Information systems modeling

UDC 004.492.3+681.518 **doi:** https://doi.org/10.20998/2522-9052.2022.4.02

Hanna Drieieva, Yelyzaveta Meleshko, Oleksandr Drieiev, Volodymyr Mikhav

Central Ukrainian National Technical University, Kropyvnytskyi, Ukraine

COMPUTER SIMULATION MODEL OF A COMPUTER NETWORK WITH FRACTAL TRAFFIC FOR TESTING ROUTING ALGORITHMS

Анотація. **The object** of research in this article is simulation modeling of a computer network and the process of traffic routing. The relevance of the research is due to the importance of ensuring the quality of service in computer networks, in particular, by necessity reducing the number of lost IP-packets at high traffic intensity values. Determining the routing of traffic packets is a complex process and is based on various indicators or combinations of indicators. If the routing process takes place in a dynamic mode, then the complexity of the route calculation increases, in this case, one of the tools for research and comparison of different routing algorithms can be a computer simulation model of a computer network. **The goal** of the work is the development and research of a computer simulation model of a computer network for testing traffic routing algorithms. **The tasks:** to develop computer simulation model of a computer network to generate the network structure and simulate the traffic distribution process with the ability to test different routing algorithms. **Research methods:** theory of computer networks, theory of fractal analysis, object-oriented programming, theory of algorithms and data structures, theory of complex networks, theory of Markov processes. **Conclusions.** The paper investigated the basic principles of traffic routing in computer networks. A simulation model of a computer network for testing traffic routing algorithms has been developed. A method based on the theory of complex networks has been developed to generate the structure of a computer network. Theory of fractal analysis and Markov processes are used for traffic generation. A series of experiments was conducted on a developed model to determine how different fractal dimensions of traffic at high traffic intensity values affect the number of lost packets, and therefore the quality of service. Analyzing the results of the experiment, the following conclusions can be drawn: the least number of lost packets occurs when the process is random or has weak trends. The fewest lost packets were at fractal dimension 1.5, i.e., when the process is completely random, there were also few lost packets at fractal dimensions close to this; persistent and anti-persistent processes (those with memory) cause more packet loss for the same traffic intensity and maximum number of packets sent from one device per unit of time. Moreover, anti-persistent processes cause significantly more losses than persistent ones. Thus, when performing traffic routing and finding optimal paths for sending IP-packets, it can be useful to determine and take into account the fractal dimension of traffic at the entrance of each router and use it when calculating metrics to determine the best routes.

Keywords: computer simulation model; computer network; routing; fractal dimension; complex networks; network traffic; quality of service.

Introduction

In this work, a computer simulation model of a computer network was created based on the theory of complex networks, Markov processes and the theory of fractal analysis. This computer simulation model allows you to generate the structure of a computer network and simulate the movement of traffic between network devices for the purpose of testing routing algorithms.

Complex networks are stochastic networks with a non-trivial topology, which differ from classical stochastic networks by the presence of a small number of nodes with a large number of connections [1, 2]. Most real-world networks are complex, for example, computer, transportation, and social networks are complex.

Complex networks have the following main properties [1-4]: scalelessness, small network diameter, high clustering coefficient and high transitivity coefficient, giant connected component (that is, more than 80% of the nodes are connected to each other; in our computer network model, full connectivity is necessary), there are hierarchical connections, there are complex cluster formations (cliques, clans, etc.), assortativity (the emergence of connections between vertices that are somehow similar to each other, in the narrow sense – the emergence of connections between vertices with a large number of connections).

Network traffic has fractal properties and can be modeled using fractal dimension and Markov processes [5]. Therefore, the generation of traffic to reproduce its fractal properties is based on the theory of Markov processes, which is often used to model the traffic of various mass service systems [6-12].

Routing is the process of determining the optimal route for the passage of information in computer systems [13, 14]. Each router makes a decision about the direction of forwarding packets based on the routing table. A routing table contains a set of rules. Each rule in the set describes the gateway or interface used by the router to access a particular network. Routes can be set administratively (static routes) or calculated using routing algorithms, based on information about the topology and state of the network obtained using routing protocols (dynamic routes).

A routing protocol is a network protocol used by routers to determine possible data routing routes in a complex large computer network [13, 14]. Routing protocols are divided into two types depending on the types of algorithms on which they are based [13, 14]: Distance Vector Algorithm (DVA) and Link State Algorithm (LSA). Examples of Distance Vector

Algorithms: RIP – Routing Information Protocol; IGRP - Interior Gateway Routing Protocol (licensed protocol of Cisco Systems); BGP - Border Gateway Protocol; AODV. Examples of Link State Algorithms: IS-IS – Intermediate System to Intermediate System (OSI stack); OSPF - Open Shortest Path First; NLSP - NetWare Link-Services Protocol (Novell stack); HSRP and CARP are gateway reservation protocols in Ethernet networks. Distance Vector Algorithms (also known as Belman-Ford algorithms) require each router to forward all or part of its routing table, but only to its neighbors. Distance vector algorithms work well only in small networks. In large networks, they clog communication lines with intensive service periodic traffic. In large networks, Link State Algorithms are used. They send only small adjustments to all network nodes and do not clog communication channels with service messages. Metrics used in routing algorithms to find the shortest IP-packet forwarding path: route length, reliability, delay, bandwidth, load, communication cost.

