UDC 004.3

# RESEARCH OF SDN NETWORK PERFORMANCE PARAMETERS USING MININET NETWORK EMULATOR

Oleksandr I. Romanov, Ivan O. Saychenko, Anton I. Marinov, Serhii S. Skolets

Institute of Telecommunication Systems
Igor Sikorsky Kyiv Polytechnic Institute, Kyiv, Ukraine

**Background**. The implementation of the new modern services on existing networks requires replacement or modernization of the old equipment. This negatively affects the efficiency of providing users with new types of services and their cost. Therefore, the issues/tasks of implementing SDN technology in the construction of networks are in the center of attention of representatives of science-research organizations and telecom operators. SDN networks have the advantages of flexible scaling of the system without replacing existing server and network equipment, eliminating operator dependence on specific vendor solutions, and rapid implementation of the new network technologies and services. To determine the numerical values of indicators of the functioning of SDN networks and check their compliance with the requirements, it is proposed to use in working process simulation modelling. For these goals, the Open Networking Foundation consortium developed the Mininet network emulator, the elements of that are published publicly. The formation of the simulation model and the procedure for determining the performance indicators of the SDN network that uses Mininet has a number of features that are considered in the work.

**Objective**. The purpose of the article is to build a simulation model based on the Mininet network emulator and to determine the performance indicators of SDN networks of various structures/topologies.

**Methods**. Creation of a SDN network segment for testing process of its functioning in the overload mode of various communication routes.

**Results**. The Mininet network emulator allows simulating SDN networks of a rather complex structure, change the performance of network branches and the amount of load in communication routes, to create a network-wide overload mode. During the simulation, it is possible to determine a number of indicators of network performance, such as the RTT (Round Trip Time) parameter for each route of communication, the bandwidth of branches and routes of communication, the amount of delay on network elements, loading of OpenFlow Switch ports/interfaces, network elements with the highest delay, the number of served and lost packets.

**Conclusions**. Use of the Mininet network emulator is a fairly convenient tool for determining the performance indicators of SDN networks. However, there are also some problems. First of all, setting up the system's operability requires writing programs to ensure the interaction among standard elements from the Mininet library. Secondone, is when congestion occurs in the communication routes, the Floodlight Controller does not automatically balance traffic along the work-around route. This task has to be solved manually by making changes to the OpenFlow Switch routing tables.

**Keywords**: Software Defined Networking; SDN; Mininet; OpenFlow; Floodlight Controller; OpenFlowSwitch; NFV; virtualization.

## INTRODUCTION

Existing networks are a collection of hardware and software equipment such as switches, routers, firewalls. These devices were created on the basis of specific hardware and software platforms from various vendors. Therefore, the introduction of new modern services on existing networks, as a rule, requires replacement or modernization of the old equipment staff [1,2,3,4]. This approach leads to the emergence of long design cycles, procurement of the necessary equipment and commissioning. All this negatively affects the efficiency of providing users with new types of services.

However, the networks of telecommunications operators today are mainly composed of “monolithic physical” network elements, where the control, administration and data transfer functions are performed by physical devices. Often, a telecom operator is forced to build its network using equipment from one manufacturer, since in this case it is easier to ensure the compatibility of network elements and to carry out upgrades. Deployment of services, modification (upgrades) of equipment or services is performed in turn in each network element and requires close internal and external resources of the operator. This approach makes the operator's network inflexible, complicates the introduction of new services and functions, and increases the operator's dependence on vendor solutions.

Therefore, the focus of representatives of research organizations, universities and mobile operators is on the issues of building networks based on SDN. The largest contribution to the development of this product comes from the Open Networking Foundation (ONF) consortium. ONF is a non-profit organization dedicated to accelerating the implementation of SDN and NVF [5].

To date, the ONF has developed documents that describe the principles of construction and operation of SDN networks. In works [7, 8, 9, 10] general requirements, system approaches and generalized architecture of SDN networks are considered. In papers [6, 11, 12, 15], the tasks and features of the protocols are considered, they are used to solve various problems in SDN networks. In works [13, 14, 16] described the functions of constructing elements of the SDN network and the order of their interaction in the process of streaming information. In works [17, 18, 19, 20, 21], the principles of constructing an optical transport network are considered and recommendations for ensuring the safety of their operation are given. The most complete and systematized material is presented in works [22, 23]. Articles [24, 25, 26, 27] explore various aspects of information flows and focus on meeting the requirements for ensuring network security.

It should be noted that in most of the works, special attention is paid to the implementation of individual elements of the SDN network. However, in the practical implementation of such a complex system as a SDN network, a comprehensive assessment of the performance indicators of the system as a whole is required. For this, various types of modelling are used to predict the behaviour of the system in various modes of operation.

