Visnyk NTUU KPI Seriia – Radiotekhnika Radioaparatobuduvannia, 2017, Iss. 69, pp. 23–29

УДК 004.043

Application of the "Endless Train" Method for the SDN Controller OpenDayLight

Skulysh M. A.

National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Ukraine

E-mail: *mskulysh@gmail.com*

Application of the "Endless train" method for the SDN controller OpenDayLight. Using the cloud for software defined networks opens new possibilities for the organization of computing processes of network controllers. Methods of virtual entities maintenance in a cloud environment allow counting on relatively infinite computing resources of the cloud service called Platform as a Service. Today it is possible to evaluate the degree of utilization of computing resources for the operation of the software controller only theoretically. The article considers the possibility of using an "endless train" method for virtualization of individual functional opportunities of SDN OpenDayLight controller. This method allows to organize controller computer processes caused by the need to serve the dynamically changing stream of user applications, such as initialization of the flow or the organization of virtual networks. In the section "Architecture and basic features SDN controller" the main components and functionality of the controller OpenDayLight were described. The method of "endless train" for the distribution of tasks between virtual machines and possibility of its use for various functions of the OpenDayLight controller were considered in the section "The method of "endless train" for the organization of the computational process SDN". In the next section "Description of the experiment" the results of simulation modeling of proposed method to study its effectiveness are presented.

Key words: software-defined network; SDN controller; Controller OpenDayLight; SDN network virtualization features

Introduction

In recent years a rapid increase in mobile traffic, and change of its structure have been seen. Modern mobile network should be able to support a large number of subscribers and process large amounts of data. They need to create high-performance clusters and well scalable virtualized environment. All these factors lead to an increase in requirements for network environment and it is often the limiting factor in the development of computing infrastructure.

The main problem of today's mobile networks is that they are too static, therefore they do not meet the current dynamics of the allocation process, unlike servers that are required by this virtualization technology [1]. Nowadays applications are distributed between multiple virtual machines that exchange data very intensively (leading to an increase of east-west traffic, which begins to dominate over traditional client-server north-south traffic). Virtual machines often migrate to optimize the load of servers [2]. It modifies traffic binding points. Traditional addressing scheme, logical separation of networks and methods of appointment rules processing traffic in these dynamic environments become ineffective.

Similar difficulties arise with the reconfiguration mechanisms for Quality of Service (QoS) while adding a new multiservice network application, such as video. Procedures of security settings modification in large networks take too much time that prevents quickly respond to emerging threats [3].

Introduction of technologies of software defined networks and network functions virtualization may be just the factor that would solve existing problems and radically change the approach to the organization and management of the network.

Quality functioning indexes of software defined networks depend on the performance of software modules that implement network functions. It is assumed that the use of cloud technology will provide SDN with necessary computing resources. Today it is possible to evaluate the degree of utilization of computing resources for the operation of the software controller only theoretically. The method of endless train, proposed in this article, will organize computer processes of software defined networks in such a way that fully use the capabilities of modern computing environment.

The purpose of this paper is to describe and to formalize the method which allows to optimize the process of service applications in SDN by increasing the efficiency of the SDN controller work. The algorithm of proposed method of service computing processes in Cloud environments called "endless train"

and the results of experimental studies that confirm the effectiveness of this method were described in this paper.

1 Basic features of SDN controller

Software Defined Network (SDN) is the data network, in which the level of network management is separated from the data transmission device and implemented in software, it is a form of virtualization of computing resources.

Data is transmitted according to the routing tables stored on hardware systems as according to the existing approaches. But these tables are centrally managed by remote system, therefore it is necessary to modify the table on each switch individually. Ideally, all network components must be controlled and adjusted during one operation. Collaboration of software network components can be caused by standards-based OpenFlow.

Key principles of SDN are separation of transmission and management data processes; centralized network management by using standardized software and virtualization of physical network resources. OpenFlow protocol, which implements an independent interface between the logic controller and transport network, is one of the implementations of software defined network [4].

