Evaluation of Telecommunication Equipment Delays in Software-Defined Networks

I. G. Buzhin, Yu. B. Mironov
Moscow Technical University of Communication and Informatics
Moscow, Russia
bigvrn93@mail.ru, mistiam@gmail.com, buzhin@media-publisher.ru

Abstrat — **Mathematical models of the switch and controller of software-defined networks are built, on the basis of which formulas are derived for estimating the time delays in the telecommunications equipment of these networks. The dependencies of the time delays in the switch and the controller on the main equipment parameters, which have a major effect on the delays, are also obtained. The obtained results can be used in the design of software-configured networks in the context of various informational impacts (for various input information flows) with a given level of quality of service.**

Key words: software defined network, telecommunication equipment, time delay, modern networking technology.

I. Introduction

At the present stage of development of communication networks, a number of tasks of the session and application levels are superimposed on the telecommunications equipment of the transport and channel levels. This situation leads either to a significant increase in the cost of telecommunications equipment, or to the occurrence of congestion and lower quality of services provided.

In this regard, there is a need to separate control functions and data transfer functions between different types of telecommunications equipment. This need has led to the development of technology software-defined networks (SDN) [1].

In modern communication networks, including software-defined ones, there are high demands on the quality of telecommunications services in the context of information exposure. The quality of telecommunication services is a combination of the properties of a service, determined by the quality of the functioning of the network, which characterize the ability to meet the needs of users. Quality of service is a set of indicators that determine the degree of user satisfaction with the service provided to him. Quality of service is characterized by the properties of usability, availability, availability, continuity, integrity and security of service. The quality of the network operation determines the ability of the network to perform functions that provide communication between subscribers.

One of the characteristics of the quality of telecommunications services is the delay created by telecommunications equipment.

Thus, when designing software- defined networks in the context of various informational impacts (with different input information flows) with a given level of quality of service, it is necessary to evaluate time delays in telecommunications equipment.

II. Architecture of Software-Defined Networks

The main idea of the SDN is to separate the traffic transfer functions from the control functions (including control of both the traffic itself and its transmitting devices). This is due to the creation of special software that can work on a separate server (computer) and which is controlled by the network administrator. All routers and switches are combined under the control of a SDN controller or Network Operating System (SOS), which provides applications with access to network management and which constantly monitors the configuration of network facilities.

Thus, the control switch function is transferred to a separate central device - the SDN controller. This approach allows you to manage and monitor the state of the network on a logically centralized controller. In addition, it becomes possible for a control level to separate from the physical component using the logical representation of the network as a whole. The interaction between the level of data transmission is carried out through a single unified open interface.

In the SDN architecture, according to [1], there are 3 levels (Fig. 1):

- network infrastructure layer, representing a set of network structures (switches and communication channels);
- a management level consisting of an operating system that provides applications with network services and a software interface for managing network devices and the network;

application level.

978-1-7281-0606-9/19/$31.00 ©2019 IEEE

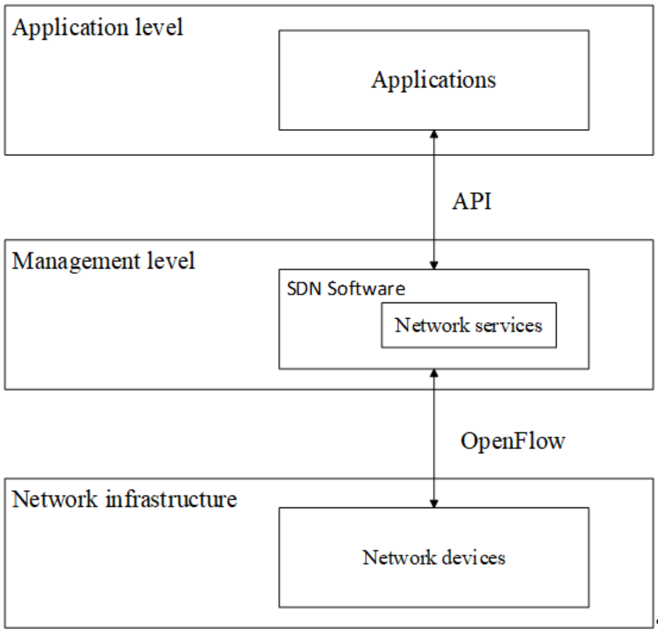

Fig. 1. Architecture SDN

One of the central places in the architecture of the SDN is the network operating system, which in a general sense is an operating system that provides processing, storage and transmission of data in the information network. A network operating system defines an interconnected group of upper-level protocols that provide basic network functions: object addressing, service operation, data security, and network management.