The larger and more complex the computer network, the more requirements are placed on routing algorithms in order for them to provide the required quality of service. When researching, improving and developing routing algorithms, their testing is important. To test routing algorithms, a computer network of a given complexity or a computer simulation model must be available. Both options have their pros and cons, but it can be safely noted that at the initial stages of development, a high-quality computer simulation model will significantly speed up the development process, and final experiments before practical implementation should be carried out already on real computer networks.

The purpose of this work is the development and research of a computer simulation model of a computer network with fractal traffic for testing routing algorithms.

Main material

The computer network in a developed model is represented by a fully connected undirected weighted graph, in which routers are nodes, and network connections between them are edges. The weight of edges is the inverse of the bandwidth of the communication channel. Nodes contain queues in which received packets are placed before determining the route of its dispatch and sending it to the next node. Time in a model is represented by discrete iterations. Routing is based on an algorithm that must be tested on a model.

The developed model provides two modes of operation:

1) at each iteration, a random number of traffic packets with random devices are generated by senders and receivers and their routing is carried out;

2) on the first iteration, a certain number of traffic packets with random devices by senders and receivers are generated once, on all subsequent iterations only their routing is carried out.

Stages of a developed computer simulation model of a computer network:

Stage 1. Generation of the computer network structure (Fig. 1) based on the Barabási-Albert model [15].

Stage 2. Checking whether the obtained network graph is fully connected. If the generated graph is not fully connected – adding edges between separated graph parts.

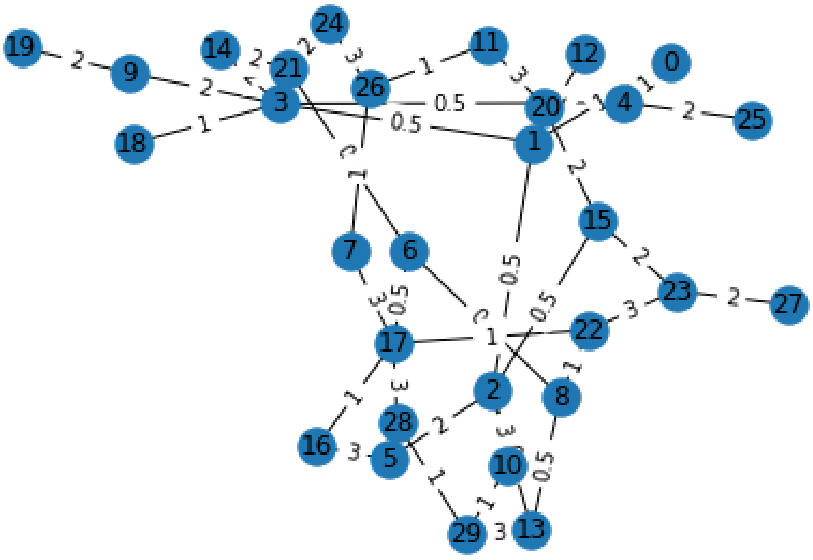


Fig. 1. An example of a structure of a computer network generated in a developed model, graph visualization is performed using the networkx library

Stage 3. Assigning weights to the edges, which depends on which vertices they connect – the more ties the nodes connecting the edge have, the lower the weight of this edge (and, accordingly, the bigger the bandwidth of the corresponding communication channel).

Stage 4. Generation of traffic packets for sending. A random number of packets with random destinations are sent to each node with some probability. The device that received the packets puts them in its internal queue. Traffic is generated with fractal properties [5]. Traffic generation is based on the theory of Markov processes, which is often used to model the traffic of various mass service systems [6-12].

Stage 5. Testing of routing algorithms. Some routing algorithm is selected for testing. Traffic packets queued at network nodes are served using the selected routing algorithm. The movement of packets on the network is simulated. If some packet does not have enough space in the queue of some node, the packet is lost. A model counts all received and lost packets.

Stage 6. A model termination. Occurs after reaching a given number of iterations (for example, 1000 iterations), or if a model works in the second mode of operation, then the stopping condition can also be a state when all queues are empty and all packets are among received or lost.

The Markov chain shown in Fig. 2 is used to generate fractal binary traffic. In this work, a binary time series was created to simulate network traffic, the persistence of which is regulated by setting the probabilities of state change to the opposite λ_1, λ_2 (Fig. 2).

