2017 5th International Conference on Future Internet of Things and Cloud Workshops

Performance Modeling of SDN with NFV under or aside the Controller

Ahmed Fahmin[1], Yuan-Cheng Lai[2], Md. Shohrab Hossain[1], Ying-Dar Lin[3], and Dipon Saha[1]
[1]Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology, Bangladesh
[2]Department of Information Management, National Taiwan University of Science and Technology, Taipei, Taiwan
[3]Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan
Email:ahmedfahmin@gmail.com, laiyc@cs.ntust.edu.tw, mshohrabhossain@cse.buet.ac.bd, ydlin@cs.nctu.edu.tw, dipon2276@gmail.com

***Abstract*—Software Defined Networking (SDN) emphasizes the separation of data and control plane while network function virtualization (NFV) decouples network function from underlying hardware. Combining SDN with NFV would have many benefits, but the problem is how to integrate them. There are two possible architectures for such integration: the controller interacts with virtualized network functions (VNFs) or the switch interacts with VNFs. In this paper, the former is referred to as NFV under the controller (NFV_C) while the latter is called NFV aside the controller (NFV_AC). To the best of our knowledge, there is no analytical model for mathematically investigating the performance of such architectures. This paper therefore aims to carry out analytical modeling of SDN with NFV under or aside the controller. We model and analyze these two SDN+NFV architectures using M/M/1 queuing model and validate our analysis with various simulations. Results show that the analytical results well match the simulation results. Also a packet delay reduction of 54.67% can be achieved for NFV_AC over NFV_C, meaning that NFV_AC is a better architecture for integrating SDN with NFV.**

***Index Terms*—SDN, Network function virtualization, M/M/1 queuing model, OpenFlow.**

I. Introduction

Traditionally, a network is built on dedicated hardware, such as routers and switches, with network software provided by the network vendors. A network engineer's ability to customize the network software is very limited and is mostly restricted by the network vendor. This has led to the concept of Software Defined Networking (SDN), where flexibility and dynamism have been introduced in the virtualization of the control plane. The basic approach is to separate a network into a control plane and a data plane, thereby being able to manage various network devices centrally. The main benefit of the SDN concept is the programability of controlling the network devices. This has enabled network engineers to change network configuration and the logic of data flow according to the business requirements. On the other hand, Network Function Virtualization (NFV) is a new approach which deploys or designs various network functions. It decouples the network functions, such as NAT, DNS caching, etc. from their proprietary hardware appliances, so that they can be implemented in virtual machines, improving their service quality.

SDN focuses on the separation of the network control plane from the physical routers' data plane. SDN is treated as the control software that sits atop a bunch of physical devices with which it communicates through interfaces. On the other hand, NFV is about virtualizing various resources into network functions in software, so that we do not need any specialized physical devices dedicated to any particular network functions. Recent works [1], [2] have shown that both SDN and NFV can be combined to provide more centralized control software and generic hardware where the utilities of SDN can be realized through virtualized robust network functions provided by NFV.

There are several works [3]–[10] on SDN modeling. None of these has considered NFV in their analytical modeling. Two recent works [1], [2] have investigated the performance of SDN and NFV coexisting architecture using simulation and experimentation. However, no analytical model was developed for the SDN architecture combining with NFV. This work is the *first* to model the architecture that combines SDN with NFV, as well as analyzing its performance.

There are different approaches to combining SDN with NFV. Since the controller determines which instance of virtualized network function (VNF) serves the packets which need network functions (NFV packets for short), the NFV packets can be redirected to the controller from the switch. The controller then forwards these NFV packets to the proper VNF for executing the required network functions. That is, the NFV packets will pass through the controller. This approach is called SDN with NFV under the controller (NFV_C). Another approach is that some of the NFV packets are sent to the controller, the controller determines which instance of VNF will serve these packets, and will set in motion the appropriate actions for the switch. Consequent NFV packets belonging to the same flow can be directly redirected to the determined instance of VNF without the controller. We term this approach SDN with NFV aside the controller (NFV_AC).

