The Impact of Human Mobility on Edge Data Center Deployment in Urban Environments

Piergiorgio Vitello♯, Andrea Capponi♯, Claudio Fiandrino⋆, Guido Cantelmo†, Dzmitry Kliazovich♦
♯ FSTC/CSC, University of Luxembourg, Luxembourg, ⋆ IMDEA Networks Institute, Madrid, Spain,
† Technical University of Munich (TUM), Germany, ♦ ExaMotive, Luxembourg
E-mails: ♯{firstname.lastname}@uni.lu, ⋆claudio.fiandrino@imdea.org, †g.cantelmo@tum.de, ♦kliazovich@ieee.org

***Abstract*—Multi-access Edge Computing (MEC) brings storage and computational capabilities at the edge of the network into so-called Edge Data Centers (EDCs) to better low-latency applications. To this end, effective placement of EDCs in urban environments is key for proper load balance and to minimize outages. In this paper, we specifically tackle this problem. To fully understand how the computational demand of EDCs varies, it is fundamental to analyze the complex dynamics of cities. Our work takes into account the mobility of citizens and their spatial patterns to estimate the optimal placement of MEC EDCs in urban environments in order to minimize outages. To this end, we propose and compare two heuristics. In particular, we present the mobility-aware deployment algorithm (MDA) that outperforms approaches that do not consider citizens mobility. Simulations are conducted in Luxembourg City by extending the CrowdSenSim simulator and show that efficient EDCs placement significantly reduces outages.**

I. Introduction

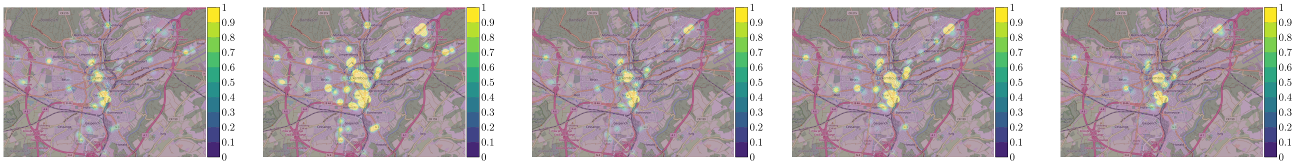
The fifth generation (5G) mobile networks relies on Software-Defined Networking (SDN) and Network Function Virtualization (NFV) to support next-generation services. Radio access and core functions are virtualized and executed in edge data centers (EDCs) according to the Multi-access Edge Computing (MEC) principle. MEC was standardized by the European Telecommunications Standards Institute (ETSI) [1] and brings applications and computing services closer to the end user, or citizen [2]. Thus, it finds applicability in scenarios where locality and low-latency are essential [3]. MEC is agnostic to the evolution of the mobile network itself and can operate with LTE, 4G, or 5G mobile networks.

The *edge*, also known as MEC host, is a data center or nano data center deployed in close proximity of base stations, i.e., inside an operator-owned infrastructure. The edge provides computing functionalities for applications and can aggregate virtualized core and radio network functions of the mobile network [3]. Offloading code execution to EDCs allows resource-constrained mobile devices to prolong battery lifetime [4] while enhancing and augmenting the performance of mobile applications [5]. Emulation platforms for research in the area have only started to appear recently [6] and little attention has been paid to the problem of EDC deployment.

EDC deployment is particularly interesting in the context of smart cities. To the best of our knowledge, only a vision paper [7] has explored such area by assessing the feasibility of leveraging three different infrastructures to deploy computing capacity, i.e., cellular base stations (BSs), routers and street lamps. The article analyzed the potential city coverage if only a subset of these elements is upgraded to furnish EDC capabilities. This study is an important step forward to solve the problems of coverage, EDC selection, and user-to-EDC assignment. However, it fails to capture the mobility dynamics of a city fully. To jointly characterize urban dynamics (temporal behaviors and spatial patterns of citizens) and the traffic dynamics is crucial to estimate computational demands and, in turns, to devise effective EDC deployments. A recent study unveils a consistent influence of urbanization level in the average traffic volume per-user. In addition, different mobile services exhibit different temporal behaviors, although spatial patterns remain uniform [8].

