UDC 004.72

Oleksii Cherevatenko, Yurii Kulakov

ANALYSIS OF TECHNOLOGIES OF SOFTWARE-DEFINED NETWORKS

Олексій Череватенко, Юрій Кулаков

АНАЛІЗ ТЕХНОЛОГІЙ РЕАЛІЗАЦІЇ МЕРЕЖ, ЩО ПРОГРАМНО КОНФІГУРУЮТЬСЯ

The article describes main modern technologies that implement the concept of software-defined networks (SDN) on different levels of management – OpenFlow, ONOS and CORD. An analysis of interconnectedness of these technologies has been conducted, innovations of each of the technologies has been studied. The future of the SDN concept has been predicted in the article.

Keywords: software-defined networks, OpenFlow, ONOS, CORD.

Fig.: 4. Tabl.: 0. Bibl.: 7.

У статті описуються основні сучасні технології, що реалізують концепцію програмно конфігурованих мереж (SDN) на різних рівнях управління мережею – OpenFlow, ONOS та CORD. Проводиться аналіз взаємопов’язаності цих технологій, вивчаються нововведення, що наявні у кожній з них. Прогнозується розвиток концепції SDN у майбутньому.

Ключові слова: програмно конфігуровані мережі, OpenFlow, ONOS, CORD.

Рис.: 4. Табл.: 0. Бібл.: 7.

Relevance of the research topic. The problem of scalability of computer networks in the XXI century has become global according to the big size of networks and variety of devices, so it requires automation of control. The concept of software-defined networks (SDN) can offer this and there are several interconnected multilevel implementations, which are OpenFlow, ONOS, and CORD, and they considered in this article.

Target setting. To date, there is no single laconic documentation that would explain the interconnection between the different levels of the software-defined network, and would describe the advantages and disadvantages of each solution in the context of the overall construction of the network.

Actual scientific researches and issues analysis. Currently, there are some studies about software-defined networks, but network equipment manufacturers have no single standard and opinion about how should the practical implementation of SDN look like, and CORD technology is still not well understood by information technologies companies due to its novelty.

Uninvestigated parts of general matters defining. The article deals with the construction and interconnection of software-defined networks based on the CORD platform (including ONOS and OpenFlow at lower levels), which has not yet received general recognition in network technologies.

The research objective. The purpose of the article is to determine if it is technically and economically feasible to create a software-defined network based on the connection of OpenFlow-ONOS-CORD multilevel technologies, according to the analysis of the advantages and disadvantages of SDN.

The statement of basic materials. The principle of a software-defined network (SDN) consists of separating and managing the processes of transmitting traffic to the network.

This interpretation roots from the principle of building network devices (such as routers and switches) that implement three logical processes and have the appropriate hardware structure: dataplane, controlplane (regulation), and managementplane (administration). Typically, all these processes are monitored on each individual network element, which results in a large amount of time spent on network configuration and significant resource utilization. The principle of programmed configuration is the separation of two processes – administration and regulation – into a separate centralized system, which will use the software tools, to configure the entire network at once, and this will save time and increase traffic flow performance [1]. The network elements in SDN thus consist of only two components - the physical chips that are responsible for the ability to transmit traffic and the easiest forwarding table that will "carry" packets of traffic "through" the router or switch (Figure 1) [2].

