Performance Analysis of SDN and NFV enabled Mobile Cloud Computing

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

***Abstract*—Mobile Cloud Computing (MCC) is regarded as a promising method to extend the battery life, increase the data storage and enhance the processing power of mobile devices. Technologies such as Software Defined Networking (SDN) and Network Function Virtualisation (NFV) will be deployed in MEC to simplify the network management and accelerate mobile service deployment. There have been some interesting research findings in the literature regarding the performance of SDN and NFV in MCC, however most of the existing work only considers these technologies in isolation and pays little attention to their cooperative and complementary relations in practical deployments. In order to achieve a deeper understanding of future MCC, a comprehensive analytical model is developed in this work to investigate the performance of MCC in the presence of both NFV service chains and SDN networks. The proposed model is capable of capturing the interactions between SDN and NFV when they share the same underlying physical infrastructure. The end-to-end latency is derived for different scales of service deployments and network configurations. Comprehensive simulation experiments are conducted and the results demonstrate that the proposed analytical model corresponds well with the simulation experiments. In addition, we should how the analytical model can be a useful tool to investigate the impact of centralised SDN control on the performance of NFV traffic transmission.**

I. Introduction

Emerging mobile services such as Virtual Reality, 4K video, and tactile internet, consume incredible amounts of compute, storage and bandwidth resources [1]. However, due to the inherent constraints of their size, weight and power, mobile devices struggle to meet the resource requirements of these new applications. Mobile Cloud Computing (MCC) [2], [3] has been considered as a key enabling technology which may allow mobile devices to provide these services. By migrating the local applications and services to a mobile cloud datacenter, MCC can extend the battery life, increase data storage and enhance the processing power of existing devices. To realise this ambition, Software Defined Networking (SDN) and Network Function Virtualisation (NFV) have been regarded as two promising and complementary technologies in MCC datacenters to simplify datacenter network management and improve resource utilisation and service flexibility.

SDN is a new networking concept that can simplify network management and accelerate network innovation. It is implemented by decoupling the network control from the underlying network infrastructure and creating a software-programmable controller. A logically centralised SDN controller maintains a global view of the network and determines how packets should be routed through the network [4], [5]. With centralised network control, network intelligence is migrated from the underlying network devices to the SDN controller, enabling network operators to manage the network consistently and holistically.

NFV is a novel network technology which allows for flexibility in service provisioning. Traditionally, services are constructed by connecting chains of purpose built network components, each performing a particular function. These may be traditional data centre functions such as firewalls and load balancers, or mobile communications functions such as the Packet and Service gateways in the 4G Evolved Packet Core. NFV decouples these functions from the hardware by implementing these network functions in software on virtual machines (VMs). These Virtual Network Functions (VNFs) can be moved, scaled or destroyed on demand, allowing for efficient placement and allocation of resources, significantly accelerating the deployment of new services.

SDN and NFV are often considered complementary technologies in practical deployments [6]. For instance, when a new MCC service needs to be deployed in the cloud datacenter, the cloud management system first constructs a VNF service chain and leverages the Virtual Network Function (VNF) manager to deploy VNFs on Virtual Machines (VMs) or containers. After instantiating the VNFs, the address of the VM or container is sent to the SDN controller, which is responsible for establishing connections between VNFs. From this example, it is clear that SDN plays an important role in the deployment, management and optimisation of NFV services. Therefore, it is necessary to jointly consider SDN and NFV when designing systems or models. In system optimisation, analytical models can provide insight into the system operation by formally defining the interactions among key parameters such as the scale and resources utilisation of the datacenter, the traffic generated by the end devices and the Quality of Service (QoS).

There have been some research efforts to build an analytical model of SDN and NFV network architectures. In SDN network modelling, Longo et al. [7] proposed a model to investigate the reliability of SDN networks based on a two layer network management architecture. Azodolmolky et al.

[8] used network calculus to investigate the worst case delay performance of an SDN network as well as the minimum buffer size required to meet a given delay constraint. To capture the burstness of the network traffic, Wang et al. [9] developed an analytical model to investigate an SDN architecture where traffic was modelled as a Markov-Modulated Poisson Process (MMPP).

