Mobile Edge Computing Partial Offloading Techniques for Mobile Urban Scenarios

Arash Bozorgchenani, Daniele Tarchi, Giovanni Emanuele Corazza
Department of Electrical, Electronic and Information Engineering, University of Bologna, Italy
Email: {arash.bozorgchenani2,daniele.tarchi,giovanni.corazza}@unibo.it

***Abstract*—Edge Computing refers to a recently introduced approach aiming to bring the storage and computational capabilities of the cloud to the proximity of the edge devices. Edge Computing is one of the main techniques enabling Fog Computing and Networking. Among several application scenarios, the urban scenario seems one of the most attractive for exploiting edge computing approaches. However, in an urban scenario, mobility becomes a challenge to be addressed, affecting the edge computing. By gaining from the the presence of two types of devices, Fog Nodes (FNs) and Fog-Access Points (F-APs), the idea in this paper is that of exploiting Device to Device (D2D) communications between FNs for assisting computation offloading requests between FNs and F-APs by exchanging status information related to the F-APs. With this knowledge, this paper proposes a partial offloading approach where the optimal tasks amount to be offloaded is estimated for minimizing the outage probability due to the mobility of the devices. In order to reduce the outage probability we have further considered a relaying approach among F-APs. Moreover, the impact of the number of tasks that each F-AP can manage is shown in terms of task processing delay. Numerical results show that the proposed approaches allow to achieve performance closer to the lower bound, by reducing the outage probability and the task processing delay.**

I. Introduction

The continuous rise of mobile applications has led to an exponential growth of demand in high computational capability in wireless cellular networks [1]. Edge computing brings this computational capabilities closer to the users and enables a large number of devices to process their tasks at the network edge instead of transmitting to the centralized cloud infrastructure by saving energy consumption, limiting the traffic to the fronthaul, and providing services with faster response. Among different scenarios, mobility and computation offloading, which are largely served within the bound of the network edge, have been adopted in internet of vehicles [2], [3]. Edge computing is also considered one of the fundamental techniques of the Fog Networking, where the focus is more on the architectural point of view, in particular toward Internet of Things (IoT) applications [4]. In this paper a partial offloading technique for edge computing environments is proposed to be used in a mobile urban scenario.

The research community is very active on computation offloading in mobile edge computing. The authors in [5] have considered the effect of mobility, users' local load and availability of cloudlets for developing an optimal offloading algorithm and compared the performance in case of always performing computation locally, always offloading or randomly selecting one of these modes. The idea of exploiting fog networking concepts applied to vehicular environment seems also a promising trend. The authors in [6] propose a vehicular fog computing infrastructure in which vehicles with more resources are considered as the computational infrastructure, to relieve the burden of the congested resource limited vehicles. In [7] a local roadside cloud-based network is proposed to deal with traffic-related data. A mobility-aware offloading decision strategy exploiting genetic algorithm for a single job, multi component is proposed in [8] to improve offloading success rate and decrease energy consumption.

In this work we have considered a partial offloading technique in an urban vehicular environment at the network edge, by considering two main types of device: Fog Nodes (FNs), smart mobile devices generating the tasks to be processed, and the Fog-Access Points (F-APs), devices able to process the offloaded tasks. In cloud computing, the users are able to offload their tasks to the centralized cloud, however, in some cases, e.g., for real time applications, the delay from centralized cloud might not be acceptable. On the other side, in edge computing, the FNs are able to exploit the other FNs and the F-APs for offloading their computational tasks and reduce the amount of traffic sent to the centralized cloud [9], [10]. Due to the storage and energy limitation of the FNs, it is not always feasible to consider direct FNs to FNs offloading; as a result, in this paper, we are considering that FNs are able to offload to the F-APs.

On the other hand, computational offloading in a mobile environment is a challenging issue, mainly due to the devices mobility. To this aim, the idea at the basis of this paper is that of exploiting FNs to FNs communications (e.g., through Device to Device (D2D) connections) for updating the FNs about the status of the system. By leveraging on a similar concept introduced in [11], an idea could be that of employing the D2D communications among FNs for sharing those parameters needed for optimally estimating the amount of data that can be offloaded to the nearby F-APs while respecting the constraints imposed by the mobility. To this aim the network

This work has been partially supported by the project "GAUChO - A Green Adaptive Fog Computing and Networking Architecture" funded by the MIUR Progetti di Ricerca di Rilevante Interesse Nazionale (PRIN) Bando 2015 - grant 2015YPXH4W_004.

can be seen as composed by two logical connections: a control plane among FNs and a data plane between FNs and F-AP for implementing the task offloading. We have here considered the possibility to have two types of F-APs, fixed and mobile. In order to reduce the outage probability due to a delayed response from the F-APs computing the offloaded task, a relaying policy has also been considered between mobile and fixed F-APs. Furthermore, we have investigated the impact of the amount of tasks, that each F-AP can manage, on the task processing delay.

