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*—Next generation communications networks are exploring the use of new technologies including software defined networks (SDN) and network function virtualisation (NFV) but existing models only consider these components in isolation. A novel analytical is developed for next generation, SDN and NFV enabled communications networks. Each network component in a layer is modelled as an M/M/1 queues and the expected latency is calculated. 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 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 naive placement of VNFs. It also serves as a first step towards developing an efficient analytical model for optimisation of VNF placement.**

I. Introduction

Next generation communications networks are faced with scaling to support greater expectations and usage of existing services [1] whilst simultanously 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).

Fast analytical models allow for evaluation and optimisation of large scale networks. Existing research into analytical models of NFV and SDN has focussed on them in isolation. Longo et al. [3] constructed a model of the reliability of a two layer hierarchical SDN controller. Similarily, Azodolmolky et al. [4] also examine the two layer SDN controller but 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. [5] 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.

Similarly in NFV modeling, Prados-Garzon et al. [6] 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. [7] 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. [8] 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. To the best of our knowledge we are the first researchers to produce an integrated model that can consider the impact of the SDN controller, NFV in a practical network topology. 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.
- Extensions for different length service chains and multiple services in the same network.
- An extension to model 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 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 run on virtual machines on commodity hardware.

A service is a sequence of several virtual network functions (VNF). They can be defined using Directed Acyclic Graphs (DAG) which encapsulate dependencies between network functions. Some network functions such as firewalls

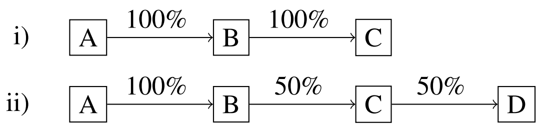

Fig. 1. Two VNF service chains represented with directed acyclic graphs. Service i has a length of three and 100% of the produced packets go through all VNFs. Service ii is four long and the proportion of packets passed between VNFs reduces at each VNF.

can reduce the data rate by filtering out packets. The task of constructing a 'concrete' service from an abstract DAG is called service composition and has received interest in optimisation [9]–[11]. Examples of some concrete services can be seen in Fig. II-A.

B. Software Defined Networking

Software Defined Networking (SDN) allow for dynamic routing of packets throughout the network and configuration of VNFs [12], [13]. A logically centralised but often physically distributed SDN controller maintains a global view of the network. A portion of the packets generated in the network will visit the SDN controller.

In this work we model a two layer hierachical SDN controller similar to that evaluated by Azodolmolky et al. [4]. In this architecture a local SDN controller exists at each server and an additional root SDN controller exists as an independant 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. A comparable implementation of this concept is VMwares NSX [14].

C. The Network Architecture

In this work we extend the Fat Tree network topology [15] to be SDN and NFV capable. Fig. II-C illustrates this network. The network topology is defined as follows:

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.

III. Analytical Model

A. Assumptions

The analytical model is based on the following assumptions:

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) At every network component packets are serviced according to an independent Poisson process with a mean rate of μ packets a cycle.

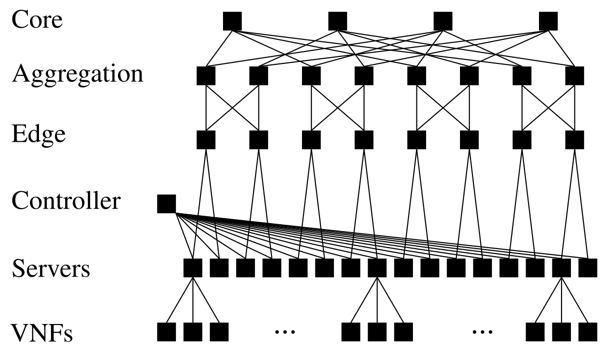

Fig. 2. The network topology modelled in this paper

3) The time taken for a packet to travel between network components is negligible.
4) The SDN controller ensures packets take the shortest path between the two destinations 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 may need to visit the root SDN controller with probability p_{sdn_root}.

The mean latency of the network can be calculated as the sum of the waiting time at each network component that a packet visits. As a result of the network architecture, packets will take shorter paths to visit VNFs under the same server, edge switch or pod. Consequently, the traffic arriving at network components at each layer in the network will also vary. The expected waiting time can be calculated as follows:

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

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 switch respectively. Similarly p_{vsw}, p_{edge}, p_{agg} and p_{core} represent the probability that the highest level a packet which reach is a virtual, edge, aggregate or core switch respectively. We will now deduce these values for arbitary settings of k and $k_v sw$.

