# INVESTIGATION OF COMMUNICATION PARAMETERS BASED ON MONITORING AND STATISCAL MODELLING

*Radi Romansky* *

Technical College – Sofia at Technical University of Sofia
Bulgaria

* Corresponding Author, e-mail: rrom@tu-sofia.bg

**Abstract:** For the effective construction of a network infrastructure, it is recommended to carry out a preliminary investigation of communications and information processing by using suitable methods and tools. In this regard, the article deals with an approach applied in the investigation of network communications in a local network structure based on program monitoring and statistical analysis of the measured data. To specify the research object and the means used, the organization of the network structure is presented, and a brief discussion of sample monitoring programs is made. The organization for the conducted program monitoring is presented and main factors of network communications are defined. Statistical modelling approach is used to investigate selected parameters of communication in the network environment. Some obtained statistical assessments are presented which permit to make conclusions for access and communication organization.

**Key words:** networking, monitoring, statistical modelling, investigation of communication, assessments.

## 1. INTRODUCTION

In the organization of distributed information processing, it is necessary to conduct a preliminary study of the processes, for which different approaches can be applied (empirical, analytical, statistical, modelling, simulation, monitoring, etc.), each of which has its advantages and disadvantages. This necessitates choosing the appropriate method for the set specific goal and ensuring the stability and stability of the designed network [1]. The main tasks that should be solved could be described as following: ✓ investigation of computer data processing [2]; ✓ architectural organization of the network infrastructure; ✓ investigation of the system characteristics of node subnets; ✓ analysis of network traffic and routing parameters

[3], ✓ investigation of remote access to distributed resources in the digital space [4]; ✓ specifying the level of service of user requests, etc. Each of them reflects on the correct maintenance of information resources in the network environment (network directory, files and databases, user mailbox, Webpage for presentation of information, etc.). It should be borne in mind that part of the information resources is confidential, and this requires additional means for their protection, which is especially important in the modern digital era [5]. In certain cases, the user himself becomes a resource in each network environment with the ability to create, modify, search, view and delete information about resources for which they are guaranteed privileges. Depending on the granted rights, the user can access different resources, which determines the corresponding responsibility for their use [6].

A heterogeneous network infrastructure enables parallel and distributed computing in more than one server with the possibility of simultaneous processing in different operating systems, database management system structures, and application servers. A survey on heterogeneous networks is presented in [7], where the discussion is directed to the smart factories and an analysis of network technologies is made. Regardless of the purpose of a built network infrastructure, its investigation is necessary to confirm or reject initial hypotheses of the project. An analysis based on measurement of the efficiency of a heterogeneous network architecture is reported in [8]. In this case the object of investigation is next-generation passive optical network type and the equation adapted from Shannon's information theory is used in this research.

The Local Area Network (LAN) infrastructure offers the relatively limited territorial distribution of hardware resources, which allows easy implementation of immediate program monitoring. The study of local structures, such as tasks for network analysis like clustering, identifying central nodes, and detecting motifs, is well studied according to [9], where an approach for deriving novel geometric features from network structure is presented. In principle, the LAN-infrastructure facilitates the construction of a unified information environment from disparate software systems installed in different nodes, as well as remote access and use of software products and web-based applications.

The purpose of this article is to present the results of a conducted study of network service parameters and supported traffic in the LAN by using program monitoring and subsequent statistical analysis of the registered data. This is presented in the next two section. The developed specialized software environment is discussed in the fourth section and conducted statistical models and carried out analyses with experimental results are presented in the last section.

## 2. MULTI-SEGMENT NETWORK STRUCTURE

The architecture of the multi-segment network structure (Figure 1) creates a generally accessible environment with opportunities for remote use of information resources in connected local segments. The common communication medium is based on twisted pairs of wires and connects existing and self-functioning local area computer networks at the nodes. Additional options allow the inclusion of workstations to existing local networks with the implementation of the necessary connections and installations.