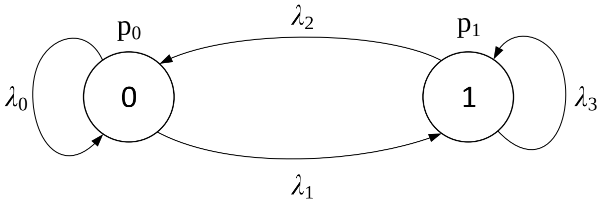


Fig. 2. Markov chain for generating fractal binary traffic

This generator is characterized by states 0 or 1, and the probabilities of being in these states as $p_0 = \lambda_2 /(\lambda_1 + + \lambda_2)$ and $p_1 = \lambda_2 /(\lambda_1 + \lambda_2)$, where λ_i – are the probabilities of the corresponding transitions [16]. The traffic intensity of such a generator will be within [0, 1] and will be equal to the probability of obtaining the output of the generator 1: p_1. The algorithm of operation of such a generator is shown in Fig. 3.

In a developed model the OSPF routing algorithm, which is based on link-state technology and uses Dijkstra's algorithm to find the shortest path, was tested. The obtained results showed the efficiency and usefulness of a developed model. In the future, the authors will test the improvements of this algorithm on a developed model.

The OSPF routing algorithm belongs to the Link State Algorithms.

Link State Algorithms

Link State Algorithms (LSA) provide each router with enough information to construct an accurate network graph. All routers work based on the same graphs, which makes the routing process more resilient to configuration changes. "Broadcast" distribution (that is, the transmission of a packet to all immediate neighbors of the router) takes place only when the state of connections changes, which does not happen so often in reliable networks. The vertices of the graph are both routers and the networks connected by them. Service information distributed over the network consists of a description of connections of various types: router-router, router-network. To understand the state of the communication lines connected to its ports, the router periodically exchanges short HELLO packets with its nearest neighbors.

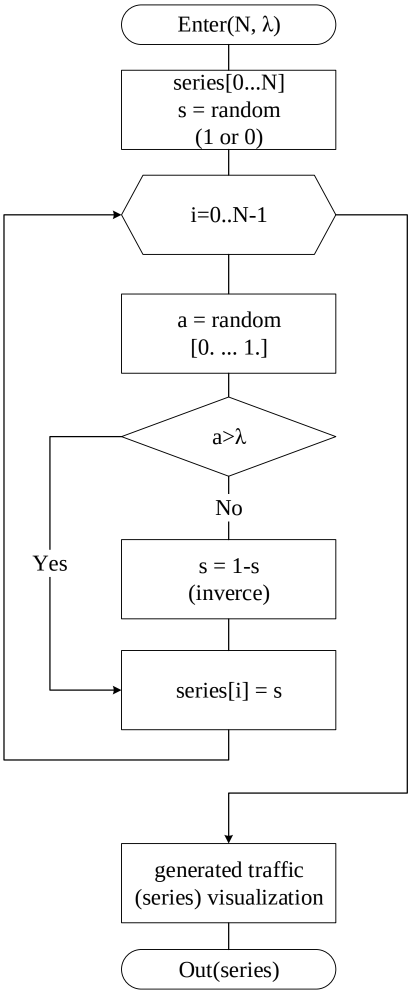


Fig. 3. Algorithm for generating fractal binary traffic

Announcements about the state of connections are not repeated periodically, as in DVA protocols, but are transmitted only when a change in the state of the connection has been established using HELLO messages. As a result, the service traffic generated by LSA protocols is much less intensive than that of DVA protocols. Protocols based on the link state algorithm are the IS-IS (Intermediate System to Intermediate System) protocols of the OSI stack, OSPF (Open Shortest Path First of the TCP/IP stack, and the NLSP protocol of the Novell stack.

Link State Algorithms (also known as algorithms of “priority of the shortest route”, Dijkstra) direct flows of routing information to all nodes of the combined network. Each router sends only the part of the routing table that defines the state of its own channels.

OSPF Link State Algorithm

The OSPF (Open Shortest Path First) protocol is a fairly modern implementation of the link state algorithm and has many features aimed at application in large heterogeneous networks.

Two stages of building the routing table in OSPF

In OSPF, the routing table construction process is divided into two major stages. In the first stage, each router builds a graph of network connections, in which the vertices of the graph are routers and IP-networks, and the edges are router interfaces.

For this, all routers exchange with their neighbors the information about the network graph that they have so far. This process is similar to the process of distributing distance vectors to networks in the RIP protocol, but the information itself is qualitatively different – it is information about the network topology. Such messages are called router links advertisement. In addition, when transmitting topological information, routers do not modify it, as RIP routers do, but transmit it unchanged. As a result of propagation of topological information, all routers of the network have identical information about the network graph, which is stored in the topological database of each router.

