Modelling and Analysis of SDN and NFV Enabled Communications Networks

Joseph Billingsley, Wang Miao and Geyong Min
Department of Computer Science
University of Exeter, UK
Email: {jb931, wang.miao, g.min}@exeter.ac.uk

***Abstract*—Developed a novel and practical analytical model for next generation, SDN and NFV enabled communications networks. Next generation communications networks are using these technologies but fast and accurate models do not yet exist. We model each network component in a layer as identical M/M/1 queues and calculate the expected latency considering possible efficient paths. We extend this work to consider services with many network functions in sequence, several services in the same network and VNFs that filter messages. After validating the models accuracy through extensive simulation experiments we, the model is used to investigate the impact of different configurations of services and VNFs. The model revealed disproportionate load on certain layers in the network as well as significant impact of naieve placement of VNFs. It also serves as a first step towards developing an efficient analytical model for optimisation of Virtual Network Function placement.**

I. Introduction

Next generation communications networks are faced with scaling to support greater expectations and usage of existing services whilst simultanously supporting demanding new use cases such as the internet of things, machine to machine, fog computing and virtual and augmented reality. Two technologies that have emerged as part of the solution to these problems are Network Function Virtualisation (NFV) and Software Defined Networking (SDN).

Fast analytical models allow for evaluation and hence optimisation of large scale networks. Existing research into analytical models of these technologies has focussed on them in isolation. Longo et al. [1] constructed a model of the reliability of a two layer hierarchical SDN controller. Similariliy, Azodolmolky et al. [2] also examine the two layer SDN controller and use network calculus to determine the worst case delay incurred by visiting it and the minimum buffer size required to prevent packet loss given the highest load. Wang et al. [3] produce a more accurate SDN model by considering the bursty and correlated arrivals of packets and a high and low priority queue at an SDN enabled switch. Unlike other research they only model the case of one switch and one SDN controller.

With regards to NFV modeling Prados-Garzon et al. [4] produced a detailed model of a single VNF which is composed of several VNF components and calculate the average response time of the VNF. Gebert et al. [5] analysed a single VNF in even more detail by modelling the packet processing process of a Linux x86 system. Their model estimates latency and packet loss within the 95% confidence interval of lab testing of a real world server. To the best of our knowledge, only Fahmin et al. [6] considered both NFV and SDN. They analysed the performance of two methods of combining SDN and NFV in the network. However they consider a simplfied network with only a single switch and VNF.

In this work we strike a middle ground between existing approaches by considering all relevant aspects of the network at an appropriate level of detail. Whilst efficient analytical models for Fat Tree networks have long been known [7], and SDN and NFV modelling widely researched, to the best of our knowledge we are the first researchers to produce an integrated model that can consider the impact of all of these components. The key contributions of this paper are:

- An efficient analytical model for a next generation NFV enabled network containing a centralised SDN controller using M/M/1 queues.
- Generally applicable extensions for different length service chains and multiple services in the same network.
- A generally applicable extension for VNFs that reduce packet rate as part of a service chain.

The rest of this paper is organised as follows. Section II reviews the preliminaries that are useful for understanding the subsequent sections. In Section III we derive the analytical model. Section IV validates this model with extensive simulation experiments. Finally Section VI concludes the paper, explores the implications of the models and examines future research directions.

II. Preliminaries

This section first introduces the concepts of Network Function Virtualisation and Software Defined Networking and presents the implications of these changes to the communication network architecture.

A. Network Function Virtualisation

In telecommunications networks, services are composed out of several network functions such as load balancers, firewalls and intrusion detection systems. Traditionally these network functions would be provided by specially engineered pieces of network hardware. In a Network Function Virtualisation (NFV) enabled network, these network functions are virtualised and can be run on industry standard servers reducing both capital and operating expenditure [].

