UDC 004.732

OPTIMIZING AN AFRICAN NETWORK: CASE of VIPNET

D. E. H. Noutsawo

Odessa National Polytechnic University

Abstract. *The fundamental idea of the SDN is to make a network programmable, that is to say more able to undergo changes related to requests for services more and more dynamic. The main technique used is the implementation of OpenFlow interfaces so that external applications can directly interact with the network.*

Key words: *SDN, SDN controller, OpenFlow, Orchestration, Network Virtualization.*

Introduction

Rapid advancement in the evolution of network technologies has made us benefit from a wide variety of applications and network services requests. Today, the newest and most popular of these innovations is the Software Defined Network (SDN). One of fundamental goals of this technology is to allow the implementation of flexible networks that can be provided dynamically.

SDN redefines the network into an underlying logical network segment using the same equipment. It is an architectural paradigm that separates the control plane of a networking device from its data plane, making it feasible to control, monitor, and manage a network from a centralized node. The SDN is divided into 2 fundamental parts. The major parts consist of the programmability of the network and the centralized control and the minor ones (but equally important) include the virtualization of the network and its orchestration.

The Key components of the SDN architecture includes a data plane ((sometimes known as the user plane, carrier plane or bearer plane) is the part of a network that carries user traffic.) consisting of network resources for data forwarding, a control plane (architecture that makes it possible to allocate control protocol functions across multiple processor levels in the network system) comprising of SDN controller(s) providing centralized control of network resources and the control/management.

Following this framework, we discuss the technical challenges of realizing SDN integration for African companies.

The aim of research: The aim of this study is to design a computer network that is capable of allowing truly differentiated services to coexist on the same infrastructure thanks to a centralized control plane.

1. African companies infirmities

VIPNET is an internet service provider. Its number of customers is > 300 and its customers are based throughout Ivory Coast. Figure 1 shows the point-to-point packet transmission scheme within the VIPNET network.

A layer may have arrows going to a lower layer; this is known as the South service interface. It can have incoming arrows coming from an upper layer; this is known as the North service interface.

There are cases where layers of the same level are struggling with each other, such as totally or partially distributed control plans. The arrows between the layers represent interfaces

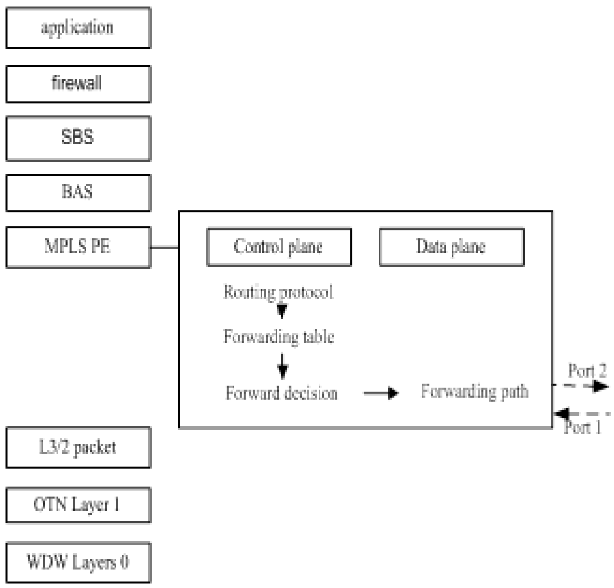

Fig. 1. packet transmission in the VIPNET network

Like most African telephone and internet companies, VIPNET equally uses this type of traditional architecture. The existing network architectures were not designed to meet the requirements of today's users, enterprises, and carriers. The network designers are therefore, constrained with some limitations from the use of this outdated network. Such limitations include:

-Complexity of the current networks: Multiple conversions of the traffic on the different traversed OS;

-Inconsistent policies;

-Inability to scale: As demands on the data center rapidly grow, so too must the network grow. Users go through multiple data centers simultaneously to access services. The client-server architecture quickly becomes obsolete;

-Vendor dependence: All the equipments are not compatible to each other; hence, sometimes they are bound to stay with the same operator;

-Limitations imposed by the physical equipments;

-The Command-line interface implementation (subject to many errors);

-High cost of leased lines.

By adding the SDN it changes the previous architecture and introduces the orchestration and network control within the VIPNET. Figure 2 shows the logical architecture of the SDN with a control layer where important operations such as routing, priority policies and security are carried out.