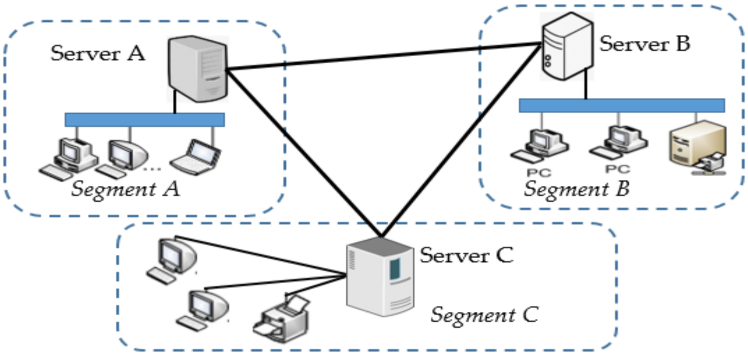



*Figure 1. Architecture of multi-segment network structure.*

The software organization is based mainly on Microsoft products, with the main server providing a platform for adding new modules to provide additional network functions. In the environment, non-dedicated, distributed file servers are configured, whose disks are designated for common use by groups of network users. For each of the individual workstations included in the network, opportunities to access and connect to the network file servers located in the two segments are provided. There is also an individual connection to the Internet for each node separately.

A multi-layered software architecture is shown in Figure 2, which is used for an information distributed environment organization. This heterogeneous language environment supports applications with requests to 2 servers (DB and Internet) and uses interpreted, compiled and scripting languages simultaneously with a different place of execution in the network. Both structured and semi- or non-structured data are processed, using the most modern programming technologies (including XML).

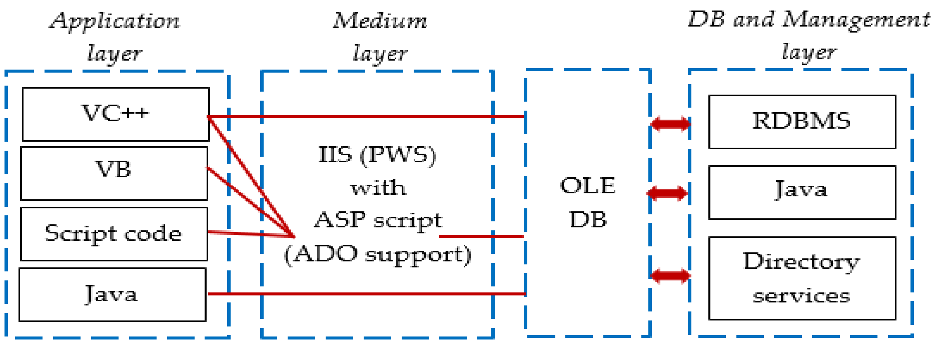



*Figure 2. Multi-layer software architecture*

The primary networking interface is NetBIOS, which allows operation over a TCP/IP environment and enables typical Windows applications to work easily with network resources, remaining independent of the TCP/IP protocol stack. Outside of the context of TCP/IP, NetBIOS applications are in most cases also independent of the transport protocol that the operating system uses. he NetBIOS mechanism for working with files is particularly versatile, for example when logically connecting a user to a domain or calling a remote procedure.

## 3. ORGANIZATION OF NETWORK MONITORING

### 3.1. Network monitoring tools

A brief overview of selected tools for program monitoring in a networked environment is presented below.

✓ *Webserver Stress Tool* is an HTTP client-server application for stress testing the performance in a distributed environment and analysing its characteristics under different load as used in [10]. By using a custom simulation toolkit, the performance of almost any type of server (static pages, JSPs/ASPs, CGIs) can be tested under normal or increased load. The program simulates independent users browsing a set of sites. The aim is to show whether the network exchange conditions required by the user can be provided in critical situations. Each user is simulated via a separate thread with its own session information, which sets URLs independently of other users, with additional settings for running the tests possible.

✓ *Iris* is an open-source software product supporting advanced pipelining technology and presents a new model for assembling a resilient and maintainable Internet metering architecture [11]. In the cited paper, it was used to create a system for distributed IP-level tracing of the routes that packets take over the IPv4 Internet, with the developed Zeph algorithm tracing the route between agents at multiple monitoring points. Network traffic management by Iris is based on the following capabilities: # network activity monitoring; # decryption and recovery of captured data; # session assembly; # filtering of connection attempts; # visualization of network statistics.

