Modelling and Analysis of SDN and NFV Enabled Datacentre Networks

Joseph Billingsley*, Wang Miao*, Geyong Min*, Nektarios Georgalas† and Ke Li*
*Department of Computer Science, University of Exeter, UK
Email: {jb931, wang.miao, g.min, k.li}@exeter.ac.uk
†Research and Innovation, British Telecom, UK
Email: nektarios.georgalas@bt.com

***Abstract*—As more devices become connected to the internet next generation datacentres must scale to provide more resources than ever before. Simultaneously there is growing pressure to ensure these datacentres become progressively more efficient. Two promising technologies that could help solve these problems are software defined networking (SDN) and network function virtualisation (NFV). SDN allows for flexible management and routing of the network allowing for more agile deployment of services and faster iteration times. NFV allows for services to be scaled to meet demand enabling more efficient resource allocation. Whilst these technologies have received much interest, existing models only consider these components in isolation. To achieve a deeper understanding of the impact of the interactions of SDN, NFV and the underlying interconnection network, this paper proposes a novel analytical model that considers these technologies in unison. The average latency in the network is calculated and the accuracy of the model verified by simulation. The model illuminates disproportionate load on certain layers of the network and serves as a step towards developing an efficient analytical model for optimisation of next generation networks.**

I. Introduction

Next generation communications networks are faced with scaling to support greater expectations and usage of existing services [1] whilst simultaneously supporting demanding new use cases such as the internet of things, smart cities and virtual and augmented reality [2]. Two technologies that have emerged as part of the solution to these problems are Network Function Virtualisation (NFV) and Software Defined Networking (SDN).

In telecommunications networks, services are composed of several network functions such as load balancers, firewalls and intrusion detection systems. Traditionally these network functions would be provided by purpose engineered network hardware. In an NFV enabled network, virtual network functions (VNFs) are run on virtual machines on commodity hardware. With NFV the resources allocated to each VNF can be dynamically committed, allowing for efficient resource allocation whilst minimising the cost and environmental impact of the datacentre.

A service is a collection of several virtual network functions where packets are sent through each of the VNFs in sequence. They can be defined using Directed Acyclic Graphs (DAG) which encapsulate dependencies between network functions as in Figure 1. Different services may be composed of different numbers and types of VNF. Additionally, one or more services may be provided by the datacentre simultaneously.

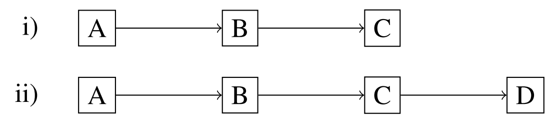

Fig. 1. Two services of different lengths, i) is three long whilst ii) is four long. These and other services may exist in the network at the same time

Software Defined Networking (SDN) allows for dynamic routing of packets throughout the network and configuration of VNFs [3], [4]. A logically centralised but typically physically distributed SDN controller maintains a global view of the network. This simplifies maintenance of the network and allows for complex routing procedures.

SDN and NFV are often considered complementary technologies [5], with the flexible placement enabled by NFV meshing well with the flexible routing allowed by SDN. Despite this existing research in modelling of both technologies has typically considered them in isolation.

Many methods of modelling SDN alone are available in the literature. Longo et al. [6] proposed a model of the reliability of a two layer hierarchical SDN controller. Azodolmolky et al. [7] also examine the two layer SDN controller but use network calculus to determine the worst case delay and the minimum buffer size required to prevent packet loss given the highest load. Wang et al. [8] developed a more realistic SDN model by considering the bursty and correlated arrivals of packets and a high and low priority queue at an SDN enabled switch. While useful, these models had objectives that did not require them to consider the wider network or other technologies.

As with SDN, NFV modelling has similarily had a narrow focus. Prados-Garzon et al. [9] produced a detailed model of a single VNF which is composed of several VNF components and calculated the average response time of the VNF. Another example is Gebert et al. [10] who analysed a single VNF in detail, modelling each queue in the packet processing process of a Linux x86 system. To the best of our knowledge, only Fahmin et al. [11] considered both NFV and SDN, they modelled the performance of two methods of combining SDN and NFV in the network. However they consider a simplified network with only a single switch and VNF.