## MAIN PART

To determine the numerical indicators of the functioning of complex systems and check them meet the requirements, three types of models are used: physical, analytical and simulation [1, 2].

Physical models assume the deployment of a network section on real equipment and practical operation in order to determine the performance characteristics. This method allows you to get values the closest to real values. However, this approach requires a lot of time and money.

Analytical models are easy to use and do not require large material investments. However, such complex systems as telecommunications are difficult to describe mathematically. Usually, when describing service processes, it is necessary to introduce a large number of restrictions. As a result, such models can have very low accuracy.

Therefore, as a rule, simulation models are used. They are less expensive than physical models, allow taking into account real network processes and have a fairly high variability of the parameters under study.

To study the performance of SDN networks using the simulation method, the Open Networking Foundation (ONF) has developed the Mininet network emulator. Elements of Mininet are available in the public domain on the developers' forum site [3, 4]. This allows you to use ready-made elementary fragments to create simulation models of SDN networks of various configurations. It should be noted that simulation models based on the MININET platform may not work immediately. They require writing matching programs and settings, the complexity of which depends on the type of structure of the network under study. However, to date, the MININET platform is the most deeply developed, it allows you to build models of networks of varying degrees of complexity, it is quite convenient to work with, and most importantly, it is in the public domain.

Let us consider the principle of constructing a SDN network model using the Mininet network emulator [5, 6]. Fig. 1 shows the structure of a data transmission network, which consists of standard switches and hosts (computers). Today, these elements independently solve both control tasks and data transmission tasks.

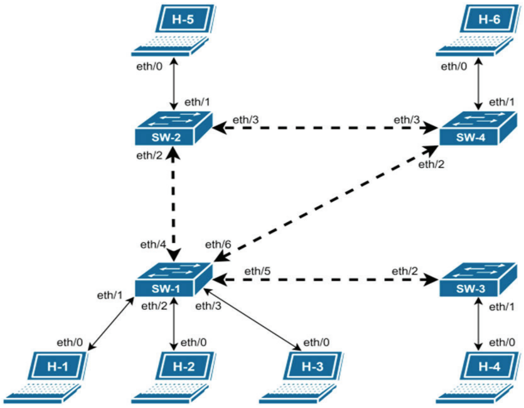

Fig. 1. Network structure emulated in Mininet

While the transition to SDN, the functional architecture of the network changes. The essence of the change is that in SDN the control plane is separated from the data plane. In this case, all control functions are transferred to the central SDN controller, and only the executive functions remain behind the data plane. Then, when switching to SDN principles, the functional architecture of the network on monolithic elements (Fig. 1) takes the form shown in Fig. 2.

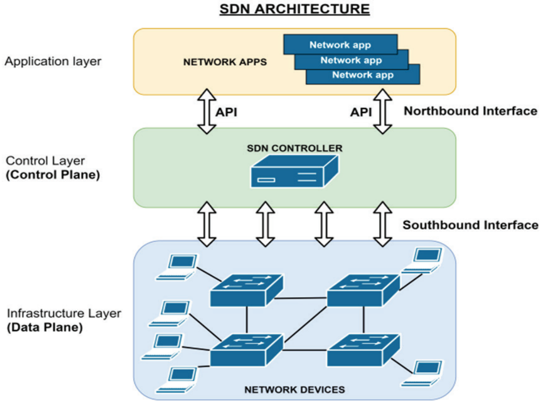

Fig. 2.SDN functional architecture

At first view, the network has become more complex due to the appearance of additional elements: a controller and functional blocks at the application level. But it must be borne in mind that the controller has taken over all the control functions. This greatly simplified the network devices at the data plane level, and there are a lot of them on the network. As for the functional blocks at the application level, they have always been. They were just not counted, as they were expected to provide additional services. A feature of functional blocks in the SDN network is that they can solve certain control tasks at the command of the controller.

The interaction of the controller with network elements is provided through the "southbound" interface using the open standard OpenFlow protocol. Today it is the main protocol recommended by the ONF for use in SDN networks.

An important advantage of SDN networks is that both hardware monolithic devices and virtual ones can be used as network elements. Fig. 3 shows an example where virtual vSwitch (for example, located in the Data Processing Center) are connected to the hardware WAN switches of the network (for example, the Inernet).