✓ *Distinct Network Monitor* is a software for capturing the packets and analysing the network traffic in the monitored network segment, providing a graphical interpretation of the collected statistics [12, 13]. Supports 10/100/1000 Mbps Ethernet and wireless TCP/IP adapters and 4/16 Token Ring cards. It allows diagnosing problems in the monitored network and discovering bottlenecks in it, and the integration of the statistical module Report Builder provides the following possibilities: # graphical interpretation of the collected data; # information about the system that uses the most traffic in the monitored network segment; # information about the connection established and the protocols used, including statistics by application protocols; # information about the most active MAC addresses and about the traffic divided by MAC addresses; # adapter statistics, number of passed packets, information about incoming and outgoing network traffic. Some of the main advantages are precise traffic capture, ability to display a single TCP stream and select fields from the IP header, use of filters, easy detection and marking of bad packets, etc.

✓ *Traffic Analyzer (TA)* is a user monitor program that successfully performs functions similar to LANAlyzer (Novell's continuous network traffic monitoring tool). Allows description of traffic categories through specially prepared templates or through created own rule files. So, it can filter by protocols and their characteristics, services and machines involved in the communication. Rule files are compiled by the protocol analyser when it is started and arranged into internal structures that are convenient to access at the time of traffic analysis. Modification of the rules file is possible when monitoring conditions change. A similar approach was applied in [14] to examine packets and determine them as local or non-local. The paper discusses the development of a hybrid selection strategy to determine the optimal channel that will allow packets to be routed to the receiver along a maximally uncongested path. The proposed strategy allows obtaining useful information about congestion and node load without adding additional load to the network traffic. In this case, TA is used to initially determine the type of traffic and if is local type, a scoring-based strategy is applied to determine the output.

The software implementation of Traffic Analyzer is based on a multi-threaded concept (Figure 3). For each received frame, a new processing thread is started, which ends its execution and releases the resources occupied by it after analysing the frame. This technology simulates a mass service system, which enables parallelism in frame processing and efficient use of system resources.

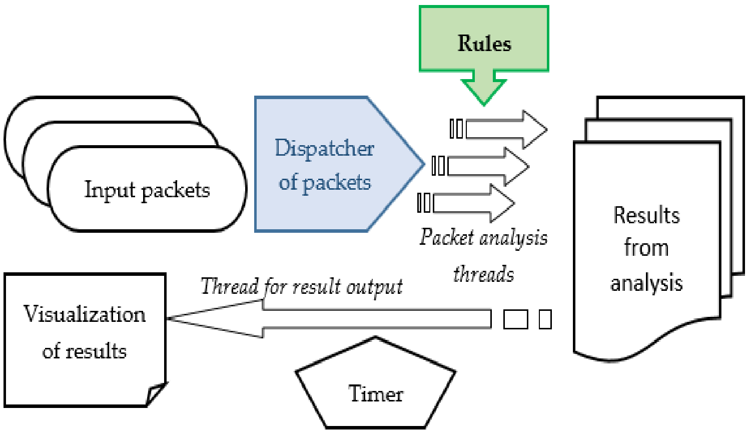



*Figure 3. Concept of the multi-threaded solution*

Traffic registration is done immediately after receiving or sending a frame over the network, processing it through a predefined set of rules and categorizing it. Immediately afterwards, the frame is removed from memory and its contents are no longer available to the monitor (without being lost to its recipient). The monitor is a console application and does not have powerful capabilities for visualizing the results of the observations (the prepared frame report), but it allows exporting to files and easy importing of the data into Microsoft Excel, their presentation as graphs and histograms, or to be processed with methods and functions of mathematical statistics.

Other products that can be mentioned are Microsoft Performance Monitor (a standard tool for monitoring parameters of various objects); *Microsoft Network Monitor* (a typical network analyser that displays the contents of "captured" packets on the network).

### 3.2. Program monitoring organization

The set goal of researching parameters of network communications requires the formulation of the main tasks before the conducted experiments, such as:

✓ Evaluation of the selected network topology regarding network traffic parameters and the impact on the overall network load.

