A Method of Routing of Fractal-like Traffic with Prediction of Router Load for Reduce the Probability of Network Packet Loss

Yelyzaveta Meleshko, Hanna Drieieva, Oleksandr Drieiev, Mykola Yakymenko, Volodymyr Mikhav and Serhii Shymko

Central Ukrainian National Technical University, 8, Universytetskyi prosp., Kropyvnytskyi, 25006, Ukraine

Abstract

In this paper, a method for routing fractal-like traffic in computer networks was proposed. This method uses the prediction of router load by analyzing the fractal dimension of network traffic to reduce the probability of packet loss. It takes into account the predicted router load as one of the metrics for determining the shortest packet transmission routes in a computer network. Additionally, a computer simulation model of a computer network based on complex network theory, Markov processes, and fractal time series was created. This computer simulation model allows the generation of a computer network structure and simulates traffic movement between network devices for testing routing algorithms. A series of experiments on the developed computer network model to determine the quality of the proposed routing method and compare it with other methods were conducted. During the analytical research and experiments, the impact of different fractal dimensions of traffic on the probability of packet loss and, consequently, on the quality of service at high traffic intensity was investigated. And it also investigated whether the proposed routing method allows for the reduction of the number of lost network packets. Analyzing the results of the experiments, the following conclusions can be drawn. The fewest lost packets were when the process was random or had weakly expressed trends, which was modeled in the experiment by traffic with the fractal dimension equal to 1.5. Persistent and anti-persistent processes (those with memory) cause more packet loss for the same traffic intensity and the same maximum number of packets generated per device per unit of time. Moreover, the anti-persistent processes modeled in the experiment by traffic with the fractal dimension equal to 1.25 cause significantly greater losses than persistent processes modeled with the fractal dimension of 1.75. Also, the results of the experiments showed that the proposed traffic routing method allows for a significantly reduced number of lost packets compared to the existing method without prediction based on fractal traffic analysis.

Keywords

Computer networks, traffic routing, network traffic, fractal-like traffic, fractal dimension, packet loss probability, router load prediction, time series forecasting, data analysis, computer simulation

1. Introduction

The relevance of the research is due to the importance of ensuring the quality of service in computer networks, in particular, reducing the number of lost IP-packages at high values of traffic intensity, which will significantly improve the quality of service at peak loads on the network [1, 2]. Determining the route of transmission of traffic packets is a complex process and is based on various metrics or combinations of metrics. If the process of routing takes place in a dynamic mode, the complexity of

COLINS-2023: 7th International Conference on Computational Linguistics and Intelligent Systems, April 20-21, 2023, Kharkiv, Ukraine
EMAIL: elismeleshko@gmail.com (Ye. Meleshko); gannadreeva@gmail.com (H. Drieieva); drey.sanya@gmail.com (O. Drieiev); m.yakymenko@gmail.com (M. Yakymenko); mihaw.wolodymyr@gmail.com (V. Mikhav); shymko.sv@meta.ua (S. Shymko)
ORCID: 0000-0001-8791-0063 (Ye. Meleshko); 0000-0002-8557-3443 (H. Drieieva); 0000-0001-6951-2002 (O. Drieiev); 0000-0003-3290-6088 (M. Yakymenko); 0000-0003-4816-4680 (V. Mikhav); 0000-0002-1132-484X (S. Shymko)

route determining increases, in this case, one of the tools for researching and comparing various routing algorithms can be a computer simulation model of a computer network [3-5].

The paper research the basic principles of traffic routing in computer networks. It was revealed that existing methods of traffic routing can be improved based on the use of predicting the load of routers [6-9]. Also, the research showed that computer traffic has fractal properties [10-12], which can be used in the development of methods for predicting the load of network devices.

A computer model of a computer network for testing traffic marching algorithms has been developed. To generate the structure of a computer network, a method based on the theory of complex networks has been developed. Markov processes and fractal time series were used to generate traffic. An improved traffic routing method was also proposed, using router load prediction based on fractal analysis to reduce the likelihood of losing network packets. On the developed computer simulation model of a computer network, a series of experiments were conducted to determine the quality of work of a developed method of routing fractal-like traffic.

1.1. Related works and problem statement

Network traffic has fractal properties and can be modeled using fractal time series [13]. The generation of traffic to reproduce its fractal properties [14, 15] can be based on the theory of Markov processes [16, 17], which is often used to model the traffic of various mass service systems [18-20] and will be useful for creating high-quality models of computer networks and testing their work methods. Another possible application of the analysis and synthesis of fractal traffic can be the detection of information attacks in computer [21-23] and social networks [24] because mathematical and computer models of networks and network traffic are also widely used for development and testing of methods of attack detection [25-27].

A structure of computer networks is often modeled using complex networks – stochastic networks with non-trivial topology, differing from classical stochastic networks by their properties [5, 28]. Most real networks – complex ones, for example, computer, transport, and social networks are complex. Complex networks have the following basic properties [5, 29, 30]: scalelessness, the small diameter of a network, in high clustering coefficient and high transitivity coefficient, giant connected component. (i.e, more than 80% of nodes are interconnected, in our model of a computer network, complete connectivity is necessary), there are hierarchical connections, there are complex cluster formations (cliques, clans, etc.), assortativity (an emergence of connections between vertices that are somehow similar to each other, in the narrow sense – a emergence of connections between vertices with a large number of connections).

Routing is the process of determining the optimal route for information to pass through computer networks [31, 32]. Each router makes a decision on the direction of packet forwarding based on a routing table. The routing table contains a set of rules, with each rule describing the gateway or interface used by the router to access a particular network. Routes can be configured administratively (static routes) or computed using routing algorithms based on information about network topology and state obtained through routing protocols (dynamic routes). A routing protocol is a network protocol used by routers to determine possible routes for data transmission in a complex large computer network [31, 32]. Routing protocols are divided into two types depending on the types of algorithms they are based on [31, 32]: distance vector protocols and link state protocols. Examples of distance vector protocols include Routing Information Protocol (RIP), Interior Gateway Routing Protocol (IGRP), Border Gateway Protocol (BGP), and Ad hoc On-Demand Distance Vector (AODV). Examples of link state protocols include (Intermediate System to Intermediate System (IS-IS), Open Shortest Path First (OSPF), NetWare Link-Services Protocol (NLSP), Hot Standby Router Protocol (HSRP), Optimized Link-State Routing (OLSR), Topology broadcast based on reverse-path forwarding (TBRPF). Distance vector protocol 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, channel 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 path to forward an IP-packet: route length, reliability, delay, bandwidth,

load, and communication cost.

The larger and more complex a computer network, the greater the demands placed on routing algorithms to ensure the required quality of service. Testing is important in researching, improving, and developing routing algorithms. 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 confidently stated that a quality computer simulation model will significantly speed up the development process in the initial stages, and final experiments before practical implementation should be conducted on real computer networks.

The goal of this work is to develop a method of routing fractal-like traffic with prediction of router load to reduce the probability of network packet loss and investigate the quality of proposed method in proposed computer simulation model of a computer network with a complex structure and fractal traffic.

2. Development of a method of routing fractal-like traffic with the prediction of router load to reduce the probability of network packet loss

2.1. Theoretical justification and essence of proposed method

Research was conducted based on mathematical modeling of how the fractal dimension of traffic affects the probability of queue overflow in a router and network packet loss.

A Markov chain, depicted in Fig. 1, was used to model binary fractal traffic.

