A control and data plane split approach for partial offloading in mobile fog networks

A Control and Data Plane Split Approach for Partial Offloading in Mobile Fog Networks

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*—Fog Computing offers storage and computational capabilities to the edge devices by reducing the traffic at the fronthaul. A fog environment can be seen as composed by two main classes of devices, Fog Nodes (FNs) and Fog-Access Points (F-APs). At the same time, one of the major advances in 5G systems is decoupling the control and the data planes. With this in mind we are here proposing an optimization technique for a mobile environment where the Device to Device (D2D) communications between FNs act as a control plane for aiding the computational offloading traffic operating on the data plane composed by the FN - F-AP links. Interactions in the FNs layer are used for exchanging the information about the status of the F-AP to be exploited for offloading the computation. With this knowledge, we have considered the mobility of FNs and the F-APs' coverage areas to propose a partial offloading approach where the amount of tasks to be offloaded is estimated while the FNs are still within the coverage of their F-APs. Numerical results show that the proposed approaches allow to achieve performance closer to the ideal case, by reducing the data loss and the delay.**

I. Introduction

The increase in mobile applications has led to an exponential growth of demand in high computational capability in wireless cellular networks [1]. To address the computational capability issue, Cloud-Radio Access Network (C-RAN) system architecture has been proposed to enable mobile devices put their computational burden in the cloud [2]. To reduce the latency in C-RAN and perform some further enhancement, Fog Radio Access Networks (F-RANs), which can be considered as an extension of C-RAN, has been proposed as a promising solution [3].

Unlike C-RAN architecture that performs the computation in the centralized cloud, F-RANs enables to process part of the signals closer to the network edge. This technique is known as fog computing. Fog Nodes (FNs), that are smart mobile devices in F-RANs, can either offload their tasks to the neighboring FNs, by exploiting Device-to-Device (D2D) communications, or offload to the smart remote radio heads called Fog-Access Points (F-APs) to reduce the amount of traffic sent to the centralized cloud [4], [5]. However, transmissions to neighboring FNs and centralized cloud has its own drawbacks. In one hand, in some cases, e.g., for real time applications, the delay from centralized cloud might not be acceptable. On the other hand, due to the limitation in the cache of the FNs and the high energy required for an FN to feed another FN, it is not always feasible to perform an edge computing task between FNs in a D2D communication. As a consequence, in this paper best effort is made to optimize the amount of tasks to be offloaded, while exploiting the D2D communications between FNs for assisting the process. In particular by exploiting one of the most recent trends within the 5G standardization, we aim at defining a solution where the control and the data planes are split [6]. To this aim the D2D communications between FN can be seen as operating at the control plane, allowing to share the information about the status of the F-AP, while the data plane is constituted by the links between the FN and the F-AP, whose utilization is based on the control plane information.

The research community is very active on computation offloading in fog computing. Computation offloading and interference management in wireless cellular networks with mobile edge computing has been investigated in [7]. The issue of load balancing in fog computing has been addressed in [8] by processing the requests locally in small cell clusters. The coverage probability and ergodic rate with three user access modes were analyzed in [4] in a F-RAN environment. One work closest to our is [9]. They consider 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.

In this work, we have considered F-APs' coverage area and users' mobility to estimate the portion of a task which can be offloaded to have the result back from the same F-AP in a partial offloading problem. We have utilized the D2D communication in our scenario for informing the other FNs about the load of F-APs when there is a computation offloading which is performed in a FN - F-AP mode. The proposed approach results to minimize the data loss due to the mobility by exploiting the network status information shared through the D2D links.

II. System Model

In this work a two layer architecture for fog 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 which should be exploited in a

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.

proper way; FNs can communicate among themselves within a specific range depending on the deployed wireless technology. On the other hand, in the second layer, there are some F-APs, whose set is shown as $\mathcal{C} = \{c_1, \ldots, c_j, \ldots, c_M\}$, with higher computational and storage capabilities able to communicate with the FNs. The F-APs have a wider range of communication comparing with the FNs and are able to aggregate the FNs' traffic requests.

