

Modeling and Assessment Performance of OpenFlow-Based Network Control Plane

Samer Salah Al_Yassin

Assistant Teacher, Al_Mamon University College

Abstract: *The subject of the paper is the features of network architecture that built on the concept of Software-Defined Networking (SDN). The main communicating processes of OpenFlow protocol are determined. OpenFlow is one of the most commonly used protocols for communication between controller and forwarding element (OpenFlow switch) in SDN. Efficiency characteristics of an OpenFlow-based network are also identified. Mathematical tools and analytical model for analysis and evaluation such characteristic as performance and reliability of OpenFlow-based network are also suggested in the paper. The proposed model is based on querying theory (M/M/1 queues considered) and distribution information models. The evaluation performance and reliability of network is analyzed due to capture the interaction between the OpenFlow switch and controller. The average packet executed time, sojourn time, waiting probability metrics of control plane are suggested for evaluation.*

Keywords: Software-defined networking, OpenFlow protocol, performance, blocking probability, analytical model.

1. Introduction

Today many big corporations and organizations are being used clouding computing technology. Cloud computing gives opportunity to increase amount of different types of services, business and users applications. Therefore, multiple applications are being migrated from traditional desktop solutions to centralized Web-based ones. However, traditional computing networks not ready to solve all tasks that connected with effective management of multiples distributed applications (traditional routers and switches should maintain and handle many different types of protocols).

A new paradigm that called software-defined networking (SDN) was created to solve these problems. SDN propose to split the control plane from the data plane [1]. In this way, many new opportunities opens in the area of cloud computing. The main advantages of SDN are a flexible adaptation of network resources to the services requirements, networks virtualization, centralized management process, remote administration, etc.

However, the centralized management mechanism faces a number of challenges: the number of processed requests and quality of service depend on the speed of type of controller (query processing, load distributing). According to the requirements of specification controller is a key element that determines the performance of the major part of cloud computing [2]. For example, two of the most common used SDN controller create quote delays of around 0.01 milliseconds and a throughput 50 000 flows per second (NOX) and offers delays of around 0.06 milliseconds and a throughput of around 31,000 flows per second (POX).

To effective future functioning of the SDN networks should to be analyzed such characteristics as performance and reliability. Also the Focus Group of Cloud computing have been formulated requirements to the quality of services in such types of networks [3]. These characteristics depend primarily on the number of controlled switches and routers,

intensity of the incoming requests and processing time by the controller and provide the appropriate quality of service.

In this way important to model the controller-switch interaction for the performance analysis of SDN networks. The modeling of SDN networks and performance estimation will help to the answer questions such as how many applications and data can be transferred, what is the packet sojourn time, what is the bottleneck in a network. Thus, the modeling processes for SDN-based network and evaluation performance, reliability of network that built on SDN concept are important objects for new designed networks architecture and existing network too.

2. Overview of SDN. Openflow Protocol

The SDN architecture consists of tree main elements: a controller – centralized element, a switch – middle forwarding devise (software program or hardware device that forwards packets) and end node – customer’s devises (virtual machine, mobile devises, computer etc) [1]. End node devises belong to data plane, controllers and switches – to control plane of SDN architecture. The effective interaction between controller and switches be able due to OpenFlow protocol [4].

The OpenFlow is a core of Software-Defined Networking. OpenFlow protocol enables controllers to determine the path and method to pass packets through the network of switches [5]. The protocol offers a higher flexibility in the routing of network flows and the freedom to change the behavior of a part of the network without influencing other traffic.

On the switch, there is a flow table, which matches packets to its entries based on their source, destination Ethernet address, IP address or port, IP type of service, ICMP code/type, VLAN ID or incoming port [6]. The controller is a software process that listens for OpenFlow messages. The Figure 1.1 depicts the common SDN or OpenFlow-based network architecture and interaction between elements.