Similarly, some researchers have considered NFV systems in their models. Prados-Garzon et al. [10] designed an analytical model to investigate the average response time of a single NFV service provisioning. Gebert et al. [11] modeled each step in the packet processing pipeline of a single VNF. To analyse the stochastic performance of multiple VNFs, an analytical model was proposed in [12] which used stochastic network calculus to investigate the worst case end-to-end performance of an NFV service provisioning for a given QoS requirements.

Although, these existing works provide some insights into the performance of SDN and NFV network architecture in various network scenarios, they do not jointly consider these two technologies in the performance analysis. From the perspective of service deployment and provisioning, SDN and NFV are complementary technologies and are often deployed together. Therefore, it is important to investigate the performance of network infrastructure with both SDN and NFV support, especially identifying how their interactions can affect the performance of service provisioning. To the best of our knowledge, only Fahmin et al. [13] have considered both NFV and SDN in their analytical model. However the network infrastructure adopted in [13] consists of only one switch and one VNF, which is too small a model in comparison to actual datacenter networks. In order to reap the benefits of SDN and NFV for MCC applications, there is an urgent need to develop a novel analytical model which can jointly consider the two complementary technologies for large-scale datacenter networks.

To fill this gap, a comprehensive analytical model is proposed in this work to investigate the performance of SDN and NFV enabled MCC datacenter networks. To capture the unique features of real-world SDN and NFV deployments we consider a network providing multiple services with NFV and a virtualised SDN implementation, where the SDN controller determines how traffic is routed among the VNFs. The analytical model is developed with the aim of understanding the interactions between SDN and NFV when they are deployed on the same infrastructure, e.g. the impact of the length of NFV service chain on the traffic engineering performance of SDN networks. The end-to-end performance in terms of the average latency is obtained by the developed model and validated through extensive simulation experiments under different network configurations. In addition, we show how the proposed analytical model can provide useful insights when designing services and networks for MCC datacenters.

The remainder 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. Finally Section V concludes the paper.

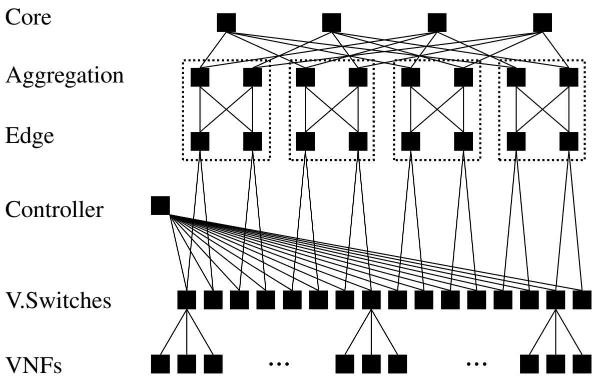

Fig. 1. An example SDN and NFV enabled 4 port fat-tree network.

II. Network Architecture

When a MCC service is requested in the cloud datacenter, the cloud management system analyses the performance and functional requirements of the service. Using Network Function Virtualisation (NFV), a MCC service is provided by defining a service chain consisting of several VNFs (Virtual Network Functions). These VNFs must be placed in VMs or containers. Packets for that service must then be routed through each of the VNFs in sequence. Multiple service chains can coexist and different service chains can have different numbers of VNFs. The SDN controller determines packet routing by maintaining routing tables in every SDN enabled switch. In this work, we assume that due to physical storage limitations, it is impossible to store the instructions for all possible destinations in a switch. If a virtual switch receives a packet that does not match any entry in its routing table, a request will be sent from the switch to the controller for instructions on how to process the received package.