In this paper, we bring the research in edge computing one step forward. Specifically, we tackle the problem of EDCs deployment in a smart city context by considering two factors. First, we consider cellular connectivity for network access and assume that EDCs should only be deployed at current Base Stations (BSs) sites to re-use already deployed infrastructure (e.g., power supply, cabinets on roofs). Thus, our solution is capital-expenditure free for mobile network operators. Second, we focus on human mobility. Within a city, complex dynamics regulate the inter-dependency of land use and citizens movements [9], i.e., the spatial distribution of citizens and locations they visit that determine mobility patterns. Similarly to [10], we make use of crowdsensed data to infer and predict human mobility with the goal of determining estimates of computing demand and the optimal EDC deployment that minimizes outages. Specifically, we leverage Google Popular Times to estimate citizens mobility that reflects daily urban patterns. For these reasons, we focus on citizens mobility during a day and consider LTE traffic generated from mobile users. Fig. 1 shows the heatmaps of the potential computing demand of the BSs in Luxembourg City according to the citizens mobility based on Google Popular Times[1] in a weekday. BSs under heavy loads are around the railway stations when citizens commute (i.e., H: 8:00 and H: 17:00) in the city center at lunch and dinner times, in the university area during the day. By considering different hours of a day, loads of BSs vary and, in turns, the potential computational demand at EDCs, which motivates our research in EDC deployment.

[1] https://support.google.com/business/answer/2721884



(a) H: 8:00 (b) H: 12:00 (c) H: 15:00 (d) H: 17:00 (e) H: 20:00

Fig. 1. Traffic generation in Luxembourg City at different hours of a working day

II. Related Works

MEC allows resource-constrained mobile devices to offload computational workload to nearby EDCs. How to enable flexible MEC platform service consumption at different localities is studied in [11]. The paper presents a stochastic model to determine the total processing time. Through control plane traffic, the solution allows for signaling between the MEC entities so that applications instances always remain within a certain zone with tolerable latency. In [12], the authors propose a resource scheduling management and use machine learning to predict offloading costs, focusing on a protest crowd incident case study. Paradrop provides computing and storage capabilities and allows third parties to develop new types of services [13]. It includes a flexible hosting substrate in the WiFi APs that supports multi-tenancy, a cloud-based backend, and an API for third-party developers. The integration of lightweight virtualization (LV) with edge networks is discussed in [14]. The authors show how LV solutions can bring flexibility and highlight the benefits for three different use cases, i.e., autonomous vehicles, smart city, and augmented reality.

The human-driven edge computing (HEC) is a paradigm that leverages the mutual benefits of MEC and mobile crowd-sensing (MCS) [15]. The authors show the advantages of this approach by integrating the MEC platform Elijah[2] and the MCS platform ParticipAct Living Lab [16]. Some approaches propose a resource allocation model at the edge [17] and a multi-cloudlet infrastructure in the context of smart city scenarios [7]. The closest work to ours is [18], where the authors present an optimal cloudlet placement and allocation of users to cloudlets in Wireless Metropolitan Area Networks (WMANs). Although this work assumes a paramount importance for cloudlets placement, it does not capture user movements within the network and user scalability in large urban environments, which are fundamental factors in nowadays smart cities characterized by the increasing mobility of citizens that influence the workload and computational demand of edge resources. This paper aims to fill this gap as our focus is to quantify the importance of user mobility at the city scale to estimate the computational demand and deploy EDC accordingly.

III. Models for Edge Data Center Deployment and Urban Environments' Dynamics

This Section formulates the problem of EDC distribution over urban environments and presents the system model that captures MEC dynamics (both at computing and networking level) and citizens mobility.

A. Problem Formulation

Let $\mathcal{B} = \{b_1, ..., b_{N_B}\}$ be the set of BSs, each characterized by latitude and longitude that define the location of b_i in the city and $\mathcal{E} = \{e_1, ..., e_{N_E}\}$ the set of EDCs to be deployed, where $N_E < N_B$ and $\mathcal{E} \subset \mathcal{B}$. EDCs are deployed at BS sites to re-use already deployed infrastructure (e.g., power supply, cabinets on roofs). This provides full spatial coverage and is more cost-effective than creating new EDC sites [7]. Similarly to [19], we consider as Key Performance Indicator (KPI) the *latency outage probability of the system* O, defined as:

$$O = Pr\{L \geq D_{max}\}, \tag{1}$$

where L is the latency the user observes and D_{max} is the maximum acceptable delay bound defined in the form of SLA agreement for the current application. While in [19] the delay outage probability indicates the one-way latency, we need for the *latency outage probability of the system* to measure the Round-Trip-Time (RTT) to capture the fact that if a user does not receive a reply from the EDC, the task is not accomplished and O increases. This can happen for two reasons:

- the EDC rejects the incoming task because it is overloaded;
- the user does not receive the reply in due time because of either processing or networking delays.

