ISSN: 2350-0328
International Journal of AdvancedResearch in Science, Engineering and Technology
Vol. 12, Issue 4, April 2025

Analysis of network simulation and emulation tools with OpenFlow

Mukhammedinov K.K.

Assistant Professor, Department of Info-communication Engineering, Tashkent University of Information Technologies named after Muhammad al-Khwarizmi, Tashkent, Uzbekistan

ABSTRACT: The development of technology is making the use of simulation tools common practice in conducting experiments and analyses in every field. Network simulation tools are widely used, especially in the telecommunications industry. For researchers conducting scientific work, working with real equipment can pose a number of challenges, including financial constraints and the inability to replicate experiments. Using simulation tools provides some relief by allowing you to experience and mimic real reality, while emulation tools allow you to conduct experiments on more real devices. This allows us to work with results that are very close to the results of real network experiments. There are several emulators/simulators that support the OpenFlow protocol. There are several scientific studies that analyze these tools. But as these platforms continue to be updated, using previously released source data may lead to some inconsistencies. This work analyzes several updates in this regard.

KEY WORDS: Software defined networking, OpenFlow, Simulation and Emulation tools.

I. INTRODUCTION

Exploring the capabilities of a real network requires the availability of a very large amount of equipment, which inevitably leads to high costs. Analysis using analytical calculation methods can also be quite different from reality. For this reason, researchers in this field often use simulation methods. This helps to achieve results that are closer to reality [1].

There are a large number of emulation/simulation systems that allow you to replicate network performance, but most of them do not offer ready-made modules for working with SDN [2]. Most systems offering solutions for SDN modeling focus on the controller and do not pay enough attention to the operation of OpenFlow switches [33].

There are several emulation and simulation tools that support OpenFlow, and there has been much research on which of them to use and the performance obtained when used in scientific projects. Researchers studying the hybrid SDN paradigm require Simulation tools, Emulation tools, and testbeds that allow them to evaluate multiple hSDN scenarios, its performance, and behavior under specific conditions. If pure SDN experiments are to be conducted, the most popular tools/platforms are: Mininet, EstiNet, NS-3, and MiniNEXT [5]. However, many developing countries have hesitations about moving to full SDN from economic and reliability perspectives, and therefore, exploring hSDN solutions remains relevant. Support for older protocols is decreasing as simulation tool updates are released. There are several scientific studies that have analyzed their effectiveness or convenience at different times. However, over time, updates can introduce many incompatibilities with previous versions, or some systems may no longer be supported. Therefore, it is of great importance to always analyze the news in this process. This article provides an analysis of emulation/simulation systems that support OpenFlow based on updates (at the time of writing).

II. RELATED WORK

In the works [5, 31,33,34,35] on the topic, analyses of network simulation and emulation tools were conducted; at the time of these analyses, some tools were no longer supported or some of their features had changed in subsequent updates. When working with new versions, the result is increased productivity and fewer errors. No matter which tool a new researcher is struggling to use, they can use the results of analyzing similar studies to make informed decisions and produce better results for their future projects. It is effective to conduct such analytical work repeatedly, along with updates.

III. SIMILARITIES AND DIFFERENCES BETWEEN EMULATION AND SIMULATION TOOLS

Network emulation is a hybrid testing method designed to bridge the gap between network simulation and real-world testing. As in simulation, the communication conditions in the simulated network are strictly controlled, ensuring flexibility and repeatability [4]. There are many simulation/emulation tools available today, and it is possible to do the work of two tools at the same time. There are several similarities and differences between emulation and simulation models. Similarities cited in the literature 3:

- **3D representation**. Simulation and emulation models should be three-dimensional if they are to be easy and understandable.
- **Accurate Movement**. In order to be useful, a model must be accurate.
- **Zero-Abstraction Elements**. The less time it takes for observers to understand what they are seeing, the easier it is for them to accept new ideas. Emulation models are not used for experimentation like simulation models. Emulation models are used in a more defined way; to test the operation of a control system under different system load conditions and as a risk-free means of training system operators and key tenants.

