Modeling of OpenFlow-Based SDN Node with Taking into Account the Differences of Serving TCP and UDP Traffic Streams

Sergey Stepanov, Mikhail Stepanov
Moscow Technical University of Communications and Informatics
Moscow, Russia
stpnvsrg@gmail.com, m.s.stepanov@mtuci.ru

***Abstract*—OpenFlow is one of the widely used protocols controller to switch communications in software-defined networking (SDN). The queueing model that capture the communication between single switch and controller is constructed. In the model two types of traffic flows are used to take into account the differences of serving TCP and UDP traffic streams. The first flow represents the process of coming and serving TCP-packets. The corresponding flow is described by Poisson model. The second flow represents the process of coming and serving UDP-packets. The corresponding flow is also described by Poisson model. It is supposed that switch and controller have finite buffers and maximum allowed time for packets to be in the buffers is restricted. In the model it is assumed that UDP-packets have priority in occupying the switch. It means that when the process of packet servicing is about to complete in the switch, and if there are TCP and UDP packets in the buffer of the switch, the priority will be given to the UDP packets over TCP for servicing. All random variables used in the model have exponential distribution with corresponding mean values. Using the model the main performance measures of interest are given with help of values of probabilities of model's stationary states. The model and derived algorithms of characteristics calculation can be used for estimation of performance characteristics of controller to switch communications and size of buffers.**

I. Introduction

Software-Defined Networking (SDN) is an emerging networking technology that has attracted great interest from both the industry and research societies [1]–[9]. Traditional networks used by majority of providers are static and they can't provide satisfactory Quality-of-Service (QoS) for ever-increasing network multimedia traffic handling. They don't fit for the successful implementation of new IT-services which require more flexibility. SDN technology can solve these problems because it makes networks more scalable, flexible and requires less time for reconfiguration. It takes the network control out of forwarding device and use a separate controller to change the forwarding rules in switches.

In SDN architecture, packet forwarding is based on logically centralized controller usually through the OpenFlow protocol. Such architecture increases the performance of the controller and its capacity which is important, especially for large and highly distributed networks [1]–[3]. OpenFlow protocol is used for data exchange between OpenFlow controller and OpenFLow switch. This protocol allows controller to execute operations of adding, updating and deleting actions for packets in flow tables. The OpenFlow network architecture is shown at Fig. 1.

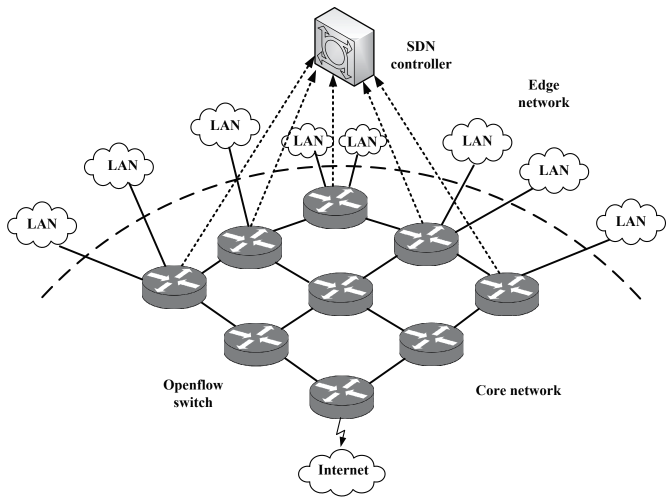

Fig. 1. The OpenFlow network architecture

In SDN, the separation of the control plane from the data plane brings new flexibility in the transportation of traffic packets through the network. This architecture raises the performance bottlenecks in the process of communication in the switch and the controller [3]–[9]. To overcome the consequences of mentioned problems it is necessary to model the controller-to-switch communication for the performance analysis of SDN networks based on OpenFlow principles. A preliminary analysis of the performance of OpenFlow networks is essential for network architects and designers. The results of modeling will give the answers considering how much traffic we can put inside the network, what is the packet service characteristics (mean waiting time, probability of losses, sojourn time); what segment of the network (switch or controller) is the bottleneck.

Because of evident complexity of communications based on SDN-principles the majority of the research results on performance analysis of SDN networks is based on computer simulations [1]–[3], [5], [6]. Compare to investigations based on simulation or practical experiments the analytical modeling is much more efficient because setting up and carry out SDN experiment or performing a computer simulation can

ISSN 2305-7254

take hours for one set of values of input parameters. The real strength of analytical modeling is the fact that analytical model can capture actual OpenFlow working principles of communications and at the same time continue to be flexible enough to process any mixture of query packet flows moving to the switch and controller [4]–[9]. Another good feature of analytical model is the possibility to construct approximate versions of algorithms for cases when more than one switch node is considered in the data plane [10]–[17].

As we have mentioned early because of the complexity of communications based on SDN-principles the analytical modeling of OpenFlow-based networks has only been attempted in a handful of papers [4]–[9]. The obtained results [4]–[9] mostly are devoted to application of closed-form formulas of queueing theory like Jackson model for estimation of performance measures of SDN-networks. This approach restricts the assumptions used in description of controller-to-switch communication.

The aim of the paper is to construct and analyze an analytical framework for modeling the process that capture the communication between single switch and controller of SDN-network. In the model two types of traffic flows are used to take into account the differences of serving TCP and UDP traffic streams. We suppose that the process of coming TCP and UDP-packets is described by Poisson model. It is supposed that switch and controller have finite buffers and maximum allowed time for packets to be in the buffers is restricted. In the model it is assumed that UDP-packets have priority in occupying the switch. All random variables used in the model have exponential distribution with corresponding mean values. The formulated assumptions are based on results of [5],[6] and of course are the limitation of the model but it allows to perform mathematical modeling of the communication between single switch and controller of SDN-network. Using the model the main performance measures of interest are given through values of probabilities of model's stationary states. The model and derived algorithms of performance measures estimation are based on results of [10]–[17] and can be used for estimation of performance characteristics of controller to switch communications and size of buffers. The novelty of the paper compare to [5],[6] consist of in taking into account the differences of serving TCP and UDP traffic streams, the possibility of discarding packet after long waiting time and finite size of buffers.

