

# Multi-Criteria Synthesis of the Software-Defined Network Structure

Albert Voronin <sup>1</sup> [0000-0001-7201-2877], Maksym Kuklinskyi <sup>1</sup> [0000-0002-2028-9206],  
Tetyana Holyavkina <sup>1</sup> [0000-0003-2595-9405], Iryna Gyza <sup>1</sup> [0000-0002-8612-5538],  
Liudmila Kharlai <sup>2</sup> [0000-0002-7633-933X]

<sup>1</sup> National Aviation University, Kyiv, Ukraine,

<sup>2</sup> Kyiv College of Communication, Kyiv, Ukraine

alnv@ukr.net, maximum\_inc@ua.fm, holyavkina.t@gmail.com,  
irynagyza@gmail.com

**Abstract.** A software-defined network is a complex system, the basis of which, first of all, is formed by a central controller, switching nodes and data transmission channels ensuring the interaction of these nodes. In the process of designing such systems, special attention must be paid to its topological structure. This is due to several factors. Firstly, the issues of improving the efficiency and reliability of the entire network directly depend on the topological structure. Secondly, topology affects the number of factors when making all the decisions of its design. And finally, the topology completely defines the process of traffic management in a software-defined network, which occurs using a central controller that stores all the information about the network structure. For a comprehensive solution of the general problem of designing software-defined network structure and solving particular problems of optimizing design decisions during its synthesis, the article proposes optimization of system topology using the random search method and the tree structure method. The results of the proposed methods are compared and recommendations are given for combining these methods.

**Keywords:** Software-Defined Network, Data Transfer System, Topology, Optimization, Design.

## 1 Introduction

Like all modern technologies, software-defined networks (SDN) are constantly being improved, undergoing processes of modification and modernization. Taking into account the complexity of these technologies, improvement processes for objective reasons are quite lengthy. And although all of them are aimed at the possibility of exchanging various kinds of information, increasing its reliability, transmission speed and reliability of the network elements functioning, their common component is the topological structure of the system. Therefore, the question of the software-defined network structure optimal synthesis is relevant and quite popular.

Copyright © 2020 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0) CMiGIN-2019: International Workshop on Conflict Management in Global Information Networks.

## 2 Analysis of publications and goal setting

According to the range of services provided, software-defined networks belong to data transmission systems [1]. An analysis of the work showed that today, close attention is paid to the research of data transmission systems. This is confirmed by a large number of publications, and caused by the fact that this topic covers a very wide range of tasks in various fields. There are many publications that classify and disclose various ways and methods of data transfer [2-5]. But in most cases they are considered either in the context of generally accepted topologies of transmission systems, or without any description of the latter. There are also publications that are dedicated to the design of data transmission systems, but they mainly relate to certain industries, which narrows their scope, since they are focused on the characteristics of this industry only. Even fewer articles address the issue of optimizing data transmission systems [6], especially in the context of software-defined networks [7]. Therefore, the purpose of this article is to develop a method for optimizing the software-defined network structure formation, and assess its effectiveness.

## 3 Determining performance indicators of a software-defined network

When solving the problem of a software-defined network structure synthesizing, the following are used as performance indicators:

- reduced costs for the creation and operation of the network;
- characteristics of software-defined network structure reliability;
- the reliability of consumer information;
- average response time of network nodes to a request;
- average message transmission time;
- network congestion.

Consider those of the indicators that are most often used to assess the effectiveness of software-defined network.

The cost of software-defined network  $W_s$  is estimated by the quoted annual costs, which take into account both capital and operating costs. Topology of any information transmission system, and SDN is no exception, can be represented formally as a symmetric  $G$  – graph. Network elements form the vertices of the graph, and data transmission channels form the edges of the graph [6].

Evaluation of network topology effectiveness can be based on indicators that characterize the structure of the graph. Such indicators may be:

- redundancy of the structure  $R_s$ . This indicator determines excess of the total number of connections  $m$  over their minimum necessary number  $n-1$  (where  $n$  is the number of nodes), which ensures network connectivity.
- uneven structure of  $N_s$ . The exponent  $N_s$  determines quadratic deviation of connections distribution of given network from a uniform distribution.

- diameter of the structure  $D_s$ . This indicator determines maximum distance of network nodes.
- compact structure  $B_s$ . The indicator  $B_s$  characterizes general proximity of the nodes to each other.
- the degree of  $C_s$  structure centralization. This indicator displays relative number of connections that are established through the central controller.