SDN main idea is to separate the traffic management and transfer functions (including control of the traffic and of the devices that carry its transfer). In traditional switches and routers, these processes are inseparable from each other and implemented in a "box": special chip network equipment providing forwarding packets from one port to another, and overlying software determines the rules of the transfer, performs the necessary analysis packet and changes service information in it, and so on. To determine the route of transmission or prevent looping traffic devices certainly exchange data, with the help of different protocols such as Open Shortest Path First (OSPF), Border Gateway protocol (BGP) and Spanning Tree, but each device operates quite autonomously [5].

Fig.1 shows the concept of SDN, according to which all logic control is made in the so-called controllers that are able to monitor the entire network.

The main driving force of the concept of software defined network and basic element of its architecture is OpenFlow. OpenFlow is a protocol, which purpose is to manage network devices with SDN-controller. It provides an opportunity for direct programming of network equipment (such as switches and routers), both physical and virtual, making the network more dynamic and controlled.

The main characteristic of OpenFlow is the use of streams for identifying network traffic. These streams are based on pre-defined rules that can be statically or dynamically programmed using SDN-controller [6].

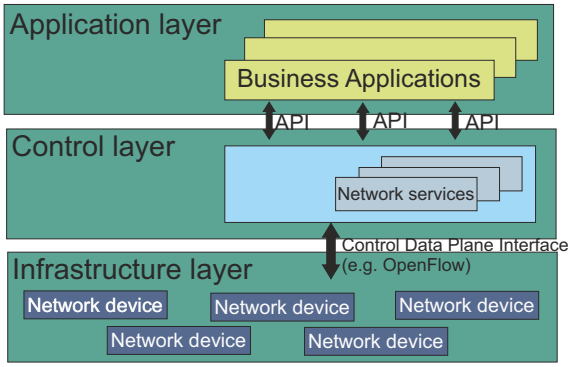

Fig. 1. The concept of software defined network

Consider in more detail the components of SDN controller architecture in order to highlight features that require variable number of resources for implementation.

SDN controller architecture consists of three levels:

1. Southbound plugins and protocols forming the network device layer.
2. Service adaptation and network functions forming the coordination and control layer.
3. Northbound APIs and applications forming the application layer.

The controller acts like middleware in the ecosystem. It is the framework that glues together the applications requiring services of the network devices and the protocols that talk to the network devices for extracting services. The controller allows the applications to be agnostic about the network device specifications, thereby allowing the application developers to concentrate on the development of application functionality rather than writing device-specific drivers. Fig. 2 shows an enhanced architecture of OpenDayLight controller, further, describe all the important components of the architecture.

*Southbound Protocols.*The southbound interface is capable of supporting multiple protocols (as separate plugins), e.g. OpenFlow 1.0, OpenFlow 1.3, BGP-LS, LISP, SNMP, etc. These modules are dynamically linked to a service abstraction layer, which determines how to fulfill the service requested irrespective of the underlying protocol used between the controller and the network devices.

Service Abstraction Layer. In OpenDaylight, the service abstraction layer (SAL) is the key design that enables the abstraction of services between the services' consumers and producers. SAL acts like a large registry of services advertised by various modules and binds them to the applications that require them. Modules providing services, or producers, can register their APIs with the registry. When an application, or a consumer, requests a service via a generic API, SAL is responsible

for assembling the request by binding producer and consumer into a contract, brokered and serviced by SAL. SAL has two architecturally different ways of implementing this registry: application-driven SAL and module-driven SAL.