B. Probability of Highest Level

As packets will always take the shortest path, a packet will just visit a virtual switch if the destination VNF is in under the same virtual switch as the source. Given $n = (k^3/4) \cdot k_{vsw}$ different VNFs:

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

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.

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

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

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

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

Finally we can calculate the probability that a packet will visit the root SDN controller by considering the probability that a packet leaving a server will visit the controller and excluding all packets that will not leave the server:

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

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 with [16]:

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

Where nc is the network component under question. We can now calculate the arrival rate, λ_{nc}, for each network component.

As destinations are evenly distributed over the VNFs, each VNF will send an equal proportion of packets to every other:

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

A portion of the packets being produced will visit the root SDN controller:

$$\lambda_{sdn} = n \cdot p_{sdn} \cdot \lambda \quad (9)$$

Virtual switches can receive packets from three sources: generated from VNFs on the server, received from other VNFs and reflected packets that it had sent to the root SDN controller. Following the same format:

$$\begin{aligned} \lambda_{vsw} &= k_{vsw} \cdot \lambda \\ &+ (n - k_{vsw}) \cdot \frac{k_{vsw}}{n-1} \cdot \lambda \\ &+ k_{vsw} \cdot \lambda \cdot p_{sdn} \end{aligned} \quad (10)$$

Note that the packets sent back from the SDN controller do not impact higher level switches as the portion of packets that are sent to the controller are not sent to higher switches till later cycles and that this absence is filled by packets 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:

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

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:

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

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

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

By substituting 8 to 13 for the arrival rates at each network component into 7 for the average waiting time we can calculate the waiting times w_{vnf}, w_{vsw}, w_{edge}, w_{agg} and w_{core} i.e. all network components except for the SDN controller.

When a packet is sent to the root controller it will incur added latency both waiting at the controller and at the virtual switch again when it returns. The expectation of the waiting time incurred by the SDN controller is:

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

By substituting equations 2 to 6 and 8 to 14 into equation (1), we can calculate the average latency in the network for the base case where only one service exists and only one hop is required.

D. Mean Latency of Long Services and Many Services

As discussed in the preliminary section telecommunications services result in packets being passed through several network functions. As a result packets persist in the network for a longer period of time for longer services.

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

After the first cycle all VNFs will have sent and received 1 packet. After the second cycle all VNFs will have sent 2 packets, forwarding the 1 received in the previous step and a new packet from this cycle, and also received 2 packets, a packet with no hops remaining and one with 1 hop remaining. At the third cycle 1 packet will be destroyed having completed the service, leaving 1 packet to be forwarded and 1 new packet created for each VNF. The net result is that on average each VNF is producing 2 packets per cycle.

We can extend this intuition to arbitary length chains:

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

where λ_{long} is the effective number of packets that are generated by a VNF 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 packet has probability $p(chain_i)$ of belonging to a particular service the average number of hops that a packet persists for and hence the impact on the production rate is the weighted mean:

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

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

Finally we must also consider that packets 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:

$$\begin{aligned} Latency_{eff} = Latency(\lambda_{eff}, \mu) \cdot \\ \sum_{i=1}^{ns} p(chain_i)(len(chain_i) - 1) \end{aligned} \tag{17}$$

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 packets that they receive. This impacts subsequent network functions in a service chain that will have lower arrival and hence lower production rates.

Consider a situation where we have a service chain with 4 VNFs with 2 filtering VNFs that filter 50% of the packets they receive, illustrated in IV. After the first cycle all VNFs will have sent and received 1 packet. After the second cycle all VNFs will have sent at least one packet, and half of the VNFs will have forwarded another packet. After the third cycle all VNFs will have produced at least 1 packet, another half will have forwarded the packet received in the previous cycle and a quarter will have forwarded the packet sent in the first cycle bringing the average production rate to 1.75.

Using this intuition we can calculate the effective production rate as the sum of the production rates considering the impact of earlier filtering VNFs. To incorporate multiple services we again must average the production rates of each service. The complete algorithm for calculating the effective production rate considering all aspects is as follows:

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

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:

- $k = 4$, $k_{vsw} = 2$ 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
- The network holds one service with two VNFs
- 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.