Given a fixed number of EDCs N_e to deploy, our problem consists in finding a match with the location of existing BSs to minimize the average latency outage probability of the entire system (e.g., to maximize the computational capacity of the system):

$$\min_{E} O. \tag{2}$$

where E is the set of BSs that are chosen to place the EDCs.

B. MEC Model

To assign EDCs to BSs, the city environment is divided into a set of $\mathcal{R} = \{r_1, ..., r_l\}$ regions. The EDC of each region is connected to all BSs within the region and responsible for both applications and baseband processing (e.g., according to the cloud-RAN paradigm and the chosen functional split [20]). Each EDC has a fixed number of N_s servers with an equal service rate μ. When the service rate is not sufficient to fulfill a task in due time, it is rejected by the system. We assume no service migration across different EDCs when the users move within the same region. Otherwise, the service is simply detached from a EDC and attached to another according to, e.g., a micro-services stateless paradigm. Finally, we assume the mobile users are always connected to the closest BS.

[2] http://elijah. cs.cmu.edu/

Users generate heterogeneous application-dependent types of tasks [18]. The task arrival is modeled with a Poisson process with arrival rate λ_i for each user u_i.

To access the EDC processing, a user sends a message which is acknowledged if EDC resources permit its execution. The latency L of the request/reply exchange includes both *network* and *processing delays* [11]. The network delay D_p consists of different components, such as transmission, propagation, queuing, and routing. The processing delay D_c depends on application processing and packet processing at the network level. They will be discussed in Sec. V-A. For our purposes, L is given by:

$$L^i = D_p^{i,k} + D_c^k + D_p^{k,i}, \tag{3}$$

where $D_p^{i,k}$ is the *network delay* from user i to EDC k, D_c^k is the *processing delay* at EDC k, and $D_p^{k,i}$ is the *network delay* from EDC k to user i.

Each EDC is modeled as a $M/M/N_s$ queue with N_s servers. The processing delay, which represents the task execution time in a EDC, is calculated as follows [18]:

$$D_c^k = f_Q\left(\phi_k \cdot \sum_{u_i \in \mathcal{U}_|} \lambda_i\right) + 1/\mu, \tag{4}$$

where ϕ_k is the fraction of accepted tasks in the EDC, and $f_Q(\lambda)$ determines the average queueing time. ϕ_k is given by:

$$\phi_k = \begin{cases} 1, & \text{if } \lambda_{max} > \lambda(k); \\ \frac{\lambda_{max}}{\lambda(k)}, & \text{otherwise.} \end{cases} \tag{5}$$

$f_Q(\lambda)$ takes in input the task arrival rate $\lambda(k) = \sum_{u_i} \lambda_i$ at EDC e_k and returns the average queuing time:

$$f_Q(\lambda) = \frac{C\left(N_s, \frac{\lambda}{\mu}\right)}{N_s\mu - \lambda}. \tag{6}$$

C is given by the Erlang's Formula [21]:

$$C(N_s, \rho) = \frac{\left(\frac{(N_s\rho)^{N_s}}{N_s!}\right)\left(\frac{1}{1-\rho}\right)}{\sum_{k=0}^{N_s-1} \frac{(N_s\rho)^k}{k!} + \left(\frac{(N_s\rho)^{N_s}}{N_s!}\right)\left(\frac{1}{1-\rho}\right)}, \tag{7}$$

where ρ is the ratio of arrival rate to service rate (λ/μ).

C. Citizens Mobility Model

Users mobility defines spatial patterns of citizens movements and their social interactions, influencing the demand for computing resources. In this work, to characterize urban mobility, we exploit the popularity of Local Businesses (LB) taken from Google Popular Times, given in per-hour values normalized between the weekly maximum and minimum number of customers of each LB. Note that the real number of customers remains unknown. Hence, one of our contributions explained hereafter, is a new approach to estimate the number of customers from the coarse measurements available. The popularity metric is then used to approximate users temporal distribution among different LBs.

