Modelling Adversarial Flow in Software-Defined Industrial Control Networks Using a Queueing Network Model

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*—In recent years, software defined networking (SDN) has been proposed for enhancing the security of industrial control networks. However, its ability to guarantee the quality of service (QoS) requirements of such networks in the presence of adversarial flow still needs to be investigated. Queueing theory and particularly queueing network models have long been employed to study the performance and QoS characteristics of networks. The latter appears to be particularly suitable to capture the behaviour of SDN owing to the dependencies between layers, planes and components in an SDN architecture. Also, several authors have used queueing network models to study the behaviour of different application of SDN architectures, but none of the existing works have considered the strong periodic network traffic in software-defined industrial control networks. In this paper, we propose a queueing network model for software-defined industrial control networks, taking into account the strong periodic patterns of the network traffic in the data plane. We derive the performance measures for the analytical model and apply the queueing network model to study the effect of adversarial flow in software-defined industrial control networks.**

***Index Terms*—Adversarial flow, SDN, QoS, Industrial control systems, Queueing network model**

I. Introduction

Recent advances in networking technology have witnessed a gradual shift in industrial control networks from serial-based communication to Ethernet technology. This provides the opportunity of exploiting the benefits of SDN for industrial control networks because the correctness of industrial control system protocols rely on the guaranteed QoS. SDN is a network paradigm in which “the control and data plane are decoupled, network intelligence and state are logically centralized, and the underlying network is abstracted from the applications” [1]. The separation of the control and data plane provides several advantages which include easier network management, network programmability, increased visibility into the network, efficient use of network resources and dynamic updating of network policies. However, SDN not only offers benefits but also raises questions about the ability to satisfy the guaranteed QoS in the presence of adversarial flow.

Adversarial flow is an additional flow usually malicious, that may be introduced into the SDN architecture by an attacker. The arrival rate and service time distribution of the adversarial flow can be manipulated by the attacker to achieve desired ends. We had described the idea of adversarial flow in our previous work [2] and opined that modelling based on aggregating distribution or modelling based on analysing the effect of two queues separately may be deployed to study the effect of adversarial flow in the analysis of SDN using queueing network models.

Queueing network models have been employed in the literature for performance evaluation of different application of SDN architectures [3]–[7]. The approach adopted by several authors involves the use of well-established results from queueing theory to study the interaction between the forwarding plane switches and the control plane controllers. Although simulations and experimentations could be desirable, analytical models, allow the study of wider configurations and how different parameter choices would affect the behaviour of network traffic.

The existing literature on the analysis of SDN using queueing network models has examined the different features of SDN and also its application in different domains. However, *none of the existing literature has considered software-defined industrial control networks and the strong periodic patterns of the network traffic*. Also, the effect of adversarial flow in SDN architectures has not been investigated using queueing network models. We had proposed the use of queueing network models to study the effect of adversarial flow in [2]. We exploit the idea here, to investigate the effect of adversarial flow in software-defined industrial control networks.

In this paper, we propose an analytical model for software-defined industrial control networks taking into account the strong periodic patterns of the network traffic in the data plane. Analytical modelling of software-defined industrial control networks provides useful insights for benchmarking and facilitates the identification of factors that could cause the network to breach the stringent QoS requirements. In addition, we use the analytical model to study the effect of adversarial flow in software-defined industrial control networks.

The rest of this paper is organised as follows. Section II presents related works on the analysis of SDN using queueing network models. Section III describes the queueing network model used in this paper to model the behaviour of network traffic in software-defined industrial control networks and the performance measures. Section IV discusses the application of the developed queueing network model to investigate the effect of adversarial flow in software-defined industrial control networks. Section V concludes the paper and presents future works.

II. Related Works

The first known analytical modelling of SDN using a queueing network model is presented by Jarschel et al. in [8]. The work focus on characterizing the interaction between SDN switch and the controller without consideration for any specific application of the SDN architecture. In the same way, authors in [9]–[12] employ queueing network models to study different features of SDN. The tradeoffs between buffer sharing mechanisms are investigated using queueing network models in [9]. The authors in [10] propose a queueing network model to characterize the performance of hardware switches and software switches in SDN. The paper in [11] presents the performance study of SDN switches using a queueing network model. And the authors in [12] use a queueing network model to examine the performance of SDN with network virtualization function under or aside the controller.