Unfortunately existing models are not useful for optimisation of the network. In particular they do not typically consider the datacentre interconnection network which is an important factor when considering the placement of services. Additionally existing models do not consider the interaction between NFV and SDN and the effect this can have on latency. Further none of the reviewed literature considered the impact that different length services, or multiple different services may have on the performance of the network. In practice it is unlikely that a datacentre will provide only a single, short service. To this end the main contributions of this paper are:

- An efficient analytical model of a joint NFV and SDN enabled datacentre network is proposed that considers the underlying datacentre interconnection network
- Additional extensions for modelling of multiple services of different lengths are constructed
- The accuracy of the model is verified through comparison with a simulation

The rest of this paper is organised as follows. Section II discusses the details of the network architecture that is modelled in this work. In Section III we derive the analytical model for the network. Section IV validates the accuracy of this model with extensive simulation experiments. Section V explores the implications of the model and Section VI concludes the paper and examines future research directions.

II. Network Architecture

The interconnection network modelled in this work is the fat-tree topology [12]. The fat-tree or folded-Clos topology is currently the most common topology used for datacentre communication networks [13]. The fat-tree topology is formed of many physical switches split into three layers: Core, Aggregation and Edge. Switches at the edge layer are additionally connected to servers. In an NFV enabled network each of these servers contains one or more VNFs which are managed by a virtual switch. The switches can direct packets towards any other network component they are connected to. In an SDN enabled network an SDN controller provides logically centralised management, instructing the switches how to route traffic to ensure it is routed through an efficient path from the source to destination VNF. Figure 2 illustrates an SDN and NFV enabled fat-tree topology.

The fat-tree topology is dependent upon the number of ports at each switch. We define k as the number of ports for each physical switch and k_{vm} the number of ports for each virtual switch. There are $(k/2)^2$ core switches. Each core switch connects to one switch in each of k pods. Each pod contains two layers (aggregation and edge) of $k/2$ switches. Each edge switch is connected to each of the $k/2$ aggregation switches of the pod. Each edge switch is connected to $k/2$ servers. Each server contains a virtual switch connected to k_{vsw} VNFs. This topology results in $n = (k^3/4) \cdot k_{vsw}$ VNFs.

Whilst the SDN controller is logically centralised, in practice a physically centralised SDN architecture cannot scale to support a large datacentre network [7]. A simple extension is the two layer hierarchical structure. In this architecture, a

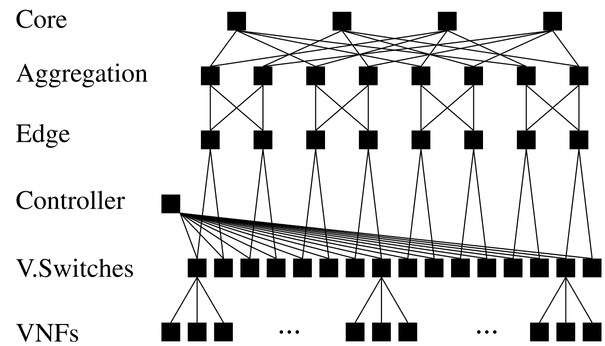

Fig. 2. An example SDN and NFV enabled fat-tree network with 4 ports for each hardware switch and 3 for the virtual switches.

local SDN controller exists at or close to each server and an additional root SDN controller exists as an independent element. Arriving or produced packets are first evaluated at the local controller. If the local controller cannot process the packet it is sent to the root controller and then back to the server once resolved. As a result only a portion of the packets generated in the network will visit the root SDN controller. A comparable real world implementation is VMWare NSX [14] which as of 2016 had been deployed on over 2400 networks [15].

III. Analytical Model

A. Assumptions

In a communications network each switch, server and SDN controller contains a queue where packets are buffered before they are processed. Subsequently the network is modelled in the same way with each network component modelled as an M/M/1 queue. The following assumptions are used in the construction of this model:

1) At each VNF, packets are generated according to an independent Poisson process with a mean rate of λ packets a cycle. Furthermore, packet destinations are uniformly distributed across the VNFs.
2) Each switch, VNF and the root controller services packets according to an independent Poisson process with a mean rate of μ, mu_{vnf} and mu_{sdn} packets a cycle respectively.
3) The time taken for a packet to travel between network components is negligible.
4) The SDN controller ensures packets take one of the shortest paths between source and destination and that packets are evenly distributed over the switches in the network.
5) Queues at each network component have infinite capacity.
6) Packets leaving a server will visit the root SDN controller with probability p_{sdn_root}.

Packets are required to visit every switch on an efficient path between the source VNF and the destination VNF. As a result of the network architecture, packets will take shorter paths to visit VNFs under the same virtual, edge or aggregate switch. Consequently, the traffic arriving at network components at each layer in the network will also vary. The latency for a given path is the sum of the time spent at each network component on the path.