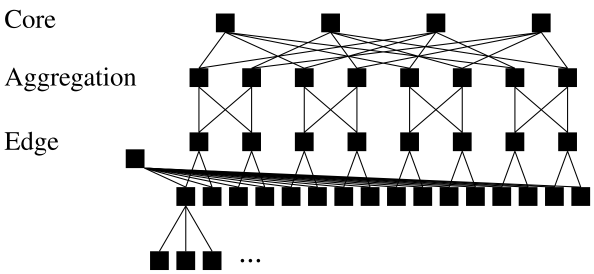

Fig. 1. The three layer topology modelled in this network

Services can be defined using Directed Acyclic Graphs (DAG) which encapsulate dependencies between network functions such as in Fig. **??**. Some network functions can affect the data rate. Network functions such as firewalls or can reduce the rate by filtering out packets. Other network functions such as video decoders can increase the data rate by creating more packets for subsequent network functions to process. The task of constructing a 'concrete' service from an abstract DAG is called service composition and has received much interest in optimisation [**?**]. In Section III we describe how our model can be used to evaluate the performance of a given concrete service.

B. Software Defined Networking

Software Defined Networking (SDN) allow for dynamic routing of packets throughout the network [] and configuration of VNFs. A logically centralised SDN controller exists in the network which maintains a global view of the network. When a packet arrives at a SDN enabled device a flow lookup is performed to determine how the packet should be routed. If an appropriate flow does not exist in the router the device sends a request to the controller requesting an update to the flow table. Depending on the implementation, portions or the entirety of the flow table may be populated in advance.

There may exist one or more SDN controllers. In a centralised SDN solution, a single control entity has a global view of the network. In contrast hierachical and distributed approaches use several control entities and either decompose or share the network information respectively.

C. The Network Architecture

In this work we extend the Fat Tree network [8] frequently found in datacentres [] to be SDN and NFV capable. Fig. II-C illustrates this network. Generally the network is laid out as follows:

We define k as the number of ports for physical switches and k_{vm} the number of ports for the virtual switches. There are $(k/2)^2$ core switches. Each core switch connects to one switch in each of the k pods. There are k pods each containing two layers (aggregation and edge) of $k/2$ switches. Each switch in the edge layer is connected to each of the $k/2$ aggregation switches. Each of the remaining switches in the edge layer is connected to $k/2$ servers. Each server has a virtual switch with k_{vm} ports connecting to a VNF.

For each pod, each aggregation switch connects to the core switch with the same index as it and all subsequent switches $k/2$ steps away. As an example, if $k = 4$ the first aggregation switch in a pod connects to core switch 1 and 3, the second aggregation switch connects to 2 and 4 with this pattern repeating for each pod. This topology results in $n = k^3/4{*}k_{vm}$ VNFs.

For this research we constructed a network where each server is connected to a single centralised SDN controller. This topology is used in practice, most notably resembling VMwares NSX architecture []. Alternatively all elements in the network could be connected to the SDN [] or a distributed SDN. This exploration is left to future research.

III. Analytical Model

A. Assumptions

The analytical model is based on the following assumptions which are commonly accepted in the literature []. Assumption 3 is based on the high speed of data centre interconnection networks and the short distances between components.

1) At each VNF, messages are generated according to an independant Poisson process with a mean rate of λ messages a cycle. Furthermore, message destinations are uniformly distributed across the VNFs.
2) At each network component messages are serviced according to an independant Poisson process with a mean rate of μ messages a cycle.
3) The time taken for a message to travel between network components in negligible.
4) The SDN controller ensures messages take the shortest path between the two destinations and that messages are evenly distributed over the switches in the network.
5) Queues at each network component have infinite capacity.
6) Messages sent from a VNF to VNFs on other servers, may need to visit the SDN controller with probability p_{sdn_user}. The same does not apply to messages arriving at a server as the route to the destination will be known.

The mean message latency of the network can be calculated as the sum of the waiting time at each network component a message visits. As a result of the network architecture, messages will take shorter paths to visit VNFs under the same server, edge switch or pod. Consequently, the traffic arriving at servers at each layer in the network will also vary. The mean message latency for a single hop service chain can hence be calculated as:

$$\begin{aligned} Latency_{base} &= ((w_{vm} + w_{sdn} + w_{srv}) \cdot p_{srv} \\ &+ (w_{vm} + w_{sdn} + 2w_{srv} + w_{edge}) \cdot p_{edge} \\ &+ (w_{vm} + w_{sdn} + 2w_{srv} + 2w_{edge} + w_{agg}) \cdot p_{agg} \\ &+ (w_{vm} + w_{sdn} + 2w_{srv} + 2w_{edge} \\ &\qquad + 2w_{agg} + 2w_{core}) \cdot p_{core}) \end{aligned} \tag{1}$$