Yen and Su in [3] deploy a queueing network model to examine SDN-based cloud computing architecture. The queueing network model is based on M/M/1 (using Kendall's notation) queue, and they demonstrate it is appropriate for modelling the operation of SDN-based cloud computing architecture. Also, they show that the proposed SDN-based cloud computing architecture can provide QoS guarantees for cloud services. Chowdhary and Huang in [4] use a queueing network model to consider SDN-based network function parallelism in the cloud. They use a M/M/c queue for optimizing the service function allocation for every service function chain and show that service functions with independent action sets can be parallelized to reduce the performance overhead.

Queueing network models have also been used to study the behaviour of SDN in ultra dense network and satellite communication networks [5], [6]. The authors in [5] examine the use of SDN to ease the management of ultra dense data plane with distributed controllers. They propose a distributed flow management model based on queueing network model and characterize the distributed controllers by considering the flow characteristics and outage. M/M/1 and M^X/M/1 queues are use to model the incoming mice and elephant flows respectively, with an additional M/M/c queue for detection of an outage. In [6], the authors propose an analytical model for software defined satellite networks using a queueing network model. They place the controllers on geosynchronous earth orbit (GEO) satellites and the forwarding functions on medium earth orbit (MEO) satellites and low earth orbit (LEO) satellites. M/M/1 queues are then used to model both the control plane and the data plane; finally, numerical analysis is employed to analyse the effect of different parameters on the file sojourn time.

Similar to our work is the performance modelling and analysis of SDN under bursty multimedia traffic [7]. The authors use Markov modulated Poisson process (MMPP) to investigate the performance of SDN in the presence of bursty and correlated arrivals. However, they assume that the packet departure process from the MMPP queue is MMPP to allow for tractable analytical model. Unlike them, we consider network traffic in software-defined industrial control networks. We use the results from [13] to better model the packet arrival process at the control plane. Also, we observe how adversarial flow may impact network traffic in software-defined industrial control networks.

In contrast to all the works presented above, we exploit queueing network model to analyse the behaviour of network traffic in software-defined industrial control networks. We consider the strong periodic patterns of the network traffic in industrial control networks and approximate the arrival process using MMPP. Also, we use results from [13] to obtain a realistic characterization of the interaction between the data plane and the control plane. We then apply the analytical model to study the effect of adversarial flow in software-defined industrial control networks.

III. System Modelling

In this section, we present a discussion on the queueing network model used in this study to model the behaviour of network traffic in software-defined industrial control networks. Also, we describe the performance measures that may be used for investigating how different parameter choices would affect the behaviour of network traffic. These performance measures are deployed in the application of the model to study adversarial flow in software-defined industrial control networks as presented in section IV.

A. The Software-Defined Industrial Control Network

We consider the software-defined industrial control network proposed in [14]. The data plane consists of Raspberry Pis (RPis), sensors, and actuators. RPis are used for receiving packets from sensors and instructing the relevant actuators to take actions based on the respective flow retrieved from the flow table or corresponding controller [14]. An existing flow is retrieved from the flow table while a new flow is sent to the controller via a Packet-In message. On getting the Packet-In message, the controller instructs the RPi on how to forward the flow via a Packet-Out/Flow-MOD message. The interaction between the data plane and the control plane occurs through the southbound interface of the control plane.

Moreover, many recent measurement studies have shown that network traffic in industrial control networks exhibits strong periodic patterns [15]–[17]. These are usually bursty and cannot be properly modelled using Possion arrival process. Also, a peculiar characteristic of industrial control network traffic is that there are a number of components interacting with each other, which implies that the bursty and correlated nature of the traffic can be captured by superposing multiple bursty sources. And considering that the MMPP has shown to be an effective tool for capturing time-varying arrival, it would be appropriate for modelling the bursty and correlated arrival patterns of network traffic in software-defined industrial control networks [7], [18].