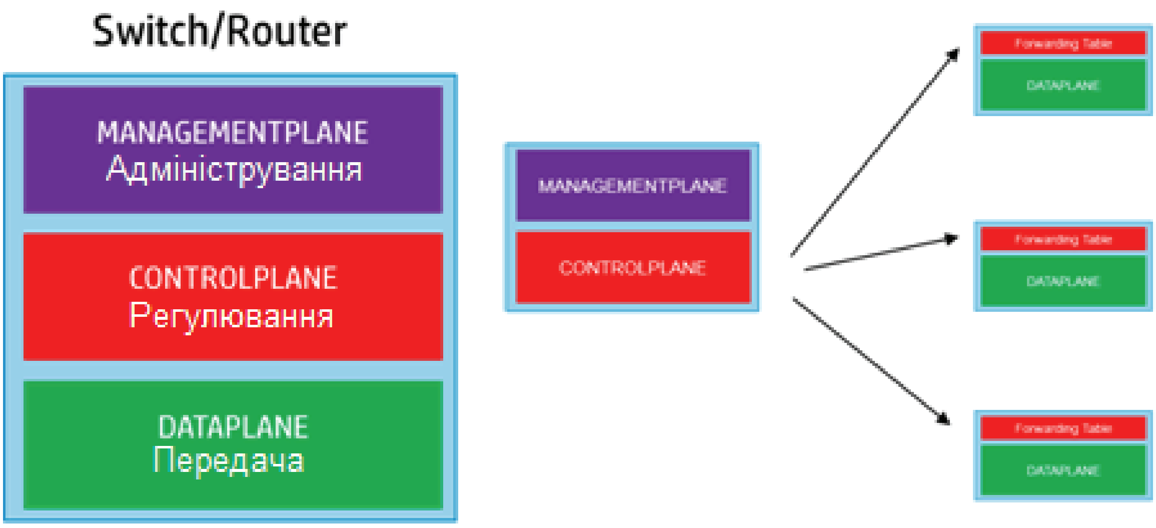

Figure 1. Separation of the levels of management of network devices in the SDN concept

The lowest level of the system under consideration is the open network protocol OpenFlow, which allows you to perform the adjustment not directly from the network device, but from the central controller of the network segment. In a basic implementation, OpenFlow connects a switch that supports OpenFlow (OpenFlow Switch, OFS) and an OpenFlow controller (OFC).

The OFS consists of two parts: a flow table that stores records received from the controller and a secure channel through which the OFS and the OFC communicate. Together with OpenFlow, some protocol such as SSL can be used in pair to provide more secure connection. After the controller software created a new command, the controller sends it to the OFS secure channel, and then adds to the flowchart as a new record. The commands received from OFC may be different - sending a frame to the port or IP address specified in the record title, dropping the frame, sending the frame back to the controller. The OFC itself can be various devices, including a personal computer (in this case, it will be the easiest for the programmer to make software adjustments), although usually a specialized server is selected as a controller. The main advantage of OpenFlow is the significant time savings required to reconfigure the system by establishing direct connections between the network elements [3]. The basic scheme of the SDN network using the OpenFlow protocol is shown in Figure 2.

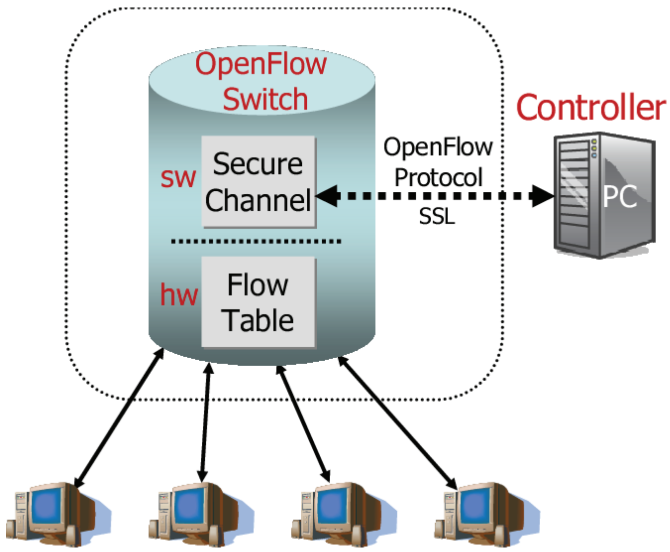

Figure 2. The basic structure of a simple SDN network, the elements of which are connected through the OpenFlow protocol

The next level of the software-defined network is ONOS, which is an open operating system and is positioned as a network solution consisting of a service provider and its clients. ONOS is a linker for simpler SDNs, being a peculiar cluster for multiple nodes. The advantage of using ONOS as a system for a large network controller is the ability of this system to cope with system failure in case of a breach of the setting of individual nodes. In addition, the node where the addressing or connection error occurred may be promptly reconfigured by the software controller.

