INFORMATION TECHNOLOGY, COMPUTER SCIENCE, AND MANAGEMENT

UDC 004.722:519.172 https://doi.org/10.23947/2687-1653-2021-21-3-284-289

A method for generating a local network graph based on the analysis of address sets

V. V. Galushka ✉[1], **D. V. Fatkhi** [1], **E. R. Gazizov** [2]
[1] Don State Technical University (Rostov-on-Don, Russian Federation)
[2] Kazan Agricultural State University (Kazan, Russian Federation)
✉ galushkavv@yandex.ru

Introduction. The paper deals with the problem of automated construction of a local area network using tools and methods for traffic analysis at the link layer of OSI model. The problem is caused by two factors. These are difficulties of the manual determination of the communication between equipment and the lack of physical access to communication lines of an already functioning network. The purpose of the work is to reduce the time spent on building a local network diagram through automating the process of determining the communication between the equipment.
Materials and Methods. To solve the set tasks, a method for determining the relative location of devices is proposed. The network adapters of a specialized software and hardware complex, which are connected to a communication line break at different points of the network, are used in opposite directions. The method used is based on calculations of intersections of address sets received from these adapters. The structural schemes of the construction of such a software and hardware complex and the requirements for it are given. The methods of obtaining MAC addresses from transit packets are described. Examples of libraries of software components for performing this operation are given. The structure of a relational database is proposed for storing the received data. The format and content of the fields of its table are described.
Results. Using the developed methods, a typical example of an Ethernet network shows a way to determine the relative location of end devices specified by their MAC addresses, as well as at least two switches located between them. The signs by which it is possible to judge the presence of switching equipment in a particular segment are determined. A method is proposed that enables through using a set of relational operations, to sequentially refine the network topology until the required accuracy is achieved.
Discussion and Conclusions. The results obtained can be used under the administration of large local networks with an extensive structure. The proposed approach allows you to reduce the time required for building a scheme. This is possible due to the automation of the process of obtaining information about devices operating on the network and their mutual location.

Keywords: network topology, graph, tree, local network, traffic analysis, sets, relational operations.

***For citation*:** V. V. Galushka, D. V. Fatkhi, E. R. Gazizov. A method for generating a local network graph based on the analysis of address sets. Advanced Engineering Research, 2021, vol. 21, no. 3, pp. 284–289. https://doi.org/10.23947/2687-1653-2021-21-3-284-289

Introduction. Large local area networks are characterized by a complex configuration of physical connections, which to a great extent determines the efficiency of their work [1, 2]. In practice, the organization does not always have a detailed scheme or other documentation describing the network equipment and its interconnections. This significantly complicates the administration procedures and conditions the urgency of the problem of determining the structure of connections in the operated network for further construction of the layout and connection of nodes.

Communication lines are most often hidden behind the elements of the structure or decoration of the building, only switchgear is available. In this case, it is impossible to understand to which of the network nodes each connected

cable leads. Therefore, the task arises on constructing a network diagram based on the analysis of data obtained from the traffic captured at certain points. We are talking about places that are potentially available for connecting additional software and hardware that analyze traffic. The objective is to reduce the time spent on building a local network scheme.

The described task is complicated by the fact that all the information related to the functioning of the local network belongs to the second (data link) layer of the OSI model, and a significant part of the important data in the packet belongs to a higher level — the network [3, 4].

Most methods of traffic analysis are designed for processing network-level information [5]. In this regard, there is a need to develop methods that allow you to get all the necessary data for building a network diagram from the packet headers of the data link layer of the OSI model. On the other hand, network topologies at the data link layer are simpler than at the network layer, and they are always strictly regulated by the relevant standards[1].

Materials and Methods. The Ethernet standard, which is widely used for building local computer networks, provides for the use of the "tree" topology for organizing connections between nodes [5]. In graph theory, a "tree" is defined as a connected circuit-free graph [6]. An important consequence of this definition is that there is one and only one path between any pairs of vertices in the tree [7]. This allows you to abandon the search for routes within such a network and greatly simplify the operation of the equipment.

