Information systems modeling

UDC 004.94 doi: 10.20998/2522-9052.2020.1.03

Yu. Davydovskyi[1], O. Reva[1], O. Malyeyeva[1], V. Kosenko[2]

[1] National Aviation University "Kharkiv Aviation Institute", Kharkiv, Ukraine
[2] State Enterprise “National Design & Research Institute of Aerospace Industries”, Kharkiv, Ukraine

APPLICATION OF THE SLIDING WINDOW MECHANISM IN SIMULATION OF COMPUTER NETWORK LOADING PARAMETERS

Abstract. **The subject** of the study in the article is the data transfer processes in computer networks from the point of view of the network performance indicators, in particular, overload parameters. **The purpose** of the work is to create a simulation model of the behavior of a computer network, taking into account the mechanism of the "sliding window" and to demonstrate its work using a test example in combination with the previously proposed algorithms. The following **tasks** are solved in the article: analysis of the network as a complex multilevel system and isolation of the problems related to the transport layer of the OSI model; consideration of tasks that are solved by the transport layer of the data network; creation of the mechanism of modeling of functions of "sliding window"; demonstration of model work. The following research **methods** are used: basics of system analysis, models of network functioning, simulation modeling method. The following **results** were obtained: Based on the results of previous studies, the article proposes a new model of computer network behavior over a period of time. This model is based on the principles of simulation modeling, which became possible because of the certain fractal properties of incoming traffic, that is one of the initial data for the simulation. In the process of development, the seven-level OSI model was taken as the basic network model and its four lower levels were selected. Ensuring reliable information transmission at all protocol levels is based on the mechanisms of acknowledgement and "sliding window". The proposed model can reduce the amount of user data directly for each of the directions in a proportion directly proportional to their utilization of the specified congested channel. It is proposed to use a correction vector for each direction of information transmission in the model. **Conclusions:** The adequacy of the developed model is confirmed on the basis of practical calculations of the test case. The practical value of this model is the ability to predict bottlenecks when creating a computer network, or vice versa, to point out the redundancy of certain solutions in order to save significant funds in the future for providers and operators of communication services.

Keywords: computer network; modeling; transport level; traffic; sliding window.

Introduction

The growth of computer network traffic shows a tenfold growth trend over the last decade, with the upward trend increasing year by year [1-3]. Communication quality assurance is also a problem that needs to be addressed. Thus, mobile users lose TCP connections when they move or change their IP addresses, or some nodes along the routing path cannot provide the connection. In addition, there may be temporary losses. Such problems also occur in cable networks. These situations have a particularly detrimental effect on the continuous exchange of large volumes of data or the transfer of funds from a bank account. Thus, in the processes of ensuring the good functioning of computer networks, an important role is played by the data flow management subsystem [4], which solves the problem of: congestion management, queue management, transmission recovery after loss of network connectivity, management of lost packets and packets recovery, errors that have occurred.

Analysis of recent studies and publications. Today, there are many studies related to the problems of improving the quality of communication at the transport level of network representation. Thus, Jungang Zhang and Son Tao [5] propose to map logical and physical endpoints in terms of virtual address. This virtual address must consist of three tuples: a socket handle, an IP address, and a TCP port number. To uniquely identify a connection, such components as a random number, socket status information, IP address, and TCP port number are used [6-8]. In addition, it is suggested to add a new layer (Mobile Socket Layer) to the transport layer in order to hide disconnection from the application [9-11].

Data flow management techniques (Tahoe, Vegas, Reno, Newreno, Random Early Detection, etc.) are used to improve TCP performance. However, existing methods are oriented on static or quasi-static conditions of MR operation and do not allow to determine the causes of packet loss [12-14]. To date, the main focus of TCR researchers is to ensure interoperability with protocols in other layers of the OSI (network and channel) model.

Most types of traffic are sensitive to network overload, delays, and packet loss with the data transmitted therein and therefore require implementation of data flow management methods [15-26]. Therefore, efficient data flow management is a topical issue when creating networks of this class.

Highlighting previously unsolved parts of a common problem. The goal of the work

One way to investigate the factors of poor communication is to model its behavior over time. In this case, the analysis of the network should be carried out by different views on its functional properties. One such feature is network load over a period of time. According to a previous study, network traffic is a predictable value over time, so it can be used as a constant value [27]. This opens up the possibility to apply not only agent modeling to computer networks, but also methods of constructing simulation models.

Article [28-30] identified the main characteristics of a computer network that have an impact on data transmission processes, and proposed a method for formalizing metrics that are directly applicable in the modeling process. The following network characteristics were considered:

- bandwidth;
- delays in communication channels;
- quality of service;
- the extent of transmission errors.