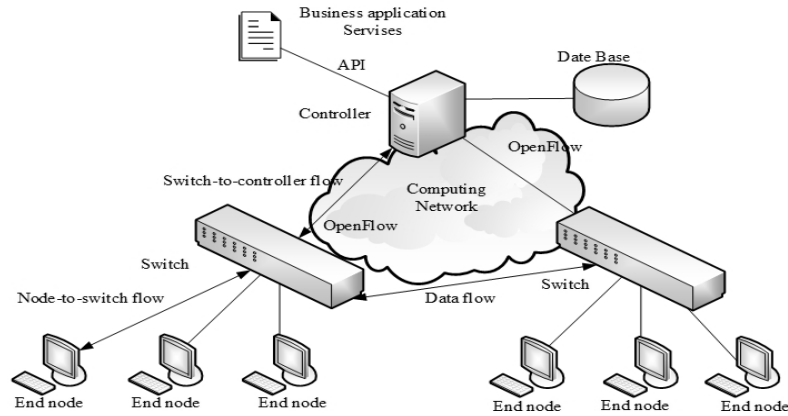


Figure 1.1: Software-Defined Networking architecture

According to requirement of OpenFlow protocol specification the controller-switch interaction takes place as follows:

- 1) Packets from end devices arrive to the switch port.
- 2) Switch checking the match entries;
- 3) Switch listed in the relevant flow table entry are executed if the match is find in flow table. The actions available are forward or drop packets;
- 4) Switch sent packet to the controller if the match is not find in flow table;

- 5) The controller computes the forwarding path and updates the required nodes in the data path by sending entries will add to the flow tables.
- 6) Subsequent packets of the flow are forwarded based on pre-calculated forwarding decisions and do not need any control plane action.

The Figure 1.2 represents the general interaction process between elements of SDN architecture (an OpenFlow-based network).

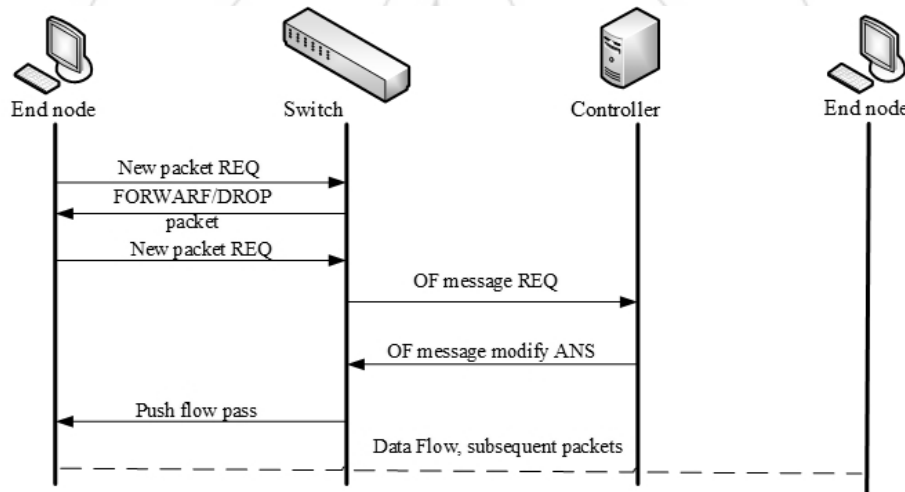


Figure 1.2: General interaction process

OpenFlow protocol has a set of requests that should be take into account during designed new SDN-based network. This set is processing at the SDN control plane: monitoring and evaluation of the network state, recovery after treats, network configuration upgrade [7]. Initialization process or new packet REQ is always the initial phase of interaction when adding new network elements or change the configuration.

More often controller is a bottleneck of the network. However, switches can add delay to the interacting process and reduce performance greatly too. Therefore, it is important to model the controller-switch communication and interaction for the performance analysis of OpenFlow networks.

3. The Model of Openflow-Based Network

Most of the existing researcher works [8-10] are based on simulations or experimentations. Despite this, analytical modeling is a time efficient alternative. Analytical modeling gives opportunity to experiment without taking much time. It can be used for analyzing networks and confidence that could be put in the obtained results.

The paramount importance to have an analytical model which can capture the feedback interaction between the controller and the switch, is able to model any amount of traffic going from switch to controller (or controller to switch), and can be extended to more than one switch [11]. The models suggested in this paper try to obtain all this results.