Where w_{vm}, w_{sdn}, w_{srv},w_{edge},w_{agg}, w_{core} represent the average time spent in a VM, the SDN controller, a virtual, edge, aggregate and core switch respectively. Similarily p_{srv}, p_{edge}, p_{agg} and p_{core} represent the probability that the highest level a message which reach is a virtual, edge, aggregate or core switch respectively. The remainder of this section will deduce these values for arbitary settings of k and $k_v m$.

B. Probability of Highest Level

As messages will always take the shortest path, a message will visit only a virtual switch if the destination VM is in under the same virtual switch as the source. Given $n = k^3/4 \cdot k_{vm}$ total VMs:

$$p_{vm} = \frac{k_{vm} - 1}{n - 1} \tag{2}$$

Similarily the probability of a message visiting at highest an edge switch is the proportion of destinations that are under the edge switch, excluding those destinations that could be visited via a shorter route.

$$p_{edge} = \frac{(k/2) \cdot k_{vm} - k_{vm}}{n - 1} \tag{3}$$

This same techinque can be used to find the remaining probabilities:

$$p_{agg} = \frac{(k/2)^2 \cdot k_{vm} - (k/2) \cdot k_v m}{n - 1} \tag{4}$$

$$p_{core} = \frac{n - (k/2)^2 \cdot k_{vm}}{n - 1} \tag{5}$$

Finally we can calculate the probability p_{sdn} by excluding all messages that will not leave the server:

$$p_{sdn} = p_{sdn\_user} \cdot (1 - p_{vm}) \tag{6}$$

C. Calculation of Mean Waiting Time

To determine the mean waiting time at each network component, we model each component as a M/M/1 queue where the mean waiting time is calculated as []:

$$MM1(\mu, \lambda_{nc}) = \frac{1}{\mu - \lambda_{nc}} \tag{7}$$

Where nc is the network component under question. Despite messages being distributed evenly across switches in each layer by the SDN controller, the arrival rate, λ_{nc}, will be different for each layer as not all messages will visit all layers.

As destinations are evenly distributed over the VNFs, all VNFs will send an equal proportion of messages to all others:

$$\begin{aligned} \lambda_{vm} &= \frac{n-1}{n-1} \cdot \lambda \\ &= \lambda \end{aligned} \tag{8}$$

A portion of the messages being produced by every VNF will require the SDN controller to be consulted:

$$\lambda_{sdn} = n \cdot p_{sdn} \cdot \lambda \tag{9}$$

Servers can receive messages from three sources: generated from VNFs it is running, received from other VNFs and reflected messages that it had sent to the SDN. Following the same format:

$$\begin{aligned} \lambda_{srv} = & \; k_{vm} \cdot \lambda \\ & + (n - k_{vm}) \cdot \frac{k_{vm}}{n-1} \cdot \lambda \\ & + k_{vm} \cdot \lambda \cdot p_{sdn} \end{aligned} \tag{10}$$

Where line two can be understood as the number of VNFs that are not hosted by the server, sending an equal proportion of their messages to each of the VNFs hosted by the server.

Note that the SDN controller does not affect switches as the portion of messages that are sent to the controller are not sent to higher switches till later cycles and that this absence is filled by messages returned from the SDN controller from earlier cycles.

The arrival rate for the edge switches can then be deduced in the same way. Following the same order:

$$\begin{aligned} \lambda_{edge} = & \; ((k/2) \cdot k_{vm}) \cdot (n - k_{vm}) \cdot \frac{1}{n-1} \cdot \lambda \\ & + (n - ((k/2) \cdot k_{vm}) \cdot \frac{(k/2) \cdot k_{vm}}{n-1} \cdot \lambda \end{aligned} \tag{11}$$