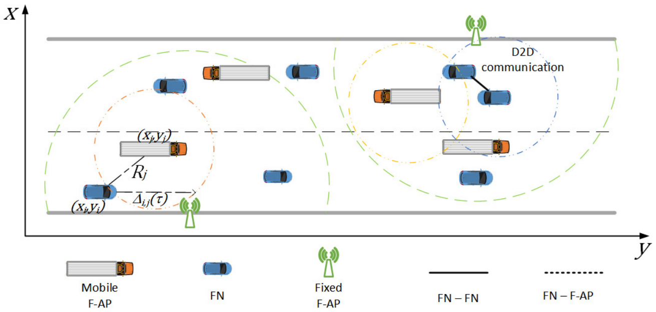

Fig. 1. Partial offloading mobile urban scenario.

II. System Model and Problem Formulation

In this work a two layer Fog architecture for edge computing is considered. On one hand, $\mathcal{U} = \{u_1, \ldots, u_i, \ldots, u_N\}$ represents the set of FNs in the first layer. All the FNs have computational and storage capabilities; FNs can communicate among them within a specific range depending on the deployed wireless technology. On the other hand, in the second layer, there are two types of F-APs, fixed and mobile. The set of mobile F-APs is shown as $\mathcal{C} = \{c_1, \ldots, c_m, \ldots, c_M\}$, and fixed F-APs as $\mathcal{F} = \{f_1, \ldots, f_k, \ldots, f_K\}$. Fixed F-APs have higher computational and storage capabilities comparing with mobile F-APs and they both have higher capabilities comparing with the FNs. F-APs are able to communicate with the FNs and compute the offloaded tasks. The fixed F-APs have a wider coverage range comparing with the FNs and the mobile F-APs, and are able to aggregate the FNs' traffic requests, while mobile F-APs and FNs are supposed to have the same coverage range.

Each FN having a task to be computed can have different choices: perform a local computation, offload to either a fixed or mobile F-AP in proximity or partially offload to the F-APs; the goal of the proposed partial offloading technique is to estimate the amount of data to offload in order to minimize the outage probability and the task processing delay. In our work, the outage probability corresponds to the probability that an offloaded task cannot be received back by the offloading FN due to the devices mobility, while the task processing delay, corresponds to the time needed for processing the task by taking into account both local and offloaded amount.

We have considered a street scenario, as shown in Fig. 1, where the generic ith car, acting as FN, can move with velocity $\vec{v}_i$ in two directions: left to right or the reverse depending on the lane they are located. Likewise, the mth mobile F-AP, which can be a bus or truck, is moving with a velocity $\vec{v}_m$ in a direction depending on the lane they are located [12]. Moreover, there are some fixed F-APs (e.g., located on light poles) at the roadside with a broader coverage area to cover the street when there is no mobile F-APs available. The priority from each FN is offloading to the mobile F-APs, and, then to the fixed F-APs.

In general, the computational time for the lth task by any device is defined as:

$$T_c^l = O_l/\eta_c \tag{1}$$

where O_l represents the number of operations required for computing the lth task and η_c is the Floating-point Operation Per Second (FLOPS) depending on the CPU of the processing device, which can be an FN or an F-AP.

In case of offloading, each task should be transmitted, hence, the transmission time for the lth task can be written as:

$$T_{tx,ij}^l = L_{s_l}/r_{ij} \tag{2}$$

where L_{s_l} is the size of the lth task requested by the ith FN and r_{ij} is the data rate of the link between the ith FN and the jth F-AP which could be either fixed or mobile. Later the result of the processed task should be sent back to the ith FN, leading to a reception time defined as:

$$T_{rx,ij}^l = L_{r_l}/r_{ij} \tag{3}$$

where L_{r_l} is the size of the result of the requested task sent back from the F-AP to the offloading FN, when we suppose a symmetric channel in terms of data rate between the ith FN and the jth F-AP. Each F-AP is supposed to have a buffer holding the tasks of the requesting FNs to be processed. The waiting time of the lth task at the jth F-AP can be defined as:

$$T_{w_j}^l(p) = \sum_{\lambda=1}^{p-1} T_{c_j}^{\lambda} \tag{4}$$

where p is the number of tasks already in the queue of the jth F-AP. The waiting time for the task to be processed plus the computing time at the F-AP corresponds to the FN idle time when the FN waits for the result back.