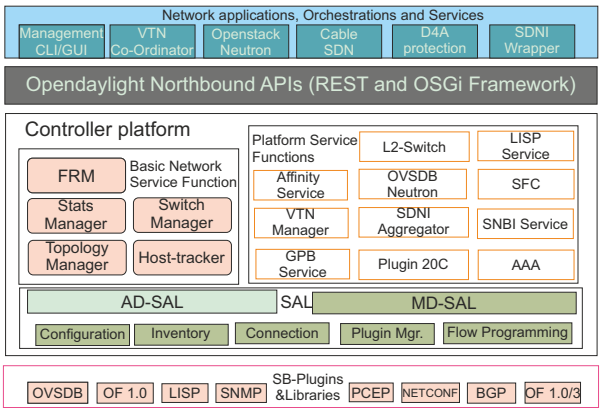

Fig. 2. Enhanced architecture of OpenDayLight controller

Service Functions. The controller has various basic network functions included as part of the shipped base. This includes services for topology discovery and dissemination, a forwarding manager for managing basic forwarding rules, and a switch manager for identifying networking elements in the underlying physical topology. SAL acts as an active registry for brokering contracts between service providers, such as protocol plugins and basic network function plugins, and the service consumers, such as the applications. These contracts are honored by the SAL without any direct dependency on the respective plugins. For example, a topology service plugin is responsible for discovering nodes and physical paths between them to generate a graph. This plugin exposes functions that can be used by an application to get a complete view of the physical layer. The basic service functions are:

1. Topology – a service for learning the network layout by subscribing to events of node addition or removal. Application requiring network view can use this service.

2. Statistics – service for managing stats counters across nodes, flows and queues (ports).

3. Switch Manager – along with the Topology Manager, responsible for storing the nodes discovered on the physical layer.

4. Forwarding – in conjunction with Switch Manager, can provide service for registering and maintaining network flow state. Applications using this can be agnostic to network device specific.

*Vendor Components (Platform Services).*The controller platform itself contains a collection of dynamically pluggable modules to perform needed network tasks. In addition to basic network services, platform-oriented services and other extensions are also added into the controller platform for enhanced SDN functionality. Some of the platform-oriented services are (a) a VTN component to realize a multi-tenant network virtualization application using OpenFlow, (b) affinity services that expose APIs to express workload relationships and service levels, (c) BGP-LS/PCEP to support traffic engineering with BGP-LS (BGP protocol library and topology model) and PCEP (path programming model), (d) an Openstack neutron to provide Neutron API handling for multiple implementations, (e) a group-based policy (GBP) that introduces a novel notion of groups of endpoints and policy (application-focused) abstractions that govern communication between these groups, (f) service function chaining (SFC), which provides the ability to define an ordered list (a chain) of network services, mainly for network function virtualization (NFV), (g) a LISP mapping service that can be used to implement virtual networks, and (h) an SDNi aggregator that collects various information, such as topology, statistics, etc., to enable inter - SDN controller communication [7].

2 Logical functioning architecture of SDN controller

The complexity of the processes taking place with the support of the telecommunications network using SDN technology caused by the high load of SDN controller individual subsystems. The functioning of the controller associated with the provision of certain network functions, which service has a different character.

On Fig. 3 architecture of SDN controller in which all functional elements of conventionally divided into three parts: the functional blocks perform complex calculations, policies and triggers that perform modifying flow switches, etc., based on preset rules and the results of the computer operations functional blocks is proposed. The sequence of operations is set at the level of services abstraction. Thus the controller works as a computer network.

SAL (service abstraction level) determines the path of request through the network. It depends on the type of application and type of computer procedures. SAL - is a key component of the controller responsible for the performance of all network functions. It is a registry of services that connects them with appropriate applications which they need. Modules of service (or service providers) may register their API in the registry. When an application or user requests the service through a common API, SAL is responsible for creating a request, linking the supplier and the consumer in the treaty, which is served by SAL.

Computing node is a system in which realized the principles of cloud storage: set of containers, which are

great physical servers with specified characteristics and the number of computing resources (Fig. 4).