In the simplest case the network contains only one service with two VNFs. In this case a packet will only pass through the network once. The mean latency in this case is dependent only on the latency for each path and the probability of a packet taking that path:

$$\begin{aligned}
&Latency_{base}(\lambda, \mu, \mu_{vnf}, \mu_{sdn}) \\
&\quad = (w_{vnf} + w_{vsw}) \cdot p_{vsw} \\
&\quad + (w_{vnf} + w_{sdn} + 2w_{vsw} + w_{edge}) \cdot p_{edge} \\
&\quad + (w_{vnf} + w_{sdn} + 2w_{vsw} + 2w_{edge} + w_{agg}) \cdot p_{agg} \\
&\quad + (w_{vnf} + w_{sdn} + 2w_{vsw} + 2w_{edge} \\
&\qquad\qquad + 2w_{agg} + w_{core}) \cdot p_{core}
\end{aligned} \tag{1}$$

where w_{vnf}, w_{sdn}, w_{vsw}, w_{edge}, w_{agg} and w_{core} represent the average time spent at a VNF, the root SDN controller and virtual, edge, aggregate and core switches respectively. Similarly p_{vsw}, p_{edge}, p_{agg} and p_{core} represent the probability that the highest level switch a packet visits is a virtual, edge, aggregate or core switch respectively. We now deduce these values for arbitrary settings of k and k_{vsw}.

B. Probability of Highest Level

As packets will always take the shortest path, a packet will not leave the server if the destination and source VNFs share the same virtual switch. Hence the probability of a packet only visiting the virtual switch is the proportion of destinations that are under the virtual switch compared to the total number of destinations:

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

Similarly the probability of a packet 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:

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

This same principle can be used to deduce the probability of visiting an aggregate or core switch:

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

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

Finally, as the SDN controller will only be consulted if the destination VNF is on a different server to the source VNF, the probability of a packet visiting the root controller is the probability of the destination being outside of the server and the local controller being unable to process it:

$$p_{sdn} = (1 - p_{vsw}) \cdot p_{sdn\_root} \tag{6}$$

C. Calculation of Mean Waiting Time

To determine the mean waiting time at each network component, each component is modelled as a M/M/1 queue where the mean waiting time is calculated with [16]:

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

where λ_{nc} is the arrival rate for a particular network component.

As destinations are evenly distributed over the VNFs, each VNF will receive an equal proportion of packets from every other VNF. Hence the arrival rate for each VNF is:

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

Virtual switches can receive packets from three sources: packets generated from VNFs on the server, packets intended for VNFs on the server and returned packets that had been sent to the root SDN controller. Regardless of destination, packets generated by the VNFs on the server must at a minimum pass through the virtual switch. Additionally, an equal portion of the traffic generated by the other VNFs in the network will be intended for each of the VNFs under the virtual switch. Finally all of the traffic sent to the root SDN controller must return to the virtual switch to be processed. Therefore the arrival rate at the virtual switch can be calculated as:

$$\begin{aligned}
\lambda_{vsw} = k_{vsw} \cdot \lambda& \\
+ (n - k_{vsw}) \cdot \frac{k_{vsw}}{n-1} \cdot \lambda& \\
+ k_{vsw} \cdot p_{sdn} \cdot \lambda&
\end{aligned} \tag{9}$$

Packets that are sent to the SDN controller do not need to considered when calculating the arrival rate for higher level switches. While packets that are sent to the root controller are not forwarded to higher switches till later cycles, their absence is filled by packets returned from the SDN controller from earlier cycles.

The arrival rate for the edge switches can be deduced in a similar way. The incoming traffic will be greater for the edge switch as there are more VNFs under it. Whilst more VNFs depend on the edge switch to access other VNFs than for the virtual switch, the packets that are intended for destinations on the same server will not pass through the edge switch. With these factors considered, and excluding packets that can take shorter paths, the arrival rate at the edge switch can be calculated as:

$$\lambda_{edge} = (k/2) \cdot k_{vsw} \cdot \frac{(n - k_{vsw})}{n-1} \cdot \lambda + (n - ((k/2) \cdot k_{vsw}) \cdot \frac{(k/2) \cdot k_{vsw}}{n-1} \cdot \lambda \tag{10}$$