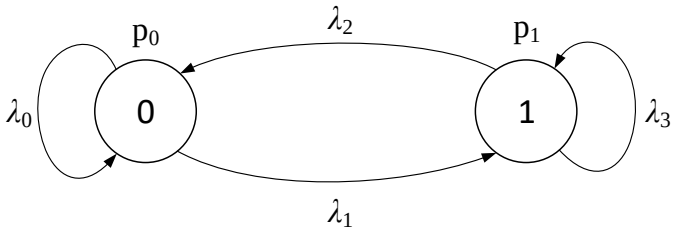


Figure 1: A model of a generator for fractal-like traffic based on a Markov chain

For a fractal binary traffic generator shown in Fig. 1, the fractal dimension by introduced metric M is expressed through a probabilities λ_1 and λ_2 of changing a current state to a opposite as follows [13]:

$$d(\lambda_1,\lambda_2) = 2 + \frac{\lambda_2(1-\lambda_1)\ln(1-\lambda_1) + \lambda_1(1-\lambda_2)\ln(1-\lambda_2)}{2\lambda_1\lambda_2}. \quad (1)$$

The flow of unit values takes values in the range [0; 1] and can be found as follows:

$$\tau = \frac{\lambda_1}{\lambda_1+\lambda_2}. \quad (2)$$

To simulate a real binary sequence of network packets, it is sufficient to estimate the probabilities λ_1 and λ_2. To simulate traffic through the flow value τ and fractal dimension d, it is necessary to find the probabilities λ_1 and λ_2, which are unknown. Therefore, the following problem is formulated:

Given the fractal dimension d and the average flow of unit values τ. The input values are limited as follows: $d \in (1..2), \tau \in (0..1)$.

The probabilities λ_1 and λ_2 are sought, where $\lambda_1, \lambda_2 \in (0..1)$.

Let's solve the formulated problem.

From formula (2), we determine the probability of transition from "1" state to "0" state λ_2 through another probability λ_1:

$$\lambda_2 = \lambda_1 \frac{1-\tau}{\tau}. \quad (3)$$

Thanks to this, equation (1) can be reduced by substitution (3) to an equation with one variable:

$$d = 2 + (1-\lambda_1)\frac{ln(1-\lambda_1)}{2\lambda_1} + (1-\lambda_1(1-\tau)/\tau)\frac{ln(1-\lambda_1(1-\tau)/\tau)}{2\lambda_1(1-\tau)/\tau}. \quad (4)$$

To simplify further notation, we assume that $\lambda=\lambda_1$:

$$d = 2 + (1-\lambda)\frac{ln(1-\lambda)}{2\lambda} + (1-\lambda(1-\tau)/\tau)\frac{ln(1-\lambda(1-\tau)/\tau)}{2\lambda(1-\tau)/\tau}. \quad (5)$$

Equation (5) is nonlinear with respect to λ and has no analytical solutions. Therefore, to solve for λ, it is necessary to transform it to search for zeros:

$$f(\lambda) = 2 - d + (1-\lambda)\frac{ln(1-\lambda)}{2\lambda} + (1-\lambda(1-\tau)/\tau)\frac{ln(1-\lambda(1-\tau)/\tau)}{2\lambda(1-\tau)/\tau}, \quad (6)$$

and use one of the numerical methods for iterative approximation. The method of tangent lines is proposed to be used. The following algorithm is shown, where the superscript denotes the result of the current iteration:

Stage 1. $\lambda^0 = 0.00001$.

Stage 2. $\lambda^{i+1} := \lambda^i - k\, f(\lambda^i)/f'(\lambda^i)$, where $k \in (0..1]$ is a coefficient for improving the convergence of the method (the smaller the coefficient, the more stable the method, but it requires $1/k$ more iterations). Repeat step (2) until the desired accuracy is achieved, after which the sought-after values are calculated:

Stage 3. $\lambda_1 = \lambda$, $\lambda_2 = \lambda(1-\tau)/\tau$.

It is also possible to use other numerical methods.

The small initial probability value corresponds to the region of stable solutions, where the method of tangent lines more often leads to a solution of the equation.

The derivative for the expression from step (2) is calculated using the following expression:

$$f'(\lambda) = -\frac{\lambda + ln(1-\lambda)}{2\lambda^2} - \frac{\lambda\cdot(1-\tau)/\tau + ln(1-\lambda(1-\tau)/\tau)}{2\lambda^2(1-\tau)/\tau} \quad (7)$$

The obtained iterative process allows you to obtain parameters for traffic generation λ_1 and λ_2 from the fractal dimension d and the flow intensity of generation of ones τ. As a result of numerical simulation modeling, at different values of fractal dimension and input traffic intensity, the average queue length in the node device and the probability of queue overflow for 10 packets were obtained.

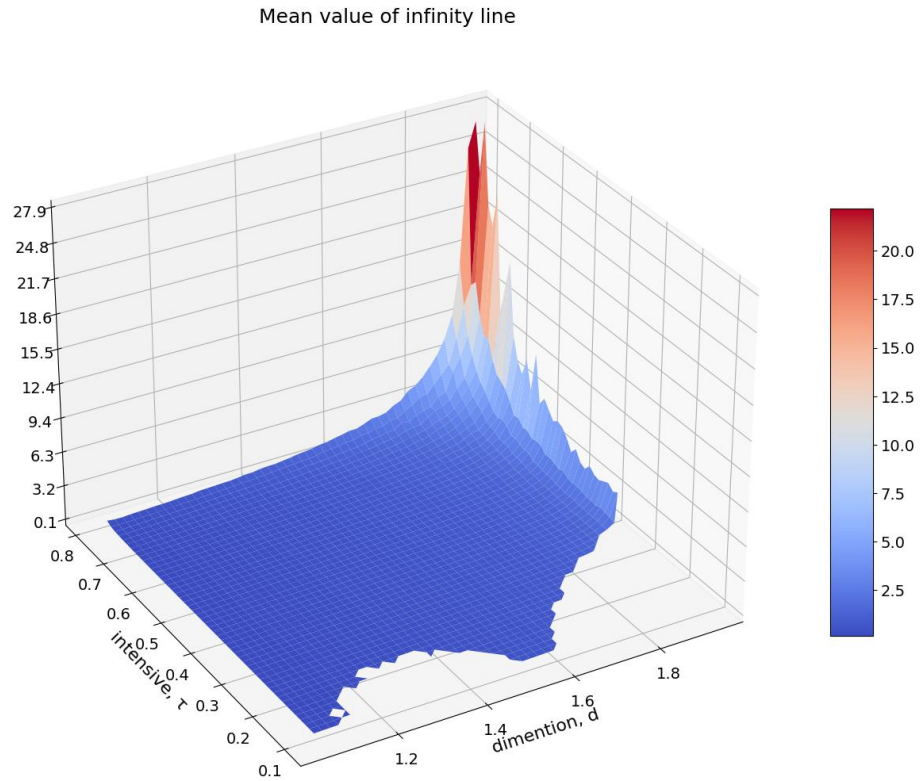


Figure 2: Average values of the queue length in a nodal device at different values of the fractal dimension and the intensity of input traffic, obtained as a result of numerical simulation

