UDC 004.7:519.87(043.3)

Chang Shu, Ph. D. (Computer Sciences)
(Northwest University of Politics & Law, People Republic of China)

THE MODELS OF DATA FLOWS AND METHODS OF SPECIAL APPLICATIONS OF SOFTWARE-DEFINED NETWORKS

Чанг Шу. Моделі потоків даних і методи спеціальних застосувань визначених програмним забезпеченням мереж. Розглядається новий тип централізовано керованих мереж або мереж, що програмно конфігуруються. Проаналізовані архітектура і основні особливості мереж, що програмно конфігуруються. На основі теорії векторних дискретних напівмарківських випадкових процесів розроблена математична модель потоків даних. Вивчені переваги мереж, що програмно конфігуруються, важливі для використовування в спеціальних системах, зокрема, в системах критичного застосування.

Ключові слова: програмно-конфігурована мережа, SDN, потік даних, математична модель, система критичного застосування випадковий процес

Чанг Шу. Модели потоков данных и методы специальных приложений определенных программным обеспечением сетей. Рассматривается новый тип централизованно управляемых сетей – программно-конфигурируемые сети. Проанализированы архитектура и основные особенности программно-конфигурируемых сетей. На основе теории векторных дискретных полумарковских случайных процессов разработана математическая модель потоков данных. Изучены преимущества программно-конфигурируемых сетей, важные для использования в специальных системах, в частности, в системах критичного применения.

Ключевые слова: программно-конфигурируемая сеть, SDN, поток данных, математическая модель, система критичного применения, случайный процесс

Chang Shu. The models of data flows and methods of special applications of software-defined networks. New kind of centralized controlled networks – Software-Defined networks (SDN) is considered. The architecture and main features of SDN are analysed. Mathematical model of data flows in SDN is developed on the basis of the discrete vector semi-Markov random processes. The advantages of Software-Defined Networks important for application in special systems including systems of critical applications are learned.

***Keywords*:** software-defined networks, SDN, data flow, mathematical model, systems of critical applications, random process

I. Introduction. Over the past several years, software-defined networking (SDN) has emerged as a compelling reason for developing and deploying new network capabilities and services. Centralizing the network control plane – the key idea behind SDN – has led to innovative approaches to traffic engineering, reducing network energy consumption, and data centre network management, e.g., [1, 2]. As SDN applications and configurations continue to grow in scale and complexity, a primary challenge is how to develop and test new designs so that they behave as expected when deployed in live settings and in systems of critical application [3].

In the SDN levels of management by a network and data communications are divided due to the transfer of functions of management (by routers, switches etc.) in applications working on a separate server (controller).

The personal interest of IT-companies in SDN is caused to those that such technologies allow to promote efficiency of network equipment on 25–30%, reduce on 30% of expense on exploitation of networks, convert the management by networks from the art in engineering, to promote safety and give possibility programmatic to create new services and operatively load them in the network equipment to the users [4, 5].

Basic tasks of SDN are:

- division of processes of transmission and data management;
- single, compatible, independent from a supplier interface between the level of management and level of data communication;
- logically centralized management by a network, carried out by a controller with the set network operating system and realized over by network applications;
- virtualisation of physical resources of network.

This work is also related to recent efforts to improve the state of the art in debugging and testing the correctness of SDN applications. Particularly, there are no some mathematical models of traffic flows transfer and routing in SDN. So we consider the details of the design and implementation of semi-Markov model of data distribution and transfer in the software-managed network.