✓ Analysis of the characteristics of different types of network traffic and assessment of their relative contribution to the network load.

✓ Carrying out a statistical analysis of the information service with determination of correlation and regression relationships between basic performance indices and calculation of quantitative assessments for them.

To perform the defined tasks, it is necessary to determine the main observable factors, as well as to select appropriate means of accumulating the necessary data for analysis. The program measurement organization will use some of the monitoring programs presented in the previous section, defining the main observables factors presented below.

- basic number of users;
- the average value of the selection times (‘*average click time’*);
- waiting time for user selection by pressing a mouse button (‘*click time’*);
- bandwidth of the server (‘*server bandwidth’*) and user access (‘*user bandwidth’*);
- number of URLs (‘*count URL’*);
- average request service time (‘*average request time’*);
- supported network traffic (‘*network traffic’*), incl. incoming traffic and outgoing traffic;
- time for interpreting the URL domains using the DNS server of the client machine (‘*time for DNS*’);
- time to establish a connection with the server (‘*time to connect*’);
- time to receive the first byte of the response to a request to the server (‘*time to first byte*’ – TFB);
- average length of packages;
- number of packets for the different network protocols and number of packets for the different IP protocols.

Various sets of experiments were conducted using selected program monitors from those listed above. In addition, a user application Network Analyser has been developed to carry out the statistical processing of the data collected by the monitoring. Program monitors provide a different output format of empirical data files (*.csv, *.doc, *.htm), which requires their coordination for proper import into the developed programming environment, for example as Excel files. In this reason, Iris is not a particularly suitable tool for extracting statistical information from large volumes of traffic, because it uses a buffer to store the packets that have passed through the monitored network segment. Although the size of this buffer can be adjusted and incoming information (packets) can be filtered, there is the real possibility of requiring significant disk space and reducing server performance.

## 4. STATISTICAL MODELLING AND ANALYSIS OF MONITORED DATA

The object of the research presented in this section is to determine the level of dependence of basic parameters such as *'server bandwidth'*, *'click times'*, '*hits*', '*time to first byte*', '*time to connect*' and '*time for DNS*' (defined as dependent factors) of the number of active users ('*users*' – independent controllable factor). To form the initial sample of empirical data, the Webserver Stress Tool was mainly used with access to a selected dynamic site in the network space. The controllable factor is controlled statically at a fixed measurement period of 5 min. The measured maximum values used in the statistical processing are summarized in Table 1.

*Table 1. Empirical data from monitoring*

| 'users' | 2 | 5 | 8 | 10 |
|---|---|---|---|---|
| 'server bandwidth' (kb/s) | 40 | 42 | 40 | 40 |
| 'click time' (ms) | 57 | 150 | 275 | 312 |
| 'hits' (sec) | 34 | 35 | 35 | 34 |
| 'time to connect' (ms) | 28 | 100 | 190 | 210 |
| 'time for DNS' (ms) | 24 | 90 | 180 | 210 |
| 'time to first byte' (ms) | 23 | 75 | 130 | 150 |

Test results from the measurement of values for the '*server bandwidth*' parameter are summarized in Figure 4, and the table confirms the absence of a clear dependence on the '*users*' factor. Observed factor values remain relatively constant in the 40-42 kb/s range, with Network Analyser statistical testing showing a correlation coefficient of –0,238.

The results of the measurements for parameters '*click time*' and '*hits*' are presented in Figure 5, and the Figure 6 presents the result of statistical model investigation of the regression dependency of '*click time*'=*F*('*users*') realized by using application Network Analyzer. With a correlation coefficient of 0,994, the regression model confirms the existence of a strong linear dependence, while for the dependence '*hits*'=*F*('*users*') the calculated estimates (correlation coefficient 0,082 and regression coefficient $b_1$=0,014) reject the existence of such a hypothesis. The results of the analysis confirm a direct dependence of the increase in the execution time of a given request to a web site (‘*click time*’) from the increasing number of users, while the time for a completed HTTP request (‘*hits*’) remains an almost constant value, independent of the number of users.