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 proportionally the same increase in traffic but as aggregate switches are already the most loaded, they 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

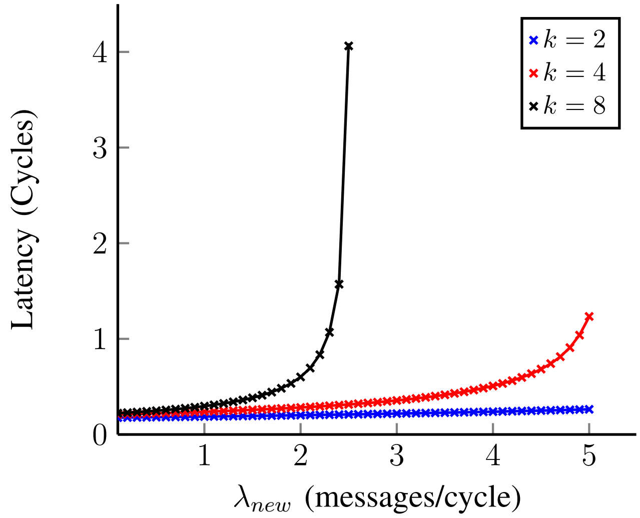

Fig. 3. Latency predicted by the model and simulation for different numbers of ports (k).

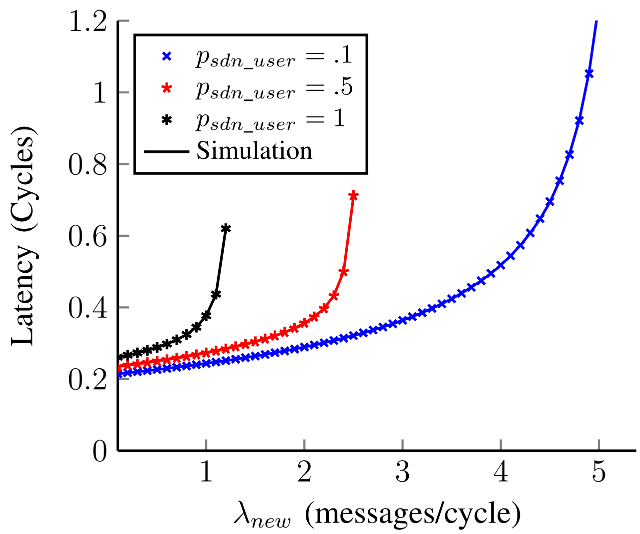

Fig. 4. Latency predicted by the model and simulation with different proportions of packets routing via the SDN controller.

the ordering of filtering VNFs considering dependancies could help reduce the issues caused by increased lifetimes of packets in 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 10, an additional $k_{vsw} \cdot \lambda * p_{SDN}$) when a large proportion of the packets are required to visit the root 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.

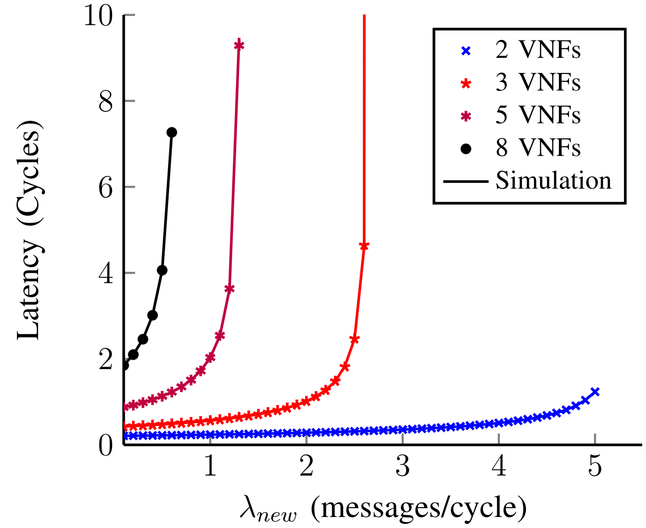

Fig. 5. Latency predicted by the model and simulation for different length service chains.