II. Background. In the SDN architecture it is possible to select three levels (Fig. 1).

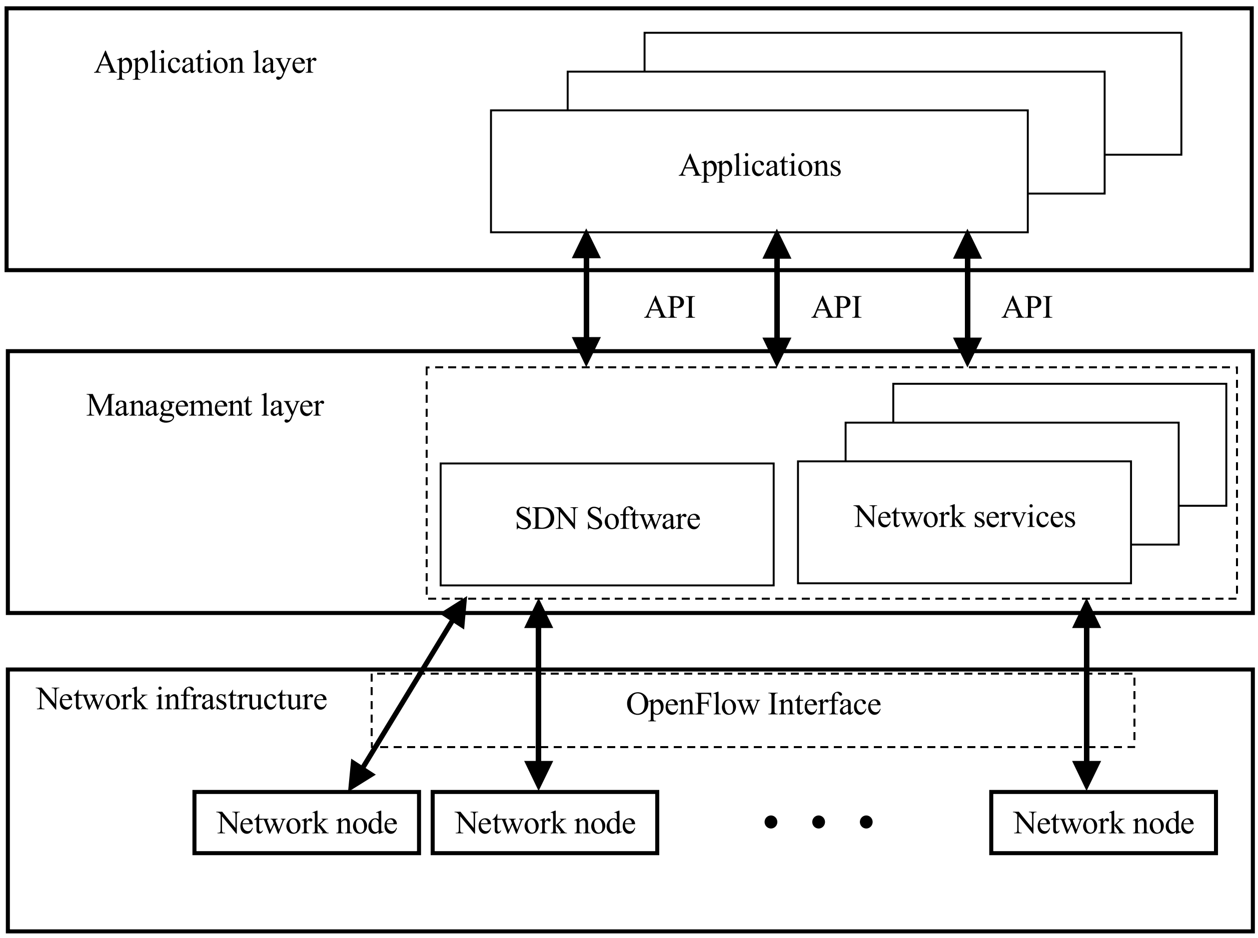

Fig. 1. Architecture of the Software Defined networks. API – applied program interface

1. Level of infrastructure is giving the set of networking issues (switches and data channels).
2. Management level, including the network operating system which provides to applications network services and programmatic interface for the management by the networking issues

and network.

3. Level of network applications for the flexible and effective management by a network.

For SDN OpenFlow is the most perspective and actively developing standard. It is the open standard, in which the requirements, produced to the switch supporting the OpenFlow protocol for the remote management, are described. Modern routers usually decide two basic tasks:

– data forwarding communication is advancement of packet from input port to certain output port. It corresponds to the level of data communication, which collects the tools of transmission (communication lines, path defining equipment, routers, switches);
– data management is treatment of packet and decision-making about that where to pass it further, on the basis of current status of router, and level of management by the state of tools of data communication.