The rest of the paper is organized as follows. In Section II, the mathematical description of the model will be presented. Here the system of state equations that relates the model's stationary probabilities is outlined and main performance measures and algorithm of their estimation will be defined. Numerical assessment is performed in Section III. Conclusions are drawn in the last section. The functional model of communication of single switch and controller of SDN-network when serving TCP and UDP-packets is shown at Fig. 2.

II. Mathematical description of the model

A. Parameters

Under an OpenFlow network, the controller-to-switch communications taking into account the differences of serving TCP and UDP traffic streams is modeled in the following way. We assume that the overall traffic process of TCP-packets and UDP-packets at the switch follows Poison process with intensities λ_1 and λ_2 correspondingly. Service time of TCP-packet at the switch has exponential distribution with parameter equals to μ_1. If the switch is occupied at the moment of arrival of TCP packet, then the TCP-packet is moved to the buffer (TCP-queue). Let us denote by i_1 the number of waiting TCP-packets and suppose that $i_1 \leq \ell_1$, where ℓ_1 is the maximum number of TCP-packets in TCP-queue. The maximum allowed time for TCP-packet to be in the TCP-queue has exponential distribution with parameter equals to σ_1. If this time has elapsed and TCP-packet is still waiting for servicing, then it leaves the buffer without resuming.

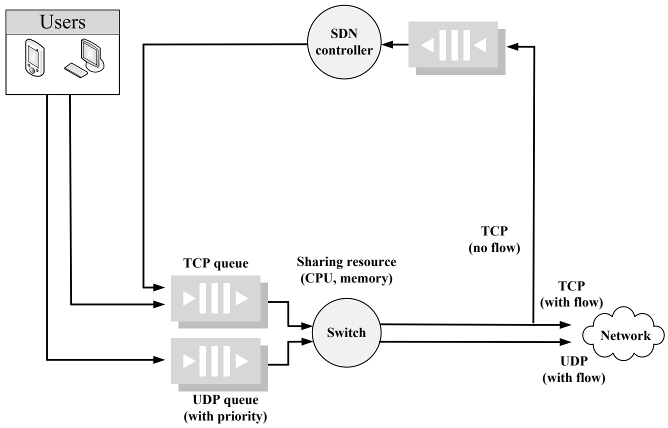

Fig. 2. The functional model of communication of single switch and controller of SDN-network when serving TCP and UDP-packets

Similar assumptions are valid for modeling the servicing of UDP-packets. Let us suppose that service time of UDP-packet at the switch has exponential distribution with parameter equals to μ_2. If the switch is occupied at the moment of arrival of UDP-packet, then the UDP-packet is moved to the buffer (UDP-queue). Let us denote by i_2 the number of waiting UDP-packets and suppose that $i_2 \leq \ell_2$, where ℓ_2 is the maximum number of UDP-packets in UDP-queue. The maximum allowed time for UDP-packet to be in the UDP-queue has exponential distribution with parameter equals to σ_2. If this time has elapsed and UDP-packet is still waiting for servicing then it leaves the buffer without resuming.

Let us denote by i_4 the state of the switch. If i_4 equals to 0 it means that switch is empty. In this case at the switch there are no waiting or serving TCP- or UDP-packets. If i_4 equals to 1 it means that switch is occupied by serving TCP-packet and both buffers may contain waiting TCP- or UDP-packets. If i_4 equals to 2 it means that switch is occupied by serving UDP-packet and both buffers may contain of waiting TCP- or UDP-packets. It is supposed that UDP-packets have priority in occupying the switch. It means that when the process of packet servicing is about to complete in the switch, and if there are TCP and UDP packets in the buffer of the switch, the priority will be given to the UDP packets over TCP for servicing.

Let us define for OpenFlow network by q_{nf} the probability that the TCP-packet goes to the controller in situation when no flow entry can be found in flow-table of the switch. After serving at the controller this packet is served at the switch and goes to the net without visiting the controller again. By applying the approach suggested in [5] instead of using the probability q_{nf} for fresh TCP-packet arrivals we are using in the model the probability $q_c = \frac{q_{nf}}{1+q_{nf}}$ that any TCP-packet after serving at the switch goes to the controller. This assumption simplify the description of the analyzed model. In order to simplify the model it is supposed that UDP-packet after serving at the switch goes to the net without visiting the controller. Here we are taking into account the nature UDP traffic requiring the minimum processing time.

We suppose that service time of TCP-packet at the controller has exponential distribution with parameter equals to μ_3. If at the moment of TCP-packet coming the controller is occupied then TCP-packet moves to buffer. Let us denote by i_3 the number of waiting and serving TCP-packets at the controller and suppose that $i_3 \leq \ell_3 + 1$, where ℓ_3 is the maximum number of TCP-packets in the buffer. The maximum allowed time for TCP-packet to be in the buffer has exponential distribution with parameter equals to σ_3. If this time has elapsed and TCP-packet still waiting for the beginning of servicing it leaves the buffer without resuming. After finishing servicing at the controller TCP-packet moves to the switch for servicing or waiting. If switch has ℓ_1 waiting TCP-packets then coming TCP-packet is lost without resuming.

The mathematical model of communication between single switch and controller of SDN-network when serving TCP and UDP-packets is shown at Fig. 3.