The same principle can also be followed for the aggregate switches only now all traffic will be split between each aggregate switch in the pod. In the same order:

$$\begin{aligned} \lambda_{agg} = & \; (k/2)^2 \cdot k_{vm} \cdot \frac{n - (k_{vm} \cdot (k/2))}{n-1} \cdot \frac{\lambda}{k/2} \\ & + n - ((k/2)^2 k_{vm}) \cdot \frac{(k/2)^2 \cdot k_{vm}}{n-1} \cdot \frac{\lambda}{k/2} \end{aligned} \tag{12}$$

Finally, as all VNFs are descendants of all core switches the arrival rate at each core can be calculated as the portion of messages arriving at a core switch, split evenly between each of the core switches.

$$\lambda_{core} = p_{core} \cdot n \cdot \frac{1}{(k/2)^2} \cdot \lambda \tag{13}$$

By substituting 8 to 13 for the arrival rates at each network component into 7 for the average waiting time we can calculate the latency incurred when visiting a network component, w_{nc}, for all network components except for the SDN controller.

When a message is sent to the controller it will incurr added latency both waiting at the controller and waiting at the server again when it returns. The expectation of the waiting time incurred by the SDN controller is:

$$w_{sdn} = (MM1(\mu, \lambda_{sdn}) + w_{srv}) * p_{sdn} \tag{14}$$

D. Mean Latency of Long Services and Many Services

As discussed in the preliminary section telecommunications services are typically composed from several network functions that pass messages from one to the other. As a result messages persist in the network for a longer period of time.

Consider a situation where each VNF send a message to an adjacent network function every cycle, under the same server or otherwise, so that all network functions will receive a message each cycle. Consider also that we have a service chain of three network function so that messages will be required to make two hops.

After the first cycle all messages will have sent and received one message. After the second cycle all messages will have sent two messages, forwarding the one received in the previous step and a new message from this cycle, and also received two messages, a message with no hops remaining and one with one hop remaining. At the third cycle one message will be destroyed having completed, leaving one message to be forwarded and one new message created for each VNF. The net result is that on average each VNF is producing 2 messages per cycle.

We can extend this intuition to arbitary length chains. Applying this back to the original analytical model we get:

$$\lambda = \lambda_{new} \cdot (len(chain_i) - 1) \tag{15}$$

where λ_{new} is the average number of new messages that are generated each cycle, len is the number of network functions that compose a given service chain and $chain_i$ is the service being modelled.

We can further extend this to several services, each of which may have different numbers of network functions. If a given message has probability $p(chain_i)$ of belonging to a particular service the average number of hops that a message persists for and hence the impact on the production rate can be calculated as:

$$\lambda = \lambda_{new} \sum_{i=1}^{ns} p(chain_i) \cdot (len(chain_i) - 1) \tag{16}$$

where ns is the number of different services and $\sum_{i=1}^{ns} p(chain_i) = 1$.

Finally we must also consider that messages that require several hops must also traverse the network several times. We can extend equation (1) by multiplying it by the average number of hops in the network considering each service:

$$Latency = Latency_{base} \cdot \sum_{i=1}^{ns} p(chain_i) \cdot (len(chain_i) - 1) \tag{17}$$

E. Mean Latency with Filtering VNFs

We make one final extension for the case where one or more network functions in a service may not forward all of the messages that they receive. As a result subsequent network functions in a service chain have lower production rates.

We can use the same conceptual model as before to solve this. Consider a situation where we have a service chain with 4 VNFs with 2 filtering VNFs that remove 50% of the traffic, as illustrated in **??**. After the first cycle all VNFs will have sent and received 1 message. After the second cycle all VNFs will have sent at least one message, and half of the VNFs will have forwarded another message bringing the average production rate to 1.5. After the first cycle all VNFs will have produced at least 1 message, another half will have forwarded the message received in the previous cycle and a quarter will have forwarded the message sent in the first cycle bringing the average production rate to 1.75.

We can extend this process to several services in the same manner as before by averaging the production rates of each service. The complete algorithm for calculating the production rate considering all aspects is as follows:

for $i = 1 : ns$ **do**
  $multiplier \leftarrow 1$
  for $j = 1 : len(chain_i)$ **do**
    $multiplier \leftarrow multiplier \cdot chain_i(j)$
    $\lambda_i \leftarrow \lambda_i + \lambda_{new} * multiplier$
  end for
end for
$\lambda \leftarrow \sum_{i=1}^{ns} \lambda_i \cdot p(chain_i)$

IV. Validation

The described model has been validated using a discrete event simulator. Each simulation experiment was run until the network reaches it's steady state where further network cycles do not change the collected statistics appreciably.

