Improvement of Software-Defined Network Performance Using Queueing Theory: A Survey

Ava Tahmasebi[1], Ahmad Salahi[2*], Mohammad Ali Pourmina[3]
1- Faculty of Mechanical, Electrical and Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran.
Email: ava.tahmasebi@srbiau.ac.ir
2- Communications Technology Institute, Iran Telecommunication Research Center, Tehran, Iran.
Email: salahi@itrc.ac.ir (Corresponding author)
3- Faculty of Mechanical, Electrical and Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran.
Email: pourmina@srbiau.ac.ir

Received: November 2020 Revised: January 2021 Accepted: February 2021

ABSTRACT:
Software-Defined Networking (SDN) is a growing network technology that has brought significant benefits to a wide range of disciplines, from science to technology to various fields. This structure can be used in network-based environments, data centers and various research sites. OpenFlow is one of the most widely used protocols for interaction between a controller and a switch in a Software-Defined Networking. Understanding the performance and limitations of the network defined by open source software, including bottlenecks and security vulnerabilities due to the centralized network structure, are important prerequisites for the efficient deployment of these systems. These points of view led researchers to examine various related mathematical models to address these issues. Queueing theory provides the most important and accurate model for evaluating the performance of the SDN networks affected by these restrictions, which has attracted the attention of researchers recently. Regarding to extensive mathematical modeling, this theory has also been used to improve network efficiency. Researchers have used this theory to investigate operational power and improve time consumption and control of data planes due to the nature of classification and storage of packets in SDN buffers. These methods overcome controller performance bottlenecks and increase SDN control capacity, especially for large distributed networks. In this paper, we examine the queueing models for different applications in different layers of SDN, in which researchers use these methods to monitor network loads, evaluate and predict performance changes due to diversity in network traffic. We introduce and review a collection of articles that explore, different applications of Queueing theory in SDN networks. In addition, in order to increase the efficiency of this research, detailed comparisons are performed in terms of structure, mathematical models and the final simulation results.

KEYWORDS: Queueing Theory, OpenFlow, Performance enhancement, Software-Defined Networking.

1. INTRODUCTION

Today, Software-Defined Networking (SDN) is regarded as one of the most important Internet approaches for the future [1-3]. Several factors of software-defined networks make it technologically promising, a more flexible and orderly network with higher initiative power. The network model removes controller from the forwarding layer and enables separated controller entities to modify the rules of transmission in modern switches [4], initiating new service operations and making the SDN respond positively to change network demands and layer topology modifications [5,6].

In SDN architecture, packet forwarding layer are logically manipulated by a centralized controller through an interface. This architecture, of course, enhances the bottlenecks of controller's performance and capacity and makes it easier to manage all switches in large and distributed networks. OpenFlow is considered as the most important SDN protocol which establishes communication between the data layer and the controller layer. More recently, the concept of OpenFlow SDN is finding a way for commercial applications which comes with new challenges [7].

Understanding the performance and limitations of OpenFlow-based SDN is a requirement in operating software applications. Predicting the performance of

Paper type: Research paper
DOI: https://doi.org/10.52547/mjtd.10.1.33
How to cite this paper: A. Tahmasebi, A. Salahi and M. A. Pourmina, **"Improvement of Software-Defined Network Performance Using Queueing Theory: A Survey"**, *Majlesi Journal of Telecommunication Devices*, Vol. 10, No. 1, pp. 33-43, 2021.

OpenFlow networks for network architecture and design is a key issue [8]. Simulation studies are mainly used for performance evaluation, while analytical methods and models have their own advantages. Some researchers used mathematical models to examine the limitations of controller and switch performance. Some other researchers, such as Bing Xiong et al. [9], focused on the average performance of packet forwarding in OpenFlow networks with balanced status. Also, Zuo [10] modeled SDN controller as Mk/M/1 queueing model and with respect to the demands of running the controller as a batch input process. The batch input cannot, of course, recognizes the pattern of the flow requests from multiple switches. While the concept of SDN [11,12] simplifies network operation and reduces the cost of network system and equipment, it also comes with a challenging problem in network operations, such as packet transmission rates and network performance. This issue refers back to controller operation as a remote system to manage and control all transition network systems such as switches. Network control programming and infrastructure capabilities are also deployed in various network services as a factor of defense against DDoS attacks [13]. Each OpenFlow switch has flow tables that accomplish the task of looking up operation as well as sending the packets. As a packet reaches the switch, the packet is compared with the entities of the flow table. If a match is found, the set of instructions containing the flow entity is executed, and if no match is found, the packet is sent to table-miss and subsequently to the controller via the control channel.

In the meantime, the controller will understand how to operate this packet and apply the requirements to all switches with the new packet routing rules. As the number of switches increases, the OpenFlow operates in a complex manner due to the proliferation of the packets and the controller cannot manage the entire network anymore. Therefore, different types of control in SDN topology should be considered [14,15]. The control layer of SDN processes a large number of packets. One of the key performance indicators of this layer, is the average time required to process a packet. This level is performed by simulations which considers an index for single-controller and being an early start to understand the functional complexities of the network. Researchers in [7] considered a parallel method for clustering and parallel processing of position-based queues. Computational modeling of control layers is an interesting approach for evaluating performance improvement. This type of modeling has attracted the attention of Y. Goto and L. Yao [17]. Although research efforts have led to the study of hierarchical structures [18], the computational complexity and network security against attacks have not been solved in such networks.

Queueing theory is a well-supported evaluation tools applied in network based systems. Utilization of such mathematical tools on existing software defined solutions could evaluate the behavior of service systems and relevant key actors. Therefore, network functionality can be analyzed applying the queueing based fundamental principles and analytical framework. Queueing theory considers the workloads of different traffic models in a network and presents a classification of tools that determine the performance of the network.

The remaining part of the paper is organized as follows: in section 2, related works are introduced. Section 3 presents SDN architecture. Queueing theory is widely explained in section 4 and general models which are typically applied to SDN networks are presented specifically with their parameters. In section 5, performance enhancement using this theory is extensively discussed and other applications of queueing theory such as energy consumption, shared buffer model, and security are presented is section 6, 7 and 8 respectively.