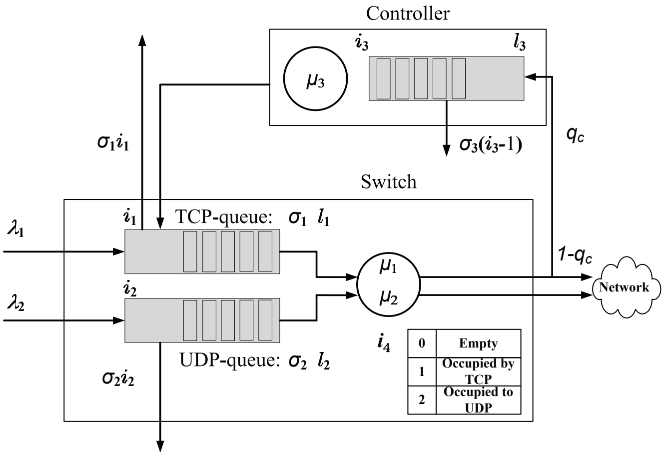

Fig. 3. The mathematical model of communication between single switch and controller of SDN-network when serving TCP and UDP-packets

B. Markov process

Let us denote by $i_1(t)$ and $i_2(t)$ correspondingly the number of TCP-packets and UDP-packets waiting at the switch at time t, by $i_3(t)$ we denote the number of TCP-packets being waiting and servicing at the controller and by $i_4(t)$ we denote the state of the switch at time t. We differ three states of the switch: empty ($i_4 = 0$), occupied by servicing TCP-packets ($i_4 = 1$) and occupied by servicing UDP-packets ($i_4 = 2$). The dynamic of a model states changing is described by multidimensional Markov process with components

$$r(t) = (i_1(t), i_2(t), i_3(t), i_4(t)),$$

defined on the finite set of model's states S. Let us denote by (i_1, i_2, i_3, i_4) the state of $r(t)$, and by S denote the set of all possible states. The vector (i_1, i_2, i_3, i_4) belongs to S when components i_1, i_2, i_3, i_4 varies according to the following rules. If $i_1 + i_2 = 0$ then $i_3 = 0, 1, \ldots, \ell_1;\ \ i_4 = 0, 1, 2$. If $i_1 + i_2 > 0$ then $i_1 = 0, 1, \ldots, \ell_1;\ i_2 = 0, 1, \ldots, \ell_2;\ \ i_3 = 0, 1, \ldots, \ell_3 + 1;\ i_4 = 1, 2$.

Let us denote by $p(i_1, i_2, i_3, i_4)$ the value of stationary probability of state $(i_1, i_2, i_3, i_4) \in S$. It can be interpreted as portion of time the model stays in the state (i_1, i_2, i_3, i_4). This interpretation gives the possibility to use the values of $p(i_1, i_2, i_3, i_4)$ for estimation of model's main performance measures.

C. System of state equations

System of state equations is obtained after equating the intensity of transition $r(t)$ out of the arbitrary model's state (i_1, i_2, i_3, i_4) to the intensity of transition $r(t)$ into the state (i_1, i_2, i_3, i_4). By using the indicator function we represent all equations of the system of state equations in one relation [10], [11]. It will be shown latter that this is very convenient for realizing the iterative algorithms of solving the corresponding system [12], [17]. In the model we have three types of events that can change the state of $r(t)$. The first type is coming of TCP- or UDP-packets. The second type is finishing of service time of TCP- or UDP-packets. The third type is finishing of waiting times of TCP- or UDP-packets. Let us consider in more details the output of realization of these events. Let us define the coefficients for $p(i_1, i_2, i_3, i_4)$ in the left part of the system of state equations.

The coming of TCP or UDP-packets (intensity λ_1 or λ_2) changes the state (i_1, i_2, i_3, i_4) with probability equals to one if switch is empty or switch is occupied and has free waiting positions to accept a packet. Necessary condition of this event is inequality $i_1 < \ell_1$ or $i_2 < \ell_2$ correspondingly. In this case with intensity λ_1 or λ_2 the model state changes from (i_1, i_2, i_3, i_4) to the state with components depending on values of i_1, i_2, i_3, i_4.

The finishing of servicing of TCP or UDP-packet at the switch (intensity μ_1 or μ_2) changes the state (i_1, i_2, i_3, i_4) with probability equals to one if there is a TCP or UDP-packet on servicing. Necessary condition of this event is equality $i_4 = 1$ or $i_4 = 2$ correspondingly. In this case with intensity μ_1 or μ_2 the model state changes from (i_1, i_2, i_3, i_4) to the state with components depending on values of i_1, i_2, i_3, i_4. The finishing of servicing of TCP-packet at the controller (intensity μ_3) changes the state (i_1, i_2, i_3, i_4) with probability equals to one if there is a TCP-packet on servicing. Necessary condition of this event is inequality $i_3 > 0$. In this case with intensity μ_3 the model state changes from (i_1, i_2, i_3, i_4) to the state with components depending on values of i_1, i_2, i_3, i_4.

The ending of waiting of TCP or UDP-packet at the switch (intensity $\sigma_1 i_1$ or $\sigma_2 i_2$) changes the state (i_1, i_2, i_3, i_4) with probability equals to one if there is a TCP or UDP-packet on waiting. Necessary condition of this event is inequality $i_1 > 0$ or $i_2 > 0$ correspondingly. In this case with intensity $\sigma_1 i_1$ or $\sigma_2 i_2$ the model state changes from (i_1, i_2, i_3, i_4) to the state $(i_1 - 1, i_2, i_3, i_4)$ or to the state $(i_1, i_2 - 1, i_3, i_4)$ correspondingly. The finishing of waiting of TCP-packet at the controller (intensity $\sigma_3(i_3-1)$) changes the state (i_1,i_2,i_3,i_4) with probability equals to one if there is a TCP-packet on waiting. Necessary condition of this event is inequality $i_3>1$. In this case with intensity $\sigma_3(i_3-1)$ the model state changes from (i_1,i_2,i_3,i_4) to the state $p(i_1,i_2,i_3-1,i_4)$.