The second stage consists in finding optimal routes using the obtained graph. Each router considers itself the center of the network and searches for the optimal route to each known network. In each route found in this way, only one step is remembered - to the next router according to the principle of one-step routing. The data about this step gets into the routing table. The task of finding the optimal path on the graph is quite complex and time-consuming. In the OSPF protocol, Dijkstra's iterative algorithm is used to solve it. If several routes have the same metric to the destination network, then the routing table remembers the first steps of all these routes.

HELLO route announcements in the OSPF algorithm

After the initial construction of the routing table, it is necessary to monitor changes in the state of the network and make adjustments to the routing table. To control the state of connections and neighboring routers, OSPF routers do not use the exchange of a complete routing table, as RIP routers do not very rationally. Instead, they transmit special short HELLO messages. If the state of the network does not change, OSPF routers do not adjust their routing tables and do not send communication announcements to their neighbors. If the state of the connection has changed, then a new announcement is sent to the nearest neighbors, which applies only to this connection, which, of course, saves network bandwidth. After receiving a new announcement about a change in the state of communication, the router rebuilds the network graph, again searches for optimal routes (not necessarily all, but only those affected by this change) and adjusts its routing table. At the same time, the router relays the announcement to each of its nearest neighbors (except the one from which it received the announcement).

With the appearance of a new connection or a new neighbor, the router learns about it from new HELLO messages. HELLO messages contain enough information about the router that sent the message, as well as about its nearest neighbors, that the router can be uniquely identified. HELLO messages are sent every 10 seconds to increase the speed at which routers adapt to changes in the network. The small volume of these messages enables such frequent testing of the status of neighbors and connections with them.

Since routers are among the vertices of the graph, they must have identifiers.

OSPF metrics and announcements

The OSPF protocol usually uses a metric that takes into account the bandwidth of networks. In addition, it is possible to use two other metrics that take into account the requirements for the quality of service in an IP-packet - packet transmission delay and network packet transmission reliability. For each metric, the OSPF protocol builds a separate routing table. The selection of the desired table depends on the requirements for the quality of service of the incoming package.

Routers are connected both to local networks and directly to each other by global point-to-point channels.

The OSPF protocol advertises two types of connections: router-to-router and router-to-network. If point-to-point links are given IP-addresses, they become additional vertices in the graph, just like LANs. Network mask information is also transmitted along with the network IP-address.

After initialization, OSPF routers only know connections to directly connected networks, just like RIP routers. They start spreading this information to their neighbors. Simultaneously, they send HELLO messages on all their interfaces, so that almost immediately each router learns the IDs of its nearest neighbors, which replenishes its topological database with new information that it learned directly. Next, topological information begins to spread through the network from neighbor to neighbor and after some time reaches the most distant routers. Each connection is characterized by a metric. The OSPF protocol supports standard for many protocols (for example, for the Spanning Tree protocol) distance values for metrics that reflect network performance: Ethernet – 10 units, Fast Ethernet – 1 unit, T1 channel – 65 units, 56 kbit/s channel – 1785 units etc.

When choosing the optimal path on a graph, a metric is associated with each edge of the graph, which is added to the path if this edge is part of it.

The OSPF protocol allows multiple routes to the same network to be stored in the routing table if they have equal metrics. If such entries are created in the routing tables, then the router implements load balancing mode, sending packets alternately along each of the routes.

OSPF stability

Each entry in the topology database has a lifetime. Each connection record has a timer associated with it, which is used to control the lifetime of the record. If any topological base entry of a router received from another router becomes out of date, then the router can request a new copy of it using a special Link-State Request message of the OSPF protocol, which should receive a Link-State Update response from the router, which directly testing the request. To initialize routers and more reliably synchronize topological databases, routers periodically exchange all database entries, but this period is much longer than that of RIP routers.