In the context of random flow, calculation of the degree performance or network load (characteristics quality of service) is based on the probability distribution functions of the system states. Effective mathematical apparatus developed for the analysis of systems that uniform chains and Markov processes.

SDN network like all type of distributed networks can be modeling as systems with losses and systems with queries [12].

For systems with losses main numerical characteristics of QoS is blocking probability ($P(B)$). The blocking probability is determine as the ratio of the amount of blocking or drop packets $S_B(t_1, t_2)$ to the total customer's packets arriving at the same time $S(t_1, t_2)$:

$$P(B) = \frac{S_B(t_1, t_2)}{S(t_1, t_2)}, \quad (1)$$

For the queuing systems main numerical characteristics of QoS is probability of customer's packet arriving and waiting ($P_w, P_w > 0$). The probability customer's packet arriving and waiting is determined as the ratio of the amount of queuing packets $S_Q(t_1, t_2)$ to the total customer's packets arriving at the same time $S(t_1, t_2)$:

$$P_w = \frac{S_Q(t_1, t_2)}{S(t_1, t_2)}. \quad (2)$$

According to suggested, approach OpenFlow-based network control plane layer (switch-controller interaction) can be represented as a feedback-oriented queuing system model, divided into a forward queuing system and a feedback queuing system of the type M/M/1 [11]. The general model of switch-controller interaction depict on Figure 2.1.

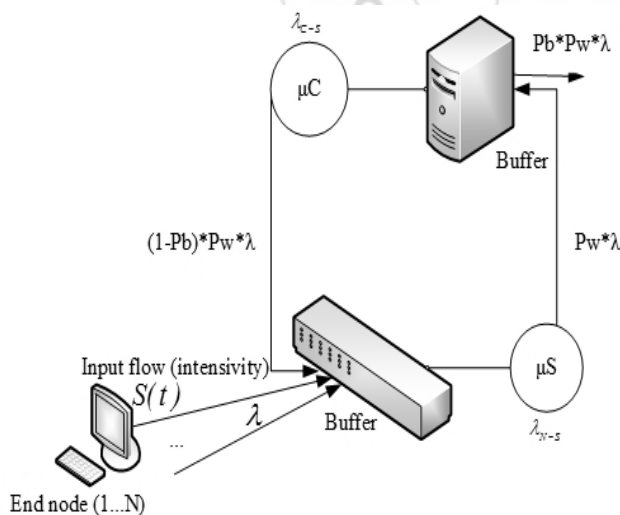


Figure 2.1: The model of OpenFlow-based network (switch-controller interaction)

In the initial state all buffer queues are empty. The probability that controller or switch busy is P_i . According to the requirements of QoS the $\sum_{i=0}^m P_i \leq 1$. P_i Can be written as [10]:

$$P_i = \frac{S(t_i)}{\sum_{i=0}^m S(t_i)}, \quad (3)$$

Where m – the average amount of equipment at the control plan of SDN architecture, $S(t)$ - the intensity of input load (a packet flow).

The arrival process at the switch is a combination of the arrival process of packets received from the line cards with rate λ and of packets being forwarded from the switch buffer after the controller has determined the appropriate action and the corresponding entry in the flow table was created.

Let's assume that the queue length of the controller is limited (Q) in order to model the possibility of dropped packets. The arrival rate in this system is p_w . A packet can be blocked at the controller with probability p_b . The drop packet can be queued in the switch and pass for a second time. Switch query time μS is inversely proportional to the number requests from end node. Controller query time μC is inversely proportional to the number requests from OpenFlow switch.

The feedback controller queue used to model the controller actually comprises out of the lineout card of the switch towards the controller, the buffer, and processing at the controller itself. The transmission time of packets from the switch to the controller is encapsulated in the service time of the controller.

4. Performance and Reliability Evaluation

The proposed model of OpenFlow interaction is using to evaluate performance measures. The main characteristics of model of OpenFlow interaction are: blocking probability, the average packet executed time and summery delay time (switch proceeding delay time and controller proceeding delay time).