Based on these details of SDN and NFV implementations, the SDN enabled fat-tree networking topology shown in Fig. 1 is considered. The fat-tree network topology is formed from three layers of switches: core, aggregation and edge switches. In most modern datacenters, the switches at the edge layer are connected to Top-of-Rack (ToR) switches, and the VNFs are hosted in VMs or containers on servers on these racks. In this work we consider an SDN architecture where only the virtual switches connect to the SDN controller. This architecture is representative of those used in industry, most notably a comparable architecture is used in VMWare's NSX solution [14]. As shown in Fig. 1, the fat-tree topology is defined by the number of ports at each switch. Let k denote the number of ports for each physical switch and k_v be the number of ports for each virtual switch. Each core switch connects to one aggregation switch of each of k pods. Within each pod, a layer of aggregation switches is fully connected to a layer of edge switches. In addition, each edge switch is connected to $k/2$ servers. Each server contains a virtual switch connected to k_v VNFs. In total, a three layer k port fat-tree topology has $(k/2)^2$ core switches, k pods, $k^2/2$ aggregation switches, $k^2/2$ edge switches, $k^3/4$ virtual switches, and $(k^3/4)\cdot k_v$ VNFs.

III. Analytical Model Derivation

In this section, we derive an analytical model of an SDN and NFV enabled MCC datacenter. The model is capable of analysing the end-to-end service performance of multiple coexisting service chains, each of which can have different numbers of VNFs. The impact of the centralised SDN controller on the end-to-end performance provisioning is also considered. With the aim of increasing the readability of the model derivation, we first consider a scenario with a single service with only two VNFs. We then extend the simplified model into the realistic scenario of multiple NFV service chains with varying length.

1) Simple NFV Deployment Scenario: Following the work on SDN and NFV performance analysis in [7], [11], [12], our models assume that packets will arrive independently from each other and the expected time to process a packet is not dependent on earlier packets. Hence the arrival and service rates of packets in this study follow independent probability distributions. For each service, we assume the traffic entering the MCC datacenter follows an independent Poisson process with a mean rate of λ_i packets per second. Each physical/virtual node, e.g. switch, VNF and the controller, service incoming packets according to an independent Poisson process with service rates μ_s, μ_v and μ_c packets per second respectively. If a packet fails to match the routing table in the SDN-enabled virtual switches, routing information for the packet will be requested from the SDN controller. Let p_m denote the probability there is no routing entry for an incoming packet.

Further, in this work we assume no knowledge of the placement of VNFs in the network. Therefore we assume that all VMs may contain a VNF and that traffic leaving a VM can be intended for any other VM. However, in an actual placement of VNFs only the first VNF creates packets and the final VNF in a service chain does not create or forward any packets. Hence the effective traffic rate leaving each VNF in the model must be lower than the actual traffic rate for the service. In the case of a service with only two VNFs we would expect half of the used VNFs to be final VNFs hence the effective arrival rate, $\lambda^e = \lambda/2$.

For the SDN and NFV enabled fat-tree network established in Section II, the end-to-end latency of a service is dependent on the probability that a packet will visit each layer of switches. If we apply the widely used Equal-Cost Multi-Path (ECMP) [15] routing protocol to this network, the expected arrival rate and hence the expected waiting time is the same for every component on each layer. For this case of a service with two VNFs, packets will only need to cross the network once and hence the expected latency depends on the expected level the packets must utilise to reach the other VNF. The latency from one VNF to another can then be written as,

$$L_h = w_v + \sum_{i=0}^{3} L_i(\lambda_i,\mu_i)\cdot p_i \tag{1}$$

where "$i = 0$" represents the virtual switch layer, and "$i = 1, 2, 3$" denotes the edge, aggregation and core layers respectively. $L_i(\lambda_i,\mu_i)$ denotes the end to end latency when the packets need to travel to the i th layer of switches and w_v gives the expected waiting time at the VNF. Similarly p_i represent the probability that a packet must use the i th layer of switches.

$L_i(\lambda_i,\mu_i)$ is the sum of the waiting time from the VNFs to the i th layer switch and back, and can be calculated by,

$$L_i(\lambda_i,\mu_i) = w_i + \sum_{j=0}^{i-1} w_j + \sum_{j=1}^{i-1} w_j \tag{2}$$

where w_j is the processing time at the jth layer of switches. For the virtual switches, the waiting time w_0 includes the time waiting at the switch and also for a reply from the SDN controller: $w_0 = p_c(w_c + w_{vs}) + w_{vs}$, where p_c is the probability that a packet will be forwarded to the SDN controller and w_{vs} is the waiting time at the virtual switch.

