

## A DELAY MODELLING IN SDN-BASED MOBILE NETWORKS WITH SINGLE CONTROLLER

**Strahil Panev, Pero Latkoski**

*Faculty of Electrical Engineering and Information Technologies,  
"Ss. Cyril and Methodius" University in Skopje,  
Rugjer Bošković bb, P.O. box 574, 1001 Skopje, North Macedonia  
strahil22@gmail.com*

**Abstract:** The Software-Defined Networking (SDN) is a concept that is already widely used in data centers and mobile core networks to provide network-wide abstraction, open interfaces, programmability, and hiding of network complexities. In mobile networks, a traditional challenge is the mobility management and providing the users a seamless experience with minimal interruption when performing a handover. In this work we are interested in performance analysis of OpenFlow-based SDN mobile networks by using analytical modelling based on queueing theory. We analyze the packet sojourn time of both the switch and the controller and the impact of key factors, such as the number of switches, probability of packet-in messages, arrival rate, and total system throughput. Our findings can be used to quantify the handover delay due to OF-signaling messages that the Mobile Node (MN) experiences when performing a “hard” handover.

**Key words:** queueing model; SDN; performance evaluation; mobile networks; mobility management

### МОДЕЛИРАЊЕ НА ДОЦНЕЊЕТО ВО МОБИЛНИТЕ МРЕЖИ БАЗИРАНИ НА SDN СО ЕДЕН КОНТРОЛЕР

**Апстракт:** Софтверски дефинираните мрежи (SDN) денес масовно се користат во податочните центри и во мобилните јадрени мрежи каде што обезбедуваат мрежна апстракција, отворени интерфејси, програмабилност и криење на мрежната комплексност. Менаџирањето со мобилноста на корисниците во целуларните мобилни мрежи е посебен предизвик за операторите. Целта е гарантирање на минимален прекин кога корисникот извршува процедура на хендовер. Во овој труд ги анализираме перформансите на мобилните софтверски дефинирани мрежи SDN базирани на OpenFlow преку аналитичко моделирање со користење на теоријата на редови. Го разгледуваме времето на процесирање на пакети и влијанието на клучните фактори како што се: број на комутатори, веројатност на пораки *Packet-In*, брзина на пристигнување на пакети и вкупен проток низ системот. Нашите заклучоци можат да се искористат за квантификација на доцнењето кое го искусува мобилната станица како резултат на OpenFlow-сигнализацијата во рамките на процедурата хендовер.

**Клучни зборови:** теорија на редови; софтверски дефинирани мрежи; евалуација на перформанси; мобилни мрежи; менаџирање на мобилност

### INTRODUCTION

The Software Defined Networking paradigm has been introduced to tackle some of the challenges that the mobile operators are experiencing, such as: inflexibility to network changing conditions, vendor lock-in and use of proprietary protocols, exaggerated network architectures as the traffic demand increases and new applications are deployed, high up-

grade costs, and inefficient use of network resources. It is an emerging technology that fully separates control and user plane and enables network abstraction and programmability via open standard protocols. The user plane usually consists of many low-cost and simple devices, while the control plane consists of single or multiple SDN controllers which provide a centralized network intelligence and a global state management. The SDN interfaces are shown on Figure 1.