Herein FNs are considered to be mobile devices with the possibility of offloading their tasks to the upper layer F-APs for computation. Due to the mobility of FNs and lack of knowledge about the load of the task queue in each F-AP, we have considered a D2D communication among FNs to exchange information about the load of the F-APs they are aware of. The focus in our work is to find a portion of a task that can be requested for processing to the F-APs while the requested FN is within the coverage of same F-AP to receive the result back. In this way, no exchange of data between F-APs is required to send the result of the requested task to the FN which leads to a lower delay. To this aim, each FN having a task to be computed can have different choices: perform a local computation, offload to an F-AP in proximity or partially offload to the F-AP; the goal of the proposed partial offloading technique is to estimate the amount of data to be offloaded in order to minimize the data loss and the task processing delay. In our work, a task is assumed to be lost if the requesting FN goes out of the coverage of the F-AP which is processing the task. By focusing on the data plane, each FN can be in one of four possible states $\mathcal{S} = \{tx, rx, comp, id\}$: transmitting, receiving, computing or idle. While the first two states are referred to the interaction with an F-AP, the computing state refers to the computation performed locally by the FN itself, while the idle state refers to the idling occurring otherwise (i.e., while waiting for the task in process offloaded to the F-AP or, in any case, if the FN has no task to be processed).

We have considered a street scenario, as shown in Fig. 1, where pedestrians, acting as FNs, can move with velocity v_i in two directions of left to right or the reverse. The coverage area of the F-APs partially overlap to cover the whole area.

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

$$T^l_{comp} = O_l / \eta_{comp} \tag{1}$$

where O_l represents the number of operations required for computing the lth task and η_{comp} is the Floating-point Operation Per Second (FLOPS) which depends 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:

$$T^l_{tx,i} = L_{s_l} / r_{ij} \tag{2}$$

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

$$T^l_{rx,i} = L_{r_l} / r_{ij} \tag{3}$$

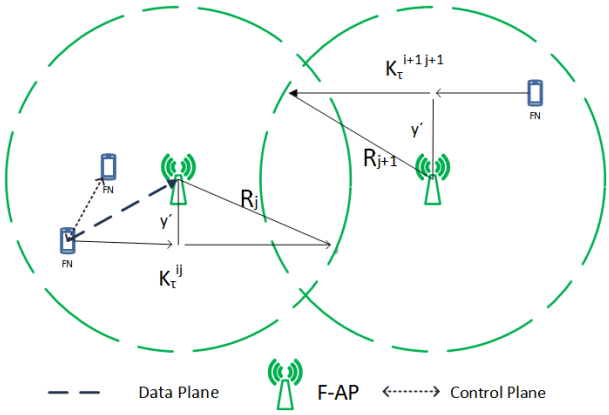

Fig. 1. The application scenario with data and control plane communications

where L_{r_l} is the size of the result of the requested task sent back from the F-AP to the source FN, when we suppose a symmetric channel in terms of data rate between the ith FN and the jth F-AP. By considering the Shannon capacity formula, the data rate between the ith FN and the jth F-AP can be defined as:

$$r_{ij} = B_{ij} \log_2 \left(1 + \frac{P^i_{tx}}{L(d_{ij}) P_{N_j}}\right) \tag{4}$$

where B_{ij} represents the bandwidth of the link, P^i_{tx} is the transmission power of the ith FN, $L(d_{ij})$ is the path loss at a distance d_{ij} between the ith FN and the jth F-AP and P_{N_j} is the noise power. Noise power can be defined as $P_{N_j} = N_T B_{ij}$, where N_T is the thermal noise. Each F-AP is supposed to have a queue 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:

$$T^l_{w_j}(p) = \sum_{\lambda=1}^{p-1} T^\lambda_{comp_j} \tag{5}$$

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 computation load to another device to optimize energy and time [10]. We define α^l_{loc} as the portion of the lth task that can be performed locally and α^l_{off} as the amount that can be offloaded where $\alpha^l_{off} = 1 - \alpha^l_{loc}$. As a result, the time needed 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:

$$T^l_{off,i}(\alpha^l_{off}) = \alpha^l_{off} T^l_{tx,i} + T^l_{w_j} + \alpha^l_{off} T^l_{comp_j} + \alpha^l_{off} T^l_{rx,i} \tag{6}$$