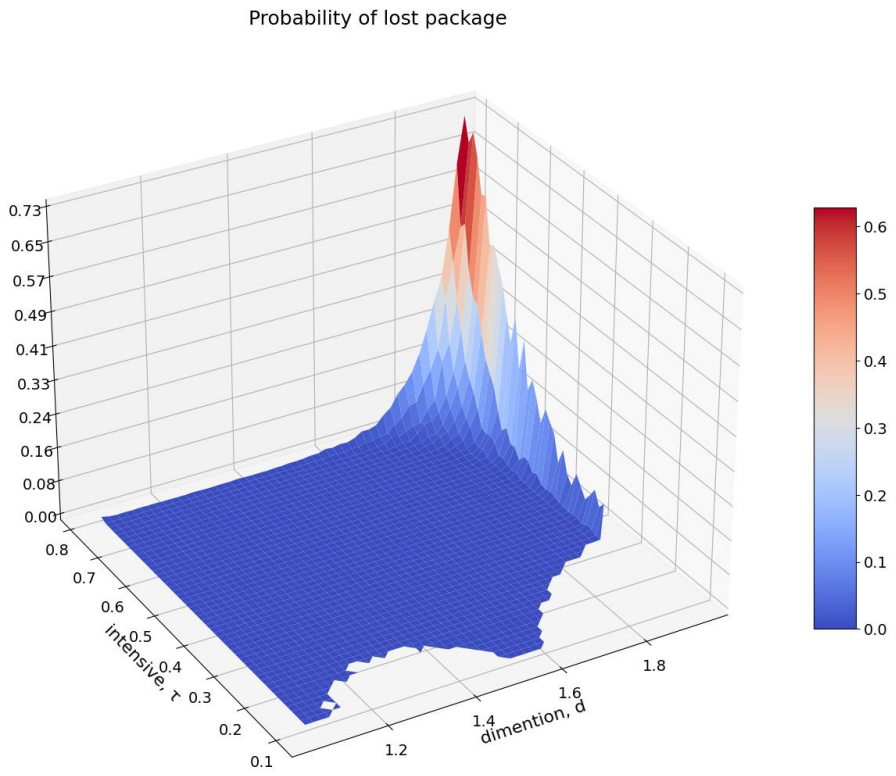


Figure 3: Probability of queue overflow by 10 packets at different values of fractal dimension and input traffic intensity, obtained as a result of numerical simulation

The empty areas correspond to inadmissible solutions when searching for the probabilities of state change of the generator λ_1 and λ_2. Accordingly, for the given d and τ, the generator is unable to produce a sequence with the specified characteristics, which is visualized by the region of admissible arguments (domain of definition), Fig. 4:

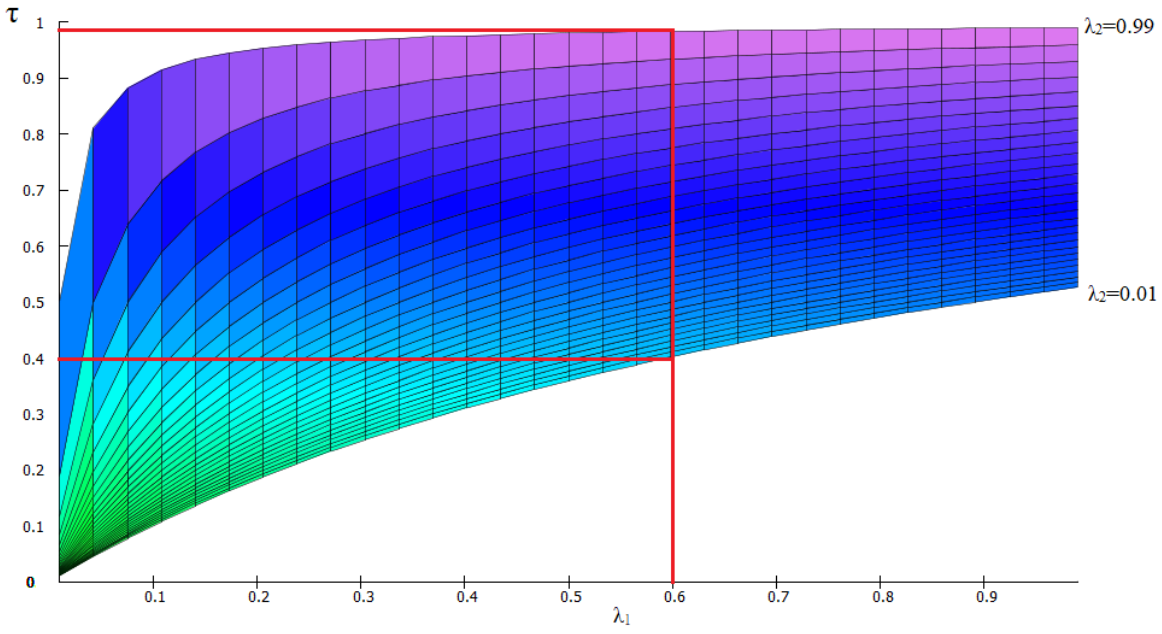


Figure 4: Possible values of the flow intensity depend on a fixed value of one of the probabilities. That is, not all triples of $\tau, \lambda_1, \lambda_2 \in (0..1)$ are possible

Let there be a probability of packet loss due to queue overflow $p=f(\tau, d)$. Let's find its approximation using the obtained tabular data (Fig. 3). We look for an approximation in the form of:

$$p_i \approx e^{-a(2-d_i)^n - b(1-\tau_i)^n}, \tag{8}$$

where a, b are the sought coefficients, n is given.

However, in this form, the method of least squares cannot be applied. Therefore, it is necessary to take the logarithm of both sides of the equation:

$$ln(p_i) \approx -a(2-d_i)^n - b(1-\tau_i)^n, \tag{9}$$

The approximation error will be considered as the difference:

$$\Delta_i^2 = (ln(p_i) + a(2-d_i)^n + b(1-\tau_i)^n)^2, \tag{10}$$

or:

$$\begin{aligned}\Delta_i^2 = ln(p_i)^2 + 2aln(p_i)(2-d_i)^n + 2bln(p_i)(1-\tau_i)^n + a^2(2-d_i)^{2n} \\ + 2ab(2-d_i)^n(1-\tau_i)^n + b^2(1-\tau_i)^{2n},\end{aligned} \tag{11}$$

where used the power of 2 to enhance sensitivity to large deviations and to exclude situations of mutual compensation of positive and negative deviations. Then, the quality of the approximation can be determined as the sum of all square deviations, where a smaller value corresponds to a better approximation:

$$\begin{aligned}\sum_i \Delta_i^2 = \sum_i ln(p_i)^2 + 2a\sum_i ln(p_i)(2-d_i)^n + 2b\sum_i ln(p_i)(1-\tau_i)^n + \\ +a^2\sum_i (2-d_i)^{2n} + 2ab\sum_i (2-d_i)^n(1-\tau_i)^n + b^2\sum_i (1-\tau_i)^{2n}.\end{aligned} \tag{12}$$

To obtain an approximation, it is necessary to solve the problem of finding a and b such that:

$$\sum_i \Delta_i^2 \underset{a,b}{\rightarrow} min. \tag{13}$$

The approximation error function with respect to the variables a and b has derivatives equal to zero. With respect to n, the system will be nonlinear and will not have an analytical or unique solution. This makes it possible to construct the following system:

$$\begin{cases}\left(\sum_i \Delta_i^2\right)'_a = 0, \\ \left(\sum_i \Delta_i^2\right)'_b = 0.\end{cases} \tag{14}$$