When constructing a graph, the set of its vertices and the connections between them is determined [8]. In relation to the network graph, vertices are network hardware. To address it within the local network, the MAC addresses assigned by the manufacturer are used. They are unique for each device and have 6 bytes in size [9]. The header of each network packet contains two MAC addresses: the sender and the recipient. They do not change during the transmission of a packet within the local network, and therefore in the problem under consideration, they can be used to identify network nodes.

When building a network graph, the major difficulty is determining the connections. Each connection links two vertices, whose relative location, as noted above, is unknown due to their great distance or hidden telecommunications routing. Connections can link devices of different types: switch-computer or switch-switch. The latter constitute the data transmission infrastructure and are of the greatest interest in terms of analyzing the network topology. In contrast, the connections of switchgear with computers describe the final vertices of the graph. At the same time, computers connected to the same switch can be conditionally combined into a group, since their mutual location relative to other computers will be the same. As a group, we can also consider larger sets of nodes, including computers connected to two or more nearby switches (i.e., those responsible for communication within one floor of a building or several offices of one department). In general, nodes that are part of a set should be located closer to each other than to nodes that are not part of the set or are part of another set [10]. At the first approximation, the entire local network can be considered as such a set, because its nodes are closely interconnected and separated from other networks [11].

The research idea is to consistently refine the network topology. For this, we will divide the set of MAC addresses of the devices included in it into smaller subsets, up to the definition of groups of computers connected to separate switches.

The division into subsets is performed relative to the points at which a hardware device capable of analyzing network packets and extracting address and other information from them is connected to the network. Such a device can be a laptop or a single-board computer that can work simultaneously with two network adapters. This will allow them to be connected to break the connection. As a result, a part of the network will be connected to each of the two network adapters.

It is important to note the difference between the terms "vertex" and "point". A vertex is a part of the network graph that denotes some equipment: a switch or an end device. A point is the connection point of the specified hardware complex, which is always located between two vertices.

Taking into account the linking of the analyzing device to the connection break, it is required to provide the operability of the communication line in which this break occurs. To that end, the network adapters must be connected through the operating system. A "bridge" type connection is used, when packets arriving at one of the interfaces are transmitted to the other one using the OSI model data link layer mechanisms, that is, without taking into account IP

[1] IEEE 802.3-2018 — IEEE Standard for Ethernet. IEEE Standard Association. standards.ieee.org. URL: https://standards.ieee.org/standard/802_3-2018.html (accessed: 11.04.2021).

addresses, routing, NAT, etc. This method of organizing the connection is completely transparent to other devices on the network, it does not change packets and does not manifest itself in any other way.

The main task of the device under consideration is to extract MAC addresses from transit packets. At this (first) stage of building a local network graph, traffic capture utilities (to write it into a file and analyze) or specialized libraries of software components (to analyze traffic in real time) are used [12, 13]. Depending on the operating system, the libraries may differ; however, as a rule, they are all based on Pcap (Packet Capture).

Regardless of the method of obtaining MAC addresses, information on them must be stored in the database. Taking into account the previously described features of the building a network diagram, we note the following requirements. For each MAC address, additional information is recorded:

— about the point to which the device that received the MAC address is connected;

— about the network interface from which the MAC address was received as the sender's address [14].

As a result, the database table will be described by relation A with the following scheme:

$$A\ (\underline{id}, address, point, side).$$

Here, *id* — primary key used only for identifying records in the table; *address* — MAC address of the device in the network extracted from the passing packet; *point* — network connection point (physical location); *side* — symbol of the network interface that transmitted the packet from which the MAC address was extracted.

After the formation of the MAC address database for a certain number of traffic capture points, the next stage begins — building a network diagram. It is based on information about the distribution of MAC addresses obtained for different connection points. Let us denote two arbitrary of them as p_1 and p_2. For each point, two sets of addresses should be received, each — from a separate network adapter. Let us denote X and Y — sets of addresses for point p_1, Z and V — sets of addresses for point p_2 (Fig. 1).

