Modelling and Analysis of SDN and NFV Enabled Communications Networks

Joseph Billingsley, Wang Miao, Geyong Min, Nektarios Georgalas
Department of Computer Science
University of Exeter, UK
Email: {jb931, wang.miao, g.min}@exeter.ac.uk,
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 making efficient resource allocation possible. Whilst these technologies have received much interest, existing models only consider these components in isolation. To achieve a deep understanding, 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 illuminate disproportionate load on certain layers of the network and serves as a step towards developing an efficient analytical model for optimisation of next generations 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).

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 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 to meet demand allowing for efficient resource consumption, allowing demand to be met 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 or types of VNFs. Additionally, one or more services may be provided by the datacentre simultaneously.

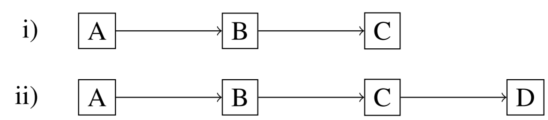

Fig. 1. Two services of different lengths, i) is three long whilst ii) is four long. These and other services may exist in the network at the same time

Existing research in NFV modelling has focussed on NFV in isolation. Prados-Garzon et al. [3] 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. [4] 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. [5] considered both NFV and SDN. They analysed 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.

B. Software Defined Networking

Software Defined Networking (SDN) allow for dynamic routing of packets throughout the network and configuration of VNFs [6], [7]. 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.

As with NFV, SDN modelling and performance analysis has focussed on SDN in isolation. Longo et al. [8] constructed a model of the reliability of a two layer hierarchical SDN controller. Similarly, Azodolmolky et al. [9] 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. [10] 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.

Existing work in NFV and SDN only considers each technique in isolation and does not consider the wider network that these technologies are integrated into. A model of the entire system is necessary to model the interaction between these components. 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 an NFV and SDN enabled datacentre network is proposed
- Additional extensions for modelling of multiple services of different lengths are provided
- 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 [11]. The fat-tree or folded-Clos topology is currently the most common topology used for datacentre communication networks [12]. 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 down an efficient path from the source to destination VNF. An illustration of an SDN and NFV enabled version is provided in Figure 2

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 [9]. A simple extension is the two layer hierarchical structure. In this architecture, a 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 [13] which as of 2016 had been deployed on over 2400 networks [14].

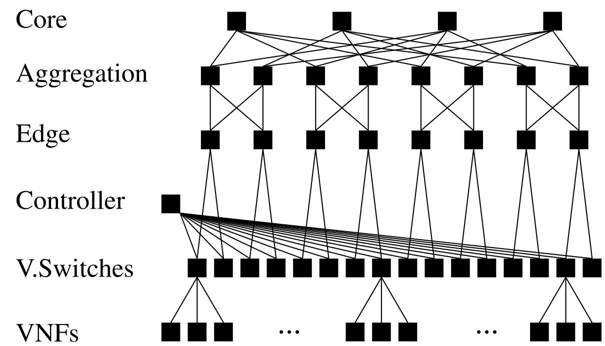

Fig. 2. An example SDN and NFV enabled fat-tree network with 4 ports for each hardware switch and 3 for the virtual switches.

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. We model the network in the same way, each network component is modelled as an M/M/1 queue based on some particular 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.
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}.

Packets are required to visit every switch on an efficient path between the source VNF and the edge 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 one time. The mean latency in this case is dependent only on the latency for each path and the probability of a packet taking that path:

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

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

B. Probability of Highest Level

As packets will always take the shortest path, a packet will not leave the server if the destination VNF is in under the same virtual switch as the source. Hence the probability of a packet only visiting the virtual switch is the proportion of destinations that are under the virtual switch:

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

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 technique can be used to find the remaining probabilities:

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

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

Finally 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 [15]:

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

where nc is the network component under question.

As destinations are evenly distributed over the VNFs, each VNF will receive an equal proportion of packets from every other VNF.

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

Virtual switches can receive packets from three sources: packets generated from VNFs on the server, packets intended for VNFs on the server and reflected packets that it had sent to the root SDN controller. All traffic generated by the VNFs on the server will go through a virtual switch. An equal portion of the traffic generated by the other VNFs in the network will be incoming 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 is:

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

Note that the packets sent to the SDN controller will not affect higher level switches as though the packets that are sent to the root controller are not forwarded to higher switches till later cycles, this 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 can be calculated in the same way as for the virtual switch except there are more VNFs under the edge switch. For the outgoing traffic, packets that are intended for destinations on the same server will not visit the edge switch and must be excluded. The arrival rate at the edge switches is hence:

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

The same occurs for the aggregate switches only now all traffic will be split between each aggregate switch in the pod and a smaller proportion of generated packets must visit the aggregate switch:

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

Finally, as all VNFs are descendants of all 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:

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

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:

$$\lambda_{sdn} = n \cdot p_{sdn} \cdot \lambda \quad (13)$$

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

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

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 two long VNF service.

D. Mean Latency of Long Services and Many Services

Most networks will provide more complex services than the base case. A consequence of longer services is that individuals packets persist in the network for a longer period of time. Consider a situation where each VNF sends a packets to an adjacent network function every cycle 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 passes through the network.

