

# Traversing Virtual Network Functions from the Edge to the Core: An End-to-End Performance Analysis

Emmanouil Fountoulakis\*, Qi Liao<sup>†</sup>, Manuel Stein<sup>†</sup>, Nikolaos Pappas\*

\*Department of Science and Technology, Linköping University, Sweden

<sup>†</sup>Nokia Bell Labs, Stuttgart, Germany

E-mails: {emmanouil.fountoulakis, nikolaos.pappas}@liu.se, {qi.liao, manuel.stein}@nokia-bell-labs.com

**Abstract**—Future mobile networks supporting Internet of Things are expected to provide both high throughput and low latency to user-specific services. One way to overcome this challenge is to adopt network function virtualization and Multi-access Edge Computing (MEC). In this paper, we analyze an end-to-end communications system that consists of both MEC servers and a server at the core network hosting different types of virtual network functions. We develop a queueing model for the performance analysis of the system consisting of both processing and transmission flows. We provide analytical approximations of the performance metrics such as system drop rate and average number of tasks in the system. Simulation results show that our approximations perform quite well. By evaluating the system under different scenarios, we provide insights for the decision making on traffic flow control and its impact on critical performance metrics.

## I. INTRODUCTION

In future communications systems, mission-critical mobile applications, e.g., augmented reality, connected vehicles, eHealth, will provide services that require ultra-low latency [1], [2]. To satisfy the low latency requirements, Multi-access Edge Computing (MEC) has been proposed as a key solution [1]. The idea of MEC is to locate more computational resources closer to the users, e.g., at the base stations. Besides latency constraints, these services may have strict function chaining requirements. In other words, each service has to be processed by a set of network functions (e.g., firewalls, transcoders, load balancers, etc.) in a specific order [3]. Furthermore, the requirements of 5G networks for flexibility and elasticity of the network inspire the idea of Network Function Virtualization (NFV) [3], [4]. The idea of NFV is to decouple the network functions from dedicated hardware equipment. Instead of dedicated hardware equipment, general purpose servers can host one or more types of network functions. However, the computational capabilities and the available resources of MEC servers are still limited compared to the high-end servers in the cloud. Therefore, it is interesting to further investigate the cooperation between the edge and the core, and the cooperation among MEC servers.

Recently, Virtual Network Function (VNF) placement and resource allocation problem has attracted a lot of attention, e.g., [5], [6]. In these works, the authors formulate the VNF placement problem as mixed integer linear problem under the

assumption of known traffic demand. In a dynamic environment with unknown traffic, the authors in [7] develop dynamic algorithms in order to control the flow by applying Lyapunov optimization theory. There are few works on analyzing networks and deriving key performance metrics such as delay. Authors in [8] analyze the end-to-end delay for embedded VNF chains. They consider two types of services that traverse different VNF chains and provide the delay analysis for each chain. However, this work considers a specific system model where multiple VNF chains embedded on a common determined network path, while routing and flow control are not considered. Furthermore, the authors in [9] and [10] estimate the end-to-end delay in Software Defined Network (SDN) environment by using local node measurements for single flow and multi-flow cases, respectively.

In this paper, we model and analyze an end-to-end communications system consisting of two MEC servers at the edge network and one at the core network hosting different types of VNFs by applying tools from queueing theory. In order to simplify the analysis, we introduce the approach of decomposing the system into subsystems, which can be further applied in analyzing scale-up system with arbitrary number of servers. We provide analytical expressions for the key performance metrics such as average number of tasks in the system and system drop rate for each subsystem. Simulation results validate our analysis and show that our analytical model is accurate. Furthermore, by evaluating the system under different scenarios we provide insights of how the routing decision affects the key performance metrics of our interest.

## II. SYSTEM MODEL

We consider an end-to-end communications system consisting of a mobile device, two MEC servers, and one server located in the core network as depicted in Fig. 1. A task traverses a service chain of two consecutive VNFs: VNF 1 and VNF 2. In this system, an MEC server, called Server 1, is co-located with the base station and hosts one copy of VNF 1 as the primary MEC server. A secondary MEC server, called Server 2, is located nearby and also hosts a copy of VNF 1. In addition, Server 3 in the core network hosts VNF 2 and has more advanced computational capabilities than Servers 1 and 2. We assume a slotted time system. At each time slot, the device transmits a task in form of a packet to a base station over a wireless channel. Because of the presence of fading in

This work has been supported by the European Unions Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 643002.