The main *objective* of this work is to carry about analytical modeling of NFV_C and NFV_AC. We also demonstrate a comparison between the two architectures based on the delay of NFV packets. Our work covers: (i) developing the analytical models for the two architectures, (ii) performing simulation to validate our analytical model, and (iii) performing sensitivity analysis of certain system parameters (such as arrival rates,

978-1-5386-3281-9/17 $31.00 © 2017 IEEE
DOI 10.1109/FiCloudW.2017.76

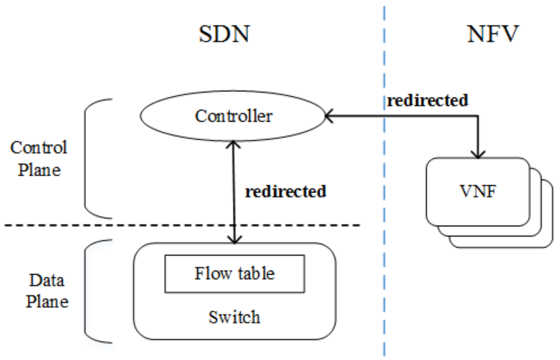

Fig. 1. Traditional SDN architecture where NFV is under controller.

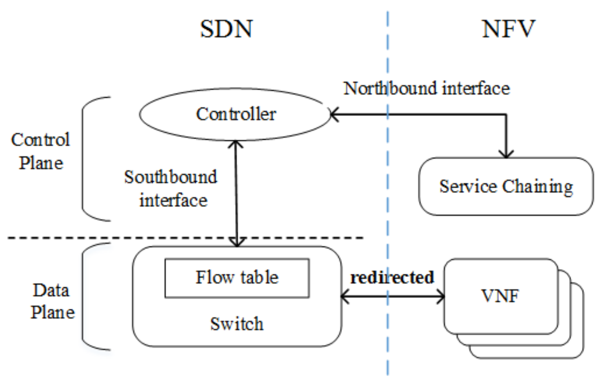

Fig. 2. SDN architecture where NFV is aside controller.

service rates, etc.) on the performance of these two architectures.

The rest of the paper is organized as follows. In Section II, we briefly explain the two SDN architectures along with related works on SDN modeling. In Section III, we present our analytical models for the two SDN architectures and in Section IV, we present the analytical and simulation results. Finally, Section V contains the concluding remarks.

II. Preliminaries

In this section, we briefly explain the two SDN architectures where NFV is under or aside the controller, followed by the current works on SDN modeling.

In the NFV_C architecture shown in Fig. 1, the controller interacts directly with VNFs. The overall procedure is as follows. First, the NFV packet enters the switch. Then, depending on the action in the flow table, the NFV packet is forwarded to the controller. The controller will determine which instance of VNF serves it and forwards it to the selected instance. After the NFV packet receives its required network function, it returns to the controller and the controller sends it back the switch. Since the action in the flow table is redirected to the controller, all NFV packets belonging to the same flow will be sent to the controller. The main advantage of this architecture is that NFV packets can be dispatched to different instances of VNF for load balancing.

In NFV_AC architecture, shown in Fig. 2 [2], the switch interacts directly with VNFs. There is a service chaining module, which selects the proper instances of VNFs and determines the order of chaining. Service chaining module communicates with the controller via northbound interfaces. In this architecture, the controller is responsible for extracting network events, collecting statistics, and analyzing payload for selecting the proper instances of VNFs and their chaining to support NFV. If table miss of a NFV packet happens, the packet visits the service chaining module through the controller. After receiving a response from the service chaining module, the controller sets the proper action into the switch and sends this NFV packet back the switch. This packet, experiencing the service chaining module, still needs to go to VNF from the switch to obtain its required network function. Consequent NFV packets belonging to the same flow can be directly redirected from the switch to determined instances of VNF without the controller. The main advantage of this architecture is that most NFV packets can be directly forwarded to VNF, significantly reducing the controller's loading, However, most NFV packets belonging to the same flow will be forwarded to the same instance. Thus the instances of VNFs are not load balanced, result in a higher delay in providing network functions.