Redundancy of  $R_s$  structure in a certain extent characterizes reliability, that is, one of the stability components and affects the efficiency of the SDN. At  $R_s = 0$  (star, tree topology) failure of any element (link or node) produces a partial or complete decay of the network. An increase in the  $R_s$  value reflects an increase in the reliability of SDN structure, however, a large redundancy (for example, complete interconnection) makes the network economically disadvantageous and difficult to implement.

Non-uniformity of  $N_s$  structure also characterizes reliability of SDN structure, but only from the point of view of the effect of individual elements failures. Structures with large values of  $N_s$  (star, ring tree, distributed star) are less reliable than with lower values of unevenness. At  $N_s = 0$  (ring, regular, complete interconnection) disconnection or failure of a structure node does not disable the network, but only reduces its performance [12-14].

Diameter of  $D_s$  structure and  $B_s$  structure compactness determine the average residence time of information in the network when it is sent between nodes. An increase in  $D_s$  and  $B_s$  values generally reflects an increase in the transmission time of messages in the network. Length of data transmission path in structures with large values of  $D_s$  and  $B_s$  (ring, tree, irregular) is longer than in structures with small values of  $D_s$  and  $B_s$  (complete relationship). Thus, indicators  $D_s$  and  $B_s$  characterize both the inertia of information processes and, to a certain extent, the survivability and noise immunity of SDN. The lower values of  $D_s$  and  $B_s$  are the more efficient network structure is, since its topology with such indicators has a shorter data transmission path and, consequently, a short residence time of the message in the transmission system. In addition, in the network structure of this type there are no communication channels, failure of which leads to the breakdown of the SDN into individual elements.

In general case, degree of  $C_s$  structure centralization indicates the presence or absence of a central node in networks, and this determines the control method (centralized, decentralized, and centralized-decentralized). Since the software-defined network is completely controlled from the central controller, the degree of centralization in the SDN affects its reliability, operating efficiency in terms of performance of its individual elements, as well as the software and technical complexity of the processes for managing all network resources.  $C_s$  values are close to one and equal to it (star, ring tree, tree) show that the systems structures are centralized, and failure of central node leads to complete destruction of network. Large degree of centralization in SDN is also associated with uneven distribution of the load among the system elements, which determines the necessity to create a central controller with high output. In the case of a network structure with  $C_s$  values close to zero or equal to zero (ring, regular, complete interconnection), network should spend a considerable part of time on or-

ganizing and managing the exchange of information between nodes, in other words, a software-defined network works like a regular peer-to-peer network.

Analysis of typical structures shows that  $R_s$ ,  $N_s$ ,  $D_s$ ,  $B_s$  and  $C_s$  characterize their survivability, reliability, performance, and cost. Therefore, using these indicators, it is possible to evaluate the network efficiency in a topological sense.

Considered indicators have different dimensions and are subject to various methods of extremization (some of them require minimization, and the rest require maximization). Therefore, it is necessary to bring them to a dimensionless form and to a single method of extremization, for example, to make them minimized. After these operations, we obtain normalized particular criteria of  $R_0$ ,  $N_0$ ,  $D_0$ ,  $B_0$ ,  $C_0$  effectiveness and  $W_0$  cost.

It is believed that these indicators equally characterize the components of efficiency, therefore, their weight differentiation is impractical. In this case, aggregated criterion of SDN topology effectiveness can be determined by the following formula:

$$F_0 = \frac{R_0 + N_0 + D_0 + B_0 + C_0}{5}, F_0 \in [0;1].$$

In the case of nonlinear compromise scheme [8], the following expression is used as the objective function:

$$\Phi = a_1(1 - F_0)^{-1} + a_2(1 + \varepsilon - W_0)^{-1}, a_1 + a_2 = 1,$$

where:  $a_1$  and  $a_2$  – weighting coefficients;  $\varepsilon$  – some sufficiently small value, which serves to avoid dividing by zero at  $W_0=1$ .

#### 4 Solving the problem of software-defined network structure synthesis

Consider two algorithms to solve the problem. This is optimization by random search and using the tree structure method [8, 9].

In the random search method, optimization occurs by generating a certain number of connected communication networks and choosing the best option from them. As additional settings for the optimization process are:

- minimum number of edges to be removed when generating a network  $R_{\min}$ ;
- maximum number of edges to be removed when generating a network  $R_{\max}$ ,
- number of iterations  $I_{\max}$ .