2. RELATED WORKS

SDN brings dynamic, adaptive, and programming capabilities of network systems with many other benefits such as centralized control, low complexity, better user experience and a noticeable reduction in network equipment costs [19,20]. The controller in the SDN structure is an operating node that manages the sub-switch devices and provides a regular interface to user-centric network-based applications [21]. Of course, there are multiple controllers in different languages on the market, but these differences make each controller have different performance characteristics and work better for specific situations. Therefore it is important to understand the performance of controllers and their effects while designing Software-Defined Networks.

Zhu et al. [22] proposed criterion and several metrics to analyze the performance characteristics of different SDN controllers. In this paper, comprehensive evaluation of different SDN controllers' performance in different networks is presented and 34 different types are categorized in terms of their properties and capabilities. Then, based on the availability of controller source code or implementation, 9 controllers are selected among them which are: Beacon, NOX, Floodlight, POX, OpenMUL, Maestro, ODL, ONOS, and Ryu. The criterion for selecting these controllers was the implementation and availability of controllers' source code. According to the results obtained from the emulations performed on these controllers, Ryu and OpenMUL controllers have shown less average round trip time which indicates less communication delay. However, these controllers missed the most number of

flows compared to the others. Shalimov et al. [23] presented Hcprobe which is a framework for testing OpenFlow controllers. This framework includes a library that provides tools for working with OpenFlow protocols. It can also provide a set of test situations with evaluation parameters to obtain reliable and measurable parameters as well as security issues. With this framework, they presented a functional analysis of OpenFlow controllers.

One of the approaches used in analytical modeling research on SDN is queueing theory. Using such a mathematical tool can also provide insight into how the system works when traffic rate is changed. Modeling the performance analysis of network components by queueing theory plays an important role in early identification of potential traffic points and bottlenecks as well as accurate evaluation of network configurations that are not yet launched. Bozakov and Rizk used the queueing model to determine the behavior of the interface between the controller and switches the number of messages at different time intervals [8]. They also provided a simple interface for the control framework that enables the network operator to specify time delay ranges for the transition process of control messages. Due to the cumulative input and operating parameters of SDN controllers, the network designer is able to calculate the upper limits of the time delay and buffer requirements of the SDN controllers. Osgouei et al [24] developed an analytical performance model of virtual SDN networks to determine the high latency of the virtual SDN controller and the service process of each virtual network. Goto et al [17] proposed a queueing model for SDN network that incorporates the consideration of input packets into a switch. Miao et al [25] studied a newfound analytical model to evaluate SDN controller's performance. A priority-based queueing system for modeling the data layer of this network is intended to further address the multipurpose nature of packet forwarding. In addition, they examined the performance of the controller and average service time using queueing theory. The performance of control plane with several compatible controllers have been studied [18] [26]. Although in these models, attempts have been made to maximize the network performance using queueing theory and classification algorithms, but these achievements are possible only by considering the limited rate of input packets and buffer length analysis has not been considered for investigating higher rate of incoming traffic flows.

Controllers basically monitor the network topology with the help of request management. Estimating the processing capacity for these requests is one of the key elements in evaluating SDN performance. Therefore, by simulating the operational delay and system flow management, performance of the control plane can be examined. The flexibility of SDN networks allows network researchers to quickly provide prototypes and test equipment for performance evaluation. In [27] the performance of OpenFlow-based networks is studied using a feedback-based queueing model. In this case, the switch is modeled as a buffer and the controller acts as a feedback queue system. In another model expansion [25], the Jackson network was exploited for modeling of the data layer since the controller was modeled separately as M/M/1 queue that considered limited and unlimited buffer states. Switched-in traffic in the correct and standard OpenFlow structure coming from the controller should not be returned again. However, in the two situations (limited and unlimited buffer states) traffic from the controller is not specified and just a rough approximation of the connection between the controller and the switch is provided. At the same time, by stabilizing and strengthening the level of control, the performance and scalability of the controller architectures become a major issue [27] [28]. In the analytical model proposed by [29], the switch buffer is modeled as a M/M/1 queue. In this model, the controller and switch queues operate independently and part of the incoming traffic to the data plane is forwarded to the control plane.

Since increasing network efficiency is related to optimizing the performance of each network node's operation when dealing with different signaling flows, the general strategy for this purpose includes better identification of flows based on its statistical parameters, network node performance based on scheduling time, capacity and rate of their input/output signaling and the controller management process after monitoring all the information. The remaining important issues to be addressed are the precise identification of the SDN structure as well as the theoretical mathematics for accurately modeling these goals.

3. SDN ARCHITECTURE

SDN broadly consists of three layers which are application layer, control layer and infrastructure layer (Figure 1). In the top layer which is the application layer, the programs reside that communicate behaviors and needed resources with the SDN controller via APIs. This layer is an open area to provide services for end users. In SDN, a complete separation of control layer from the data layer is involved as well as implement management and orchestration by a concentrate control layer which brings adaptability and dynamics to systemic network traffic. Due to the growth of the network as well as its traffic, the controller performance is subject to bottlenecks that can lead to a loss of network capability and security [29]. The controller usually manages OpenFlow switches through a group

of hosts. As a packet reaches the switch, the packet is compared with the entities of the flow table. If a match is found, the set of instructions containing the flow entity is executed, and if no match is found, the packet is sent to table-miss and subsequently to the controller via the control channel. In a study by Tootoonchian et al. [16], NOX controller was upgraded and compared with two other types of controllers named Beacon, and Maestro in terms of performance enhancement scheme.