The structure and volumes of traffic (transfer requirements) that occur over time have been separately indicated. As a result, the behavior of the computer network was modeled at its lower three levels: physical (network topology), channel (packet volume, and transmission loss) and network (building packet data routes).

However, during the simulation, the question of traffic reallocation in case of congestion of transport nodes or data transmission channels naturally arose. In a real network, this problem is handled by the transport layer using a so-called "floating window».

Therefore, the purpose of this article is to further disclose behavior of the created model of computer network.

The following tasks are solved:

- analysis of the network as a comprehensive multilevel system and highlighting the issues related to the transport layer of the OSI model;
- consideration of tasks that are solved by the transport layer of the data network;
- creation of a mechanism for modeling of the “sliding window” functions;
- demonstration of model work in combination with the algorithms previously proposed in previous works.

Materials and methods

The quality of transmission of different types of traffic on networks also depends on the lower levels of the OSI model (Fig. 1). When considering the transport layer of the OSI model, it should be noted that the transport layer provides data transportation services [28]. In particular, the primary purpose of the transport layer is to address issues such as ensuring reliable transport of data through a unified network. The transport layer is intended for the delivery of data without errors, losses and duplication in the order in which they were transmitted. It does not matter what data is transmitted, from where and where, that is, it determines the mechanism of transmission itself. It divides the data blocks into fragments, the size of which depends on the protocol.

Ensuring reliable transmission of information at all protocol levels is based on acknowledgement and "sliding window" mechanisms (methods for reducing redundancy in the transmission of information – Nagle, Clark, methods for detecting TCP Vegas overload, methods for controlling the size of the window – Slow Start, Binary Increase, and the retransmission timer – Jacobson) [31]. Before considering the floating window mechanism, it is worth describing the principle of operation of the TCP Protocol, namely the need to receive confirmation of the sent bytes of information. This significantly reduces the speed of data transfer, since several connection sessions will be spent on transmitting a single packet of information. For example, before sending, TCP must make sure that the recipient exists, and then perform a three-way-handshake to set up the session. Also, since the TCP Protocol differs from UDP in the reliability of data transmission, each segment during transmission is marked with a sequence number that the recipient sends back to the sender to confirm the sending and to indicate the next segment to be sent.

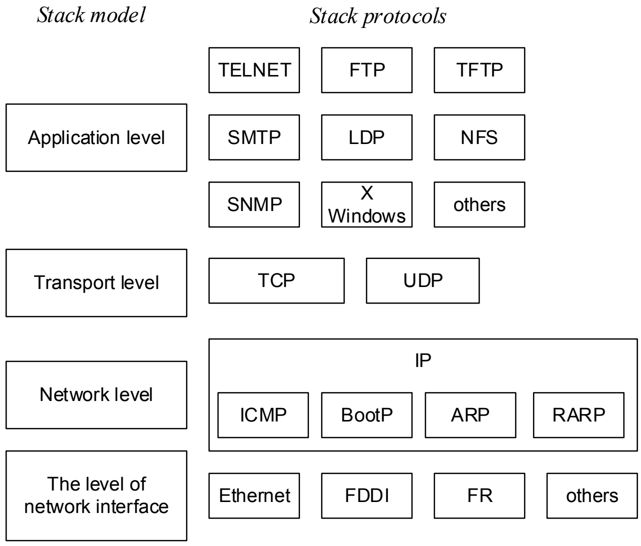

Fig. 1. TCP/IP stack model protocols

TCP packets have the following fields: local sender port and local recipient port (Fig. 2) carrying the contents of program entry points, such as Telnet on one side, and entry point (in this context encapsulation) to the IP layer. In addition, in the TCP/IP stack, the TCP layer is responsible for generating packets from the data stream coming from the application.

Local sender port		Local recipient port
Segment position		
The first expected byte		
Data offset	Flag	Window size
Checksum		Note the urgency of the data
Options		Filler

Fig. 2. The structure of the TCP packet header

A TCP connection is identified by a pair of full addresses of both interacting processes (endpoints). The addresses of each endpoint include IP addresses (network number and computer number) and port number.

The connection is made in the following sequence:

– one of the parties initiates the connection. It sends a request to TCP to open an active port;

– after the TCP port is opened, the initiator process sends a process request that requires a connection;

– TCP from the receiver opens the port for receiving data (passive open) and returns a receipt confirming the receipt of the request;

– in order for the transmission to be possible in both directions, the protocol from the receiver also opens an active port and transmits the request to the opposite party;

– the initiating party opens the receiving port and returns the receipt.