- Assumed that the proposed model has some restriction:
- The overall traffic arrival process at the switch and the controller is Poison. Further exponentially distributed service times are used for the switch and the controller. This allows to use results based on M/M/1 queues.
 - A packet has to traverse the switch system at least once.
 - Infinite buffer is assumed at the switch, i.e. switch don't drop packets.
 - Requests are serviced in accordance with the First Come First Served service policy.

The blocking probability can be calculated according to the next formula:

$$P_b = \frac{S(t)}{\sum_{i=0}^{s-1} \frac{S(t)}{i} + \frac{S(t)}{S(t) - S_i(t)}} \quad (4)$$

The average packet executed time ($E[S(t)]$) is defined as the time spent by a packet in the network from the moment it enters the network at its source node, until it leaves through the destination. $E[S(t)]$ for the SDN architecture that depicted on Figure 2.1 can be given as:

$$E[S(t)] = \frac{1}{\lambda} \left(\frac{1}{\mu_S} + \frac{1}{\mu_C} \right). \quad (5)$$

Where $\frac{1}{\mu}$ is average packet transmission time. The load

intensity (load per controller) can be represent as: $\rho = \frac{\lambda}{\mu}$

$$P_{b(C-S)} = \frac{\alpha_1(S(t) - S_i(t))P}{\lambda_{C-S}}; P_{b(N-S)} = \frac{\alpha_2(S(t) - S_i(t))P_{w_i} + \sum_{i=0}^{m-i} \alpha_2(S(t) - S_i(t))P_w}{\lambda_{N-S}} \quad (6)$$

So, probability of customer's packet arriving and waiting can be calculated as:

$$P_w = \frac{1}{\frac{1-\rho}{E[S_n(t)]} + \rho} \quad (7)$$

According to formula (6-7) sojourn time controller-switch can be written as:

$$P_{b(N-S)} = \frac{1}{\mu} \left(\frac{\sum_{i=0}^m \alpha_1(S(t) - S_i(t))P_{w(C-S)}}{\lambda_{C-S}} + \frac{\sum_{i=0}^{m-i} \alpha_2(S(t) - S_i(t))P_{w(N-S)}}{\lambda_{N-S}} \right). \quad (9)$$

As a result proposed a set of different scenarios for proposed OpenFlow models. The characteristics of NOX controller were consider during calculation. The main attention take to the blocking probability as more important characteristic of OpenFlow-based network performance.

The Figure 3.1 depicts the dependence of blocking probability P_b , 100% from controller load ρ , 100%.

In proposed model λ_{C-S} and λ_{N-S} should be analyzed.

λ_{C-S} - intensity inside the switch-controller interaction:

$\lambda_{C-S} = \sum_{i=0}^m \alpha_1(S(t) - S_i(t))P_i$. λ_{N-S} - intensity the switch-

node interaction: $\lambda_{C-S} = \sum_{i=0}^{N+m} \alpha_2(S(t) - S_i(t))P_w$

In this way, the pocking probability can be written as:

$$t_{wi} = \frac{\alpha_1(S(t) - S_i(t))P_{w_i}}{\mu \lambda_{(C-S)w_i}} \quad (8)$$

and sojourn time to packerts for the switch-controller interaction models (Figure 2.1) can be written as:

Proposed model has a maximum blocking probability 0,006 in situation, when controller load is maximum and using 65 End-nodes devices. Values are given for controller loads from 5 to 100 (edge meaning) percent in 10 percent steps. Therefore, controller handles the complete traffic going through the switch.