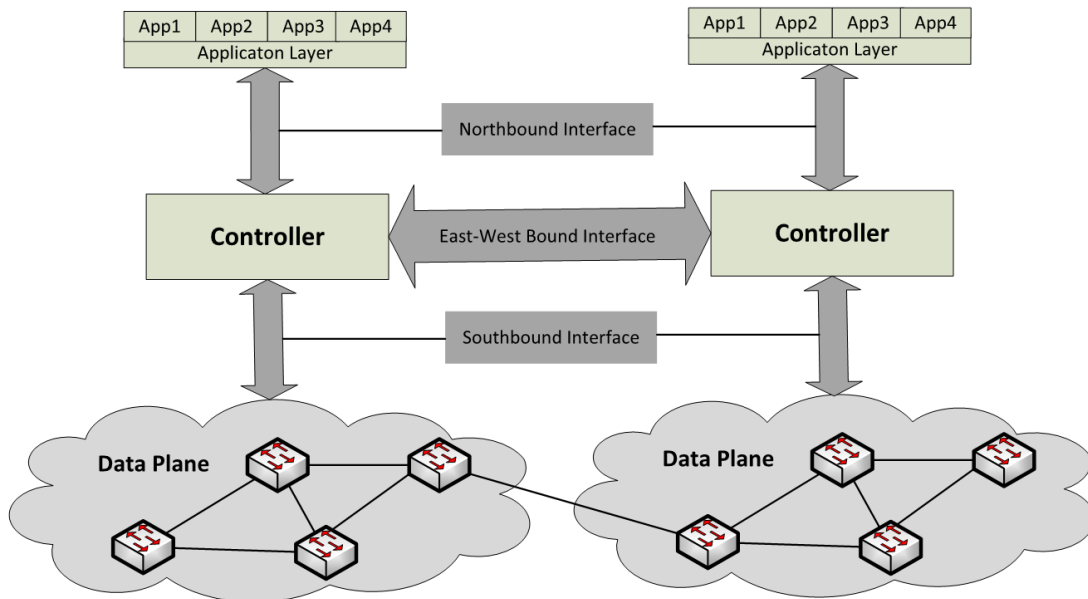


Fig. 1. SDN interfaces

The Northbound Application Programming Interface (API) defines the orchestration of the user plane by the applications and the execution of typical tasks such as load balancing, firewall, policy enforcement etc. The east-west API is used for exchanging controller synchronization and invocation messages that are needed when installing new flows or topology updates in the user plane. The southbound protocols define a set of communication rules for exchanging control information between the controllers and the user plane switches. The most popular implementation of the Southbound API is OpenFlow (OF).

Today OF is the most popular and the only open protocol that is used in SDN for the communication between the switch and the controller. The basic elements of an OF switch are one or more *flow tables* that are used to perform packet look-up and forwarding. Each *flow table* contains *flow entries* that are added, updated or deleted at initialization or in case of network changes. A flow entry comprises of *match fields*, *counters*, and a *set of instructions* that are executed if an incoming packet is matched on the defined header fields. The controller-to-switch communication usually consists of the following: if a new flow comes at the switch interface and it cannot be matched to a flow entry, then the following actions are executed: (i) the first packet of the flow is sent to the controller, (ii) the controller decides for the best optimal path and updates the flow tables of all the switches on the forwarding path, (iii) all the packets of that flow are directly forwarded by the switches based on the already sent

forwarding rules and there is no need for additional control plane action. In this paper we are interested in the OF-signaling related delay that the MN is experiencing when roaming in an OF-based mobile network.

A major challenge in today's mobile/cellular networks is ensuring the continuity of the user's ongoing session and providing minimal interruption time during the procedure of handover. In the case of SDN-based core networks, a successful handover procedure consists of exchanging OF signaling messages between the control and data plane. The specific case of "hard" handover involves a breakage of the ongoing session and involves reconfiguration and management messages exchanged between the switch and controller. In our work, we concentrate on the handover of hosts between different switches, and our aim is to model the handover delay introduced by the OF control messages by using a new mathematical model.

Analytical models based on queuing theory have been extensively used to provide performance modelling of systems based on OF architecture. By analyzing the desired parameters of the OF architecture under specific conditions, novel algorithms and valuable conclusions for the network performance can be drawn. One of the first OF-based analytical models was introduced by Jarschel et al, in [1]. The authors use a feedback-oriented queuing model with a forwarding queue system (switch), and a feedback queue system (controller). They use M/M queue with infinite buffer size for the switch, and M/M-S

finite size queue for the feedback queue. However, they only model a single switch communicating with a single controller. Azodolmolky et al. [2] propose an analytical model based on network calculus theory, as an addition to using queuing theory for SDN modelling. The authors are interested in evaluating the switch and the controller for queue length and delay and manage to provide the upper limit of the packet processing delay of the switch, and the upper limit of the controller queue length. The shortcoming in their work is the fact that they use a single node model. Thieme in [3] investigates different aspects and challenges of using queuing theory for performance analysis of SDN. He is mainly interested in the latencies that occur when moving data between different caches, SW/HW triggered interruptions, design of interfaces as queues etc. An important subject of analysis is the finite capacity of the system and the limitation on the buffer size, however, as the author explains, most if not all recent work has completely omitted the last two assumptions and considered them both as infinite.

The work by Mahmood et al. [4] is of great importance as it considers for the first time the multiple-node case, meaning multiple switches constituting the data plane. Their work is extended in [5] with modelling of *Packet-In* messages. The authors propose an analytical approach to quantify the controller's sojourn time and the solidity of the packets pumped into the network. They do not analyze the propagation delay, but their model considers both finite and infinite buffers. Shang et al. [6] are also interested in the *Packet-In* messages and how they impact the performance of the controller's sojourn time. They use M/H2 (2-phase hyperexponential distribution) queue to model the switches, while modelling the controller as M/M. The authors do not provide extensive model validation, but their numerical results clearly point that if the probability of the *Packet-In* messages is high, the performance of the network degrades rapidly. Sarkar et al. [7] provide a new analytical model based on queuing theory and probability distribution function. They use M/M/1-N feedback queue to increase the approximation of the actual controller performance. The model analyzes the packet waiting time of an OF-switch as well as the probability to dropped packets under high load. The authors in [8] investigate how to reduce the packet sojourn time by deploying multiple controllers. Their main aim is to evaluate the flow setup time and optimize the number of the controllers. An analytical model is proposed and a prototype of multiple controllers is implemented. By

measuring the packet service time, the authors fit a hyper-Erlang distribution to the response time, which they later use in their mathematical model to determine the optimal number of controllers.