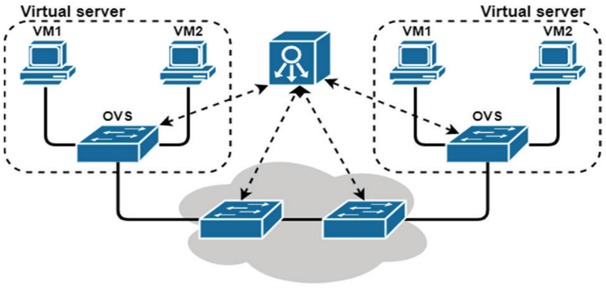

Fig. 3.Building an SDN network using hardware and virtual network elements

Each switch, hardware Switch or virtual vSwitch, contains one or more flow tables, a group table, and supports an OpenFlow channel to communicate with the SDN Controller (Fig. 4). Each flow table in OpenFlow Switch contains a set of flow entries and rules.

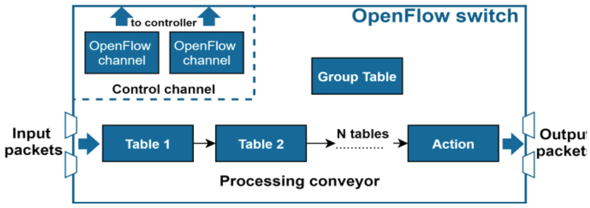

Fig. 4. Functional diagram of OpenFlow Switch

In Fig. 4 shows that a switch can use multiple forwarding tables. Tables are numbered starting from 0. The packet is sequentially transmitted for analysis across all tables. Once a package meets the selection criteria, a specific command is executed, such as a redirect. The forwarding entry indicates the port, which can be physical or virtual. There are special virtual ports that can change the logic of packet processing, for example, to provide that it is transmitted through all ports of the OpenFlow switch.

There are two modes of work switch :

1. Packet processing mode for data that already contain information about packet in the forwarding tables;

2. The mode of processing a packet that first entered the switch and for which there is no data in the forwarding tables.

The first processing mode is shown in Fig. 5.

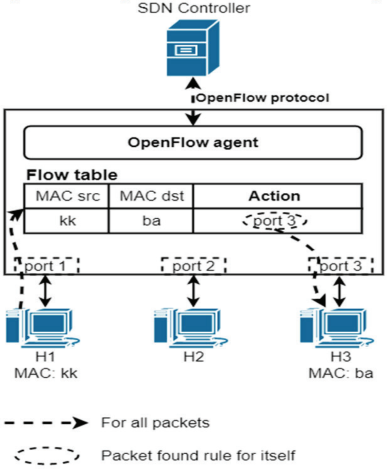

Fig. 5. Mode of processing packet for which all data contains in the Flow table.

The essence of the processing process is that a packet from the H1 host enters port 1 and then is transmitted for analysis to the Flow table. Checking the set of packet fields will allow you to determine the action - to transmit the packet to port 3, to which the H3 host is connected.

The processing mode of a packet that first entered the switch and for which there is no data in the forwarding tables is shown in Fig. 6.

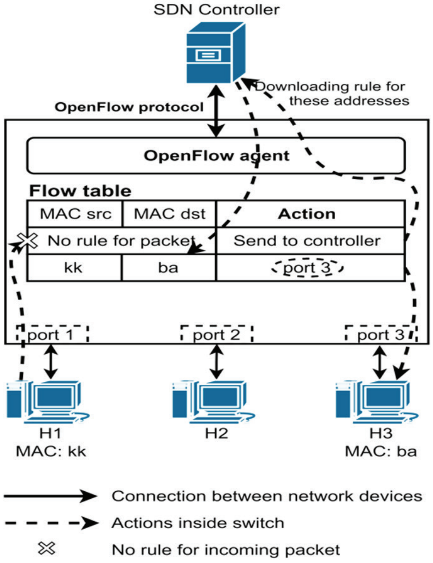

Fig. 6. Packet processing mode for which there is no data in the Flow table

The essence of the processing process is that a packet from the H1 host enters port 1 and then is transmitted for analysis to the Flow table. Checking the set of packet fields will not determine the port to which the packet is to be sent. Then, as a necessary action, the packet will be transferred to the OpenFlow agent, which will establish a secure communication channel with the SDN Controller and transfer the packet to it for analysis. SDN Controller will determine the destination port of the packet, make changes to the Flow table, and all subsequent packets of this type will be processed in the first mode. The packet processing algorithm in this mode is shown in Fig. 7.

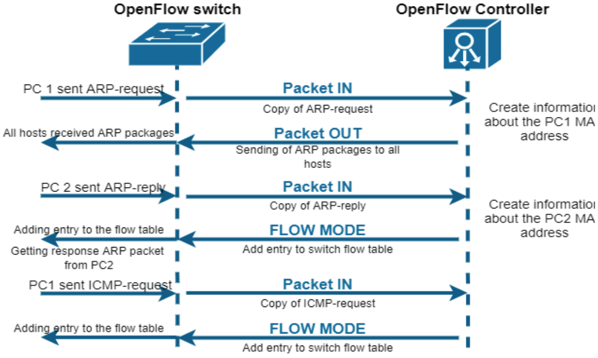

Fig. 7. Algorithm for processing a package for which there is no data in the Flowtable.