TABLE I
Related Works on SDN Modeling

Paper	Device	#	Methodology	Performance metric
[3]	Controller	1	M/M/1/k	Avg. packet delay
	Switch	1	M/M/1	Avg. packet delay
[4]	Controller	1	Net. Calculus	Buffer size bound
	Switch	N	Net. Calculus	Packet delay bound
[5]	Controller	1	M/M/1	Avg. packet delay
	Switch	1	M/M/1 (adjusted λ)	Avg. packet delay
[6]	Controller	1	M/M/1/k	Avg. packet delay
	Switch	N	M/M/1	Avg. path delay
[7]	Root controller	1	M/M/1	Avg. packet delay
	Local controller	N	M/M/1/k	Avg. packet delay
[8]	Controller	1	M/G/1	Avg. packet delay
	Switch	N	M^X/M/1	Avg. packet delay
[9]	Controller	1	3D state	Avg. packet delay
	Switch	1	(controller, HPQ, LPQ)	Avg. packet delay Packet loss prob.
[10]	Controller	1	MMPP/M/1	Ave. packet delay
	Switch	1	HPQ: MMPP/M/1 LPQ: MMPP/M/1/k	Avg. packet delay Avg. throughput

A. Related works on SDN modeling

There have been a few works on the analytical modeling of SDN. In Table I, we give the key points of such previous works, together with their methodologies and performances matrices.

First, modeling on SDN was carried out in [3], where feedback orientated queueing theory is used to show the interaction between the control plane and the data plane. In [4], network calculus was used to develop an analytical model of a SDN network. This work developed the buffer size bound and packet delay bound.

Mahmood et al. [5] presented an improvement on [3] by modeling SDN as a modified Jackson network. They estimated the packet rate from the controller to the switch, so that overall packet arrival rate into the switch could actually be obtained. They further extended their previous work to propose an analytical model for the SDN having multiple switches [6]. In this model, they calculated the average path delay from the source to the destination, rather than the average packet delay in a switch.

Wang et al. [7] adopted the concept of hierarchical-controller architecture, which has a root controller and some local controllers for improving flexibility of the control plane. With this architecture, they analyzed the average packet delay spent in the controllers. However, this paper did not consider the switches. On the other hand, Xiong et al. [8] thought that the packet arrivals should have the batch characteristics, rather than a Poisson process. Thus, their model is of packet batch arrivals following a Poisson process, and the number of packets in a batch conforming to Poisson distribution. Finally, they model the behaviors of switches and the controller as the queueing systems M^X/M/1 and M/G/1, respectively.

Goto et al. [9] considered the switches to have two queues: a high-priority queue for those packets sent back from the controller and a low-priority queue for newly-arrival packets from other switches. Therefore, those packets coming back from the controller have a higher priority for delivery, in order to reduce their delay. A three-dimensional state (controller's queue length, high-priority queue length, and low-priority queue length) was created to represent this system. The authors derived state transition probabilities and tried to reduce the complexity of obtaining the steady state probability.

Miao et al. [10] considered the realistic nature of multi-media traffic, and used a Markov Modulated Poisson Process (MMPP) to model the burst of packet arrivals. They also adopted two queues: a high-priority queue and a low-priority queue in the switch, which is similar to the work [9]. They solved this problem by using MMPP/M/1 and MMPP/M/1/k for the high-priority queue and low-priority queue, respectively.

However, none of these works considered both SDN and NFV in their models. Therefore, in this paper, we develop analytical models for combining SDN with NFV. To the best knowledge, this work is the *first work* that considers the presence of NFV while modeling the performance of SDN.

III. System Model

We have used classical queueing theory to develop mathematical models for the NFV_C (case I) and NFV_AC (case II). The queueing model of NFV_C is shown in Fig. 3, where it has three M/M/1 queues. For NFV_AC the model has four queues, as shown in Fig. 4.

