Resilience Analysis of Software-Defined Networks Using Queueing Networks

Livinus Obiora Nweke
Information Security and Communication Technology
Norwegian University of Science and Technology (NTNU)
Gjøvik, Norway
livinus.nweke@ntnu.no

Stephen D. Wolthusen
School of Mathematics and Information Security
Royal Holloway, University of London
Egham, United Kingdom
Information Security and Communication Technology
Norwegian University of Science and Technology (NTNU)
Gjøvik, Norway
stephen.wolthusen@rhul.ac.uk, stephen.wolthusen@ntnu.no

***Abstract*—Software-Defined Networks (SDN) are being adopted widely and are also likely to be deployed as the infrastructure of systems with critical real-time properties such as Industrial Control Systems (ICS). This raises the question of what security and performance guarantees can be given for the data plane of such critical systems and whether any control plane actions will adversely affect these guarantees, particularly for quality of service in real-time systems. In this paper we study the existing literature on the analysis of SDN using queueing networks and show ways in which models need to be extended to study attacks that are based on arrival rates and service time distributions of flows in SDN.**

***Index Terms*—Software Defined Networks, Queueing Theory, Queue Networks, Real-Time Systems, Industrial Control Systems**

I. Introduction

Networks with hard and firm real-time constraints such as industrial control systems networks historically relied upon dedicated field bus systems with limited performance but well-defined, deterministic characteristics. In recent years, cost and performance considerations have encouraged migration to standard, if heavily over-provisioned and somewhat modified, networks such as Industrial Ethernet variants (e.g. PROFINET IRT or EtherCAT). As network infrastructures outside this domain increasingly rely on the advantages offered by software-defined networks (SDN) and network function virtualisation (NFV), it is likely that this will also need to be considered for real-time environments.

However, the more flexible and dynamic SDN architecture not only offers benefits, but also raises questions on the ability to satisfy security and quality of service (QoS) guarantees, particularly in the presence of malfunctioning or malicious entities interfering with control or data plane.

The purpose of this paper is to analyse the existing literature on the analysis of SDN using queueing networks and to show how the models can be extended to study attacks that are based on arrival rates and service time distributions of flows in SDN. We only consider attacks that can be analysed at the flow abstraction level, without considering the semantics of the flows, a limitation that is inherent in using queueing networks. Although simulations and experimentation are desirable for the study of SDN, analytical modelling permits the study of a wider range of configurations and parameters as well as the optimisation not only of performance [1], but also an understanding of the severity of attacks.

Understanding the performance of SDN through its analysis using queuing networks is of particular importance for systems with strong QoS requirements. These requirements include particularly the delay, loss, and jitter parameters and respective requirements for different types and classes of service [4]. Systems that will fail if the QoS requirements are not met are referred to as hard real-time systems [5]. Although it is easy to apply queuing networks to the general performance analysis of most SDN architectures, systems with hard real-time requirements must be more comprehensive to capture all relevant interactions. Many of the QoS requirements in hard real-time systems, moreover, are immediately usable for security considerations since the feasibility and effort required by adversaries for denial of service (DoS) attacks, both immediately and transitively, are a particular concern.

We therefore provide a review on the analysis of SDN using queueing networks in this paper, analysing the still nascent body of work currently emerging in this domain. We note that thus far the focus is on relatively straightforward models such as the M/M/1 model (using Kendall's notation) being used as the queueing model in characterising the SDN behaviour, with M/G/1 and GI/M/1/K models considered more appropriate model for SDN controllers and switches, respectively. This appears to be a lacuna in existing work as a more precise characterisation of both arrival processes and service time distributions for regular operation, configuration changes, and adversarial action is thus far not being considered. This, however, is critical to understand vulnerabilities and security requirements.

The rest of this paper is organised as follows. In section II, basic concepts and terminologies use in this paper are discussed. A literature review of the works on analysis of SDN using queueing networks is presented in section III. Section IV presents our proposed models and metrics for security analysis of SDN using queueing networks. Section V provides an illustration using DoS attack, on how the models proposed

may be applied to study attacks that are based on arrival rates and service time distributions of the packet flows in SDN. Section VI offers a brief discussion on the results from the survey and the lessons learned. Finally, section VII concludes the paper and presents future works.

II. Background

This section discusses the basic concepts and terminologies used in this survey. In particular, SDN and queueing theory are contextualised.