Let us define terms for the right part of the system of state equations. The coming of TCP-packets changes the state (i_1-1,i_2,i_3,i_4) to the state (i_1,i_2,i_3,i_4) with intensity λ_1 if $i_1>0$ and $i_4>0$. As result TCP-packet moves to TCP-queue. The coming of UDP-packets changes the state (i_1,i_2-1,i_3,i_4) to the state (i_1,i_2,i_3,i_4) with intensity λ_2 if $i_2>0$ and $i_4>0$. As result UDP-packet moves to UDP-queue. The coming of TCP-packets changes the state $(i_1,i_2,i_3,0)$ to the state (i_1,i_2,i_3,i_4) with intensity λ_1 if $i_1=0$, $i_2=0$ and $i_4=1$. As result TCP-packet moves to switch. The coming of UDP-packets changes the state $(i_1,i_2,i_3,0)$ to the state (i_1,i_2,i_3,i_4) with intensity λ_2 if $i_1=0$, $i_2=0$ and $i_4=2$. As result UDP-packet moves to switch.

The expiration of waiting of TCP-packet at the switch changes the state (i_1+1,i_2,i_3,i_4) to the state (i_1,i_2,i_3,i_4) with intensity $(i_1+1)\sigma_1$ if $i_1<\ell_1$. As result TCP-packet leaves TCP-queue at switch. The finishing of waiting of UDP-packet at the switch changes the state (i_1,i_2+1,i_3,i_4) to the state (i_1,i_2,i_3,i_4) with intensity $(i_2+1)\sigma_2$ if $i_2<\ell_2$. As result UDP-packet leaves UDP-queue at switch. The finishing of waiting of TCP-packet at the controller changes the state (i_1,i_2,i_3+1,i_4) to the state (i_1,i_2,i_3,i_4) with intensity $i_3\sigma_3$ if $i_3<\ell_3+1$. As result TCP-packet leaves queue at controller.

The finishing of servicing of TCP-packet at the switch changes the state $(i_1+1,i_2,i_3,1)$ to the state (i_1,i_2,i_3,i_4) with intensity μ_1 and probability $(1-q_c)$ if $i_4=1$, $i_1<\ell_1$, $i_2=0$. As result TCP-packet from TCP-queue moves to switch and TCP-packet from switch moves to net. The ending of servicing of TCP-packet at the switch changes the state $(i_1+1,i_2,i_3-1,1)$ to the state (i_1,i_2,i_3,i_4) with intensity μ_1 and probability q_c if $i_4=1$, $i_1<\ell_1$, $i_2=0$, $i_3>0$. As result TCP-packet from TCP-queue moves to switch and TCP-packet from switch moves to controller. The termination of servicing of TCP-packet at the switch changes the state $(i_1+1,i_2,i_3,1)$ to the state (i_1,i_2,i_3,i_4) with intensity μ_1 and probability q_c if $i_4=1$, $i_1<\ell_1$, $i_2=0$, $i_3=\ell_3+1$. As result TCP-packet from TCP-queue moves to switch and TCP-packet from switch dropped because of full queue at controller. The finishing of servicing of TCP-packet at the switch changes the state $(i_1,i_2+1,i_3,1)$ to the state (i_1,i_2,i_3,i_4) with intensity μ_1 and probability $(1-q_c)$ if $i_4=2$, $i_2<\ell_2$. As result UDP-packet from UDP-queue moves to switch and TCP-packet from switch moves to net. The expiration of servicing of TCP-packet at the switch changes the state $(i_1,i_2+1,i_3-1,1)$ to the state (i_1,i_2,i_3,i_4) with intensity μ_1 and probability q_c if $i_4=2$, $i_2<\ell_2$, $i_3>0$. As result UDP-packet from UDP-queue moves to switch and TCP-packet from switch moves to controller. The finishing of servicing of TCP-packet at the switch changes the state $(i_1,i_2+1,i_3,1)$ to the state (i_1,i_2,i_3,i_4) with intensity μ_1 and probability q_c if $i_4=2$, $i_2<\ell_2$, $i_3=\ell_3+1$. As result UDP-packet from UDP-queue moves to switch and TCP-packet from switch dropped because of full queue at controller.

The ending of servicing of UDP-packet at the switch changes the state $(i_1+1,i_2,i_3,2)$ to the state (i_1,i_2,i_3,i_4) with intensity μ_2 if $i_4=1$, $i_1<\ell_1$, $i_2=0$. As result TCP-packet from TCP-queue moves to switch and UDP-packet from switch moves to net. The expiration of servicing of UDP-packet at the switch changes the state $(i_1,i_2+1,i_3,2)$ to the state (i_1,i_2,i_3,i_4) with intensity μ_2 if $i_4=2$, $i_2<\ell_2$. As result UDP-packet from UDP-queue moves to switch and UDP-packet from switch moves to net. The termination of servicing of TCP-packet at the controller changes the state (i_1-1,i_2,i_3+1,i_4) to the state (i_1,i_2,i_3,i_4) with intensity μ_3 if $i_1>0$, $i_3<\ell_3+1$. As result TCP-packet from controller moves to TCP-queue at the switch. The finishing of servicing of TCP-packet at the controller changes the state (i_1,i_2,i_3+1,i_4) to the state (i_1,i_2,i_3,i_4) with intensity μ_3 if $i_1=0$, $i_2=0$, $i_3<\ell_3+1$, $i_4=1$. As result TCP-packet from controller moves to switch.

The expiration of servicing of TCP-packet at the switch changes the state $(i_1,i_2,i_3,1)$ to the state (i_1,i_2,i_3,i_4) with intensity μ_1 and probability $(1-q_c)$ if $i_1=0$, $i_2=0$, $i_4=0$. As result TCP-packet from switch moves to net. The ending of servicing of TCP-packet at the switch changes the state $(i_1,i_2,i_3-1,1)$ to the state (i_1,i_2,i_3,i_4) with intensity μ_1 and probability q_c if $i_1=0$, $i_2=0$, $i_3>0$, $i_4=0$. As result TCP-packet from switch moves to controller. The termination of servicing of TCP-packet at the switch changes the state $(i_1,i_2,i_3,1)$ to the state (i_1,i_2,i_3,i_4) with intensity μ_1 and probability q_c if $i_1=0$, $i_2=0$, $i_3=\ell_3+1$, $i_4=0$. As result TCP-packet from switch moves to controller and dropped.