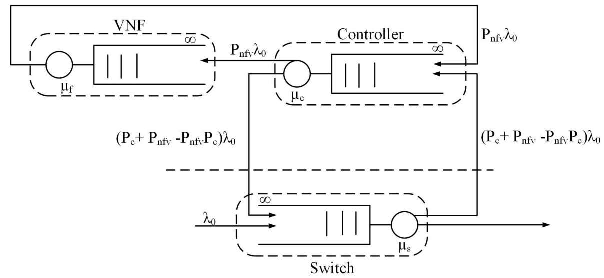

Fig. 3. Case I: queueing model for NFV_C.

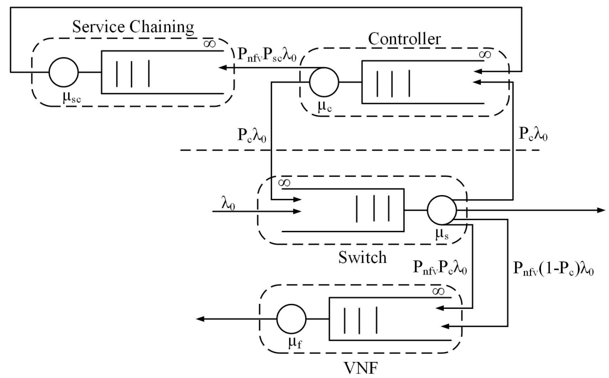

Fig. 4. Case II: queueing model for NFV_AC.

The flow of NFV packets in the two SDN architectures are illustrated through the phase diagrams in Fig. 5 and Fig. 6, respectively. These two figures show the way a NFV packet progresses through the system in two cases. T_s, T_c, T_f and T_{sc} represent the average packet delay at the switch, controller, VNF and service chaining module, respectively.

A. Assumptions and Notations

Following are the assumptions of the model:

- Data arrival process at switch is a Poisson process.
- The service time of packets in switch, controller and VNF are assumed to follow exponential distributions.
- For switch, controller, VNF and service chaining module, the queue size is infinite.

The notations used in the analysis are listed in Table II. To denote different parameters, we have used superscript 1 and 2 for NFV_C and NFV_AC, respectively. Moreover, we have used subscript X to indicate devices where X can be replaced by s, c, f and sc for switch, controller, VNF and service chaining module, respectively.

B. Analysis for NFV_C

To calculate the average packet delay for NFV packets, we first calculate average packet delay in different queues. Fig. 3 shows two types of packets entering the switch: (i) new packets

TABLE II
NOTATIONS USED IN THE ANALYSIS

Symbol	Parameter Name
λ_0	Arrival rate of packets at the switch
$\Lambda_X^{(1)}$	Total arrival rate at X ϵ {s, c, f} for NFV_C
$\Lambda_X^{(2)}$	Total arrival rate at X ϵ {s, c, f, sc} for NFV_AC
P_c	The probability of redirecting to the controller from switch
P_{nfv}	The probability of NFV packets
μ_s	Service rate at switch
μ_c	Service rate at controller
μ_f	Service rate at VNF
μ_{sc}	Service rate at Service chaining module
$T_X^{(1)}$	Average Packet delay at X ϵ {s, c, f} for NFV_C
$T_X^{(2)}$	Average Packet delay at X ϵ {s, c, f, sc} for NFV_AC
$T_{Total}^{(1)}$	Average packet delay of NFV packets for NFV_C
$T_{Total}^{(2)}$	Average packet delay of NFV packets for NFV_AC

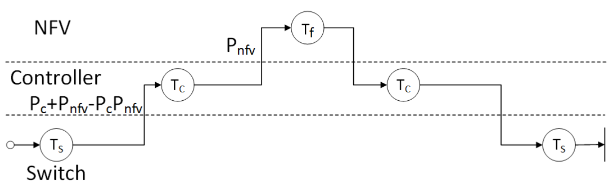

Fig. 5. Phase diagram of delays for NFV packets for NFV_C.

arriving the switch, and (ii) packets that are redirected from the controller go back the switch. For the former, the packet arrival rate is λ_0. For the latter, it can be carefully obtained as follows.