A similar situation occurs for the aggregate switches only now all traffic will be split between each aggregate switch in the pod. Additionally, a smaller proportion of generated packets will visit the aggregate switch. The arrival rate at the aggregate switch is thusly:

$$\lambda_{agg} = \Big((k/2)^2 \cdot k_{vsw} \cdot \frac{(n - k_{vsw} \cdot (k/2))}{n-1} \cdot \lambda + (n - (k/2)^2 \cdot k_{vsw}) \cdot \frac{(k/2)^2 \cdot k_{vsw}}{n-1} \cdot \lambda\Big) \cdot \frac{1}{k/2} \tag{11}$$

Finally, as all VNFs are under each of the core switches the arrival rate at each core is the portion of packets arriving at a core switch, split evenly between each of the core switches:

$$\lambda_{core} = p_{core} \cdot n\lambda \cdot \frac{1}{(k/2)^2} \tag{12}$$

By substituting Equations 8 to 12 for the arrival rates at each network component into Equation 7 we can calculate the average waiting times w_{vnf}, w_{vsw}, w_{edge}, w_{agg} and w_{core}.

When a packet is sent to the root controller it will wait at the controller and at a virtual switch again when it returns. All VNFs will send a portion of the messages they produce to the root controller:

$$\lambda_{sdn} = n \cdot p_{sdn} \cdot \lambda \tag{13}$$

Accounting for the additional waiting time at the server, the expectation of the waiting time incurred by the SDN controller is:

$$w_{sdn} = (MM1(\mu_{sdn}, \lambda_{sdn}) + w_{vsw}) \cdot p_{sdn} \tag{14}$$

By substituting the probabilities of the different paths and the mean waiting times at each component into Equation 1, we can determine the average latency in the network for the base case of a single path through the network.

D. Mean Latency of Long Services and Many Services

Most networks will require more complex services with more than two VNFs. A consequence of longer services is that individual packets persist in the network for a longer period of time.

Consider a situation where each VNF sends a packet to an adjacent network function every cycle so that all network functions receive a packet each cycle. Consider also that we have a service chain of three network functions so that packets will be required to make two passes through the network. After the first cycle all VNFs will have sent and received one packet. After the second cycle all VNFs will have sent two packets, forwarding the one received in the previous step and a new packet from this cycle, and also received two packets, a packet with no hops remaining and a packet with one hop remaining. At the third cycle one packet will be destroyed having completed the service, leaving one packet to be forwarded and one new packet created for each VNF. Effectively, each VNF is producing two packets per cycle on average.

We can extend this intuition to arbitrary length chains. The longer the service chain grows, the longer messages will be persisted in the network leading to larger effective production rates. Following the intuition, the effective production rate for an arbitrary length chain is:

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

where len is the number of network functions that compose a given service chain and $chain_i$ is the service being modelled.

Similarly it is unlikely that the network will provide only one service. We can further extend the model to several services, each of which may have different numbers of network functions. If a given packet has probability $p(chain_i)$ of belonging to a particular service the expectation of the service chain length determines the average effective production rate:

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

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

Finally, longer services will necessarily require more passes through the network. As Equation 1 provides the average latency for a single pass through the network, the waiting time for a variable number of passes is dependant on Equation 1 and the average number of passes in the network:

$$Latency = Latency_{base}(\lambda_{eff}, \mu, \mu_{vnf}, \mu_{sdn}) \cdot \sum_{i=1}^{ns} p(chain_i)(len(chain_i) - 1) \tag{17}$$

IV. Validation

To verify the accuracy of the analytical model it has been validated using a discrete event simulator built in OMNeT++ [17]. Each simulation experiment was run until the network reaches its 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, and p_{sdn_root}. To remain concise latency results are presented for selected cases. For all cases, where not otherwise specified, the following parameter settings are used:

- $k = 4$, $k_{vsw} = 2$ and $p_{sdn\_root} = 0$
- All switches and the root SDN controller have the same service rate of 40 messages per cycle ($\mu = 40$, $\mu_{sdn} = 40$)