The MMPP is a doubly stochastic Poisson process with time-varying arrival and can be obtained by varying the arrival rate of a Poisson process according to an m-state irreducible continuous Markov chain that is independent of the arrival process [18]. Several studies have exploited a two-state MMPP to evaluate the performance of a network [7], [19]. Whilst it is possible to model the network traffic using a single distribution, we use multiple similar distributions to capture the superposition of multiple bursty sources. The superposition of MMPPs has shown to generate MMPP and if the process to be superposed is identical, the complexity is greatly reduced [18]. This allows us to derive the analytical model; as the infinitesimal generator matrix $\mathbf{Q}$ of the Markov chain and arrival matrix $\mathbf{\Lambda}$ can be parametrized as follows:

$$\mathbf{Q} = \begin{bmatrix} -\varphi_1 & \varphi_1 \\ \varphi_2 & -\varphi_2 \end{bmatrix} and\, \mathbf{\Lambda} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$

where φ_1 is the transition rate from state 1 to 2, and φ_2 is the rate out of state 2 to 1. Also, λ_1 and λ_2 represent the traffic rates when the Markov chain is in states 1 and 2, respectively. We can then obtain the average rate of arrival λ_s of network traffic in industrial control networks as:

$$\lambda_s = \frac{\varphi_1\lambda_2 + \varphi_2\lambda_1}{\varphi_1 + \varphi_2} \tag{1}$$

B. Queueing Network Model for Software-defined Industrial Control Networks

We employ queueing network model here, to study the behaviour of network traffic in software-defined industrial control networks. The arrival process of packet at the data plane of SDN deployed in industrial control networks consists mainly of the network traffic modelled using MMPP. This makes up the existing flow in the software-defined industrial control networks. However, there could be a new flow that needs the attention of the control plane. This type of flow does not have an entry in the flow table of the RPi (switch) and it would have to be processed by the control plane.

For the control plane, we consider the case of multiple controllers. Multiple controllers are usually deployed to improve the resilience of SDN architecture. This is to ensure that the network continues to function even if one of the controllers fails. The handover between the controllers in the case of failure is initiated by the controllers themselves, which facilitates fast recovery from failure and also controllers load balancing [20]. The mechanism used by the controllers to manage the data plane devices among themselves is beyond the scope of this paper. Also, according to the OpenFlow specification [20], the data plane devices must connect to all the controllers they are configured with and would try to maintain connectivity with all of them concurrently. Hence, the packets departing from the data plane are simultaneously fed into the controllers in the control plane of the SDN architecture. A more complex architecture for the controllers have been considered in [21], [22] but for the purpose of the discussion in this paper, we are concentrating on this simple model.

The arrival process of packets at the controllers in the control plane is dependent on the departure process of packets from the data plane. Tian [13] have shown that the output process of the Markovial arrival process is not a Markovial process, but rather, it becomes a Poisson process. Thus, like most existing works on the analysis of SDN using queueing network models, we employ a Poisson arrival process with exponential service time distribution (M/M/1) to model the behaviour of the controllers in the control plane [8], [10]–[12], [23]–[25]. Also, to obtain a more realistic characterization of the controllers' behaviour, we use a finite capacity queue of M/M/1 (M/M/1/K) for the controllers at the control plane.

The queueing network model used for this study is shown in Figure 1, such that $n < m$.

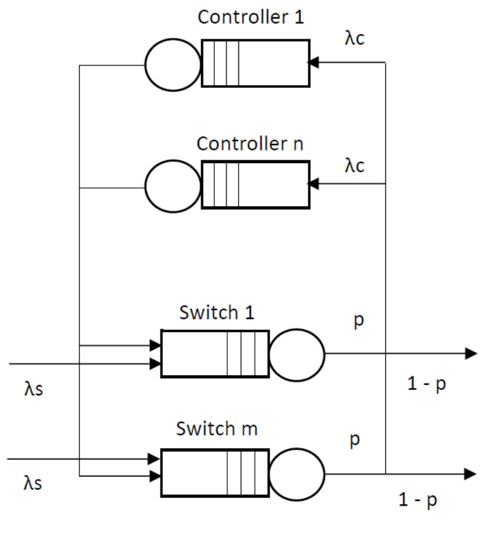

Fig. 1. Queueing Network Model for Software-defined Industrial Control Networks

From the queueing network model of the software-defined industrial control network shown in Figure 1, an arriving packet at the data plane is checked to see if it belongs either an existing flow or if it is a new flow. The probability that the arriving packet is a new flow is given as p. If the arriving packet is a new flow, the packet is sent to the controllers in order to obtain information on how the packet should be forwarded. In the case where the arriving packet belongs to an existing flow, the packet is forwarded with a probability of $1 - p$ without consulting the controllers.