Numerous validation experiments were performed for several combinations of network sizes, service lengths, number and probability of selection, filtering VNFs and p_{user_sdn}. To remain concise, latency results are presented for selected cases. For all cases, where not otherwise specified, the following parameter settings are used:

- $k = 4$, $k_{vm} = 2$ and $p_{sdn_user} = 0$
- All switches have the same service rate of 40 messages per cycle
- The VNFs have a service rate of 20 messages per cycle
- One service with two VNFs is used
- Services are selected with equal probability

Additionally, when multiple services in the same network are considered (Fig. IV) each service is assigned a length the same as it's index plus one. As a consequence, tests with more services have higher average lengths.

Figures IV to IV depict mean message latency predicted by the model plotted against those provided by a discrete event simulator for a range of parameter settings. Results are only shown where the network is in a steady state, i.e. where the arrival rate for any network element is lower or equal to the service rate at the element. The figures demonstrate that the simulation results closely match that predicted by the model. The tractability and accuracy of the model analytical model make it suitable for analysis of next generation NFV and SDN enabled communications networks.

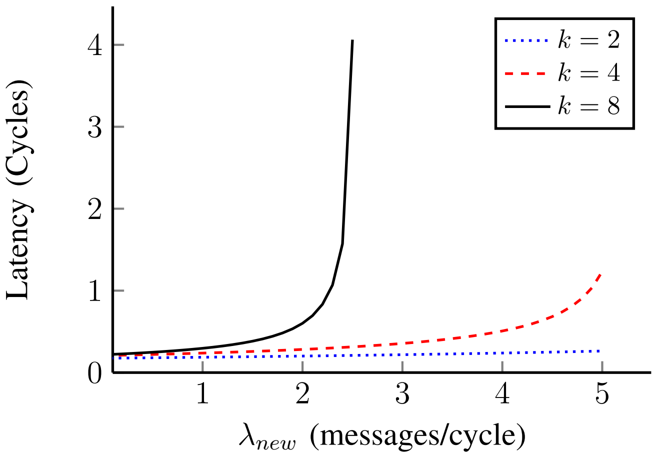

Fig. 2. Latency predicted by the model and simulation for different numbers of ports (k).

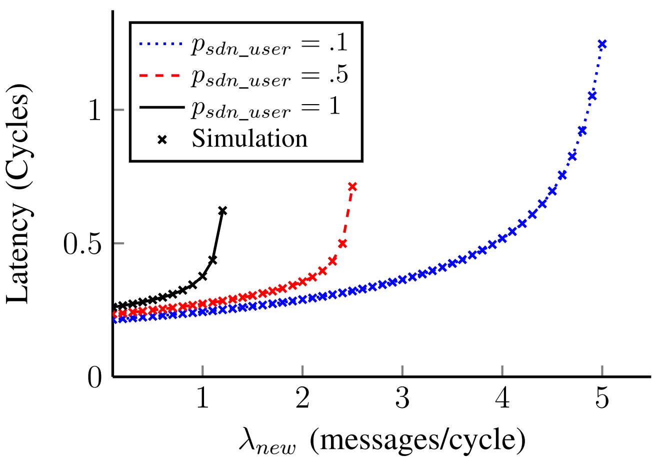

Fig. 3. Latency predicted by the model and simulation with different proportions of messages routing via the SDN controller.

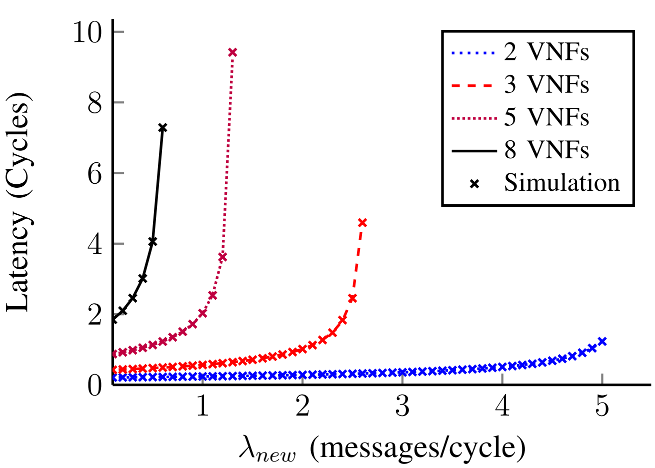

Fig. 4. Latency predicted by the model and simulation for different length service chains.