We now calculate the remaining waiting times and probabilities. If the source and destination VNFs share the same virtual switch, then packets between two VNFs will not visit a higher layer switch. Let the probability of a packet only visiting a virtual switch be p_0 which can be calculated by

$$p_0 = \frac{k_v - 1}{n_v - 1} \tag{3}$$

where $n_v = k^3/4$ denotes the total number of VNFs in the datacenter.

In the fat-tree topology, the higher layer switches connect more VNFs. The probability of a VNF sending packets via an edge switch is the proportion of destination VNFs under the edge switch, excluding destinations that can be visited via a shorter path i.e. via a shared virtual switch. Therefore, p_1 can be derived with the following equation,

$$p_1 = \frac{\left(\frac{k}{2}-1\right)\cdot k_v}{n_v - 1} \tag{4}$$

Using the same method the probability of visiting the aggregate and core layers can be calculated with,

$$p_2 = \frac{\left(\frac{k}{2}-1\right)\cdot k\cdot k_v}{2\cdot(n_v - 1)} \tag{5}$$

$$p_3 = \frac{n_v - \left(\frac{k}{2}\right)^2\cdot k_v}{n_v - 1} \tag{6}$$

At the virtual switch, the SDN controller will only be consulted if the destination VNF is located in another physical server, and the destination is not contained in the routing table of the virtual switch in that server. The probability that the packets will be sent to the SDN controller for routing information can be calculated by,

$$p_c = (1 - p_0) \cdot p_m \tag{7}$$

After obtaining the probability that a packet will be processed at the different layers of switches and at the SDN controller. The following subsection derives the waiting time at each component of the routing path. According to [16], the waiting time for a M/M/1 queue is obtained by

$$w(\mu, \lambda) = \frac{1}{\mu - \lambda} \tag{8}$$

where μ is the service rate and λ is the arrival rate for an M/M/1 queue. In the following, we aim to calculate the arrival rate at the VNFs, SDN controller and different layers of switches. As the destinations are evenly distributed over the VNFs, each VNF will receive an equal proportion of packets from other VNFs. Hence each VNF receives traffic at rate λ^e. Virtual switches realise the communications among VMs, so virtual switches can receive packets from three sources: 1) packets generated by VNFs on the same server as the virtual switch 2) traffic generated by the VNFs in the other servers and 3) packets sent back from the SDN controller. Hence the arrival rate at the virtual switch can be calculated by,

$$\lambda_0^e = k_v \cdot \lambda^e \left(1 + \frac{n_v - k_v}{n_v - 1} + p_c\right) \tag{9}$$

The arrival rate for the edge switches can be calculated similar to the virtual switch. It should be noticed that packets that are intended for destinations on the same server do not need to visit the edge switch. The arrival rate at the edge switch can hence be calculated by,

$$\lambda_1^e = \frac{k \cdot k_v \cdot \lambda^e}{2(n_v - 1)} \left(2n_v - k_v \left(\frac{k}{2} + 1\right)\right) \tag{10}$$

Under the ECMP protocol, traffic will be balanced among aggregate switches in a pod and VNFs sharing the same virtual or edge switches will not use the aggregation switches. The arrival rate at each aggregate switch can then be computed by,

$$\lambda_2^e = \frac{\frac{k^2}{2} \cdot k_v \cdot \lambda^e}{k(n_v - 1)} \left(2n_v - k_v \left(\left(\frac{k}{2}\right)^2 + \frac{k}{2}\right)\right) \tag{11}$$

As all VNFs are connected by the core switches, the arrival rate at each core switch is the portion of traffic that cannot be reached by any other switch. Using ECMP, the traffic leaving the aggregation layer will be evenly split amongst the different core switches. Therefore the arrival rate at the core switch can be calculated with,

$$\lambda_3^e = \frac{\lambda^e \cdot p_3 \cdot n_v}{(k/2)^2} \tag{12}$$

Finally, let us calculate the traffic rate for the SDN controller. Given the number of VNFs $n_v = (k^3/4) \cdot k_v$ the arrival rate at the SDN controller is computed by,

$$\lambda_c = \lambda \cdot n_v \cdot p_c \tag{13}$$

By substituting the arrival rates (Eqs. 9 to 13) and service rates of each network component into the M/M/1 latency equation (Eq. 8), we can obtain the average waiting time at each VNF, and at the different layers of switches. Substituting the probabilities of the different paths and the mean waiting time into Eq. (1), we obtain the average end-to-end latency for the simple NFV deployment scenario.