Let's find the partial derivatives:

$$\left(\sum_i \Delta_i^2\right)'_a = 2\sum_i ln(p_i)(2-d_i)^n + 2a\sum_i (2-d_i)^{2n} + 2b\sum_i (2-d_i)^n(1-\tau_i)^n, \tag{15}$$

$$\left(\sum_i \Delta_i^2\right)'_b = 2\sum_i ln(p_i)(1-\tau_i)^n + 2a\sum_i (2-d_i)^n(1-\tau_i)^n + 2b\sum_i (1-\tau_i)^{2n}. \tag{16}$$

Then the system of equations will have the following form:

$$\begin{cases}2\sum_i ln(p_i)(2-d_i)^n + 2a\sum_i (2-d_i)^{2n} + 2b\sum_i (2-d_i)^n(1-\tau_i)^n = 0, \\ 2\sum_i ln(p_i)(1-\tau_i)^n + 2a\sum_i (2-d_i)^n(1-\tau_i)^n + 2b\sum_i (1-\tau_i)^{2n} = 0.\end{cases} \tag{17}$$

Solution of the system yields the following result:

$$a = \frac{-\sum_i ln(p_i)(2-d_i)^n \sum_i (1-\tau_i)^{2n} + \sum_i ln(p_i)(1-\tau_i)^n \sum_i (2-d_i)^n(1-\tau_i)^n}{\sum_i (2-d_i)^{2n} \sum_i (1-\tau_i)^{2n} - (\sum_i (2-d_i)^n(1-\tau_i)^n)^2}, \tag{18}$$

$$b = \frac{-\sum_i ln(p_i)(1-\tau_i)^n \sum_i (2-d_i)^{2n} + \sum_i ln(p_i)(2-d_i)^n \sum_i (2-d_i)^n(1-\tau_i)^n}{\sum_i (2-d_i)^{2n} \sum_i (1-\tau_i)^{2n} - (\sum_i (2-d_i)^n(1-\tau_i)^n)^2}. \tag{19}$$

On Fig. 5 shows the dependence of the sum of deviations $S = \sum_i \Delta_i^2$ on the exponent *n*.

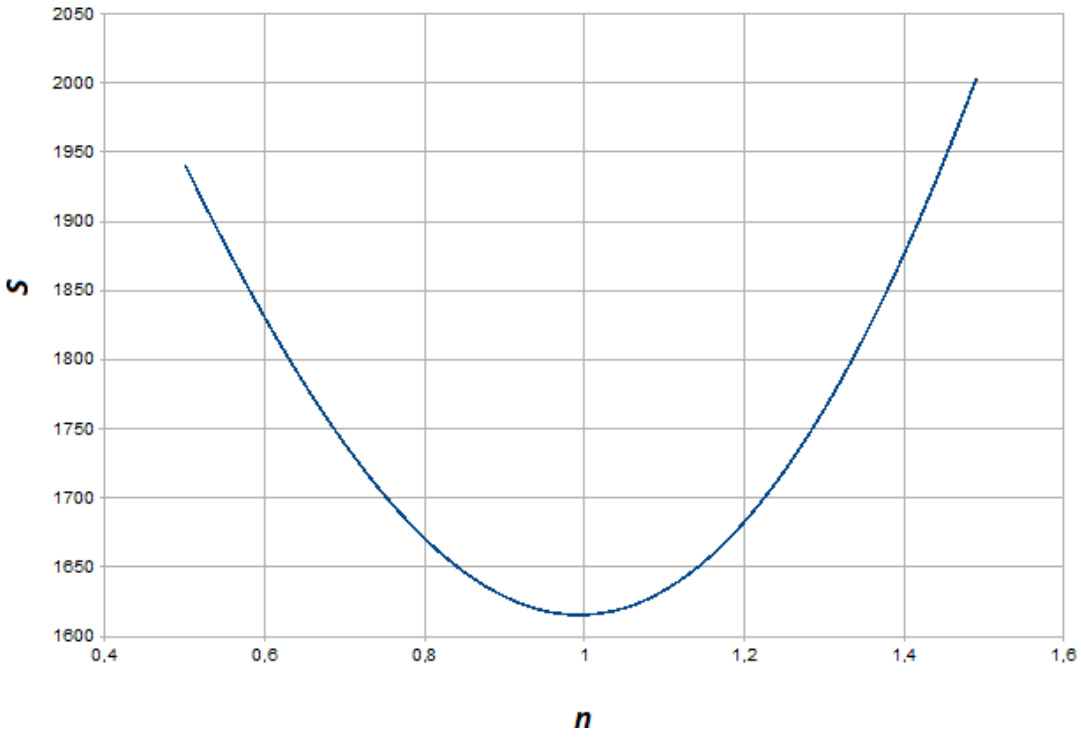


Figure 5: Dependence of the sum of deviations $S = \sum_i \Delta_i^2$ on the exponent *n*.

Due to the nonlinearity of the equations obtained with respect to n, a series of experiments were conducted with different values of the power exponent n. Based on the results, the following parameters were selected:

$$n = 0{,}99,\ a = 6{,}6297,\ b = 11{,}1654,\ \sum_i \Delta_i^2 = 1615{,}4651 \quad (20)$$

Finally, one can use the following approximation to determine the probability of losing network packets for a given known fractal dimension of traffic:

$$p(d,\tau) \approx e^{-6{,}6297(2-d)^{0{,}99}-11{,}1654(1-\tau)^{0{,}99}}, \quad (21)$$

where *p* is the probability of packet loss when transmitted over a separate channel, which depends on the traffic intensity τ and its fractal dimension *d*.

Link State Protocol algorithms, also known as "first-priority shortest path" algorithms, are commonly based on Dijkstra's algorithm and use different metrics to determine the shortest path for forwarding network packets [31, 32, 34]. Their implementation involves the ability to collect statistics on traffic passing through a router, which enables traffic analysis and prediction. The ability to use different distance metrics and combine multiple metrics allows adding packet loss probability as a metric.

OSPF (Open Shortest Path First) protocol is a modern implementation of the link-state algorithm and has many features designed for use in large heterogeneous networks [35-39].

In OSPF, the process of building a routing table is divided into two main stages. In the first stage, each router builds a network topology graph, where a nodes of a graph are routers and IP networks, and a edges are a router interfaces. All routers exchange information about a network graph with their neighbors, which they have at that time. This process is similar to the process of spreading distance vectors to networks in RIP protocol, but information itself is qualitatively different – it is information about a network topology. Such messages are called router link advertisements. Moreover, during the transmission of topological information, routers do not modify it, as RIP routers do, but transmit it in an unchanged form. As a result of spreading topological information, all routers in a network have identical information about a network graph, which is stored in a topological database of each router.

The second stage involves finding a optimal routes using the obtained graph. Each router considers itself the center of a network and seeks the optimal route to each network it knows. In each found route, only one step is remembered – to a next router according to the principle of single-step routing. This data about this step is entered into a routing table. Finding a optimal path on a graph is quite complex and time-consuming. OSPF protocol uses an iterative Dijkstra algorithm to solve it. If several routes have a same metric to a destination network, then the first steps of all these routes are recorded in the routing tables.

After the initial routing table is built, it is necessary to track changes in a network and make adjustments to a routing table. To monitor a state of links and neighboring routers, OSPF routers send special short HELLO messages. If the network state does not change, OSPF routers do not make any adjustments to their routing tables and do not send link state advertisements to their neighbors. However, if a link state changes, a router sends a new advertisement that pertains only to that link, saving network bandwidth. Upon receiving a new advertisement about a link state change, a router reconstructs a network graph, searches for optimal routes (not necessarily all, but only those affected by the change), and adjusts its routing table accordingly. Simultaneously, a router retransmits the advertisement to all its nearest neighbors (except for the one from which it received the advertisement).