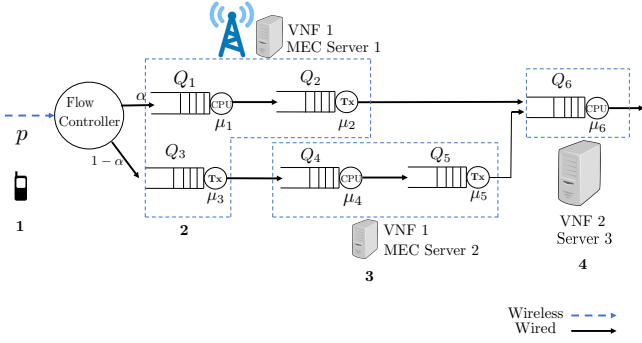


Fig. 1: The system model. The blue dashed lines group the queues located in the same server.

the wireless channel, transmissions may face errors. A task is successfully transmitted to the base station with a probability  $p$  that captures fading, attenuation, noise, etc. The device attempts for a new task transmission only if the previous task is successfully received at the base station. The received tasks need to be distributed between the queue for local processing and the queue for transmission to the secondary MEC server. Thus, there are two possible routes to pass through the service chain. A flow controller at the base station decides randomly the routing for each task<sup>1</sup>. With probability  $\alpha$  the task is processed by Server 1 first, and then forwarded to Server 3. With probability  $1 - \alpha$  the task is forwarded to Server 2, to be processed by VNF 1, and then forwarded to Server 3 for being processed by VNF 2.

Each task that arrives at a server first waits in a queue for being processed by a VNF. Then, after the processing, it is stored in the transmission queue, waiting to be forwarded and processed by the next VNF. Let  $Q_i$  denote the  $i$ -th queue, where  $i \in \mathcal{K}$ , and  $\mathcal{K}$  is the set of the queues in the system. Note that the queues follow an early departure-late arrival model: at the beginning of the slot the departure takes place and a new arrival can enter the queue at the end of the slot. The queues for task transmission are  $Q_2$ ,  $Q_3$ , and  $Q_5$ , and the queues for task processing are  $Q_1$ ,  $Q_4$ ,  $Q_6$ . The arrival rates for queues  $Q_1$  and  $Q_3$  are  $p\alpha$  and  $p(1 - \alpha)$ , respectively. We denote by  $\mu_i$ ,  $i \in \mathcal{K}$ , the service rates of the queues. We assume that the service times are geometrically distributed. Furthermore, given that  $Q_1$ ,  $Q_3$ , and  $Q_4$  are non empty, the arrival rates of  $Q_2$ ,  $Q_4$ , and  $Q_5$  are equivalent to the service rates of  $Q_1$ ,  $Q_3$ , and  $Q_4$  (i.e.,  $\mu_1$ ,  $\mu_3$ , and  $\mu_4$ ) respectively.

Furthermore, the queues at Servers 1 and 2 are assumed to have finite buffer. Let  $M_i$  denote the buffer size of each queue  $i \in \mathcal{K} \setminus \{6\}$ . If a queue is full and no task departs at the same time that a new one arrives, the new task is dropped and removed from the system. However, the queue of Server 3 (where  $Q_6$  is located) is assumed to have infinite length of buffer. In practice, the buffer in the core network has limited size, which is usually quite large. Our analysis based

on the infinite buffer size assumption can capture this scenario with minor modifications. However, we can extract insights on finding the appropriate queue size by performing the analysis assuming infinite queue size.

### III. PERFORMANCE ANALYSIS

In this section, we perform the modeling and the performance analysis that allow us to derive the critical performance metrics. We model the considered queueing system utilizing Discrete Time Markov Chain (DTMC). Modeling the whole system as one Markov chain can drive in a quite complicated system difficult to be analyzed in terms of closed-form expressions. Thus, in order to simplify the analysis, we decompose the system into different subsystems. We consider the following four subsystems: 1)  $Q_1$  and  $Q_2$ , 2)  $Q_3$  and  $Q_4$ , 3)  $Q_5$ , and 4)  $Q_6$ . The performance metrics for the whole system are approximated with the analytical expressions derived from the subsystems.