Considering that OpenFlow and other similar network protocols are integrated with ONOS will be incorrect, because OpenFlow is used only to establish a connection between network devices, but the ONOS operating system itself is provided with its own applications and software solutions. They are divided into tiers, each of which has its own software models (one of them interacts with OpenFlow) and can work independently. According to this level isolation, the system becomes versatile, because it is not tied to a single model or protocol and can interact at different levels with other network elements across the topology (Figure 3) [4].

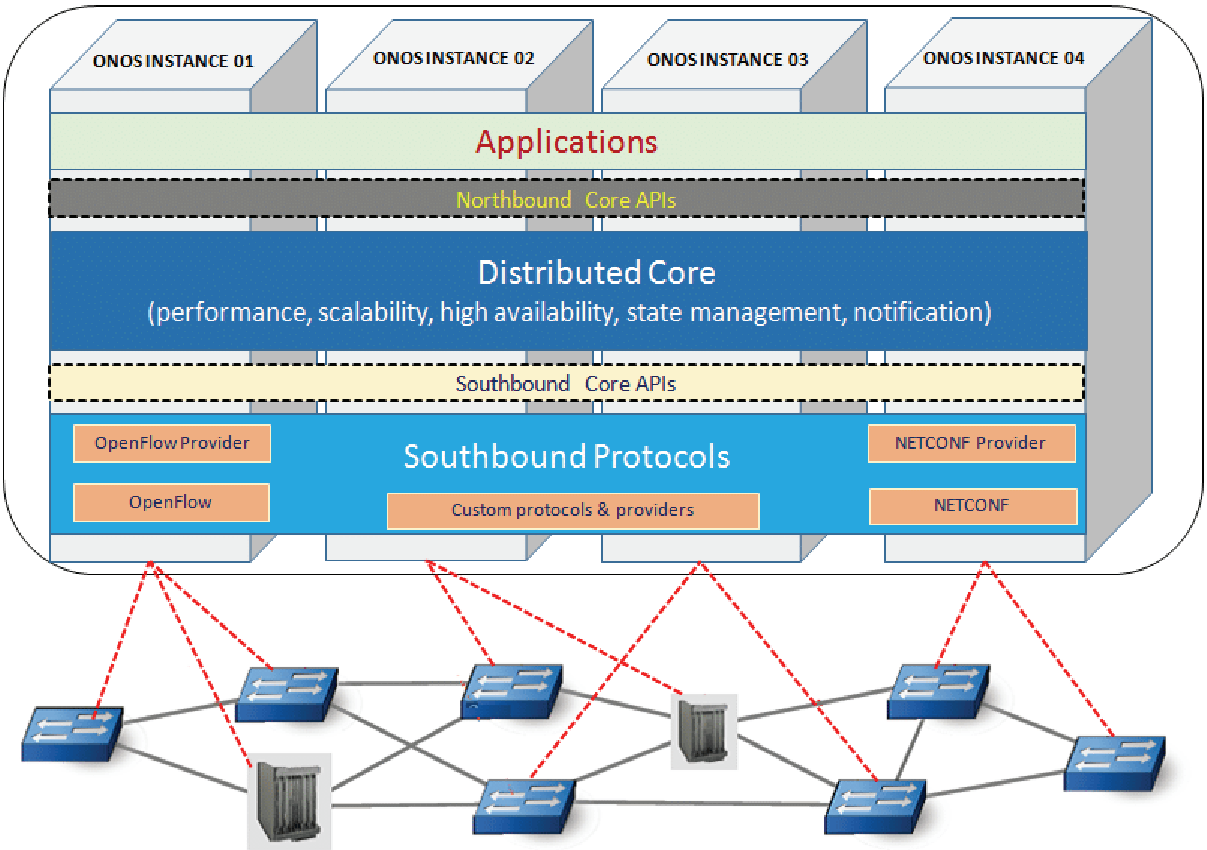

Figure 3. Dividing of the ONOS system to levels.
OpenFlow switches communicate via the OpenFlow Provider (bottom left)