A. Software-Defined Networks (SDN)

SDN is a current network architecture in seeking to decouple the control plane from the data plane. Unlike in a traditional network architecture where control and data plane are embedded in the networking devices, SDN separates roles such that networking devices can become purely forwarding devices with the forwarding instructions pushed to them via the control plane and allowing the use of commodity components [6]. The main goals of the SDN architecture were to simplify the deployment of control plane functions and to enable the applications to deal with a single abstracted network device without concern for the implementation details whilst lowering dependency on dedicated components [7]. Thus, with SDN, network control functions become directly programmable enabling the automation of network functions which in turn facilitate the building of highly scalable and flexible networks that can readily adapt to the dynamic nature of today's environment [8].

The SDN architecture sketched in Figure 1, comprises of the application layer, control layer, and infrastructure layer [6]. The SDN applications exist in the application layer, and interact with the control layer via the northbound interfaces. In the middle of the SDN architecture is the SDN controller, which translates applications' requirements and exerts low-level control over the network elements, while providing relevant information up to the SDN applications. The infrastructure layer comprises of the network elements, which expose their capabilities toward the control layer via the southbound interfaces. It can be inferred from this setup that the network intelligence is logically centralised in the SDN controllers which maintain a global view of the network [6]. Therefore, the network appears to the applications and policy engines as a single, logical switch; and the network devices no longer need to understand and process several protocols but merely accept instructions from the SDN controllers [8].

What is not visible from the illustration above is that a reasonable large SDN architecture will have multiple controllers and several number of switches. Also, the architecture does not show the information flows and communication between these components. Further, the realisation of the concept of SDN entails that two requirements must be met: the need for a common logical architecture in the network devices to be managed by the SDN controller and the need for a standard, secure protocol between the SDN controller and the network devices (and further on to the application layer) [9]. These

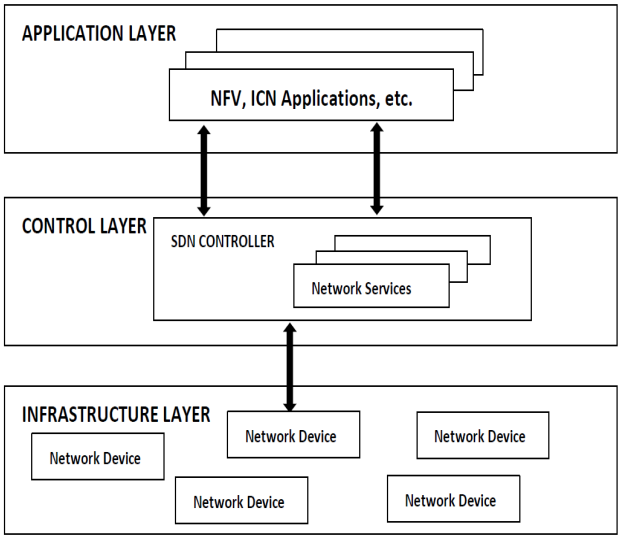

Fig. 1. SDN Architecture

requirements are addressed e.g. by OpenFlow [10], which is both a protocol between SDN controllers and network devices, as well as a specification of the logical structure of the network switch function. The protocol specifies how the applications can directly access and manipulate the network devices via the SDN controller without regards to the details of how the switch are implemented, and also it uses the concept of flows to identify network traffic based on pre-defined match rules that can be statically or dynamically programmed [10].

The SDN architecture and OpenFlow standard provide an open architecture in which control functions are separated from the network devices and placed in a logically centralised controller. The centralisation of the network state in the control plane provides the flexibility to configure, manage, secure, and optimise network resources with far-reaching automation. They also enable the underlying infrastructure to be abstracted for applications and network services, enabling the network to be treated as a logical entity [8].

B. Queueing Theory

Queueing theory started by the study of queues, devising analytical mechanisms and tools for the design and evaluation of the performance of queueing systems [2]. To characterise a queueing system, there is need to identify the probabilistic properties of the arrival processes, service times and service disciplines [2]. Conventionally, the arrival process is characterised by the distribution of the inter-arrival times of the customers. These inter-arrival times are usually assumed to be independent and identically distributed random variables. They are denoted by A(t):

$$A(t) = P(\textit{interarrival time} < t).$$

The service time is used to express how long the service will take. It is usually assumed that the service time for a customer is independent and does not depend upon the arrival

process. Another common assumption about service time is that it is exponentially distributed. Its distribution function is denoted by B(x):

$$B(x) = P(\textit{service time} < x).$$

The service discipline is used to decide how the next customer waiting on the queue is served. The most common methods used include: First-in, First-out (FIFO); Last-come, First-out (LIFO); Random Service (RS); Priority.