#### A. Subsystems 1 and 2: Two queues in tandem

The two queues in tandem  $Q_1$  and  $Q_2$  are considered a subsystem. The arrival rate for  $Q_1$  is:  $\lambda_1 = p\alpha$ . The Markov chain  $\{(X_n, Y_n)\}$  is described by  $P_{i,j;u,k} = \Pr\{X_{n+1} = i, Y_{n+1} = j \mid X_n = u, Y_n = k\}$ , where  $X_n$  and  $Y_n$  denote the states (in terms of queue length) of  $Q_1$  and  $Q_2$  at the  $n$ -th time slot, respectively, and  $i$  and  $j$  are referred to as the level  $i$  and phase  $j$ , respectively. The Markov chain is a Quasi-Birth-and-Death (QBD) DTMC [12]. Note that the QBD only goes a maximum level up or down, the transition matrix has a block partitioned form:

$$\mathbf{P}_1 = \begin{bmatrix} \mathbf{B} & \mathbf{C} & & & \\ \mathbf{E} & \mathbf{A}_1 & \mathbf{A}_0 & & \\ & \mathbf{A}_2 & \mathbf{A}_1 & \mathbf{A}_0 & \\ & & \ddots & \ddots & \\ & & & \mathbf{A}_2 & \mathbf{A}_0 + \mathbf{A}_1 \end{bmatrix}.$$

For the sake of simplicity, given a probability of an event, denoted by  $p$ , we denote the probability of its complementary event by  $\bar{p} \triangleq 1 - p$ . To derive the block matrices  $\mathbf{B}, \mathbf{C}, \mathbf{E}$  and  $\mathbf{A}_i$ , for  $i = 0, 1, 2$ , we first define the following matrices: First we define the following matrices

$$\mathbf{P}_1^{(1)} = \begin{bmatrix} 1 & 0 & & & \\ \mu_2 & \bar{\mu}_2 & & & \\ & \ddots & \ddots & & \\ & & \mu_2 & \bar{\mu}_2 & \end{bmatrix}, \mathbf{P}_1^{(2)} = \begin{bmatrix} 0 & 1 & 0 & & \\ 0 & \mu_2 & \bar{\mu}_2 & & \\ 0 & 0 & \mu_2 & \bar{\mu}_2 & \\ & & \ddots & \ddots & \\ & & & 0 & 1 \end{bmatrix}.$$

Then, the block matrices of the transition matrix are calculated as

$$\mathbf{B} = \bar{\lambda}_1 \mathbf{P}_1^{(1)}, \mathbf{C} = \lambda_1 \mathbf{P}_1^{(1)}, \mathbf{E} = \bar{\lambda}_1 \mu_1 \mathbf{P}_1^{(2)},$$

$$\mathbf{A}_0 = \lambda_1 \bar{\mu}_1 \mathbf{P}_1^{(1)}, \mathbf{A}_1 = \bar{\lambda}_1 \bar{\mu}_1 \mathbf{P}_1^{(1)} + \lambda_1 \mu_1 \mathbf{P}_1^{(2)}, \mathbf{A}_2 = \bar{\lambda}_1 \mu_1 \mathbf{P}_1^{(2)}.$$

Following the steps described above, utilizing the properties of a QBD DTMC, we can construct the transition matrix of Subsystem 1 for arbitrary finite buffer sizes.

<sup>1</sup>Probabilistic routing is a common strategy widely used in the literature, see for example [11].

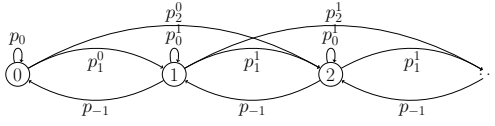


Fig. 2: Markov chain for  $Q_5$ .

Our goal is to derive the steady state distribution of the Markov chain defined above. We can apply direct methods in order to find the steady state distribution [12, Chapter 4]. Note that there are several efficient algorithms that can be used for this purpose, e.g., logarithmic reduction method.

We denote the steady state distribution of Subsystem 1 by a row vector  $\pi^{(1)} = [\pi_{0,0}^{(1)}, \pi_{0,1}^{(1)}, \dots, \pi_{0,M_2}^{(1)}, \pi_{1,0}^{(1)}, \dots, \pi_{M_1,M_2}^{(1)}]$ . We find  $\pi^{(1)}$  by solving the following linear system of equations  $\pi^{(1)}\mathbf{P}_1 = \pi^{(1)}$ ,  $\pi^{(1)}\mathbf{1} = 1$ , where  $\mathbf{1}$  denotes the column vector of ones. Hereafter we use  $\pi^{(n)}$  to denote the steady state distribution vector of the  $n$ -th subsystem for  $n = 1, 2, 3, 4$ .