Number of iterations means number of networks that will be generated, and from which the best option will be selected. The larger value of this parameter, the higher results reliability, as the number of considered cases increases.

Minimum number of edges to be removed – allows you to initially exclude very strongly connected networks from consideration, if it is a priori known that there is no optimal solution among them. As this value increases, the maximum number of edges in the generated networks decreases, thus, connectivity of the networks decreases.

Maximum number of edges to be removed – at the initial stage allows to exclude from the consideration very weakly connected networks, if it is known that there is no optimal solution among them. When this value decreases, connectivity of the created networks increases [15-17].

Network generation is as follows. A fully connected network is created. After that, number  $R$  is randomly generated that sets number of edges removed from the network:

$$R_{\min} \leq R \leq R_{\max}.$$

After that it is randomly removed from the fully connected network of,  $R$  edges. It is controlled so that the network remains connected. For the network thus obtained, the calculation of the values of particular criteria is performed. After that, the generalized criterion is calculated using the nonlinear compromise scheme [8] taking into account the significance coefficients of the criteria. The result obtained is compared with the optimal solution received at the moment, and if it is better, it is taken as the optimal solution at this step. This procedure is repeated  $I_{\max}$  times.

Consider the description of the algorithm more detailed.

Step 0. Create a random network  $S_0$ . Set is as initial and optimal.  $S_{\text{OPT}} = S_0$ .  $\Phi_{\text{OPT}} = \Phi_0$ .

Step 1. While  $I < I_{\text{MAX}}$  repeat. Create a random network  $S_I$ . If  $\Phi_I < \Phi_{\text{OPT}}$ , then put  $S_{\text{OPT}} = S_I$ .  $\Phi_{\text{OPT}} = \Phi_I$ .

Step  $n$ . Output optimization result  $S_{\text{OPT}}$ ,  $\Phi_{\text{OPT}}$ .

#### **Random network** generation algorithm

Step 0. Create a fully connected network.

Step 1. Randomly determine the integer  $R$ :  $R_{\min} < R < R_{\max}$

Step 2. Repeat the procedure for removing a random edge  $R$  times. To remove an edge, it is randomly selected. If while deleting, the graph remains connected, then it is deleted and if not, then this edge is excluded from further consideration and another edge is selected for deletion.

Now consider the structure **tree method**.

In this case, optimization occurs by selecting the optimal edge for removal at each step of the iterative process.

As additional settings for the optimization process are:

- amount of iteration  $I_{\max}$ ;
- a sign of a stop in the absence of leafs of optimization tree with better performance than non-leaf nodes.

For optimization, a fully connected network is built, and then the edges are removed in accordance with the value of the generalized criterion. For optimization, a tree of optimal solutions is built. Its nodes are network options, and edges are the fact of removing one corresponding edge from the top-level network. Thus, the lower we move along the tree, the more edges are removed and the less remains. At each next step, the leaf vertex of the tree with the best value of the generalized optimization

criterion is selected, and all possible options for removing one edge from the selected network are considered. All the obtained options and values of the generalized criterion are entered in the optimization tree, and the selected vertex ceases to be leafy. Now the best leaf vertex is again selected and the process is repeated until all leaf vertices are exhausted, the iteration counter is exhausted, or (with the stop sign turned on) a situation does not occur when the optimal solution does not belong to the leaf node.

Algorithm can be interrupted at any iteration. In this case, we get the best result for a given number of iterations.

Consider an optimization algorithm.

Step 0. Create a fully connected network and set it as the root of the optimization tree  $S_0$ .

Step 1. Find in the optimization tree leaf with the best value of the generalized criterion (in the first step it will be the root of the tree – since it is the only node of the tree)  $S = S_1$ . Calculate all possible options for removing one edge from the network and enter the result into the optimization tree. Repeat this step until the counter reaches the limit of the number of iterations or there are no valid leaf nodes left.

## 5 Practical application of the proposed algorithms

Briefly consider the results obtained on a test example. First, we will set the initial data for optimization and the coordinates of the nodes Tables 1-2.