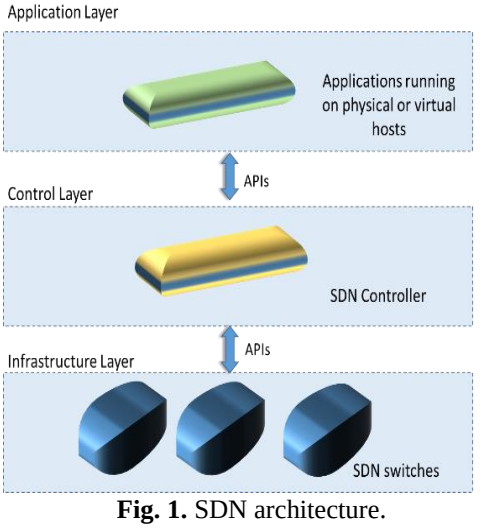

Fig. 1. SDN architecture.

The researchers concluded that the improved controller scheme using the buffered structure-packet forwarding algorithm performed better than the others. Similar controller performance improving method was also examined by Jiang et al. [30]. In such articles, an attempt has been made to review the various conventional methods that have been proposed to increase network performance, especially at the control layer. In all these methods, they have tried to evaluate and eliminate the weaknesses of SDN networks, which include security problems, lack of monitoring of incoming traffic in the face of high system load and system performance in buffering the packets, with optimal conceptual and statistical modeling.

4. QUEUEING MODELS

In queueing theory each queue can be represented by a 5-tuple A/B/C/D/E. If D and E are not mentioned, their values are infinite. 'A' represents distribution of inter arrival time if A=M then random variable inter arrival time has negative exponential density function like:

$$f(\tau) = \lambda e^{-\lambda\tau} \quad (1)$$

Where λ is the average arrival rate. For this arrival density function, it is easy to show that the random variable which represents the number of arrivals in equal periods T follows Poisson distribution. If A=G then it has a general density function. B represents distribution of service time. If B=M then, the random variable service time has negative exponential density function like:

$$f(t) = \mu e^{-\mu t} \quad (2)$$

Where μ is the average service rate. If B=G then service time has a general distribution.

Besides, C represents the number of servers, D represents the buffer size and E represents the number of population which generate the input traffic.

A priority queue is a type of queue in which each input additionally has a priority associated with it. In priority queue input with highest priority served first.

The mathematical queueing models are used in network modeling to evaluate the performance of network functions. The most commonly used models among the general models of queueing theory are: M/M/1/L, M/M/c/L (Figure 2), M/D/1 and M/G/1.

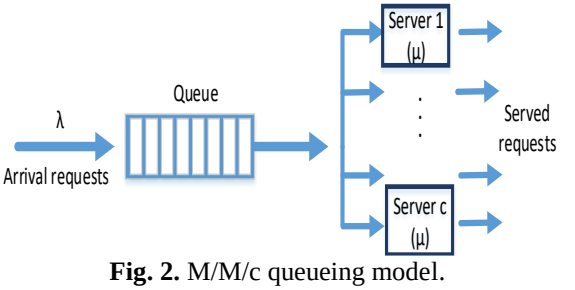

Fig. 2. M/M/c queueing model.

In M/M/c queue, by defining ρ=λ/cμ as the system stability index, which introduces the ratio of packet entry rate to system service rate and accepting the equilibrium condition by having cμ≥λ ,by using Little law, the average of packets in the queue can be obtained. In fact , if W_q is considered as the average waiting time in the queue and λ is the entry rate, according to Little's law the average queue length L_q is calculated as follows:

$$L_q=\lambda W_q \quad (3)$$

Switching in Software-Defined Networks utilizes queueing theory in two ways. Single shared buffer for control and data layer traffic as well as two priority buffers for separating control and data layer traffic. Deepak Singh [31] found that shared buffer requires 85% higher delay time to launch network flow table entities in switches and requires more than 82% more buffer capacity than priority queue buffer.

Poisson distribution is the prevalent model deployed for evaluating traffic in most analytical models of OpenFlow networks [32]. In these models [33], arrival rates at both the data layer and control layer

are assumed to follow the Poisson distribution. One controller manages multiple switches in the SDN [34]. However, different hosts cause frequent data flow requests. When requests for flow are made, they will have access to the controller at once completely and randomly. It is in this situation that the controller can manage only one flow at a time and the queue system is formed.

The flow tables of each OpenFlow switch being used in two important tasks to accomplish which are looking up operation and sending the packets [35]. As a packet reaches the switch, the packet is evaluated with the entities of the flow table. Whenever a match is found, the set of instructions containing the flow entity is executed. Else, the packet is sent to table-miss and subsequently to the controller via the control channel. The controller updates the switches via new rules. These rules contains list of actions which should be performed in the order that they are defined. If no action is described, it indicates that the packet should be dropped. Controllers basically monitor the network topology with the help of request packets received in control plane. The processing capacity of the controller is one of the key elements in evaluating SDN performance. . Aside from the number of switches, the effect of multiple switches on performance as well as the accuracy of the model from the processing power and latency dimensions should be examined. Requests are always high and operating capacity must increase as the number of switches increases. There must be regular information about each switch to make this cycle works.

5. PERFORMANCE IMPROVEMENT

Queueing theory has a successful record of modeling computer networks and systems. In the case of computer-based networks, and despite widespread worldwide efforts, there is still little research to evaluate and improve the performance of software-defined or OpenFlow networks. There is considerable research on queueing theory that has examined OpenFlow or OpenFlow networks. However, there has been no documented source and research to assist in-depth networking professionals, and much of the research remains theoretical and substantive.

There are, of course, several tools for network administrators to help them monitor the performance of a network. These tools show the network topology to the network administrator with performance and device information. Factors are embedded within the network structure to monitor. The agents are the software that provides the device information and then communicates with the central server, which aggregates and displays all the data. Basically, communication between agents and central servers is done by simple network management protocols. Modeling techniques in the form of queueing theory can help to better evaluate the performance of SDN. Performance evaluation typically enables early detection of network hotspots as well as bottlenecks in which network providers can resolve them quickly and before they encounter problems.