The concept behind partial offloading is to delegate only a portion of the computational load to another device to optimize energy and time [13]. We define α_l as the portion of the lth task that is offloaded. As a result, the time required for offloading a task can be written as the sum of the time for sending the portion of the task, the time the task should wait in the F-AP processing queue, the time for computing that task at the F-AP and the time needed for having the result back:

$$T_{off,i}^l(\alpha_l) = \alpha_l T_{tx,ij}^l + T_{w_j}^l + \alpha_l T_{c_j}^l + \alpha_l T_{rx,ij}^l \tag{5}$$

while the time for local computation, can be defined as the time needed for computing the remaining portion of the task:

$$T_{loc,i}^l(\alpha_l) = (1 - \alpha_l) T_{c_i}^l \tag{6}$$

Thus, in case of partial offloading, the total delay for processing a task can be rewritten as the maximum of the two delays, i.e.,

$$D_i^l(\alpha_l) = \max\{T_{off,i}^l(\alpha_l), T_{loc,i}^l(\alpha_l)\} \tag{7}$$

In order to estimate the amount of data that can be offloaded we have to estimate the amount of time that the ith FN remains under the coverage of the jth F-AP for avoiding to have the result back when the FN is out of coverage. The remaining distance before going out of the coverage of the jth F-AP at time instant τ, as defined in [11], is equal to:

$$\Delta_{i,j}(\tau) = \sqrt{R_j^2 - (y_j(\tau) - y_i(\tau))^2} \pm (x_j(\tau) - x_i(\tau)) \tag{8}$$

where $\{x_i(\tau), y_i(\tau)\}$ and $\{x_j(\tau), y_j(\tau)\}$ are, respectively, the position of the ith FN and the jth F-AP at time τ and R_j is the radius of the jth F-AP's coverage area[1]. Thus, the time that the ith FN remains in the coverage area of the jth F-AP (i.e., sojourn time) can be written as:

$$\bar{T}_\tau^{i,j}(\alpha_l) = \frac{\Delta_{i,j}(\tau)}{\partial v_{ij}} \tag{9}$$

where $\partial v_{ij} = |\vec{v}_i - \vec{v}_j|$ is the modulo of the vector speeds of the ith FN and jth F-AP taking into account their relative direction. The outage for the lth task of the ith FN can be defined as:

$$\Omega_l^i(\alpha_l) = \begin{cases} 1 & \text{if } \bar{T}_\tau^{i,j}(\alpha_l) < T_{off,i}^l(\alpha_l) \\ 0 & \text{if } \bar{T}_\tau^{i,j}(\alpha_l) \geq T_{off,i}^l(\alpha_l) \end{cases} \tag{10}$$

corresponding to the occurrence that, due to the FNs and F-APs mobility, the time needed for offloading a task is higher than the FN sojourn time within the F-AP coverage area. Having the goal of minimizing the outage probability and processing delay, we define our minimization problem as:

$$\begin{cases} \min_{\boldsymbol{\alpha}} \left\{ \sum_{i=1}^N \sum_l \Omega_l^i(\alpha_l) \right\} \\ \min_{\boldsymbol{\alpha}} \left\{ \sum_{i=1}^N \sum_l D_i^l(\alpha_l) \right\} \end{cases} \tag{11}$$

subject to

$$T_{c_i}^l > T_{c_q}^l > T_{c_k}^l > T_{tx,ij}^l > T_{rx,ij}^l > 0 \tag{12}$$

$$R_m < R_k \tag{13}$$

$$|\vec{v}_j| < |\vec{v}_i| \tag{14}$$

$$0 \leq \alpha_l \leq 1 \tag{15}$$

where $\boldsymbol{\alpha}$ is the set of the offloaded portion of all the tasks in a given time instant.Hence, there are two objectives in the formulation, i.e., minimizing the sum of the tasks not successfully received due to devices mobility during the offloading phase and the sum of all the total delays suffered by all of the tasks, respectively shown in (11). Constraint (12) introduces the hypothesis that the FNs computing time is higher than that of the mobile F-APs, that is even higher than that of the fixed F-APs. All these computational times are supposed to be higher than FNs transmission and receiving times. Constraint (13) set the fixed F-APs coverage area higher than the mobile F-APs. Constraint (14) means that the velocity of the FNs is higher than that of the mobile F-APs. Finally, the offloaded portion is always between 0 and 1 as shown in constraint (15).