2) Realistic NFV Deployment Scenario: Although there are several papers investigating the performance of NFV, existing research only considers the case of a single NFV service, paying little attention to the case of multiple coexisting services. As datacenter infrastructure is always used to simultaneously support multiple services, it is necessary to investigate the performance of an SDN and NFV enabled datacenter network with different numbers of services. In this subsection, we will extend the simplified NFV deployment scenario into the case of multiple VNFs of varying lengths.

Let N_s denote the number of NFV services deployed in the datacenter, K_i represent the length of the i th service and λ_i the traffic rate of the service. In an actual VNF placement, $K - 1$ of the K VNFs would emit traffic hence the effective traffic rate for the i th service is $\lambda_i^e = \lambda_i \cdot \frac{K_i - 1}{K_i}$. For the case of multiple coexisting services, several services of different lengths may produce traffic simultaneously at different rates. The expected total effective traffic rate is the effective traffic rate of each service considering the probability of a packet belonging to each service which can be calculated from the relative arrival rates,

$$\lambda^e = \sum_i^{N_s} \lambda_i^e \cdot \frac{\lambda_i}{\sum_j^{N_s} \lambda_j} \tag{14}$$

Similar to the simple NFV deployment case, the expected end-to-end latency is the sum time spent taking each path. By substituting the effective traffic rate in Eq. (14) into Eqs. (9-13), we can calculate the effective network traffic in each network layer. Given the service rates μ_v, μ_e, μ_a, and μ_s, the latency, w_j can then be obtained for each network layer. After calculating the probability that a packet visits a certain network layer, the expected latency between two VNFs can be calculated by Eq (1). Finally, as longer services require visiting several VNFs the total latency is given by the expected latency between two VNFs times the expected length of the service,

$$L_t = L_h \sum_i^{N_s} K_i \cdot \frac{\lambda_i}{\sum_j^{N_s} \lambda_j} \tag{15}$$

For convenience, pseudocode for the entire process is given in Algorithm 1.

IV. Validation and Performance Analysis

A. Model Validation

To verify the accuracy of the analytical model, a discrete event simulator has been built using OMNeT++ [17] to simulate a NFV and SDN enabled datacentre network. Each simulation experiment was run until the network reaches its

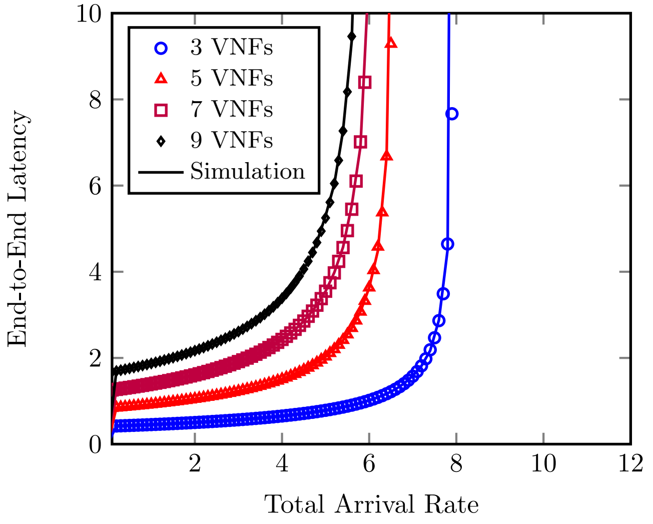

Fig. 2. Latency predicted by the model and simulation for different numbers of ports ($N_s = 1$, $K_i = 2$, $k = 4, 6, 8$, $k_v = 2$, $p_m = 0$).