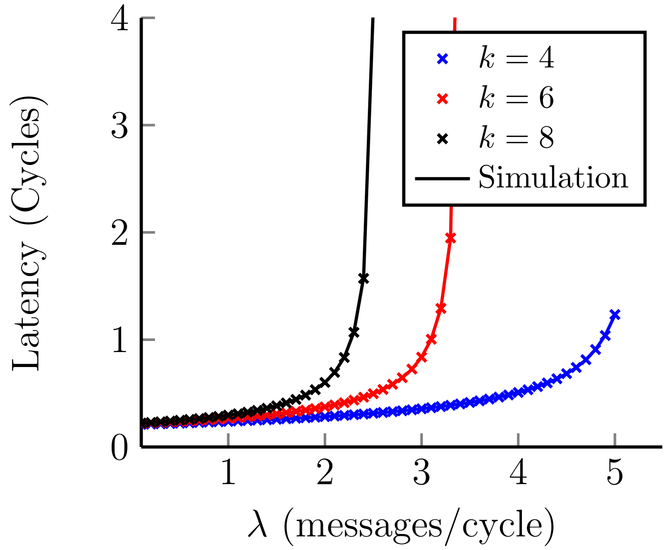

Fig. 3. Latency predicted by the model and simulation for different numbers of ports (k).

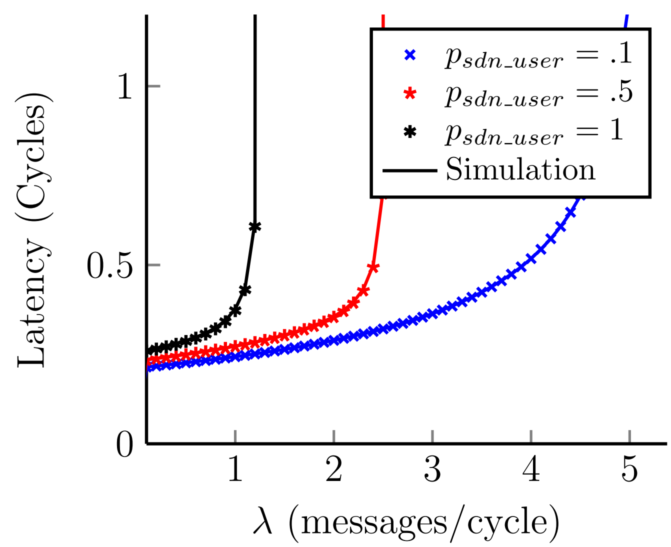

Fig. 4. Latency predicted by the model and simulation with different proportions of packets routing via the SDN controller (p_{sdn_root}).

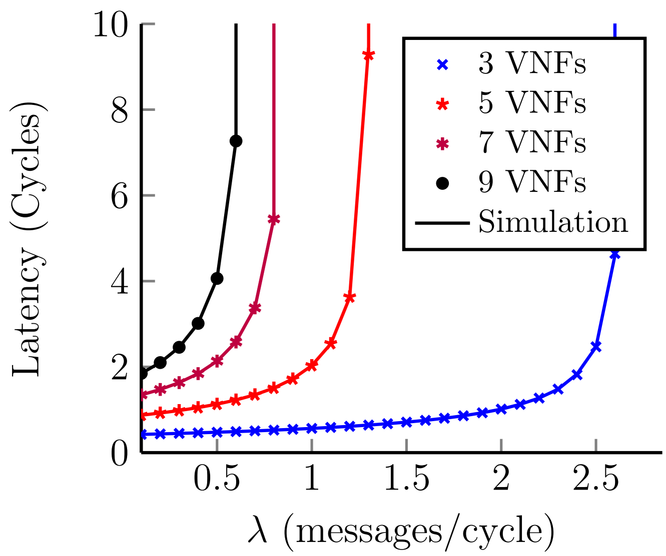

Fig. 5. Latency predicted by the model and simulation for different length service chains.

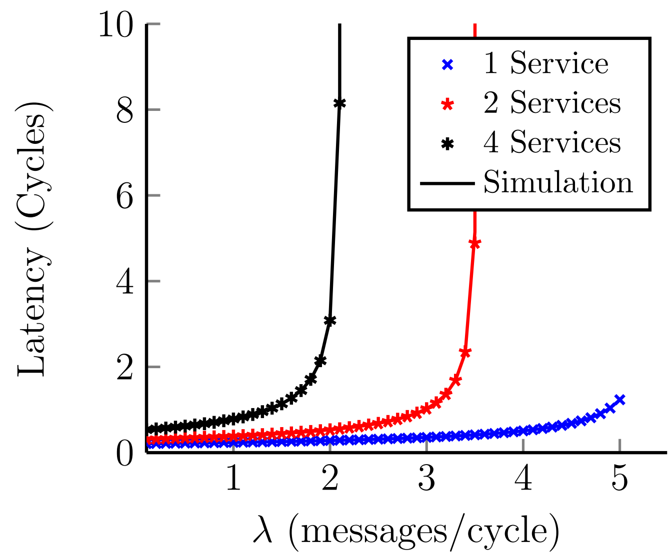

Fig. 6. Latency predicted by the model and simulation for several services of different lengths.