Kendall's notation is the standard notation used to describe and classify queuing systems [2]. It is given as:

$$A/B/m/K/n/D,$$

where; A is the distribution function of the inter-arrival times, B is the distribution function of the service times, *m* is the number of servers, K is the capacity of the system, *n* is the population size, and D is the service discipline.

It is common practice to denote exponentially distributed random variables by *M* meaning Markovain or memoryless [11]. If the population size and the capacity are infinite, the service discipline is FIFO and is usually omitted. Thus, M/M/1 denotes a system with Poisson arrivals, exponentially distributed service times and a single server [2]. Basic queueing systems are used to describe the system as a unique resource, but to describe the system as a set of interacting resources; queueing networks are usually used. A queueing network can be defined as a collection of service centres representing the system resources which are used provide service to a collection of customers that represent the users [2]. Queueing networks have been shown to be appropriate tool for system performance evaluation and have also been shown as being helpful in the modelling of attacks against distributed systems.

III. State-of-art in Analysis of SDN using Queueing Networks

Whilst queue network research spans decades, here we concentrate on queueing distributions for modelling software-defined networks as a method for characterisation and categorisation as this is highly pertinent to the adaptation for security modelling

A. Modelling Based on M/M/1 Distribution

The M/M/1 distribution consists of a Poission arrival, one exponential server, infinite FIFO queue and unlimited customer population. Although, these are very strong assumptions not satisfied by real systems, they provide useful insights which may be used to study how real systems will perform given certain parameters. The first known analysis of SDN using queueing model, presented by Jarschel et al. in [12] is based on M/M/1 distribution. The model assumed that packet arrivals form Poisson streams and that the queue length of the controller system is finite so as to model the possibility of dropped packets under high load condition. In addition, the model has some limitations in that it does not capture the fact that packets arriving at the switch are queued one queue per port and that it is limited to a single switch per controller. The results obtained from the analytical model were validated using a packet based simulation in OMNeT++.

Some of the limitations in [12] were subsequently addressed by Chilwan, Mahmood, Østerbø and Jarschel in [13]. It was achieved by modelling both the controller and the switch as Jackson Network, with some modifications to suit OpenFlow-based SDN. Whilst maintaining the assumptions made in the previous paper, the authors also showed that the model can be extended to handle the case where more than one switch exist in the data plane. The model was then validated using a discrete event simulation which resembles the queuing behaviour of the proposed model. In addition, the model was further extended by Mahmood, Chilwan, Østerbø and Jarschel in [14] for the performance analysis of OpenFlow-based SDN with multiple nodes in the data plane.

Yen and Su in [15] proposed SDN based cloud computing architecture which was implemented using open source Open vSwitch and POX controller packages. The queueing network model based on M/M/1 was used to model the operation of the OpenFlow architecture to show the correctness of the architecture. In this model, it was assumed that the arrival rate of cloud service request is an exponential distribution and that the service rate of both the Open vSwitch and the POX controller was exponentially distributed. Considering that the average packet queue length is an important parameter of the SDN-based cloud environment, it was used as the performance metric for the evaluation of the architecture. The numerical results of the average queue length presented in the work show that the proposed SDN-based cloud computing architecture can provide QoS guarantees for cloud services.

A Preemption-based Packet Scheduling (P^2S) scheme to improve global fairness and reduce packet loss rate in SDN data plane was presented by Miao, Min, Wu, and Wang in [16]. The performance of the proposed scheme was analysed using queueing networks based on M/M/1 distribution. In this model, the arrival and departure of packets in the system were characterised as a birth-death process. Miao et al. assumed that the packet arrivals followed a Poisson distribution, and that the service time follows a negative exponential distribution. The performance of the P^2S scheme was compared to the traditional FIFO scheme in terms of packet loss probability and service fairness. The results presented in the work showed that FIFO cannot guarantee fairness among packets and suffers from high packet loss, while P^2S scheme offers priority for the packets from the controller and as such, achieves better performance in terms of global fairness and packet loss probability. Also, the performance of the model was validated using simulations by varying the traffic arrival rate, flow table hit probability and the service rates of the switch and controller.