Furthermore, the arrival rate of  $Q_2$  depends on the service rate of  $Q_1$ . However, the arrival rate of  $Q_2$  is equal to  $\mu_1$  if and only if  $Q_1$  is non-empty. Therefore, the arrival rate of  $Q_2$  is  $\lambda_2 = \Pr\{Q_1 > 0\} \mu_1 = \left(\sum_{j=0}^{M_2} \sum_{i=1}^{M_1} \pi_{i,j}^{(1)}\right) \mu_1$ . Similarly, we can construct the transition matrix  $\mathbf{P}_2$  and the steady state distribution  $\pi^{(2)}$  for the second subsystem consisting of  $Q_3$  and  $Q_4$ . The arrival rates of  $Q_3$  and  $Q_4$  are  $\lambda_3 = p(1-\alpha)$  and  $\lambda_4 = \Pr\{Q_3 > 0\} \mu_3 = \left(\sum_{j=0}^{M_4} \sum_{i=1}^{M_3} \pi_{i,j}^{(2)}\right) \mu_3$ , respectively.

### B. Subsystem 3: $Q_5$ with finite buffer

We consider  $Q_5$  as an independent subsystem.  $M_5$  is the buffer size of the queue. We first define the arrival rate of  $Q_5$ :

$$\lambda_5 = \Pr\{Q_4 > 0\} \mu_4 = \left(\sum_{i=0}^{M_3} \sum_{j=1}^{M_4} \pi_{i,j}^{(2)}\right) \mu_4.$$

We model the subsystem as one Markov chain whose transition matrix is shown below

$$\mathbf{P}_3 = \begin{bmatrix} \bar{\lambda}_5 & \lambda_5 & & & \\ \bar{\lambda}_5 \mu_5 & \lambda_5 \mu_5 + \bar{\lambda}_5 \bar{\mu}_5 & \lambda_5 \bar{\mu}_5 & & \\ & \lambda_5 \mu_5 & \lambda_5 \mu_5 + \bar{\lambda}_5 \bar{\mu}_5 & \lambda_5 \bar{\mu}_5 & \\ & & \ddots & \ddots & \ddots \\ & & & \bar{\lambda}_5 \mu_5 & \bar{\lambda}_5 \bar{\mu}_5 + \lambda_5 \end{bmatrix}.$$

We denote the steady state distribution of Subsystem 3 by  $\pi^{(3)} = [\pi_0^{(3)}, \pi_1^{(3)}, \dots, \pi_{M_5}^{(3)}]$ . To derive  $\pi^{(3)}$ , we solve the following linear system of equations:  $\pi^{(3)}\mathbf{P} = \pi^{(3)}$ ,  $\pi^{(3)}\mathbf{1} = 1$ . Using balance equations, we obtain  $\pi_i^{(3)} = \frac{\lambda_5 \bar{\mu}_5^{i-1}}{\lambda_5^i \mu_5^i} \pi_0^{(3)}$ , for  $1 \leq i \leq M_5$ , and  $\pi_0^{(3)} = \left[1 + \sum_{i=1}^{M_5} \frac{\lambda_5^i \bar{\mu}_5^{i-1}}{\lambda_5^i \mu_5^i}\right]^{-1}$ .

### C. Subsystem 4: $Q_6$ with infinite buffer size

The arrival rate for  $Q_6$  depends on the service rate of  $Q_2$  and  $Q_5$ , and the probability that the queues are non-empty. Note that the departures from  $Q_2$  and  $Q_5$  can be considered

independent stochastic processes. The arrival rates that occur due to  $Q_2$  and  $Q_5$  are  $\lambda_{6,2} = \Pr\{Q_2 > 0\} \mu_2$  where  $\lambda_{6,5} = \Pr\{Q_5 > 0\} \mu_5$ , respectively. The arrival rate of  $Q_6$  is:  $\lambda_6 = \lambda_{6,2} + \lambda_{6,5}$ . We model the system as a Markov chain as shown in Fig. 2, where