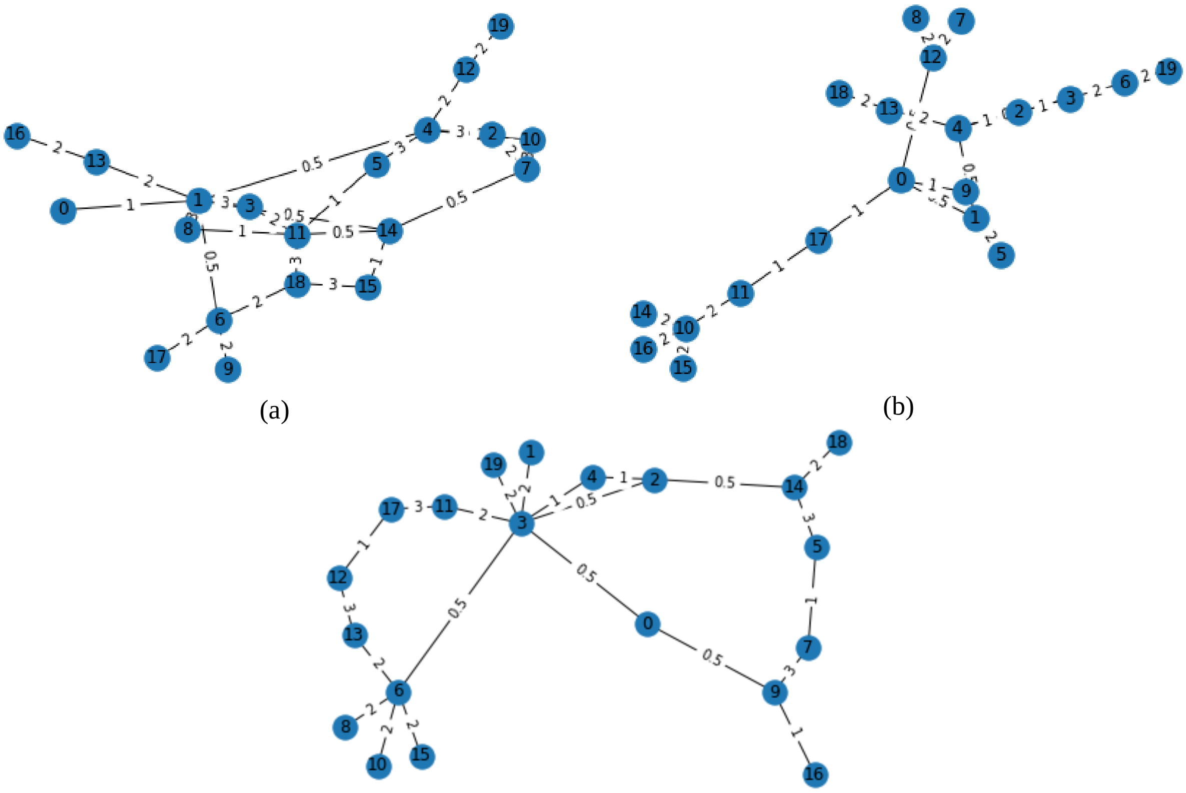
Since information about a certain connection is initially generated only by the router that found out the actual state of this connection by testing with HELLO messages, and the rest of the routers only relay this information without conversion, unreliable information about the reachability of networks, which can appear in RIP routers, cannot appear in OSPF routers, and outdated information is quickly replaced by new information, since a new message is generated immediately when the communication state changes.

OSPF networks may have periods of unstable operation. For example, in the event of a connection failure, when the information about this has not reached any router, it continues to send packets to the destination network, considering this connection to be operational. However, these periods do not last long, and packets do not "get stuck" in routing loops, but are simply discarded due to the impossibility of transmitting them due to an inoperable connection.

Disadvantages of the OSPF protocol include its computational complexity, which rapidly increases with the increase in the size of the network, that is, the number of networks, routers and connections between them. To overcome this drawback, the OSPF protocol introduces the concept of a network area, which should not be confused with an autonomous Internet system. Routers belonging to a certain area build a graph of connections only for this area, which reduces the size of the network. Information about connections is not transferred between areas, and border routers for the areas exchange only information about the addresses of networks that are in each of the areas and the distance from the border router to each network. When transferring packets between regions, one of the border routers of the region is selected, namely the one with the shortest distance to the desired network. This style resembles the RIP style of operation, but the instability here is eliminated by the fact that loop connections between areas are prohibited. When transferring addresses to another area, OSPF routers aggregate several addresses into one if they detect a common prefix. Quality of Service (QoS) – the technology of giving different classes of traffic different priorities in service. Any prioritization makes sense only if there is a queue for service. Right here in the queue, an important IP-packet can go through first based on its priority. The queue is formed where it is narrow (usually such places are called "bottle-neck"). QoS is not a panacea: if the "neck" is too narrow, the physical buffer of the interface, where all the packets that are going to leave through this interface, are often overflowed. And then new packets will be destroyed, even if they are very important and prioritized. Therefore, if the queue on the interface exceeds 20% of its maximum size on average (on cisco routers, the maximum queue size is usually 128-256 packets), there is a reason to think hard about the network design.

An experiment series was conducted on a developed model to determine how different fractal dimensions of traffic at high traffic intensity values affect the number of lost packets, and therefore the quality of service.

Three computer networks were generated, shown in Fig. 4 (a-c). There are 20 routers in each of the networks. The queue length in each router is 128 packets. The traffic intensity was 0.7. The values of the fractal dimension were taken as follows: 1.01, 1.25, 1.37, 1.50, 1.75, 1.87 and 1.99. The first operating mode of a model was used - at each iteration, a random number of traffic packets with random devices of senders and receivers generating and their routing is carried out. The number of lost traffic packets after passing 100 iterations of a model was calculated. The results of an experiments are presented in Table 1, where are shown the average values based on experiments with networks from Fig. 4 (a-c).