The connection is considered established. Then data is exchanged within this connection.

Within the connection, the correctness of the transfer of each segment must be confirmed by the receipt of the recipient. Acknowledgment is one of the traditional methods of ensuring reliable communication.

The TCP protocol implements a kind of window-based acknowledgement algorithm. The peculiarity of this algorithm is that the window is defined on the set of numbered bytes of unstructured data stream coming from the top level and buffered by the TCP protocol.

The sliding window principle [32] allows using the TCP protocol in networks of different power and reliability. The sliding window mechanism allows changing the number of bytes that can be sent without acknowledgment. Thus, the higher the reliability of the network, the larger the size of such a window and the more information can be sent without confirmation. This allows the user to increase the data rate. When TCP sees that data is being lost, the size of the window decreases, reducing transmission speed by increasing reliability and the number of checks.

The expanded algorithm of the "sliding" window will not be considered in this article, since its implementation in the proposed model is significantly different and preserves only its main stages, namely:

- control over the number of errors;
- window size determination;
- adjust the amount of traffic that will be transmitted.

Suppose that the amount of data to be transmitted is df_c, and that the existing delay on the communication channel is

$$d_c = \frac{1}{C_c - df_c}, \tag{1}$$

where C_c - channel c bandwidth;

When d_c will exceed the maximum Q_k link quality factor, a sliding window model will be enabled, which in the real network would increase the number of bytes that need to send acknowledgments, and in its turn would reduce the amount of user data that can be transmitted in the specified amount of time. The proposed models can reduce the amount of user data directly.

In this way, let the link c be overloaded with O_c. This may mean that the amount of traffic to be transmitted by this channel will be reduced to the next:

$$df_c^* = (d_c - O_c) f_c. \tag{2}$$

Considering that information of different directions can be transmitted by one channel [16], the next step of the model will be to determine all transmission directions that trigger the specified channel.

And also one of the key features of the OSI model is that the levels do not interfere with each other's work, but extend their functionality. This leads to the fact that at the transport level there may not be information about the route structure that was built at the previous network level. Therefore, the correction effect should not extend to the specified route, but to the entire direction of data transmission.

Thereafter, the model can reduce the amount of information for each of the directions in a size that is directly proportional to their utilization of the specified overloaded channel.

Let us denote the total set of possible transmission directions R. Then the amount of data that should be deferred on the direction $r \in R$:

$$df_r^* = \left(d_c - O_c\right) f_r, \tag{3}$$

where d_c – the amount of delay at the channel c, exceeding the maximum allowed coefficient,

O_c – the amount by which the allowed number of transmitted data is exceeded. It is calculated based on d_c and the constant M, which must be specified before starting the simulation,

f_r – the amount of data that must be passed on a specific iteration of the model in the direction of r.

After that, the iteration should be recalculated with the changed indicators df_r.

However, the calculations at the test bench showed the inadequacy of such a model, which is caused by the possibility of multiple congestion of the network during one iteration. This results in multiple corrections of the same transmission directions r from the subset $R' \in R$, which in its turn leads to an unreasonably large reduction in traffic df_r.

To solve this problem it is suggested to use a correction vector for each direction $r \in R$. Thus for each direction r the maximum amount of overload is calculated, which is equal:

$$v_r = \max_i (df_{c_1}^*, \ldots, df_{c_i}^*, \ldots, df_{c_n}^*), \tag{4}$$

where $df_{c_i}^*$ – volume of all overloads, at all communication channels that are used in the direction r, $i = \overline{1, n}$.

Therefore, in a situation where the communication direction r repeatedly causes overloads, the number of its adjustments will always be equal to one, and the value of the adjustment is its maximum network overload.

To determine the overload in the model, we will use an additional calculation cycle using the following formula:

$$O_c = \frac{df_c - MC_c}{df_c}. \qquad (5)$$

In the proposed model, an additional round should be performed in case of network congestion, in particular when channel delays are negative or greater than one.

The results of the simulation on a test sample

Here is an example of modeling the behavior of a computer network, where the first step is to describe the network topology, that is, the geographical location of the nodes and communication channels. Consider a simplified version of the network topology shown in Fig. 3 [28].