$$\begin{aligned} p_0 &= \bar{\lambda}_{6,2} \bar{\lambda}_{6,5}, p_1^0 = \lambda_{6,2} \bar{\lambda}_{6,5} + \lambda_{6,5} \bar{\lambda}_{6,2}, p_2^0 = \lambda_{6,2} \lambda_{6,5}, \\ p_0^1 &= \bar{\lambda}_{6,5} \bar{\lambda}_{6,2} \bar{\mu}_6 + \bar{\lambda}_{6,5} \lambda_{6,2} \mu_6 + \lambda_{6,5} \bar{\lambda}_{6,2} \mu_6, p_2^1 = \lambda_{6,2} \lambda_{6,5} \bar{\mu}_6 \\ p_1^1 &= \lambda_{6,2} \bar{\lambda}_{6,5} \bar{\mu}_6 + \bar{\lambda}_{6,2} \lambda_{6,5} \bar{\mu}_6 + \lambda_{6,2} \lambda_{6,5} \mu_6, \\ p_2^1 &= \lambda_{6,2} \lambda_{6,5} \bar{\mu}_6, p_{-1} = \bar{\lambda}_{6,2} \bar{\lambda}_{6,5} \mu_6. \end{aligned}$$

The transition matrix that describes the Markov chain above is shown below

$$\mathbf{P}_6 = \begin{bmatrix} a_0 & b_0 & 0 & 0 & \cdots \\ a_1 & b_1 & b_0 & 0 & \cdots \\ a_2 & b_2 & b_1 & b_0 & \cdots \\ 0 & b_3 & b_2 & b_1 & \cdots \\ \vdots & 0 & b_3 & b_2 & \cdots \end{bmatrix},$$

where  $a_0 = p_0^0$ ,  $a_1 = p_1^0$ ,  $a_2 = p_2^0$ ,  $b_0 = p_{-1}$ ,  $b_1 = p_1^1$ ,  $b_2 = p_2^1$ ,  $b_3 = p_2^1$ . The transition matrix is a lower Hessenberg matrix. We denote the steady state distribution of Subsystem 4 by  $\pi^{(4)} = [\pi_0^{(4)}, \pi_1^{(4)}, \dots]$ . The general expression for the equilibrium equations of states is given by the  $i$ -th term in the following equation:  $\pi_i^{(4)} = a_i \pi_0^{(4)} + \sum_{j=1}^{i+1} b_{i-j} \pi_j^{(4)}$ . For the DTMC with infinite state space, we apply  $z$ -transform approach to solve the state equations. The  $z$ -transforms for the state transition probabilities  $a_i$  and  $b_i$  are  $A(z) = \sum_{i=0}^{\infty} a_i z^{-i}$  and  $B(z) = \sum_{i=0}^{\infty} b_i z^{-i}$ , respectively. The  $z$ -transform for the steady state distribution vector  $\pi^{(4)}$  is  $\Pi(z) = \sum_{i=0}^{\infty} \pi_i^{(4)} z^{-i} = \pi_0^{(4)} \frac{z^{-1} A(z) - B(z)}{z^{-1} - B(z)}$ . The solution for  $\pi_i^{(4)}$  is given by

$$\pi_0^{(4)} = \frac{1 + B'(1)}{1 + B'(1) - A'(1)}, \pi_i^{(4)} = c_i + \sum_{j=1}^m r_j (p_j)^{(i-1)}, i > 0,$$

where  $r$ ,  $p$ , and  $c$  are the residues, poles, and direct terms, respectively. Since  $Q_6$  has infinite buffer size, we need to characterize the conditions under which the queue is stable. The Loynes' theorem states: if the arrival and service processes of a queue are strictly jointly stationary and the average arrival rate is less than the average service rate, then the queue is stable. Therefore,  $Q_6$  is stable if and only if the following inequality holds:  $\lambda_6 < \mu_6$ .

### D. Discussion on the analysis of scaled-up systems

In this work, we analyze a simple end-to-end system that consists of three connected servers. We can analyze systems with arbitrary number of servers by decomposing the system into subsystems and analyze each subsystem individually. Then, we use the results of each subsystem in order to derive the analytical expressions for the whole system. A full version of this work can be found in [13].