and differences:

- **Different Aims**. Simulation models are used to test and develop different solutions to achieve the best solution based on an accepted set of predefined metrics.
- **High-Speed Execution for Simulation**. Most control systems are designed to operate in real time, so emulation experiments must be performed in real time. Simulation models can respond faster than real time, which can be a potential source of error.
- **Repeatability**. Simulation models always work the same way and produce exactly the same results if the parameters do not change. Emulation experiments, on the other hand, can produce different results even when the event is repeated, because they work with a device.

Nowadays, these platforms are very advanced and bring additional opportunities. For example, most emulators are used for simulation purposes only and provide results without connecting actual hardware externally. Some tools do not have a GUI or traffic analysis but integrate with other tools or add relevant modules.

IV. SOFTWARE DEFINED NETWORKING AND OPENFLOW PROTOKOL

Nowadays, while the number of Internet users in the world is growing rapidly, the number of public and corporate networks is also increasing. In this regard, consumers want to enjoy high-quality and secure data transmission over network channels. As for legacy networks, they can no longer cope with the increasing demands for successful operation and increased load. The solution may be the use of software-defined networking (SDN). With SDN, there is no need to increase the number of network elements or purchase advanced routers, switches, firewalls. This technology offers the implementation of networks based on software platforms that are easy to manage and make adjustments to management operators. Through special software applications developed based on the OpenFlow protocol, it is possible to optimize traffic flow, ensure efficient data processing and secure channel transmission. The use of a software-defined network based on OpenFlow is a modern, relevant and promising solution to the problem of efficiency and effectiveness of existing networks. The SDN paradigm has two main concepts: 1) separation of the control layer and the data layer, and 2) a centralized software-based controller in the control layer [1,2]. The controller uses the OpenFlow protocol as the main protocol for its southbound interface. The SDN architecture is shown in Figure 1.

As in many industries, conducting real experiments in the Telecommunications industry can be both financially and expertly challenging. Performing experiments using simulation tools can solve a number of problems and help increase confidence. However, not all tools support the OpenFlow protocol. Some support the OpenFlow protocol, but it is quite complex and can require expertise from the user. Some have stopped supporting it. This can affect the accuracy of the results. There are currently many types of simulation and emulation tools that support the OpenFlow protocol, and you may have difficulty choosing which one to work with. This chapter provides an analysis of some of the tools, and often reviews are released based on these updates, which will help you work with a suitable tool based on the latest benchmarks.

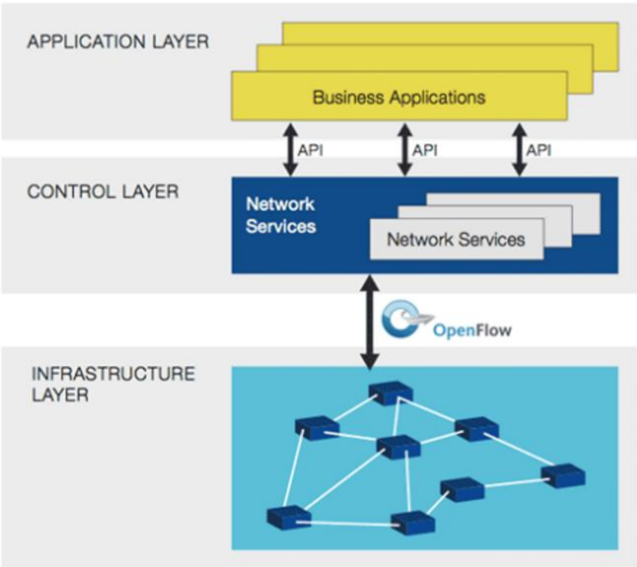

Fig1. Software-Defined Network Architecture [36]