First we can obtain the arrival rate of NFV packets as $P_{nfv}\lambda_0$, because the probability of packets requiring a network function is assumed to be P_{nfv}. Then, the rate of packets sent to the controller from the switch due to table miss is $P_c\lambda_0$. However, these two rates slightly overlap because some NFV packets encounter table miss. Thus, we should subtract the probability of their intersection. Therefore, total arrival rate sent from the switch to the controller is $(P_c + P_{nfv} - P_cP_{nfv})\lambda_0$. This rate is also the rate sent from the controller to the switch because we assume the controller has infinite buffer, i.e., no loss. Finally the total packet rate entering into

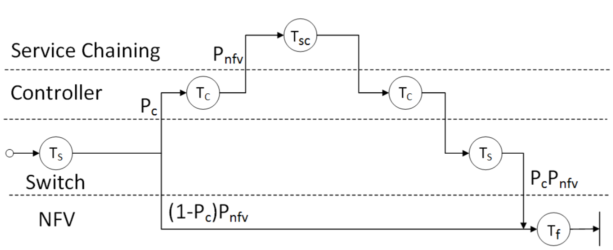

Fig. 6. Phase diagram of delays for NFV packets for NFV_AC.

the switch is expressed as

$$\begin{aligned}\Lambda_s^{(1)} &= \lambda_0 + (P_c + P_{nfv} - P_cP_{nfv})\lambda_0 \\ &= (1 + P_c + P_{nfv} - P_cP_{nfv})\lambda_0.\end{aligned} \quad (1)$$

As switch is a M/M/1 queue, average packet delay at switch can be derived using the total arrival rate at switch, $\Lambda_s^{(1)}$, and service rate at switch, μ_s, as

$$T_s^{(1)} = \frac{1}{\mu_s - (1 + P_c + P_{nfv} - P_cP_{nfv})\lambda_0}. \quad (2)$$

Some of the packets from the controller will enter the VNF module. Hence, total arrival rate at VNF is as

$$\Lambda_f^{(1)} = P_{nfv}\lambda_0. \quad (3)$$

Therefore, packet delay at VNF can be expressed as

$$T_f^{(1)} = \frac{1}{\mu_f - P_{nfv}\lambda_0}. \quad (4)$$

For the controller, packet arrival occurs in two ways. First, some packets from switch enter the controller. Second, all the packets leaving VNF enter the controller queue. The former has a rate $(P_c + P_{nfv} - P_cP_{nfv})\lambda_0$, as described above. The latter is $P_{nfv}\lambda_0$. Combining the two cases, the total arrival rate at controller can be written as

$$\begin{aligned}\Lambda_c^{(1)} &= (P_c + P_{nfv} - P_cP_{nfv})\lambda_0 + P_{nfv}\lambda_0 \\ &= (2P_{nfv} + P_c - P_cP_{nfv})\lambda_0.\end{aligned} \quad (5)$$

As the controller is a M/M/1 queue, packet delay at controller can be calculated as

$$T_c^{(1)} = \frac{1}{\mu_c - (2P_{nfv} + P_c - P_cP_{nfv})\lambda_0}. \quad (6)$$

Since each NFV packet has to visit switch and controller queue twice and VNF module queue once, we can calculate average packet delay for NFV packets using Eqns. (2), (4) and (6) as

$$T_{Total}^{(1)} = 2T_s^{(1)} + 2T_c^{(1)} + T_f^{(1)}. \quad (7)$$

C. Analysis for NFV_AC

From Fig. 4, two types of packets enter the switch: (i) new packets to the switch and (ii) packets that are redirected from the controller. Therefore, total arrival rate at switch is as

$$\begin{aligned}\Lambda_s^{(2)} &= \lambda_0 + P_c\lambda_0 \\ &= (1 + P_c)\lambda_0.\end{aligned} \quad (8)$$