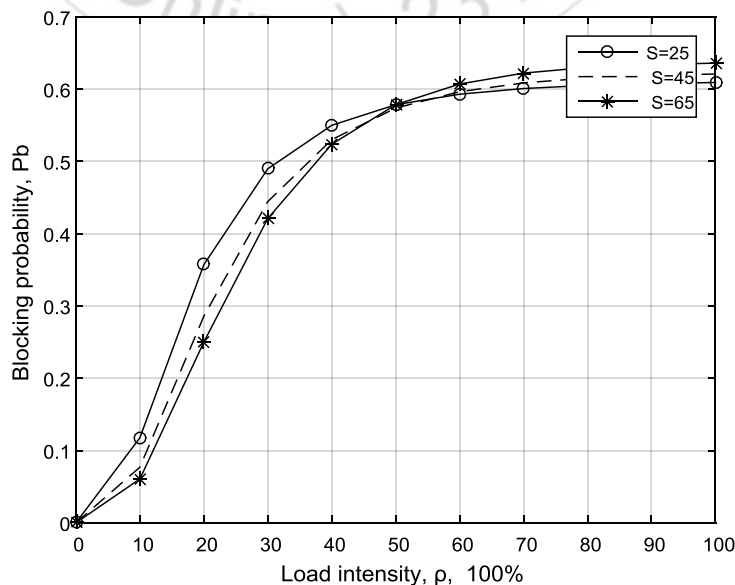


Figure 3.1: Impact load controller intensity to the blocking probability

Figure 3.2 illustrates the modeled the depend of sojourn time of a packet on the controller load for an expected NOX controller service time value of maximum $E[S(t)] = 530msec$.

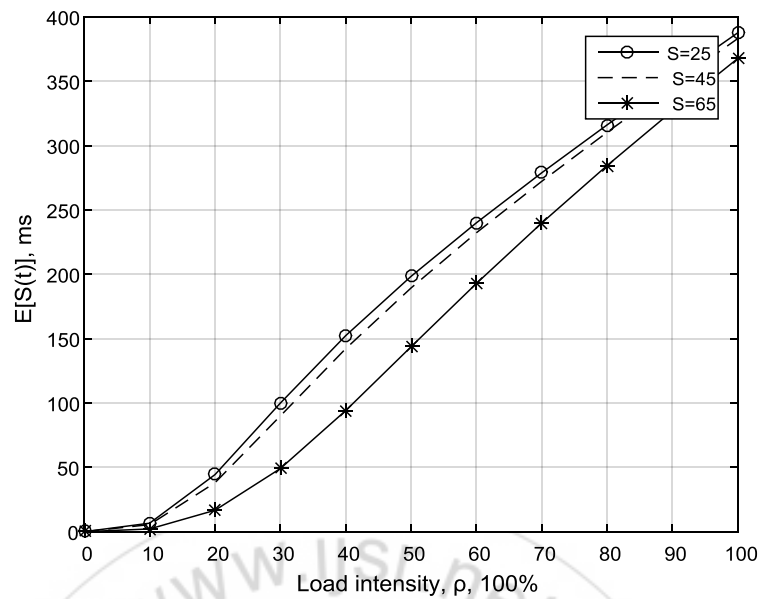


Figure 3.2: Impact the load controller intensity to the average packet executed time

Diagrams on the Figure 3.2 are shown for several intensity distributions for different amount of End-node devise and different controller load. According there diagrams the average packet executed time is almost equal for different intensity and depend only from controller load.

In Figure 3.3 the coefficients of variation for the sojourn times are shown also dependent on the controller load. This is caused by the fact that with a smaller p_w .