Based on levels and models, ONOS has applications that allow the programmer to configure the network and control the traffic. Network topology information is stored in the control center device where the operating system is running. From the same center, you have the ability to connect directly to a particular device on the network and send it some data. This expands network control capabilities and simplifies application development and administration. Actual applications can be added or removed from the network dynamically without the need to stop traffic and disable or restart the network, which is a huge advantage for providers that require uninterrupted connection to subscribers [5].

Based on the CORD operating system, controllers are built using the CORD architectural solution, which is described below.

The highest level of the network described is the CORD technology, which is positioned as a solution for Internet service providers and implements the idea of a software-defined network on a large scale. It involves the creation of a single virtual management center that can process data dynamically, using cloud computing for this. This center communicates with intermediate elements of the network and end users. With the help of CORD, the service provider has the ability to set up a network that will be programmatically configured on remote computing power, and this provides the whole system with several important benefits - with automated control, the platform becomes simpler and faster (because it does not require human intervention, all configurations are made by the software), more flexible (because the program dynamically configures network elements, constantly synchronizing them and making necessary updates), while using cloud computing opens a perspective of decreasing the amount of physical hardware what brings tangible savings and reduce the technical complexity of the construction of the network.

CORD can be used to transfer data to three types of clients - mobile users, corporate and home users. Users are connected to intermediate data centers (one for each of the described types) and themselves are connected to a CORD controller that configures the entire network. The controller itself consists of many specialized controllers, each of which executes its own part of the software code - for example, one of these controllers can perform routing management, and another interconnection of different network levels. The CORD controller is connected to the server base, which stores the configuration of the system and performs operations that are sent to it by the controller. Additionally, it joins the core architecture of the CORD architecture, the configurable multiplexer, ROADM, which provides multiple inputs and one access output to the global network (Figure 4) [6].