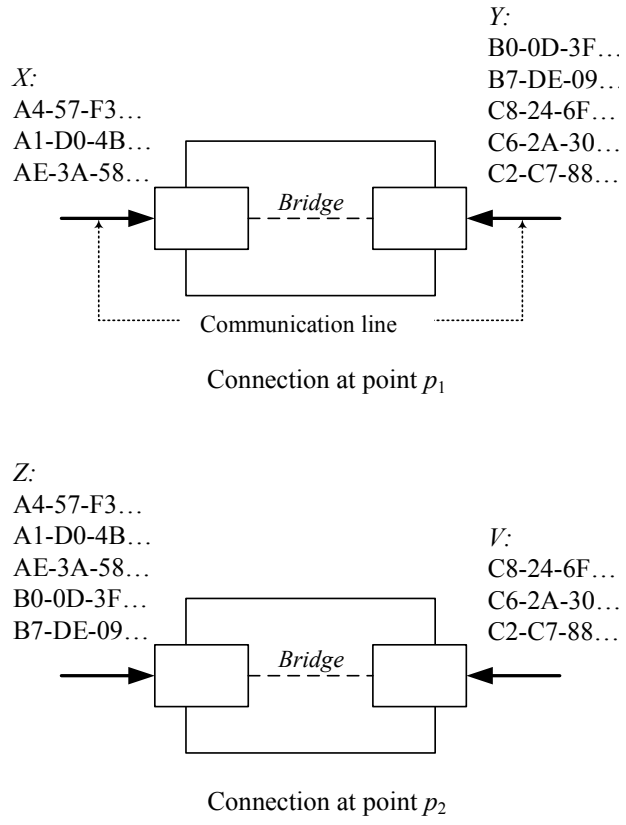

Fig. 1. Distribution of addresses across sets when connecting to different points of the network

From now on, only the first part is specified for the MAC addresses to shorten the record. Within the framework of the example under consideration, it is unique, and this is enough to reflect the work of the method.

Based on the obtained distribution of addresses across sets corresponding to different network interfaces, it is possible to draw initial conclusions about the mutual location of devices. To do this, you need to calculate all possible intersections for two points, that is, $X \cap Z$, $X \cap V$, $Y \cap Z$, $Y \cap V$.

It is advisable to calculate intersections by means of a database management system. This is because:

— information on the address belonging to a set is stored in the database,

— operations on sets are supported in relational algebra [15].

You need to execute queries equivalent to the following set of expressions:

$$X \cap Z = \Pi_{address}\,(\sigma_{point = 1 \wedge side = 1}\,(A)) \cap (\sigma_{point = 2 \wedge side = 1}\,(A))),$$
$$X \cap V = \Pi_{address}\,(\sigma_{point = 1 \wedge side = 1}\,(A)) \cap (\sigma_{point = 2 \wedge side = 2}\,(A))),$$
$$Y \cap Z = \Pi_{address}\,(\sigma_{point = 2 \wedge side = 1}\,(A)) \cap (\sigma_{point = 1 \wedge side = 1}\,(A))),$$
$$Y \cap V = \Pi_{address}\,(\sigma_{point = 2 \wedge side = 2}\,(A)) \cap (\sigma_{point = 1 \wedge side = 2}\,(A))).$$

Research Results. Let us consider an example of the application of the proposed methodology for constructing a network topology based on the distribution of sets of MAC addresses shown in Fig. 1. Determine the required intersections of the sets:

$$X \cap Z = \{\text{A4} - 57 - \text{F3}, \text{A1} - \text{D0} - 4\text{B}, \text{AE} - 3\text{A} - 58\},$$
$$X \cap V = \emptyset,$$
$$Y \cap Z = \{\text{B0} - 0\text{D} - 3\text{F}, \text{B7} - \text{DE} - 09\},$$
$$Y \cap V = \{\text{C8} - 24 - 6\text{F}, \text{C6} - 2\text{A} - 30, \text{C2} - \text{C7} - 88\}.$$

You can notice that one of the intersections (X and V) — is an empty set. This result is obtained for oppositely directed sides. Accordingly, the other sets (Y and Z), on the contrary, represent the sides directed at each other, and the result of their intersection is the addresses located between the measurement points, that is, between p_2 and p_1.

The remaining intersections represent the addresses located on opposite sides of the measurement points. X and V represent oppositely directed sides. Therefore, the remaining intersection, in which X (that is, $X \cap Z$) participates, includes addresses located on the side of point p_1, $Y \cap V$ — on the side of point p_1. Thus, it is possible to make an initial conclusion on the relative location of all the addresses obtained under the analysis, as well as on their location relative to the measurement points (Fig. 2).