For each type of LB t, we consider a random value between 0 and N_{max}^t, where N_{max}^t is the maximum value of customers. This value and the average waiting time of staying in a LB permits to compute the number of people who remained at that LB. The aggregation of different LBs in a region defines how crowded a district is. In summary:

- For each local business L of typology t we draw the maximum number of people N_L^t visiting that specific LB from the uniform distribution $[0; N_{max}^t]$.
- The time-dependent demand for a certain L is obtained by combining the number of visits with the popular times:
$$D_{L,h} = P_{h,L} \cdot N_L^t, \tag{8}$$
where $D_{L,h}$ is the number of users visiting location L during time interval h and $P_{h,L}$ is the popularity index according to our data.
- Finally, for a certain geographical area d, the overall demand is obtained as the combination of all LB located in that specific area as:
$$A_{d,h} = \sum_{l \in L_d} D_{l,h}, \tag{9}$$
where $A_{d,h}$ is the demand of a certain geographical area d at time h and L_d is the sub set of LB located in d.

IV. EDCs Deployment Policies

The placement policy and BSs assignment to EDCs are keys for effective EDCs deployment. This work proposes and compares two placement policies. The first one is called *distributed deployment algorithm (DDA)* and deploys EDCs so that they are the centroids of a cluster composed of a set of base stations that all share a similar distance. The second policy, called *mobility-aware deployment algorithm (MDA)*, considers the mobility of citizens and their social interactions in urban environments to calculate the expected computational demand and distribute edge resources by exploiting them as weights. In the following, we explain in details both policies.

A. Distributed Deployment Algorithm (DDA)

The Distributed Deployment Algorithm (DDA) distributes EDCs according to the k-medoids clustering algorithm. DDA places EDCs and assigns BSs to them by exploiting the k-medoids clustering algorithm, which is similar to the more famous k-means. While in the k-means algorithm the centers of clusters are not necessarily input data points, the k-medoids method chooses the centroids between the input data. In other words, it clusters BSs and chooses EDCs as centers of the clusters between the BSs within a cluster, which fits our purpose perfectly. More specifically, DDA assigns EDCs among BSs by computing a cost based on the distances between BSs candidates as EDCs and all other BSs assigned to it. The main shortcomings of this approach consist in some under-utilized EDCs and others that suffer of big delays due to overloads of computational demand leading to high values of outage probability, as discussed more in details in V-B. To overcome these issues, two possible directions can be investigated. First, to propose a more effective placement of EDCs among the BSs, which is discussed in IV-B. Second, to allocate servers among EDCs proportionally with the computational demand, which is presented in IV-C.

Algorithm 1 MDA

```
1: Input: $B$, $R$, $n$ ▷ BSs, requests at BSs, number of EDCs
2: Output: $E$, $C$ ▷ Set of BSs chosen as EDCs, Set of BSs-EDCs connections
3: $AVG\_reqs = sum(R)/n$ ▷ Average num. of requests per EDC
4: $E \leftarrow Random(B, n)$ ▷ Randomly select $n$ BSs as EDC
5: $Cost \leftarrow getCost(E, AVG\_reqs, B, R)$ ▷ Compute cost
6: while $Swaped$ do
7:     $Swaped \leftarrow False$
8:     for $b \in B$ do
9:         for $i = 1 \rightarrow n$ do
10:            $E\_dup = E$ ▷ Copy of $E$
11:            $E\_dup_i \leftarrow b$
12:            $Cost\_dup = getCost(E\_dup, AVG\_reqs, B, R)$
13:            if $Cost\_dup < Cost$ then
14:                $E \leftarrow E\_dup$
15:                $Swaped \leftarrow True$
16:            end if
17:        end for
18:    end for
19: end while
20: procedure GETCOST($E, AVG\_reqs, B, R$)
21:     $R\_tot = \emptyset$
22:     $Cost = 0$
23:     for $b \in B$ do
24:         $e = ClosestEDC(b, E)$ ▷ Find EDC in $E$ at minimum distance from $b$
25:         $C_b \leftarrow e$ ▷ Connect BS $b$ to the nearest EDC $e$
26:         $R\_tot_e \leftarrow R\_tot_e + R_b$ ▷ Sum requests at EDC
27:     end for
28:     for $e \in E$ do
29:         $Cost \leftarrow Cost + |R\_tot_e - AVG\_reqs|$ ▷ Compute cost
30:     end for
31:     return Cost
32: end procedure
```