In the network structure, constraints related to bottlenecks can be met by analysis models without any simulation or physical configuration and deployment operations. But there is a sharp learning approach that limits the use of analytics and queueing experiences to practitioners in a network. So far, various structures have been presented that bring queueing theory closer to what is in the minds of SDN researchers [36]. SDN controller decides how packets are sent from one switch input to the output, while the data layer performs those decisions. The separation of control and data layers is a particular advantage in investigating buffer prerequisites over different time intervals in packet processing and traffic rate at control and data layers.
In shared-buffer switches, input ports and output buffers are shared in order to reduce computational operations. Therefore, the factor of packet time delays in the switch is obtained by using the service rate and the capacity of the output buffers.

In order to evaluate the impact of the number of switches on the efficiency and accuracy of the method, many simulations are performed and the results are evaluated in terms of processing power and delay. Since there are many features that have impacts on system efficiency such as topology updating capabilities, distribution speed and flow table, etc., the efficiency of controller requires careful modeling and analysis of results. The single-layer controller cannot handle large network requests. Therefore, it is necessary to use several controllers in order to have a scalable architecture for the controller. In [37], an attempt has been made to use a multi-control structure to optimize scalability and network expansion. The dormant structure is also used to reduce cost and increase system efficiency. In this idea, part of controllers is allowed to get into the dormant state under light traffic condition to evaluate the efficiency of this system. The M/M/c queue and genetic algorithm have been used to achieve the best values of designing variables for deployment decision making and operating cost reduction. In [38] to increase the efficiency of resource-limited mobiles, the SDN cloud network has been used to offload their computationally intensive tasks to cloud servers using the resource management feature. In this case because service time density function of offloaded tasks are not known, M/G/1 queue has been used.

In [39], queueing theory was used to better manage the proposed multi-controller and load balancing systems. Traffic propagation delay and controller

capacity were the two main keys for evaluating and achieving better system performance. By using queueing theory, optimal parameter values were extracted. There are similar systems in this field that used queueing theory in different parts of the SDN network and modeling different network functions to obtain optimal values and achieving better system performance. Table (1) shows queueing model based techniques to enhance performance.

Table 1. Queueing model-based techniques to enhance performance.

Reference	Key Performance	Queueing Model	Network structure
Yonghong et al [37]	Cost Managing	M/M/c	Multi-controller based SDN network
Chamola et al [38]	Latency aware task assignment	M/G/1	Cloudlet network
Li et al [39]	Traffic propagation delay and controller capacity	M/M/1	Multi-controller network with 34 nodes and 42 links
Hu et al [40]	Routing optimization	M/M/1	Hierarchical multi-controller architecture
Ansell et al [41]	Performance evaluation and traffic visualization	M/M/1	SDN network using Ryu controller
Sood et al [42]	Flow-table size and packet arrival rate and number of rules	M/G/1	SDN network with 1000 nodes

6. ENERGY CONSUMPTION

Energy consumption is an important factor in green software-based networks. The layers of control [43] and data [44,45] consume a large amount of energy. The control layer acts as a software-based network operating system. Since control layer consumes a large part of energy, assessing the energy consumption of the control layer is a common topic in the SDN research [46,47].

Few researches have focused on modeling the energy consumption of network control layer, except STSC [48]. It is difficult to provide a special control layer for a given level of energy consumption. The queueing theory framework enables the modeling of performance and energy consumption in software-based networks. Huang et al [49] proposed a method that provided energy modeling and performance of network control layer in terms of integrated evaluation and comparative framework. They developed a computational model for several threading and group of controllers to enhance available frameworks for hierarchical patterns. They also proposed a framework for software-based network control layer based on analytical models that can analyze and evaluate the performance and sustainable energy of network control layers. In this method, six levels of control in single-threaded controllers, cluster controllers, local strategy of flat structures, global strategy of flat structures and hierarchical structures are simulated. Besides, the processing time and energy consumption of these levels at a same data layer and similar capacity and service rates are examined. The overall saturation is equal to the total number of packets that the control layer can match and the service rate is equal to the average packets that can be processed by the controller at any time. The simulation results can accurately show the performance and power levels of software-centric network control and the proposed framework can test various software-centric network control levels as well. This method is an effective way for control levels identification and appropriate for software-based sustainable network applications.

The subject of modeling the energy consumption of different layers of SDN, has been considered in various dimensions in datacenters. Due to the lack of intelligent infrastructure for selecting and deriving switches with the desired flow-table capacity, achieving lower costs and saving more energy in the network, faced with obstacles. In a study by [46], three life process-split of flow-table entry are defined as packet-in process (modeled as M/M/1), handling process (modeled as M/M/1) and serving process (modeled as M/G/c/L). The packet-in process indicates the procedure of sending the request message to the controller in regard of non-compliance with the routing table entries. While, the handling process is related to the controller operation in dealing with such new packets and serving process is the step of updating the flow-table with the rule assigned by the controller. As it supposed that the input packets have Poisson distribution, the arrival rate of new flows from all connected hosts to a switch is concluded to be:

$$\lambda_{in} = \sum_{i=1}^{n} \lambda_i \quad (4)$$

λ_i indicates the arrival rate of new flows from different *n* ports (i=1,…,n) and λ_{in} indicates the arrival rate of these flows from the ports to the switch. Since the input packets are modeled as M/M/1, the arrival rate and the departure rate of the packets from switch to the controller are equal. This logic is repeated for the controller process because the queue model selected for

this process is the same as the previous step. Thus the arrival rate for each processing of the control plane is:

$$\lambda_{packet-in} = \sum_{i=1}^{m} \lambda_i(S_i) = \sum_{i=1}^{m} \lambda_{in}(S_i) \quad (5)$$

Which S_i indicates the i^{th} switch and m indicates the number of total switches and $\lambda_{packet-in}$ is the arrival rate of the packet-in messages. While the likelihood of flow-mod message (which is sent from controller to other switches, after generating the final rule for the switch's packet-in request) is defined as $\pi(s_i)$, the arrival rate of flow-mod is obtained as follows:

$$\lambda_{flow-mod} = \sum_{i=1}^{m} \lambda_{packet-in}(S_i).\pi(S_i) \quad (6)$$