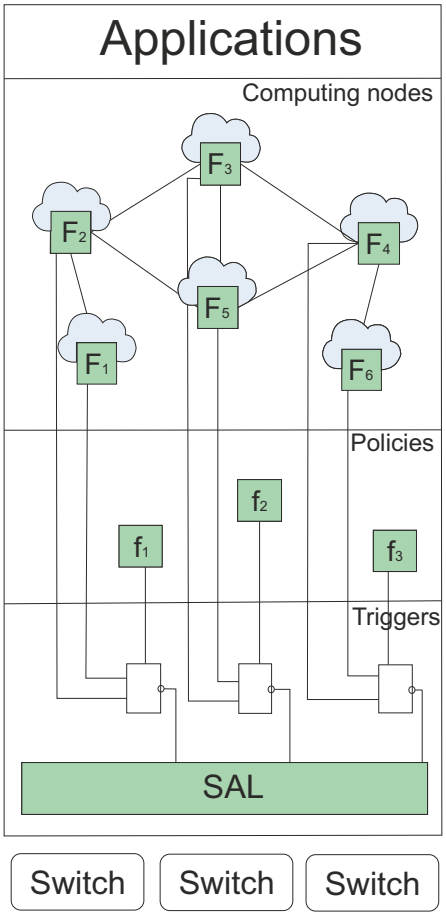

Fig. 3. SDN controller architecture

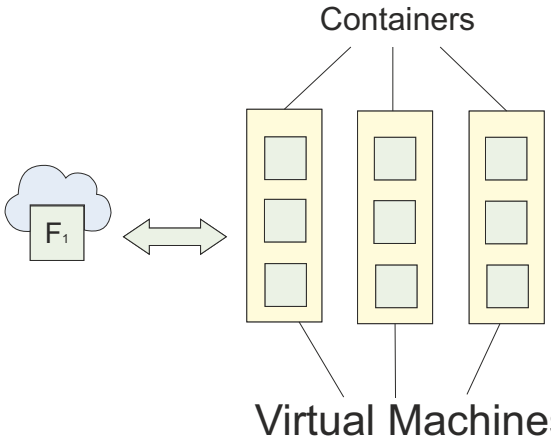

Fig. 4. The structure of the controller computing node

Operations correspond to computing devices in the appropriate virtual machine. Each virtual machine is aimed at performing one narrow task defined by the type of function implemented in this node. The sequence of units bypass depends on the type of procedure (a connection setting, the choice of virtual network topology, etc.).

Computing nodes refer to policies to perform the necessary procedures. These are simple calculation functions that perform the registration of new rules based on network statistics, forecasting network congestion, etc. For example, network reconfiguration is performed by computing nodes based on rules defined by nodes of policies. Triggers are logical devices that make decisions based on a set of rules defined for this unit. Rules can be changed according to the external calculations.

3 The method of "endless train" for the organization of the computational process SDN

Today, there are several approaches to organization of computational process in SDN controller. Their main disadvantage is that they do not consider the possibility of unlimited resources which provide cloud technologies. All existing approaches based on the fact that the amount of system resources is limited and depends on the technical capabilities of the server that hosted virtual machines.

The result of the application of these approaches is reducing of the quality of service at certain time periods that are related to the work of the monitoring system is to collect and analyze data about the network quality of service. If the QoS parameters for certain services are below the threshold, but other services are provided with a sufficiently high quality, the system cannot react to lower quality in a certain period of time, counted as only an average performance.

The solution to this problem is the use of heterogeneous cloud environments, namely certain decisions for platform as infrastructure (PaaS). Platform-as-Infrastructure is an isolated cluster, consisting of a group of servers and services that work together as a complete system, allowing easy deployment, test, maintain and scale the system. PaaS lets you create and maintain an unlimited number of virtual machines whose job is to serve the computing functions of the controller. The use of an unlimited number of resources for virtual machines will avoid periods of reduced quality of service.

The method proposed in this article is called "effect of endless train". Its main idea is that after a certain virtual machine for service received a specified number of tasks, a new virtual machine will be created, which receives all of the following tasks (fig. 5).

The proposed method is based on the method of dynamic migration of virtual machines, developed by Jelastic [8], which provides server load balancing cloud storage by creating a platform for the automatic control of virtual machine containers. Also considered the method of choice container for migration of virtual machines described in [9], which allows you to analyze and predict the load on the network based on an assessment of resources used.