V. NETWORK SIMULATION AND EMULATION TOOLS THAT SUPPORT OPENFLOW

NS-3 is an open source network simulator for Internet systems. This is a very complex tool that runs simulations described by user-generated code, so programming skills may be required to use it [6]. The NS-3 simulator comes with the OpenFlow protocol and a simple controller. This controller lacks the basic functionality provided by real SDN controllers, making it insufficient for experimentation in the scientific community. Additionally, this simulator does not support Spanning Tree Protocol. It can be extended by adding an NS-3 module that supports OpenFlow 1.3 [13], which includes an OpenFlow 1.3 gateway module and a management application interface. The module implementation has a number of drawbacks: it is only available for GNU/Linux platforms, there is only one connection between the switch and the controller, each switch can only be controlled by one controller. Support is available and the last update was ns-3.43, released in 2024 [7].

EstiNet is a network simulator and OpenFlow emulator for studying software-defined networks. It supports both simulation mode and emulation mode. In simulation mode, real open source OpenFlow controllers such as NOX, POX, Floodlight, OpenDaylight, and Ryu controllers can be run directly on a controller node in the simulated network to manage these simulated OpenFlow switches without any modifications [14]. The authors used EstiNet to functionally test and evaluate several NOX/POX components and protocols, such as Learning Bridge and Spanning Tree Protocols. Support is available and the last update was released on August 12, 2020 with EtsiNet version 11 [11].

Mininet is a network emulator whose main elements are virtual nodes, switches, controllers, and communication channels for creating a network. Mininet nodes use standard Linux software [8]. The switches in Mininet support the OpenFlow protocol for dedicated routing and software-defined networks. Mininet version 2.2.0 only provides four models: nodes, SDN switch, legacy switch, and legacy router model. Updated in 2020, Mininet 2.3.0 integrates with Docker, is extended with virtual containers, and makes OVS (Open vSwitch) easier to work with [9]. This allows you to work with the OpenFlow protocol through legacy switches. Only one communication line is implemented, only the bandwidth, delay, and loss probability can be configured in detail. The disadvantage is the lack of support for Windows

ISSN: 2350-0328
International Journal of AdvancedResearch in Science, Engineering and Technology
Vol. 12, Issue 4, April 2025

operating systems. This can be overcome by connecting through a virtual machine. Mininet is not suitable for working with large-scale networks, as this emulator can only run on a single machine, which can lead to a lack of hardware resources. Therefore, MaxiNet [10], based on Mininet, was developed, which increases performance in large networks, but the last support is 2015. This creates a number of imbalances. Additionally, there are limited features for a private traffic generator and performance analysis.

OMNeT++ [15] simulation framework is an open, extensible, modular component-based discrete-event simulator in C++ for modeling communication networks, multiprocessors, and other distributed or parallel systems. This simulates any system consisting of devices interacting with each other. OMNeT++ has a comprehensive graphical interface, and thanks to its modular architecture, the simulation engine and models can be easily incorporated into applications. The INET framework, which enables network modeling, supports many well-known protocols. A unique feature of OMNeT++ is that it decomposes the model into Level 1 of the OSI model with maximum detail of all technologies, protocols, and data formats used. Using traditional tools like Wireshark, it is possible to use real traffic dumps for modeling and collect model traffic for analysis. OMNeT++ provides an OpenFlow 1.5.1 extension module to work with software-defined networks [16]. Support is high, Omnet++ version 6.1.0 was developed in 2024. We can see significant improvements in the update. For example, in the IDE, simulation, and Python libraries, etc. [17].

OFNet is an SDN emulator with similar functionality to the Mininet network emulator, but in addition, it has a number of useful tools for generating traffic and monitoring OpenFlow messages, and evaluating SDN controller performance. OFNet has gained additional necessary emulation capabilities, such as easy debugging of SDN-enabled networks, detection of connection and flow generation errors in controllers, testing controller behavior beyond ping, and obtaining matrices to create a traffic scenario close to the real world [20]. The OFNet project was released in 2016 with version v1.0 and although new versions have not been released or supported, it can be extended by connecting additional components, with the latest components being released on Github in 2024 [21]. This can be confusing for new users.