In which $\lambda_{flow-mod}$ is the arrival rate of the flow-mod messages .For the serving process, M/G/c/c model is considered and supposed that each switch's capacity of flow entry equals *c*. thus, the steady state probability of the *j* flow tables in each switch can be calculated as:

$$p_j = \frac{(\lambda_{flow-mod}/\mu_f)^j/j!}{\sum_{i=0}^{c}(\lambda_{flow-mod}/\mu_f)^i/i!} \quad (7)$$

Which f indicates a flow table entry. The probability of packet loss can be obtained due to non-response of the flow table as below:

$$p_{packet-loss} = \frac{(\lambda_{flow-mod}/\mu_f)^c/c!}{\sum_{i=0}^{c}(\lambda_{flow-mod}/\mu_f)^i/i!} \quad (8)$$

As can be seen from the above statement, the rate of capacity is inversely related to the probability of packet loss. Therefore, by determining the maximum acceptable probability, the flow table capacity in SDN switches can be optimally designed.

Kuap et al. [50] proposed an energy assessment framework, provided a structure for the controller, and finally extracted the energy model that enables the assessment of energy consumption. This model captures network energy consumption and traffic. Faraci et al [51] developed a logical framework to appraise the coordination server's operation of SDN/NFV(Network Function Virtualization) network when VNFs(Virtual Network Function) are located at the edge of the network by the Telco operators. In this method, the flows, assuming to be the same in terms of preference and service quality, enter the set of queues of the system. These flows are fragmented in blocks based on the functions they are supposed to perform on different parts of the network. Each block represents an operation which is provided by a VM(Virtual Machine) running on the enhanced Customer Premises Equipment (eCPE) node. If no VM is available, the blocks are lined up and serviced by eCPE node according to the FIFO service policy.

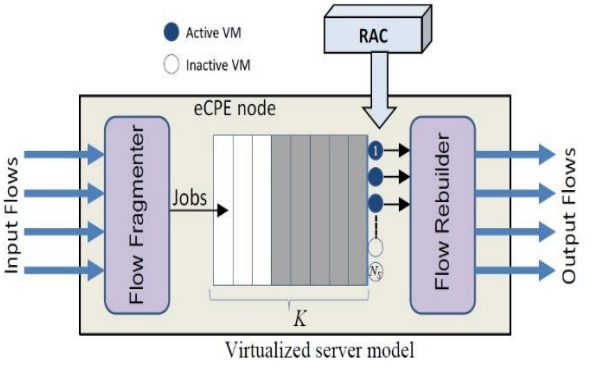

Fig. 3. Proposed virtual system model in [51].

It should be noted that these methods have been mathematically analyzed and simulated, and the practical results in terms of real queueing traffic in SDN systems have not been studied yet. Also, in queueing modeling, no priority order is considered in the incoming traffic in order to observe its effect on the queue parameters and consequently the energy consumption.

7. SHARED BUFFER MODELS

Buffering in switch is concerned with varied and temporary traffic absorption. Output buffers also exist in the form of a single queue or with dual precedencies [52]. In the shared queueing model, packets proceed the controller and switches and the queue is distributed according to the FIFO service method, but with two precedencies, packets are individually queued, in different FIFOs of service providers. The system uses priority and limited capacity queues to represent the switch. The lower precedence queue is served when there is no packet in the higher precedence queue. This structure shows the OpenFlow switch. However, by utilizing the limited capacity queueing model for switches, the queueing model does not provide any input from queueing solutions which is much more difficult to analyze. The key point about software-based network switches is the size of the output buffer. To solve this challenge, buffers are usually considered as an infinite queue. Therefore, two queueing models are defined to investigate the effect of buffer sharing. These models are called SE and SPE, which are relevant to the divergent switch queue forms [31]. "S" in these models is referred to the switches in the software data layer and "E" is referred to the part where the packets are encapsulated and sent to the control plane and "P" is referred to the data queue precedence. The SE shared buffer model and SPE precedence queue buffer model (depicted in Figure 4-(a) and 4-(b) respectively) are compared and the three concepts of buffer size, choice of queue with or without precedency, and controller server capacity are taken into account for software-

based performance as well as software network switching performance.

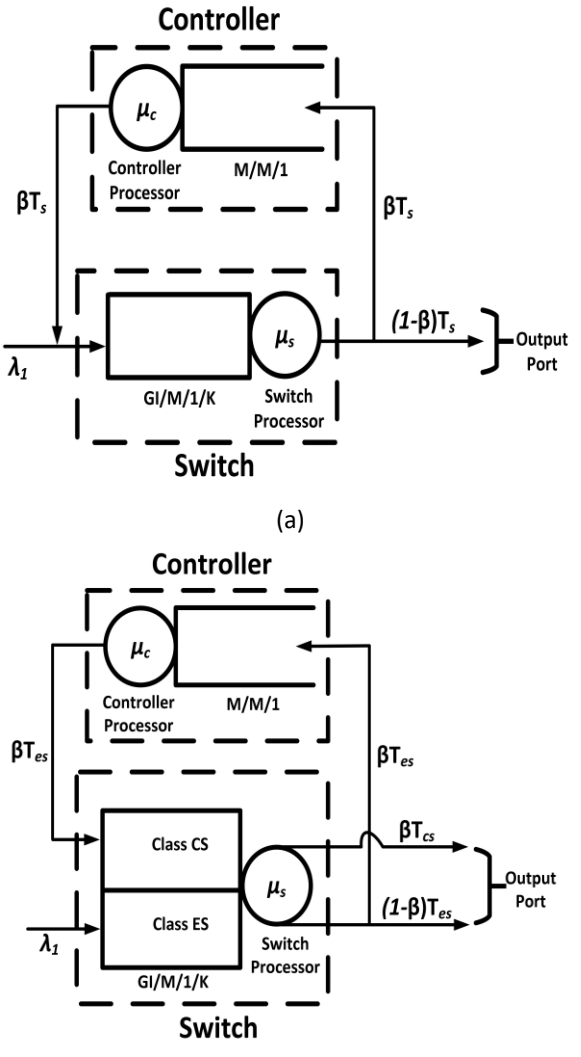

Fig. 4. (a): shared buffer (modeled as PE) and (b): precedence based queuing buffer (modeled as SPE) [31].