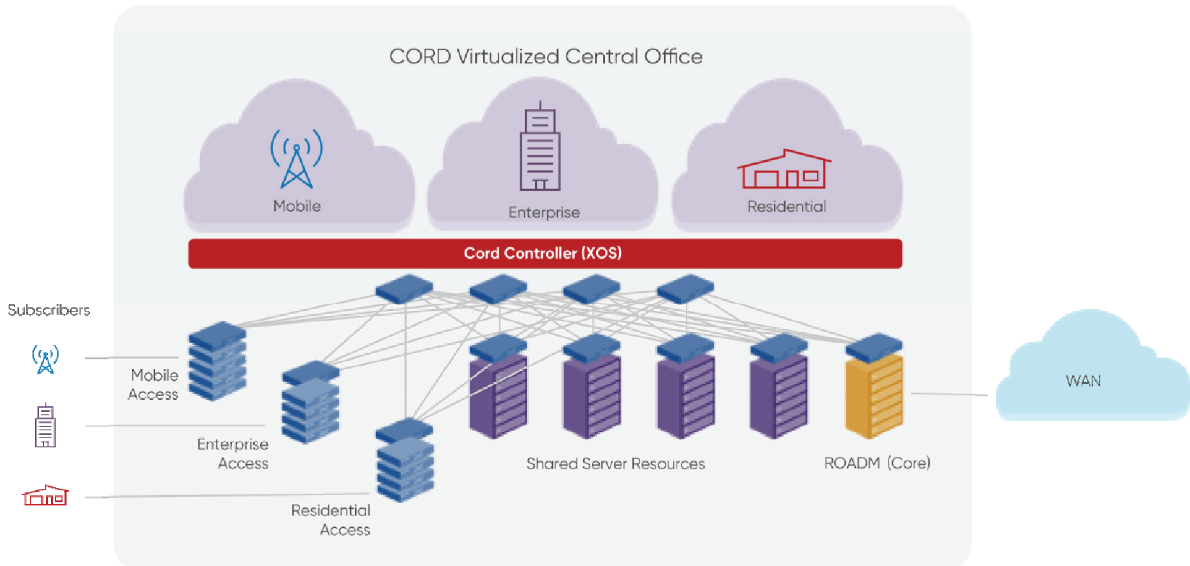

Figure 4. CORD hardware architecture

Although CORD is a large-scale solution, there are now concepts in which it is either an integral part or one of the possible solutions, for example, in the Blue Planet Platform, which offers very wide opportunities for setting up communication between the operator and users. However, these and other similar solutions are quite new and insufficiently tested so they will not be mentioned in this article.

Thus, a system that uses a general CORD-based plan, the main controllers of which are based on the ONOS operating system, which in turn relies on the OpenFlow protocol, which provides a connection to the network elements, is described. The obvious advantages of the system are the versatility described above, high scalability and stability of work, provided by automation of the administration and the exclusion of the human factor. Disadvantages of such a system until recently were the requirements for equipment and a specialist who need to be experienced with the software setup. However, today both minuses are less significant, because, for example, OpenFlow is supported in the vast majority of network devices from major manufacturers, and operators can use various devices as controllers, like not only specialized hardware, but also a fairly common, for example, a personal computer. The requirements for a specialist who need to be experienced are less necessary because of the possibilities of the ONOS system - the programs for network configuration in this system are created on rather simple commands of the Java language, therefore a qualified system administrator is enough effective to work with the controller and companies don't need a specialized programmer [7].

Conclusions. This article describes and analyzes key modern technologies that implement the concept of software-defined networks (SDN). Based on the advantages and expected disadvantages, the conclusion is made on the technical feasibility of creating OpenFlow-ONOS-CORD communication networks. Such a solution should be implemented among ISPs, as well as among diverse users, both private and corporate, this will allow companies to save material costs, since SDNs can be followed by much fewer specialists and allow users to use a stable and fast connection. According to the automation and high speed regulation and connectivity on SDN networks, it is sensibly to assume that this approach to network management will gain popularity in the future, as the number and size of networks are becoming larger and their administration needs more resources.

References

1. Kulakov, Y., Kohan A., Kopychko S. (2019). *Traffic Orchestration in Data Center Network Based on Software-Defined Networking Technology.* In Proceedings of the 2nd International Conference on Computer Science, Engineering and Education Applications (ICCSEEA2019) (pp. 228-237).

2. Лебеденко, Е. В. (2018). *Управление качеством обслуживания в системах информации на основе гистерезисного метода с двумя типами*

порогов. Диссертация на соискание учёной степени кандидата технических наук. Получено из: http://niiae.ru/components/com_chronoforms/uploads/Dissertation//20180427104156_%20.pdf

3. Koponen, T., Casado, M., Gude, N., Stribling, J. (2010). *Onix: A Distributed Control Platform for Large-scale Production Networks.* In Proceedings of the 9th USENIX Symposium on Operating Systems Design and Implementation (pp. 351-364).