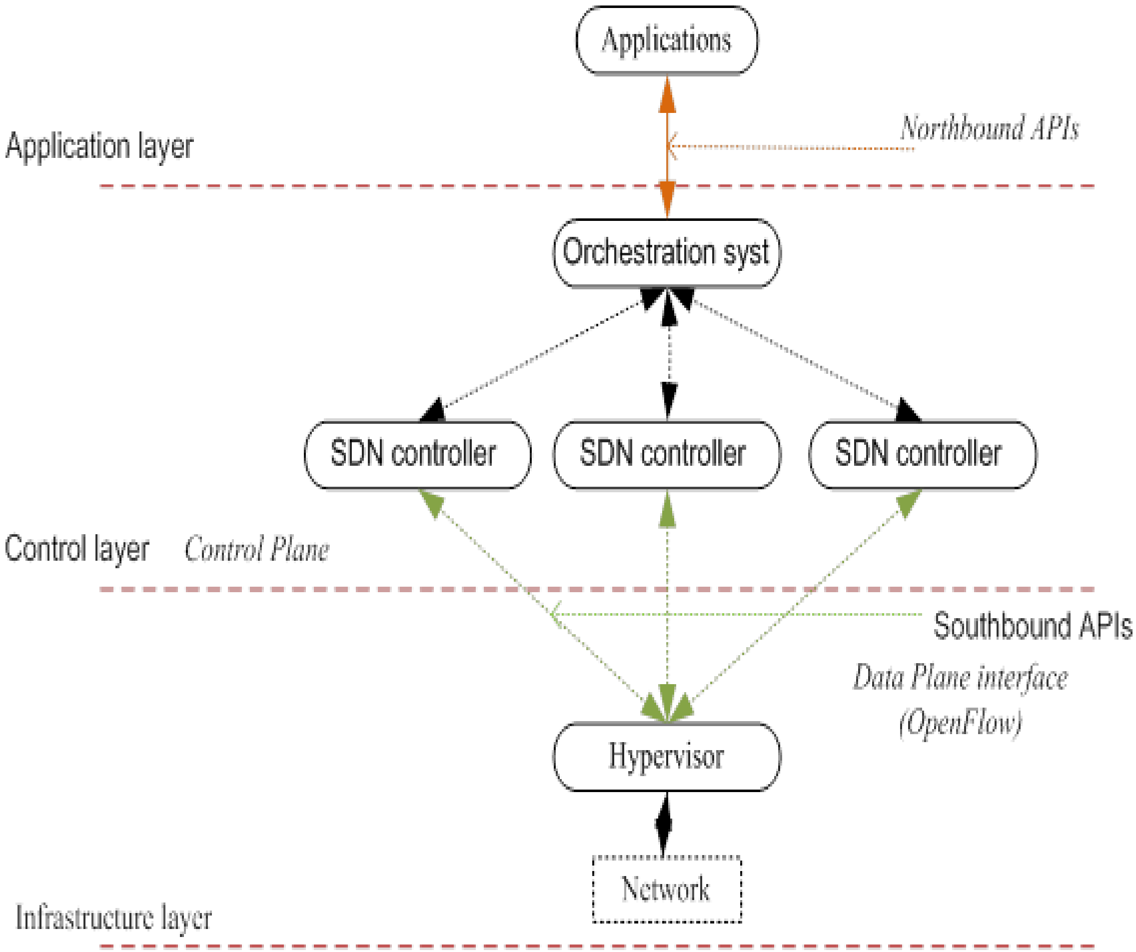

Fig. 2. SDN logical architecture in VIPNET

This structure will collaborate with the old one to maintain the production of the company and to allow a slow but careful migration towards a completely virtualized network. Figure 3 is a hybrid network representing the results of SDN integration