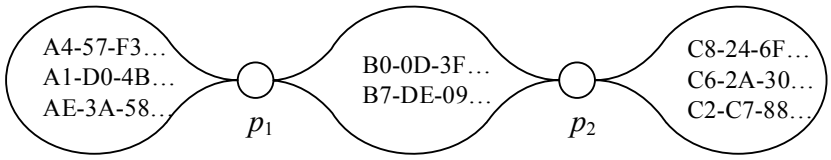

Fig. 2. Mutual arrangement of devices and points

It should be remembered that the points on this diagram are not network nodes (in particular, switches). However, the results obtained allow us to make the following assumption: if a set includes several addresses, it means that there is at least one switch inside it. We will substantiate the statement as follows: several computers cannot be connected directly; this requires appropriate network equipment (Fig. 3).

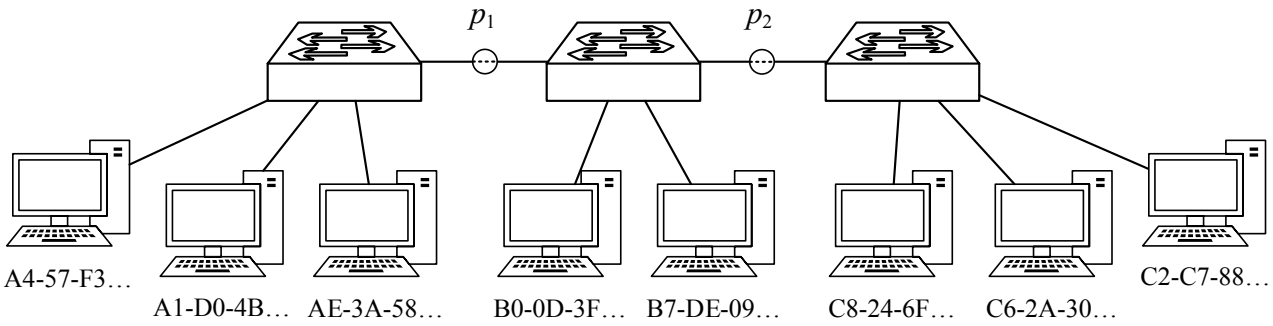

Fig. 3. Network scheme

The scheme shown in Fig. 3 is not final since there may be not one, but several switches inside each of the sets. At the next stages of the method, you should perform similar operations for each of the obtained sets, receiving MAC addresses at other points in the network. Each new measurement will provide refining the scheme and supplementing it with new switching nodes.

Discussion and Conclusions. A method for efficiently constructing a network scheme is proposed. The approach is based on an automated analysis of open information extracted from packets transmitted over the network. This technique is an alternative to the physical search for communication lines and the determination of the devices

connected by them. The application of the proposed solutions can significantly reduce the time spent by system administrators on determining the location of all devices and drawing them on the network diagram. The advantage of the method is the possibility of sequential refinement of the topology of network connections to obtain the required accuracy.

References

1. Kuz'menko NG. Komp'yuternye seti i setevye tekhnologii. St. Petersburg: Nauka i tekhnika; 2013. 368 p. (In Russ.)

2. Galushka VV. Seti i sistemy peredachi informatsii. Rostov-on-Don: DSTU Publ. House; 2016. 105 p. (In Russ.)

3. Stefano-Niko Orzen. Interaction understanding in the OSI model functionality of networks with case studies. IEEE 9th Int. SACI, 2014. P. 327–330. URL: www.researchgate.net/publication/269301474_Interaction_understanding_in_the_OSI_model_functionality_of_networks_with_case_studies (accessed: 18.08.2021). 10.1109/SACI.2014.6840086

4. Piyush Saxena. OSI Reference Model — A Seven Layered Architecture of OSI Model. International Journal of Research. 2014;1(10):1145–1156.

5. Lagutin IA. Opredelenie topologii s pomoshch'yu protokola LLDP v setyakh Juniper. Perspektivy razvitiya informatsionnykh tekhnologii. 2013;16:66–70. URL: https://cyberleninka.ru/article/n/opredelenie-topologii-s-pomoschyu-protokola-lldp-v-setyah-juniper/viewer (accessed: 10.04.2021). (In Russ.)