In [9] authors investigate the key factors that impact the overall latency in SDN-based mobile core networks. They argue that latency is contributed mainly by two factors: processing delay and transmission delay. Processing delay refers to the packet sojourn time within the switch and the controller, while transmission delay describes the time needed for the data to be transmitted between the switches and the controller. The authors assess the impact of the number of hops, load of the controller, and the buffer handling. Tantayakul et al. [10] investigate the impact of SDN on the mobility management and how mobile devices can move and use the SDN network. They propose a solution based on OF protocol that has several advantages: it is a mobility solution without participation of the mobile node, it is easy to be implemented with significant ease of use, avoids the need of tunneling and tunneling packet overhead, and minimizes the packet loss and the handover delay. Alotaibi et al. [11] are interested in quantifying the handover delay due to OF-signaling message exchange. They model both *Packet-In* and *Port-Status* messages and define a proper performance metric to evaluate the total delay the MN experiences in case of a "hard handover". Furthermore, they propose an LTE architecture and compare it via simulations with the existing solutions. The major drawback in their proposal is the fact that in their performance evaluation, they present findings obtained via simulations in relation to the proposed distributed LTE architecture, and their focus is not to extensively validate the proposed analytical model. The major contribution of the authors in [12] is to open a discussion on how the SDN technology can be efficiently used in mobile networks for mobility management in future mobile networks (5G) and evolved LTE. They analyze the most important factors related to mobility management, such as preservation of session continuity and scalability of handovers in dense mobile networks. The presented concepts of mobility management can be used to incorporate the functionality of LTE Self-Organizing Network (SON), but to speed up the deployment of the proposed concepts, additional SDN protocols need to be defined.

In this paper we propose an analytical model for SDN-based mobile networks with a single controller. We put our focus on the handover of hosts between different switches, and we try to quantify

the handover latency by investigating the delay caused by the exchange of OF controller messages.

The main contributions of the work presented here are summarized as follows.

1) We propose an analytical approach to model the performance of the switches and the controller based on queuing theory. The switches and the controllers are modelled by using M/M/1 queue, and a suitable performance metric for assessing the handover delay is proposed. We model only the *Packet-In* messages as we assume that this is the dominant type of exchanged messages. We assume that as MN performs a “hard” handover, and the first messages exchanged in the control plane are the *Packet-In* messages.

2) Based on the proposed mathematical model, we provide numerical results that are obtained with *Wolfram Mathematica 12.0*. The findings can be used when designing a handover delay guarantees of a mobile network for a given amount of traffic directed towards the controller. In addition, we analyze the key factors, such as the impact of the number of switches, arrival rate at the switches, probability of *Packet-in* messages, and total system throughput. The obtained results prove that the above performance metrics are critical factors when modelling SDN-based networks.

The remainder of this paper is organized as follows: In Section 2, we describe our network model and assumptions, and we elaborate on the proposed analytical model. Section 3 gives the numerical results, and we state our conclusions in Section 4.

## 2. ANALYTICAL MODEL

We first start with describing our system and the assumptions used. We continue with modeling the switch and the controller with aim to find out the relation between controller’s packet sojourn time and the total system throughput. Finally, we define the performance metric that will allow us to quantify the handover delay. The parameter of interest is the controller’s packet sojourn time, and this is the total time a packet spends at the controller, including the time it is being serviced.

### 2.1. System description and assumptions

In this section, we describe the scenario of interest and the assumptions used. We are interested in modelling the controller’s packet sojourn time in a network that consists of  $n$  switches, controlled by a single controller. In general, when a packet arrives

at the interface of the switch, it is matched to an existing *flow entry*, if no match is found then a *Packet-In* messages is sent to the controller. Similarly, when MN connects or disconnects to a switch, it sends a *Packet-In* message. OF includes other asynchronous messages, such as *Port-status*, *Flow-mod/removed*, and *Error*, but in this work we are only interested in the *Packet-In* messages as those are the dominant type of messages in a mobile network.