Therefore, average packet delay at switch can be expressed as

$$T_s^{(2)} = \frac{1}{\mu_s - (1 + P_c)\lambda_0}. \quad (9)$$

It is easily observed that the arrival rate of NFV packets, $\Lambda_f^{(2)}$, is as

$$\Lambda_f^{(2)} = P_{nfv}\lambda_0. \quad (10)$$

Therefore, average packet delay at VNF for NFV_AC can be expressed as

$$T_f^{(2)} = \frac{1}{\mu_f - P_{nfv}\lambda_0}. \quad (11)$$

The packets that need processing from the controller have to be sent to it from the switch. Also the packets departing from the service chaining module will enter the controller's queue again. Combining the two cases, the total arrival rate at controller can be written as

$$\begin{aligned}\Lambda_c^{(2)} &= P_c\lambda_0 + P_cP_{nfv}\lambda_0 \\ &= (1 + P_{nfv})P_c\lambda_0.\end{aligned} \quad (12)$$

As the controller is a M/M/1 queue, packet delay at controller can be calculated as

$$T_c^{(2)} = \frac{1}{\mu_c - (1 + P_{nfv})P_c\lambda_0}. \quad (13)$$

In Fig. 4, service chaining module is connected to the controller and NFV packets will visit it depending on the probability P_{nfv}. The arrival rate of the service chaining module can be expressed as

$$\Lambda_{sc}^{(2)} = P_cP_{nfv}\lambda_0. \quad (14)$$

Therefore, average packet delay at service chaining module can be expressed as

$$T_{sc}^{(2)} = \frac{1}{\mu_{sc} - P_cP_{nfv}\lambda_0}. \quad (15)$$

Using Eqns. (9), (11), (13) and (15), we can calculate the average packet delay for NFV packets as

$$\begin{aligned}T_{Total}^{(2)} &= P_c(2T_s^{(2)} + 2T_c^{(2)} + T_{sc}^{(2)} + T_f^{(2)}) \\ &\quad +(1 - P_c)(T_s^{(2)} + T_f^{(2)}) \\ &= (1 + P_c)T_s^{(2)} + 2P_cT_c^{(2)} + P_cT_{sc}^{(2)} + T_f^{(2)}\end{aligned} \quad (16)$$

TABLE III
BASELINE PARAMETERS FOR THE ANALYSIS AND SIMULATION

Parameter Name	Value
Probability of redirecting to controller, P_c	0.04
Probability of redirecting to NFV, P_{nfv}	0.5
Arrival Rate at the Switch, λ_0	75000 pkts/sec
Service rate at switch, μ_s	100000 pkts/sec
Service rate at controller, μ_c	90000 pkts/sec
Service rate at NFV, μ_f	95000 pkts/sec
Service rate at service chaining, μ_{sc}	85000 pkts/sec

IV. RESULTS

In this section, we give the analytical and simulation results for NFV_C and NFV_AC by varying different system parameters, including the probability of redirecting from switch to controller, P_c, service rates of NFV, μ_f, and the probability of NFV packets, P_{nfv}. We have listed all the default setting used for the analysis and simulation in Table III. It was reported in [11] that in a typical OpenFlow network, the probability of new flows is about 4%. The probability of NFV packets is set 50% by default. For simulation, 1.5 million packets are generated for stability.

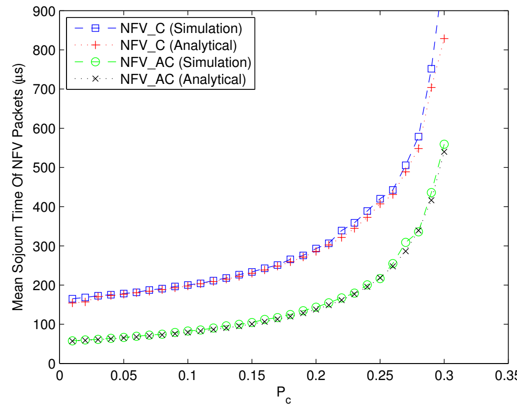

Fig. 7. Average packet delay of NFV packets vs. P_c.