In the following, we resort to a suboptimal solution by relaxing some of the hypotheses and employing D2D communications among FNs for sharing information to be used for the partial offloading estimation.

III. D2D Assisted Partial Offloading

The optimization procedure is based on evaluating a closed form expression for the optimized $\boldsymbol{\alpha}$ by relaxing some of the problem constraints. However, due to the mobility of FNs, some of the parameters cannot be considered as known by FNs. Hence, we aim at exploiting D2D communications among FNs to exchange information related to the status of the F-APs (i.e., waiting time, node position and direction, velocity). Then, the estimated information is used by the FNs to calculate the amount of data to be offloaded. In the end a relaying method between mobile and fixed F-APs is also proposed in order to reduce the outage probability.

A. Ideal partial offloading estimation

As a first step for the optimization procedure the F-APs within the coverage area of a given FN are selected as potential candidates for offloading. All FNs prioritize the mobile F-APs in the network edge for offloading, and if there is no mobile F-AP they will offload the task to a fixed F-AP. In order to minimize the outage probability the ith FN having a task to be processed selects the F-AP allowing to maximize the sojourn time within its coverage area. Hence, the jth F-AP is selected such that:

$$\max_j \left\{ \psi_\tau^{i,j}(\alpha_l) \right\} = \max_j \left\{ \bar{T}_\tau^{i,j}(\alpha_l) - T_{w_j}^l \right\} \tag{16}$$

corresponding to select the F-AP with the highest available time $\psi_\tau^{i,j}$, that is a function of both sojourn time (9) and task waiting time (4) in the buffer of the F-APs due to previous ongoing computations. It is worth to be noticed that the sojourn time (9) is a function of velocities and directions of both ith and jth devices.

In order to minimize the outage probability, we aim at optimizing the portion of the tasks to be offloaded. To avoid outage, the offloading time of the task portion from an FN should be less than the sojourn time in the coverage area of the selected F-AP for offloading, as shown in the second condition in (10). To find the portion of the lth task which can be offloaded considering the offloading time and the velocity, exploiting (8) and (9), we can rewrite the second condition in (10), corresponding to no outage, as:

$$\alpha_l \frac{L_{s_l}}{r_{ij}} + T_{w_j}^l + \alpha_l \frac{O_l}{\eta_{c_j}} + \alpha_l \frac{L_{r_l}}{r_{ij}} \leq \frac{\Delta_{i,j}(\tau)}{\partial v_{ij}} \tag{17}$$

that allows to find the optimal α_l parameter, as:

$$\alpha_l \leq \frac{\Delta_{i,j}(\tau) - T_{w_j}^l \cdot \partial v_{ij}}{\partial v_{ij} \cdot \left\{ \frac{L_{s_l}}{r_{ij}} + \frac{O_l}{\eta_{c_j}} + \frac{L_{r_l}}{r_{ij}} \right\}} \tag{18}$$

[1] In case the vehicles are in the lower lane the operator between the first and the second term is +, otherwise is -.

The above condition allows to minimize the outage condition by setting an upper limit on the amount of data to be offloaded. However, the reliability of the calculated α_{off}^{l} parameter depends on the knowledge of some input information, i.e., direction and velocity, and task waiting time in the F-AP computing buffer.

B. D2D assisted information sharing

In order to know the parameters to be used for estimating (18) we rely on the D2D communications among FNs that is used for sharing information related to waiting time, velocity and direction of movement.

Hence, we suppose that when an FN receives back the result of its offloaded task, it is also able to estimate the amount of time the task has waited in the queue of that specific F-AP, as well as its velocity and direction, and also the time instant this information has been estimated, corresponding to τ. The updated set of information at time instant τ of the information obtained by the ith FN from the jth F-AP corresponds to $\left\{\tilde{T}_{w_j}(\tau), \tilde{\vec{v}}_j(\tau)\right\}$, where $\tilde{T}_{w_j}(\tau)$ corresponds to the waiting time in the jth F-AP and $\tilde{\vec{v}}_j(\tau)$ corresponds to the velocity and direction of the jth F-AP, both estimated by ith FN at time instant τ.