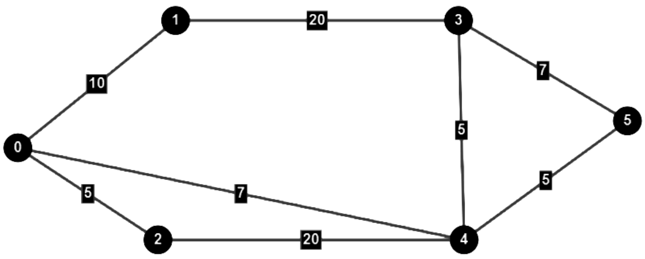

Fig. 3. Test network topology

The next step is to select the input for modeling. We choose the structure of the input data according to the average load on computer networks in the world [27]. The following is the distribution of orders for data transmission by destination. The destinations for data transfer will share this load and can be presented as a table of distribution of applications for data transmission (Table 1).

Two routes will be constructed at the beginning of the simulation using the Dijkstra algorithm. The next step is to saturate these edges with the data transfer requirements of Table 1. The state of the network after saturation is shown in Fig. 4.

Table 2 shows the distribution of data transfer orders after the second iteration.

Table 1 – **Distribution of data transfer orders**

Directions of data transfer	**Modeling time parameters**				
	10 am	2 pm	4 pm	8 pm	10 pm
1	5	7	3	7	5
2	1	3	3	1	

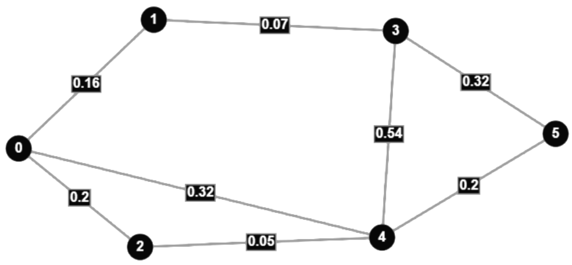

Fig. 4. A network full of data transfer requirements

Table 2 – **Distribution of data transfer orders after the second iteration**

Directions of data transfer	**Modeling time parameters**				
	10 am	2 pm	4 pm	8 pm	10 pm
1	5	**7.75**	**4,1625**	7	5
2	1	**3.15**	**3,4725**	1	

Add in the condition that the amount of data to be transmitted at 4 pm will increase significantly. We obtain the distribution shown in Table 3.

Table 3 – **Distribution of data orders with increased metrics**

Directions of data transfer	**Modeling time parameters**				
	10 am	2 pm	4 pm	8 pm	10 pm
1	5	7.75	**20**	7	5
2	1	3.15	**15**	1	

Using Dijkstra's algorithm, two more routes were found for both transmission directions. The transmission volume for routes of the first direction will be equal to 6.6, and for routes of the second 10 and 5, respectively, their priority (Fig. 5).

After recalculating the delays, we get the following structure of the transport network (Fig. 6). As can be seen from the figure, the overload occurred simultaneously on six communication channels and the requirements need to be corrected.

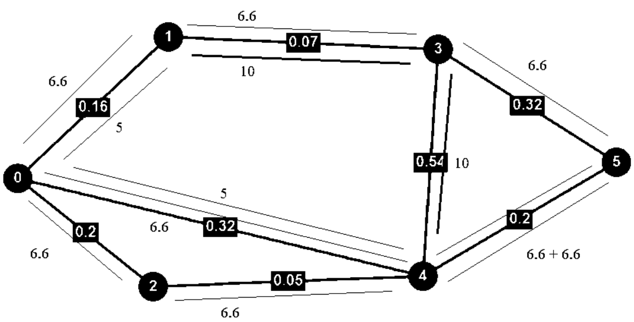

Fig. 5. Data routes on the third iteration after saturation with requirements

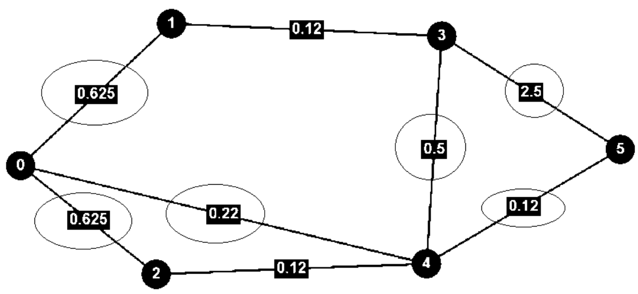

Fig. 6. Delays in communication channels during overload

The model now needs to reallocate some traffic based on congestion.