Let's make a simulation of the network, the structure of which is shown in Fig. 8, using the Mininet network emulator.

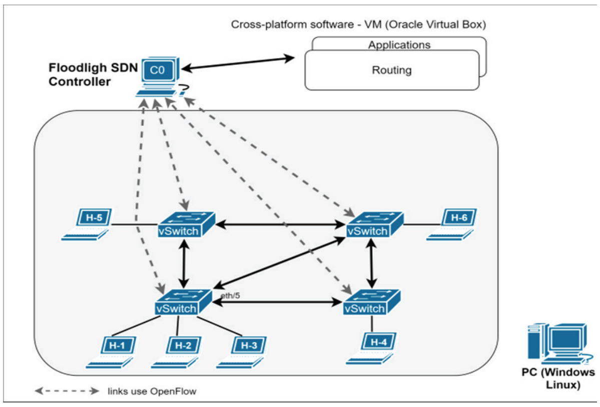

Fig. 8.The structure of the investigated network Mininet

Using Mininet, it is possible to explore SDN networks of various structures. The main advantages of Mininet are:

- fast network creation using standard components.
- the ability to build networks of complex structure;
- creation of virtual network elements such as OpenFlow Switch, web servers, monitoring tools, Wireshark;
- transfer and installation of configured network structures from Mininet to hardware switches;
- deploying a network in Mininet on a laptop, server, virtual machine or cloud.

There are three ways to build a simulation model using Mininet:

1. Simultaneously installation of the network structure and controller in one file. In this case, the controller has an embedded network.
2. Separate installation of the network structure and controller in different files. In this case, we have a controller, which is a remote device that is not built into the network and is launched when the network file is started.
3. Combined installation:
   - established network structure with built-in controller;
   - an additional Floodlight controller is installed and runs separately from the network.

The first method allows you to build only the simplest network structures. The second method provides the construction of more complex structures, however, it has limitations on the modes of operation of the network under study. The third method has more possibilities than the first two. It allows you to build large-scale network topologies. Floodlight controller is widely used in research in educational institutions and is actively developing [7, 8]. Therefore, the work used the third method for constructing the Mininet simulation model.

Floodlight controller was developed by an open community of developers who support the OpenFlow protocol. Modelling SDNs on the Mininet platform using the Floodlight controller has next set of advantages [1]:

- controller makes it easy to adapt and develop applications that are written in Java;
- there are included REST APIs, that simplify interface programming;
- the Floodlight website provides coding examples to help developers create own product;
- controller works with both physical and virtual switches which compatible with OpenFlow;
- controller can support networks on traditional switches which does not support OpenFlow.

The structure and main functional elements of the Floodlight controller are shown in Fig. 9.

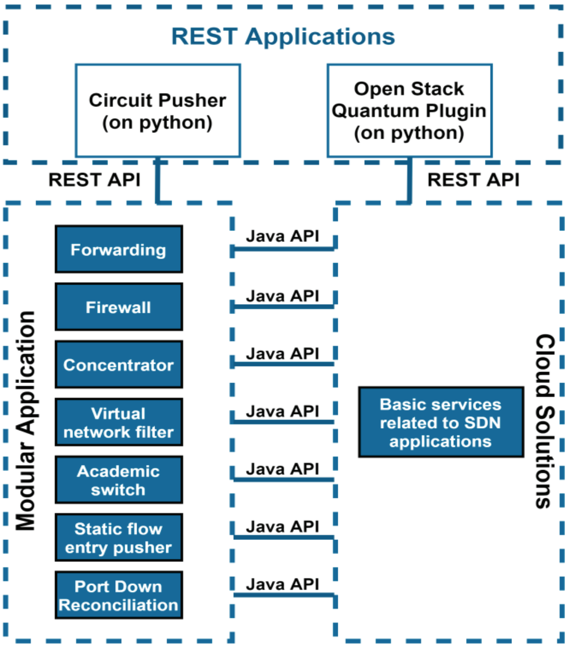

Fig. 9. Functional structure of the Floodlight controller

Now let's consider the purpose of the basic elements. The main component in the work of Floodlight Controller is REST Applications – programs that use services which use REST API, controlled by controller modules and modular applications [2, 3].

REST Applications in their turn interact with Circuit Pusher which from REST API provides management of packets of flows on all switches on the basis of IP-addresses with the set priority. Also REST Applications interact with modular applications that acting like the applications with higher bandwidth for communication with the controller.