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. Effectively, each VNF is producing 2 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. More formally the effective production rate for an arbitrary length chain is:

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

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 effective production rate:

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

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

Finally, longer services will require more passes through the network. This is given by multiplying Equation 1 by the average number of passes in the network considering each service:

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

IV. Validation

To verify the accuracy of the analytical model it has been validated using a discrete event simulator built in OMNeT++ [16]. 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 (6) 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 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. 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

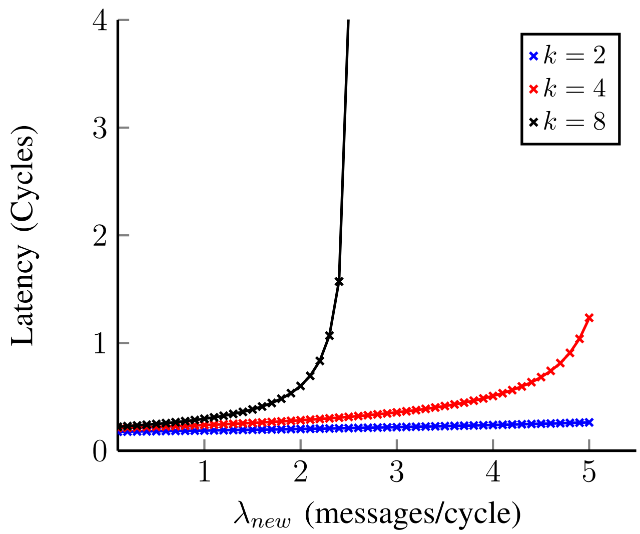

Fig. 3. Latency predicted by the model and simulation for different numbers of ports (k).

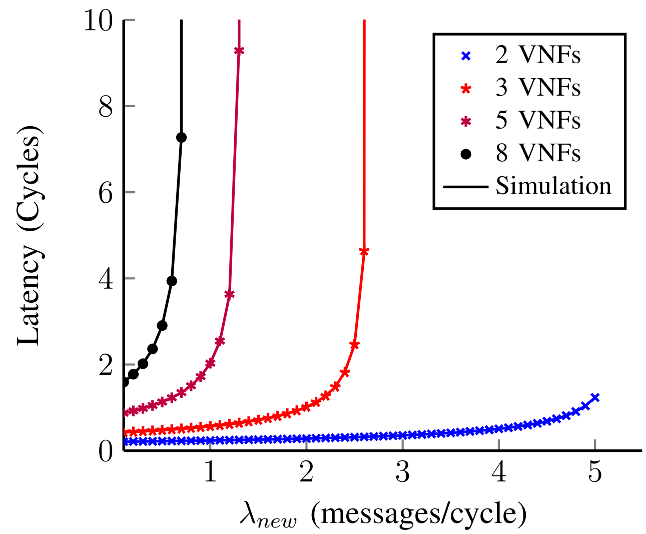

Fig. 5. Latency predicted by the model and simulation for different length service chains.

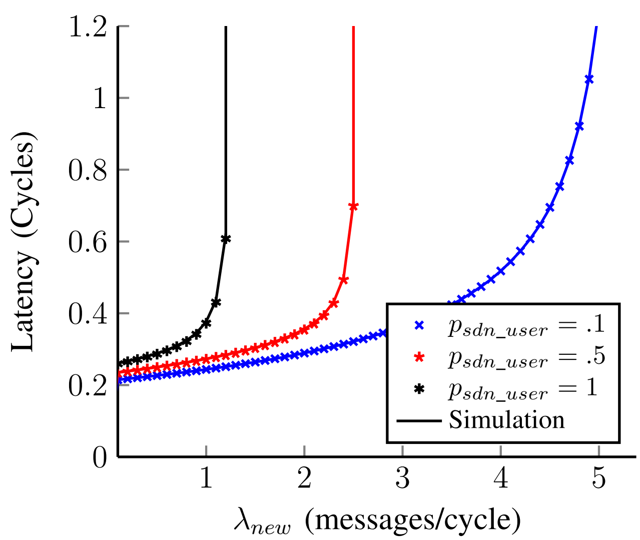

Fig. 4. Latency predicted by the model and simulation with different proportions of packets routing via the SDN controller.

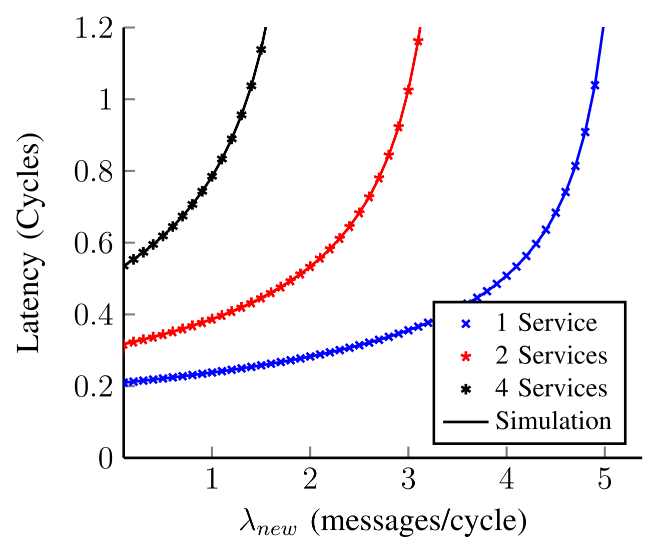

Fig. 6. Latency predicted by the model and simulation for several services of different lengths.

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 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 impact of VNFs that increase or reducethe outgoing packet rate has not been explored. Optimisation of the ordering of filtering VNFs considering dependancies would likely reduce the impact of increased lifetimes of packets in the network.

Further, M/M/1 queues have been shown to underestimate the latency when faced with more realistic bursty traffic [17]. MMPP queues have been used in the literature to model bursty traffic [10], [17] 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] 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.

[4] 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.

[5] 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.

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

[7] 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.

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

[9] 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.

[10] 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.

[11] 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.

[12] 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

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

[14] ——, "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/

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

[16] 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

[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.