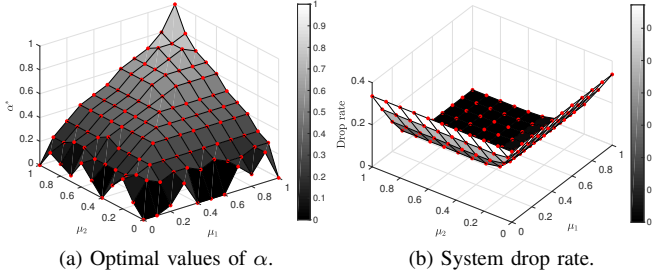


Fig. 3: Objective: To minimize the system drop rate.  $\mu_3 = \mu_4 = \mu_5 = 0.5$ ,  $\mu_6 = 0.9$ ,  $p = 0.8$ .  $M_i = 10$ , for  $1 \leq i \leq 5$ .

#### IV. KEY PERFORMANCE METRICS

In this section, we provide analytical expressions of the performance metrics of our interests, i.e., system drop rate and average number of tasks of the system by utilizing the results of the previous section. The probabilities to have a dropped task at each time slot for  $Q_1 - Q_5$  are shown respectively in below

$$P_{D_1} = \lambda_1 \bar{\mu}_1 \sum_{j=0}^{M_2} \pi_{M_1,j}^{(1)}, P_{D_2} = \lambda_2 \bar{\mu}_2 \sum_{i=1}^{M_1} \pi_{i,M_2}^{(1)},$$

$$P_{D_3} = \lambda_3 \bar{\mu}_3 \sum_{j=0}^{M_4} \pi_{M_3,j}^{(2)}, P_{D_4} = \lambda_4 \bar{\mu}_4 \sum_{i=1}^{M_3} \pi_{i,M_4}^{(2)}, P_{D_5} = \lambda_5 \bar{\mu}_5 \pi_{M_5}^{(3)},$$

where  $P_{D_i}$  is the probability to have a dropped task of queue  $i$ . The average length of each queue is given by

$$\bar{Q}_1 = \sum_{i=0}^{M_1} \sum_{j=0}^{M_2} \pi_{i,j}^{(1)} i, \bar{Q}_2 = \sum_{j=0}^{M_2} \sum_{i=0}^{M_1} \pi_{i,j}^{(1)} j, \bar{Q}_3 = \sum_{i=0}^{M_3} \sum_{j=0}^{M_4} \pi_{i,j}^{(2)} i,$$

$$\bar{Q}_4 = \sum_{j=0}^{M_4} \sum_{i=0}^{M_3} \pi_{i,j}^{(2)} j, \bar{Q}_5 = \sum_{i=0}^{M_5} \pi_i^{(3)} i, \bar{Q}_6 = \sum_{i=0}^{\infty} \pi_i^{(4)} i.$$

Therefore, the system drop rate and the average number of task in the system can be described as

$$P_D = \sum_{i \in \mathcal{K} \setminus \{6\}} P_{D_i} \text{ and } \bar{Q} = \sum_{i \in \mathcal{K}} \bar{Q}_i, \text{ respectively.}$$

#### V. NUMERICAL RESULTS

In this section, we evaluate the accuracy of our derived mathematical model in terms of key performance metrics by comparing the analytical with simulation results. Furthermore, we provide results regarding the system performance under different setups. We developed a MATLAB-based behavioural simulator and each case run for  $10^6$  timeslots.

##### A. Effect of $\mu_1$ and $\mu_2$ on the drop rate in systems with small buffers

In this subsection, we observe the performance of the system in terms of the drop rate when the size of the buffers is small. In Fig. 3a, we provide the optimal values of  $\alpha$  (probabilistic routing decision) for different values of  $\mu_1$  and  $\mu_2$ . Note that

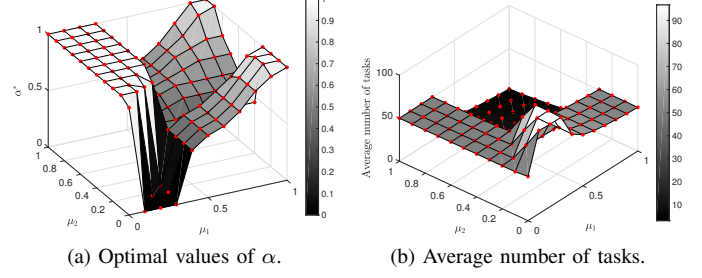


Fig. 4: Objective: To minimize the average number of tasks.  $\mu_3 = \mu_4 = \mu_5 = 0.5$ ,  $\mu_6 = 0.9$ ,  $p = 0.8$ .  $M_i = 50$ , for  $1 \leq i \leq 5$ .