**Table 1.** Initial data

$K_{length}=50$	Conversion factor of distances from some conventional units to kilometers. Thus, coordinates of the nodes can be set in any unit of measurement (dimensionless)
$W_{cap} = 1000$	Cost of capital charges for the construction and operation of one kilometer of communication network (in c.u.)
$E_n = 1$	Return on investment (dimensionless)
$N = 20$	Number of network nodes (dimensionless)

**Table 2.** Nodes coordinates

Node number	Nodes coordinates	
	X	Y
1	9,5	9,7
2	10,1	14,1
3	10,6	16,7
4	10,2	15,2
5	10	12,9
6	9,9	10,9
7	8,7	11,3

8	7,7	8,8
9	7,7	6,6
10	9,1	7,7
11	6,9	4,7
12	6,8	4,2
13	8,6	3,4
14	6,2	5,3
15	7,9	5,6
16	13,4	4,4
17	17,4	4,8
18	11,5	2,2
19	14,8	5
20	13,4	4,1

After optimization by random search and tree structure methods, respectively, the following intermediate and final results of the experiment were obtained Table 3. In addition, the related nodes obtained by these methods are shown Tables 4-5.

**Table 3.** Comparison of optimization results

Description	Random search method	Tree method structure
Number of iterations	1000	200
Advanced options of methods settings	R <sub>min</sub> = 50 R <sub>max</sub> = 171	
Criteria values	Rs: 2,10526315789474 Ns: 0,00205331184815511 Ds: 3 Bs: 0,768421052631579 Cs: 0,214285714285714 Ws: 16944651,7100879 Kr: 0,766081871345029 Kn: 0,0020533118481551 Kd: 0,3 Kb: 0,180246913580247 Kc: 0,214285714285714 Kw: 0,274133975648503 Fs1: 0,292533562211829 Fs: 1,39557965832538 <b>Best Fs:</b> <b>1,39557965832538</b>	Rs: 2,78947368421053 Ns: 0,00143203681593325 Ds: 2 Bs: 0,621052631578947 Cs: 0,376623376623377 Ws: 12557937,8514079 Kr: 0,690058479532164 Kn: 0,0014320368159333 Kd: 0,2 Kb: 0,145679012345679 Kc: 0,376623376623377 Kw: 0,203164838561053 Fs1: 0,282758581063431 Fs: 1,32459769651696 <b>Best Fs:</b> <b>1,32417186843977</b>

**Table 4.** Connected nodes during random optimization

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1		+			+			+			+		+	+	+		+			
2	+			+	+			+					+							
3										+	+			+	+					+
4		+				+						+			+					
5	+	+							+	+										+
6				+						+				+	+		+			
7								+		+					+	+				+
8	+	+					+				+	+		+						+
9					+				+	+	+	+	+	+	+					
10			+		+	+	+		+				+	+		+				+
11	+		+					+	+				+						+	
12				+				+	+							+			+	
13	+	+							+	+	+				+		+			
14	+		+			+		+	+	+					+		+			
15	+		+	+		+	+		+				+	+		+				
16							+			+		+			+					
17	+					+							+	+				+	+	
18											+							+	+	
19												+						+	+	+
20			+		+		+	+		+									+	

**Table 5.** Connected nodes during optimization using the structure tree method

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1		+	+	+	+	+	+	+	+				+	+	+					+
2	+		+	+	+	+	+			+										
3	+	+		+		+				+										
4	+	+	+		+	+				+										
5	+	+		+		+	+			+										
6	+	+	+	+	+		+	+								+				
7	+	+			+	+		+		+										
8	+					+	+		+	+	+	+			+					
9	+							+	+	+	+	+	+	+	+					
10		+	+	+	+		+	+	+		+	+		+			+	+		+
11								+	+	+		+	+	+	+				+	
12								+	+	+	+		+	+	+				+	
13	+								+		+	+			+				+	+
14	+								+	+	+	+			+					
15	+							+	+		+	+	+	+		+				
16						+									+		+	+	+	+
17										+						+		+	+	+
18										+	+	+	+			+			+	+
19	+															+	+	+	+	+
20										+			+			+	+	+		

## 6 Conclusion

Thus, we see that the tree structure method provides the best indicator of efficiency, but requires large computational resources. In this regard, it seems interesting to combine both methods. A fully connected network a priori has a poor value of the optimality criterion and is far from the optimal solution. In this regard, it is inefficient to use it as a base. Thus, the random search method is effectively to be used to generate initial basic networks, and the tree structure method is to be used to improve their characteristics.

In this case, optimization is carried out in two stages. At the first stage, a random optimal network is built using the random search method, and then at the second stage it is used as the base (root) network for the structure tree method – with the help of which it is improved. Using this approach in software-defined networks will allow the central controller to significantly reduce its computing load, both in the case of the initial design of software-defined networks, and in the case of its further improvement.