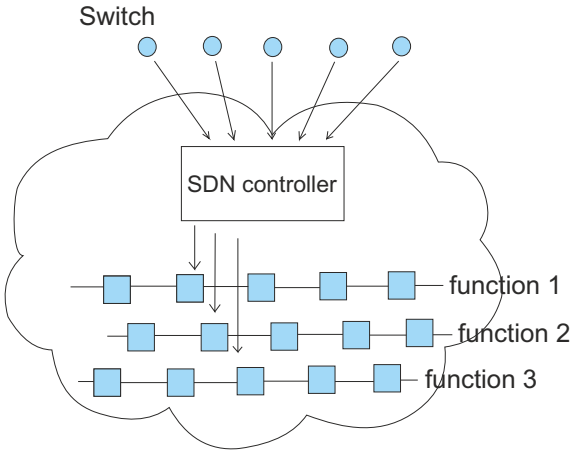

Fig. 5. The method of “endless train”

The use of cloud storage provides that all data is stored on a large number of distributed servers in the network. Two basic tasks must be performed:

- Interactive division of client tasks between virtual machines that are located in one or more clusters. On the one hand it is the task of load balancing, on the other - the task of ensuring the reliability of service.

- Monitoring and management of a cluster of virtual machines. Cluster resources should always be enough for all virtual machines that run on all servers in the cluster at the same time.

The method of “endless train” implies that all requests for service are sent to the current virtual machine as long as it is filled. The number of applications that can be processed by one virtual machine depends on the amount of resources that is allocated to it at creation. After completing this machine, a new one, which will accept all subsequent applications, will be created. Using this method is possible if the amount of system resources is relatively unlimited. The use of heterogeneous cloud environment provides this opportunity [10].

Old virtual machines continue to serve their information flows as long as their number does not run out. Then virtual machines are fold, or wait for re-commissioning as “empty wagon”. The maximum number of requests that can be served in the "wagon depends on the configuration of the cloud platform, or can be obtained by experience. The number of requests depends on the resource volume that controller uses to perform computing tasks. Flexibility of migration processes — technical processes of virtual machines maintenance, is also affects maximum number of requests. Illustration of described method is shown in fig. 6.

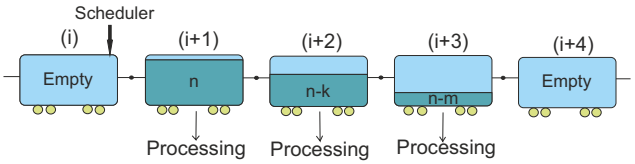

Fig. 6. The effect of “endless train”

n - maximum number of tasks for a single VM; k - number of tasks managed to be processed in the (i + 2) virtual machine since its start of filling to the current moment – the moment of completion for applications in a virtual machine (i + 1); m - number of tasks that managed to be processed in the (i + 3) virtual machine from the start of filling to the current moment.

The method of "Endless train"is a link between level of abstraction and level of computational procedures. The functions of the controller, which implementation requires interaction with a database, you may be considered separate tasks, independent of each other. These functions can be implemented for a given method. These individual tasks can be made in the cloud environment to improve the efficiency of the controller by increasing the amount of computing resources.

The main components of the described system is a set of basic controller functions and a large number of applications that use these functions. Application of the proposed method is appropriate when the system consists of the flow of applications need to be served.

Consider the possibility of using “endless train” method for each of the service functions of the controller.

Topology. Topologies used to store information about network scheme necessary for some applications. The scheme is based on network activity information subscribers exchanged between network nodes. Data from units recorded in a database that is constantly updated. Thus, the function of topology interacts with the database, so the use of endless train method is not considered for it.

Statistics. All applications that work with the function of statistics require computing calculations. The work of statistics function includes processing large amounts of information (data on the network status and activity), execution of computing operations to the received data (calculation of probability and average delay, queue length, etc.) and forecasting network activity based on data obtained two previous stages. Thus, this feature is a stream of requests to perform certain operations that do not require interaction with the database. Therefore, it is appropriate to use the method of “endless train” to increase the speed of processing applications.

Switch Manager. This function is responsible for building a logical network architecture based on statistics function. Physical network topology is used to identify network elements. The method of “endless train” is not applicable for this function.

Forwarding. Forwarding involves searching of the shortest path for the transmission of data flow, analysis of information about reserves and other flow control transactions. The main objectives of this function is the initiation of flow and finding the best way for it. These tasks involve the simultaneous maintenance of a large number of applications and require different execution of computing operations. Therefore, to optimize the flow manager can apply the method of "endless train".