Fahmin, Lai, Hossain, Lin and Saha in [17] presented the performance modeling of SDN with network virtualisation function (NFV) under or aside the controller, which are the different methods of combining SDN with NFV; using M/M/1 queueing model. Fahmin et al. used queueing theory to develop mathematical models for both scenarios resulting to queueing

networks which were then analysed. In this model, they assumed that the arrival process at the switch follows Poisson process, the service time of packets in switch, controller and virtualised network function (VNF) follow exponential distribution, and the queue size of a switch, controller and VNF is infinite. The average packet delay was used as the performance metric for the analysis of both approaches. Simulation was used for the validation of the analytical process. Also, the results presented showed that the packet delay for SDN with NFV aside the controller is significantly less than that for SDN with NFV under the controller, and that the service rate at VNF does not affect the delay gap between SDN with NFV aside and under the controller.

B. Modelling Based on GI/M/1/K Distribution

The GI/M/1/K distribution is a type of queueing distribution where inter-arrival times are independent and identically distributed with general distribution, and service times are independent and exponentially distributed. Also, the system is made of a finite waiting space and the arriving customers are served on a FIFO basis. The properties of GI/M/1/K distribution is suitable for the study of packet arrivals at SDN switch. Goto, Masuyama, Ng, Seah, and Takahashi in [1] proposed a queueing model of an OpenFlow-based SDN that takes into account classful treatment of packets arriving at a switch. They argued that the different packets arriving at the switch should be treated differently and to that end, they classified these packets as follows: external packets arriving at the switch according to a Poisson process, Class S; packet whose forwarding information is missing in the flow table and are forwarded to the controller, Class C; and packets processed by the controller and sent back to the switch, Class F. The switch was then model using GI/M/1/K distribution that can enqueue no more than K1 packets. In this model, three performance measures, namely, packet loss probabilities in Class S and F, and average packet transfer delay through the system was used in the analysis. Simulation was used in the validation of the model and the simulation results presented matches the average delay obtained by the queueing analysis and thus confirming the validity of the model.

Queueing model was used by Singh, Ng, Lai, Lin and Seah in [18] to investigate the effect of a buffer sharing in SDN switch. Using GI/M/1/K distribution for modelling the switch, they proposed two models: Shared Buffer referred to as SE Model; and Priority Queueing Buffer referred to as SPE Model. In this work, Model SE uses a single queue for the switch while Model SPE uses priority queue for the switch. They assumed that the controller had an infinite capacity queue and that the external packet arrival at the switch follows Poisson process. The relative minimum capacity and relative time to install the flow table entries were used as the performance measures to compare the performance of the models. Discrete event simulation of the SE and SPE queuing networks was used in the validation of the analytical results.

Singh, Ng, Lai, Lin and Seah in [19], presented a unified queueing model for characterizing the performance of hardware switches and software switches in SDN. They started by first modelling SDN with software switch and hardware switch, and then used queuing models to model the software (SPE) and hardware (HPE) SDN switches. They assumed that the controller had an infinite capacity queue and that the external arrival at the switch follow Poisson process. In this model, the switches were modelled with GI/M/1/K distribution, however, M/M/1/K distribution was added in modelling the hardware processor of the hardware switch. The average packet transfer delay was used as the performance metric for comparing the models. They validated the accuracy of the analytical results for both models using simulation. The results from the study show that a hardware switch performs better than a software switch in terms of average delay and packet loss probability.

C. Modelling Based on M^X/M/1 Distribution

The M^X/M/1 distribution is a type of M/M/1 distribution with batch arrivals of random size. The arrival stream forms a Poisson process and the batch size is a random variable. Xiong, Yang, Zhao, Li, and Li in [20] investigated the packet arrival process and forwarding procedure at an OpenFlow switch. They modelled its packet forwarding performance using M^X/M/1 distribution. In this work, they argued that packet arrivals at the switch does not follow Poisson process but instead results in packet batch arrivals. They assumed that the packet arrived at the switch as Poisson stream, the number of packets in a switch conforms to Poisson distribution, and the packet processing time of the switch conforms to negative exponential distribution. The average sojourn time and average queue length were used as the performance measures for evaluating the performance of the model. Also, Bilen, Ayvaz, and Canberk in [21] used M^X/M/1 distribution to model elephant flows. They proposed a distribution flow management model in SDN ultra-dense network based on queueing theory.