B. Mobility-aware Deployment Algorithm (MDA)

MDA aims to decrease the overall outage probability of the system by considering where the computational demand is higher according to the spatial patterns of citizens. In other words, the MDA solution is based on the idea to consider the complex dynamics of a city (e.g., user mobility and social interactions) to propose a more effective placement of edge resources. As DDA, MDA places EDCs among BSs by exploiting the k-medoids algorithm. Differently from DDA, it assigns EDCs among BSs by computing a cost based on the number of requests received by BSs and corresponding computational demand for EDCs. MDA is based on an iterative approach (see the MDA pseudocode in Algo 1) that computes how far the total number of requests for each EDC is from the average number of requested each EDC should have. Specifically, note that line 29 calculates the previously explained cost.

C. Allocation of Servers among EDCs

Another fundamental aspect to investigate is how to choose the number of servers to allocate among EDCs. We focus on this issue by formulating the following problem: given a certain fixed total amount of servers, how can we allocate them among EDCs? To this end, we compare a fixed number of servers (FNS) approach and a proportional number of server (PNS) approach. The former simply allocates the same number of servers for each EDC. The latter distributes the total number of servers proportionally with the computational demand of each EDC (taking into account its temporal evolution).

V. Performance Evaluation

A. Simulation Set-up

Next, we expose assumptions, parameters related to the computational demand and the urban scenario, which includes the city layout and user mobility (summarized in Table I).

TABLE I
SETUP PARAMETERS

SYMBOL	VALUE	DESCRIPTION
N_u	100 000	Number of users
N_b	141	Number of BSs
N_e	8	Number of edge data centers
N_l	1083	Number of total LBs
N_t	13	Number of LBs typologies
N_s	10	Number of servers in each EDC
λ_i	$0 < \lambda_i < 2.99$	Task arrival rate for user i
μ	100	Server service rate

MEC set-up: The 141 considered BSs represent the real infrastructure of Luxembourg City of BSs for public mobile communication network over 50 W[3], which are imported as a layer on a map to extract their coordinates (latitude and longitude)[4]. Each user generates traffic with an arrival rate of λ_i set in the range $[0 - 2.99]$ [18]. The service rate of each server is $\mu = 100$.

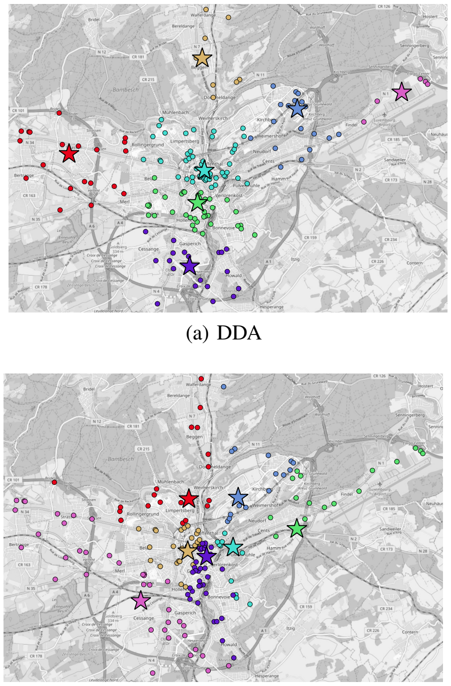
City layout and user mobility: To simulate user mobility in realistic urban environments, we extend CrowdSenSim [22] originally developed for mobile crowdsensing [23]. Specifically, we consider a street network graph based on Open Street Maps and augment its precision with an algorithm [24] to determine user trajectories accurately. We operate in Luxembourg City and consider 1 083 LBs belonging to 13 different categories (e.g., restaurants, pubs, public offices, etc.) from Google Popular Times. 100 000 pedestrian walk on the city street network. They are distributed over the city according to the mobility weighted through the popularity of local businesses for a simulation period of 24 hours of a working day given by the average of days between Monday and Friday.