A preliminary analysis of simulation/emulation tools, i.e. type, year of first release, most recent version, and whether open source or commercial, and the results are presented in Table 1. This helps new users get their first impression.

Table 1. Emulator/Simulators first characteristics

Tools	Type	first produced year	Last update	Source
MiniNet [9]	Emulator	2010	Mininet 2.3.0 (2021)	Open Source
MaxiNet [10]	Emulator	2014	MaxiNet 1.1.0 (2015)	Open Source
EstiNet [11]	Emulator/Simulator	2011	EtsiNet 11 (2020)	Commercial
NS-3 [7]	Simulator	2008	ns-3.42 (2024)	Open Source
Kathará [12]	Emulator	2003	Kathará 3.7.8(2024)	Open Source
OMNeT++ [17]	Simulator	1997	OMNeT++6.1.0(2024)	Open Source
OFNet[29]	Emulator	2016	OFNet v1.0	Open Source
OPNet (Riverbed Modeler) [23]	Simulator	2000	Riverbed Modeler 18.11.1	Commercial
Netsim [24;25]	Emulator/Simulator	2005	Netsim Lite v14.2	Commercial
GNS3 [30]	Emulator	2006	GNS3 version 2.2.53	Open Source

Distributed OF Testbed (DOT) [18] offers a highly scalable emulator for SDN that provides a simulated network across a cluster of machines. It can simulate large SDN deployments by distributing the workload across a cluster of compute nodes. Through extensive experiments, we have shown that DOT overcomes the limitations of Mininet and emulates larger networks [19].

OPNET. The Commercial modeling system OPNET Modeler (currently called Riverbed Modeler) implements network models from various manufacturers that reflect the technical characteristics of real equipment. OPNET also provides the ability to configure detailed traffic parameters. Despite all the advantages of OPNET, it has a number of disadvantages: there are no logical ports between the switch and the controller, there is no hybrid switching, there is no IP fragmentation

ISSN: 2350-0328
International Journal of AdvancedResearch in Science, Engineering and Technology
Vol. 12, Issue 4, April 2025

reassembly, and multiple controllers are not provided. In addition, OPNET is a commercial system [22]. The first version was released in 2000 and was re-released under the name Riverbed Modeler after being acquired by Riverbed Technology in 2012. The last update was in 2024, with Riverbed Modeler version 18.11.1 released separately for Windows and Linux operating systems [23]. You can use older versions of OpNet, or you can achieve better results in Riverbed Modeler with updates.

NetSim is a commercial emulator that implements an SDN module equipped with SDN controllers. Controllers can be used to manage packet forwarding across all Layer 3 devices in the network [24]. The SDN module is currently available for networks such as Internetworks, WSN/IoT, MANET, VANET, and LTE. With this module, any Layer 3 device can be configured as an SDN controller [25]. Support is high and the last update was released in 2024 in NetSim Lite (v14.2) [25].

Kathará is a network emulation system that accurately replicates real system behavior and can utilize multiple virtualization technologies using modularity [26]. Kathará is based on Docker containers and Quagga software routers [27] and allows you to easily reproduce a network topology on a single host machine. That's why Kathará works on all major operating systems (Windows, MacOS, and Linux). Netkit is a network emulator, renamed Kathará in 2020, and is an emulator with many years of experience [28].