Modular applications include applications such as: Firewall, Forwarding, Concentrator (Hub), Academic Switch, Entry Pusher, Virtual Network Filter, PortDown Reconciliation. The interaction of all these elements creates the controller's main functionality.

In the research, using the Mininet simulation model with the Floodlight Controller, were investigated the following performance indicators of the SDN network, which is shown in Fig. 8:

1. Network performance;
2. Bandwidth of network channels during transmitting UDP and TCP traffic packets;
3. Possibility of balancing traffic by the Floodlight controller in the overload mode of network paths;
4. RTT Packets delay values.

To discovery these parameters, there are required following input data:

1. Network structure that can be specified as a connectivity matrix;
2. Data transfer protocols that are used:
   - UDP;
   - TCP.
3. Amount of the load in the paths of communication.
4. Set performance of network branches.
5. Packet processing time in network elements.

As a result of modelling the process of network work, were determined the following indicators:

1. Network bandwidth depends on the size of the input load to the network.
2. Total real network branch performance.
3. Load on the interfaces of each network switch.
4. RRT for all communication routes for two types of load:
   - RTT for TCP packets;
   - RTT for UDP packets;

RTT represents the sum of the time it takes for delivery the package to the consumer and the time it takes to get confirmation that the packet has been received.

5. Impact of latency in network elements on bandwidth performance.

To generate traffic in the network, were used Iperf tool to generate packets over TCP and Iperfudp to generate packets over UDP.

The results of network modelling in Fig. 8 are presented in Table 1.

Table 1.Networkperformanceindicators

| Network route | Load | Bandwidth | Latency (ms) | Transmitted and lost packets |
|---|---|---|---|---|
| H1-H5 | 6 Mbps | 3.7 Mbps | 18.621 ms | 15201/5238 (34% loss) |
| H3-H5 | 4 Mbps | 2.4 Mbps | 1.787 ms | 10178/3741 (37% loss) |
| H4-H5 | 5 Mbps | 2.97 Mbps | 1.683 ms | 12674/4847 (38% loss) |

During the research, Host 5 was used as a server that receives traffic from three clients - Host 1, Host 3, Host 4. At the initial setup, the performance of all channels in the network was 10 Mbit / s. Further, the load in the network for each client increased until the network was overloaded.

For example, a branch of the network has a performance of 10 Mbps. The load in the communication route was increased for reaching the total incoming load in 15 Mbps. The goal was for traffic to be served not only along the shortest path, but also along the roundabout path. That is, there was a need to balance traffic. In our case, to prevent the occurrence of packet loss in the direction from sw-1 to sw-2, some of the sent packets were serviced in the transmission direction from sw-1 to sw-4. The obtained results are shown in Table 1.The visualization of the overloading process is shown in Fig. 10.

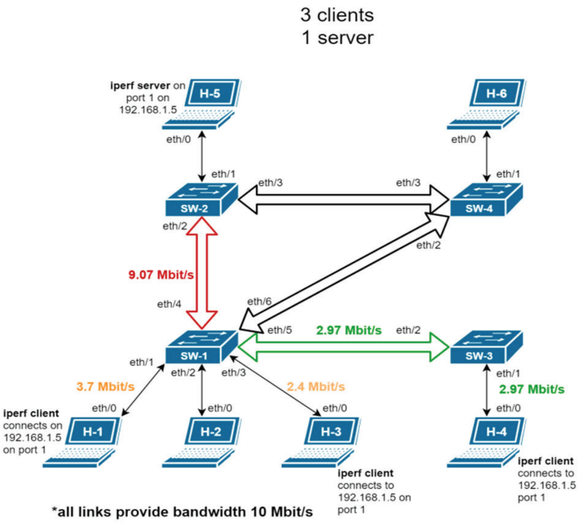

Fig.10.Network congestion (overload)

The Wireshark utility allows analysing the amount of traffic that went through the corresponding ports/interfaces. Fig. 11 shows an example of loading the eth4 interface of the switch sw-1 during traffic transmission to Host 5, which belongs to the switch sw-2.

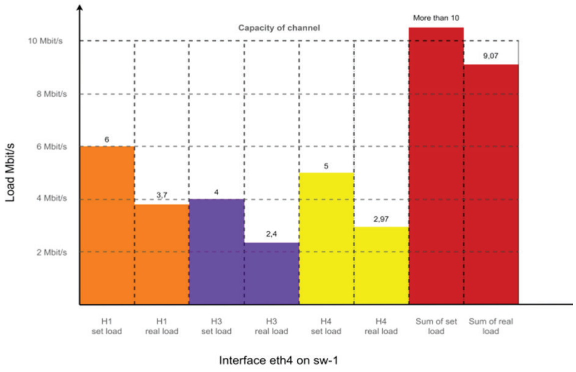

Fig. 11 Load on the eth4 interface of the switch sw-1

The total amount of transmitted and lost packets is shown in Fig. 12.