B. Simulation Results

Fig. 2 presents the deployment of 8 EDCs with DDA (§ IV-A) and MDA (§ IV-B) approaches in Luxembourg City. Circles and stars represent BSs and EDCs respectively. BSs are assigned to an EDC of the same color. This result unveils that considering the mobility of citizens leads to a significantly different EDC deployment. On the one hand, DDA (see Fig. 2(a)) deploys EDCs so that all the controlled BSs experience a similar distance. On the other side, the MDA approach deploys EDCs among BSs that experience higher computational demands (see Fig. 2(a)). Specifically, with MDA most of the EDCs tend to be deployed closer to

[3] https://data.public.lu/fr/datasets/cadastre-gsm/

[4] https://map.geoportail.lu/



(a) DDA

(b) MDA

Fig. 2. Distributed (DDA) and Mobility-aware (MDA) Deployment Algorithms

the city center and two of them in the north-eastern district of the city (Kirchberg area), which are the most important working and business districts of Luxembourg City and are very crowded during working days, especially at lunchtimes.

Fig. 3 shows the per-hour outage probability in a working day for the proposed approaches with a fixed number of servers per EDC ($N_s = 10$) and varying the number of EDCs. Fig. 3(a) illustrates the outage probability for the DDA approach. MDA clearly outperforms DDA, as shown in Fig. 3(b). Interestingly, the results show that the variation of the number of EDCs shows a different behavior of the two approaches. By increasing the number of EDCs in the city makes the outage probability of decreasing proportionally for DDA. This is not true in MDA as having 9 or 10 EDCs makes little difference.

We now fix the number of deployed EDCs in the city ($N_e = 8$) and investigate in Fig. 4 the per-hour outage probability by varying the number of servers for each EDC. Fig. 4(a) illustrates the DDA approach. The increase of the number of servers per EDC does not decrease the outage probability as it does the deployment of additional EDCs (see result discussed in Fig. 3(a)). With a fixed number of EDCs, MDA still outperforms DDA, as Fig. 4(b) shows.

We now analyze the impact of server allocation in the EDCs (see Subsection IV-C). Fig. 5 compares DDA and MDA with the two allocation policies FNS and PNS. The number of EDCs is set to 8 and the number of servers per EDC is set to 10 for FNS. Fig. 5(a) compares FNS and PNS for DDA. As expected, PNS outperforms FNS for most of the time, especially during busy hours. Surprisingly, the PNS approach at 8:00 and 18:00 performs worse than PNS. The reason is that few servers are assigned to EDCs with low computational demand and it leads to outage when other EDCs around the city do not have requests to satisfy. Thus, to reach a low level of outage probability without wasting resources, it is fundamental to plan an effective deployment of EDCs around the city before planning how to set the number of servers per EDC. To this end, Fig. 5(b) shows that under the MDA approach, i) the outage probability is significantly lower than any DDA strategy and, ii) that the previous insight is not any longer valid as with MDA, PNS is always lower or equal FNS.

VI. Concluding Remarks

This paper tackles the problem of EDCs deployment in urban environments. By considering citizens mobility and their social interactions, we show that overall performance can improve. We model the computational demand and citizens mobility and formulate a problem to minimize the outage probability. Then, we propose two heuristic algorithms. The first one (DDA) deploys EDCs on the sole basis of the spatial distances EDCs-BSs, while the second (MDA) is aware of citizens mobility and the expected computational demand. We further consider two different approaches for server allocation within EDCs. FNS deploys a fixed number of servers per EDC, while PNS deploys servers proportionally to the computational demand. The results show that the policy MDA with PNS makes lower the outages, thereby proving that considering citizens mobility for EDCs deployment in urban environments is effective.

Acknowledgment

Dr. Fiandrino is supported by the Juan de la Cierva grant from the Spanish Ministry of Economy and Competitiveness (FJCI-2017-32309). Dr. Cantelmo is supported by the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No. 754462. Mr. Vitello is supported by the Luxembourg National Research Fund (PRIDE17/12252781/DRIVEN).

References

[1] F. Giust, G. Verin *et al.*, "MEC deployments in 4G and evolution towards 5G," Feb 2018, ETSI White Paper.

[2] T. Taleb, K. Samdanis *et al.*, "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, Third Quarter 2017.