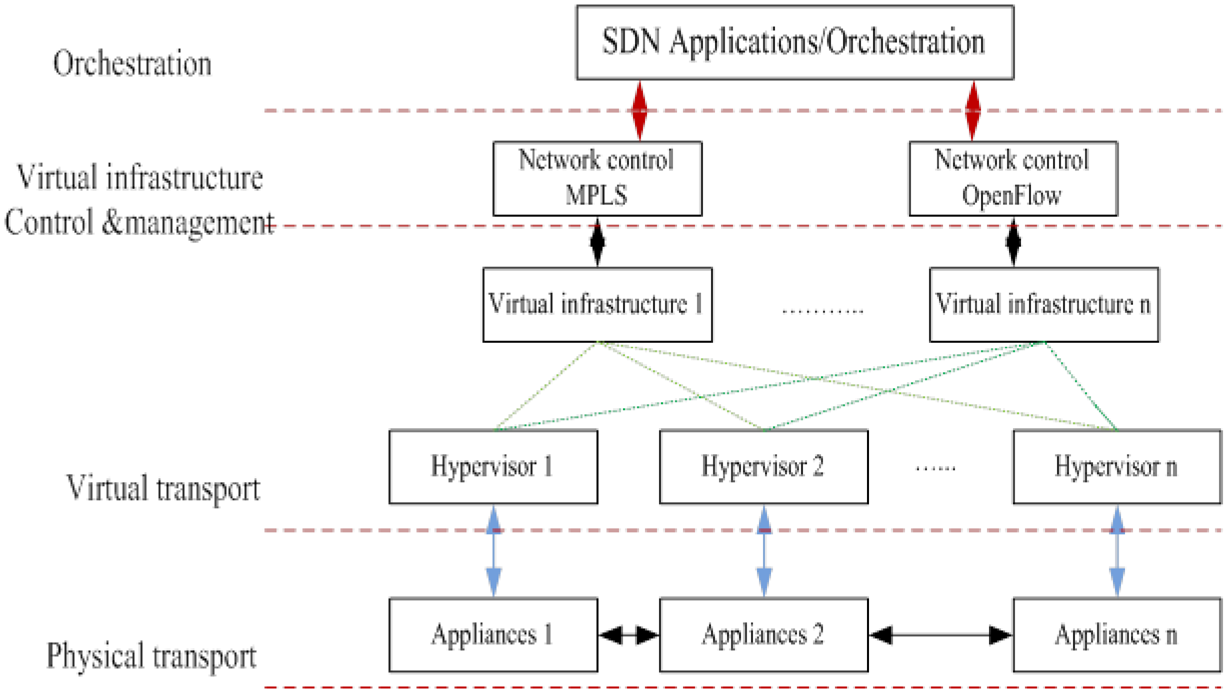

Fig. 3. SDN over traditional network of VIPNET

The most important changes in this structure are due to orchestration, virtualization, controllers and the OpenFlow application whose function and complexity is explained bellow.

1.1. Virtualization

In the traditional design, virtualization has the merit of simulating multiple virtual machines on a single physical machine. We find ourselves always lack of the capacity to support the volume of more recent data. Added to the fact that the programming is very complex, it is clear that this network becomes difficult to administer and / or to repair. This is the state of the VIPNET network.

Instead of a server virtualization that runs multiple virtual servers on a physical server, we propose to virtualize the network. Virtualizing a network amounts to running multiple virtual networks on a physical network knowing that each virtual network has the illusion that it functions as a physical network. The program that provides an abstraction layer for the network hardware is a hypervisor.

Network hypervisors create completely decoupled virtual networks independent of the underlying physical network. The hypervisor allows segments of a virtual network to be independently managed and provisioned dynamically. It virtualizes the transport network and submits only an abstract representation to the SDN. The physical components that make up the network (such as switches, routers, and firewalls) can be virtualized to meet the traffic demands created by server virtualization.

1.2. Orchestration

In the SDN architecture, the Orchestrator provides an end-to-end programming function. The latter receives an order from an application and performs the following actions necessary to complete the requested task. To achieve its mission, the orchestrator relies on each element of the chain on the deployed controllers.

The orchestrator usually works in a "transactional" mode. It must be able to validate each step when it performs an action to go back in case of failure.

SDN orchestration techniques which are monitoring, design, control and QoS leverage the power of programmable networks to facilitate all the networking functionalities at the fingertips of networking administrators and architects.

1.3. Controller

A centralized SDN controller acts as a consistent global database and the SDN specific mechanisms ensure that a packet entering the network is consistently managed by all SDN switches. It defines the data flows that occur in the SDN data plan. Each stream crossing the network must first obtain the controller's authorization, which verifies that the communication is authorized by the network policy. If the controller allows a stream, it calculates a route for the stream and adds an entry for that stream in each of the switches along the path.