We model both the switches and the controller as a single queue and not per interface, and we suppose that all switches have the same service rate. We assume that both switches and controller have infinite buffer sizes and the service rates are load independent. The packet arrivals and service times are modelled using a Poisson distribution which allows us to model the switches and the controller as M/M/1 system. The destinations of the packet flows are uniformly distributed, and we have an open queuing network, where packets can enter and depart the system from any node. The service discipline is FIFO. We assume that at each *Packet-In* message only the header of the first packet is sent to the controller, and after that all subsequent packets are forwarded directly without controller intervention. We also take the assumption that the controller only responds with a *Flow-mod* (install a new flow) message to the switch. Finally, we suppose that the controller has full knowledge of the network topology and always configures the switches in the most optimal way.

### 2.2. Modelling the switches and controller

We model only *Packet-in* messages. We analyze the case when MN disconnects from one switch, say  $s_i$ , and it attaches to another switch, say  $s_j$ , where  $i, j \in \{1, 2, \dots, n\}$ , and  $n$  is the number of switches in the domain. This triggers two *Packet-In* messages sent to the controller to update it on MN’s mobility. When receiving the above messages, the controller sends *Flow-mod* to some of the switches to change the flow entries for the specific MN. The exchange of messages initiated by the mobility of the node is presented in Figure 2, where “(1)” denotes the messages initiated by the switch  $i$ , and “(2)” the messages initiated by the switch  $j$ .

We define the general arrival rate of packets at node  $i$  as  $\lambda_i$ ,  $q_i^{pc}$  is the probability of a *Packet-in* message to go to the controller,  $c_{ij}$  is the routing probability between switches  $i$  and  $j$ , and  $b_{ji} \in \{0, 1\}$  is an indicator that describes whether the update from node  $j$  involves node  $i$  in the flow path, or not.

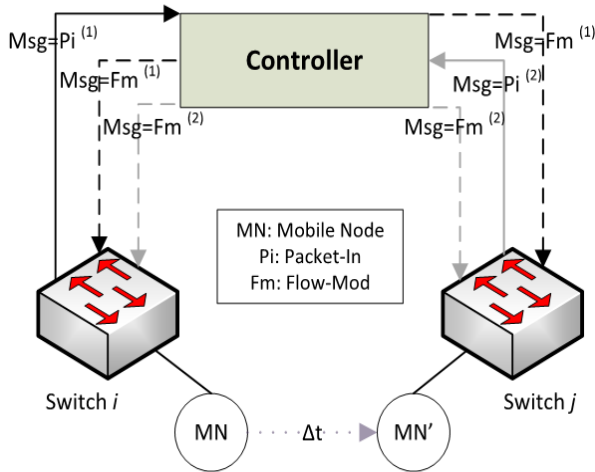


Fig. 2. Mobility triggered exchange of messages

The total arrival rate of packets  $\Lambda_i$  can be presented as

$$\Lambda_i = \lambda_i + \sum_{j=1, j \neq i}^n \lambda_j \times c_{ij} + q_i^{pc} \lambda_i + \sum_{j=1, j \neq i}^n (q_j^{pc} \times b_{ji}) \lambda_j. \quad (1)$$

The service rate of node  $i$  is  $\mu_i$ , and we use the Jackson's theorem for open queuing networks. The load of the switch  $i$  is given as

$$\rho_i = \frac{\Lambda_i}{\mu_i}. \quad (2)$$

We denote the number of packets and the sojourn time at node  $i$  with  $N_i$  and  $T_i$ , respectively. The mean value  $E[.]$  for M/M/1 queuing model is given in [13] as

$$E(N_i) = \frac{\rho_i}{1-\rho_i}. \quad (3)$$

By using the Little's law, for the  $i$ -th switch, the average packet processing time is

$$E(T_i) = \frac{E(N_i)}{\Lambda_i}. \quad (4)$$

If we substitute Eq. (2) and Eq. (3) in Eq. (4), we get

$$E(T_i) = \frac{1}{\mu_i - \Lambda_i}. \quad (5)$$

The average processing time of the switches is given by

$$\begin{aligned} E(T_s) &= \sum_{i=1}^n \frac{\Lambda_i}{\sum_{i=0}^n \Lambda_i} E(T_i) \\ &= \sum_{i=1}^n \frac{\Lambda_i}{\sum_{i=0}^n \Lambda_i} * \frac{1}{\mu_i - \Lambda_i}. \end{aligned} \quad (6)$$

If the packet header can be matched into a flow table, then the switch forwards the packet on the packet flow path, otherwise it must contact the controller. The probability of a *Packet-In* message to be sent to the controller is  $q_i^{pc}$ . The total rate of *Packet-In* messages sent to the controller from  $n$  switches is

$$\Lambda_c = \sum_{i=1}^n q_i^{pc} \lambda_i. \quad (7)$$

As we have stated previously, the *Packet-In* message processing at the controller can be characterized as M/M/1 queuing model. If  $p_i$  is the probability of  $i$  *Packet-in* messages at the controller, then the mean value of these messages is

$$\begin{aligned} E(N_c) &= \sum_{i=0}^{\infty} i p_i = \\ &= \sum_{i=0}^{\infty} i \left(\frac{\Lambda_c}{\mu_c}\right)^i \left(1 - \frac{\Lambda_c}{\mu_c}\right) = \frac{\Lambda_c}{\mu_c - \Lambda_c}. \end{aligned} \quad (8)$$