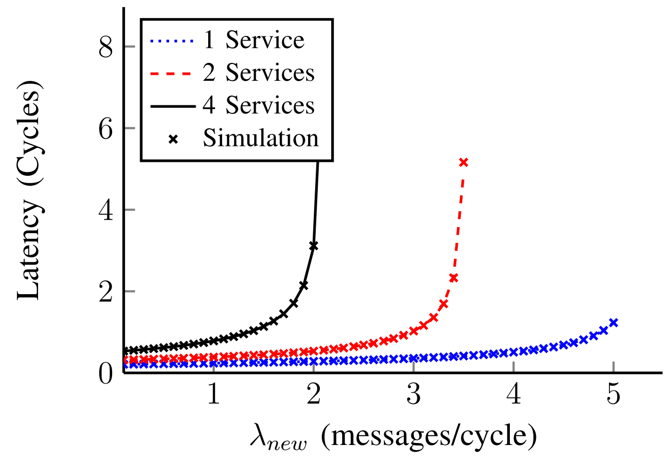

Fig. 5. Latency predicted by the model and simulation for several services of different lengths.

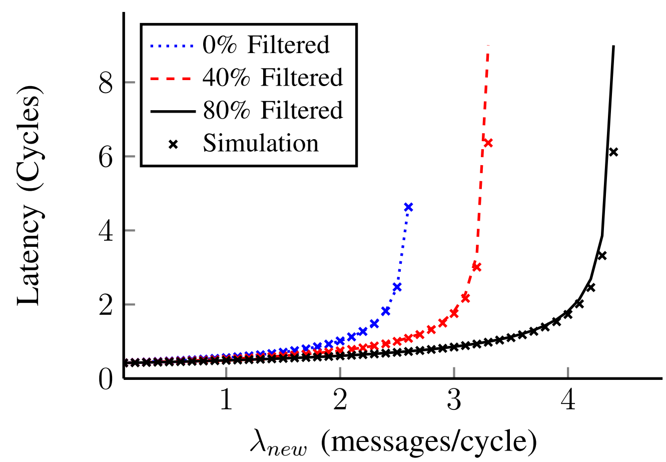

Fig. 6. Latency predicted by the model and simulation for a three long service chain where the second VNF only forwards a portion of the traffic.

V. Performance Analysis

Having validated it's accuracy, the analytical model is now used to investigate the performance of SDN and NFV enabled networks under load.

Analysis of the average arrival rate to each node shows that the edge switches typically receive the most traffic, followed by the aggregate and core switches. Under heavy loads the edge switches will be overwhelmed first resulting in high latency or packet loss.

Fig. IV shows how this issue is aggravated by the addition of NFV features into the network. All switches receive the proportionally the same increase in traffic but as aggregate switches are already the most loaded, they tend to fail whilst other network elements have a non-negligible amount of spare capacity. IV shows that effective filtering network functions can reduce the load on the network. Optimisation of the ordering of filtering VNFs considering dependancies could help reduce the issues caused by persistance of messages throughout the network.

In IV we see how increasing the proportion of traffic that visits the SDN causes the network to become unstable at lower arrival rates. Analysis of the average arrival rate at servers shows that whilst they typically experience lower load than the network switches, they can receive substantially more traffic (from the model, an additional $k_{vm} \cdot \lambda * p_{SDN}$) when a large proportion of the messages are required to visit the SDN controller. If the virtual switch is installed on a server that is able to handle the same rate of traffic as the other switches the edge switch will still be overloaded first. However if the server is substantially less powerful, as in these tests, the servers will be overwhelmed first.