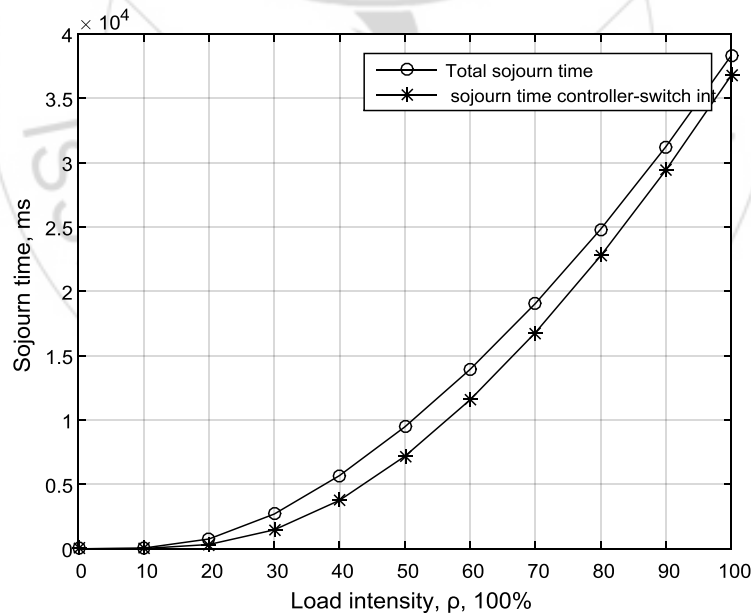


Figure 3.3: Sojourn time distribution

As a result the total sojourn times is depending from controller time delay. The less packets are subject to the delay imposed by the controller and therefore, the deviation from the mean value for these packets is much higher

5. Conclusion

In this work we have proposed an analytical model for an OpenFlow enabled SDN based on M/M/1 network. Proposed in the article OpenFlow-based model of controller-

switch interaction can be used to estimate the packet sojourn time and the probability of lost packets in such a system and can give hints to developers and researchers on questions how an SDN architecture will perform given certain parameters.

As a result, performance of OpenFlow-based model was evaluating. NOX controller characteristics were consider as an input data. The applicability of the model is determined by establishing tree performance probability measures: the blocking probability (assumed the worst variants when

$p_b > 0$), the sojourn time distribution (inside controller and for all fragment), the average packet executed time (inside controller).

The article found these characteristics depending on the number of End node and the intensity of the load that coming to the controller. According to the obtained result the sojourn time not more than 3,7 ms even for edge controller load. Thus, the presented model helps to see the importance of the controller performance for installing new flows.

References

- [1] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: Enabling Innovation in Campus Networks," ACM SIGCOMM Computer Communication Review, vol. 38, no. 2, 2008, - 69 p.
- [2] Architecture SDN [Electronic resource] // Open Networking Foundation. — [2014]. — Mode of access: <https://www.opennetworking.org/>
- [3] Software-Defined Networking: The New Norm for Networks [Electronic resource] // Open Networking Foundation. — [2012]. — Mode of access: <https://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdn-newnorm.pdf>
- [4] OpenFlow Switch Specification (Series) [Electronic resource] // Open Networking Foundation. — [2014]. — Mode of access: <https://www.opennetworking.org/sdn-resources/onf-specifications/openflow>
- [5] Tkachova Olena, Issam Saad, Mohammed Jamal Salim. Mathematical methods for analysis of Software-Defined Networks // International Journal "Information Technologies & Knowledge", Vo9, № 2, - pp. 111 - 123
- [6] ONF, Open Networking Foundation, [Online] Available: <https://www.opennetworking.org>.
- [7] P. Jogalekar, M. Woodside, "Evaluating the scalability of distributed systems," Parallel and Distributed Systems, IEEE Transactions on, vol. 11, no. 6, 2000, - pp. 589–603.
- [8] Azodolmolky, Siamak Nejabati, Reza, Pazouki, Maryam Simeonidou, Dimitra. An Analytical Model for Software Defined Networking: A Network Calculus-based Approach // IEEE Globecom, - 2013
- [9] Naous, Jad & Erickson, David & Covington, G. Adam & Appenzeller, Guido & McKeown, Nick. Implementing an OpenFlow Switch on the NetFPGA Platform // Proceedings of the 4th ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS '08), - 2008, - pp 1-9.
- [10] V. Tanyingyong, M. Hidell, and P. Sjodin. Improving PC-Based OpenFlow Switching Performance // 6th ACM/IEEE Symposium on Architectures for Networking and Communications Systems [Electronic resource], New York, NY, USA, 2010, — Mode of access: <http://doi.acm.org/10.1145/1872007.1872023>
- [11] G. A. Korn, T. M. Korn "Mathematical Handbook for Scientists and Engineers," McGraw-Hill, New York, 1968 -112 p.
- [12] F. Machihara, Transition probabilities of Markovian service system and their application. Rev. Elect. Commun. Labs. Vo.2/29, 1981, - pp. 170-188.