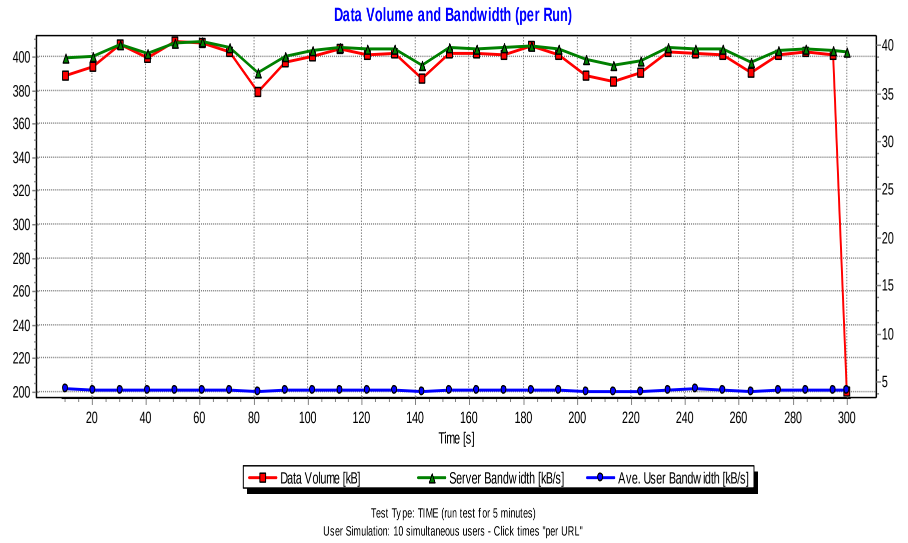



*Figure 4. Test result for ‘bandwidth’ in the time for ‘users’=10*

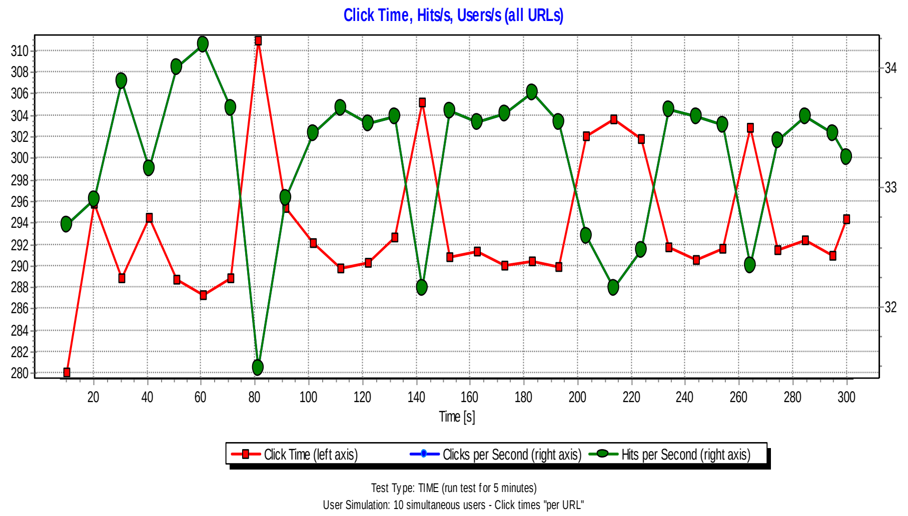



*Figure 5. Text result by monitoring of ‘click time’ and ‘hits’ (‘users’ = 10)*

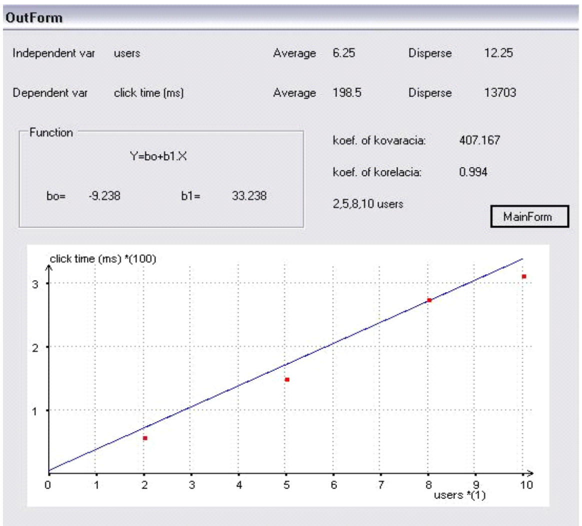
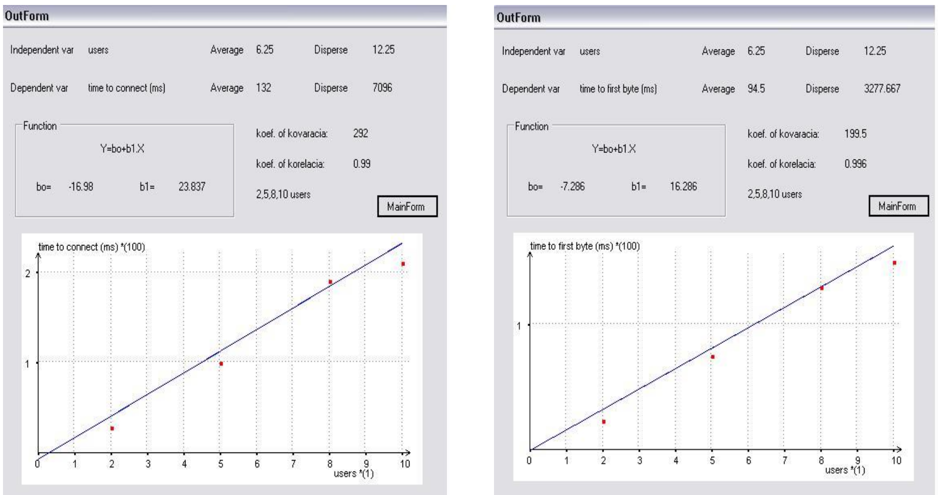



*Figure 6. Statistical modelling of 'click time'=F('users')*

The last three observable variables from Table 1 are sequential actions associated with executing a single HTTP request. The statistical modelling carried out with Network Analysis (Figure 7) confirms for all three factors the hypothesis of a strong correlation dependence on the manageable factor 'users', with the correlation coefficient being 0,99 and a linear increase of the regression line, as for '*time for DNS*'=*F*('*users*') and '*time to first byte*'=*F*('*users*') the average values calculated by the program are almost the same.




*'time to connect'=F('users')* *'time to first byte'=F('users')*

*Figure 7. Results for statistical modelling and analysis*

## 4. CONCLUSION

The article is devoted to the investigation of communication parameters in a LAN structure to determine the efficiency of supported processes. Such a study is recommended by various authors, and in [15] a model framework is proposed for this purpose using a mathematical analytical model to investigate different layers of the abstractions of each network topology using a network simulator. Another approach in the direction of studying the suitability and efficiency of processes is presented in [16], where software testing and the selection of a suitable tool for this in user interface analysis are discussed.

The research presented in this article combines two possible approaches with the aim of combining their advantages and obtaining heterogeneous estimates of the monitored factors. For program monitoring, tools suitable for research in a limited network environment and providing results in a format suitable for further use have been selected. To automate statistical modelling, an application providing estimates of defined dependencies between selected groups of parameters of network communications has been created. The main goal is to form a complete research process for the system organization of a distributed environment and determine the levels of efficiency and relative dependence of parameters.

## REFERENCES

[1] Liu, T.; Bai, G.; Tao, J.; Zhang, Y.A.; Fang, Y.; Xu, B. Modeling and evaluation method for resilience analysis of multi-state networks. *Reliability Engineering & System Safety*, vol. 226,2022, art. 108663 (https://doi.org/10.1016/j.ress.2022.108663).

[2] Romansky, R. Mathematical modelling and study of stochastic parameters of computer data processing. *Mathematics*, vol. 9, no. 18, 2021, art. 2240.