Some vendor components of the controller can also be made in a cloud environment. This applies to the following platform-oriented services as coordinator of Virtual Tenant Network (VTN) and routing protocol BGP.

BGP routing protocol is used by SDN controller as protocol management platform for interaction between physical routers, switches and network services (such as firewall). This approach allows networks exist in various multi-vendor environments without the need to modernize its infrastructure. This module searches for the shortest path for the flow of applications. Thus, each application must refer to it. Therefore, these two components are advisable to implement the method of "endless train" and move some of their functionality modules in the cloud environment.

4 Description of the experiment

To assess the effectiveness of the proposed method the simulation in GPSS environment was conducted. The delay in servicing requests authentication flow, as well as requests to determine the optimal transmission of information flow was estimated. Three methods of controllers work organization were simulated. Namely:

- Centralized planning of controllers load. This method was modeled as a queuing system with limited amount of service devices.
- Cluster controller organization was simplified simulated as the system of queuing with limited amount of service devices, for which it was established label control delay. If specified conditions are not fulfilled, the load between servicing devices is redistributed.
- Method of "endless train" was simulated as the system of queuing with unlimited (very high) amount of service devices, each function processed by a separate group of service devices.

During the experiment, the procedure of connection setting for the subscriber that runs by the SDN controller was modeled. The controller is represented as a queuing system. Each of the elements of the controller was presented as a service unit. The scheme of the procedure is as follows:

1. Unknown packet is sent to the controller;
2. Controller processed the packet; Packet processing can occur in various ways. It depends on the type of operating system, controller, embedded processing algorithms, volume buffer memory.
3. The controller determines the best route across the network (route with the least cost, satisfying routing policies);
4. The relevant OpenFlow policies are set to switches and back route is laid;
5. Data is transmitted to the switch on the specified routes;
6. The switch receives a packet;

OpenFlow switch gets packet through the port specified in the packet header. The packet is processed in accordance with the established priority queues per port switch, packet data recorded in forwarding table. Storing a new entry in the table of forwarding is performed in accordance with the established priority queues beginning to switch. After creating a new record switch performs the data transmission network elements in accordance with the instructions received from the controller.

Modeled process looks as the passing through a sequential chain of service devices. Allocation of resources took place in three different concepts: centralized, clustered and the proposed method of "Endless train."To confirm the effectiveness of the method the value of the average delay in the processing of packets for each of the three concepts was assessed. Input data is the number of service devices, the amount of resources needed to service requests for each of the devices and resources that a proper system can provide, the number of applications that can be served simultaneously on each of the devices to service requests for each stage. The result of the simulation was the average delay in servicing requests for each of the concepts. The result was obtained dependence (fig. 7).

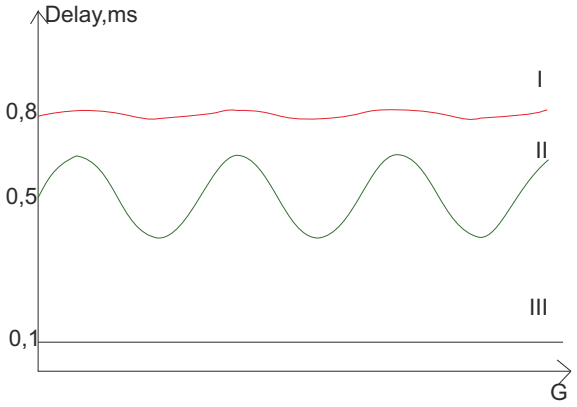

Fig. 7. Experimental time of delay for three different methods

The graph shows that for the method of Endless train delay is the smallest. At the same time limitation of this model is the amount of resources used for

maintenance. This article has not been investigated excessive amount of resources used for initialization and operation of the excess number of virtual machines.

As can be seen from the graphs, the method for Endless train delay was the lowest since the concept involves the use of conditional number of virtual machines, and therefore there in no additional delay caused by overloading of computing resources. Increasing the delay time for the cluster architecture can be explained by fading moments in case of the cluster overloading.

The limitation of the proposed model is the amount of resources used for the service. This article has not been investigated excessive amount of resources used for initialization and operation of the excess number of virtual machines.

