128 MB, however 256 MB is a more safe value. Deterministic network can be obtained at small cost.

4.4. ACQUISITION PROCESS ANALYSIS

Additionally to the network measurements, examinations of the DAQ card efficiency were performed. As one of the computers (107) works as the data acquisition module, the sampling speed affects the whole instrument efficiency. The whole time of the data processing is defined by the following formula:

$$t_{ovr} = t_{trans}^{ap} + t_{trans}^{pv} + t_{acq} + t_{proc} \quad \text{s} \qquad (1)$$

and includes:

t_{trans}^{ap} – time of the data transmission between the acquisition and processing computers

t_{trans}^{pv} –time of the data transmission between the processing and control computers

t_{acq} – time of the data acquisition (determined by the DAQ card)

t_{proc} – time of the data processing

The difference between the theoretical and measured acquisition time is presented in Fig. 9. The difference increases for the longer samples vectors, which means that the sampling clock on the DAQ card works slower than the computer system clock. The difference during the hour reaches 2,5 ms. The detailed knowledge about the speed of the acquisition is essential, as it should not be longer than the cycle of the deterministic network. Otherwise, if the samples vectors are updated faster than they are transported to the processing computer, there is the loss of data.

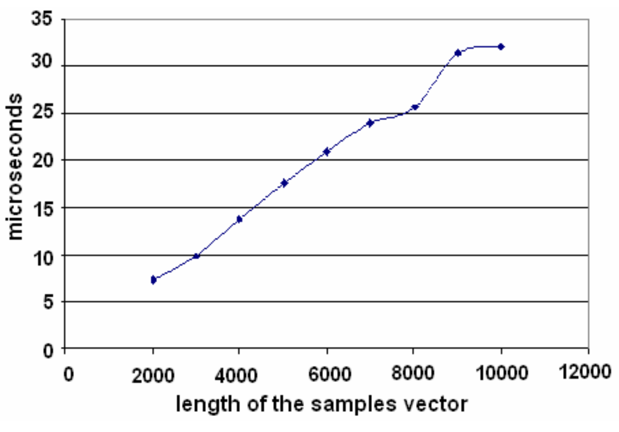

Fig. 9 – Illustration of the difference between the theoretical and measured acquisition time.

5. CONCLUSIONS

The time triggered networks are the next step in the reliable DMS designed using VI technology. Ensuring deterministic data transfer between the modules increases area of possible applications, including education and industrial processes of critical importance. Progress of the RTOS will influence stability of RT targets and the only drawback is the limited group of the compatible hardware (especially network interface cards). The advantages are easy configuration and low overall cost of the DMS (even the slowest existing processors fulfill RT requirement). The designer's task is to focus on the RT-specific programming techniques and ensure that the deterministic data travels between the modules without lockups and stalls.

More advanced RT configuration may require specialized hardware, such as PXI modules or advanced DAQ cards. However, as our research shown, even with basic computer configuration it is possible to obtain efficient distributed system.

6. REFERENCES

[1] M. Auboiroux. MVI Technologies, LabVIEW Real-Time manages an endurance test bench for renault engine components, Available: www.ni.com/realtime

[2] P. Bilski, W. Winiecki. A Low-Cost Real-Time Virtual Spectrum Analyzer, *IEEE IMTC'05 Conf.* (Ottawa, Canada May, 17-19, 2005), pp. 2216-2221.

[3] P. Bilski, W. Winiecki, Virtual Real-Time Instrumentation Using ETS Configuration, *IMEKO XVIII World Congress,* (Rio de Janeiro, Brazil, Sept. 17-22, 2006)

[4] W. Nawrocki, M. Wawrzyniak, LAN-Based Measurement Systems, *IMEKO 2005 Conf.* (Gdynia/Jurata, Poland, September 12-15, 2005), pp. 261-265.

[5] Using Time-Triggered Networks to Communicate Deterministically Over Ethernet with the LabVIEW 8 Real-Time Module, Available: zone.ni.com

[6] Sharing Data using Time-Triggered Shared Variables, Available: zone.ni.com

[7] Using the LabVIEW Shared Variable, Available: zone.ni.com

[8] LabVIEW Measurements Manual, National Instruments Company, 2001.

[9] Ethereal User's Guide, Available: http://www.ethereal.com

Piotr Bilski *(Eng. 2000, Msc. 2001, PhD 2006) was born in 1977 in Olsztyn, Poland. He graduated from Warsaw University of Technology, Institute of Radioelectronics. Currently he is an Assistant Professor and Vice-Dean in Charge of Students in Department of Informatics, Warsaw University of Life Sciences. His main scientific interests are diagnostics of analog systems, design and analysis of virtual instrumentation, application of artificial intelligence and machine learning methods to the environmental sciences and relational databases.*

Wieslaw Winiecki *was born in 1950 in Gdansk, Poland. He received the M.Sc., Ph.D., and D.Sc. degrees from Warsaw University of Technology (WUT), Warsaw, Poland, in 1975, 1986 and 2004, respectively.*

Since 1975, he has been with the Institute of Radioelectronics, WUT, where he is currently a Professor. He is also the Head of the Computer-Aided Measurement Laboratory He is the author or co-author of 4 books and over 150 scientific publications. Since the beginning of his professional career, he has been involved in the activities of the group that is involved with computer-aided measurements concerning the hardware and software for measuring systems. His main field of activity has been focused on virtual instrumentations and distributed measuring systems.

Prof. Wieslaw Winiecki is a member of the Metrology and Instrumentation Committee, Polish Academy of Sciences; and is the President of the Polish Society for Measurement, Automatic Control and Robotics (POLSPAR), and a member of IEEE.