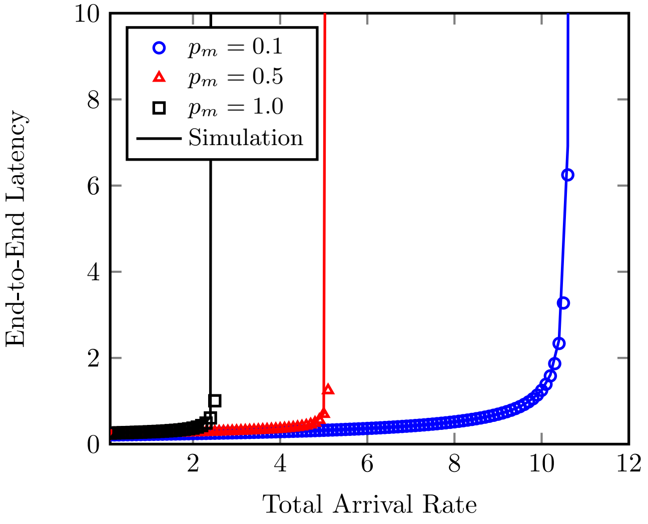

Fig. 3. Latency predicted by the model and simulation for different miss rates ($N_s = 1$, $K_i = 2$, $k = 4$, $k_v = 2$, $p_m = 0.1, 0.5, 1.0$).

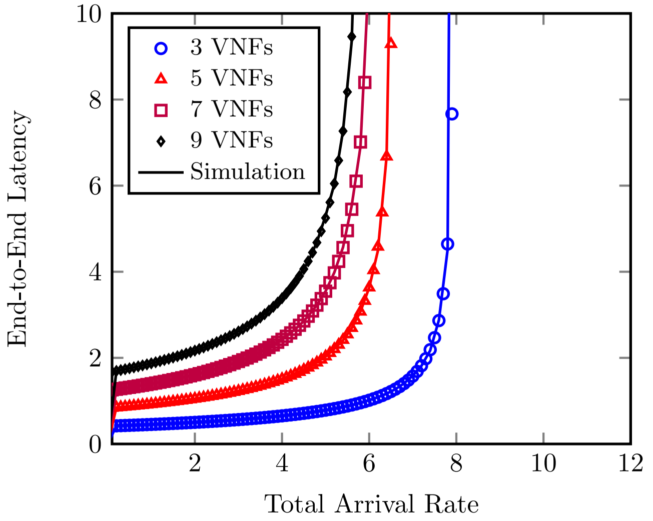

Fig. 4. Latency predicted by the model and simulation for a single service with different lengths ($N_s = 1$, $K_i = 3, 5, 7, 9$, $k = 4$, $k_v = 2$, $p_m = 0$).

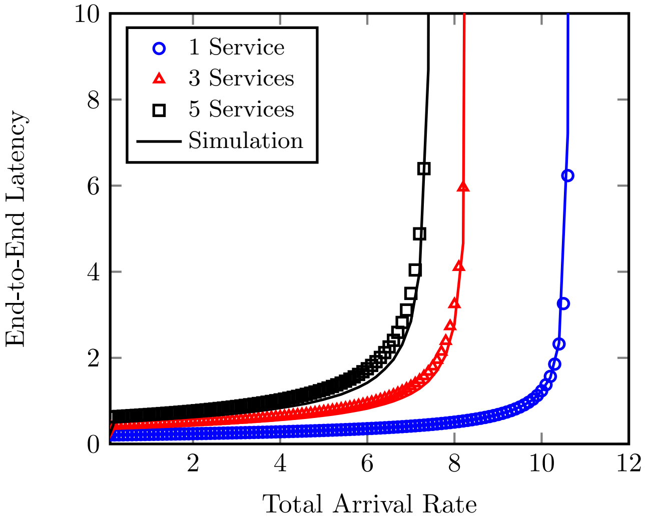

Fig. 5. Latency predicted by the model and simulation for several services with different length service chains ($N_s = 1, 3, 5$, $K_i = 2 : (N_s+1)$, $k = 4$, $k_v = 2$, $p_m = 0$).