C. Performance Measures

Let T_s be the forwarding time of a packet through the data plane, and T_c be the forwarding time of a packet through the control plane, then we can obtain the total forwarding time of a packet through the queueing network model, T by dividing the packet forwarding into two cases: forwarding without the intervention of control plane and forwarding with the involvement of the control plane. The latter case consists of the sojourn time of the packet (the expected time spent by the packet) in the data plane T_s, and the sojourn time of the packet in the control plane T_c, while the former is just the sojourn time of the packet in the data plane. Thus, we have:

$$T = \begin{cases} T_s & \text{with probability}\, 1-p \\ T_s + T_c & \text{with probability}\, p \end{cases}$$

Also, we can obtain the total forwarding time of packet through the queueing network model as:

$$\begin{aligned} T &= (1-p)T_s + p(T_s + T_c) \\ &= T_s + pT_c \end{aligned} \tag{2}$$

The value of T_s, which is the average sojourn time in the data plane can be calculated using the method proposed by Fischer and Meier-Hellstern [18]:

$$\begin{aligned} T_s = \frac{1}{\rho}(\frac{1}{2(1-\rho)}(2\rho + \lambda_s m_1 - 2m_1((1-\rho)g + m_2\pi\Lambda) \\ (Q + e\pi)^{-1}\lambda_e) - \frac{1}{2}\lambda_s m_2) \end{aligned} \tag{3}$$

where ρ is the traffic intensity at the data plane, given by $\rho = \lambda_s/\mu_s$; m_1 and m_2 are the mean and the second moment of the service time, given by $m_1 = 1/\mu_s$ and $m_2 = 2/{\mu_s}^2$, respectively; $ge = 1$ and $g\mu_s = (1-\rho)^{-1}$ [26]; $\pi = (\varphi_1, \varphi_2)/(\varphi_1 + \varphi_2)$; and $\lambda_e = (\lambda_1, \lambda_2)$.

In addition, the blocking probability P_{b_s} that an arriving packet finds the buffer full can be obtained by calculating the probability P'_n, $0 \leq n \leq K_s$ that there are n packets in a MMPP/M/1/K queue [27] and it is given by [28]:

$$P'_n = \left(\sum_{n=0}^{K_s} \mathbf{P}_n \times \mathbf{\Lambda} \times \mathbf{e}\right)^{-1} \mathbf{P}_n \times \mathbf{\Lambda} \times \mathbf{e} \tag{4}$$

Thus, the blocking probability P_{b_s} can be written as [27]:

$$P_{b_s} = P'_{K_s} \tag{5}$$

For the value of T_c, the packet arriving at the control plane is obtained from the packet departing from the data plane. Unlike the assumption made by Maio et al. [7] that the packet departure process from the MMPP queue is MMPP to allow for a tractable analytical model, we use results from Tian [13] to model the packet arrival process at the control plane. The results show that the departure process of the Markovial arrival process is not Markovial but rather, it becomes a Poisson process [13]. Hence, the packet departure process of the Markovial arrival at the data plane, which is the same as the packet arrival at the control plane λ_c can be obtained using the Laplace transform matching method and it is given as [13]:

$$\lambda_c = \mu_s(\lambda_1\lambda_2 + \lambda_1\varphi_2 + \lambda_2\varphi_1) \tag{6}$$

Given that N_c is the number of packets in the M/M/1/K queue at the control plane, the average number of packets at the control plane E[N_c] can be calculated using the standard formula given as:

$$E[N_c] = \frac{\rho(1 - (K+1)\rho^K + K\rho^{K+1})}{(1-\rho)(1-\rho^{K+1})} \tag{7}$$

where ρ is the traffic intensity at the control plane (λ_c/μ_c) and K is the buffer size .