The finishing of servicing of UDP-packet at the switch changes the state $(i_1,i_2,i_3,2)$ to the state (i_1,i_2,i_3,i_4) with intensity μ_2 if $i_1=0$, $i_2=0$, $i_4=0$. As result UDP-packet from switch moves to net. The ending of servicing of TCP-packet at the controller changes the state (i_1,i_2,i_3+1,i_4) to the state (i_1,i_2,i_3,i_4) with intensity μ_3 if $i_1=\ell_1$, $i_3<\ell_3+1$, $i_4>0$. As result TCP-packet from controller moves to switch and dropped.

By equating the left and right parts of system of state equations we obtain the following relation valid for all $(i_1,i_2,i_3,i_4)\in S$.

$$
\begin{gathered}
P(i_1,i_2,i_3,i_4)\Big\{\lambda_1 I(i_1<\ell_1)+\lambda_2 I(i_2<\ell_2)+ \\
+\mu_1 I(i_4=1)+\mu_2 I(i_4=2)+\mu_3 I(i_3>0)+ \\
+\sigma_1 i_1 I(i_1>0)+\sigma_2 i_2 I(i_2>0)+\sigma_3(i_3-1)I(i_3>1)\Big\}= \\
=P(i_1-1,i_2,i_3,i_4)\lambda_1 I(i_1>0,i_4>0)+ \\
+P(i_1,i_2-1,i_3,i_4)\lambda_2 I(i_2>0,i_4>0)+ \\
+P(i_1,i_2,i_3,0)\lambda_1 I(i_1=0,i_2=0,i_4=1)+ \\
+P(i_1,i_2,i_3,0)\lambda_2 I(i_1=0,i_2=0,i_4=2)+ \\
+P(i_1+1,i_2,i_3,i_4)(i_1+1)\sigma_1 I(i_1<\ell_1)+ \\
+P(i_1,i_2+1,i_3,i_4)(i_2+1)\sigma_2 I(i_2<\ell_2)+ \\
+P(i_1,i_2,i_3+1,i_4)i_3\sigma_3 I(i_3<\ell_3+1)+ \\
+P(i_1+1,i_2,i_3,1)\mu_1(1-q_c)I(i_4=1,i_1<\ell_1,i_2=0)+ \\
+P(i_1+1,i_2,i_3-1,1)\mu_1 q_c I(i_4=1,i_1<\ell_1,i_2=0,i_3>0)+ \\
+P(i_1+1,i_2,i_3,1)\mu_1 q_c I(i_4=1,i_1<\ell_1,i_2=0,i_3=\ell_3+1)+ \\
+P(i_1,i_2+1,i_3,1)\mu_1(1-q_c)I(i_4=2,i_2<\ell_2)+
\end{gathered}
\tag{1}
$$

$$+P(i_1,i_2+1,i_3-1,1)\mu_1 q_c I(i_4=2,i_2<\ell_2,i_3>0)+$$
$$+P(i_1,i_2+1,i_3,1)\mu_1 q_c I(i_4=2,i_2<\ell_2,i_3=\ell_3+1)+$$
$$+P(i_1+1,i_2,i_3,2)\mu_2 I(i_4=1,i_1<\ell_1,i_2=0)+$$
$$+P(i_1,i_2+1,i_3,2)\mu_2 I(i_4=2,i_2<\ell_2)+$$
$$+P(i_1-1,i_2,i_3+1,i_4)\mu_3 I(i_1>0,i_3<\ell_3+1)+$$
$$+P(i_1,i_2,i_3+1,i_4)\mu_3 I(i_1=0,i_2=0,i_3<\ell_3+1,i_4=1),$$
$$+P(i_1,i_2,i_3,1)\mu_1(1-q_c)I(i_1=0,i_2=0,i_4=0),$$
$$+P(i_1,i_2,i_3-1,1)\mu_1 q_c I(i_1=0,i_2=0,i_3>0,i_4=0),$$
$$+P(i_1,i_2,i_3,1)\mu_1 q_c I(i_1=0,i_2=0,i_3=\ell_3+1,i_4=0),$$
$$+P(i_1,i_2,i_3,2)\mu_2 I(i_1=0,i_2=0,i_4=0),$$
$$+P(i_1,i_2,i_3+1,i_4)\mu_3 I(i_1=\ell_1,i_3<\ell_3+1,i_4>0).$$

Values $P(i_1,i_2,i_3,i_4)$ are satisfying to the normalizing condition

$$\sum_{(i_1,i_2,i_3,i_4)\in S} P(i_1,i_2,i_3,i_4)=1.$$

D. Performance measures

The model performance measures can be defined by summing probabilities $p(i_1,i_2,i_3,i_4)$ over corresponding subsets of S. Let us define by L_1 the mean number of waiting TCP-packets at the switch, by L_2 define the mean number of waiting UDP-packets at the switch, by L_3 define the mean number of waiting TCP-packets at the controller. The introduced characteristics are looking in the following way

$$L_1=\sum_{\{(i_1,i_2,i_3,i_4)\in S\ |\ i_1>0\}} p(i_1,i_2,i_3,i_4)i_1;$$

$$L_2=\sum_{\{(i_1,i_2,i_3,i_4)\in S\ |\ i_2>0\}} p(i_1,i_2,i_3,i_4)i_2;$$

$$L_3=\sum_{\{(i_1,i_2,i_3,i_4)\in S\ |\ i_3>1\}} p(i_1,i_2,i_3,i_4)(i_3-1).$$

The mean number y_1 of TCP-packets being on service at the switch, the mean number y_2 of UDP-packets being on service at the switch and the mean number y_3 of TCP-packets being on service at the controller may be written as