As it is shown in Figure 4, In the SE model, a single queue is used for the switch represented as GI/M/1/K to indicate independent arrival flows with general distribution. While in the SPE model, precedence based queue is presented in which two class of packets defined as CS (controller to switch) and ES (external to switch) in order to specify different packet processing paths. Due to the perfect isolating scheme for control packets and traffic packets, the waiting time and loss probability have been reduced in the SPE model compared to the SE model.

8. SECURITY AND PROTECTION AGAINST ATTACKS

The centralized structure of SDN networks allows botnets to disable network components by generating massive traffic. Although prevention of traffic from entering into switches during the attack completely neutralizes the attack, it also prevents legitimate traffic from entering the switch. In the technique proposed in [53] queueing theory is used to prevent flooding attacks. In this method, buffers with different priorities are used and input packets are prioritized and placed in different queues according to some criteria which can be gained from their headers. Packets that are suspected of being malicious packets are placed in lower-priority buffers, resulting in later service.

Another method proposed by zhang et al. [54] uses a multi-layer for queueing to confront the DDoS attack on the SDN controller. In this method, the queues have the ability to be integrated when normal traffic enters the network and reduce the load of the controller, and instead, if the attack is detected, it can be distributed dynamically and separate the attack traffic from normal.

Other methods also have been suggested by researchers for using queueing theory to achieve higher security in SDN, which are summarized in Table 2.

In all the mentioned methods, conventional databases and real IPs have been used. While these systems can be threatened with IP spoofing attacks and low-rate attacks. In these methods, it is also assumed that the queues that are deployed in these systems in order to efficiently manage the incoming traffic and service all the legitimate packets using the flow tables in SDN switches. But it must be considered that if the flow table also overflows the system performance will be degraded. Therefore the algorithms need to be expanded and upgraded to enhance the efficiency of the defense mechanisms against unpredictable attacks.

9. CONCLUSION

SDN has been popular for more than a decade among network providers and researchers. The network deals with service providers and carriers, and organizations are able to monitor the movement of their information packets over the network. Network providers can make optimal decisions about the flow of network traffic from one location to another and will be able to publish these decisions to the routers and switches of the network, which ultimately creates the structure of a network.

Table 2. Queueing model-based techniques used in SDN security systems.

Reference	Method	Attack	Network Topology	Achievements
Wei et al [53]	Request priority in flow ranger using queueing theory	DoS flooding attack	One controller and three switches with total 100 users	43% more in performance compared to FCFS
Zhang et al [54]	Multiple layers of dynamic queues	DoS attack	One Controller, two switches and 6 hosts	Lowers the RTT
Yan et al [55]	Time slice allocation-based multi-queueing algorithm in SDN switches	DDoS attack	One Controller, Three switches and four client groups	Decreasing delay to 1.7s and failure ratio to 0.2%
Eom et al [56]	Graphical security metrics using queueing model and genetic algorithm	Cyber attack	One controller and three switches connected to the security module	Exceeds 0.9 in probability of attack success
Wang et al [57]	Time slice allocation and controller scheduling method using queueing algorithm	DoS attack	Floodlight controller and two switches connected to 6 hosts	Achieving 5% failure ratio at attack rate of 3000 $\text{packets}/_{s}$

The key mathematical tools that are brought using various models of queueing theory, not only result in performance improvement, but also helps researchers understand the behavior of network elements when facing different types of network traffics. Getting useful results from queueing models can help gather network status data across all segments of the network as well as with pre-processing. Thus, to meet the critical performance requirements of SDN networks, queue based solution has been proposed by several researchers in the literature. However, there was a lack of a detailed and extensive review classification for the applied queue management models. In this paper we tried to fill this gap by presenting a wide range of comprehensive study of theoretical approaches of queueing models in software defined networks.

We provide a comprehensive and novel classification of queueing theory applications in SDN networks. In each part, we indicate the key factors of queueing models and highlight the solutions and network structures. The state-of-art researches' results, which are addressed in this paper, show that the throughput and time delay in the controller platform increases as the switches are expanded so as not to exceed the controller processing capacity. In this regard, factors that affect SDN performance improvement are the capacity for updating topology and time delay which requires a precise queueing model. We also discussed some significant issues in buffer shared problems and defense mechanisms in SDN that need traffic management and monitoring modeled by queueing models. For the sake of carrying out evaluation of their results, detailed comparison of different deployed queue models is made as well. Besides, as queueing theory has been widely used for balancing congestion and attack detection in SDN networks, the last part of this paper is dedicated to present and compare the relevant methods developed using queueing models for SDN security and protection against attacks. An important issue in queueing theory applications that needs more attention in SDN research is the unconventional behavior of packets that reach SDN nodes and buffers.

REFERENCES

[1] A. Hakiri, G. Aniruddha, B. Pascal, C. Douglas, T. Gayraud, **"Software-Defined Networking: Challenges and research opportunities for future internet"**. Computer Networks, 2014. 75: pp. 453-471.

[2] J.Pan, S. Paul, and R. Jain, **"A survey of the research on future internet architectures"**. IEEE Communications Magazine, 2011. 49(7): pp. 26-36.

[3] S. Rowshanrad, W. Yonggang, H. Chuan, **"A survey on SDN, the future of networking"**. Journal of Advanced Computer Science & Technology, 2014. 3(2): pp. 232-248.

[4] A. Lara, A. Kolasani, B. Ramamurthy, Network innovation using OpenFlow: A survey. IEEE communications surveys & tutorials, 2013. 16(1): pp. 493-512.