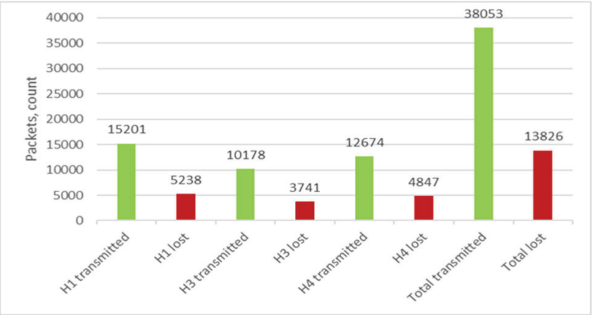

Fig. 12 Packet transfer statistics between clients and server

Next, the value of the delay of RTT packets was researched [4].

The *pingall* tool is responsible for determining the RTT parameter. But to get more detailed statistics use the command: *pingallfull*.

The results of the RTT research are shown in Table2.

Table 2.Theresultsofthe RTT research

| Delay, ms | Average RTT, ms | Number of working routes |
|---|---|---|
| 0 | 0.146 | 30 |
| 2.5 | 22.266 | 30 |
| 5 | 38.578 | 30 |
| 50 | 308.414 | 30 |
| 100 | 909.721 | 30 |
| 200 | 1792.005 | 30 |
| 400 | 3358.825 | 28 |
| 450 | 3693.478 | 27 |
| 500 | 3131.509 | 18 |

Visualization of the obtained results is presented in Fig.13 and Fig. 14.

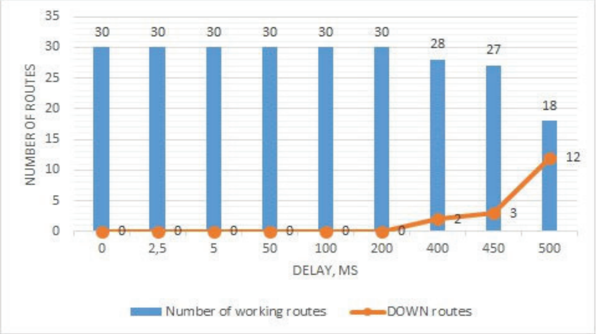

Fig. 13. Dependence of the number of working and denied communication routes with different delays in the network

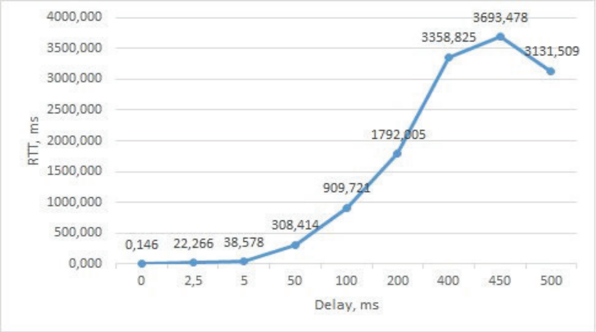

Fig. 14. Dependence of RTT change at different network delays

As can be seen from Fig. 14, when packets with a set delay of 400 ms are transmitted in the network, the first disconnections of the routes begin to occur. Namely, 2 directions with the largest number of hops, with 30 directions in the network (where hop is the distance between two nodes in the network, so the more hops – the more complicated the routing path and the further are the nodes from each other).

According to Fig. 14, you can see a decrease in RTT with a delay of 450 ms, due to the failure of 3 routes. With a delay of 500 ms, we see a much larger number of rejections, namely 12 routes.

Also, as a result of the experiment during seting up all the above delays (Table 2), we noticed rutes with the largest delays: h1-h4, h1-h5, h1-h6, h2-h4, h2-h5, h2- h6, h3-h4, h3-h5, h3-h6, h4-h5, h4-h6, h5-h4, h5-h6, h6-h4.

Thus, the use of Mininet allows you to model a fairly complex SDN network and get important performance indicators, such as bandwidth, quality of service, packet delay using different protocols, such as TCP, UDP and others. Mininet is publicly available. Therefore, it deserves a more detailed study.