NOX is the original OpenFlow controller, in which case the one that was installed to upgrade the VIPNET network. It has of the following components:

-Event Dispatcher: Provides tools that allow application components to communicate with each other by sending events and listening to them.

-OpenFlow manager: An application that provides an interface that operators use to manage OpenFlow networks and deploy them;

-Dynamic deployment of shared objects: Analyzes the directory structure for all implemented components;

-Entry / output: asynchronous file;

-OpenFlow API: This is a south-oriented API that facilitates efficient network control and allows the SDN controller to make dynamic changes based on real-time requests and needs;

-Core - Services: threading and event management;

-Other: Network protocols, data structures, utilities.

The controller consists of channels that use the Nordbound APIs to communicate with external networks, business and SDN applications, and southdbound APIs to communicate with the Openflow switch.

The northbound API allows you to configure, analyze, troubleshoot, and deploy new devices on the network, while with the Southbound API, the network controller will discover network devices, detect service configurations, and gather all the information necessary to manage the network.

When an OpenFlow switch cannot reach its controller, the stream remains unavailable until the connection is restored.

The following network controller features enable you to configure and manage physical and virtual equipment and services.

-Firewall management;

-Manage software load balancing;

-Virtual network management;

-Management of the Remote Access Service gateway.

1.4. Openflow

OpenFlow is the primary interface protocol that manages communication between the control and data layer devices (the controller and the switch) and defines how a controller and device communicate with each other to ensure that the traffic flow is switched and / or controlled as programmed. Keeping in mind that the controller can be configured quickly and change the behavior of the entire subnets or change the behavior of a single device as is necessary for a specific traffic flow. We have two types of OpenFlow; simple OpenFlow and OpenFlow Hybrid (which we used). In the later case, the switcher can also switch the packets according to its traditional mode as Ethernet switch or IP router. The hybrid in this case can be seen as an OpenFlow switch that resides next to a traditional and independent switch. This type of hybrid switch requires a classification mechanism that either directs packets to the OpenFlow controller or processes them in a traditional way. That's why we have a Nox controller which is a platform that implements southbound APIs.

The two functions of an OpenFlow switch are; Control support function and Data forwarding function. The control support function; the switch will communicate with the controller and the controller will manage the switch with the OpenFlow protocol in order to support the programmability of the network resources. Meanwhile the Data forwarding function will direct incoming flows from other network devices and end systems to a path that has been previously calculated and established from the rules defined by the SDN applications and passed to the controller.

The main components of the OpenFlow switch are the ports, the flow tables, the group tables, and the counter tables. We associate an entrance tail and an exit tail. But we will linger only on the flow tables.

The NOX controller manages the set of switches in the VIPNET network that will allow it to add, delete and update the flow tables. During processing, when a packet reaches an entry, there is a match, the instructions associated with that entry are executed, if there is no match and we have the following cases:

- The packet continues through pipeline processing to the next flow table;

- The packet is switched to the controller;
- The package is abandoned.

We have noticed that generally when pipeline processing stops, the packet is then routed to a terminal network and not dropped (almost no packet loss).

Each feed entry contains a set of instructions that may be 'required' or 'optional'. We quoted the most important.

-Meter id: Directs the packet to the specified counter. The result of the measurement may be to drop the packet; it depends on the configuration of the meter;

-Apply-Actions: This applies the specific actions immediately, without any modification of the set of actions. This statement can be used to change or modify the packet between two tables;

-Clear-Actions: Clears all actions in the action set immediately;

-Write-Actions: Adds a new action on an existing set of actions;

-Write-Metadata / mask: Write the hidden metadata value;

-Goto-Table next-table-id: Indicates the next table in the pipeline processing.

All these statements are executed in the exact order specified by the preceding list. Flow tables may not support all matches, instructions, or actions.

These instructions and those executed in the group table and the counter table makes all the complexity and gives all the advantages of an OpenFlow technology.

2. Results

2.1. User’s level

-The changes are directly felt by customers who complained of latency and sometimes lack of signal.

-The possibility for customers to pay only their consumption was well received compared to the old pricing which a fixed rate was corresponding to a bandwidth or a service.

-A carrier could introduce a video service that offers premium subscribers the highest possible resolution in an automated and transparent manner.