- The VNFs have a service rate of 20 messages per cycle ($\mu_{vnf} = 20$)
- The network holds one service with two VNFs
- Services are selected with equal probability

Additionally, when multiple services in the same network are considered (as in Figure 6) each service is assigned a length the same as its index plus one. As a consequence, tests with more services have higher average lengths. This decision was made to ensure the different tests were distinct from each other.

Figures 3 to 6 depict mean message latency predicted by the model plotted against those provided by a discrete event simulator for a range of parameter settings. For the model, 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 those predicted by the model. The tractability and accuracy of the analytical model make it suitable for analysis of next generation NFV and SDN enabled communications networks.

V. Performance Analysis

Having validated its accuracy, the analytical model can now be used to investigate the performance of SDN and NFV enabled networks under load. In particular it is useful to be able to identify potential bottlenecks that would lead to poor performance.

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 the first network components to receive a higher arrival rate than their service rate leading to high latency or packet loss. Figure 5 shows that this issue is aggravated by the addition of NFV features into the network. With longer service chains, although all switches receive proportionally the same increase in traffic, aggregate switches are already the most loaded so they can be overloaded whilst other network elements have spare capacity.

In Figure 4 we see that 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 Equation 9, an additional $k_{vsw} \cdot p_{sdn} \cdot \lambda$) when a large proportion of the packets are required to visit the root SDN controller. If the virtual switches are able to handle the same rate of traffic as the physical switches the edge switch remains as the bottleneck. However if the virtual switches are substantially less powerful, as in these tests, the virtual switches can become the bottleneck.

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 accurately model the impact of one or more services of different lengths in the network.

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. 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 for practical applications.

This model cannot determine the effect of VNFs that increase or reduce the outgoing packet rate. Optimisation of the ordering of filtering VNFs could reduce the impact of increased lifetimes of packets on the network.

Further, M/M/1 queues have been shown to underestimate the latency when faced with more realistic bursty traffic [18]. MMPP queues have been used in the literature to model bursty traffic [8], [18] and may be an appropriate extension.

References

[1] J. G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. E. Lozano, A. C. K. Soong, and J. C. Zhang, "What will 5g be?" *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 6, pp. 1065–1082, 2014.

[2] GSA, "The road to 5g: Drivers, applications, requirements and technical development," 2015, [Online; accessed 2018-05-04]. [Online]. Available: https://goo.gl/TsRqKT

[3] H. Kim and N. Feamster, "Improving network management with software defined networking," *IEEE Communications Magazine*, vol. 51, no. 2, pp. 114–119, 2013.

[4] S. Hares and R. White, "Software-defined networks and the interface to the routing system (I2RS)," *IEEE Internet Computing*, vol. 17, no. 4, pp. 84–88, 2013.

[5] J. Matías, J. Garay, N. Toledo, J. Unzilla, and E. Jacob, "Toward an sdn-enabled NFV architecture," *IEEE Communications Magazine*, vol. 53, no. 4, pp. 187–193, 2015.

[6] F. Longo, S. Distefano, D. Bruneo, and M. Scarpa, "Dependability modeling of software defined networking," *Computer Networks*, vol. 83, pp. 280–296, 2015.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] Cisco, "Cisco global cloud index: Forecast and methodology, 2016-2021," 2018, [Online; accessed 2018-05-04]. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/global-cloud-index-gci/white-paper-c11-738085.html

[14] VMware, "Network virtualisation and security platform - nsx," 2018, [Online; accessed 2018-05-04]. [Online]. Available: https://www.vmware.com/uk/products/nsx.html

[15] ——, "Nsx growth and success in 2016," 2016, [Online; accessed 2018-05-04]. [Online]. Available: https://blogs.vmware.com/networkvirtualization/2017/01/nsx-growth-success-2016.html/

[16] L. Kleinrock, *Theory, Volume 1, Queueing Systems*. Wiley-Interscience, 1975.

[17] A. Varga and R. Hornig, "An overview of the omnet++ simulation environment," in *Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops, SimuTools 2008, Marseille, France, March 3-7, 2008*, S. Molnár, J. R. Heath, O. Dalle, and G. A. Wainer, Eds. ICST/ACM, 2008, p. 60. [Online]. Available: https://doi.org/10.4108/ICST.SIMUTOOLS2008.3027

[18] 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.