$$y_1=\sum_{\{(i_1,i_2,i_3,i_4)\in S\ |\ i_4=1\}} p(i_1,i_2,i_3,i_4);$$

$$y_2=\sum_{\{(i_1,i_2,i_3,i_4)\in S\ |\ i_4=2\}} p(i_1,i_2,i_3,i_4)i_2;$$

$$y_3=\sum_{\{(i_1,i_2,i_3,i_4)\in S\ |\ i_3>0\}} p(i_1,i_2,i_3,i_4).$$

Let us denote by Λ_1 and by Λ_2 the intensity of TCP-packets or TCP-packets correspondingly left the system without servicing because of insufficient number of waiting positions or expiring the maximum allowed waiting time. The introduced characteristics are define in the following way

$$\Lambda_1=\lambda_1\sum_{\{(i_1,i_2,i_3,i_4)\in S\ |\ i_1=\ell_1\}} p(i_1,i_2,i_3,i_4)+ \quad (2)$$
$$+\sum_{\{(i_1,i_2,i_3,i_4)\in S\ |\ i_4=1;\, i_3=\ell_3+1\}} p(i_1,i_2,i_3,i_4)\mu_1 q_c+$$
$$+\sum_{\{(i_1,i_2,i_3,i_4)\in S\ |\ i_1=\ell_1;\, i_3>0\}} p(i_1,i_2,i_3,i_4)\mu_3+L_1\sigma_1+L_3\sigma_3;$$
$$\Lambda_2=\lambda_2\sum_{\{(i_1,i_2,i_3,i_4)\in S\ |\ i_2=\ell_2\}} p(i_1,i_2,i_3,i_4)+L_2\sigma_2.$$

The ratio π_t of lost TCP-packets and the ratio π_d of lost UDP-packets may be written as

$$\pi_t=\frac{\Lambda_1}{\lambda_1};\quad \pi_d=\frac{\Lambda_2}{\lambda_2}. \quad (3)$$

The probability π_{tb} and π_{db} of losses for TCP-packets and UDP-packets correspondingly because the buffer is full can be defined as

$$\pi_{tb}=\sum_{\{(i_1,i_2,i_3,i_4)\in S\ |\ i_1=\ell_1\}} p(i_1,i_2,i_3,i_4);$$

$$\pi_{db}=\sum_{\{(i_1,i_2,i_3,i_4)\in S\ |\ i_2=\ell_2\}} p(i_1,i_2,i_3,i_4).$$

Let us denote by T_t and T_b the mean time TCP-packets or UDP-packets correspondingly being in the system. The values of T_t and T_d can be calculated in the following way

$$T_t=\frac{L_1+L_3+y_1+y_3}{\lambda_1(1-\pi_{tb})};\quad T_d=\frac{L_2+y_2}{\lambda_2(1-\pi_{db})}. \quad (4)$$

E. Solution of the system of state equations

Performance measures introduced in the previous chapter are based on the values of stationary probabilities $p(i_1,i_2,i_3,i_4)$. To calculate $p(i_1,i_2,i_3,i_4)$ it is necessary to solve the system of state equations (1). This system doesn't have any special features that simplify the solution. Because of this reason the values of $p(i_1,i_2,i_3,i_4)$ can be found by Gauss-Zeidel iterative algorithm [10],[11]. Let us denote by $L(i_1,i_2,i_3,i_4)$ the coefficient for probability $P(i_1,i_2,i_3,i_4)$ in the left part of (1) and by $R(P(j_1,j_2,j_3,j_4))$ denote the right part of (1). The left part is a function of the components i_1,i_2,i_3,i_4 of state $(i_1,i_2,i_3,i_4)\in S$. The right part is a function of probabilities $P(j_1,j_2,j_3,j_4)$ listed in the right part of (1).

Let us define by $P^{(s)}(i_1,i_2,i_3,i_4)$ approximation number s for unnormalized values of $P(i_1,i_2,i_3,i_4)$ obtained by Gauss-Zeidel iterative algorithm, $(i_1,i_2,i_3,i_4)\in S$. To find the values of $P(i_1,i_2,i_3,i_4)$ according to Gauss-Zeidel iterative algorithm the following recursion formula is used

$$P^{(s+1)}(i_1,i_2,i_3,i_4)=\frac{1}{L(i_1,i_2,i_3,i_4)}\times \quad (5)$$

$$\times R(P^{(s,s+1)}(j_1, j_2, j_3, j_4)),$$

where $(i_1, i_2, i_3, i_4) \in S$. Upper index $(s, s+1)$ in $P^{(s,s+1)}(j_1, j_2, j_3, j_4)$ means usage in process of calculation the already found approximation $P^{(s+1)}(j_1, j_2, j_3, j_4)$ and if it unknown to this moment then the usage of known approximation $P^{(s)}(j_1, j_2, j_3, j_4)$. The initial approximation can be found from relations

$$P^{(0)}(i_1, i_2, i_3, i_4) = 1, \qquad (i_1, i_2, i_3, i_4) \in S.$$

At each step of iterative algorithm the convergence is checked with help of relation

$$\frac{\sum\limits_{(i_1,i_2,i_3,i_4)\in S} \left|P^{(s+1)}(i_1, i_2, i_3, i_4) - P^{(s)}(i_1, i_2, i_3, i_4)\right|\Big)}{\sum\limits_{(i_1,i_2,i_3,i_4)\in S} P^{(s+1)}(i_1, i_2, i_3, i_4)} \le \varepsilon,$$

where ε is taken from interval $10^{-6} \ldots 10^{-10}$. In order to guarantee the convergence it is sufficient to put one of unknown probabilities $P(i_1, i_2, i_3, i_4)$ in (1) to one and after to solve obtained heterogeneous system of linear equations by iterative algorithm (5).

III. Numerical assessment

In this section, we apply the constructed analytical model and elaborated algorithms to produce the quantitative and qualitative analysis of the dependence of model's performance measures on the values of input parameters and taking into account the differences of serving TCP and UDP traffic streams in particular the priority of serving UDP-packets, finite buffers and restriction on maximum allowed time for packets to be in the buffers. Another important area of model application is the estimation of the switch and controller throughput required for serving incoming traffic with given values of performance indicators. Let us consider a few numerical examples that illustrate the solution of the listed problems. The values of performance measures are obtained after solving the system of state equations (1) and using the definitions of characteristics through values of stationary probabilities formulated in subsection II-D.