With the appearance of a new link or neighbor, a router learns about it from new HELLO messages. HELLO messages contain fairly detailed information about a router that sent the message, as well as its nearest neighbors, to uniquely identify this router. HELLO messages are sent every 10 seconds to increase the speed of router adaptation to changes in a network. The small size of these messages enables frequent testing of the state of neighbors and their connections.

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

OSPF protocol typically uses a metric that takes into account the network's bandwidth. In addition, it is possible to use two other metrics that take into account the quality of service requirements for IP-packet – packet transmission delay and packet transmission reliability in the network. OSPF protocol builds a separate routing table for each metric. The choice of the required table depends on the quality of service requirements for an input packet.

Taking into account the features of OSPF algorithm, an improved method of routing fractal-like traffic was developed.

The stages of the proposed method of routing fractal-like traffic with predicting the load on the routers to reduce the probability of loss of network packets are:

Stage 1. OSPF routing protocol is launched. At first, it operates in normal mode, but on each router, traffic statistics are accumulated for a certain time interval T_n.

Stage 2. Based on the obtained traffic statistics for time T_n, the fractal dimension of traffic is calculated, and based on it, the probability of packet loss $p(d, \tau)$ is predicted in the future on each router. Routers exchange their packet loss probability $p(d, \tau)$ with others via HELLO messages.

Stage 3. The predicted packet loss probability is used as an additional metric in SPF routing algorithm of OSPF protocol (SPF algorithm uses Dijkstra algorithm to find a shortest path for packet transmission). We add the predicted probability of packet loss to the standard path length metric:

$$m = k_1 \cdot m_1 + k_2 \cdot m_2, \quad (22)$$

where m_1 is the standard metric, m_2 is the predicted packet loss probability, and k_1 and k_2 are weighting coefficients, which were set to 1 in this work. The metric $m_2 = p(d, \tau)$ is determined by formula (21).

2.2. Computer simulation model of a computer network for testing traffic routing methods

Developed simulation model of a computer network is represented by a fully connected undirected weighted graph, in which nodes are routers and edges are network connections between them. The weight of edges is the value inversely proportional to the bandwidth of the communication channel. Nodes contain queues in which received packets are placed before determining the route for their transmission and sending them to next node. Time in the model is represented by discrete iterations. Routing is carried out based on an algorithm that needs to be tested on the model.

Developed model includes two operation modes:

On each iteration, a random amount of traffic packets with random sender and receiver devices is generated and routed.

On the first iteration, a certain amount of traffic packets with random sender and receiver devices is generated once, and only their routing is performed on all subsequent iterations.

The stages of the developed computer simulation model of a computer network are:

Stage 1. A computer network structure, where nodes are routers, and edges are traffic transmission channels, is generated (Fig. 6) based on Barabási-Albert model [33].

Stage 2. Checking whether a generated network graph is fully connected. If a generated graph is not fully connected, add edges between disconnected components of graph.

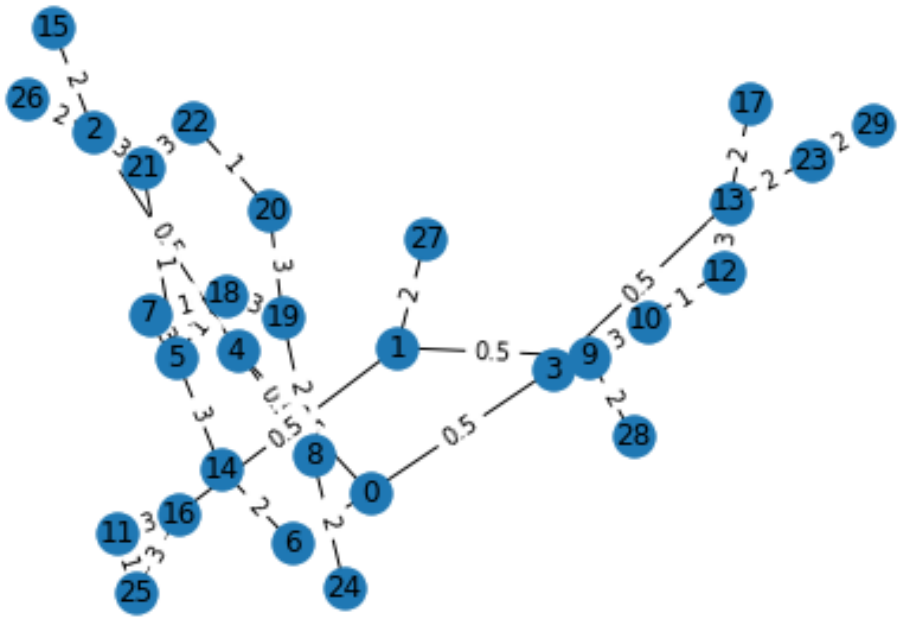


Figure 6: Example of a computer network structure generated by developed model

Stage 3. Assigning weights to edges that depend on nodes they connect – the more connections a node has, the lower the weight of edge connecting it (and accordingly, the higher the bandwidth of the corresponding communication channel).

Stage 4. Generating traffic packets for transmission. A random number of packets with random destinations is sent to each node with a certain probability. A device that received packets puts them in its internal queue. A traffic is generated with fractal properties [13]. Traffic generation is based on the theory of Markov processes, which is often used to model traffic in various queuing systems [16-20].

Stage 5. Testing routing algorithms. An algorithm for routing testing is selected. Traffic packets queued in network nodes are serviced using selected routing algorithm. The movement of packets through a network is modeled. If a packet does not fit in the queue of a node, it is lost. The model calculates all received and lost packets.

Stage 6. Completion of the model's work. The model stops when it reaches a certain number of iterations (for example, 1000 iterations), or if the model is working in the second mode, the stopping condition can also be when all queues are empty and all packets are either sent or lost.

To generate fractal binary traffic, a Markov chain was used, shown in Fig. 1.

In this work, to simulate network traffic, a binary time series was created, the persistent of which is regulated by the probabilities of changing the state to the opposite λ_1, λ_2 (Fig. 1).

This generator is characterized by states 0 or 1, and probabilities of being in these states are $p_0=\lambda_2/(\lambda_1+\lambda_2)$ and $p_1=\lambda_2/(\lambda_1+\lambda_2)$, where λ_i are the probabilities of the corresponding transitions [13]. The traffic intensity of such a generator will be in the range of [0, 1] and will be equal to the probability of obtaining "1" at the output of the generator: p_1.

The algorithm of such a generator is shown in Fig. 7.