In the proposed idea as two FNs are approaching, they update their set by comparing the time in which the information regarding the corresponding F-AP has been updated in order to record only the most recent values. If the sender's updating time is more recent, the information about that F-AP will be updated in the recipient FN's set. This corresponds to say that the information in the buffer of each FN, can be written as:

$$\mathcal{B}_i = \left\{\tilde{T}_{w_j}(\bar{\tau}), \tilde{\vec{v}}_j(\bar{\tau}) | \bar{\tau} = \max_{\iota}(\tau_\iota), d_{i\iota} \leq R_i\right\} \; \forall j \tag{19}$$

where $\bar{\tau}$ is the maximum updating time instant, i.e., the most recent time instant, among all the approaching FNs that are in the D2D coverage area of the ith FN, that is equal to R_i, while $d_{i\iota}$ us the distance between the ith and the ιth FNs. Information related to the waiting time, direction and velocity of each F-AP is spread out through the D2D connections whenever FNs are approaching and used as an input for estimating α_l for minimizing the outage probability and the task processing delay.

In order to see the impact of the parameters in (19) on the results, we are considering two types of information spread among the FNs. In case the information related to the velocity and direction of F-APs is spread through the FNs, by exploiting (18), we could rewrite the offloading parameter estimation as:

$$\dot{\alpha}_l \leq \frac{\Delta_{i,j}(\tau)}{\partial \tilde{v}_{ij} \cdot \left\{\frac{L_{s_l}}{r_{ij}} + \frac{O_l}{\eta_{c_j}} + \frac{L_{r_l}}{r_{ij}}\right\}} \tag{20}$$

while, when the waiting time is also spread through the FNs D2D connection, by exploiting (18), we could rewrite the offloading parameter estimation as:

$$\ddot{\alpha}_l \leq \frac{\Delta_{i,j}(\tau) - \tilde{T}_{w_j}^{l} \cdot \partial \tilde{v}_{ij}}{\partial \tilde{v}_{ij} \cdot \left\{\frac{L_{s_l}}{r_{ij}} + \frac{O_l}{\eta_{c_j}} + \frac{L_{r_l}}{r_{ij}}\right\}} \tag{21}$$

where $\partial \tilde{v}_{ij}$ is the estimated velocity modulo of the vector difference between ith FN and jth F-AP.

C. F-AP Relaying

After the task computation by the mobile F-AP, the result will be sent to both the FN and the nearest fixed F-AP, so that in case the result can not be received due to the devices mobility, the fixed F-AP with its broader coverage area will send the result back to the requesting FN. This will lead to a significant reduction of outage probability. In this case, the outage becomes:

$$\hat{\Omega}_l^i(\alpha_l) = \begin{cases} 1 & \text{if } \bar{T}_\tau^{i,k} < \hat{T}_{off,i}^{l} \\ 0 & \text{if } \bar{T}_\tau^{i,k} \geq \hat{T}_{off,i}^{l} \end{cases} \tag{22}$$

by considering the sojourn time of the ith FN in the coverage area of the kth fixed F-AP, $\bar{T}_\tau^{i,k}$, while the delay for the offloading phase becomes:

$$\hat{T}_{off,i}^{l}(\alpha_l) = \begin{cases} \alpha_l T_{tx,im}^{l} + T_{w_m}^{l} + \alpha_l T_{c_m}^{l} \\ \qquad + \alpha_l T_{tx,mk}^{l} + \alpha_l T_{rx,ik}^{l} & \text{(23a)} \\ \alpha_l T_{tx,ij}^{l} + T_{w_j}^{l} + \alpha_l T_{c_j}^{l} + \alpha_l T_{rx,ij}^{l} & \text{(23b)} \end{cases}$$

where the first equation refers to the case with relaying while the second one for the case with no relaying; the additional term in the relaying delay is due to the transmission time between the mth mobile F-AP and the kth fixed F-AP.