The packet sojourn time at the controller is

$$E(T_c) = \frac{E(N_c)}{\Lambda_c}, \quad (9)$$

$$E(T_c) = \frac{1}{\mu_c - \Lambda_c}. \quad (10)$$

If we substitute Eq. (7) in Eq. (10), we get

$$E(T_c) = \frac{1}{\mu_c - \sum_{i=1}^n q_i^{pc} \lambda_i}. \quad (11)$$

### 2.3. Performance metric

Our aim is to quantify the handover delay which is a product of exchanging OF signaling messages in SDN-based mobile networks. For a *Packet-In* event, we denote the mean interaction time between switch  $i$  and the controller  $c$  as

$$\begin{aligned} E[T_i^{tot}] &= \\ &= E[T_i] + E[T_c] + E[T_{i_{prop}}^c] + E[T_{i_{prop}}^{c,max}], \end{aligned} \quad (12)$$

where  $T_i$  and  $T_c$  are the respective service time of the switch  $i$  and the controller in the domain,  $T_{i_{prop}}^c$  is the propagation time of the *Packet-In* message from the switch  $i$  to the controller, and  $T_{i_{prop}}^{c,max}$  is the maximum propagation delay (required to reach the

furthest switch) for the *Flow-mod* message sent to the subset of switches as a reaction of the controller to the *Packet-In* message. The complete handover delay  $T_{ho}$  that the MN experiences is sum of the delays introduced by both *Packet-In* message at switch  $i$  and the *Packet-In* message sent by switch  $j$ :

$$E[T_{ho}] = E[T_i^{tot}] + E[T_j^{tot}], \quad (13)$$

$$E[T_{ho}] = 2 \left( \sum_{i=1}^n \frac{\Lambda_i}{\sum_{i=0}^n \Lambda_i} * \frac{1}{\mu_i - \Lambda_i} + \frac{1}{\mu_c - \sum_{i=1}^n q_i^{pc} \lambda_i} \right) + E[T_{prop}^{tot}], \quad (14)$$

where

$$E[T_{prop}^{tot}] = E[T_{i_{prop}}^c] + E[T_{i_{prop}}^{c,max}] + E[T_{j_{prop}}^c] + E[T_{j_{prop}}^{c,max}]. \quad (15)$$

In practice, the processing delay of the switch is much smaller than the processing delay of the controller, as it will be shown in the next section. The propagation time of the messages should be considered for mobile networks due to the large distances.

### 3. NUMERICAL RESULTS

In this section we validate our analytical model by performing numerical analysis. We use *Wolfram Mathematica 12.0* for the graphs. For  $q^{pc}$  we use 4%,  $\mu_s = 9.8 \mu s$ ,  $\mu_c = 240 \mu s$  [1], and we assume that the packets arrive at all switches with the same rate of  $\lambda_i$ . The average distance between the switches and the controller is 100 km (we try to model a small mobile operator), and the link between the switches and the controller is optical, 1 Gbps. We take the number of switches,  $n = 10$ , and the average packet size is 128 B.

Figure 3 shows the packet sojourn time of the switches as a function of  $q^{pc}$ . Value of  $q^{pc} = 0$  means that the switch can forward all the incoming packets without the need of contacting the controller, while value of  $q^{pc} = 1$  means that all the traffic passes through the controller. We notice that the average service time increases as the probability of *Packet-In* messages gets higher. If the packets arrive

at a low rate, the increase in the sojourn time is slower, for  $\lambda_i = 10k$  and  $\lambda_i = 20k$  it is almost a linear increase. However, for an arrival rate of  $\lambda_i = 40k$ , the packet sojourn time of the switch increases exponentially, and becomes very high for higher values of  $q^{pc}$ . We can see that for  $\lambda_i = 40k$ , the increase of the average service time is about 10 times when comparing the case of  $q^{pc} = 0$  to  $q^{pc} = 0.3$ . As it was shown in [1], in a normal network the expected probability of *Packet-In* messages is 0.04. This means that OF should not degrade the packet sojourn time in normal network conditions.