(a)

(b)

(c)

Fig. 4. Structures of computer networks generated for the experiment

Table 1 – **Results of a series of experiments to determine how different fractal dimension of traffic at its intensity of 0.7 affects the number of lost packets**

№	Traffic fractal dimension	Maximum quantity packets to be sent from one device to unit of time	Average value lost packets	Average amount devices with lost packets
1	1.01	55	0 .00000	0
2	1.01	60	0 .00000	0
3	1.01	65	54.7981	11
4	1.01	70	54.8461	11
5	1.01	75	54.8521	11
6	1.01	80	54.8568	11
7	1.01	85	54.8718	11
8	1.01	90	54.8866	11
9	1.25	55	0 .00000	0
10	1.25	60	0 .00000	0
11	1.25	65	54.7981	11
12	1.25	70	54.8461	11
13	1.25	75	54.8521	11
14	1.25	80	54.8568	11
15	1.25	85	54.8718	11
16	1.25	90	54.8866	11
17	1.37	55	0 .00000	0
18	1.37	60	0 .00000	0
19	1.37	65	0 .00000	0
20	1.37	70	0 .00000	0
21	1.37	75	0 .00000	0
22	1.37	80	0 .00000	0
23	1.37	85	31.2903	7.3
24	1.37	90	98.817 0	20
25	1.5	55	0 .00000	0
26	1.5	60	0 .00000	0
27	1.5	65	0 .00000	0
28	1.5	70	0 .00000	0
29	1.5	75	0 .00000	0
30	1.5	80	0 .00000	0
31	1.5	85	52.6565	12
32	1.5	90	98.8946	20
33	1.75	55	0 .00000	0
34	1.75	60	0 .00000	0
35	1.75	65	0 .00000	0
36	1.75	70	0 .00000	0
37	1.75	75	15.7478	3.6
38	1.75	80	54.019 0	12
39	1.75	85	96.012 0	19.6
40	1.75	90	99.3858	20
41	1.87	55	0 .00000	0
42	1.87	60	0 .00000	0
43	1.87	65	0 .00000	0
44	1.87	70	0 .00000	0
45	1.87	75	35.7258	8.3
46	1.87	80	92.6598	19.6
47	1.87	85	98.54 00	20
48	1.87	90	99.3523	20
49	1.99	55	0 .00000	0
50	1.99	60	0 .00000	0
51	1.99	65	94.3441	19
52	1.99	70	94.419 0	19
53	1.99	75	92.8931	18.6
54	1.99	80	97.861 0	19.6
55	1.99	85	94.6751	19
56	1.99	90	96.3461	19.3

The fractal dimension varies in the range (1, 2), and its value can be interpreted as follows:

– values smaller than 1.5 – the process is persistent, i.e., maintains its trend, the less the fractal dimension, the stronger the trend is maintained, the closer to 1.5 – the more random the process.

– value 1.5 – the process is completely random.

– values greater than 1.5 – the process is anti-persistent – any trend tends to change to the opposite.

Analyzing the results of the experiment, the following conclusions can be drawn:

– the least number of lost packets when the process is random or has weak trends. The least number of packets were lost at the fractal dimension of 1.5, and there were also few lost packets at the fractal dimensions of 1.37 and 1.75.

– persistent and anti-persistent processes (those with memory) cause more packet loss at the same traffic intensity and maximum number of packets sent from one device per time unit.

Moreover, anti-persistent processes cause significantly greater losses than persistent ones.

Thus, when performing traffic routing and finding optimal paths for sending IP-packets, it can be useful to determine and take into account the fractal dimension of traffic at the entrance of each router and use it when calculating metrics to determine the best routes.

Conclusions

The paper investigated the basic principles of traffic routing in computer networks. A computer model of a computer network for testing traffic routing algorithms has been developed. A method based on the theory of complex networks has been developed to generate the structure of a computer network. Theory of fractal analysis and Markov processes are used for traffic generation. A series of experiments was conducted on a developed model to determine how different fractal dimensions of traffic at high traffic intensity values affect the number of lost packets, and therefore the quality of service.

Analyzing the results of the experiment, the following conclusions can be drawn:

– the least number of lost packets when the process is random or has weak trends. The least number of packets were lost at the fractal dimension of 1.5, and there were also few lost packets at the fractal dimensions of 1.37 and 1.75.

– persistent and anti-persistent processes (those with memory) cause more packet loss at the same traffic intensity and maximum number of packets sent from one device per time unit.

Moreover, anti-persistent processes cause significantly greater losses than persistent ones.

Thus, when performing traffic routing, it can be useful to determine and take into account the fractal dimension of traffic at the entrance of each router and use it when calculating metrics to determine the best routes.