VI. Conclusions and Future Work

In this paper we have presented an efficient analytical model capable of modelling an SDN and NFV enabled datacentre communications network. We also propose extensions that allow the evaluation of impact of services with different lengths and properties running in the network aswell as support for network functions that filter a portion of the traffic.

Whilst this work has evaluated a specific network structure, the same approach for determining the impact of integrating SDN and NFV should be transferable to other network topologies. In particular the extensions made to the model to handle services and reducing network functions have been shown to only impact the arrival rate at servers and so this work should be generalisable to other network topologies.

There remain many useful extensions that can be made to the network. In particular the current model assumes uniform placement of network functions which makes it unsuitable as a model for VNF placement problems. More powerful models that can efficiently consider the impact of concrete and/or fuzzy placement of network functions are required.

Further, M/M/1 queues have been shown to underestimate the latency when faced with more realistic bursty traffic [9]. Similarily the current model can not accurately model NFs that increase the load on the network. This is as these network functions can produce more than one packet at a time, and hence also create bursty traffic patterns. MMPP queues have been used in the literature to model bursty traffic [3], [9] and may be able to solve these problems.

References

[1] F. Longo, S. Distefano, D. Bruneo, and M. Scarpa, "Dependability modeling of software defined networking," *Computer Networks*, vol. 83, pp. 280–296, 2015. [Online]. Available: https://doi.org/10.1016/j.comnet.2015.03.018

[2] S. Azodolmolky, P. Wieder, and R. Yahyapour, "Performance evaluation of a scalable software-defined networking deployment," in *Second European Workshop on Software Defined Networks, EWSDN 2013, Berlin, Germany, October 10-11, 2013*. IEEE Computer Society, 2013, pp. 68–74. [Online]. Available: https://doi.org/10.1109/EWSDN.2013.18

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

[4] J. Prados-Garzon, P. Ameigeiras, J. J. Ramos-Muñoz, P. Andres-Maldonado, and J. M. López-Soler, "Analytical modeling for virtualized network functions," in *2017 IEEE International Conference on Communications Workshops, ICC Workshops 2017, Paris, France, May 21-25, 2017*. IEEE, 2017, pp. 979–985. [Online]. Available: https://doi.org/10.1109/ICCW.2017.7962786

[5] S. Gebert, T. Zinner, S. Lange, C. Schwartz, and P. Tran-Gia, "Performance modeling of softwarized network functions using discrete-time analysis," in *28th International Teletraffic Congress, ITC 2016, Würzburg, Germany, September 12-16, 2016*, T. Hoßfeld, B. L. Mark, S. G. Chan, and A. Timm-Giel, Eds. IEEE, 2016, pp. 234–242. [Online]. Available: https://doi.org/10.1109/ITC-28.2016.140

[6] A. Fahmin, Y. Lai, M. S. Hossain, Y. Lin, and D. Saha, "Performance modeling of SDN with NFV under or aside the controller," in *5th International Conference on Future Internet of Things and Cloud Workshops, FiCloud Workshops 2017, Prague, Czech Republic, August 21-23, 2017*. IEEE Computer Society, 2017, pp. 211–216. [Online]. Available: https://doi.org/10.1109/FiCloudW.2017.76

[7] R. I. Greenberg and L. Guan, "An improved analytical model for wormhole routed networks with application to butterfly fat-trees," in *1997 International Conference on Parallel Processing (ICPP '97), August 11-15, 1997, Bloomington, IL, USA, Proceedings*. IEEE Computer Society, 1997, pp. 44–48. [Online]. Available: https://doi.org/10.1109/ICPP.1997.622554

[8] M. Al-Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity data center network architecture," in *Proceedings of the ACM SIGCOMM 2008 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, Seattle, WA, USA, August 17-22, 2008*, V. Bahl, D. Wetherall, S. Savage, and I. Stoica, Eds. ACM, 2008, pp. 63–74. [Online]. Available: http://doi.acm.org/10.1145/1402958.1402967

[9] Y. Wu, G. Min, K. Li, and B. Javadi, "Modeling and analysis of communication networks in multicluster systems under spatio-temporal bursty traffic," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 5, pp. 902–912, 2012. [Online]. Available: https://doi.org/10.1109/TPDS.2011.198