Let us suppose that packet processing time at the switch $\frac{1}{\mu_1} = 10$ μs [5]. We accept as a unit of time the mean duration of service a packet at the switch. The values of model's parameters in chosen time units are taken as follows: $\lambda_1 =$ 0,3; $\lambda_2 =$ 0,15; $\sigma_1 = \sigma_2 = \sigma_3 =$ 0,1; $\mu_1 = \mu_2 = 1$; $\mu_3 =$ 0,1; $\ell_1 = 4$; $\ell_2 = 5$; $\ell_3 = 6$; $q_{nf} =$ 0,2.

We begin the model's numerical assessment with Fig 4 that presents the values of π_t and π_d defined by formulas (3) vs. the value of switch throughput expressed in Gb/s according to the excepted mean time of serving TCP-packet of size 1500 byte. According to the presented results the values of packet blocking are decreasing with increasing of switch throughput. Obtained curves can be used for estimation the switch throughput required for serving incoming traffic with given values of π_t and π_d if their values are chosen as performance indicators. As additional measure of quality of serving we can chose the mean values of time packets stay in the considered system. The values of T_t and T_d vs switch throughput measured in Gb/s are shown on Fig 5.

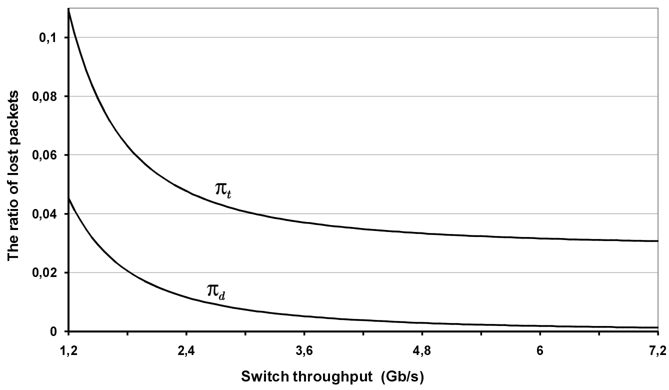

Fig. 4. The results of estimation of π_t and π_d vs switch throughput measured in Gb/s

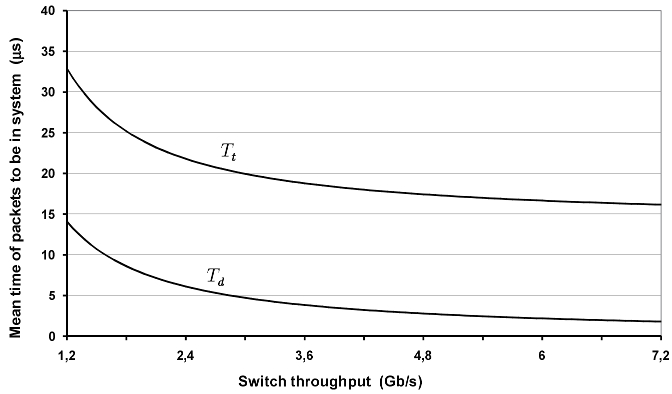

Fig. 5. The results of estimation of T_t and T_d vs switch throughput measured in Gb/s

Fig. 4 and 5 shows that for some moment increasing of switch throughput decreases the values of chosen performance indicators. Further this decreasing is stabilized because system bottleneck now will be controller throughput. Let us denote as Λ_{1k} the k-th term $k = 1, 2, 3, 4, 5$ in the expression for Λ_1 defined by (2). Then Λ_{11} is the intensity of blocked TCP-packets because of no free space in the switch buffer, Λ_{12} is the intensity of blocked TCP-packets because of no free space in the controller buffer, Λ_{13} is the intensity of blocked TCP-packets after serving at the controller because of no free space in the switch buffer, Λ_{14} is the intensity of leaving of blocked TCP-packets after unsuccessful waiting at the switch buffer and Λ_{15} is the intensity of leaving of blocked TCP-packets after unsuccessful waiting at the controller buffer. The values of Λ_{1k}, $k = 1, 2, 3, 4, 5$ vs switch throughput measured in Gb/s are shown on Fig 6.

We see that the main contribution into Λ_1 gives the intensity of of leaving of blocked TCP-packets after unsuccessful waiting at the controller buffer. It is clear that to decrease this value it is necessary to increase controller throughput.

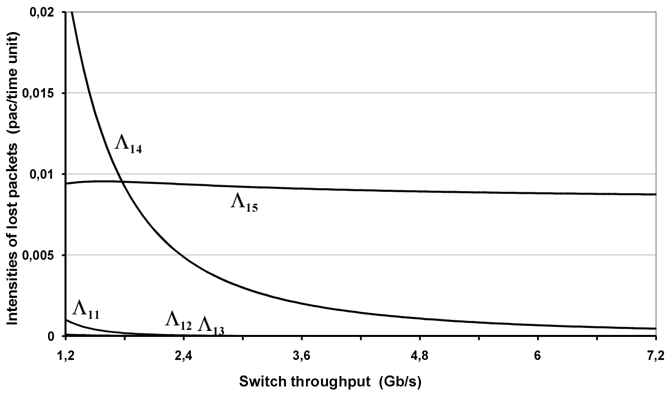

Fig. 6. The intensities of lost TCP-packets depending on the different reasons taken into account in the model vs switch throughput measured in Gb/s

IV. Conclusion

In this work we have proposed an analytical model for an OpenFlow enabled software-defined networking based on construction and analysis of corresponding Markov process that capture the communication between single switch and controller. In the model two types of traffic flows with taking into account the differences of serving TCP and UDP traffic streams are considered. The process of coming and serving TCP-packets and UDP-packets is described by Poisson model. It is supposed that switch and controller have finite buffers and maximum allowed time for packets to be in the buffers is restricted. In the model it is assumed that UDP-packets have priority in occupying the switch. It means that when the process of packet servicing is about to complete in the switch, and if there are TCP and UDP packets in the buffer of the switch, the priority will be given to the UDP packets. All random variables used in the model have exponential distribution with corresponding mean values. Using the model the main performance measures of interest are given with help of values of probabilities of model's stationary states.