This can allow to take into account traffic trends and unload routers that are too loaded, thereby reducing the number of lost packets.

References

1. Barabási, A.-L. (2018), *Network science*, Cambridge University Press, 475 p., available at: http://networksciencebook.com.
2. Snarskyi, A.O. & Lande, D.V. (2015), *Modeling Complex Networks: Tutorial*, Engineering, Kyiv, 212 p., available at: http://dwl.kiev.ua/art/mss/ (in Russian).
3. Traag, V.A. (2014), *Algorithms and Dynamical Models for Communities and Reputation in Social Networks*, Springer International Publishing, 229 p., doi: https://doi.org/10.1007/978-3-319-06391-1.
4. Watts, D.J. & Strogatz, S.H. (1998), "Collective dynamics of "small-world" networks", *Nature*, Vol. 393(6684), pp. 440-442, available at: https://www.nature.com/articles/30918.
5. Drieieva, H., Drieiev, O., Meleshko, Ye., Yakymenko, M. & Mikhav, V. (2022), "A method of determining the fractal dimension of network traffic by its probabilistic properties and experimental research of the quality of this method", *CEUR -WS*, Vol. 3171, Gliwice, Poland, R. 1694-1707, available at: http://ceur-ws.org/Vol-3171/paper120.pdf.
6. Meleshko, Ye., Drieiev, O., Yakymenko, M. & Lysytsia, D. (2020), "Developing a model of the dynamics of states of a recommendation system under conditions of profile injection attacks", *Eastern-E* doi: https://doi.org/10*uropean Journal of Enterprise Technologies*, Vol. 4, No. 2(106), pp. 14-24, doi: https://doi.org/10.15587/1729-4061.2020.209047.
7. Meleshko, Ye., Raskin, L., Semenov, S. & Sira, O. (2019), "Methodology of probabilistic analysis of state dynamics of multi-dimensional semi-Markov dynamic systems", *Eastern-European Journal of Enterprise Technologies*, Vol. 6, No. 4(102), pp. 6-13, doi: https://doi.org/10.15587/1729-4061.2019.184637.
8. Dimitrakos, T.D. & Kyriakidis, E.G. (2008), "A semi-Markov decision algorithm for the maintenance of a production system with buffer capacity and continuous repair times", International Journal of Production Economics, Vol. 111(2), pp. 752-762, doi: https://doi.org/10.1016/ j.ijpe .2007.03.010.
9. Li, Q.-L. & Lui, JCS, (2014), "Block-structured supermarket models", *Discrete Event Dynamic Systems*, Vol. 26(2), pp. 147-182, doi: https:// doi.org/10.1007/s10626-014-0199-1.
10. Okamura, H., Miyata, S. & Dohi, T., (2015), "A Markov Decision Process Approach to Dynamic Power Management in a Cluster System", IEEE Access, Vol. 3, pp. 3039-3047, doi: https://doi.org/10.1109/access.2015.2508601.
11. Li, Q.-L. (2016), "Nonlinear Markov processes in large networks", Special Matrices, Vol. 4(1), doi: https://doi.org/10.1515/spma-2016-0019.
12. Feinberg, E.A. & Yang, F. (2015), "Optimal pricing for a GI/M/k/N queue with several customer types and holding costs", Queuing Systems, Vol. 82(1-2), pp. 103-120, doi: https://doi.org/10.1007/s11134-015-9457-7.
13. Olifer, N. & Olifer, V. (2005), "Computer Networks: Principles", Technologies and Protocols for Network Design 1st Edition, Wiley, 1000 p.
14. Cisco (2022), "IP Routed Protocols", available at: https://www.cisco.com/c/en/us/tech/ip/ip-routed-protocols/index.html.
15. Barabási, A.-L. & Albert, R. (1999), "Emergence of scaling in random networks", Science, Vol. 286, No. 5439, P. 509-512, doi: https://doi.org/10.1126/science.286.5439.509.

16. Drieieva, H., Smirnov, O., Drieiev, O., Polishchuk, Y., Brzhanov, R. & Aleksander, M. (2020), “Method of Fractal Traffic Generation by a Model of Generator on the Graph”, COAPSN, CEUR-WS, Vol. 2616, Lviv, Ukraine, available at: http://ceur-ws.org/Vol-2616/paper31.pdf.

Received (Надійшла) 17.08.2022
Accepted for publication (Прийнята до друку) 19.10.2022

ВІДОМОСТІ ПРО АВТОРІВ / ABOUT THE AUTHORS

Дрєєва Ганна Миколаївна – аспірант кафедри кібербезпеки та програмного забезпечення, Центральноукраїнський національний технічний університет, Кропивницький, Україна;
Hanna Drieieva – Postgraduate student of Cybersecurity and Software Department, Central Ukrainian National Technical University, Kropyvnytskyi, Ukraine;
e-mail: gannadreeva@gmail.com; ORCID ID: https://orcid.org/0000-0002-8557-3443.