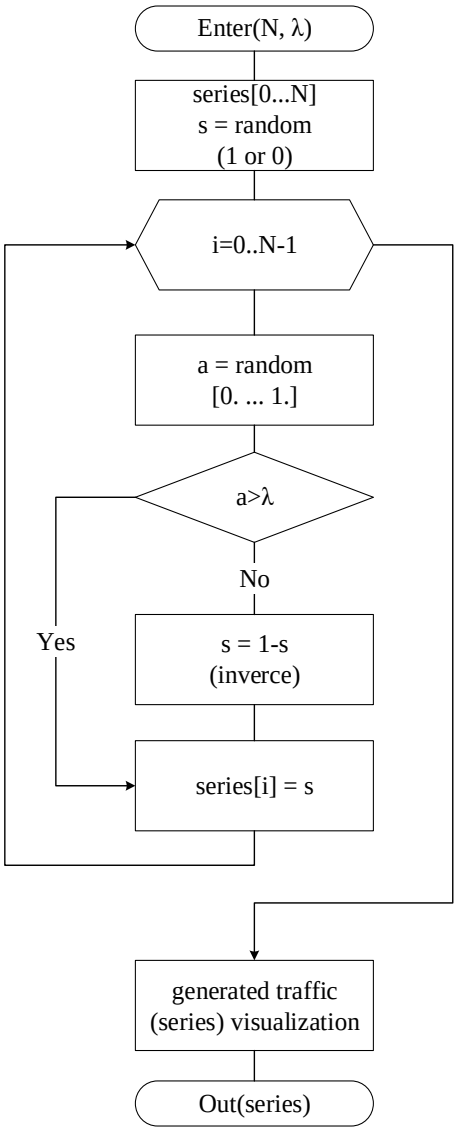


Figure 7: Algorithm for generating fractal binary traffic

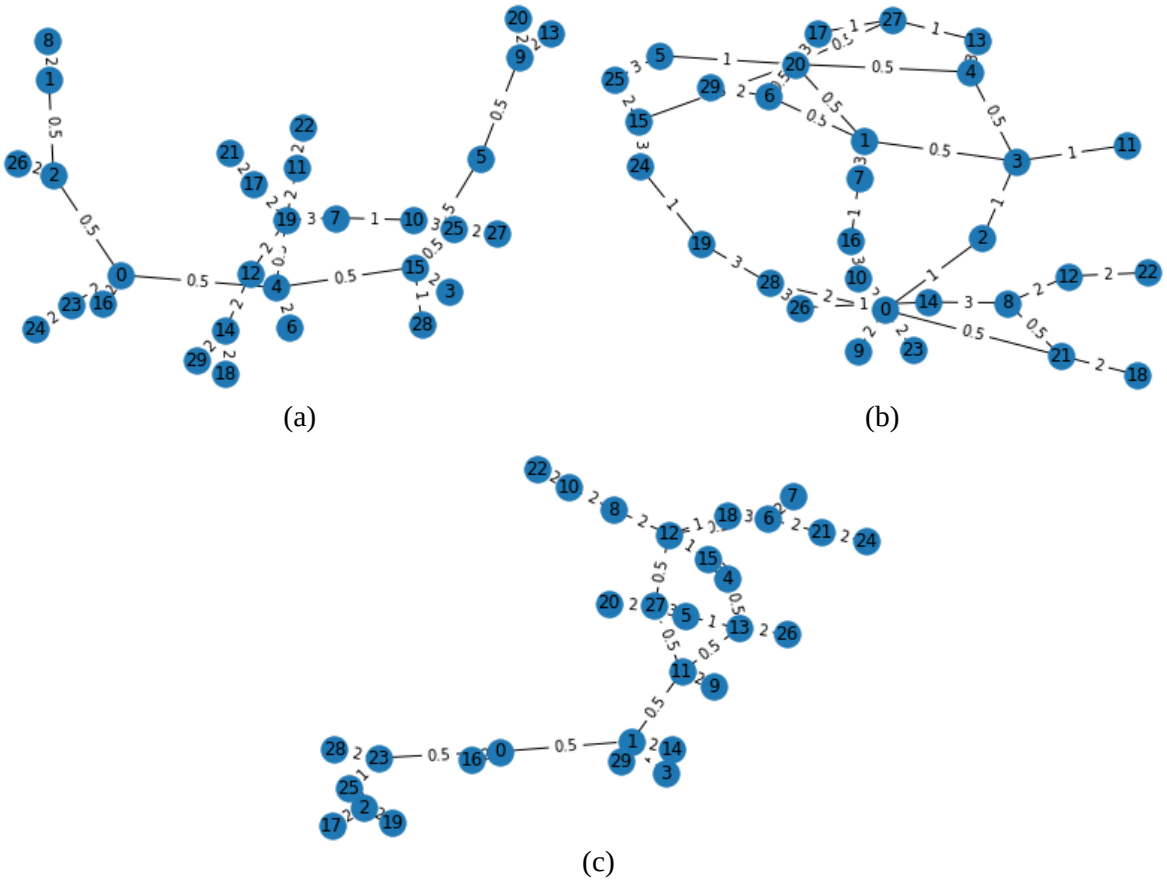
On developed model, OSPF routing algorithm was tested, which is based on link-state technology technology and uses Dijkstra's algorithm to find a shortest path. The obtained results showed the performance of developed model. In the future, on developed model, the authors will test the improvements of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

2.3. Experimental research of the quality of proposed method of routing fractal-like traffic

2.3.1. Experiment

A series of experiments were conducted on developed model to determine how the different fractal dimensions of traffic affect the probability of packet lost and therefore the quality of service at high traffic intensities. Three computer networks were generated, as shown in Fig. 3(a-c), each containing 30 routers with a queue length of 128 packets. The traffic intensity was set to 0,8, and the fractal dimension values used were 1,25, 1,50, and 1,75. The first mode of the model was used, where a random number of traffic packets with random sender and receiver devices were generated and routed on each iteration. The number of lost traffic packets was calculated over 1000 iterations of model time.

The results of experiments are presented in Table 1, which shows the average values based on the experiments conducted on the networks shown in Fig.8(a-c).



(a) (b)

(c)

Figure 8: Computer network structures generated for experiment

The results of the experiments are presented in Table 1. Computer networks with 30 routers were simulated during the transmission of packets between routers over a period of 1000 units of model time.

The following abbreviations were used in the table:

- WM1 – well-known OSPF routing method without predicting router load.
- WM2 – well-known OSPF routing method using traffic load prediction based on the moving average method.
- PM – proposed method for improving OSPF routing method, using the prediction of the probability of network packet loss due to router load based on fractal traffic analysis.

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

- values greater than 1,5 indicate an antipersistent process – any tendency tends to change in the opposite direction. The larger the fractal dimension, the stronger the antipersistent. As it approaches 1,5, the process becomes more random.
- value of 1,5 indicates a completely random process.
- values less than 1,5 indicate a persistent process, which means that it maintains its trend. The smaller the fractal dimension, the stronger the trend is maintained. As it approaches 1,5, the process becomes more random.

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

- at high network loads, the least packet loss occurs when the traffic is antipersistent, and the most occurs when the traffic is random or persistent.
- at high network loads, the proposed routing method significantly reduces packet loss, with significant improvements occurring for random and persistent traffic, and some improvement also occurring for antipersistent traffic, although it is not significant.