We can then derive the value of T_c, which is the average sojourn time in the control plane using Little's Law as follows:

$$T_c = \frac{\rho^{K+1}(K\rho - K - 1) + \rho}{\lambda_c(1-\rho)(1-\rho^K)} \tag{8}$$

The blocking probability, which is the probability that an arriving packet finds the system full is given as:

$$P_{b_c} = \rho^K \frac{1-\rho}{1-\rho^{K+1}} \tag{9}$$

IV. Modelling the Adversarial Flow Using the Queueing Network Model

In this section, we investigate the effect of adversarial flow in software-defined industrial control networks using the developed queueing network model. We model the adversarial flow by aggregating the adversarial traffic and the regular traffic. We assume that the attacker has probabilistic knowledge of the regular traffic and using this assumption, we observe how an attacker may vary the arrival rate of the adversarial traffic to increase the average sojourn time of traffic in the network leading to a breach of QoS requirement. The main objective of the attacker is to cause a denial of service (DoS) attack.

A. The Adversarial Flow Model

In order to model the adversarial flow, there are different hierarchies of models that may be considered. These hierarchies of models would be based on how much knowledge the adversary is using to formulate the adversarial flow model. The first level of the model involves the superposition of the adversarial traffic and the regular traffic. In the second level of the model, a stochastic model of the legitimate traffic can be used for the modelling the adversarial flow. Furthermore, the third level of the model involves the use of actual observation of the regular traffic for modelling the adversarial flow.

Although the hierarchy of models describes a spectrum of how much information we allow the attacker to have when formulating the model, we present the first level of the model in this subsection. Along that spectrum, what we then describe is the simplified version of the adversarial flow model to show the utility of our model. We assume that the attacker has some knowledge of the existing flow and is able to create the model of the regular flow. This will allow the attacker to adapt future adversarial flow by looking at the historical flows that have been seen. The attacker does not need to see the actual traffic, but rather the model of the traffic can be employed to adapt the adversarial flow. It is then possible for the attacker to use the knowledge of the model to modify the adversarial flow in such a way that it could increase the average sojourn time of network traffic which may cause a DoS attack.

The first level of the modelling, which is the superposition of the adversarial traffic and the regular traffic is shown in Figure 2. The arrival rate of traffic λ_s to the queueing network model is then the ratio of the arrival rate of regular traffic λ_R as in (1), and that of the adversarial traffic λ_A (which is under the control of the attacker). Already, we have assumed that the adversary knows the model of the regular traffic, which implies that the adversary is aware that the regular traffic follows MMPP arrival process. We can then evaluate how an adversary can adapt the arrival rate of the adversarial traffic by looking at the regular traffic such that the average sojourn time of network traffic is increased in the next subsection.

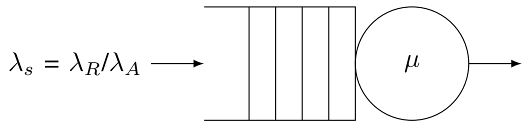

Fig. 2. Superimposing Regular Flow and Adversarial Flow

B. Evaluation of the Model

The purpose of this evaluation is to study the effectiveness of adding an adversarial flow in breaching QoS parameters. To do this, we parametrize the queueing network model by first assuming that the traffic rates of the regular traffic are same as the traffic rates of the adversarial traffic. This is to allow us to examine how an intelligent attacker may manipulate the arrival rate of the adversarial flow against the regular traffic in such a way that the probability of the network traffic breaching QoS requirement is high. An explicit estimate of the combined traffic can then be obtained such that we take the regular traffic and express the adversarial traffic as an addition with some probability of extra traffic.

The remaining variables of the queueing network model are parametrized as follows. We assume that the service rate of the data plane is same as the control plane, the infinitesimal generator of the MMPP arrivals is set as $\varphi_1 = 0.06$ and $\varphi_2 = 0.03$, and we use the same buffer sizes for both the data plane and the control plane. Also, we use the result from [8] that show the probability that an arriving packet is a new flow, $p = 0.04$ as the worst case estimate. By varying the arrival rate of the adversarial traffic against the regular traffic, we observe the condition under which the adversarial traffic would increase the average sojourn time of traffic in the network.

In the scenario that we described, an attacker with some knowledge of the regular traffic can increase the arrival rate of the adversarial traffic in order to increase the average sojourn time of traffic in the network. The attacker may continue to increase the arrival rate of the adversarial traffic in a stealthy way to avoid detection until it causes the network to breach the QoS requirement. This implies that given a regular traffic with parameters in the preceding paragraphs, if the attacker is able to create adversarial flow with parameters described in figure 3 then there would be a breach of QoS requirement.