In accordance with specification 1.3 of standard OpenFlow, co-operation of controller with a switch is carried out by means of the OpenFlow protocol is every switch must contain one or more flow tables, group table and to support a channel (OpenFlow channel) for communication with a remote controller – server. Specification does not regulate architecture of controller and API for its applications. Every table of flows in a switch contains the set of records (flow entries) about flows or governed.

The OpenFlow Device consists, at least, from three components:

– flow tables;
– secure channel used for the management by a switch by the peripheral “intelligent” device (by a controller);
– support of the OpenFlow protocol, used for the management.

The logically centralized data management in a network supposes taking away of all functions of management by a network on a separate physical server, urgent by a controller, which is under the authority of administrator of network. A controller can manage both one and by a few OpenFlow-switches and contains the network operating system, giving network services on the low-level management by a network, segments of network and state of network elements, and also applications carrying out the high levelling management by a network and data flows. Network OS (NOS) provides to applications access to the management by a network and constantly watches configuration of tools of network. Unlike traditional interpretation of term of OS, under NOS they understand programmatic system, which is providing monitoring, access and management by the resources of whole network, instead of its concrete node. Like the traditional operating system, NOS provides a programmatic interface for applications of management by a network and will realize the mechanisms of management by the tables of switches: addition, delete, modification collection of various statistics governed. Thus, NOS allows to create applications as the centralized programs using the high levelling names, on the basis of such algorithms, as, for example, algorithm of the Dijkstra search of short cut in a column, in place of the difficult distributed algorithms like the Bellman – Ford algorithm, in terms of low-level addresses which are used in modern routers. For controllers in SDN the requirement is very important of that all applications of one controller at every the instant must have the identical picture of topology of network. As every controller can be connected with a few switches, and every switch – with a few controllers, controllers managing the same switch can be united in a group controller (GK). All controllers of the same GK must have the concerted picture of topology of that part of network, to which they provide access. Such approach to construction of the distributed controller decides the problem of scaled and promotes SDN break tolerance.

One of ideas, actively developed within the framework of SDN, is virtualisation of networks with the purpose of more effective use of network resources. Under virtualisation of network the isolation of network traffic is grouping (multiplexing) of a few data flows with different descriptions are understood. Within the framework of one logical network, which can divide a single physical network with other logical networks or network, cuts (network slices). Every such cut can use addressing, the algorithms of routing, quality of services managements and etc.

Virtualisation of network allows:

- to promote efficiency of allocation of network resources and balance loading on them;
- to insulate the flows of different users and applications within the framework of one physical network;
- administrators of different cuts to use the politicians of routing and rule of flow control;
- to conduct the experiments in a network, using the real physical network infrastructure;
- to use in every cut only those services which concrete applications need.

Administrator can look on the centralized controller the SDN system after all network in single presentation, due to what rise comfort of management, safety and implementation of row of other tasks is simplified. Indeed, as an administrator sees all flows of traffic, it is easier to notice encroachments, to appoint priorities to different types of traffic and develop the rules of reaction of network at congestions and problems with the equipment.

III. Markov model of software-defined data flows. Before introducing notions of update consistency, develop some formal model. Let the sequence of queries and responses of permanent-update requests and atomic operations (e.g., packet arrival and departures at the switching node), is discrete vector semi-Markov process $\mathbf{\Phi}(n)$. We suppose legal preposition that request q_{k-1} precedes request q_k in $\xi(n)$, $k<n$ and the response s_{k-1} precedes the request q_k. Let $\tilde{\xi}(n)$ is a partial order component set of requests in $\mathbf{\Phi}(n)$. A request q_k is considered as complete in $\mathbf{\Phi}(n)$ if process $\mathbf{\Phi}(n)$ contains a response s_k. A complete request is committed in $\mathbf{\Phi}(n)$ if its response is Ack, or cancelled otherwise.