In contrast to the traditional interpretation of the term SOS as an operating system integrated with the network protocol stack, in this case, SOS means a software system that provides monitoring, access, resource management of the entire network, and not a specific node. It is in this definition that SOS can be considered a SDN controller. The operating system does not manage the network, it provides a programming interface (API) for management. Thus, in fact, the solution of network management tasks is placed on the level of applications implemented based on the network operating system API.

The technology of software-defined networks simplifies the process of operation and maintenance of networks, connecting to networks automatically. This technology is widely used in the Internet of Things. It is assumed that in the future “things” will become active participants in business, informational and social processes, where they can interact and communicate with each other, sharing environmental information, reacting and influencing the processes occurring in the outside world, without human intervention. Many Internet of Things devices are serviced by radio technologies that use unlicensed radio frequency bands and are designed to communicate in small radius cells with limited QoS and security requirements applicable to the home environment.

III. Principle of Functioning Telecommunication Equipment of Software-Defined Networks

The separation of the level of control and data transfer is solved within the framework of the SDN. In the SDN switch, only the data transfer level is implemented. Each SDN switch has several input and output ports. Switch ports are connected to ports of another switch by physical communication channels. The management port is connected to the SDN controller node — OpenFlow exchanges messages on this channel to control the switch.

The controller does not manage the network, it provides a programmatic interface (API) for management. Thus, in fact, the solution of network management tasks is placed on the level of applications implemented on the basis of the network operating system API.

The SDN controller generates data on the state of all network resources and provides access to them for network management applications. These applications manage various aspects of network operation, including building topologies, routing decisions, load balancing.

To implement this idea, an open protocol OpenFlow has been developed for managing network equipment that is not focused on products from a specific supplier. Using this protocol, specialists themselves can determine and control which nodes, under what conditions and with what quality they can interact in the network.

The switch is equipped with a set of addressing tables (flow tables) that form the addressing pipeline (pipeline), which consists of one or more serially connected addressing tables. A packet arriving at one of the input ports of the switch is first processed (information is read from the packet header), then it goes to the addressing pipeline. The sequential processing of the packet in the addressing tables begins. The package here refers to a bit string from which two parts can be distinguished: a header and a payload. The operations performed on packets in the addressing tables do not change the load of the packet, but are capable of changing its header.

The package header consists of several fields. As a rule, these fields indicate the identifiers of the network protocols that must process the packet and the service information used by them. During the packet passing through the addressing tables pipeline, service fields (metadata) can be added to the packet header, which are intended to transfer service information inside the pipeline, and are reset when a packet arrives at one of the output ports of the switch. The composition and size of service fields is determined by the technical characteristics of a specific SDN switch. Thus, data transfer decisions are made on the basis of streams, which are a combination of packet header fields.

The addressing table contains flows records and instructions for applying actions to a packet. Stream entries and relevant information from the packet headers begin to match. If a match is found, the corresponding action is taken on the packet. It may result in removing a packet from the switch, switching to another (larger by number) addressing table, sending it to the controller via a secure OpenFlow channel, or sending a packet to the desired output port of the SDN switch. If the switch cannot execute the instruction of applying the action to the packet, then the packet information is sent to the controller via the secure OpenFlow channel. If the information from the packet header did not match any entry about the streams, then the packet is transmitted to the next addressing table.

The controller manages the contents of the addressing tables of the subordinate switches. The OpenFlow protocol assumes that the main reason for a change in the contents of the switch tables is the controller's reaction to events from the network, and involves several types of notifications so that the switches can notify the controller of such events, as well as several types of commands with which the controller can make modifications to the switch tables.

The switch processes each message received from the controller, with the possibility of forming a response, if necessary. If the switch cannot fully process the message received from the controller, it should send back an error message to the controller. This can occur due to a switch reset, QoS policy, or if it is sent to a blocked or faulty port.

The OpenFlow protocol provides reliable message delivery and processing, but does not provide automatic delivery confirmations or orderly processing of messages. Message processing is provided for the primary connection and additional connections that use reliable data transfer, but is not supported on additional connections that use unreliable data transfer. Message delivery is guaranteed as long as the OpenFlow channel is operational.

IV. ASSEMENT OF DELAYS OF TELECOMMUNICATION EQUIPMENT OF SOFTWARE-DEFINED NETWORKS

A mathematical model of the functioning of the SDN switch and the SDN controller was built in [2]. Based on this model, we obtain the formulas for estimating delays in the telecommunications equipment of software-defined networks.