IV. Numerical Results

In this section, the numerical results obtained through computer simulations in Matlab are presented; the parameters used for the scenario are shown in Tab. I. The computer simulations are carried out in terms of average task delay and outage probability, defined as:

- Average Task Delay: The average time spent for offloading or for performing the local computation (See (7)).
- Outage probability: The average probability of number of unsuccessful receptions by FNs, due to devices mobility, over total number of generated tasks (See (10) and (22)).

In this section we will compare the performance of the D2D approaches with a benchmark that considers to know perfectly all the needed parameters, and labeled as ideal. The comparison is done with two possible D2D approaches, taking into account the impact of the information spread through the nearby FNs on the performance. In particular, when we suppose that the FNs share the information only about the velocity and direction of movement of the F-APs, as defined in (20), the scenario is labeled as D&V, while when the information regarding the waiting time is also known, as defined in (21), it is labeled as D&V&T_w.

We have compared the performance of these three approaches in terms of delay and outage probability for different number of FNs and F-APs, by considering a task generation

TABLE I
SIMULATION PARAMETERS

Parameter	Value
Dimension	500m x 20m
Task size (L_s)	5 MB
Task result size (L_r)	1 MB
Channel Model	Extended Vehicular A model (EVA) [14]
FN and mobile F-AP coverage range (R_i, R_m)	15 m
Fixed F-AP coverage range (R_k)	50 m
Task Operation (O_l)	50G
FN Flops (η_{c_i})	15G FLOPS (= 1 CPU)
Mobile F-AP Flops (η_{c_j})	30G FLOPS (= 2 CPUs)
Fixed F-AP Flops (η_{c_j})	60G FLOPS (= 4 CPUs)
FN Velocity ($\|v_i\|$)	[8-12] m/s
Mobile F-AP Velocity ($\|v_m\|$)	[5-7] m/s

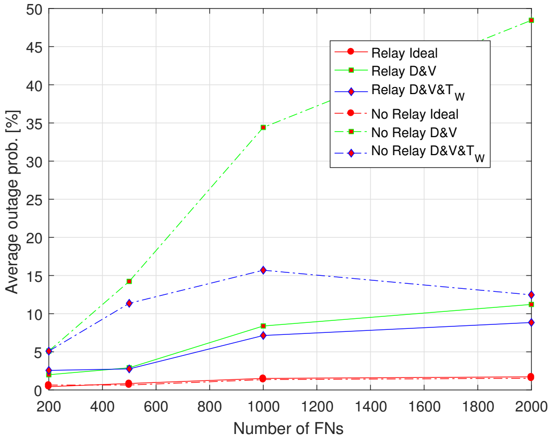

Fig. 2. Outage Probability of 11 fixed F-APs with high capacity

rate equal to 0.1 task per second. In the following we will refer as computational capacity as the amount of task that each F-AP can manage; in particular with low computational capacity each F-AP can backlog an amount of tasks equal to the number of CPUs (i.e., p=2 per CPU), while with high computational capacity can backlog two task for each CPU (i.e., p=3 per CPU). Moreover the effect of the relaying between mobile and fixed F-APs is considered. The location of 20 mobile F-APs is generated randomly; 10 of them are on the right lane while and 10 in the left lane.

Figs. 2 and 3 depict the average outage probability of the FNs for different scenarios in the presence and absence of relaying among the F-APs. As seen, whenever more information regarding F-APs waiting time, velocity and direction is known, the performance is better because of the better estimation of the portion to be offloaded. Moreover, when there is a relay among the F-APs, the outage probability is reduced, and this is due to the receiving back the result from the fixed F-APs because of the higher coverage area. Furthermore, we can notice that in Fig. 2 where the F-APs have higher capacity, the average outage probability is slightly decreased comparing with Fig. 3 where the F-APs have lower capacity.

The average outage probabilities when there are 5 fixed F-APs are depicted in Figs. 4 and 5. The performance order of the techniques is the same as for 11 fixed F-APs, however, it can be noticed that when number of fixed F-APs decreases from 11 to 5, the outage probability increases. This is because fixed F-APs have a broader coverage area and higher computational capabilities and by having fewer of them in the scenario more tasks will be offloaded to the mobile F-APs which increases the outage probability. Furthermore, in Fig. 5 where the F-APs can manage more tasks, the outage probability is lower comparing with Fig. 4 in which the capacity of the F-APs is lower.