2.2. Network’s level

-Reduced complexity through automation: OpenFlow-based SDN offers a flexible network automation and management framework, which makes it possible to develop tools that automate many management tasks that were done manually before;

-Offers a centralized view of an organization’s entire network, making it easier to streamline enterprise management and provisioning;

-Centralized control of multi-vendor environments: SDN control software can control any OpenFlow-enabled network device from any vendor, including switches, routers, and virtual switches;

-Remote deployment and without intervention: Automated deployment, secure and without intervention all the necessary components at remote sites;

-More granular network control: OpenFlow‘s flow-based control model allows to apply policies at a very basic level (session, user, device, and application) in a highly abstracted and automated fashion;

-Real-Time Network Optimization: In the face of changing network conditions, such as the initiation of high-priority file transfer, a controller can respond by inserting streams to ensure that the specific handover receives the quality service required or to ensure that it is redirected to the most optimal route;

- Wireless LAN Controller (WLAN) Bypass: A WLAN controller can detect the initiation of a high bandwidth stream between two wireless clients and can inform the OpenFlow controller to redirect that specific stream along a shorter path so that the WLAN controller can be excluded from the high bandwidth flows;

Conclusion

IT environments are becoming more dynamic, powerful, distributed and hybrid, but the equipment and the architectures used in Africa have not evolved much to meet the standards they deserved. Some are already outdated and obsolete. The strength of SDN is in the fact that it reinvents the network by rules of automation and orchestration of all the digital components within a unified network. If African companies are slowly evolving, they are still evolving.

It is therefore time to update the architecture of African networks. For that, it would be advantageous to look as much at the SDN-WAN as at the SDN-carrier who implement innovative techniques to manage a global network and that of automatic way.

References

1. Nunes, B., et al. (2014): "A Survey of Software-Defined Networking: Past, Present, and Future of

2. Programmable Networks," IEEE Communications Surveys & Tutorials, PP.99 1-18.

3. Gude, Natasha, et al. (2008): "NOX: towards an operating system for networks." *ACM SIGCOMM*

4. Computer Communication Review 38.3 105-110.

5. D. Drutskoy, E. Keller, and J. Rexford. (March/April 2013) " Scalable network virtualization in software-defined networks". IEEE Internet Computing.

6. Tootoonchian, Amin, et al., (2012) "On controller performance in software-defined networks," in

7. USENIX Workshop on Hot Topics in Management of Internet, Cloud, and Enterprise Networks snd Services (Hot-ICE), p. 54.

8. Hybrid OpenFlow Using the Normal Action, (2013). Available at: http://networkstatic.net/hybrid-openflow-using-the-normal-action/ (accessed fev 2018).

9. Orchestration of Ethernet Services in Software-Defined and Flexible Heterogeneous Optical Networks - the EU/JP Project STRAUSS, (April 2014). Available at: https://www.slideshare.net/ADVAOpticalNetworking/ (accessed jan 2018).

10. SDN Architectural Limitations: Towards a Full Software Network Vision. Available at: https://sdn.ieee.org/newsletter/may-2016/sdn-architectural-limitations-towards-a-full-software-network-vision (accessed jan 2018).

11. Fonctionnalites indispensables du SD-WAN, (2017). Available at: https://www.whitepapers.em360tech.com/ wp-content/uploads/Key-Requirements-French.pdf (accessed jan 2018).

12. Software-Defined Networking: The New Norm for Networks,(2012). Available at: https://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdn-newnorm.pdf

13. McKeown, Nick, al. (2008) "OpenFlow: enabling innovation in campus networks." ACM SIGCOMM Computer Communication Review 38.2 pp. 69–74.

14. Shalimov, Alexander, et al. (2013) "Advanced study of SDN/OpenFlow controllers." Proceedings of the 9th Central & Eastern European Software Engineering Conference in Russia.

15. B. Raghavan et al. (2012). "Software-defined internet architecture: Decoupling architecture from infrastructure" in Proc. 11th ACM Workshop Hot Topics Network, pp. 43–48.

16. N. Mckeown, (2011) "How SDN will shape networking". Access Mode: http://www.youtube.com/watch?v=c9-K5O_qYgA.

17. S. Schenker, (2011) "The future of networking, the past of protocols". Available at: http://www.youtube.com/watch?v=YHeyuD89n1Y.