6. Alekseev VE, Talanov VA. Grafy i algoritmy. Struktury dannykh. Modeli vychislenii. Moscow: Binom. Laboratoriya znanii; 2012. 320 p. (In Russ.)

7. Ifenthaler D, Gibson D, Dobozy E. Informing learning design through analytics: Applying network graph analysis. Australasian Journal of Educational Technology. 2018;34(2):117–132. https://doi.org/10.14742/ajet.3767

8. Asel'derov ZM, Donets GA. Predstavlenie i vosstanovlenie grafov. Kiev: Naukova dumka; 2001. 96 p. (In Russ.)

9. Mao Yan, Kam-Yiu Lam, Song Han, et al. Hypergraph-based data link layer scheduling for reliable packet delivery in wireless sensing and control networks with end-to-end delay constraints. Information Sciences. 2014;278:34–55. 10.1016/j.ins.2014.02.006

10. Grigor'yan A. Introduction to Analysis on Graphs. Providence, Rhode Island: American Mathematical Society; 2018. 150 p.

11. Anduo Wang, Xueyuan Mei, Jason Croft, et al. Ravel: A Database-Defined Network. In: Proc. Symposium on SDN Research. 2016;5:1–7. URL: www.researchgate.net/publication/304918854_Ravel_A_Database-Defined_Network (accessed: 21.08.2021). https://doi.org/10.1145/2890955.2890970

12. Jiaqian Li, Chengrong Wu, Jiawei Ye, et al. The Comparison and Verification of Some Efficient Packet Capture and Processing Technologies. 2019 IEEE International Symposium on DASK, 2019. P. 967–973. URL: www.ieeexplore.ieee.org/abstract/document/8890423 (accessed: 18.08.2021). 10.1109/DASC/PiCom/CBDCom/CyberSciTech.2019.00177

13. Saavedra MZNL, Yu W. Towards Large Scale Packet Capture and Network Flow Analysis on Hadoop. In: Proc. 6th Int. Workshop on Computer Systems and Architectures; 2018. P. 186–189. URL: www.researchgate.net/publication/329905189_Towards_Large_Scale_Packet_Capture_and_Network_Flow_Analysis_on_Hadoop (accessed: 18.08.2021). 10.1109/CANDARW.2018.00043

14. József Marton, Gábor Szárnyas, Dániel Varró. Formalising openCypher Graph Queries in Relational Algebra. In: Proc. 21st European Conf. on Advances in Databases and Information Systems. 2015;10509:53–68. 10.1007/978-3-319-66917-5_13

15. Alekh Jindal, Samuel Madden, Malu Castellanos, et al. Graph Analytics using Vertica Relational Database. IEEE Xplore, 2015. P. 1191–1200. URL: www.ieeexplore.ieee.org/document/7363873 (accessed: 18.08.2021).

Received 26.07.2021
Revised 09.08.2021
Accepted 24.08.2021

About the Authors:

Galushka, Vasiliy V., associate professor of the Computer Systems and Information Security Department, Don State Technical University (1, Gagarin sq., Rostov-on-Don, 344003, RF), Cand.Sci. (Eng.), ORCID: http://orcid.org/0000-0003-2369-065X, galushkavv@yandex.ru

Fatkhi, Denis V., associate professor of the Information Technology Department, Don State Technical University (1, Gagarin sq., Rostov-on-Don, 344003, RF), Cand.Sci. (Eng.), ORCID: https://orcid.org/0000-0003-1538-1363, Zmey2257@mail.ru

Gaziziov, Evgeniy R., associate professor of the Physics and Mathematics Department, Kazan State Agrarian University (65, K. Marx St., Kazan, 420015, Republic of Tatarstan, RF), Cand.Sci. (Eng.), ORCID: http://orcid.org/0000-0002-9837-9850, pim.kazgau@mail.ru

Claimed contributorship

V. V. Galushka: basic concept formulation; research objective and tasks setting; developing a multi-address separation method. D. V. Fatkhi: development of a hardware complex for traffic capture; practical implementation of the proposed methods. E. R. Gaziziov: definition of the database structure and generation of relational set intersection operations.

All authors have read and approved the final manuscript.