## References

1. A.E. Kolomeets, L.V. Surkov, Software-defined SDN Networks – A Fundamentally New Approach to Networking, Modern computer systems and technologies: department collected works “Computer systems and networks” MSTU. named after N.E. Bauman, 2014, pp. 33-42.
2. Telecommunication systems and networks: Textbook: T.1 – Modern technologies / B.I. Kruk, V.N. Popantonolulo, V.P. Shuvalov; under the editorship of professor V.P. Shuvalova. – Ed. 3rd, fixed and added, Moscow: Hot line – Telecom, 2003, 647 p.
3. Marder N.S. Modern telecommunications, M.: IRIAS, 2006, 384 p.
4. Abilov A.V. Communication networks and switching systems: Textbook for universities, Moscow, Radio and communications, 2004, 288 p.
5. V.G. Olifer, N.A. Olifer, Computer networks. Principles, technologies, protocols, St. Petersburg. : Publishing house “Peter”, 2000, 672 p.
6. Gostev V.M. System for optimizing the design of data transmission networks, Memoirs of KNU: Physics and mathematics, 2007, Vol. 149. Book 2, pp. 35-48.
7. Mazin Al Hadidi, J. Samih Al-Azzeh, O. Tkalich, R. Odarchenko, S. Gnatyuk, Yu. Khokhlachova. ZigBee, Bluetooth and Wi-Fi Complex Wireless Networks Performance Increasing, International Journal on Communications Antenna and Propagation, Vol. 7, № 1, pp. 48-56, 2017.
8. Vetelina E. Study of development trends of modern network technologies using software-defined networks as an example, New information technologies in automated systems, № 18, 2015, pp. 502-508.
9. A.N. Voronin, Y.K. Ziatdinov, M.V. Kuklinskyi, Multicriteria solutions: models and methods: monograph, Kyiv, NAU, 2011, 348 p.
10. V.S. Mikhalevich, A.I. Kuksa, Methods of sequential optimization, Moscow: Nauka, 1983, 208 p.
11. Roman Odarchenko, Serhii Dakov, Olexandr Oksiuk and Larisa Dakova, Software-Controlled Network SDN Reliability Calculation 5th International Scientific-Practical

- Conference Problems of Infocommunications Science and Technology, PICS&T 2018, Conference Proceedings, pp. 99-103.
12. S. Gnatyuk, A. Okhrimenko, M. Kovtun, T. Gancarczyk, V. Karpinskyi, Method of Algorithm Building for Modular Reducing by Irreducible Polynomial, Proceedings of the 16th International Conference on Control, Automation and Systems, Oct. 16-19, Gyeongju, Korea, 2016, pp. 1476-1479.
  13. Al-Azzeh J.S., Al Hadidi M., Odarchenko R., Gnatyuk S., Shevchuk Z., Hu Z. Analysis of self-similar traffic models in computer networks, International Review on Modelling and Simulations, № 10(5), pp. 328-336, 2017.
  14. R. Odarchenko, V. Gnatyuk, S. Gnatyuk, A. Abakumova, Security Key Indicators Assessment for Modern Cellular Networks, Proceedings of the 2018 IEEE First International Conference on System Analysis & Intelligent Computing (SAIC), Kyiv, Ukraine, October 8-12, 2018, pp. 1-7.
  15. Z. Hassan, R. Odarchenko, S. Gnatyuk, A. Zaman, M. Shah, Detection of Distributed Denial of Service Attacks Using Snort Rules in Cloud Computing & Remote Control Systems, Proceedings of the 2018 IEEE 5th International Conference on Methods and Systems of Navigation and Motion Control, October 16-18, 2018. Kyiv, Ukraine, pp. 283-288.
  16. M. Zaliskyi, R. Odarchenko, S. Gnatyuk, Yu. Petrova. A.Chaplits, Method of traffic monitoring for DDoS attacks detection in e-health systems and networks. CEUR Workshop Proceedings, Vol. 2255, pp. 193-204, 2018.
  17. I. Parkhomey, S. Gnatyuk, R. Odarchenko, T. Zhmurko et al, Method For UAV Trajectory Parameters Estimation Using Additional Radar Data, Proceedings of the 2016 4th International Conference on Methods and Systems of Navigation and Motion Control, Kyiv, Ukraine, October 18-20, 2016, pp. 39-42.