4. Shah, S. A. R., Jaikar, A. (2016). *An adaptive load monitoring solution for logically centralized SDN controller*. (from 18th Asia-Pacific Network Operations and Management Symposium (APNOMS). Retrieved from: https://www.researchgate.net/publication/309917884_An_adaptive_load_monitoring_solution_for_logically_centralized_SDN_controller

5. Lantz, B., O'Connor, B., Hart, J., Berde, P. and others. (2014). *ONOS: Towards an Open, Distributed SDN OS.* In the technical program of the ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking (HotSDN 2014). Retrieved from http://delivery.acm.org/10.1145/2630000/2620744/p1-berde.pdf?ip=176.37.26.167&id=2620744&acc=OPENTOC&key=4D4702B0C3E38B35%2E4D4702B0C3E38B35%2E4D4702B0C3E38B35%2EE994ED6114094BD1&__acm__=1556545733_366840876660aed76d8c4ea681cec66b

6. Open Networking Projects. (2017). *CORD.* Retrieved from https://www.opennetworking.org/cord/

7. Van der Meer, S., Grasa, E. (2016). *SDN Architectural Limitations: Towards a Full Software Network Vision.* IEEE Softwarization, A collection of short technical articles, May 2016. Retrieved from: https://sdn.ieee.org/newsletter/may-2016/sdn-architectural-limitations-towards-a-full-software-network-vision

Authors

Cherevatenko Oleksii – student, Department of Computer Engineering, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute".

E-mail: chereva@ukr.net

Череватенко Олексій Володимирович – студент, кафедра обчислювальної техніки, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського».

Yurii Kulakov – professor, Doctor of Engineering Sciences, Department of Computer Engineering, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute".

E-mail: ya.kulakov@gmail.com

Кулаков Юрій Олексійович – професор, доктор технічних наук, кафедра обчислювальної техніки, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського».

РОЗШИРЕНА АНОТАЦІЯ

Олексій Череватенко, Юрій Кулаков

АНАЛІЗ ТЕХНОЛОГІЙ РЕАЛІЗАЦІЇ МЕРЕЖ, ЩО ПРОГРАМНО КОНФІГУРУЮТЬСЯ

Актуальність теми дослідження. Проблема масштабованості комп'ютерних мереж у XXI столітті стала глобальною і вимагає автоматизації управління. Концепція програмно конфігурованих мереж (SDN) може запропонувати це і існує кілька взаємопов'язаних багаторівневих реалізацій, якими є OpenFlow, ONOS і CORD, які розглядаються в цій статті.

Постановка проблеми. До теперішнього часу не існує єдиної лаконічної документації, яка пояснювала б взаємозв’язок між різними рівнями мережі, що програмно конфігурується, та проводив би опис переваг і недоліків кожного з рішень у контексті загальної побудови мережі.

Аналіз останніх досліджень і публікацій. Наразі існує деяка кількість досліджень в області мереж, що програмно конфігуруються, а також технологій, що імплементують концепцію SDN. Однак серед виробників мережевого обладнання немає єдиного стандарту та бачення, як має виглядати практична реалізація SDN, а технології імплементації ще недостатньо вивчені компаніями сфери інформаційних технологій.

Виділення недосліджених частин загальної проблеми. У статті розглядається побудова та взаємозв’язок програмно конфігурованих мереж на базі платформи CORD, у яку входять на більш низьких рівнях система ONOS та протокол OpenFlow.

Постановка завдання. Завданням статті є визначити на основі аналізу переваг і недоліків, чи є технічно та економічно обґрунтованою побудова мережі, що програмно конфігурується, на базі зв’язки різнорівневих технологій OpenFlow–ONOS–CORD.

Викладення основного матеріалу. Описаний основний принцип SDN і технологій їх реалізації OpenFlow, ONOS і CORD. Проведено аналіз переваг і недоліків централізованої архітектури управління програмами для провайдера та клієнтів. Результати аналізу були достатньо інформативними та підтвердили очікувані висновки.

Висновки. У статті було описано та проаналізовано ключові сучасні технології, які реалізують концепцію програмно конфігурованих мереж (SDN). Зроблено висновок про технічну доцільність створення мереж з використанням зв’язки OpenFlow–ONOS–CORD. Припущено, що цей підхід буде набирати популярність у майбутньому.

Ключові слова: програмно конфігуровані мережі, OpenFlow, ONOS, CORD.