First of all, you need to detect the congestion of each of the communication channels by the formula (5) (in this example, the load factor M = 0.85):

$$O_{0-1} = \frac{11.6 - 0.85 * 10}{11.6}.$$

In accordance

$$O_{0\text{-}2} = 0.36;\ O_{0\text{-}4} = 0.49;\ O_{3\text{-}4} = 0.58;$$

$$O_{3\text{-}5} = 0.10;\ O_{4\text{-}5} = 0.68.$$

As follows, we define the correction vector for both transmission directions. For this purpose, we will distribute the congestion on the communication channels according to the directions of data transmission:

$$v_1 = \max\{0.27, 0.36, 0.49, 0.58, 0.1, 0.68\} = 0.68\ ;$$

$$v_2 = \max\{0.27, 0.49, 0.58\} = 0.58\ .$$

Now, the model is moving the rest of the data to the next iteration in two stages, first 15% of the amount of data will be transferred automatically due to transmission losses, and secondly, the amount of data that causes the overload will be transferred. Thus, we obtain Table 4 of the distribution of orders.

Table 4 – **Distribution of orders after reduction of network load**

Directions of data transfer	**Modeling time parameters**				
	10 am	2 pm	4 pm	8 pm	10 pm
1	5	7.75	**6.4**	**7 + 3 + 13.6**	5
2	1	3.15	**6.3**	**1 + 2.25 + 8.7**	

The last step in this iteration is to recalculate the modified load model. The recalculation of the delays gives the result shown in Figure 7. It is obvious that the congestion has been corrected, but the greatest value of this model is that the bottleneck is visible after correction - the bottlenecks of the system that need to be changed to increase the utilization rate of the network resources. These bottlenecks are circled at Fig. 7. Channels 3-4 and 4-5 are dangerously close to the maximum load factor *M*, which is equal to 0.85.

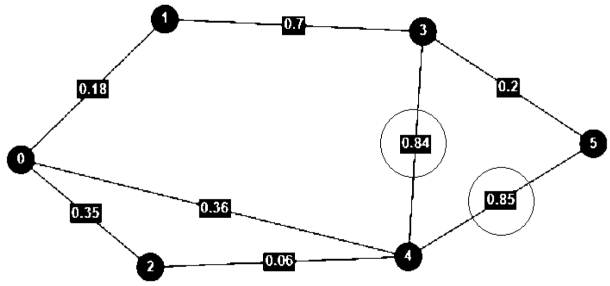

Fig. 7. Delays in communication channels after load adjustment

Conclusions

Based on the results of previous studies, the article proposes a new model of computer network behavior over a period of time. This model is based on the principles of simulation, which is made possible by certain fractal properties of incoming traffic, which is one of the initial data for the simulation.

In the course of development, the OSI seven-tier model was taken as the basic network model and its four lower levels were selected. Providing reliable information transmission at all protocol levels is based on acknowledgment and sliding window mechanisms. The proposed models can reduce the amount of user data directly for each of the directions directly in proportion to their utilization of the specified congested channel.

The model proposes to use a correction vector for each direction of information transmission.

The adequacy of the developed model is confirmed on the basis of practical calculations of the test case. The practical value of this model is the ability to predict bottlenecks when creating a computer network, or vice versa to point out the redundancy of certain solutions in order to save considerable money on providers and service providers in the long run.

References

1. (2019), *Internet World Stats: Usage and Population Statistics*, available at: https://www.internetworldstats.com/stats.htm
2. (2019), Cisco VNI., "Cisco Visual Networking Index: Forecast and Trends, 2017–2022", available at: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.html
3. Bihun, N., Shyshatskyi, A., Bondar, O., Bogrieiev, S., Sova, O., Trotsko, O. and Nalapko. O. (2019), Analysis of the peculiarities of the communication organization in NATO countries, *Advanced Information Systems*, Vol. 3, No. 4, pp. 39-44, DOI: https://doi.org/10.20998/2522-9052.2019.4.05
4. Davydovskyi, Y., Reva, A. and Malyeyeva, O. (2018), "Method of modelling the parameters of data communication network for its upgrading", *Innovative technologies and scientific solutions for industries*, No. 4. pp. 15-22, DOI: https://doi.org/10.30837/2522-9818.2018.6.015
5. Zhang, Y. and Dao S. (1995), "A Persistent Connection Model for Mobile and Distributed Systems", *Proc. Int. Conf. on Computer Communications and Networks*, p. 300.
6. Kuchuk, G.A., Gakhov, A.A. and Pashnev, R.P. (2006), *Infocommunication Resource Management*, Fizmatlit, Moscow, 218 p.
7. Ekwall, R., Urban, P. and Schiper, A. (2003), "Robust TCP Connections for Fault Tolerant Computing", *Proc. Int. Conf. on Parallel and Distributed Systems*, pp. 501-508.
8. Babenko, V.G., Shadkhin, V.Yu. and Shevchenko O.O. (2010), "Retrieving the principles of organizing the transfer of data in TCP/IP measures", *Bulletin of the Cherkasy Sovereign Technological University,* No. 3, pp. 3-5.
9. Manner, J. and Kojo M. (2004) Mobility Related Terminology. *Network Working Group.* RFC 3753.
10. Eddy, W-E. and Ishac, J. (2005), "Location Management in a Transport Layer", *Mobility Architecture*, NASA/TM-2005-213844.