Algorithm 1 Calculation of Average Latency of SND and NFV-enabled MCC Datacenter Networks

```
Calculate the effective network traffic: $\lambda_f^e$          (Eq. (14))
Calculate the traffic rates: $\lambda_i$                        (Eqs. (3-7))
Calculate the probabilities: $p_i$                              (Eqs. (9-13))
Calculate the waiting times: $w_i$, $w_v$ and $w_{vs}$          (Eq. (8))
Calculate the latency for each path: $L_i$                      (Eq. (2))
Calculate the latency for a single hop: $L_h$                   (Eq. (1))
Calculate the end-to-end latency: $L_t$                         (Eq. (15))
```

steady state where further network cycles do not change the collected statistics appreciably. Comprehensive simulation experiments were conducted to validate the performance of the proposed analytical model under different network configurations. However for the sake of specific illustration only a selection of tests are presented here and the results comparison between the analytical model and simulation experiments are presented in terms of the average end-to-end latency.

In practice a datacentre can contain on the order of tens of thousands of servers [18], with each switch supporting 1 to 100Gbits/s traffic a second. It is not feasible to simulate the scale of datacenter network in lab environment. Therefore, a scaled down version of a typical datacentre is modelled. Except where otherwise stated, the following parameters are used in our tests:

- $k = \{4, 6, 8\}$, $k_v = 2$ and $p_m = \{0.1, 0.5, 1\}$
- The service rate of the switches and SDN controller are set to be 40 packets per second ($\mu_v = 40$, $\mu_c = 40$)
- The service rate of the VNFs is set to be 20 packets per

second ($\mu_f = 20$)

- Services are selected with equal probability
- Case I: The network holds one service with two VNFs
- Case II: The network holds multiple services ($N_s = \{1, 2, 4\}$ and $K_i = \{3, 5, 7, 9\}$)

Figs 2 to 5 depict the mean message latency predicted by the model plotted against those provided by the simulation experiments 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 is lower than or equal to the service rate for all queues. 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 MCC datacentres.

B. Performance Analysis

We now show how the proposed analytical model can be a useful tool to optimise the design of an SDN and NFV enabled MCC datacenter network. We demonstrate the utility of the model for three key parameters: the scale of the network infrastructure, the table miss probability and the number and length of services.

1) Impact of the size of the datacenter on the end-to-end latency: The proposed analytical model can also be used to quantitatively analyse the relationship between latency and the size of the datacentre. From Fig. 2 we can see that, counterintuitively, larger datacentres that are at capacity can provide a worse QoS than a smaller datacenter under the same conditions. From Eqs. (3 to 5) we can see this is a consequence of the high probability of packets going via higher layers of the datacentre for larger numbers of ports, hence overloading the higher layers as many packets must traverse all of the network layers. In a practical deployment, datacentre designers can adopt the strategy of placing nearby VNFs close with each other to reduce the transmission latency.

2) Impact of the miss probability in SDN flow tables on the end-to-end latency: The SDN paradigm provides the benefit of a simplified network management and centralised system optimisation. However, from the packet forwarding perspective, the centralised control incurrs extra transmission latency during packet delivery. The proposed analytical model allows us to investigate the relationship between the reliance on centralised control and the end-to-end latency of each service. From the Fig. 3 we can see that as the system is only stable for low arrival rates when the SDN controller must frequently assist with routing instructions. Considering Eq. 13, we can see that the arrival rate at the SDN controller is proportional to the network traffic that is produced by the VNFs. Due to the total connectivity between the virtual machines and the SDN controller, even a slight increase of the traffic rate can overwhelm the SDN controller. To ensure that the SDN controller does not become a bottleneck in the system, network designers should ensure that routing tables at switches contain the majority of the information required for routing so that few requests must be sent to the controller, or ensure the processing capability of the SDN controller is sufficient to handle the traffic rate from the VNFs.