The model and derived algorithms of performance measures estimation can be used to produce the quantitative and qualitative analysis of the dependence of model's performance measures on the values of input parameters with taking into account the differences of serving TCP and UDP traffic streams in particular the priority of serving UDP-packets, finite buffers and restriction on maximum allowed time for packets to be in the buffers.

The effects of key SDN network parameters are studied which include the ratio of lost TCP-packets and UDP-packets, the mean time of TCP-packets and UDP-packets being in the system and intensities of lost TCP-packets depending on the reasons of losses taken into account in the model. The constructed analytical framework additionally offers the possibility to find the values of switch and controller throughput required for serving incoming TCP and UDP traffic streams with given values of performance indicators and to find the size of buffer and maximum allowed time for packets to be in the buffers that permits to serve traffic flows with acceptable value of delay instead of increasing throughput. Proposed model can be fuhrer developed to include the possibility of packets batch arrivals.

Acknowledgment

The research is partly supported by the Russian Foundation for Basic Research, project no. 16-29- 09497ofi-m.

References

[1] F. Hu, Q. Hao, K. Bao. "A survey on software-defined network and openflow: From concept to implementation", *IEEE Commun. Surv. Tutor.*, vol. 16, No 4. 2014, pp. 2181-2206.

[2] A. Hakiri, A. Gokhale, P. Berthou, D.C. Schmidt, T. Gayraud. "Software-defined networking: Challenges and research opportunities for future internet", *Comput. Netw*, vol.75. No 24. 2014, pp. 453-471.

[3] M. Jammal, T. Singh, A. Shami, R. Asal, Y. Li. "Software defined networking: State of the art and research challenges", *Comput. Netw.*, vol. 72, No 29. 2014, pp. 74-98 .

[4] Q. Zuo, M. Chen, P. Jiang. "Delay evaluation of openflow control plane by queue model", *J. Huazhong Univ. Sci. Technol. (Nat. Sci. Ed.)*, vol. 8, No 1. 2013, pp. 44-49.

[5] K. Mahmood, A. Chilwan, O. Osterbo, M. Jarschel. "On the modeling of OpenFlow-based SDNs: the single node case", *Proceedings of the 6th International Conference on Networks and Communications (NeCoM 2014).*, 2014, pp. 207-214.

[6] K. Mahmood, A. Chilwan, O. Osterbo, M. Jarschel. "Modelling of open flow-based software-defined networks: the multiple node case", *IET Netw.*, vol 4, No 5. 2015, pp. 278-284.

[7] B. Xiong, K. Yang, J. Zhao, W. Li, K. Li. "Performance evaluation of OpenFlow-based software-defined networks based on queueing model", *Computer Networks*, vol. 102, No C. 2016, pp. 172–185.

[8] J. Ansell, W. K.G. Seah, B. Ng, S. Marshall. "Making Queueing Theory More Palatable to SDN/OpenFlow-based Network Practitioners", *NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium*. 2016, pp. 1–6.

[9] J. Raychev, G. Hristov, D. Kinaneva, P. Zahariev "Modelling and Evaluation of Software Defined Network Architecture Based on Queueing Theory", *2018 17th International Conference on Information Technology Based Higher Education and Training (ITHET)*. 2018, pp.1–5.

[10] S.N. Stepanov. *Osnovy teletraffika multiservisnykh setei (Fundamentals of Multiservice Networks)*. Eqo-Trends. Moscow, 2010, (in Russian).

[11] S.N. Stepanov. *Teoriya teletraffika: kontseptsii, modeli, prilozheniya (Theory of Teletraffic: Concepts, Models, Applications)*. Goryachaya Liniya-Telekom. Moscow, 2015, (in Russian).

[12] S.N. Stepanov "Model of Joint Servicing of Real-Time Service Traffic and Data Traffic. I", *Automation and Remote Control*, vol. 72. No 4. 2011, pp. 787–797.

[13] S.N. Stepanov "Model of Joint Servicing of Real-Time Service Traffic and Data Traffic. II", *Automation and Remote Control*, vol. 72. No 5. 2011, pp. 1028–1035.

[14] S.N. Stepanov, M.S. Stepanov. "Planning Transmission Resource at Joint Servicing of the Multiservice Real Time and Elastic Data Traffics", *Automation and Remote Control*, vol. 78. No 11. 2017, pp. 2004–2015.

[15] S.N. Stepanov, M.S. Stepanov. "The Model and Algorithms for Estimation the Performance Measures of Access Node Serving the Mixture of Real Time and Elastic Data", *In: Vishnevskiy V., Kozyrev D. (eds) Distributed Computer and Communication Networks. DCCN 2018. Communications in Computer and Information Science (CCIS)*. Springer, LNCS, vol. 919. 2018, pp. 264–275.

[16] S.N. Stepanov, M.S. Stepanov. "Planning the Resource of Information Transmission for Connection Lines of Multiservice Hierarchical Access Networks", *Automation and Remote Control*, vol. 79. No 8. 2018, pp. 1422–1433.

[17] S.N. Stepanov, A.M. Romanov, M.S. Stepanov, A.P. Vasilyev. "Analytical modeling of call admission control for real-time and elastic services in inter satellite link", *In Proc. of 2018 Systems of Signals Generating and Processing in the Field of on Board Communications*. IEEE, Moscow, Russian Federation, Moscow Technical University of Communication and Informatics. 2018, pp. 1–5. DOI: 10.1109/SOSG.2018.8350637.