The quality of telecommunication services is a combination of the properties of a service, determined by the quality of the functioning of the network, which characterize the ability to meet the needs of users. Ensuring the quality of telecommunications services is the purpose of designing communication networks. One of the factors that determines the quality of service (QoS) is the delay - the time it takes for a packet (message) after it is sent to reach its destination. When the telecommunications equipment of the network is functioning, a switching delay is formed (the time it takes for the device that received the packet (message) to start transmitting it to the next telecommunications device.

The model of the functioning of the SDN switch is a queuing network consisting of two parts (separate queuing systems): the first part is the input queue of the switch and the device for reading information from the packet header, the second part is the flow addressing table. Two independent Poisson flows of intensity λ_1 and λ_2. requests come to the input of the queuing network. These streams are numbered in order of decreasing importance of applications. At the time of termination of the service on the released device, an application is selected from a non-empty queue with a minimum number. Incoming requests with intensity λ_1 correspond to packets arriving at the control port of the switch (from the controller), requests with intensity λ_2 correspond to the rest of the packets arriving at the switch from external networks. The service times of requests for each of the streams are independent in aggregate and have an exponential distribution function. If all devices are busy, then the incoming application is waiting in the drive, while the applications of both streams form a common queue. The first priority applications (1-applications or priority applications) are packets arriving at the SDN switch from the controller. They have a relative priority compared to applications of the second stream (2-applications, or non-priority applications). Then the average waiting time w_1 of the priority application in the first part of the switch operation model is estimated using formula (1):

$$w_1 = \frac{Q_1}{\lambda_1 (1 - \pi_1)} \tag{1}$$

where Q_1 - stationary average queue length of priority requests (packets arriving at the switch from the controller), π_1 - probability of losing priority claims in the first part of the switch model.

The average waiting time of a non-priority application is estimated using formula (2):

$$w_2 = \frac{Q_2}{\lambda_2 (1 - \pi_1)} \tag{2}$$

where Q_2 - stationary average queue length of non-priority requests (packets arriving at the switch from external networks).

The second part of the switch operation model is a single-line queuing system that receives a Poisson flow with intensity Λ_1 (intensity of the output stream of the first part of the switch model). The processing of a packet by each record

$i, i=\overline{1,S}$ and the execution of further actions on the packet will be represented by the servicing device with an average time of matching $1/\sigma$ the packet with the record of the flows. Then the delay in the second part of the switch model is estimated as (3):

$$w_3 = \frac{S}{\sigma} \tag{3}$$

where S - number of entries in the stream addressing table.

Then the total delay in the SDN switch will be estimated using (4):

$$w = w_1 + w_2 + w_3 \tag{4}$$

Central to the architecture of the SDN is the SDN controller. This device manages the contents of the tables of addressing flows and lists of rules (actions) of the subordinate switches. Changing the addressing tables of switches occurs based on the state of the network and messages coming from the switches via the OpenFlow channel. We will build the model of the functioning of the SDN controller without choosing a specific network operating system. The algorithm for the operation of the SDN controller described below does not depend on the specific network operating system and is common to all network operating systems. The network operating system will influence the choice of model input parameters.

The input of the controller receives messages from the OpenFlow channel from g the switches. The controller recognizes all these messages and sends them to the controller core, which decides what needs to be done in the appropriate situation. Next, a response to this message is generated, which will contain a list of actions that the controller must perform. These actions can be: change of addressing tables, deletion or addressing of a packet to another switch, etc. The controller core selects the appropriate action from the actions database, which is located on the controller. It contains all sorts of actions that the controller can choose. Next, the response is sent to the switch, where the actions that are specified in this response are required to be performed.

The algorithm for the functioning of the SDN controller can be described by a linear queuing system with r waiting places ($r < \infty$), into which the Poisson message flow enters the OpenFlow channel from g the switches. The service times of messages (applications) are independent and, at the same time, the service time of each application on any of the c devices is distributed exponentially with a parameter γ. A message arriving in an overcrowded system (i.e., when all c devices and all r waiting places are occupied) is lost and is not returned to it again. The service device simulates the core processor processing SDN controller.