A. Impact of P_c

Fig. 7 shows the impact of P_c on the average packet delay of NFV packets. It is found that the analytical results match the simulation results, irrespective of the architecture: NFV_C or NFV_AC. This means our model and analysis can correctly emulate realistic conditions. Also, we observed that the packet delay increases with the increase in P_c. More packets sent to the controller cause heavier load on the controller, resulting in greater packet delays. Comparing NFV_C with NFV_AC, the packet delay for NFV_AC is much smaller than that for NFV_C, largely three reasons. First, each NFV packet in NFV_C has to enter the controller, and then goes to the VNF, and finally returns to the switch via the controller. This is a longer route for NFV packets in NFV_C, compared to that in NFV_AC. Second, more packets enters the controller in NFV_C than in NFV_AC. We can see that the arrival rate entering the controller is $(2P_{nfv} + P_c - P_cP_{nfv})\lambda_0$, which is much larger than $(1 + P_{nfv})P_c\lambda_0$ in NFV_AC. The heavier load the controller has, longer the delay packets experience in the controller. Third, more packets sent to the controller from the switch represent more packets sent back to the switch. Therefore, the arrival rate in switch for NFV_C is also larger than that for NFV_AC. We can easily know this from Eqs. (1) and (8). The arrival rate in switch for NFV_C is $(1 + P_c + P_{nfv} - P_cP_{nfv})\lambda_0$, which is larger than that for NFC_AC, $(1 + P_c)\lambda_0$.

B. Impact of Service rate at NFV, μ_f

Fig. 8 shows the impact of service rate at NFV, μ_f, on the average packet delay of NFV packets. For both cases, the analytical results very match the simulation results. It is found that NFV_AC performs better than NFV_C because of three reasons described above. The delay gap between NFV_C and NFV_AC is constant because μ_f only affects the packet

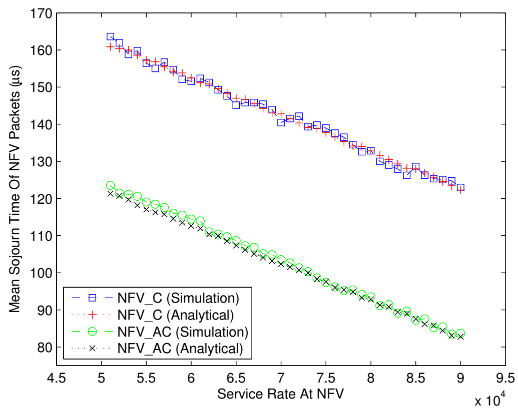

Fig. 8. Average packet delay of NFV packets vs. μ_f.

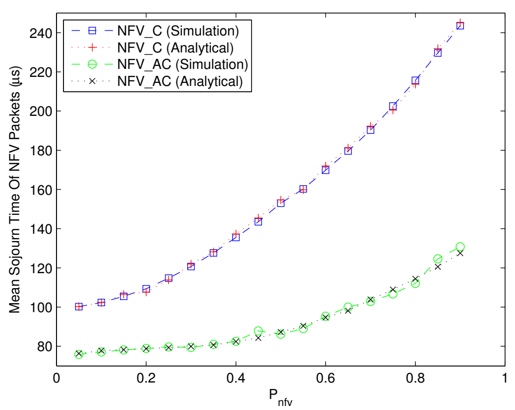

Fig. 9. Average packet delay of NFV packets vs. P_{nfv}.

delay in the VNF module. However, the packet arrival rates for NFV_C and NFV_AC are the same, meaning that their packet delay in VNF are also the same for both no matter which value of μ_f. This can be proved from Eqs. (4) and (11) because they have the same formula. The gap is caused by the packet delay differences in the controller, switch, and the service chaining model, but it is not affected by μ_f.

C. Impact of P_{nfv}

Fig. 9 shows the impact of P_{nfv} on the average packet delay of the NFV packets. It is found that the analytical result matches the simulation results, verifying the correctness of our analysis. Also we can observe that the average packet delay increases with the increase of P_{nfv}. As described above, μ_f will not affect the gap between NFV_C and NFV_AC. However, P_{nfv} actually affects this gap because a larger P_{nfv} means more packets sent to the controller, especially for NFV_C. That causes that the packet delay for NFV_C increase faster than that for NFV_AC.