D. Modelling Based on M/Geo/1 Distribution

The M/Geo/1 distribution comprises of Poisson distribution and the service times obey geometric distribution. Sood, Yu, and Xiang in [22] proposed an analytical modelling based on M/Geo/1 distribution to study the performance of SDN switches. They assumed that the packet arrival at the switch followed Poisson process and the service times obey geometry distribution. In this model, they used the Embedded Markov Chain theory and applied it to M/Geo/1 queue to obtain the number of packets in the system and the service time. The model was used for just investigating the average response time of SDN switch without considering the switch-controller interaction. Thus, the average flow response time was used as the performance metric for evaluating the performance of the switch. Simulations were used for the validation of the model and both the simulation and analytical results presented matched each other. They concluded by noting that the important factors that determine the switch's mean response time are packet arrival rate, number of flow-table entries, and the position of the targeting rule by a corresponding packet.

E. Modelling Based on M/G/1 Distribution

In M/G/1 distribution, the arrival process is Poisson and the service time for each customer is generally distributed. Also, the distribution has an infinite queueing capacity and unlimited customer population. The packet-in message process of SDN controller was modelled by Xiong, Yang, Zhao, Li, and Li in [20]. They studied the arrival process and serving process of packet-in messages in SDN controller, and modelled the SDN controller performance with the M/G/1 distribution. In this model, they assumed that the packet-in messages at the controller from its switches constituted a Poisson stream and that the processing time of packet-in messages in the controller conforms to normal distribution. The performance of the controller was evaluated using the publicly available benchmark Cbench and the sojourn time of a packet-in message was used as the performance metric. Also, they compared the performance of their model with the most common model (M/M/1) used in characterizing the performance of SDN controllers. The results presented showed that M/G/1 provided more accurate approximation of the SDN controller performance than M/M/1. In the same way, Javed, Iqbal, Saleh, Haider and Ilyas in [23] demonstrated through experiments that M/G/1 distribution using log-normal distribution mixture as the service distribution is closer to reality in terms of SDN controller performance evaluation than M/M/1 distribution.

IV. Models and Metrics for Security Analysis of SDN Using Queueing Networks

It is clear that it is inappropriate to use the already-strong assumptions on distributions for regular network traffic discussed in section III for adversarial traffic. We therefore propose to expand the models such that they can capture with additional traffic caused by attacks, differentiating this traffic as necessary. Within the limits of not considering the semantics of the flows themselves, what is needed to be extended are primarily arrival rates and service time distributions as the models reviewed above are too simple to capture such adversarial behaviour. We hence propose two modelling levels: one based on aggregating the distribution where we are primarily interested in breaches of QoS guarantees, and a more refined model which explicitly captures SDN behaviour by studying queue network representations of the SDN architecture, and distinguishing between baseline and adversary flows *with different arrival rates and service time distributions*.

A. Modelling Based on Aggregating the Distribution

It is possible to modify the arrival process of the existing models to capture the situation where adversarial actions may occur. The proposed model is similar to work done in [24], where the input to the queueing network model is considered as a combination of both the normal traffic and malicious traffic. The addition of the two probability distribution functions is justified by the fact that the distribution of a malicious traffic will not be the same as the normal traffic because of the way the attacker would craft the malicious packet to evade detection. Hence, it is not plausible to use Poisson distribution to describe the distribution of both the normal traffic and malicious traffic. Therefore, in order to capture the behaviour of an attacker, we need to as a minimum, use a different type of distribution to describe malicious traffic and then calculate the impact using the aggregate distribution as input to the queueing network model as illustrated in figure 2.

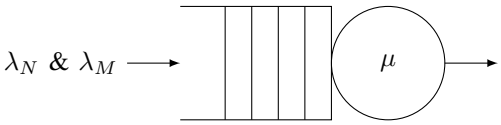

Fig. 2. Proposed Model Based on Aggregating the Distribution

B. Modelling Based on Analysing the effects of the two Queues Separately

The effects of the two queues and distinct service times can be studied separately to understand adversarial actions. This can be achieved by distinguishing between regular flows and adversarial flows, which also cause flows internal to the SDN architecture primarily in the form of additional messages sent from controllers to switches where new or altered flows must be accommodated before actually considering the aggregate flows through the switching fabric. Once a new flow is established, the malicious packet can be treated as additional traffic as above, but aggregate service times become particularly interesting in establishing possible breaches of QoS requirements, minimizing the effort required on the part of adversaries.