For the model of the SDN controller, the delay will be estimated by the stationary distribution of the message residence time in the model. Note that a message that is made when i other messages are received in the system immediately begins to be serviced if $i < c$, and the time it takes to service the fully loaded message system $i-c+1$ if $c \le i < c+r$ it waits for service to start. With a fully loaded system, messages exit from it through exponentially distributed times with the parameter $c\gamma$. Then the waiting time for the start of the service of the message $c+i, 0 \le i < r$, which made the messages in the system, is distributed according to the Erlang law $E_{i+1}(x)$ with the parameters $c\gamma$ and $i+1$. Using the formula for the total probability and taking into account that the distribution function of the stationary distribution $W_{con}(x)$ of the waiting time for the maintenance of the message received in the system is conditional, we obtain:

$$\begin{aligned} W_{con}(x) &= \frac{1}{1-\pi_{con}}\left[\sum_{i=0}^{c-1} p_i + \sum_{i=0}^{r-1} p_{c+i} E_{i+1}(x)\right] = \\ &= \frac{1}{1-\pi_{con}}\left[P_{w=0} + \sum_{i=0}^{r-1} p_{c+i} E_{i+1}(x)\right] \end{aligned} \tag{5}$$

where p_i - probability of i messages are in the system at the moment t, $P_{w=0}$ - probability of serving a message without waiting in the queue, π_{con} - message loss probability.

Moving on to the Laplace-Stieltjes transformation:

$$\begin{aligned} \omega_{con}(s) &= \int_0^{\infty} e^{-sx} dW_{con}(x) = \\ &= \frac{1}{1-\pi_{con}}\left[P_{w=0} + c\gamma p_c \sum_{i=0}^{r-1} \frac{\lambda_{con}^i}{(s+c\gamma)^{i+1}}\right] = \\ &= \frac{1}{1-\pi_{con}}\left[P_{w=0} + c\gamma p_c \frac{1-\left(\frac{\lambda_{con}}{s+c\gamma}\right)^r}{s+c\gamma-\lambda_{con}}\right] \end{aligned} \tag{6}$$

where λ_{con} - intensity of incoming messages to the controller input.

Hence, taking into account the independence of the service time from the waiting time for the start of service, it follows that the stationary distribution $W_{con}(x)$ of the residence time in the system of the message accepted for service has the form (7):

$$\varphi_{con}(s)=\int_0^\infty e^{-sx}dW_{con}(x)=$$

$$=\frac{1}{1-\pi_{con}}\left[P_{w=0}+c\gamma p_c\frac{1-(\frac{\lambda_{con}}{s+c\gamma})^r}{s+c\gamma-\lambda_{con}}\right]\frac{\gamma}{s+\gamma} \quad (7)$$

Stationary average waiting time for service start :

$$\omega_{con}=\varphi'_{con}(0)=$$

$$=\frac{c\gamma-\left(\frac{\rho}{c}\right)^r\left[(r+1)c\gamma-r\lambda_{con}\right]}{(c\gamma-\lambda_{con})^2}\frac{p_c}{1-\pi_{con}} \quad (8)$$

where $\rho=\lambda_{con}/\gamma$.

The stationary average message dwell time in the system is calculated using the Little formula:

$$w_{con}=\omega_{con}+\frac{1}{\gamma} \quad (9)$$

V. Analysis of Dependence of Delays on Parametrs of Telecommunication Equipment

Based on the formulas obtained for estimating the delays of telecommunications equipment of software-defined networks, we obtain the diagrams of dependencies of delays on the parameters of telecommunications equipment.

Based on formula (4), we construct a graph of the dependence (Fig.2) of the total delay w on the parameter $\rho_1=(\lambda_1+\lambda_2)/\mu$, which is defined as the ratio of the intensity of the incoming flow to the intensity of packet processing.

The main influence on the delay in the SDN switch is provided by the input stream intensity, the processing rate of incoming packets, and the number and number of entries in the addressing table. The values of the parameters for plotting the dependence (Fig. 2) are taken from datasheets [4, 5]. Based on Fig.2, with an increase in the ratio of the intensity of the incoming flow to the intensity of packet processing, there is an increase in the total delay in the SDN switch.

Based on formula (9), we construct a graph of dependence (Fig. 3) (delays (average residence time w_{con} in the SDN controller) against the number of processor processor cores.