V. Conclusion and Future works

In this paper, we have presented models for two SDN architectures combined with NFV. The analysis for the packet delay of NFV packets is derived using an M/M/1 model. Extensive simulations were conducted to verify our analysis. Results show that analytical results very closely match simulation results, supporting the correctness of our model and analysis. The packet delay for NFV_AC is significantly less than that for NFV_C. We recorded a significant delay reduction of 54.67% for NFV_AC compared to NFV_C, for three reasons: a shorter route, a smaller controller load, and a smaller switch load.

The service rate at VNF, μ_f, does not affect the delay gap between NFV_AC and NFV_C because the packet delay in VNF are the same for both. On the other hand, the probability exists that NFV packets, P_{nfv}, will have some effects. The larger P_{nfv} will cause a larger gap on packet delay between NFV_AC and NFV_C because the switch and the controller become more congested for NFV_C.

Although NFV_AC is obviously better than NFV_C in our model, our model assumes VNF has the same capacity to serve the packets. In a real environment, NFV_C has more flexibility to select a lightly-loaded instance to serve NFV packets to reduce packet delay in VNF. This scenario will be investigated in the future.

References

[1] P.-C. Lin, Y.-D. Lin, C.-Y. Wu, Y.-C. Lai, and Y.-C. Kao, "Balanced service chaining with traffic steering in software defined networks with network function virtualization," *IEEE Computer*, vol. 49, no. 11, pp. 68–76, November 2016.

[2] Y.-D. Lin, P.-C. Lin, C.-H. Yeh, Y.-C. Wang, and Y.-C. Lai, "An extended sdn architecture for network function virtualization with a case study on intrusion prevention," *IEEE Network*, vol. 29, no. 3, pp. 48–53, May/June 2015.

[3] M. Jarschel, S. Oechsner, D. Schlosser, R. Pries, S. Goll, and P. Tran-Gia, "Modeling and performance evaluation of an OpenFlow architecture," in *International teletraffic congress*, 2011, pp. 1–7.

[4] S. Azodolmolky, R. Nejabati, M. Pazouki, P. Wieder, R. Yahyapour, and D. Simeonidou, "An analytical model for software defined networking: A network calculus-based approach," in *IEEE Global Communications Conference (GLOBECOM)*, Atlanta, GA, Dec 9-13 2013, pp. 1397–1402.

[5] K. Mahmood, A. Chilwan, O. N. Østerbø, and M. Jarschel, "On the modeling of openflow-based sdns: The single node case," *arXiv preprint arXiv:1411.4733*, 2014.

[6] ——, "Modelling of OpenFlow-based software-defined networks: the multiple node case," *IET Networks*, vol. 4, no. 5, pp. 278–284, 2015.

[7] G. Wang, J. Li, and X. Chang, "Modeling and performance analysis of the multiple controllers' approach in software defined networking," in *International Symposium on Quality of Service*, Portland, OR, June 15-16, 2015, pp. 73–74.

[8] B. Xiong, K. Yang, J. Zhao, WeiLi, and K. Li, "Performance evaluation of openflow-based software-defined networks based on queueing model," *Computer Networks*, vol. 102, pp. 172–185, 2016.

[9] Y. Goto, H. Masuyama, B. Ng, W. K. Seah, and Y. Takahashi, "Queueing analysis of software defined network with realistic openflow–based switch model," in *IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS)*, London, UK, September 19-21, 2016, pp. 301–306.

[10] W. Miao, G. Min, Y. Wu, H. Wang, and J. Hu, "Performance modelling and analysis of software-defined networking under bursty multimedia traffic," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 12, no. 5s, pp. 77:1–19, 2016.

[11] J. R. Jackson, "Networks of waiting lines," *Operations research*, vol. 5, no. 4, pp. 518–521, 1957.