[3] M. Satyanarayanan, "The emergence of edge computing," *Computer*, vol. 50, no. 1, pp. 30–39, Jan 2017.

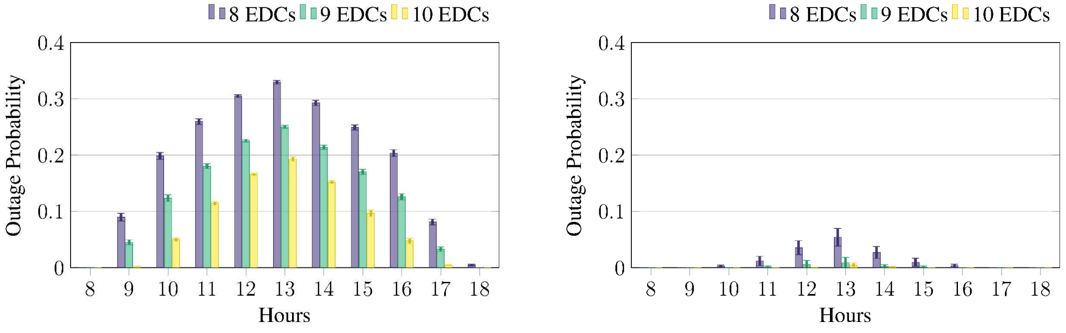
[4] C. Fiandrino, N. Allio *et al.*, "Profiling performance of application partitioning for wearable devices in mobile cloud and fog computing," *IEEE Access*, vol. 7, pp. 12 156–12 166, Jan 2019.

[5] C. Mouradian, D. Naboulsi *et al.*, "A comprehensive survey on fog computing: State-of-the-art and research challenges," *IEEE Communications Surveys Tutorials*, vol. 20, no. 1, pp. 416–464, First Quarter 2018.

[6] C. Andrés Ramiro, C. Fiandrino *et al.*, "openLEON: An end-to-end emulator from the edge data center to the mobile users," in *Proc. of ACM WiNTECH*, 2018, pp. 19–27.

[7] J. Gedeon, J. Krisztinkovics *et al.*, "A multi-cloudlet infrastructure for future smart cities: An empirical study," in *Proc. of ACM EdgeSys*, Jun 2018, pp. 19–24.

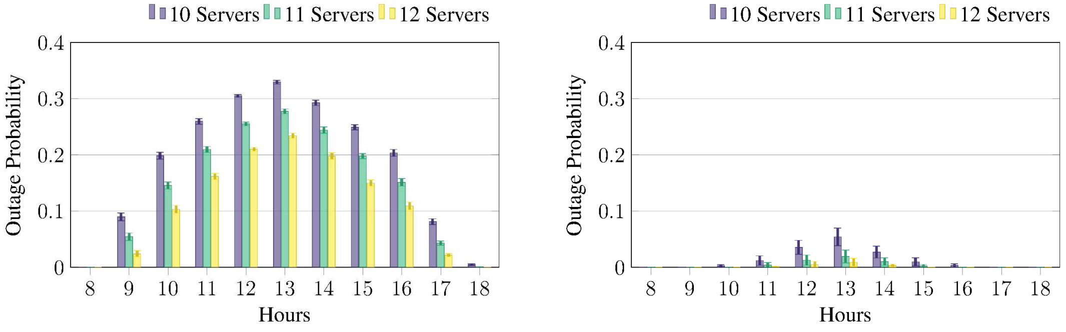
[8] C. Marquez, M. Gramaglia *et al.*, "Not all apps are created equal: Analysis of spatiotemporal heterogeneity in nationwide mobile service usage," in *Proc. of ACM CoNeXT*, 2017, pp. 180–186.