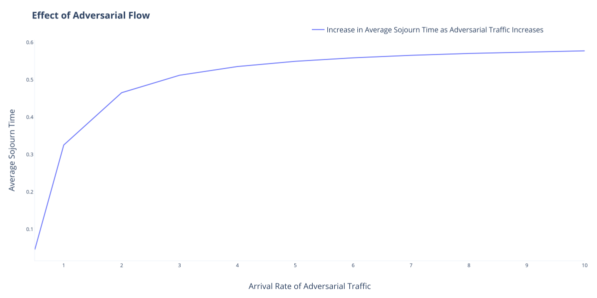

Fig. 3. Average Sojourn Time vs Arrival Rate of Adversarial Traffic

We can infer from our study that the probability of breaching the QoS requirements of network traffic in software-defined industrial control networks would increase with a knowledgeable attacker. An attacker with some knowledge of the regular traffic is able to manipulate the arrival rate of the adversarial flow according to the nature of the regular traffic. By so doing, the attacker can launch an attack at the right time so as to trigger an adverse effect or event. Therefore, the arrival rate of the adversarial flow would have to be at least higher than the arrival rate of the regular traffic, so that the probability of this breaching the boundary is high.

Considering the attacker's objective, this can effectively create an optimization problem with a number of constraints. The optimization problem is not a static optimization, but has constraints that can be expressed in the form of functions. This is because the attacker is modulating based on the existing bursty traffic. In addition, the attacker does not just create a DoS attack, but is constrained by the number of adversarial flows that can be used to maximise disruption and stealthiness at the same time. We are formulating this as standard optimization problem, but we are not dealing with the optimization part here, because it is a well-known and well-solved problem.

We can then say that given the constraints faced by the attacker which can be expressed in the form of functions, these would be the criteria for getting the sojourn time over the threshold that could result in a DoS attack. These constraints have to be solvable by the attacker to achieve the desired goal (i.e., DoS attack). Under these constraints, the attacker can adapt some parameters like the arrival rate to increase the average sojourn time of network traffic. Although there are more clever ways of modulating the attacker traffic than the superposition with some knowledge of the regular traffic, we use the idea here to show the capability of an intelligent attacker.

V. Conclusions and Future Work

Research in the applicability of SDN for industrial control networks is still on-going and consideration for how adversarial flow may impact network traffic in software-defined industrial control networks is very pertinent. In this paper, we have proposed an analytical model using a queueing network model to study the behaviour of network traffic in software-defined industrial control networks. We have observed that the network traffic in industrial control networks exhibit strong periodic patterns and we approximated the arrival process at the data plane using MMPP. Also, we derived the performance measures for the analytical model and then applied the model to study the effect of adversarial flow in software-defined industrial control networks.

Our study indicates that an adversary with some knowledge of the regular traffic is able to increase the arrival rate of the adversarial traffic in such a way that it is higher than that of the regular traffic to increase the average sojourn time of traffic in the network. It is also possible for the attacker to continue to increase the arrival rate of the adversarial traffic in a covert manner. This ensures that the intrusion detection system will not notice the additional malicious traffic until the QoS requirement has been breached.

There are several directions for future research within the analysis of software-defined industrial control networks using a queueing network model. One direction is to layer a particular application domain on top of the queueing network model. For example, it is possible to consider the condition under which the adversarial flow in SDN deployed for IEC 61850 substation could breach the performance bounds as specified by the standard. Another direction is to investigate refinements of the probability density function of the regular traffic and to derive its performance metrics. This could then be referenced when the regular traffic is combined with adversarial traffic to study the effect of adversarial flow.

References

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

[2] L. O. Nweke and S. Wolthusen, "Resilience analysis of Software-Defined networks using queueing networks," in *2020 International Conference on Computing, Networking and Communications (ICNC): Communications and Information Security Symposium (ICNC'20 CIS)*, Big Island, USA, Feb. 2020.

[3] 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.