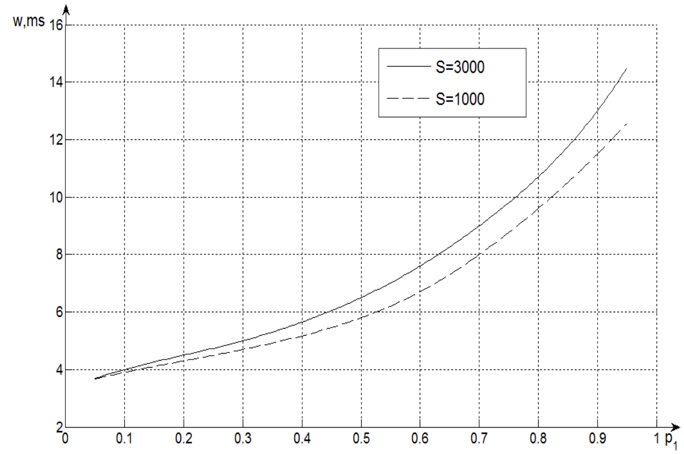

Fig. 2. The graph of the total delay w switch SDN from ρ_1

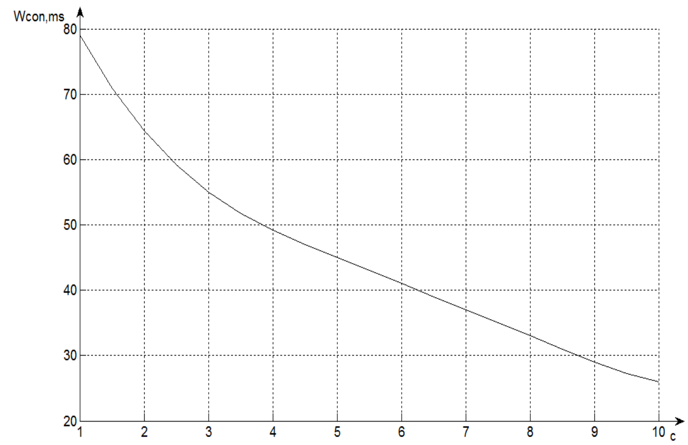

Fig. 3. The graph of the delay w_{con} controller SDN from the number c of logical processor cores

The greatest impact on the delay in the SDN controller is its input queue, which is characterized by the average waiting time for service to start (9), and the processor of processing messages and generating responses to requests from SDN switches, which is characterized by the average message processing time.

The values of the parameters for plotting the dependence (Fig. 3) are taken from datasheets [6]. On the basis of Fig.3, with an increase in the number c of logical cores of the message processing processor and the formation of responses to switch requests, the time delay in the SDN controller decreases.

VI. CONCLUSION

In this article, formulas (4), (9) were obtained for estimating the time delays in telecommunications equipment of software-defined networks, namely, in the SDN switch and the SDN controller.

The dependence of the time delay in the SDN switch on the ratio of the intensity of the incoming flow to the intensity of packet processing was obtained. As a result, it was found that with an increase in the ratio of the intensity of the incoming flow to the intensity of packet processing, there is an increase in the total delay in the SDN switch.

Also, the dependence of the average message dwell time in the SDN controller on the number of logical cores of the packet processing processor and the formation of responses by the SDN switch was obtained. As a result, it was found that with an increase in the number of logical cores of the message processing processor and the generation of responses to switch requests, the time delay in the SDN controller decreases.

The obtained results can be used in the design of software-defined networks in the context of various informational impacts (for various input information flows) with a given level of quality of service.

REFERENCES

[1] ONF TR-502: SDN Architecture [Electronic resource] // Open Networking Foundation. – URL: https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/TR_SDN_ARCH_1.0_06062014.pdf (date of the application: 01.02.2019).

[2] K. E. Samouylov, I. A. Shalimov, I. G. Buzhin, Y. B. Mironov, “Model of functioning of telecommunication equipment for software-configurated networks,” Modern Information Technologies and IT-Education, vol. 14, no. 1. 2018. doi:10.25559/SITITO.14.201801.013-026.

[3] V. Vishnevskiy, “Theoretical foundations of computer network design,” Moscow: The technosphere, 2003, 512 p

[4] D-link. Gigabit Stackable Smart Managed Switches [Electronic resource]: Electronic text data. – D-link, 2015.

[5] HP Performance Brief for External Audiences [Electronic resource]: Electronic text data. – HP, 2007.

[6] OpenDaylight Controller:MD-SAL:FAQ [Electronic resource] // wiki.opendaylight.org. - URL: https://wiki.opendaylight.org/view/OpenDaylight_Controller:MD-SAL:FAQ (date of the application: 09.02.2019).

[7] ONF TR-539 OpenFlow Controller Benchmarking Methodologies [Electronic resource] // Open Network Foundation. – URL: https://3vf60mmveq1g8vzn48q2o71a-wpengine.netdna-ssl.com/wp-content/uploads/2014/10/TR-539_OpenFlow_Controller_Benchmarking_Methodologies_v1.pdf date of the application: 06.05.2018).