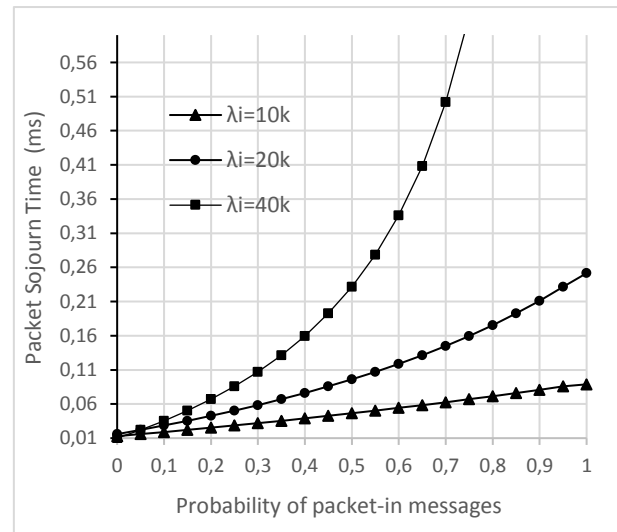


Fig. 3. Average service time of switches (probability of *Packet-In*)

We want to investigate if adding switches to the network will cause degradation of the packet service time of the controller. Adding switches to the network would mean more requests sent to the controller, as the new switches have no flow entries installed. In Figure 4 we analyze the dependency of controller's service time from the number of switches while changing the arrival rate at the switches. For the probability of *Packet-In* messages we use a fix value of 0.04. As the number of switches increases, the controller's service time increases. However, we can notice that the packet sojourn time increases very slowly as the number of switches gets higher. When packets arrive at the rate of 40k per second, the service time of a controller connecting to 100 switches is only about 0.022 ms higher than it is when connecting to 10 switches. Additionally, we notice that the packet sojourn time degrades faster as the packet arrival rate increases.

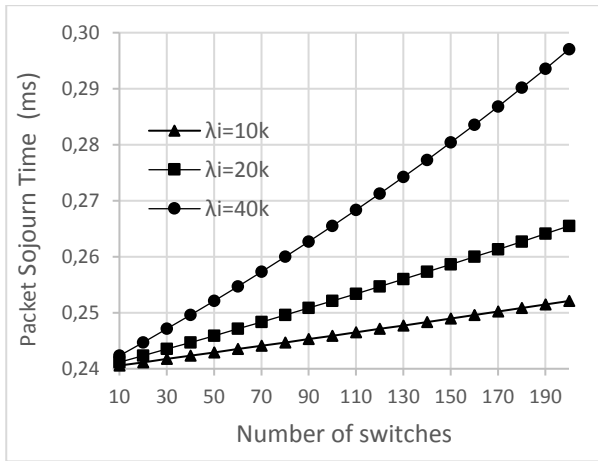


Fig. 4. Average service time of controller (number of switches)

Figure 5 shows the packet sojourn time of the controller and the dependency of the probability of *Packet-In* messages. We take that the number of switches is 10 and all of them receive  $\lambda_i$  packets per second. The way  $q^{pc}$  influences the performance is very similar to the way it influences the switches. When the packets arrive at the slower rate ( $\lambda_i = 10$  k and  $\lambda_i = 20$  k), the probability of *Packet-In* messages is not a factor that impacts the performance of the network. However, when the packets arrive at higher rate ( $\lambda_i = 40$  k), then the packet sojourn time increases rapidly for higher values of  $q^{pc}$ .

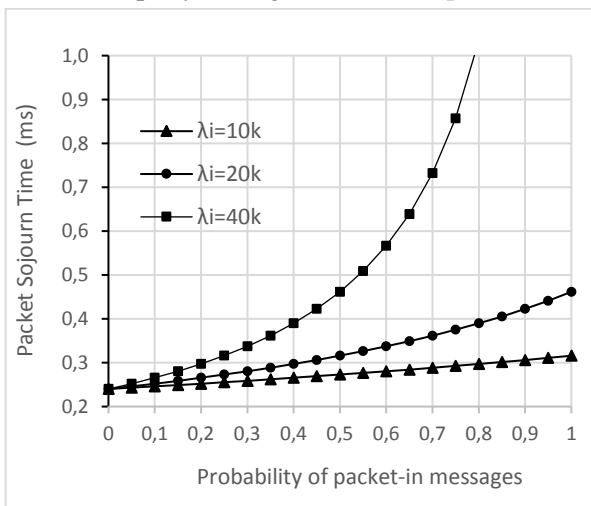


Fig. 5. Average service time of controller (probability of *Packet-In*)

We also want to investigate the impact of the controller’s mean service time to the packet sojourn time, while keeping  $q^{pc} = 0.04$ . The results are shown on Figure 6. As expected, for higher values of  $\mu_c$  the packet sojourn time increases. This plot can be used for setting guarantees on the packet sojourn time when designing a network with controller that

has a fixed average service time and having a limited load on the controller. At controller load around 0.7 to 0.8, the packet sojourn time increases sharply, and becomes unacceptably high for the higher range of load values.