[4] A. Chowdhary and D. Huang, "Sdn based network function parallelism in cloud," in *Proc. Networking and Communications (ICNC) 2019 Int. Conf. Computing*, Feb. 2019, pp. 486–490.

[5] 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.

[6] T. Li, H. Zhou, H. Luo, W. Quan, and S. Yu, "Modeling software defined satellite networks using queueing theory," in *Proc. IEEE Int. Conf. Communications (ICC)*, May 2017, pp. 1–6.

[7] W. Miao, G. Min, Y. Wu, H. Wang, and J. Hu, "Performance modelling and analysis of software-defined networking under bursty multimedia traffic," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 12, no. 5s, pp. 77:1–77:19, Sep. 2016. [Online]. Available: http://doi.acm.org/10.1145/2983637

[8] M. Jarschel, S. Oechsner, D. Schlosser, R. Pries, S. Goll, and P. Tran-Gia, "Modeling and performance evaluation of an openflow architecture," in *Proc. 23rd Int. Teletraffic Congress (ITC)*, Sep. 2011, pp. 1–7. [Online]. Available: https://ieeexplore.ieee.org/document/6038457

[9] 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.

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

[11] 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.

[12] 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.

[13] W. Tian, "Analysis and efficient provisioning of access networks with correlated and bursty arrivals," *Int. J. Communication Systems*, vol. 27, no. 4, pp. 551–570, 2014.

[14] K. Ahmed, J. O. Blech, M. A. Gregory, and H. W. Schmidt, "Software defined networks in industrial automation," *Journal of Sensor and Actuator Networks*, vol. 7, no. 3, p. 33, 2018.

[15] R. R. R. Barbosa, R. Sadre, and A. Pras, "Exploiting traffic periodicity in industrial control networks," *IJCIP*, vol. 13, pp. 52–62, 2016.

[16] Jiexin Zhang, Shaoduo Gan, X. Liu, and P. Zhu, "Intrusion detection in scada systems by traffic periodicity and telemetry analysis," in *Proc. IEEE Symp. Computers and Communication (ISCC)*, Jun. 2016, pp. 318–325.

[17] R. R. R. Barbosa, R. Sadre, and A. Pras, "Towards periodicity based anomaly detection in scada networks," in *Proc. IEEE 17th Int. Conf. Emerging Technologies Factory Automation (ETFA 2012)*, Sep. 2012, pp. 1–4.

[18] W. Fischer and K. S. Meier-Hellstern, "The markov-modulated poisson process (MMPP) cookbook," *Perform. Eval.*, vol. 18, no. 2, pp. 149–171, 1993.

[19] B. L. Mark and Y. Ephraim, "Explicit causal recursive estimators for continuous-time bivariate Markov chains," *IEEE Transactions on Signal Processing*, vol. 62, no. 10, pp. 2709–2718, May 2014.

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

[21] G. Wang, J. Li, and X. Chang, "Modeling and performance analysis of the multiple controllers' approach in software defined networking," in *Proc. IEEE 23rd Int. Symp. Quality of Service (IWQoS)*, Jun. 2015, pp. 73–74.

[22] Z. Bozakov and A. Rizk, "Taming sdn controllers in heterogeneous hardware environments," in *Proc. Second European Workshop Software Defined Networks*, Oct. 2013, pp. 50–55.

[23] A. Chilwan, K. Mahmood, O. N. sterb, and M. Jarschel, "On modeling controller-switch interaction in openflow based sdns," *International Journal of Computer Networks & Communications*, vol. 6, pp. 137–150, 2014.

[24] K. Mahmood, M. Jarschel, O. Østerbø, and A. Chilwan, "Modelling of openflow-based software-defined networks: the multiple node case," *IET Networks*, vol. 4, pp. 278–284, 2015.

[25] 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.

[26] D. M. Lucantoni, "New results on the single server queue with a batch markovian arrival process," *Stochastic Models*, vol. 7, pp. 1–46, 1991.

[27] Y. Wu, G. Min, and L. T. Yang, "Performance analysis of hybrid wireless networks under bursty and correlated traffic," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 1, pp. 449–454, Jan. 2013.

[28] K. S. Meier-Hellstern, "The analysis of a queue arising in overflow models," *IEEE Transactions on Communications*, vol. 37, no. 4, pp. 367–372, Apr. 1989.