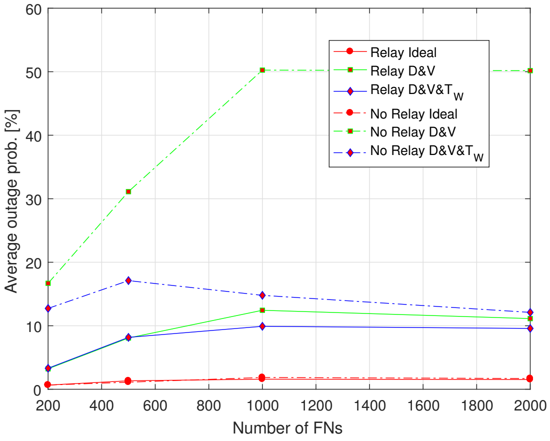

Fig. 3. Outage Probability of 11 fixed F-APs with low capacity

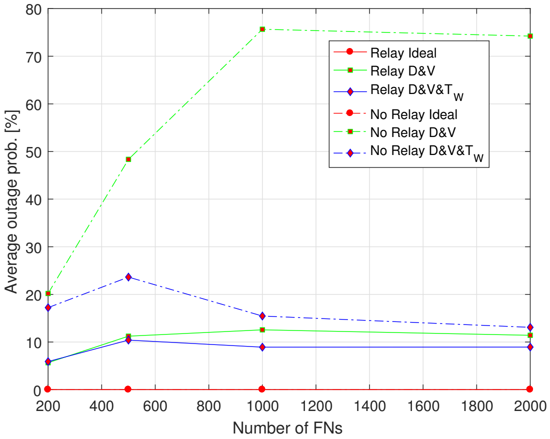

Fig. 4. Outage Probability of 5 fixed F-APs with low capacity

Figs. 6 and 7 depict the average task delay of the network with 11 fixed F-APs where there is, respectively, a relay among the F-APs in the first one and there is no relay in the second one. It can be seen that relaying does not have an impact on the delay, however, delay is highly influenced by the capacity of the F-APs. When the F-APs have higher capacity more tasks can be processed and kept in the queue which will result in parallel computation in F-APs and local computation in FNs, when partial offloading, which will result in a lower delay.

The simulation results underscore the impact of the proposed estimation approach and employing the D2D commu-

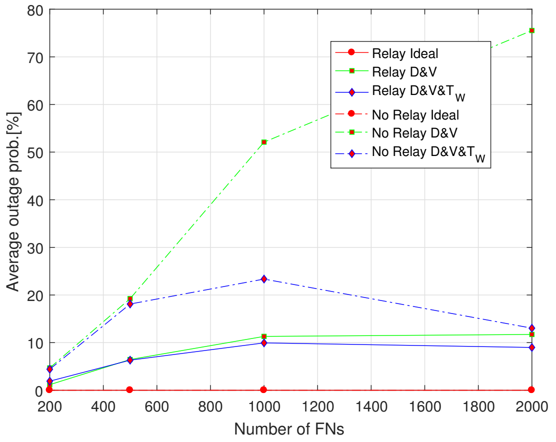

Fig. 5. Outage Probability of 5 fixed F-APs with high capacity

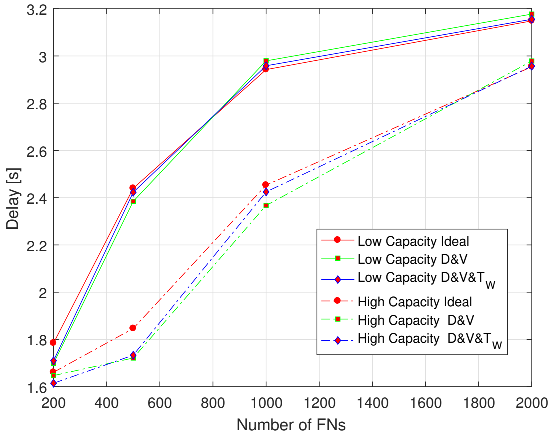

Fig. 6. Average task delay without relaying through 11 fixed F-APs

nication on the performance in terms of outage probability and delay. It is proved that the knowledge about waiting time, velocity and direction of the other nodes can greatly impact the accuracy of the estimation of the offloaded portion. By having a D2D communication for informing the other FNs about the status of the F-APs, FNs are able to better estimate how much they can offload in order to have the lowest amount of delay and outage probability.