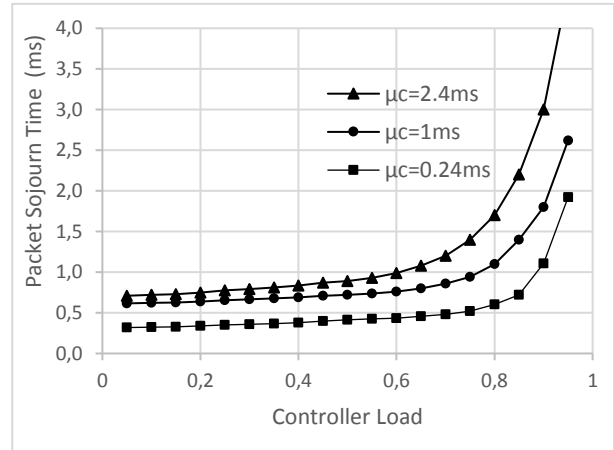


Fig. 6. Packet sojourn time (load), effect of  $\mu_c$  for a fixed value of  $q^{pc}$

Finally, Figure 7 shows the total handover delay as a function of the system throughput that the MN experiences when performing a “hard” handover. System throughput is the total traffic handled by the network. The parameter  $q^{pc}$  has three different values, hence three curves. The plot shows that for  $q^{pc} = 0.1$  and  $q^{pc} = 0.4$ , initially the handover latency increases almost linearly as the system throughput gets higher.

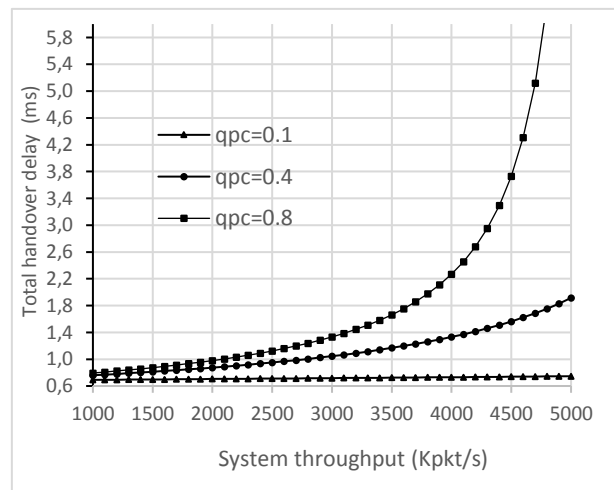


Fig. 7. Total handover delay vs. system throughput

However, for  $q^{pc} = 0.8$  the handover latency at first increases moderately up to system throughput of 3500  $K_{pkt}/s$ , and after that increases sharply as the throughput has negligible increase. This analysis

can be useful when designing the handover delay guarantees of a mobile network for a given amount of traffic that is directed towards the controller. E.g., for  $q^{pc} = 0.4$  and network throughput of  $4250 K_{pkt}/s$ , the handover delay due to OF signaling is 1.45 ms.

#### 4. DISCUSSION OF MODEL'S ASSUMPTIONS

With aim to obtain closed-form equations, we made several major model assumptions in Section 2.1, such as: unlimited buffer, modelling the switch/controller as a single queue, Poisson arrival rate, and switch/controller's service rates are load-independent. In this section we discuss the impact of each of the limitations on the model accuracy.

A real implementation OF-switch and controller has finite buffers, and additionally has multiple interfaces. In our extensions of this work, we explicitly evaluate the effects of the limited buffer on the accuracy of the model. For analyzing the effects of limited buffer, we provide analytical model by using M/M/1/K queue and we compare with the standard modelling as described in this paper. Additionally, we perform simulations in MATLAB to further investigate the effect. Our results lead to conclusion that the limited buffer has an impactful effect on the performance, especially at high loads of the controller/switch. However, the assumption of modelling the switch as a single queue (as opposed to multiple interfaces), does not have a high impact on the average packet service time.

The assumption in the proposed model for load independency of the mean service rate has a more predictable effect on the accuracy of the modelling. The average processing time in the real implementations of OF-switch and OF-controller tends to deteriorate at higher loads. In our next work, we evaluate the effect of this hypothesis on the accuracy of the modelling. We use M/M/1 queue with state dependent service, which means that we consider a system in which the server works with mean service rate  $\mu_1$  until there are  $l$  packets in the system, at which point it changes to a different rate  $\mu_2$ . Our results show that there is an impact on the average packet service time, especially as the packet arrival rate increases. However, the effect of load dependent service rate can be effectively mitigated with careful SDN design considerations, by selecting a controller/switch with sufficiently high mean service rate