we obtain the optimal  $\alpha$  for each value of  $\mu_1$  and  $\mu_2$  by applying brute force. We observe that for small values of  $\mu_1$  and  $\mu_2$ , the value of  $\alpha$  is small (around 0.2). Therefore, the routing selects to route the traffic flow to the secondary MEC server (Server 2). Furthermore, it is shown that the value of optimal  $\alpha$  is affected by the smaller value between the transmission rate and processing rate. Therefore, the buffer with the smallest transmission or computation capacity becomes the bottleneck for the subsystem. This could be the case, for example, when the connection between the MEC server 1 and the server in the core network is weak. Fig. 3b depicts the system drop rate for the corresponding optimal  $\alpha$ 's.

##### B. Effect of $\mu_1$ and $\mu_2$ on the number of tasks in systems with large buffers

In this subsection, we provide results for the performance of the system in terms of average number of tasks. Our objective is to minimize the average number of tasks in the system when the buffer size is large. In Fig. 4a, the optimal  $\alpha$ 's for different values of  $\mu_1$  and  $\mu_2$  are shown. We observe that for small values of  $\mu_1$ , the optimal value of  $\alpha$  is equal to 1. The flow controller decides to route the whole traffic to the first server. This decision is optimal in terms of minimizing the average number of tasks in the system, but it increases significantly the system drop rate. The reason is that a large percentage of the tasks are dropped and but not served. We also observe that the smallest value between  $\mu_1$  and  $\mu_2$  operates as bottleneck in the subsystem and subsequently in the whole system.

##### C. Trade-off between system drop rate and average queue length - simulation vs analytical results

In this subsection, we provide results that show the trade-off between the system drop rate and average queue length for different routing decisions. In addition, we compare the analytical with simulation results and evaluate the accuracy of our model. In this paper, we show only one scenario due to the space limitation. We observe that  $\alpha$  with values around 0.5 provide the best trade-off. In addition, an interesting result is shown for the case that  $\alpha$  takes extreme values, i.e., 0.1 and 0.9. Although the system drop rate is almost the same for these two cases, the average queue length is quite different. The reason is that when the first path is selected with higher probability, the traffic flow traverses less number of queues

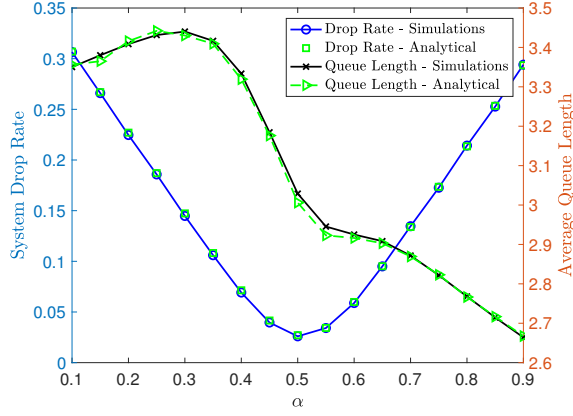


Fig. 5: System drop rate and average queue length trade-off. Analytical vs simulation results.  $\mu_i = 0.45$  for  $1 \leq i \leq 5$ ,  $\mu_6 = 0.9$ .

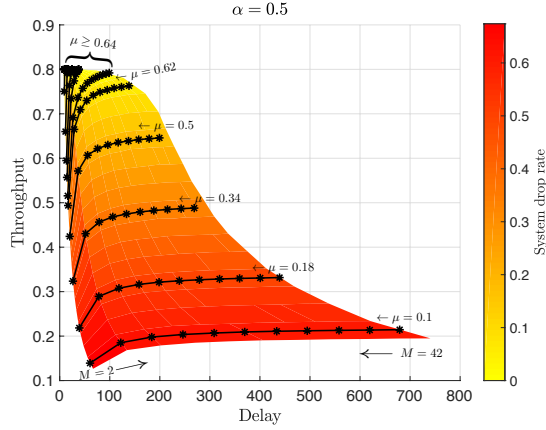


Fig. 6: Performance region.  $\mu_i = \mu$  for  $1 \leq i \leq 5$ ,  $M_i = M$  for  $1 \leq i \leq 5$ .  $p = 0.8$ ,  $\mu_6 = 0.9$ .

comparing to the second path. In this case, three of the queues, i.e.,  $Q_3, Q_4, Q_5$ , are lightly loaded. On the other hand, when the probability the second path to be selected is high, i.e., smaller value of  $\alpha_1$ , the traffic traverses larger number of queue. In this case, more queues are heavily loaded and the average queue length increases.