## REFERENCES

1. Romanov O.I., Nesterenko M.M. Telecommunication network technologies Evolution//Advances in the telecommunications 2019/ According to scientific editorship IlchenkoM.Yu., Kravchuk S.O., Kyiv, 2019, - p.p. 88-107.
2. Oleksandr I. Romanov, Mykola M. Nesterenko, Nadiia O. Fesokha, Volodymyr B. Mankivskyi. Evaluation of productivity virtualization technologies of switching equipment telecommunications networks. Information and Telecommunication Sciences, 2020, Volume 11, Number 1 (20), page 53 – 58, DOI: https://doi.org/10.20535/2411-2976.12020.53-58.
3. Romanov A.., Nesterenko M., Mankivskyi V, Saichenko, I. (2020). Modelofoptimalloaddistributioninmobileoperatoraccessnetwork.BulletinofCherkasyStateTechnologicalUniversity, (3), 20–29. https://doi.org/10.24025/2306-4412.3.2020.200346
4. Oleksandr I. Romanov, Mykola M. Nesterenko, Volodymyr B. Mankivskyi. Used regress model coefficient utilization of channels for plan load distribution in network. Visnyk NTUU KPI Seriia – RadioteknikaRadioaparatobuduvannia№67, 2016p.p. 34-42
5. Open Network Foundation. Accelerating the Adoption of SDN & NFV. http://opennetworking.wpengine.com/wp-content/uploads/2013/03/ONF%20Overview%204-pager%20FINAL.pdf, 2021
6. ONF. OpenFlow Switch Specification Version 1.5.1 ( Protocol version 0x06 ), 2014.https://opennetworking.org/wp-content/uploads/2014/10/openflow-switch-v1.5.1.pdf
7. ONF TR-502. SDN architecture. Issue 1 June, 2014
8. ONF TR-508. Requirements Analysis for Transport OpenFlow/SDN. V1.0 August 20, 2014. ONF TR-508
9. ONF specification. OpenFlow Table Type Patterns. Version No. 1.0 15 August 2014
10. IETF RFC 7426. Request for Comments: 7426. ISSN: 2070-1721 EICT. Category: Informational K. Pentikousis, Ed.,
11. ISSN: 2070-1721 EICT
12. https://tools.ietf.org/html/rfc7426
13. ONF TR-525 SDN Interoperability Event Technical Issues Report AppFest 2015
14. Testing-Interop Working Group Version 0.2
15. O.I. Romanov; M. V. Oryschuk; Y. S. Hordashnyk. Computing of influence of stimulated Raman scattering in DWDM telecommunication systems. 2016 International Conference Radio Electronics & Info Communications (UkrMiCo) Year:2016.DOI:10.1109/UkrMiCo.2016.7739622,eISBN:978-1-5090-4409-2
16. Globa L., Skulysh M., Romanov O., Nesterenko M. (2019) Quality Control for Mobile Communication Management Services in Hybrid Environment. In: Ilchenko M., Uryvsky L., Globa L. (eds) Advances in Information and Communication Technologies. UKRMICO 2018. Lecture Notes in Electrical Engineering, vol 560. Springer, Cham. DOI https://doi.org/10.1007/978-3-030-16770-7_4, Print ISBN978-3-030-16769-1
17. ONF TR-539 OpenFlow Controller Benchmarking Methodologies Version 1.0 November 2016
18. ONF TS-029. MPLS-TP OpenFlow Protocol Extensions for SPTN. Version 1.0 June 16, 2017
19. ONF TR-539. OpenFlow Controller Benchmarking Methodologies Version 1.0 November 2016
20. ONOS. Security and Performance. Analysis: Report No. 1. Date: September 19, 2017
21. ONOS Security and Performance Analysis (Report No. 2). Date: Nov. 2, 2018
22. ONF TR-537. Negotiable Datapath Model and Table Type Pattern Signing Version 1.0. ONF TR-537 2016-09-08
23. Oleksander Romanov,VolodymyrMankivskyi. Optimal Traffic Distribution Based on the Sectoral Model of Loading Network Elements /2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T), 8-11 Oct. 2019, Date Added to IEEE Xplore: 09 April 2020, Conference Kyiv, Ukraine.DOI:10.1109/PICST47496.2019.9061296, eISBN: 978-1-7281-4184-8.
24. Romanov, O., Nesterenko, M., Veres, L., Kamarali, R., Saychenko, I. Methods for calculating the performance indicators of ip multimedia subsystem (IMS). Lecture Notes in Networks and Systems, 2021, 152, c. 229-256
25. Next-Gen SDN Tutorial, 2019, https://opennetworking.org/wp-

content/uploads/2019/10/NG-SDN-Tutorial-Overview.pdf

26. Carmelo CasconeO'Connor, Thomas Vachuska, Bruce Davie. Software-Defined Networks: A Systems Approach, – 2021, p.152

27. K´evinPhemius, Mathieu Bouet and J´er´emieLeguay. DISCO: Distributed Multi-domain SDN Controllers. Thales Communications & Security. arXiv:1308.6138v2 [cs.NI] 29 Aug 2013

28. JunHuyLam,Sang-GonLee,Hoon-Jae Lee, and YustusEkoOktian. Securing SDN Southbound and Data Plane Communication with IBC. Hindawi Publishing Corporation Mobile Information Systems Volume 2016, Article ID 1708970, 12 pages http://dx.doi.org/10.1155/2016/1708970.