11. Krasilnikov, S.R., Kravchuk, S.O. and Levchuk, A.R. (2015), "Cross-level optimization for increasing the productivity of the transport protocol in wireless networks", *Proceedings of the Military Institute of Taras Shevchenko National University of Kyiv,* No. 49. pp. 61-66.
12. Yang, P., Luo, W., Xu, L., Deogun, J. and Lu, Y. (2011), "TCP Congestion Avoidance Algorithm Identification", *Proc. of IEEE ICDCS*, available at: http://cse.unl.edu/ xu/paper/peng ICDCS 2011.pdf
13. Sadouni, S., Benslama, M. and Beylot, A. (2016), "Sctp-manet new extension of sctp protocol for the optimization of manet performances", *International Journal of Wireless & Mobile Networks (IJWMN),* Vol. 8, No. 5, pp. 53-66.
14. Zhuk, P.V. (2011), "Analysis of Data Flow Management Methods in Mobile Radio Networks (MANET)", *Proceedings of the Military Institute of Taras Shevchenko National University of Kyiv*, No. 33, pp.145-155.
15. Hanabali, A. (2005), "A Survey of TCP over Ad Hoc Networks", *IEEE Communications Surveys&Tutorials*, No. 7(3), pp. 22-36.
16. Larsen, E. (2012), *TCP in MANETs – challenges and Solutions*, Norwegian Defence Research Establishment (FFI), 27 September 2012, 57 p.
17. Mozhaiev, M., Kuchuk, N. and Usatenko, M. (2019), "The method of jitter determining in the telecommunication network of a computer system on a special software platform", *Innovative technologies and scientific solutions for industries*, Vol. 4 (10), pp. 134-140, DOI: https://doi.org/10.30837/2522-9818.2019.10.134
18. Kuchuk, N., Mohammed, A. S., Shyshatskyi, A. and Nalapko, O. (2019), "The method of improving the efficiency of routes selection in networks of connection with the possibility of self-organization", *International Journal of Advanced Trends in Computer Science and Engineering*, No. 8(1), P. 1–6, DOI: https://doi.org/10.30534/ijatcse/2019/0181.22019.
19. Kuchuk, G., Kharchenko, V., Kovalenko, A., Ruchkov, E. (2016), "Approaches to selection of combinatorial algorithm for optimization in network traffic control of safety-critical systems", *East-West Design & Test Symposium (EWDTS)*, P. 1–6, DOI: https://doi.org/10.1109/EWDTS.2016.7807655
20. Sviridov, A., Kovalenko, A. and Kuchuk, H. (2018), "The pass-through capacity redevelopment method of net critical section based on improvement ON/OFF models of traffic", *Advanced Information Systems*, Vol. 2, No. 2, pp. 139–144, DOI: https://doi.org/10.20998/2522-9052.2018.2.24
21. Svyrydov, A., Kuchuk, H., Tsiapa, O. (2018), "Improving efficienty of image recognition process: Approach and case study", *Proceedings of 2018 IEEE 9th International Conference on Dependable Systems*, Services and Technologies, DESSERT 2018, pp. 593-597, DOI: http://dx.doi.org/10.1109/DESSERT.2018.8409201
22. Kuchuk G.A., Mohammad A.S. and Kovalenko, A.A. (2011), "The parametric data transmission control method for modifying the transport protocols of wireless networks ", *Information Processing Systems*, No. 8 (98), pp. 211-218.
23. Ruban, I., Kuchuk, H. and Kovalenko A. (2017), "Redistribution of base stations load in mobile communication networks", *Innovative technologies and scientific solutions for industries*, No 1 (1), P. 75–81, doi : https://doi.org/10.30837/2522-9818.2017.1.075
24. Kuchuk, G., Nechausov, S. and Kharchenko, V. (2015), "Two-stage optimization of resource allocation for hybrid cloud data store", *International Conference on Information and Digital Technologies*, *Zilina*, pp. 266-271, DOI: http://dx.doi.org/10.1109/DT.2015.7222982
25. Donets, V., Kuchuk, N. and Shmatkov, S. (2018), "Development of software of e-learning information system synthesis modeling process", *Advanced Information Systems*, Vol. 2, No 2, pp. 117–121, DOI: https://doi.org/10.20998/2522-9052.2018.2.20
26. Kuchuk, N., Mozhaiev, O., Mozhaiev; M. and Kuchuk, H. (2017), "Method for calculating of R-learning traffic peakedness", *4th International Scientific-Practical Conference Problems of Infocommunications Science and Technology, PIC S and T 2017*, pp. 359–362. URL : http://dx.doi.org/10.1109/INFOCOMMST.2017.8246416
27. Malyeyeva O., Davydovskyi Y. and Kosenko V. (2019), "Statistical analysis of data on the traffic intensity of Internet networks for the different periods of time", *Second International Workshop on Computer Modeling and Intelligent Systems (CMIS-2019)*, pp. 897-910.
28. Davydovskyi, Y., Reva, O., Artiukh, O. and Kosenko, V. (2019), "Simulation of computer network load parameters over a given period os time", *Innovative technologies and scientific solutions for industries*, No. 3 (9), pp. 72-80, DOI: https://doi.org/10.30837/2522-9818.2019.9.072
29. Kovalenko, A. and Kuchuk H. (2018), "Methods for synthesis of informational and technical structures of critical application object's control system", *Advanced Information Systems*, Vol. 2, No. 1, pp. 22–27, DOI: https://doi.org/10.20998/2522-9052.2018.1.04
30. Mozhaev, O., Kuchuk, H., Kuchuk, N., Mozhaev, M., Lohvynenco, M. (2017), "Multiservise network security metric", *IEEE Advanced information and communication technologies-2017*, Proc. of the 2th Int. Conf. Lviv, 2017, pp. 133–136, DOI: https://doi.org/10.1109/AIACT.2017.8020083
31. Svyrydov, A. S. (2018), Method of growth of data transmission reliability time of use of TCP freeze protocol by increase transmission capacity, *Control, Navigation and Communication Systems,* PNTU, Poltava, Vol. 3, No. 49, pp. 135-138, DOI: https://doi.org/10.26906/SUNZ.2018.3.135
32. (2019), *CiscoTips*, available at : http://ciscotips.ru/ccna1-3