3) Impact of the number and length of NFV services on the end-to-end latency: Finally, we also consider the related situations of a different length services and of multiple services of varying lengths. As the datacenter network will be shared by multiple services to improve resource utilisation, it is important to investigate how these parameters impact the end-to-end latency. From Fig. 4 we can observe the expected latency increasing as the length of the service increases. This is a consequence of the increased effective arrival rate that results from longer services (Eq. 14). Correspondingly, we can see that for the case of a multiple services with on average lower service lengths (Fig. 5) the expected end-to-end latency is improved for higher relative arrival rate. This further demonstrates that that the key factor to consider when deciding whether to provide services from a datacentre is not the number of services, but instead their effective arrival rate.

V. Conclusion

In this paper we have presented a comprehensive analytical model to investigate the performance of a SDN and NFV enabled MCC datacenter. First, we presented an abstracted network architecture that captures the behaviour of SDN and NFV when they are implemented in a datacenter. Based on this network architecture, we developed a novel analytical model that can obtain the end-to-end latency for a given set of services. The proposed model is capable of investigating the interactions between SDN and NFV when they share the same underlying physical infrastructure. Extensive simulations were conducted to validate the performance of the proposed analytical model. Simulation results have demonstrated that the proposed analytical model matches well with simulation experiments. The proposed resulting analytical model is fast and accurate, and hence suitable for determining the optimal design of services and networks in large scale Mobile Cloud computing datacenters.

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*, 2014.

[2] R. Li, C. Shen, H. He, X. Gu, Z. Xu, and C. Xu, "A lightweight secure data sharing scheme for mobile cloud computing," *IEEE Trans. Cloud Computing*, 2018.

[3] M. R. Rahimi, N. Venkatasubramanian, S. Mehrotra, and A. V. Vasilakos, "On optimal and fair service allocation in mobile cloud computing," *IEEE Trans. Cloud Computing*, 2018.

[4] H. Kim and N. Feamster, "Improving network management with software defined networking," *IEEE Communications Magazine*, 2013.

[5] S. Hares and R. White, "Software-defined networks and the interface to the routing system (I2RS)," *IEEE Internet Computing*, 2013.

[6] J. Matías, J. Garay, N. Toledo, J. Unzilla, and E. Jacob, "Toward an sdn-enabled NFV architecture," *IEEE Communications Magazine*, 2015.

[7] F. Longo, S. Distefano, D. Bruneo, and M. Scarpa, "Dependability modeling of software defined networking," *Computer Networks*, 2015.

[8] S. Azodolmolky, P. Wieder, and R. Yahyapour, "Performance evaluation of a scalable software-defined networking deployment," in *EWSDN'13: European Workshop on Software Defined Networks*, 2013.

[9] W. Miao, G. Min, Y. Wu, H. Wang, and J. Hu, "Performance modelling and analysis of software-defined networking under bursty multimedia traffic," *TOMCCAP*, 2016.

[10] J. Prados-Garzon, P. Ameigeiras, J. J. Ramos-Muñoz, P. Andres-Maldonado, and J. M. López-Soler, "Analytical modeling for virtualized network functions," in *ICC'17: IEEE International Conference on Communications*, 2017.

[11] S. Gebert, T. Zinner, S. Lange, C. Schwartz, and P. Tran-Gia, "Performance modeling of softwarized network functions using discrete-time analysis," in *ITC'16: International Teletraffic Congress*, 2016.

[12] W. Miao, G. Min, Y. Wu, H. Huang, Z. Zhao, H. Wang, and C. Luo, "Stochastic performance analysis of network function virtualization in future internet," *IEEE Journal on Selected Areas in Communications*, 2019.

[13] A. Fahmin, Y. Lai, M. S. Hossain, Y. Lin, and D. Saha, "Performance modeling of SDN with NFV under or aside the controller," in *Fi-Cloud'17: International Conference on Future Internet of Things and Cloud Workshops*, 2017.

[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. Chiesa, G. Kindler, and M. Schapira, "Traffic engineering with equal-cost-multipath: An algorithmic perspective," *IEEE/ACM Trans. Netw.*, 2017.

[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 *SimuTools'08: International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops*, 2008.

[18] J. Hamilton, "Aws re:invent 2016," 2016, [Online; accessed 2018-05-04]. [Online]. Available: https://www.youtube.com/watch?v=AyOAjFNPAbA