[3] Kravets, O.Ja.; I.V. Atlasov; I.A. Aksenov; A.S. Molchan; O.Yu. Frantsisko; P.A. Rahman. Increasing efficiency of routing in transient modes of computer network operation. *International Journal on Information Technologies and Security*, vol. 13, no. 2, 2021, pp.3-14.

[4] Romansky, R.; Noninska, I. Deterministic model investigation of processes in a heterogeneous e-learning environment. *Int'l Journal of Human Capital and Information Technology Professionals, vol. 13,* no. 1, 2022, pp. 1-16.

[5] Liu, L.; Zhu, H.; Chen, S. Balancing privacy risk and benefit in service selection for multiprovision cloud service composition. *Mathematics*, vol. *10*, no.10, 2022, art. 1675.

[6] Tsaregorodtsev, A.V.; Kravets, O.Ja.; Choporov, O.N.; Zelenina, A.N. Information security risk estimation for cloud infrastructure, *International Journal on Information Technologies and Security*, vol. 10, no. 4, 2018, pp. 67-76.

[7] Xia, D.; Jiang, C.; Wan, J.; Jin, J.; Leung, V.C.; Martínez-García, M. Heterogeneous network access and fusion in smart factory: A survey. *ACM Computing Surveys,* April 2022, 33 p. (https://doi.org/10.1145/3530815).

[8] Girón, A.; Rivera, E.; Gómez, G. Measurement of the spectral efficiency of a heterogeneous network architecture of the NG-PON type for a quasilinear propagation regime. *Entropy, vol. 24*, no. 4, 2022, art. 481 (https://doi.org/10.3390/e24040481).

[9] Cheng, S.; Giesen, J.; Huang, T.; Lucas, P.; Mueller, K. Identifying the sceptics and the undecided through visual cluster analysis of local network geometry. *Visual Informatics*, vol. 6, no. 3, September, 2022, pp.11-22 (https://doi.org/10.1016/j.visinf.2022.07.002)

[10] Nurwarsito, H.; Sejahtera, V.B. Implementation of dynamic web server based on operating system-level virtualization using docker stack. In *Proceedings of 12th IEEE International Conference on Information Technology and Electrical Engineering (ICITEE),* Yogyakarta, Indonesia, 6-8 October 2020, pp. 33-38. DOI: 10.1109/ICITEE49829.2020.9271710.

[11] Gouel, M.; Vermeulen, K.; Mouchet, M.; Rohrer, J.P.; Fourmaux, O.; Friedman, T. Zeph & Iris map the internet: A resilient reinforcement learning approach to distributed IP route tracing. *ACM SIGCOMM Computer Communication Review*, vol. 52, no. 1, 2022, pp.2-9.

[12] *ActiveXperts Network Monitor 2019: Network and Server Monitoring Software*. Monitortools.com, https://www.monitortools.com/software/product/distinct-network-monitor/

[13] *Distinct Network Monitor*. Distinct Corporation, 2022. Available on: https://www.distinct.com/monitor/monitor.htm

[14] Trik, M.; Molk, A.M.N.G.; Ghasemi, F.; Pouryeganeh, P. A hybrid selection strategy based on traffic analysis for improving performance in networks on chip. *Journal of Sensors*, vol. 2022, 2022, art. ID 3112170 (19 p.). https://doi.org/10.1155/2022/3112170

[15] Alsheikhy. A.. Estimating end-to-end delay on a networking environment using developed framework. *International Journal on Information Technologies and Security*, vol. 14, no. 1, 2022, pp. 3-16.

[16] Pelivani, E., Besimi,A., Cico, B. An empirical study of user interface testing tools. *International Journal on Information Technologies and Security*, vol. 14, no. 1, 2022, pp. 37-48.

### *Information about the author:*

**Radi Romansky** is a full professor at Technical University of Sofia, Doctor (Dr) in Computer Engineering and Doctor of Science (D.Sc.) in Informatics and Computer Science; Full member of European Network of Excellence on High Performance and Embedded Architectures and Compilation (HiPEAC). He has over 220 scientific publications and over 25 books. Areas of scientific interests: ICT, informatics, computer architectures, computer modelling, privacy and data protection, etc.

**Manuscript received on 15 April 2022**