Received (надійшла) 12.12.2019
Accepted for publication (прийнята до друку) 22.01.2020

ABOUT THE AUTHORS / ВІДОМОСТІ ПРО АВТОРІВ

Давидовський Юрій Костянтинович – аспірант кафедри комп'ютерних наук та інформаційних технологій, Національний аерокосмічний університет імені М.Є. Жуковського «ХАІ», Харків, Україна;
Yurii Davydovskyi – graduate student of the Department of Computer science and information technologies, National Aerospace University "KhAI", Kharkiv, Ukraine;
e-mail: davidovskyi2350@gmail.com; ORCID ID: https://orcid.org/0000-0003-2813-4169.

Рева Олександр Анатолійович – кандидат технічних наук, доцент кафедри комп'ютерних наук та інформаційних технологій, Національний аерокосмічний університет імені М.Є. Жуковського «ХАІ», Харків, Україна;
Oleksandr Reva – Candidate of Technical Sciences, Associate Professor of the Department of Computer science and information technologies, National Aerospace University "KhAI", Kharkiv, Ukraine;
e-mail: o.reva@khai.edu; ORCID ID: https://orcid.org/0000-0003-1933-1064.

Малєєва Ольга Володимирівна – доктор технічних наук, професор, професор кафедри комп'ютерних наук та інформаційних технологій, Національний аерокосмічний університет імені М.Є. Жуковського «ХАІ», Харків, Україна;
Olga Malyeyeva – Doctor of Technical Sciences, Professor, Professor of the Department of Computer science and information technologies, National Aerospace University "KhAI", Kharkiv, Ukraine;
e-mail: o.maleyeva@khai.edu; ORCID ID: https://orcid.org/0000-0002-9336-4182.

Косенко Віктор Васильович – доктор технічних наук, професор, помічник директора з наукової роботи, ДП "Південний державний проектно-конструкторський та науково-дослідний інститут авіаційної промисловості", Харків, Україна;
Viktor Kosenko – Doctor of Technical Sciences, Professor, Assistant Director for Research, State Enterprise "National Design & Research Institute of Aerospace Industries", Kharkiv, Ukraine;
e-mail: kosv.v@ukr.ua; ORCID ID: https://orcid.org/0000-0002-4905-8508.