Conclusion

To expand computing power of SDN controller it is advisable to use cloud technologies. Removal of all functions of the controller or part of them in the cloud environment will allow to operate conditionally endless resource flows to service requests. It is necessary to consider features of virtual service of entities not only in software but also on the physical level.

This article analyzes the possibility of using cloud technology to increase the amount of computing resources of software defined networks. The method of computing processes service in cloud environments called "endless train", which allows to split the flow of applications for individual units that served in separate virtual machines was proposed. This method allows to avoid the additional delays due to overload computer resources and to improve the network quality of service. The effectiveness of the method was evaluated by means of simulation modeling of the network.

References

[1] Harris M. (2012) *Traditional Network Infrastructure Model and Problems Associated with it*. Pluribus Networks

[2] Jinkyu J., Sung-Hun K. and Hwanju K. (2013) Analysis of virtual machine live-migration as a method for power-capping. *The Journal of Supercomputing*, Vol. 66, No. 3, pp. 1629-1655. DOI: 10.1007/s11227-013-0956-1

[3] Smelyanskii R. (2012) Programmno-konfiguriruemye seti [Software-defined networks]. *Otkrytye sistemy. SUBD*, No 09, pp. 23-26.

[4] McKeown N., Anderson T., Balakrishnan H., Parulkar G., Peterson L., Rexford J., Shenker S. and Turner J. (2008) Openflow: Enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69–74. DOI: 10.1145/1355734.1355746

[5] Chiosi M. (2015) *Network Functions Virtualization: Network Operator Perspectives on Industry Progress*, 20 p. Available at: https://portal.etsi.org/Portals/0/TBpages/NFV/Docs/NFV_White_Paper3.pdf

[6] Lara A. (2013) Network Innovation Using OpenFlow: A Survey. *IEEE Communications Surveys & Tutorials*, Vol. 16, Is. 1, pp. 493-512. DOI: 10.1109/SURV.2013.081313.00105

[7] Sridhar Rao (2015) *SDN Series Part Six: OpenDaylight, the Most Documented Controller*. Available at: https://thenewstack.io/sdn-series-part-vi-opendaylight

[8] Ye K., Jiang X., Huang D. and Wang B. (2013) Live Migration of Multiple Virtual Machines with Resource Reservation in Cloud computing Environments. *Cloud Computing (CLOUD), 2011 IEEE International Conference on*, pp. 267-274. DOI: 10.1109/CLOUD.2011.69

[9] Pahl C. and Xiong H. (2013) Migration to PaaS clouds – Migration process and architectural concerns. *Maintenance and Evolution of Service-Oriented and Cloud-Based Systems (MESOCA)*, pp. 86-91. DOI: 10.1109/mesoca.2013.6632740

[10] Skulysh M.A. (2017) The Method of Computing Organization in High Loaded SDN Controller System. *Experience of Designing and Application of CAD Systems in Microelectronics (CADSM)*. DOI: 10.1109/CADSM.2017.7916080

Застосування методу "Нескінченний потяг" для SDN контролера OpenDayLight

Скулиш М. А.

У статті розглянута можливість застосування методу "нескінченний потяг" для віртуалізації окремих функціональних можливостей SDN контролера OpenDayLight. Даний метод дозволяє організувати обчислювальні процеси контролера, викликані необхідністю обслуговувати потік абонентських заявок, що змінюється динамічно, таких як ініціалізація потоку або організація віртуальних мереж.

Ключові слова: програмно-визначена мережа; контролер SDN; контролер OpenDayLight; віртуалізація мережевих функцій SDN

Применение метода "бесконечный поезд" для SDN контроллера OpenDayLight

Скулыш М. А.

В статье рассмотрена возможность применения метода бесконечный поезд для виртуализации отдельных функциональных возможностей SDN контроллера OpenDayLight. Данный метод позволяет организовать вычислительные процессы контроллера, вызванные необходимостью обслуживать динамически изменяющийся поток абонентских заявок, таких как инициализация потока или организация виртуальных сетей.

Ключевые слова: программно-определенная сеть; контроллер SDN; контроллер OpenDayLight; виртуализация сетевых функций SDN