Table 2. Extended analysis of network emulators/simulators supporting the OpenFlow

Tools	Programming languages	Support for legacy protocols	GUI	Availability on Windows OS	Scalability	Virtualization technology	Architecture type
MiniNet [9]	Python	No	No	Via virtual machine	~4000 nodes	KVM	Centralized
MaxiNet [10]	Python	No	No	Via virtual machine	unlimited	KVM	Centralized
EstiNet [11]	C++, Python	Yes	Yes	Via virtual machine	thousands	None (kernel re-entering simulation)	Centralized
NS-3 [7]	C++, Python	Yes	No	Via virtual machine	unlimited	KVM	Centralized
Kathará (Netkit)[12]	Python	Yes	No	Through container technology	unlimited	Docer conteyner	Centralized
OMNeT++ [17]	C++ NED	Yes	Yes	Available	unlimited	Xen, KVM, VMWare	Centralized/ Distributed
OFNet[29]	C++	No	No	Via virtual machine	thousands	KVM	Centralized/ Distributed
Riverbed Modeler (OPNet)[23]	C, C++	Yes	Yes	Available	Up to ~5000 nodes	SteelHead-v	Centralized/ Distributed
Netsim[24;25]	C, Java	Yes	Yes	Available	unlimited	KVM	Centralized/ Distributed
GNS3 [30]	Phyton	No	Yes	Available	unlimited	Qemu, VirtualBox, VMware	Centralized

GNS3 [30] is an open-source and enterprise-grade network emulation software for simulating, configuring, testing, and troubleshooting virtual and real networks. GNS3 also supports multi-brand devices such as Huawei, Juniper, Cisco, and more. Additionally, GNS3 has the key advantage of being able to connect to real devices or other virtual machines. Other features include scaling, clustering, and paravirtualization. However, while GNS3 supports a physical wireless card, it does not support wireless network emulation [32]. The GNS3 project allows you to use various VM environments as templates for configuring servers. Additionally, it allows you to create SRv6 nodes based on SONiC and register them as VMs [31].

There are also a few other simulators/emulators, but some have stopped supporting or been merged into others, while others are newly released or not yet popular. Finding new information is difficult or has been discontinued. There are previous versions that might work, but it leads to more evasion of reality. In our future work, we will consider and expand on other tools. Using tools that have been discontinued and haven't had updates for a long time can lead to unrealistic results. Some expanded classifications of these emulators/simulators are given in Table 2.

This was mainly considered according to the following classifications:

Programming language - what is the main programming language and working in the language you know is also important when choosing the right tool. However, most tools are now integrated into multiple languages.

Support for legacy protocols - this is important for building and testing a hybrid SDN architecture. Newer versions of many tools do not support older protocols. When working with older protocols, some extras will need to be added.

GUI (Graphical User Interface) - When creating topologies, most network tools integrate with another tool for a graphical interface. This may cause some difficulties or misunderstandings during operation (especially for new users).

Availability on Windows operating system - Many network simulators/emulators run on Linux and use hypervisors to test them on Windows or another OS. While this isn't that difficult today, it would be even easier if it were installed directly into your computer's operating system.

Scalability - In recent versions of most platforms, the number of connection nodes is large enough to easily accommodate new users or small projects. However, this is of great importance for large projects. Thousands of nodes can be connected on open source platforms (For example, the Mininet community says that 4096 nodes have been successfully connected [9], NS-3 community confirms 750,000 nodes successfully connected in simulator [7]). But this is also a process that depends on the condition of your computer. Commercial platforms divide into usage tiers based on the number of nodes.

Virtualization technology - depending on the virtualization technology used, you will need to choose the tool that is compatible with your computer. Container-based platforms are relatively flexible, but any operating system can be used with containerization technology like Docker or Kubernetes.

Architecture type - many platforms have topologies that automatically create architectures in a centralized architecture. If we need a distributed architecture, we will have to build it based on additional code. This requires practice and knowledge. When working with legacy protocols, using hybrid SDN, and creating complex networks, we may need a distributed architecture.

VI. CONCLUSION AND FUTURE WORK

Today, there are many network emulation and simulation tools that support the OpenFlow protocol, and several scientific papers have been published about these platforms. However, there are some updates that have not been released recently. This can lead to some inconsistencies. For example, some simulation tools do not support older versions of the OpenFlow protocol in their latest updates or require the installation of additional modules. Therefore, in this work, we analyzed some tools with some updates. Some platforms allow you to connect a small number of nodes in their free versions, while others do not have a graphical interface, and additional modules or programs are integrated. However, most platforms now allow you to connect nodes that match the power of your computer as a result of the development and integration of virtualization technologies and containerization technologies. This can increase the efficiency of use. In the future, we will obtain SDN analysis results from our platforms based on these analyses.