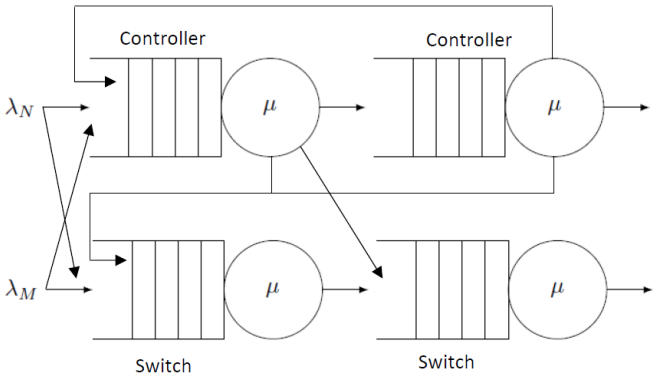

Fig. 3. Proposed Model Based on Analysing the effects of the two Queues Separately

V. Towards Security Analysis of SDN Using Queueing Networks

The previous section have identified the areas of SDN security that are amenable to study using queueing networks. In this section, we use DoS attacks to illustrate how the models proposed may be applied to study attacks that are based on arrival rates and service time distributions of the packet flows in SDN. To the best of our knowledge, queueing network

models and other analytical approaches have not been used in the literature to evaluate real-time processing security issues in SDN in particular. Although network calculus and queueing networks have been used in security research, the flexibility that comes with using well-established results from queueing theory justifies our choice for proposing the use of queueing networks for the study of DoS attacks in SDN, similar to the works done in other fields [24]–[29].

DoS attacks are among the security issues in SDN that can be captured easily using queueing networks. SDN can be characterized as queueing networks model to capture the interaction between its different components. Using the model based on aggregating the distribution proposed in the previous section, we can then explicitly model the two types of flows. The distribution for a typical DoS attacks is not a uniform distribution because of the way the attacker crafts the attacks. Then combining both the regular traffic distribution, which is usually assumed to be Poisson; and the adversarial distribution, we can then study the specific characteristics of the attacker.

Additional degree of flexibility would have to be introduced to the queueing networks model to capture the behaviour of SDN under DoS attacks. A typical approach that may be adopted from the queueing models presented in this paper is to use queueing distributions with finite capacity. The introduction of queues with finite capacity will facilitate the modelling of SDN under DoS attacks as was done in [26]–[28]. Hence, M/M/1/K or M/G/1/K may be used in characterising the behaviour of both the data plane and control plane. In using M/M/1/K or M/G/1/K for modelling the behaviour of both the data plane and control plane, blocking or loss probability (P_{loss}) may be deployed as the security metrics for studying the security properties of SDN. For example, if P_{loss} is large, SDN may be said to be under DoS attacks.

Also, the approach used in [26], which involves the use of a threshold value that is greater than 0, may be applied to provide information about the SDN security status. If P_{loss} is less than the threshold value, it can be concluded that the SDN is not under DoS attacks. However, if P_{loss} is greater than or equal to the threshold value, implying that the SDN resources is exhausted; it can be concluded that the SDN is under DoS attacks.

Other types of security challenges that can be studied using queueing networks are described in [24], [25], [29]. In particular, the framework proposed in [25] and used in [24] could be used to study some security issues in SDN as the authors opined that the framework allows for the study of DoS and performance degradation, which can also be suffered by SDN. Therefore, queueing networks offer a promising approach to studying the security properties of SDN in order to provide the basis for it deployments in industrial environments where there are stringent security and QoS requirements.

VI. Discussion

The goal of this study has been to evaluate the different methods in the literature employed for the analysis of SDN using queueing networks and to categorise them based on the queuing distribution deployed for the analysis as this is most pertinent for subsequent extension into studies of security properties. The categorisation showed that simple M/M/1 distribution is the most widely adopted model for characterizing the behaviour of SDN switch and controller. Although the distribution facilitates an easy analytical process, it seems to be a poor fit for evaluating the performance of SDN switch and controller. Thus, it is important that a more realistic approach for characterising the behaviour of SDN switch and controller is investigated.

Already, the authors in [14] suggested at the completion of their work with M/M/1 that M/G/1 is a more appropriate model for characterizing the behaviour of SDN controller that needs to be investigated. The authors in [20] went further to use M/G/1 to model the performance of SDN controller. In their work, they compare the performance of the M/G/1 with M/M/1 and the results from the work support the initial suggestions made by authors in [14] that M/G/1 is a more accurate approximation of the SDN controller performance than M/M/1. Also, the results from the experiments conducted in [23] further validate the claim that M/G/1 is a better fit than M/M/1 for the evaluation of the performance of SDN controller.