Застосування механізму ковзаючого вікна при моделюванні параметрів навантаження комп'ютерної мережі

Ю. К. Давидовський, О. А. Рева, О. В. Малєєва, В. В. Косенко

Анотація. Предметом дослідження в статті є процеси передачі даних в комп'ютерних мережах з погляду на показники якості роботи мережі, зокрема, параметри перевантаження. **Мета роботи –** створення моделі імітації поведінки комп'ютерної мережі з урахуванням механізму «ковзаючого вікна» та демонстрація її роботи за допомогою тестового прикладу в комплексі з запропонованими раніше алгоритмами. В статті вирішуються наступні **завдання**: аналіз мережі як комплексної багаторівневої системи та виділення проблематики, яка стосується транспортного рівня моделі OSI; розгляд задач, які вирішуються транспортним рівнем мережі передачі даних; створення механізму моделювання функцій «ковзаючого вікна»; демонстрація роботи моделі. Використовуються такі **методи** дослідження: основи системного аналізу, моделі функціонування мережі, метод імітаційного моделювання. Отримано наступні **результати**: З урахуванням результатів попередніх досліджень в статті запропоновано нову модель поведінки комп'ютерної мережі на протязі певного проміжку часу. Вказана модель базується на принципах імітаційного моделювання, що стало можливим завдяки певним фрактальним властивостям вхідного трафіку, який є одним із початкових даних для моделювання. В процесі розробки в якості базової моделі мережі була взята семирівнева модель OSI та обрані чотири нижні її рівні. Забезпечення надійної передачі інформації на всіх рівнях протоколів базується на механізмах квитування та „ковзаючого вікна" У пропонованій моделі можна зменшувати обсяг даних користувача безпосередньо для кожного з напрямків у розмірі, що прямо пропорційний їхній утилізації вказаного перевантаженого каналу. В моделі запропоновано використовувати вектор корекції для кожного напрямку передачі інформації. **Висновки.** Адекватність розробленої моделі підтверджена на базі практичних розрахунків тестового прикладу. Практичною цінністю даної моделі є можливість спрогнозувати «вузькі місця» (bottlenecks) під час створення комп'ютерної мережі, або навпаки - вказати на надмірність певних рішень, щоб у перспективі зекономити значні кошти провайдерів та операторів послуг зв'язку.

Ключові слова: комп'ютерна мережа; моделювання; транспортний рівень; трафік; ковзаюче вікно.

Использование механизма скользящего окна при моделировании параметров нагрузки компьютерной сети

Ю. К. Давыдовский, А. А. Рева, О. В. Малеева, В. В. Косенко

Аннотация. Предметом исследования в статье являются процессы передачи данных в компьютерных сетях с точки зрения показателей качества работы сети, в частности, параметров перегрузки. **Цель** работы - создание модели имитации поведения компьютерной сети с учетом механизма «скользящего окна» и демонстрация ее работы с помощью тестового примера в комплексе с предложенными ранее методами. В статье решаются следующие **задачи:** анализ сети как комплексной многоуровневой системы и выделение проблематики, касающейся транспортного уровня модели OSI; рассмотрение задач, которые решаются транспортным уровнем сети передачи данных; создание механизма моделирования функций «скользящего окна»; демонстрация работы модели. Используются следующие **методы** исследования: основы системного анализа, модели функционирования сети, метод имитационного моделирования. Получены следующие **результаты**: С учетом результатов предыдущих исследований в статье предложена новая модель поведения компьютерной сети в течение определенного промежутка времени. Указанная модель базируется на принципах имитационного моделирования, что стало возможным благодаря определенным фрактальным свойствам входящего трафика, который является одним из исходных данных для моделирования. В процессе разработки в качестве базовой модели сети взята семиуровневая модель OSI и выбраны четыре ее нижних уровня. Обеспечение надежной передачи информации на всех уровнях протоколов базируется на механизмах квитирования и "скользящего окна". В предлагаемой модели можно уменьшать объем данных пользователя непосредственно для каждого из направлений в размере, прямо пропорциональном их утилизации указанного перегруженного канала. В модели предложено использовать вектор коррекции для каждого направления передачи информации. **Выводы:** Адекватность разработанной модели подтверждена в дальнейшем на базе практических расчетов тестового примера. Практической ценностью данной модели является возможность спрогнозировать «узкие места» (bottlenecks) при создании компьютерной сети, или наоборот - указать на избыточность определенных решений, чтобы в перспективе сэкономить значительные средства провайдеров и операторов услуг связи.

Ключевые слова: компьютерная сеть; моделирование; транспортный уровень; трафик; скользящее окно.