REFERENCES

[1] Oleksandr I. Romanov, Ivan O. Saychenko, Anton I. Marinov, Serhii S. Skolets, "Research of SDN network performance parameters using mininet network emulator", Published by Igor Sikorsky Kyiv Polytechnic Institute, Kyiv, Ukraine, 2021, pp 24-31, (https://creativecommons.org/licenses/by/4.0/)

[2] Mohammed Al Somaidai, Survey of Software Components to Emulate OpenFlow Protocol as an SDN Implementation, American Journal of Software Engineering and Applications, 2014; 3(6): 74-82, https://www.researchgate.net/publication/276902675
[3] I. McGregor, "The relationship between simulation and emulation," Proceedings of the Winter Simulation Conference, San Diego, CA, USA, 2002, pp. 1683-1688 vol.2, doi: 10.1109/WSC.2002.1166451.
[4] B. Razvan, Introduction to Network Emulation, vol. 911. 2013.
[5] Sajad Khorsandroo, Adrián Gallego Sánchez, Ali Saman Tosun, JM Arco , Roberto Doriguzzi-Corin, Hybrid SDN evolution: A comprehensive survey of the state-of-the-art, 2021, 40 p. https://doi.org/10.1016/j.comnet.2021.107981
[6] Maygua-Marcillo, L. Creating a new model in NS3 Network Simulator./L. Maygua-Marcillo, L. Urquiza-Aguiar, M. Paredes-Paredes //Preprints.–2019.
[7] NS-3: NS-3 network simulator https://www.nsnam.org Last accessed November, 2024.
[8] Z.J. Allamuratova, K.K. Mukhammedinov SDN Software-defined network emulator performance analysis, "Scientific foundations of raising the use of information technologies to a new level and modern problems of automation" III-international scientific conference, November 20, 2024, Uzbek.
[9] Mininet: Mininet network emulator https://www.mininet.org Last accessed January 2025.
[10] MaxiNet: MaxiNet network emulator http://maxinet.github.io Last accessed December 2024.
[11] EtsiNet: EtsiNet 11 network simulator http://www.gordonsmart.com Last accessed December 2024.
[12] Kathará: Kathará network emulator https://www.Kathará.org/ Last accessed December 2024.
[13] Internet source: https://github.com/ljerezchaves/ofswitch13
[14] Shie-Yuan Wang, Chih-Liang Chou, and Chun-Ming Yang. 2013. EstiNet openfl ow network simulator and emulator. Communications Magazine, IEEE 51, 9 (September 2013), 110–117.
[15] András Varga and Rudolf Hornig. 2008. An Overview of the OMNeT++ Simulation Environment. In Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops.
[16] Salih, M.A. OpenFlow 1.3 extension for OMNeT++ / M.A. Salih, J. Cosmas, Y.Zhang //2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing. – IEEE, 2015. – C. 1632-1637.
[17] OMNeT++ : OMNeT++ network simulator https://omnetpp.org/ Last accessed January 2025.
[18] A. R. Roy, M. F. Bari, M. F. Zhani, R. Ahmed, and R. Boutaba. 2014. Design and management of DOT: A Distributed OpenFlow Testbed. In Network Operations and Management Symposium (NOMS), 2014 IEEE.
[19] Arup Raton Roy, Md. Faizul Bari, Mohamed Faten Zhani, Reaz Ahmed, and Raouf Boutaba David R. Cheriton Design and Management of DOT: A Distributed OpenFlow Testbed, Article in ACM SIGCOMM Computer Communication Review, https://www.researchgate.net/publication/266660177
[20] S. Asadollahi, B. Goswami, A. S. Raoufy and H. G. J. Domingos, "Scalability of software defined network on floodlight controller using OFNet," 2017 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT), Mysuru, India, 2017, pp. 1-5, doi: 10.1109/ICEECCOT.2017.8284567.