For a process $\mathbf{\Phi}(n)$, let $\Psi_c\left[\mathbf{\Phi}(n)\right]$ denotes the set of rules resulting from sequential composition and non-crossed queue of all committed rules in $\mathbf{\Phi}(n)$ as respects order in $\Psi\left[\mathbf{\Phi}(n)\right]$, i.e., no request q_k is processed before any request q_{k-1} such that q_{k-1}, q_k are strictly ordered pairs.

Let $\Psi_{ic}\left[\mathbf{\Phi}(n)\right]$ is the set of rules that form a series of subsets of incomplete requests in $\mathbf{\Phi}(n)$. Thus, a rule $r_c \varnothing r_{ic}$ such that $r_c \in \Psi_c\left[\mathbf{\Phi}(n)\right]$ and $r_{ic} \in \Psi_{ic}\left[\mathbf{\Phi}(n)\right]$, is an aggregation of all a priory installed rules and a subset of rules that are then being installed in $\mathbf{\Phi}(n)$.

Following [6], we stipulate that every packet considered as the point in the state-space of $\mathbf{\Phi}(n)$ joining the network generates a set of traces: sequences of located packets connected by pseudorandom way, i.e., as numbered list of pairs [packet + port], indicating the order in which the packets traverse the network. Without any loss of generality, we may suppose a path is consistent with the rule r_c if all the events of the path are ordered relatively r_c. Let for a path ζ in the process $\mathbf{\Phi}(n)$ the sub-process $\mathbf{\Phi}_{\zeta}(n)$ denotes the prefix label of $\mathbf{\Phi}(n)$ up to the 1st event of ζ, i.e., up to the moment when the corresponding packet enters the network.

IV. Conclusions. Due to the removal from the switches of loading on treatment of highway of management, SDN allows to these devices to point the all resources at acceleration of moving of traffic, that substantially promotes productivity. Thus due to virtualisation of management a network go down charges on their construction and accompaniment. On results the tests conducted on the base of the largest providers of the USA, the SDN use allows on 20–30% to multiply utilization of the Data Centre resources and in once or twice reduce running expenses. The programmatic tools SDN allow to the administrators to add a new functionality to already present network architecture. Thus new functions will work on many platforms – they will not have to be realized anew in switches software of every supplier. Due to centralized control and data distribution these networks may be successfully applied in special systems, for example, in systems of critical applications.

References

1. Masoud Moshref, Minlan Yu and Ramesh Govindan. Resource/Accuracy Tradeoffs in Software-Defined Measurement \\ ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking (HotSDN). – August 16, 2013. – PP. 73-78.

2. Brandon Heller, Colin Scott, Nick McKeown, Scott Shenker, Andreas Wundsam, Hongyi Zeng, Sam Whitlock, Vimalkumar Jeyakumar, Nikhil Handigol, Murphy McCauley, Kyriakos Zarifis, Peyman Kazemian. Leveraging SDN Layering to Systematically Troubleshoot Networks \\ ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking (HotSDN). – August 16, 2013. – PP. 37-42.

3. Chang Shu. Shaping of traffic parameters and elimination of overload in aeronautical telecommunication networks // Наукові записки Українського науково-дослідного інституту зв'зку. – 2011. – №2(18). – PP. 52–57.

4. Martin Casado, Tal Garfinkel, Aditya Akella, Michael J. Freedman Dan Boneh, Nick McKeown, Scott Shenker. SANE: A Protection Architecture for Enterprise Networks, 15-th Usenix Security Symposium, Vancouver, Canada, August 2006. – 15 PP.

5. N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, J. Turner. Openflow: Enabling innovation in campus networks. SIGCOMM Computer Communication Review, vol. 38, no. 2. – 2008. – PP. 69-74.

6. Тихонов В.И. Марковские процессы / В.И. Тихонов, М.А.Миронов. – М.: Советское радио, 1977. – 488 с.