Table 1

Results of a series of experiments to determine the quality of developed method of routing fractal-like traffic and compare it with existing methods with a traffic intensity of 0,8

№	*Fractal dimension*	*Maximum number of packets that were sent to a router from its subnet per unit of time*	*Average number of lost packets per router in network in the series of experiments, %*		
			WM1	**WM2**	**PM**
1	1,75	55	00,00000	00,00000	00,00000
2	1,75	60	00,00000	00,00000	00,00000
3	1,75	65	66,50100	66,50100	65,33100
4	1,75	70	66,55133	66,51100	65,41733
5	1,75	75	66,56100	66,52800	65,49633
6	1,75	80	66,56533	66,55000	65,57067
7	1,75	85	66,57800	66,56167	65,64267
8	1,75	90	66,57867	66,56133	65,69200
		Average values:	**49,91691**	**49,90162**	**49,14375**
9	1,50	55	00,00000	00,00000	00,00000
10	1,50	60	00,00000	00,00000	00,00000
11	1,50	65	00,00000	00,00000	00,00000
12	1,50	70	00,00000	00,00000	00,00000
13	1,50	75	89,19267	87,13133	33,78667
14	1,50	80	99,52600	99,44533	94,94133
15	1,50	85	99,81133	99,78200	98,20633
16	1,50	90	99,84533	99,82200	98,60000
		Average values:	**48,54691**	**48,27258**	**40,69179**
17	1,25	55	00,00000	00,00000	00,00000
18	1,25	60	00,00000	00,00000	00,00000
19	1,25	65	00,00000	00,00000	00,00000
20	1,25	70	40,09400	36,29100	03,67200
21	1,25	75	97,75567	97,31167	77,17200
22	1,25	80	99,56233	99,49433	95,70900
23	1,25	85	99,78700	99,75400	97,95233
24	1,25	90	99,83700	99,81233	98,49200
		Average values:	**54,62950**	**54,08291**	**46,62466**

2.3.2. Results

Analyzing Table 1, the following conclusions can be made. Developed routing method in the conducted experiment with the given network parameters reduces the loss of network packets compared to the usual OSPF routing method without predicting the router's load:

- for antipersistent traffic, on average, by 1,03%, and a maximum of 1,17%;
- for random traffic, on average, by 7,85%, and a maximum of 55,41%;
- for persistent traffic, on average, by 8,00%, and a maximum of 36,42%.

And in comparison, with the method that also used load forecasting but by means of a moving average, proposed method also shows better results, namely fewer lost packets:

- for antipersistent traffic, on average by 1,01% and a maximum of 1,17%;
- for random traffic, on average by 7,58% and a maximum of 53,34%;
- for persistent traffic, on average by 7,45% and a maximum of 32,61%.

Also, analyzing Table 1, it can be seen that depending on the network parameters and traffic characteristics, the use of the proposed method can reduce network packet loss from 1,01 to 2,57 times.

2.3.3. Discussions

The poor results of the known method of load forecasting based on a moving average may be due to the fact that it does not take into account the fractal properties of traffic and its fractal dimension.

Thus, in routing traffic and finding optimal paths for sending IP-packets, it is useful to determine and take into account the fractal dimension of traffic at the input of each router and use it to predict the probability of packet loss and when calculating metrics to determine the best routes. Therefore, the proposed routing method significantly improves the quality of service in a computer network by reducing the probability of network packet loss.

3. Conclusions

The paper investigates the main principles of traffic routing in computer networks and how the fractal dimension of traffic affects the probability of queuing overflow in routers and loss of network packets. A computer model of a computer network was developed to test traffic routing algorithms. A method based on the theory of complex networks was developed to generate the structure of the computer network, while Markov processes and fractal time series were used to generate traffic. A method for routing fractal-like traffic with prediction of router congestion was also developed to reduce the probability of packet loss. The quality of proposed routing method was studied in a complex computer network with fractal traffic using a computer simulation model.

The results of experiments with proposed method on proposed computer model showed that:

- under heavy network load, the least lost packets occurred with antipersistent traffic, while the most lost packets occurred with random or persistent traffic.
- proposed routing method significantly reduces packet loss under heavy network load, particularly with a decrease of 1,03% on average for antipersistent traffic, 7,85% on average for random traffic, and 8,00% on average for persistent traffic compared to the standard OSPF routing algorithm. Moreover, depending on the network parameters and traffic characteristics, proposed method can reduce network packet losses from 1,01 to 2,57 times.

Therefore, proposed method of fractal traffic routing with prediction of router congestion can reduce the probability of network packet loss and improve the quality of service in a computer network.

4. References

[1] S. Dalal, B. Seth, V. Jaglan, et al., "An adaptive traffic routing approach toward load balancing and congestion control in Cloud-MANET ad hoc networks," Soft Comput, vol. 26, no. 10, pp. 5377-5388, 2022. doi: https://doi.org/10.1007/s00500-022-07099-4.

[2] L. Huang, M. Ye, X. Xue, and et al., "Intelligent routing method based on Dueling DQN reinforcement learning and network traffic state prediction in SDN," Wireless Netw., 2022. doi: https://doi.org/10.1007/s11276-022-03066-x.

[3] F.J. Suárez, P. Nuño, J.C. Granda, and D.F. García, "Computer networks performance modeling and simulation," in M.S. Obaidat, P. Nicopolitidis, and F. Zarai (Eds.), Modeling and Simulation of Computer Networks and Systems, Morgan Kaufmann, 2015, pp. 187-223. ISBN: 9780128008874. doi: https://doi.org/10.1016/B978-0-12-800887-4.00007-9.

[4] A. Sharma, R. Kumar, M. W. A. Talib, S. Srivastava and R. Iqbal, "Network modelling and computation of quickest path for service-level agreements using bi-objective optimization," International journal of distributed sensor networks, vol. 15, no. 10, 2019. doi: http://dx.doi.org/10.1177/1550147719881116.

[5] A.-L. Barabási, "Network Science," Cambridge University Press, 2018, 475 p. Available: http://networksciencebook.com/.

[6] A. Svyrydov, A. Kovalenko, and H. Kuchuk, "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, 2018. doi: https://doi.org/10.20998/2522-9052.2018.2.24.