[21] Online source: https://github.com/search?q=OFNet&type=repositories Last accessed January 2025.
[22] Lee, S. Performance comparison of software defined networking simulators for tactical network: Mininet vs. OPNET / S. Lee, J. Ali, B. Roh //2019 International Conference on Computing, Networking and Communications (ICNC). – IEEE, 2019. – C. 197-202.
[23] OPNet: Rivered Modeler network simulator https://support.riverbed.com/ Last accessed January 2025.
[24] Netsim: Netsim network simulator https://www.boson.com/ Last accessed January 2025.
[25] Netsim: Online kirish https://www.tetcos.com/ Last accessed January 2025.
[26] Mariano Scazzariello, Lorenzo Ariemma, Tommaso Caiazzi Kathar´a: A Lightweight Network Emulation System, 2020 IEEE/IFIP Network Operations and Management Symposium, NOMS 2020.
[27] M. Polverini, I. Germini, A. Cianfrani, F. G. Lavacca and M. Listanti, "A Digital Twin based Framework to Enable "What-If" Analysis in BGP Optimization," NOMS 2023-2023 IEEE/IFIP Network Operations and Management Symposium, Miami, FL, USA, 2023, pp. 1-6, doi: 10.1109/NOMS56928.2023.10154393.
[28] Internet source https://github.com/KatharáFramework/Kathará Last accessed February 2025.
[29] Internet source https://brianlinkletter.com/2016/11/ofnet-a-new-sdn-network-emulator/ Last accessed February 2025.
[30] GNS3: GNS3 network simulator: https://www.gns3.com/ Last accessed February 2025.
[31] Elisa Rojas, Roberto Doriguzzi Corin, Kévin Phemius, Carmen Guerrero, Are We Ready to Drive Software-Defined Networks? A Comprehensive Survey on Management Tools and Techniques, Article in ACM Computing Surveys, February 2018, DOI: 10.1145/3165290
[32] M. Park, S. Byun and N. Ko, "5G Test Environments on GNS3 with SDN Network," 2024 15th International Conference on Information and Communication Technology Convergence (ICTC), Jeju Island, Korea, Republic of, 2024, pp. 1030-1031, doi: 10.1109/ICTC62082.2024.10826971.
[33] Kabir M. H. et al. Detail comparison of network simulators //International Journal of Scientific & Engineering Research. – 2014. – T. 5. – №. 10. – C. 203-218.
[34] Saba Siraj, Ajay Kumar Gupta, Rinku-Badgujar Network Simulation Tools Survey, International Journal of Advanced Research in Computer and Communication Engineering Vol. 1, Issue 4, June 2012 ISSN: 2278 – 1021
[35] Veepin Kumar, Sanjay Singla Performance Analysis of Commonly Used Simulators and Emulators in MANET, Journal of Positive School Psychology 2022, Vol. 6, No. 2, 5390 – 5398
[36] Internet source: https://opennetworking.org/sdn-resources/whitepapers/software-defined-networking-the-new-norm-for-networks/

ISSN: 2350-0328
International Journal of AdvancedResearch in Science, Engineering and Technology
Vol. 12, Issue 4, April 2025

AUTHOR'S BIOGRAPHY

Kobeysin Mukhammedinov
Born in 1994 in Nukus, Karakalpakstan, Uzbekistan. He received his bachelor's degree in 2018 and his master's degree in 2020 from the Tashkent University of Information Technologies named after Muhammad al-Khwarizmi. He is currently a PhD student at the same university and is conducting research in the field of Telecommunications. As part of his research, he has published articles in 7 local journals and conferences. His areas of interest include SDN, NFV, network programming and Network slicing.