[5] N. McKeown, T. Anderson, H. Balakrishnan, **"OpenFlow: enabling innovation in campus networks".** ACM SIGCOMM Computer Communication Review, 2008. 38(2): pp. 69-74.

[6] N. Feamster, J. Rexford, and E. Zegura, **"The road to SDN: an intellectual history of programmable networks".** ACM SIGCOMM Computer Communication Review, 2014. 44(2): pp. 87-98.

[7] B. Nunes, M. Mendonca, X. Nguyen, K. Obraczka, T. Turlett, **"A survey of Software-Defined Networking: Past, present, and future of programmable networks".** IEEE Communications Surveys & Tutorials, 2014. 16(3): pp. 1617-1634.

[8] Z. Bozakov, and A. Rizk. **"Taming SDN controllers in heterogeneous hardware environments".** 2013 Second European Workshop on Software-Defined Networks. 2013.

[9] B. Xiong, K. Yang, J. Zhaoc, L. Keqin, **"Performance evaluation of OpenFlow-based Software-Defined Networks based on queueing model".** Computer Networks, 2016. 102: pp. 172-185.

[10] Q. Zuo, M. Chen, and pp. Jiang, **"Delay evaluation of OpenFlow control plane by queueing model"**. Huazhong Univ. Sci. Technol. (Nat. Sci. Ed.), 2013. 8(1): pp. 44-49.

[11] I. Alsmadi, I. Alzam, M. Akour, **"A systematic literature review on Software-Defined Networking"**, in Information Fusion for Cyber-Security Analytics. 2017, Springer. pp. 333-369.

[12] W. Xia, H. Wen, C. Heng, **"A survey on Software-Defined Networking"**. IEEE Communications Surveys & Tutorials, 2014. 17(1): pp. 27-51.

[13] Q. Yan, F. Yu, Q. Gong, J. Li, **"Software-Defined Networking (SDN) and distributed denial of service (DDoS) attacks in cloud computing environments: A survey, some research issues, and challenges".** IEEE Communications Surveys & Tutorials, 2015. 18(1): pp. 602-622.

[14] N. Gude, **"NOX: towards an operating system for networks"**. ACM SIGCOMM Computer Communication Review, 2008. 38(3): pp. 105-110.

[15] H. Yeganeh, and Y. Ganjali, **"A framework for efficient and scalable offloading of control applications".** Proceedings of the first workshop on hot topics in Software-Defined Networks. 2012. ACM.

[16] A. Tootoonchian, S. Gorbunov, Y. Ganjali, **"On Controller Performance in Software-Defined Networks".** The 2nd {USENIX} Workshop on Hot Topics in Management of Internet, Cloud, and Enterprise Networks and Services. 2012. pp. 10-16.

[17] Y. Goto, H. Masuyama, N. Bryan, K. Winston, G Seah, Y. Takahashi. **"Queueing analysis of Software-Defined Networking with realistic OpenFlow–based switch model"**. IEEE 24th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS). 2016.

[18] J. Hu, C. Lin, X. Li, J. Huang, **"Scalability of control planes for Software-Defined Networks: Modeling and evaluation".** In 2014 IEEE 22nd International Symposium of Quality of Service (IWQoS), pp. 147-152. 2014.

[19] H. Farhady, H. Lee, and A. Nakao, **"Software-Defined Networking: A survey"**. Computer Networks, 2015. 81: pp. 79-95.

[20] F. Hu, Q. Hao, and K. Bao, **"A survey on Software-Defined Networking and OpenFlow: From concept to implementation"**. IEEE Communications Surveys & Tutorials, 2014. 16(4): pp. 2181-2206.

[21] F. Alencar, M. Santos, M. Santana, S. Fernandes **"How Software Aging affects SDN: A view on the controllers".** 2014 Global Information Infrastructure and Networking Symposium (GIIS). pp. 1-6, 2014.

[22] L. Zhu, M. Karim, F. Li, X. Du, M. Guizani **"SDN Controllers: Benchmarking & Performance Evaluation"**. arXiv preprint arXiv:1902.04491. 2019 Feb 12.

[23] A. Shalimov, D. Zuikov, D. Zimarina, V. Pashkov, R. Smeliansky **"Advanced study of SDN/OpenFlow controllers"**. In Proceedings of the 9th central & eastern european software engineering conference in russia. pp. 1-6, 2013.

[24] A.G. Osgouei, A.K. Koohanestani, H. Saidi, A. Fanian, **"Analytical performance model of virtualized SDNs using network calculus"**. In 2015 23rd Iranian Conference on Electrical Engineering. pp. 770-774, 2015.

[25] K. Mahmood, A .Chilwan, O. Østerbø, M. Jarschel. **"Modelling of OpenFlow-based Software-Defined Networks: the multiple node case"**. IET Networks, 4(5): pp. 278-284. 2015.

[26] Y. Fu, J.B., J. Wu, Z. Chen, K. Wang, and M. Luo, **"A dormant multicontroller model for Software-Defined Networking,"** China Communications, 2014. 11(no. 3,): pp. 44-45.

[27] M. Kobayashi, S. Seetharaman, G. Parulkar, G. Appenzeller, **"Maturing of OpenFlow and Software-Defined Networking through deployments"**. Computer Networks, 2014. 61: pp. 151-175.

[28] S. H. Yeganeh, and Y. Ganjali, **"On Scalability of Software-Defined Networking"**. IEEE Communications Magazine, 2013. 51(no. 2,): pp. 136-141.

[29] S. Sezer, S. Scott-Hayward, pp. K. Chouhan, Fraser B, **"Are we ready for SDN? Implementation challenges for Software-Defined Networks"**. IEEE Communications Magazine, 2013. 51(7): pp. 36-43.

[30] G. Jiang, B. Fu, M. Chen, L. X. Zhang., **"The survey and quantitative analysis of SDN controller"**. Frontiers of Computer Science and Technology, 2014. 8(6): pp. 653-664.