[7] F. Tang, Z. M. Fadlullah, B. Mao and N. Kato, "An Intelligent Traffic Load Prediction-Based Adaptive Channel Assignment Algorithm in SDN-IoT: A Deep Learning Approach," in IEEE Internet of Things Journal, vol. 5, no. 6, pp. 5141-5154, Dec. 2018. doi: https://doi.org/10.1109/JIOT.2018.2838574.
[8] A. A. Turky and A. Mitschele-Thiel, "Use of load prediction mechanism for dynamic routing optimization," 2009 IEEE Symposium on Computers and Communications, Sousse, Tunisia, 2009, pp. 782-786. doi: https://doi.org/10.1109/ISCC.2009.5202245.
[9] S. Choi and D.-Y. Yeung, "Predictive Q-Routing: A Memory-based Reinforcement Learning Approach to Adaptive Traffic Control," in Advances in Neural Information Processing Systems, vol. 8, D. Touretzky, M.C. Mozer, and M. Hasselmo, Eds. MIT Press, 1995, pp. 945-951. Available: https://proceedings.neurips.cc/paper/1995/file/4e2545f819e67f0615003dd7e04a6087-Paper.pdf.
[10] S. Molnar and G. Terdik, "A general fractal model of Internet traffic," Proceedings LCN 2001. 26th Annual IEEE Conference on Local Computer Networks, Tampa, FL, USA, 2001, pp. 492-499. doi: https://doi.org/10.1109/LCN.2001.990828.
[11] D. Chakraborty, A. Ashir, T. Suganuma, G. Mansfield Keeni, T. K. Roy, N. Shiratori, "Self-similar and fractal nature of Internet traffic," Network Management, vol. 14, no. 2, pp. 119-129, 2004. doi: https://doi.org/10.1002/nem.512.
[12] W. E. Leland, W. Willinger, M. S. Taqqu, and D. V. Wilson, "On the self-similar nature of Ethernet traffic," SIGCOMM Comput. Commun. Rev., vol. 25, no. 1, pp. 202-213, Jan. 1995. doi: https://doi.org/10.1145/205447.205464.
[13] H. Drieieva, O. Drieiev, Ye. Meleshko, M. Yakymenko, and V. Mikhav, "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, 2022, pp. 1694-1707. Available: http://ceur-ws.org/Vol-3171/paper120.pdf.
[14] C. Ding, Y. Chen, Z. Liu, A. M. Alshehri and T. Liu, " Fractal Characteristics of Network Traffic and Its Correlation with Network Security," Fractals, vol. 30, no. 02, 2021. doi: http://dx.doi.org/10.1142/S0218348X22400679.
[15] G. Millán Naveas, E. San Juan Urrutia, and M. Vargas Guzmán, "A simple multifractal model for self-similar traffic flows in high-speed computer networks," Computación y Sistemas, vol. 23, no. 4, pp. 1517-1521, 2019. doi: https://doi.org/10.13053/cys-23-4-2831.
[16] P.-C.G. Vassiliou and A. C. Georgiou, "Markov and Semi-markov Chains, Processes, Systems and Emerging Related Fields," Mdpi AG, 294 p., 2021. doi: https://doi.org/10.3390/math9192490.
[17] A. Farahani, A Shoja. and H. Tohidi, "Markov and semi-Markov models in system reliability," In Engineering Reliability and Risk Assessment, Elsevier, pp. 91-130, 2023. doi: https://doi.org/10.1016/B978-0-323-91943-2.00010-1.
[18] Y. Zhang, D. Yue, L. Sun and J. Zuo, " Analysis of the Queueing-Inventory System with Impatient Customers and Mixed Sales," Discrete Dynamics in Nature and Society, vol. 2022, 2022. doi: https://doi.org/10.1155/2022/2333965.
[19] Ye. Meleshko, L. Raskin, S. Semenov, and O. Sira, "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, 2019. Available: https://www.scopus.com/record/display.uri?eid=2-s2.0-85078054250&origin=resultslist.
[20] Ye. Meleshko, O. Drieiev, M. Yakymenko and D. Lysytsia, "Developing a model of the dynamics of states of a recommendation system under conditions of profile injection attacks," Eastern-European Journal of Enterprise Technologies, vol. 4, no. 2(106), pp. 14-24, 2020. Available: https://www.scopus.com/record/display.uri?eid=2-s2.0-85096707995&origin=resultslist.
[21] M. Albahar, M. Thanoon, and A. Albahr, "The use of fractal dimension (FD) analysis in detection of anomalies, sabotages, and malicious acts in a cyber-physical system using Higuchi's algorithm," International Journal on Information Technologies & Security, vol. 14, no. 2, 2022.
[22] S. Gavrylenko, V. Chelak, O. Hornostal, and S. Gornostal, "Identification of the computer system state based on multidimensional discriminant analysis," in Proceedings of the 29th International Scientific Symposium Metrology and Metrology Assurance, Sozopol, Bulgaria, 2019. doi: https://doi.org/10.1109/MMA.2019.8936011.

[23] S. Gavrylenko, V. Chelak, and O. Hornostal, "Research of intelligent data analysis methods for identification of computer system state," in Proceedings of the 30th International Scientific Symposium Metrology and Metrology Assurance, Sozopol, Bulgaria, 2020. doi: https://doi.org/10.1109/MMA49863.2020.9254252.
[24] A. M. Al-Oraiqat, O. S. Ulichev, Ye. V. Meleshko, H. S. AlRawashdeh, O. O. Smirnov, L. I. Polishchuk, "Modeling strategies for information influence dissemination in social networks," Journal of Ambient Intelligence and Humanized Computing, vol. 13, no. 5, pp. 2463-2477, 2022. doi: https://doi.org/10.1007/s12652-021-03364-w
[25] M. Syfert, J. M. Kościelny, J. Możaryn, A. Ordys and P. Wnuk, "Simulation Model and Scenarios for Testing Detectability of Cyberattacks in Industrial Control Systems," in In Intelligent and Safe Computer Systems in Control and Diagnostics, 2022, pp. 73-84. doi: 10.1007/978-3-031-16159-9_7. Available: https://link.springer.com/chapter/10.1007/978-3-031-16159-9_7.
[26] S. Semenov, L. Zhang, W. Cao, S. Bulba, V. Babenko, and V. Davydov, "Development of a fuzzy GERT-model for investigating common software vulnerabilities," Eastern-European Journal of Enterprise Technologies, vol. 6, no. 2(114), pp. 6-18, 2021. doi: https://doi.org/10.15587/1729-4061.2021.243715.
[27] S. Semenov, O. Sira, S. Gavrylenko, and N. Kuchuk, "Identification of the state of an object under conditions of fuzzy input data," Eastern-European Journal of Enterprise Technologies, vol. 1, no. 4(97), pp. 22-30, 2019. doi: https://doi.org/10.15587/1729-4061.2019.157085. Available: https://core.ac.uk/download/pdf/288839756.pdf.
[28] S. N. Dorogovtsev, J. F. F. Mendes, "The Nature of Complex Networks," Oxford University Press, 2022.
[29] G. Cimini, T. Squartini, F. Saracco, D. Garlaschelli, A. Gabrielli and G. Caldarelli, "The statistical physics of real-world networks," Nature Reviews Physics, vol. 1, pp. 58-71, 2019. doi: https://doi.org/10.1038/s42254-018-0002-6.
[30] A. S. D. Mata, "Complex networks: a mini-review," Brazilian Journal of Physics, vol. 50, pp. 658-672, 2020. doi: https://doi.org/10.1007/s13538-020-00772-9.
[31] J. Aweya, "IP Routing Protocols: Link-State and Path-Vector Routing Protocols," 1st ed., CRC Press, 2021, 438 p.
[32] Cisco, "IP Routed Protocols," Technology Support, 2022. Available: https://www.cisco.com/c/en/us/tech/ip/ip-routed-protocols/index.html
[33] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," Science, vol. 286, no. 5439, pp. 509-512, 1999. doi: https://doi.org/10.1126/science.286.5439.509.
[34] Cisco, "Dynamic Routing Protocols," Cisco Press, 2001. Available: https://www.ciscopress.com/articles/article.asp?p=24090&seqNum=4.
[35] Cisco, "Understand Open Shortest Path First (OSPF) – Design Guide," Technology Support, 2022. Available: https://www.cisco.com/c/en/us/support/docs/ip/open-shortest-path-first-ospf/7039-1.html.
[36] P. R. Tadimety, "OSPF Messages," in OSPF: A Network Routing Protocol, Berkeley, CA, Apress, 2015. doi: https://doi.org/10.1007/978-1-4842-1410-7_18.
[37] B. Yu, "OSPF-Based Network Engineering Design and Implementation," in Informatics and Management Science VI, W. Du, Ed. London, Springer, 2013, vol. 209, pp. 131-138. doi: https://doi.org/10.1007/978-1-4471-4805-0_16.
[38] J. T. Moy, "OSPF: Anatomy of an Internet Routing Protocol," Addison-Wesley Professional, 1998. A. Verma and N. Bhardwaj, "A Review on Routing Information Protocol (RIP) and Open Shortest Path First (OSPF) Routing Protocol," International Journal of Future Generation Communication and Networking, vol. 9, no. 4, pp. 161-170, 2016.
[39] N. Jain and A. Payal, "Comparison Between IPv4 and IPv6 using OSPF and OSPFv3 on Riverbed Modeler," in 2019 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS), Goa, India, 2019, pp. 1-7. doi: https://doi.org/10.1109/ANTS47819.2019.9118101.