(a) DDA

(b) MDA

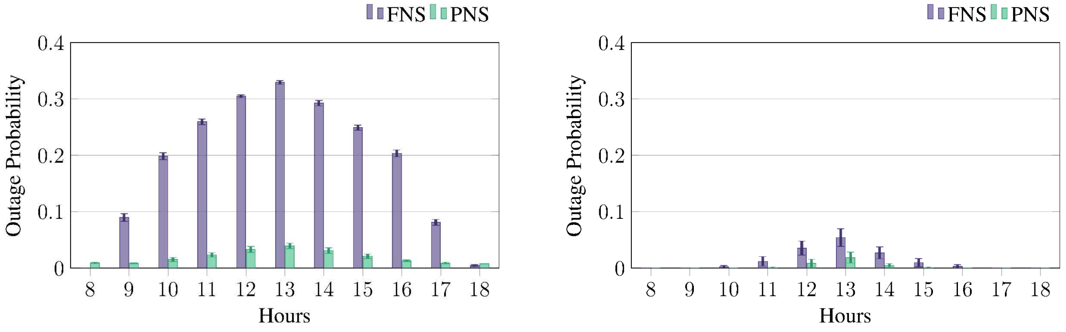
Fig. 3. Total Outage Probability in a working day with a different number of EDCs (number of servers per EDC fixed to 10)



(a) DDA

(b) MDA

Fig. 4. Total Outage Probability in a working day with a different number of servers (number of EDCs deployed in city fixed to 8)



(a) DDA

(b) MDA

Fig. 5. Total Outage Probability in a working day with FNS and PNS approaches

[9] K. Jayarajah, A. Tan *et al.*, "Understanding the interdependency of land use and mobility for urban planning," in *Proc. of ACM UbiComp*, 2018, pp. 1079–1087.

[10] Y. Zhou, B. P. L. Lau *et al.*, "Understanding urban human mobility through crowdsensed data," *IEEE Communications Magazine*, vol. 56, no. 11, pp. 52–59, Nov 2018.

[11] M. C. Filippou, D. Sabella *et al.*, "Flexible MEC service consumption through edge host zoning in 5G networks," *CoRR*, 2019.

[12] J. Patman, P. Lovett *et al.*, "Data-driven edge computing resource scheduling for protest crowds incident management," in *Proc. of IEEE NCA*, Nov 2018, pp. 1–8.

[13] P. Liu, D. Willis *et al.*, "ParaDrop: Enabling lightweight multi-tenancy at the network's extreme edge," in *Proc. of IEEE/ACM SEC*, Oct 2016, pp. 1–13.

[14] R. Morabito, V. Cozzolino *et al.*, "Consolidate IoT edge computing with lightweight virtualization," *IEEE Network*, vol. 32, no. 1, pp. 102–111, Jan 2018.

[15] P. Bellavista, S. Chessa *et al.*, "Human-enabled edge computing: Exploiting the crowd as a dynamic extension of mobile edge computing," *IEEE Communications Magazine*, vol. 56, no. 1, pp. 145–155, Jan 2018.

[16] G. Cardone, A. Cirri *et al.*, "The participact mobile crowd sensing living lab: The testbed for smart cities," *IEEE Communications Magazine*, vol. 52, no. 10, pp. 78–85, October 2014.

[17] J. Xu, B. Palanisamy *et al.*, "Zenith: Utility-aware resource allocation for edge computing," in *Proc. of of IEEE EDGE*, Jun 2017, pp. 47–54.

[18] M. Jia, J. Cao *et al.*, "Optimal cloudlet placement and user to cloudlet allocation in wireless metropolitan area networks," *IEEE Transactions on Cloud Computing*, vol. 5, no. 4, pp. 725–737, Oct 2017.

[19] J. Liang, Z. Chen *et al.*, "Delay outage probability of multi-relay selection for mobile relay edge computing systems," *CoRR*, 2019.

[20] L. M. P. Larsen, A. Checko *et al.*, "A survey of the functional splits proposed for 5G mobile crosshaul networks," *IEEE Communications Surveys Tutorials*, vol. 21, no. 1, pp. 146–172, First Quarter 2019.

[21] L. Kleinrock, *Queueing systems, Volume 1: Theory*. Hoboken, NJ, US, 1975.

[22] C. Fiandrino, A. Capponi *et al.*, "CrowdSenSim: a simulation platform for mobile crowdsensing in realistic urban environments," *IEEE Access*, vol. 5, pp. 3490–3503, Feb 2017.

[23] A. Capponi, C. Fiandrino *et al.*, "A survey on mobile crowdsensing systems: Challenges, solutions and opportunities," *IEEE Communications Surveys Tutorials*, pp. 1–49, Apr 2019.

[24] P. Vitello, A. Capponi *et al.*, "High-precision design of pedestrian mobility for smart city simulators," in *Proc. of IEEE ICC*, May 2018.