18. Open Network Foundation, (2013) OpenFlow Switch Specification Available at: https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.4.0.pdf.

19. ALTO and SDN. Available at: http://www.ietf.org/proceedings/84/slides/slides-84-alto-5.

20. S. Yeganeh, A. Tootoonchian, and Y. Ganjali, (Feb. 2013) "On scalability of software-defined networking" IEEE Commun. Mag., vol. 51, no. 2, pp. 136–141.

21. Sherwood, R. (2014). Modern OpenFlow and SDN - *Part I*. available at: http://bigswitch.com/blog/2014/05/02/modern-openflow-and-sdn-part-i

22. Colt's Carrier SDN & NFV: Experience, Learnings & Future Plans (2017) available at: https://www.slideshare.net/OpenNetworkingSummit/colts-carrier-sdn-nfv-experience-learnings-future-plans.

ОПТИМІЗАЦІЯ АФРИКАНСЬКОЇ МЕРЕЖІ: ВИПАДОК VIPNET

Д. Е. Х. Ноутсаво

Одеський національний політехнічний університет

Анотація. *Інтернет - це цифрове суспільство, де все підключено і доступно з будь-якого місця. Однак, незважаючи на їх широке впровадження, традиційні IP-мережі складні і дуже важкі в управлінні. Важко налаштувати мережу відповідно до зумовленням політиками і переналаштувати її для реагування на помилки, завантаження і зміни. SDN, нова парадигма дозволяє змінити цей стан*

речей. він відокремлює площину управління і план даних з єдиною метою - змінити таким чином призначення завдань при передачі потоку.

Основна SDN полягає в тому, щоб зробити мережу програмованою, тобто більш здатною піддавати зміни, пов'язані із запитами на послуги, більш динамічними. Це технологія, що дозволяє детермінованим чином спростити архітектуру, доставку і роботу мережевих сервісів. Основною використовуваною технікою є надання інтерфейсів програмування OpenFlow, щоб зовнішні додатки могли безпосередньо взаємодіяти з мережею.

В цій статті ми обговоримо дизайн африканської мережі, що користується перевагами SDN. VIPNET, середній провайдер інтернет-послуг використовує технологію SDN для поновлення своєї домашньої та фізичної мережі. в цьому контексті він використовується в основному для забезпечення стабільного зв'язку з клієнтами, які в більшості випадків використовують додатки реального часу, що вимагають дуже великої ємності. З цієї новою архітектурою з'являються контролери OpenFlow, які керують управлінням потоком. Інтерфейси OpenFlow реалізовані на обладнанні, такому як контролер, комутатори, маршрутизатори, що належать постачальникам і / або клієнтам. ці інтерфейси, також звані південними інтерфейсами, мають функцію управління зв'язком між контролерами та іншими вузлами мережі. Це можуть бути зв'язки з фізичним рівнем або посиланнями з віртуальними комутаторами.

Ця здатність до оркестровки і віртуалізації логічних мереж - це інновація, очікувана фахівцями мережі.

Ключови слова: *SDN, контролер SDN, OpenFlow, оркестровка, мережева віртуалізація*

ОПТИМИЗАЦИЯ АФРИКАНСКОЙ СЕТИ: СЛУЧАЙ VIPNET

Д. Е. Х. Ноутсаво

Одесский национальный политехнический университет

Аннотация. *Основная идея SDN заключается в том, чтобы сделать сеть программируемой, то есть более способной переносить изменения, связанные с запросами на услуги, более динамично. Основной используемой методикой является предоставление интерфейсов программирования OpenFlow, чтобы внешние приложения могли напрямую взаимодействовать с сетью.*

Ключевые слова: *SDN, контроллер SDN, OpenFlow, оркестровка, сетевая виртуализация.*

Received on 26.03.2018

Noutsawo Happi Diane Eddosie, PhD student in the Institute of Computer Systems, Odessa National Polytechnic University, Shevchenko ave, 1, Odessa, Ukraine, E-mail: dianehappi.dh@gmail.com, tel. 063-936-0973

Ноутсаво Хеппи Диан Еддози, аспирантка кафедры институт компьютерных систем Одесского национального политехнического университета. Просп. Шевченко, 1, Одесса, Украина, E-mail: dianehappi.dh@gmail.com, тел. 063-936-0973

ORCID ID: 0000-0002-6298-5437