In the analytical models proposed to study the OF networks, the most frequent assumption consid-

ered is the Poisson arrival rate. In most of the references that we analyzed, this assumption was applied (even in research work within the current year), however we know for a fact that the real network traffic exhibits self-similar characteristics. The self-similar traffic is a widely spread traffic phenomenon in modern networks, and its basic characteristic is: segments of the process have the very same statistical properties at different scales. The important definitions and properties of self-similarity are given by Leland et.al. [14], where they define an important parameter to quantify the degree of self-similarity, called the Hurst parameter. The simulation of self-similarity has also attracted the attention from the research community, in [15] the authors compare several traffic generation models implemented in MATLAB. In our ongoing research, we model the switch using GI/M/1/K queue to represent independent arrivals with general distribution. The reason of using this type of distribution is that the standard M/M/1 expression for the average queue length and average delay does not hold for switches modelled with finite capacity (GI/M/1/K) queue because the traffic from external arrivals mix with packets fed back from the controller resulting in the aggregate process losing its Markovian property.

#### 5. CONCLUSION

We propose an analytical model that can be used for performance modelling of OF-based SDN mobile core networks. By using queuing theory, we try to analyze the switch and controller's packet sojourn time and we are interested to quantify the total handover delay as a function of the network throughput. The numerical results show that the packet arrival rate, controller load, and number of switches are critical factors when modelling SDN-based networks. Additionally, a high probability of *Packet-In* messages impacts negatively the packet service time and degrades the network performance. For future work, we plan to incorporate the influence to the packet processing time of the synchronization and invocation messages that the controllers exchange in a distributed control plane scenario.

#### REFERENCES

- [1] Jarschel, M., Oechsner, S., Schlosser, D., Pries, R., Goll, S., Tran-Gia, P.: Modeling and performance evaluation of an OpenFlow architecture, *Proceedings of the 23rd International Teletraffic Congress*, 2011, pp. 1–7.



- [2] Azodolmolky, S, Nejabati, R., M. Pazouki, M., Wieder, P., Yahyapour, R., Dimitra, S.: An analytical model for software defined networking: A network calculus based approach, *Global Communications Conference*, 2013, pp. 1397–1402.
- [3] Thieme, C.: Challenges for modelling of software-based packet processing in commodity hardware using queuing theory, *Network Architecture and Services*, Vol. **49**, pp. 49–54 (2017).
- [4] Mahmood, K., Chilwan, A., Osterbo, O., Jarschel, M.: Modelling of OpenFlow-based software-defined networks: The multiple node case, *IET Networks*, Vol. **4**, Iss. 5, pp. 278–284 (2015).
- [5] Mahmood, K., Chilwan, A., Østerbø, O., Jarschel, M.: On the modeling of OpenFlow-based SDNs: the single node case, *Computer Science and Information Technology (CS & IT)*, Vol. **4**, pp. 207–214 (2014).
- [6] Shang, Z., Wolter, K.: Delay evaluation of OpenFlow network based on queuing model, *12th European Dependable Computing Conference (EDCC 2016)*, 2016.
- [7] Sarkar, C., Setua, S.: Analytical model for OpenFlow-based software-defined network, *Progress in Computer, Analytics and Networking*, 2018, pp. 583–592.
- [8] Zhihao, S., Wu, H., Wolter, K.: Performance Evaluation of the Control Plane in Software Defined Networks, *The 12th EAI Int. Conf*, 2019.
- [9] Marquezan, C., An, X., Despotovic, Z., Khalili, R., Hecker, A.: Identifying latency factors in SDN-based moile core networks, *Symposium on Computers and Communication*, 2016, pp. 484–491.
- [10] Tantayakul, K., Dhaou, R., Paillassa, B.: Impact of SDN on mobility management, *30th International Advanced Information Networking and Applications Conference*, pp. 260–265, 2016.
- [11] Alotaibi, M., Helmy, A., Nayak, A.: Modeling Handover Signaling Messages in OpenFlow-Based Mobile Software-Defined Network, *Journal of Computer Networks and Communications*, Vol. **2018**, month 12, pp. 1–14.
- [12] Kuklinski, S., Li, Y., Dinh, K.: Handover management in SDN-based mobile networks, *Global Communications Conference Workshops*, 2014, pp. 194–200.
- [13] Kleinrock, L.: *Queueing Systems*, Vol. **1: Theory**, Wiley Interscience, 1975.
- [14] Leland, W. E., Taqqu, M. S., Willinger, W., Wilson, D. V.: On the self-similar nature of Ethernet traffic (extended version), *Networking, IEEE/ACM Transactions on*, 1994, pp. 1–15.
- [15] Tomić, I., Maletić, N.: Comparison of Models for Self-Similar Network Traffic Generation, *X Int. Symposium on Industrial Electronics (INDEL)*, 2014.