[31] D. Singh, B. Ng, Y.C. Lai, Y.D. Lin, W.K. Seah. **"Modelling Software-Defined Networking: switch design with finite buffer and priority queueing".** 2017 IEEE 42nd Conference on Local Computer Networks (LCN). 2017.

[32] B.D. Choi, D.I. Choi, Y. Lee, D.K. Sung, **"Priority queueing system with fixed-length packet-train arrivals"**. IEE Proceedings-Communications, 1998. 145(5): pp. 331-336.

[33] R. Jain and S. Routhier, **"Packet trains--measurements and a new model for computer network traffic"**. IEEE journal on selected areas in Communications, 1986. 4(6): pp. 986-995.

[34] H. Okamura, T. Dohi, and K.S. Trivedi, **"Markovian arrival process parameter estimation with group data".** IEEE/ACM Transactions on Networking (TON), 2009. 17(4): pp. 1326-1339.

[35] Z. Wang, S. Zhao, Z. Fan, X. Wan **"Performance Modeling and Analysis of Control Plane for SDN Based on Queueing Theory".** Wireless Personal Communications, 2017. 97(1): pp. 591-601.

[36] M. Jarschel, T. Zinner, T. Hoßfeld, pp. Tran-Gia, W. Kellerer, **"Interfaces, attributes, and use cases: A compass for SDN".** IEEE Communications Magazine, 2014. 52(6): pp. 210-217.

[37] F. Yonghong, B. Jun, W. Jianping, C. Ze, W. Ke, Min L. **"A dormant multi-controller model for Software-Defined Networking"**. China Communications. 2014, 11(3): pp. 45-55.

[38] V. Chamola, C.K. Tham, S. Gurunarayanan, N. Ansari **"An optimal delay aware task assignment scheme for**

wireless SDN networked edge cloudlets". Future Generation Computer Systems. 2020. 102, pp. 862-875.

[39] G. Li, X. Wang, Z. Zhang **"SDN-based load balancing scheme for multi-controller deployment"**. IEEE Access. 2019, 7, pp. 312-322.

[40] J. Hu, C. Lin, pp. Zhang. **"Performance evaluation and optimization of hierarchical routing in SDN control plane"**. Chinese Journal of Electronics. 2018, 27(2), pp. 342-350.

[41] J. Ansell, W.K. Seah, B. Ng, S. Marshall **"Making Queueing theory more palatable to SDN/OpenFlow-based network practitioners".** InNOMS 2016-2016 IEEE/IFIP Network Operations and Management Symposium. 2016, pp. 1119-1124.

[42] K. Sood, S. Yu, Y. Xiang. **"Performance analysis of Software-Defined Networking switch using M/Geo/1 model".** IEEE Communications Letters. 2016, 20(12). pp. 22-25.

[43] M.F. Tuysuz, Z.K. Ankarali, and D. Gözüpek, **"A survey on energy efficiency in Software-Defined Networks".** Computer Networks, 2017. 113: pp. 188-204.

[44] C. Newport, and W. Zhou. **"The (surprising) computational power of the SDN data plane"**. 2015 IEEE Conference on Computer Communications (INFOCOM). 2015.

[45] K. Xie, X. Huang, S. Hao, M. Ma, **"Distributed power saving for large-scale software-defined data center networks"**. IEEE Access, 2018. 6: pp. 5897-5909.

[46] G. Shen, Q. Li, S. Ai, Y. Jiang, M. Xu, X. Jia. **"How powerful switches should be deployed: A precise estimation based on queueing theory"**. IEEE Conference on Computer Communications, 2019, pp. 811-819.

[47] A. Hammadi, and L. Mhamdi, **"A survey on architectures and energy efficiency in data center networks"**. Computer Communications, 2014. 40, pp. 1-21.

[48] T. Qiu, Y. Zhang, D. Qiao, X. Zhang, M.L. Wymore, A.K. Sangaiah. **"A robust time synchronization scheme for industrial internet of things"**. IEEE Transactions on Industrial Informatics, 2017. 14(8): pp. 3570-3580.

[49] X. Huang, F. Li, K. Cao, pp. Cong, T. Wei, S. Hu, **"Queueing Theoretic Approach for Performance-Aware Modeling of Sustainable SDN Control Planes"**. IEEE Transactions on Sustainable Computing 5, no. 1 pp.121-133. 2018.

[50] F. Kaup, S. Melnikowitsch, and D. Hausheer. **"Measuring and modeling the power consumption of OpenFlow switches"**. 10th International Conference on Network and Service Management (CNSM) and Workshop. 2014.

[51] G. Faraci, and G. Schembra, **"An analytical model to design and manage a green SDN/NFV CPE node"**. IEEE Transactions on Network and Service Management, 2015. 12(3): pp. 435-450.

[52] J. Ansell, W.K. Seah, B. Ng, S. Marshall **"Making Queueing theory more palatable to SDN/OpenFlow-based network practitioners"**. IEEE/IFIP Network Operations and Management Symposium. 2016.

[53] S. Wei, C. Fung, "FlowRanger: **A request prioritizing algorithm for controller DoS attacks in Software-Defined Networks"**. In 2015 IEEE International Conference on Communications (ICC) 2015, pp. 5254-5259.

[54] pp. Zhang, H. Wang, C. Hu, C. Lin **"On denial of service attacks in Software-Defined Networks"**. IEEE Network. 2016, 30(6), p.28-33.

[55] Q. Yan, Q. Gong, F.R. Yu. **"Effective Software-Defined Networking controller scheduling method to mitigate DDoS attacks"**. Electronics Letters. 2017, 53(7), pp. 469-471.

[56] T. Eom, J.B. Hong, J.S. Park and D.S. Kim, **"Security and Performance Modeling and Optimization for Software-Defined Networking"**. In 2019 18th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, Aug., 2019, pp. 610-617.

[57] T. Wang, Z. Guo, H. Chen, W. Liu, **"BW Manager: Mitigating denial of service attacks in Software-Defined Networks through bandwidth prediction"**, IEEE Transactions on Network and Service Management. 2018, 15(4), pp. 1235-1248.