Мелешко Єлизавета Владиславівна – доктор технічних наук, професор, доцент кафедри кібербезпеки та програмного забезпечення, Центральноукраїнський національний технічний університет, Кропивницький, Україна;
Yelyzaveta Meleshko – Doctor of Engineering Sciences, Professor, Associate Professor of Cybersecurity and Software Department, Central Ukrainian National Technical University, Kropyvnytskyi, Ukraine;
e-mail: elismeleshko@gmail.com; ORCID ID: https://orcid.org/0000-0001-8791-0063.

Дрєєв Олександр Миколайович – кандидат технічних наук, доцент, доцент кафедри кібербезпеки та програмного забезпечення, Центральноукраїнський національний технічний університет, Кропивницький, Україна;
Oleksandr Drieiev – Candidate of Engineering Sciences, Associate Professor, Associate Professor of Cybersecurity and Software Department, Central Ukrainian National Technical University, Kropyvnytskyi, Ukraine;
e-mail: drey.sanya@gmail.com; ORCID ID: https://orcid.org/0000-0001-6951-2002.

Міхав Володимир Володимирович – аспірант кафедри кібербезпеки та програмного забезпечення, Центральноукраїнський національний технічний університет, Кропивницький, Україна;
Volodymyr Mikhav – Postgraduate student of Cybersecurity and Software Department, Central Ukrainian National Technical University, Kropyvnytskyi, Ukraine;
e-mail: mihaw.wolodymyr@gmail.com; ORCID ID: https://orcid.org/0000-0003-4816-4680.

Програмна імітаційна модель комп’ютерної мережі з фрактальним трафіком для тестування алгоритмів маршрутизації

Г. М. Дрєєва, Є. В. Мелешко, О. М. Дрєєв, В. В. Міхав

Анотація. Об’єктом вивчення у статті є імітаційне моделювання комп’ютерної мережі та процес маршрутизації трафіку. Актуальність дослідження зумовлена важливістю забезпечення якості обслуговування у комп’ютерних мережах, зокрема, зменшенню кількості втрачених IP-пакетів при високих значеннях інтенсивності трафіку. Визначення маршруту передачі пакетів трафіку є складним процесом і базується на різних показниках або комбінаціях показників. Якщо процес маршрутизації відбувається у динамічному режимі, то складність розрахунку маршруту зростає, в такому разі одним з інструментів дослідження та порівняння різних алгоритмів маршрутизації може стати програмна імітаційна модель комп’ютерної мережі. **Метою** роботи є розробка та дослідження програмної імітаційної моделі комп’ютерної мережі для тестування алгоритмів маршрутизації трафіку. **Завдання:** створити програмну імітаційну модель комп’ютерної мережі для генерації структури мережі та симуляції процесу поширення трафіку з можливістю тестувати різні алгоритми маршрутизації. **Методи досліджень:** теорія комп’ютерних мереж, теорія фрактального аналізу, об’єктно-орієнтоване програмування, теорія алгоритмів та структур даних, теорія складних мереж, теорія марківських процесів. **Висновки.** У роботі було досліджено основні принципи маршрутизації трафіку у комп’ютерних мережах. Розроблено програмну модель комп’ютерної мережі для тестування алгоритмів марштуртизації трафіку. Для генерації структури комп’ютерної мережі розроблено метод на основі теорії складних мереж. Для генерації трафіку використано теорію фрактального аналізу та марківські процеси. На розробленій моделі було проведено серію експериментів для визначення як різна фрактальна розмірність трафіку при високих значеннях інтенсивності трафіку впливає на кількість втрачених пакетів, а отже і якість обслуговування. Аналізуючи результати експерименту можна зробити наступні висновки: найменше втрачених пакетів, коли процес випадковий, або має слабко виражені тренди. Найменше втрачених пакетів було при фрактальній розмірності 1.5, тобто коли процес повністю випадковий, також мало втрачених пакетів було при фрактальних ромірностях близьких до даної; персистивні та антиперсистивні процеси (такі, що мають пам’ять), викликають більше втрати пакетів при тій же інтенсивності трафіку та максимальних кількостях пакетів на відправку з одного пристрою в одиницю часу. При чому антиперситентні процеси викликають значно більші втрати, ніж персистентні. Таким чином при виконанні маршрутизації трафіку та пошуку оптимальних шляхів для відправки IP-пакетів може бути корисним визначати та враховувати фрактальну розмірність трафіку на вході кожного маршрутизатора та використовувати її при розрахунку метрик для визначення найкращих маршрутів.

Ключові слова: програмна імітаційна модель; комп’ютерна мережа; маршрутизація, фрактальна розмірність; складні мережі; мережевий трафік; якість обслуговування.