#### D. Effect of different buffer capacities and service rates on throughput and delay

In order to further investigate the performance of the system, we provide simulation results that show how different setups of the system affect the system throughput and delay. Note that the analytical expressions for the throughput and delay are calculated by using the results for the system drop rate and average queue length, respectively. We omit the analysis due to space limitation and we provide the complete analysis in an extended version of this work. However, it is interesting to show some preliminary results. In Fig. 6, we provide results for the throughput, delay, and corresponding system drop rate

for different values of  $\mu$  and  $M$ . We obtain the throughput and delay as following: We fix the service rate  $\mu$ , and change the capacity of the buffers  $M$ . Thus, we create each black horizontal line with the stars. The colormap represents the values of system drop rate.

We observe that system performance is significantly affected when we increase the service rates of the buffers. On the other hand, when the service rates are low but the capacities of the buffers are large, the system performance is not improved. From these preliminary results, we observe that it is more important to increase the service rate than the capacity of the buffers.

## VI. CONCLUSIONS & FUTURE DIRECTIONS

In this work, we consider a network topology with two MEC servers, a high-end server at core network, and VNF chains embedded in the servers. We model the network and provide an analytical study on the system performance in terms of system drop rate and average number of the tasks in the system. It is shown, through simulations results, that the approximate model performs well. Numerical results also show useful insights on the design of such systems or resource allocation at each server. Furthermore, we investigate numerically the routing policy that optimizes different objective functions.

## REFERENCES

- [1] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile edge computing—A key technology towards 5G," *ETSI white paper*, vol. 11, no. 11, pp. 1–16, 2015.
- [2] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, and D. Sabella, "On multi-access edge computing: A survey of the emerging 5G network edge cloud architecture and orchestration," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1657–1681, 2017.
- [3] R. Mijumbi, J. Serrat, J.-L. Gorricho, N. Bouten, F. De Turck, and R. Boutaba, "Network function virtualization: State-of-the-art and research challenges," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 236–262, 2016.
- [4] M. S. Bonfim, K. L. Dias, and S. F. L. Fernandes, "Integrated NFV/SDN architectures: A systematic literature review," *ACM Comput. Surv.*, vol. 51, no. 6, pp. 114:1–114:39, 2019.
- [5] R. Cohen, L. Lewin-Eytan, J. S. Naor, and D. Raz, "Near optimal placement of virtual network functions," in *proc. IEEE INFOCOM*, pp. 1346–1354, 2015.
- [6] L. Wang, Z. Lu, X. Wen, R. Knopp, and R. Gupta, "Joint optimization of service function chaining and resource allocation in network function virtualization," *IEEE Access*, vol. 4, pp. 8084–8094, 2016.
- [7] H. Feng, J. Llorca, A. M. Tulino, and A. F. Molisch, "Optimal dynamic cloud network control," *IEEE/ACM Transactions on Networking*, no. 99, pp. 1–14, 2018.
- [8] Q. Ye, W. Zhuang, X. Li, and J. Rao, "End-to-end delay modeling for embedded VNF chains in 5G core networks," *IEEE Internet of Things Journal*, pp. 1–1, 2018.
- [9] H.-N. Nguyen, T. Begin, A. Busson, and I. G. Lassous, "Approximating the end-to-end delay using local measurements: A preliminary study based on conditional expectation," in *Proc. IEEE ISNCC*, pp. 1–6, 2016.
- [10] —, "Evaluation of an end-to-end delay estimation in the case of multiple flows in SDN networks," in *IEEE CNSM*, 2016, pp. 336–341.
- [11] M. Ploumidis, N. Pappas, and A. Traganitis, "Flow allocation for maximum throughput and bounded delay on multiple disjoint paths for random access wireless multihop networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 1, pp. 720–733, 2016.
- [12] A. S. Alfa, *Applied discrete-time queues*. Springer, 2016.
- [13] E. Fountoulakis, Q. Liao, and N. Pappas, "An end-to-end performance analysis for service chaining in a virtualized network," <https://arxiv.org/abs/1906.10549>.