29.DouglasComer, AdibRastegarnia. Externalization of Packet Processing in Software Defined Networking. All content following this page was uploaded by Adib Rastegarniaon 11 January 2019. https://www.researchgate.net/publication/330276437

***Романов О.І., Сайченко І.О., Марінов А.І., Сколець С.С.***

## Дослідження параметрів продуктивності мережі SDN з використанням мережевого емулятора MININET

**Проблематика.** Принципи побудови і функціонування мереж SDN суттєво відрізняються від традиційних телекомунікаційних мереж. Для визначення чисельних значень показників функціонування такого типу мереж та перевірки їх відповідності вимогам, консорціумом OpenNetworkingFoundation розроблений мережевий емулятор Mininet. Елементи мережевого емулятора є у відкритому доступі. Однак, побудова імітаційної моделі і порядок визначення показників функціонування мережі SDN з використанням Mininet має ряд особливостей,які розглянуті в роботі.

**Мета дослідження**. Створення імітаційної моделі на базі мережевого емулятору Mininet і визначення показників функціонування мереж SDN різної структури.

**Методика реалізації.** Визначення показників функціонування мережі SDN у режимі перевантаження напрямків зв'язку.

**Результати дослідження.** Проведено моделювання мереж SDN складної структури і визначені наступні показники функціонування: RTT (RoundTripTime) для кожного напрямку зв'язку, пропускна здатність гілок і напрямків зв'язку, величина затримки на мережевих елементах, завантаження портів OpenFlowSwitch, елементи мережі з найбільшою затримкою, число обслугованих і втрачених пакетів.

**Висновки.** Використання мережевого емулятора Mininet є досить зручним інструментом для визначення показників функціонування мереж SDN. Однак, є і проблеми. По-перше, налаштування працездатності системи вимагає написання програм для забезпечення взаємодії стандартних елементів з бібліотеками Mininet. По-друге, при виникненні перевантажень в напрямках зв'язку, FloodlightController не забезпечує автоматичного балансування трафіку по обхідним шляхам. Цю задачу доводиться вирішувати в ручному режимі шляхом внесення змін до таблиці маршрутизаціїOpenFlowSwitch.

**Ключові слова**: Програмно-конфігуровані мережі; SDN; Mininet; OpenFlow; Floodlight контролер; OpenFlow комутатор; NFV; віртуалізація.

***Романов А.И., Сайченко И.О., Маринов А.И., Сколец С.С.***

## Исследование параметров производительности сети SDN с использованием сетевого эмулятора MININET

**Проблематика.** Принципы построения и функционирования сетей SDN существенно отличаются от традиционных телекоммуникационных сетей. Для определения численных значений показателей функционирования такого типа сетей и проверки их соответствия требованиям, консорциумом Open Networking Foundation разработан сетевой эмулятор Mininet. Элементы сетевого эмулятора имеются в открытом доступе. Однако, построение имитационной модели и порядок определения показателей функционирования сети SDN с использованием Mininet имеет ряд особенностей, которые рассмотрены в работе.

**Цель исследования.** Создание имитационной модели на базе сетевого эмулятор Mininet и определения показателей функционирования сетей SDN различной структуры.

**Методика реализации.** Определение показателей функционирования сети SDN в режиме перегрузки направлений связи.

**Результаты исследования.** Проведено моделирование сетей SDN сложной структуры и определены следующие показатели функционирования: RTT (RoundTrip Time) для каждого направления связи, пропускная способность ветвей и направлений связи, величина задержки на сетевых элементах, загрузка портов OpenFlowSwitch, элементы сети с наибольшей задержкой, число обслуженных и потерянных пакетов.

определения показателей функционирования сетей SDN. Однако, есть и проблемы. Во-первых, настройка работоспособности системы требует написания программ для обеспечения взаимодействия стандартных элементов с библиотеками Mininet. Во-вторых, при возникновении перегрузок в направлениях связи, FloodlightController не обеспечивает автоматической балансировки трафика по обходным путям. Эту задачу приходится решать в ручном режиме путем внесения изменений в таблицы маршрутизации OpenFlowSwitch.

**Ключевые слова**: Программно-конфигурируемые сети; SDN; Mininet; OpenFlow; Floodlight контроллер; OpenFlow коммутатор; NFV; виртуализация.