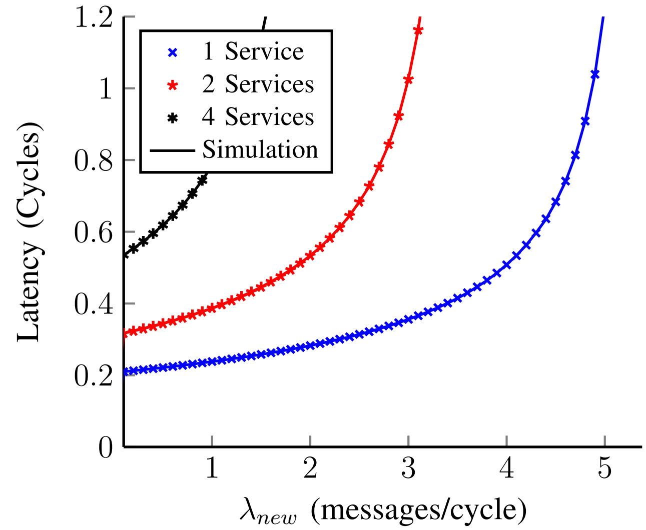

Fig. 6. Latency predicted by the model and simulation for several services of different lengths.

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 should be widely applicable.

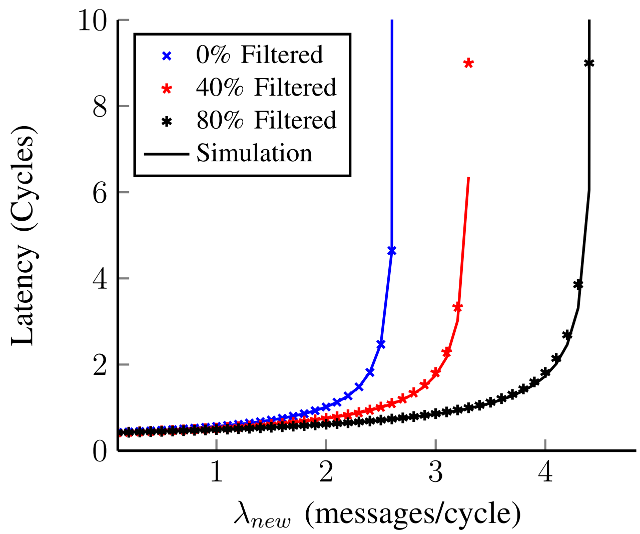

Fig. 7. Latency predicted by the model and simulation for a three long service chain where the second VNF only forwards a portion of the traffic.

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 [17]. Similarly the current model can not accurately model VNFs 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 [5], [17] and may be able to solve these problems.

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. [Online]. Available: https://doi.org/10.1109/JSAC.2014.2328098

[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] 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

[4] 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

[5] 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

[6] 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

[7] 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

[8] 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

[9] S. Mehraghdam, M. Keller, and H. Karl, "Specifying and placing chains of virtual network functions," in *3rd IEEE International Conference on Cloud Networking, CloudNet 2014, Luxembourg, Luxembourg, October 8-10, 2014*. IEEE, 2014, pp. 7–13. [Online]. Available: https://doi.org/10.1109/CloudNet.2014.6968961

[10] M. F. Bari, S. R. Chowdhury, R. Ahmed, and R. Boutaba, "On orchestrating virtual network functions," in *11th International Conference on Network and Service Management, CNSM 2015, Barcelona, Spain, November 9-13, 2015*, M. Tortonesi, J. Schönwälder, E. R. M. Madeira, C. Schmitt, and J. Serrat, Eds. IEEE Computer Society, 2015, pp. 50–56. [Online]. Available: https://doi.org/10.1109/CNSM.2015.7367338

[11] H. Wada, J. Suzuki, Y. Yamano, and K. Oba, "E^3: A multiobjective optimization framework for sla-aware service composition," *IEEE Trans. Services Computing*, vol. 5, no. 3, pp. 358–372, 2012. [Online]. Available: https://doi.org/10.1109/TSC.2011.6

[12] H. Kim and N. Feamster, "Improving network management with software defined networking," *IEEE Communications Magazine*, vol. 51, no. 2, pp. 114–119, 2013. [Online]. Available: https://doi.org/10.1109/MCOM.2013.6461195

[13] 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. [Online]. Available: https://doi.org/10.1109/MIC.2013.76

[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] 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

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

[17] 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