In the case of SDN switches, an appropriate model needs to consider both the external packet arrival rate at the switch and the packet arrival at the switch from the controller, as this will ensure that the QoS requirements of SDN environment is properly captured. Unfortunately, most of the work studied did not address this concern in their modelling of SDN switches, which is a limitation when seeking to analyse the interaction with adversarial packet arrival. However, authors in [1], [18], [19] used GI/M/1/K distributions to correctly model the packet arrival at the switch, taking into account both the external packet arriving at the switch and the packet arriving at the switch from the controller. These works suggest that GI/M/1/K distribution is a more appropriate model for characterising the behaviour of SDN switch.

Moreover, most of the earlier work reviewed relies on heavily simplified models, which do not allow insight into the internal functioning of SDN. This is because they treat flows architecture elements as black boxes, which is interesting for security analysis as an attacker may explicitly target those flows. Although none of the works presented in this survey used queueing networks for the analysis of the security requirements of SDN, earlier work [24]–[29] showed that it is possible to study DoS, DDoS, performance degradation, de-synchronization attacks, injections attacks, tails attacks, and economic denial of sustainability (EDoS) attacks using queueing networks. This can also be applied for the analysis of SDN to discover attacks and their potential solutions.

We also note that all work considered so far captured only SDN configurations with a single controller connected to one or more switches; this is not particularly realistic, particularly in critical systems requiring redundancy for resiliency. Such deployments should be seen as network of queues which could lead to analytically modelling them as queueing networks.

Hence, there is need to consider such deployments as none of the existing works addressed them because such realistic modelling will provide better performance evaluation and lead to design of a more resilient SDN architecture.

VII. Conclusions and Future Work

Research in the analysis of SDN using queueing networks is still a developing field, and clearly refinement particularly for analyses relevant for security are still called for. In this paper, we have undertaken a survey of existing work on the use of queueing networks for the analysis of SDN with a focus on the suitability of queueing disciplines to understand effective quality of service modelling and subsequent study of the security properties of SDN.

Future work will include the development of queue networks suitable for systems with hard real-time requirements and where adversarial action may occur. In such systems, there are deadlines that need to be met so as not to violate the QoS guarantees of such systems; the latter brings with it that a number of standard assumptions made in queueing theory and queue networks can no longer be fully supported, mainly on probability density functions and independence assumptions. We then seek to use queueing networks for modelling attacks against such systems. The understanding of these issues will help to address threats to reliability, resilience, and security that may arise from adopting SDN for systems with hard real-time requirements and thus, accelerate the deployment of SDN for such systems.

References

[1] Y. Goto, H. Masuyama, B. Ng, W. K. G. Seah, and Y. Takahashi, "Queueing analysis of software defined network with realistic openflow–based switch model," in *Proc. Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS) 2016 IEEE 24th Int. Symp. Modeling*, Sep. 2016, pp. 301–306.

[2] J. M. Smith, *Introduction to Queueing Networks*. Springer, Cham, 2018.

[3] J. Ansell, W. K. Seah, B. Ng, and S. Marshall, "Making queueing theory more palatable to sdn/openflow-based network practitioners," in *NOMS 2016-2016 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 2016, pp. 1119–1124.

[4] M. Karakus and A. Durresi, "Quality of service (qos) in software defined networking (sdn): A survey," *Journal of Network and Computer Applications*, vol. 80, pp. 200–218, 2017.

[5] Y. Chen, T. Farley, and N. Ye, "Qos requirements of network applications on the internet," *Information Knowledge Systems Management*, vol. 4, no. 1, pp. 55–76, 2004.

[6] O. N. Foundation, "Sdn architecture," Open Networking Foundation, Tech. Rep., 2014.

[7] O. N. Fundation, "Software-defined networking: The new norm for networks," *ONF White Paper*, vol. 2, pp. 2–6, 2012.

[8] S. William, "Software-defined networks and openflow," *The Internet Protocol Journal*, 2013.

[9] d. José Roberto, S.-B. Germán, and S.-P. Josep, "A critical review of openflow/sdn-based networks," in *Transparent Optical Networks (ICTON), 2014 16th International Conference on*. IEEE, 2014, pp. 1–5.

[10] O. Foundation, "Openflow specification," Open Networking Foundation Publication, Tech. Rep., 2015.

[11] M. Pinsky and S. Karlin, *An introduction to stochastic modeling*. Academic press, 2010.

[12] M. Jarschel, S. Oechsner, D. Schlosser, R. Pries, S. Goll, and P. Tran-Gia, "Modeling and performance evaluation of an openflow architecture," in *Proceedings of the 23rd international teletraffic congress*. International Teletraffic Congress, 2011, pp. 1–7.