V. Conclusions

The partial offloading problem in mobile edge computing in a mobile urban scenario with FNs and F-APs mobility is considered. FNs consider the remaining time in the coverage for the selection of an F-AP and estimate the portion of task to offload in order to avoid outage. By using a D2D communication the information of the F-APs among FNs is spread, allowing to better estimate the task offloading portion. A relaying technique is also proposed for minimizing the outage. Simulation results demonstrate that by benefiting from the D2D communication and relaying the result among F-APs, outage probability is minimized. Moreover, the impact of the F-APs capacity on the average task delay is shown.

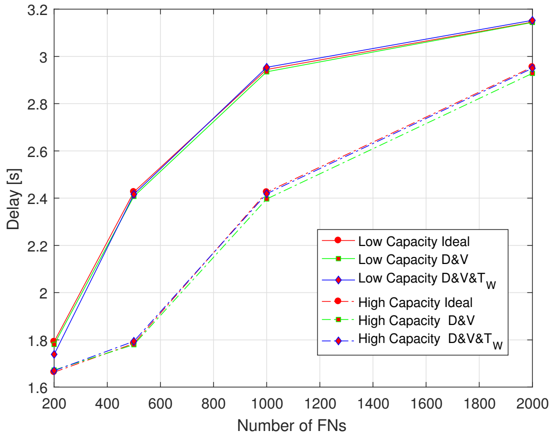

Fig. 7. Average task delay with relaying through 11 fixed F-APs

References

[1] S. Abolfazli, Z. Sanaei, E. Ahmed, A. Gani, and R. Buyya, "Cloud-based augmentation for mobile devices: Motivation, taxonomies, and open challenges," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 337–368, First Quarter 2014.

[2] S. M. Oteafy and H. S. Hassanein, "IoT in the fog: A roadmap for data-centric IoT development," *IEEE Commun. Mag.*, vol. 56, no. 3, pp. 157–163, mar 2018.

[3] W. Zhang, Z. Zhang, and H. C. Chao, "Cooperative fog computing for dealing with big data in the internet of vehicles: Architecture and hierarchical resource management," *IEEE Commun. Mag.*, vol. 55, no. 12, pp. 60–67, Dec. 2017.

[4] M. Chiang and T. Zhang, "Fog and IoT: An overview of research opportunities," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 854–864, Dec. 2016.

[5] Y. Zhang, D. Niyato, and P. Wang, "Offloading in mobile cloudlet systems with intermittent connectivity," *IEEE Trans. Mobile Comput.*, vol. 14, no. 12, pp. 2516–2529, Dec. 2015.

[6] X. Hou, Y. Li, M. Chen, D. Wu, D. Jin, and S. Chen, "Vehicular fog computing: A viewpoint of vehicles as the infrastructures," *IEEE Trans. Veh. Technol.*, vol. 65, no. 6, pp. 3860–3873, Jun. 2016.

[7] R. Yu, Y. Zhang, S. Gjessing, W. Xia, and K. Yang, "Toward cloud-based vehicular networks with efficient resource management," *IEEE Netw.*, vol. 27, no. 5, pp. 48–55, Sep. 2013.

[8] Y. Shi, S. Chen, and X. Xu, "MAGA: A mobility-aware computation offloading decision for distributed mobile cloud computing," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 164–174, Feb 2018.

[9] S. Yan, M. Peng, and W. Wang, "User access mode selection in fog computing based radio access networks," in *2016 IEEE International Conference on Communications (ICC)*, Kuala Lumpur, Malaysia, May 2016.

[10] M. Peng, S. Yan, K. Zhang, and C. Wang, "Fog-computing-based radio acees networks: issues and challenges," *IEEE Netw.*, vol. 30, pp. 46–53, July 2016.

[11] A. Bozorgchenani, D. Tarchi, and G. E. Corazza, "A control and data plane split approach for partial offloading in mobile fog networks," in *2018 IEEE Wireless Communications and Networking Conference (WCNC)*, Barcelona, Spain, Apr. 2018.

[12] D. Huang and H. Wu, *Mobile Cloud Computing: Foundations and Service Models*. Cambridge, MA, USA: Morgan Kaufman - Elsevier, 2018.

[13] D. Mazza, D. Tarchi, and G. E. Corazza, "A partial offloading technique for wireless mobile cloud computing in smart cities," in *2014 European Conference on Networks and Communications (EuCNC)*, Bologna, Italy, Jun. 2014.

[14] *3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA); Base Station (BS) radio transmission and reception*, 3GPP TS 36.104.