[13] A. Chilwan, K. Mahmood, O. N. Østerbø, and M. Jarschel, "On modeling controller-switch interaction in openflow based sdns," *International Journal of Computer Networks & Communications*, vol. 6, no. 6, p. 135, 2014.

[14] K. Mahmood, A. Chilwan, O. Østerbø, and M. Jarschel, "Modelling of openflow-based software-defined networks: the multiple node case," *IET Networks*, vol. 4, no. 5, pp. 278–284, 2015.

[15] T. Yen and C. Su, "An sdn-based cloud computing architecture and its mathematical model," in *Proc. Electronics and Electrical Engineering 2014 Int. Conf. Information Science*, vol. 3, Apr. 2014, pp. 1728–1731.

[16] W. Miao, G. Min, Y. Wu, H. Wang, and J. Hu, "Performance modelling and analysis of software-defined networking under bursty multimedia traffic," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 12, no. 5s, p. 77, 2016.

[17] A. Fahmin, Y. Lai, M. S. Hossain, Y. Lin, and D. Saha, "Performance modeling of sdn with nfv under or aside the controller," in *Proc. 5th Int. Conf. Future Internet of Things and Cloud Workshops (FiCloudW)*, Aug. 2017, pp. 211–216.

[18] D. Singh, B. Ng, Y. Lai, Y. Lin, and W. K. G. Seah, "Modelling software-defined networking: Switch design with finite buffer and priority queueing," in *Proc. IEEE 42nd Conf. Local Computer Networks (LCN)*, Oct. 2017, pp. 567–570.

[19] D. Singh, B. Ng, Y.-C. Lai, Y.-D. Lin, and W. K. Seah, "Modelling software-defined networking: Software and hardware switches," *Journal of Network and Computer Applications*, vol. 122, pp. 24–36, 2018.

[20] B. Xiong, K. Yang, J. Zhao, W. Li, and K. Li, "Performance evaluation of openflow-based software-defined networks based on queueing model," *Computer Networks*, vol. 102, pp. 172–185, 2016.

[21] T. Bilen, K. Ayvaz, and B. Canberk, "Qos-based distributed flow management in software defined ultra-dense networks," *Ad Hoc Networks*, vol. 78, pp. 24–31, 2018.

[22] K. Sood, S. Yu, and Y. Xiang, "Performance analysis of software-defined network switch using $m/geo/1$ model," *IEEE Communications Letters*, vol. 20, no. 12, pp. 2522–2525, Dec. 2016.

[23] U. Javed, A. Iqbal, S. Saleh, S. A. Haider, and M. U. Ilyas, "A stochastic model for transit latency in openflow sdns," *Computer Networks*, vol. 113, pp. 218–229, 2017.

[24] J. G. Wright and S. D. Wolthusen, "Stealthy injection attacks against IEC61850's goose messaging service," in *Proc. IEEE PES Innovative Smart Grid Technologies Conf. Europe (ISGT-Europe)*, Oct. 2018, pp. 1–6.

[25] J. G. Wright and S. D. Wolthusen, "De-synchronisation attack modelling in real-time protocols using queue networks: Attacking the iso/iec 61850 substation automation protocol," in *International Conference on Critical Information Infrastructures Security*. Springer, 2017, pp. 131–143.

[26] Y. Wang, C. Lin, Q.-L. Li, and Y. Fang, "A queueing analysis for the denial of service (dos) attacks in computer networks," *Computer Networks*, vol. 51, no. 12, pp. 3564–3573, 2007.

[27] S. Hao, H. Song, W. Jiang, and Y. Dai, "A queue model to detect ddos attacks," in *Proceedings of the 2005 International Symposium on Collaborative Technologies and Systems, CTS 2005, Saint Louis, Missouri, USA, May 15-20, 2005*, W. K. McQuay and W. W. Smari, Eds. IEEE Computer Society, 2005, pp. 106–112.

[28] H. Shan, Q. Wang, and C. Pu, "Tail attacks on web applications," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017*, B. M. Thuraisingham, D. Evans, T. Malkin, and D. Xu, Eds. ACM, 2017, pp. 1725–1739.

[29] F. Al-Haidari, K. Salah, M. Sqalli, and S. M. Buhari, "Performance modeling and analysis of the edos-shield mitigation," *Arabian Journal for Science and Engineering*, vol. 42, no. 2, pp. 793–804, 2017.