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

$$T^l_{loc,i}(\alpha^l_{off}) = \alpha^l_{loc} T^l_{comp_i} = (1 - \alpha^l_{off}) T^l_{comp_i} \quad (7)$$

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

$$D^l_i(\alpha^l_{off}) = \max\{T^l_{off,i}(\alpha^l_{off}), T^l_{loc,i}(\alpha^l_{off})\} \quad (8)$$

Let us define the location of the ith FN at time instant τ as $Loc^i_\tau(x^i_\tau, y^i_\tau)$. Besides, location of the jth F-AP is defined as $Loc^j(x^j, y^j)$ that is considered to be fixed. 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. We assume that FNs are aware of the location of fixed F-APs. Moreover, having the knowledge of moving direction and velocity, the ith FN can estimate the remaining distance before moving out of the coverage of the jth F-AP at time instant τ as:

$$K^{i,j}_\tau = \sqrt{R^2_j - \acute{y}^2} + |x^i_\tau - x^j| \quad (9)$$

where $\acute{y}$, as shown in Fig. 1, is the minimum distance between the ith FN and the jth F-AP and R_j is the radius of the jth F-AP's coverage area. On the other side, it is possible to calculate the distance traversed during the offloading time, that is:

$$\bar{K}^{i,j}_\tau(\alpha^l_{off}) = v_i \cdot T^l_{off,i}(\alpha^l_{off}) \quad (10)$$

Now, let us define the data loss for the lth task of the ith FN as:

$$DL^i_l(\alpha^l_{off}) = \begin{cases} 1 & \text{if } K^{i,j}_\tau < \bar{K}^{i,j}_\tau(\alpha^l_{off}) \\ 0 & \text{if } K^{i,j}_\tau \geq \bar{K}^{i,j}_\tau(\alpha^l_{off}) \end{cases} \quad (11)$$

which means that if the distance that the ith FN has traversed for offloading a portion of task is higher than the distance it was able to traverse to remain in the coverage area of the F-AP, the task is considered to be lost. Having the goal of minimizing the data loss and delay in the network, we define our minimization problem as:

$$\begin{aligned} &\min_{\boldsymbol{\alpha_{\text{off}}}} \left\{ \frac{\sum_{i=1}^N \sum_l DL^i_l(\alpha^l_{off})}{\sum_{i=1}^N \sum_l DG^i_l} \right\} \\ &\min_{\boldsymbol{\alpha_{\text{off}}}} \left\{ \frac{\sum_{i=1}^N \sum_l D^l_i(\alpha^l_{off})}{\sum_{i=1}^N \sum_l DG^i_l} \right\} \end{aligned} \quad (12)$$

subject to

$$T^l_{comp_i} > T^l_{comp_j} > T^l_{tx,i} > T^l_{rx,i} > 0 \quad (13)$$

$$T^l_{w_j} \geq 0 \quad (14)$$

$$d_{ij} \leq R_j \quad (15)$$

$$\alpha^l_{loc} + \alpha^l_{off} = 1 \quad (16)$$

where $\boldsymbol{\alpha_{\text{off}}}$ is the set of the offloaded portion of all the tasks, and

$$DG^i_l = \begin{cases} 1 & \text{if the } i\text{th FN generates a task to be processed} \\ 0 & \text{otherwise} \end{cases} \quad (17)$$

allows to consider the number of tasks generated by the FNs in total. Hence, there are two objectives in the formulation, i.e., minimizing the average data loss, which is sum of the tasks which have been lost during network time for all FNs over sum of all the generated tasks, and average task delay respectively shown in (12). Constraint (13) introduces the hypothesis that the computing time of FNs is higher than F-APs, which itself is higher than transmission and receiving time of FNs. Waiting time for task l could be zero or more depending on the queue load of the jth F-AP and this is shown in (14). Constraint (15) ensures the distance between an FN and an F-AP should be less than the maximum F-AP coverage distance in order to have an interaction. Moreover, the portion of the task which is offloaded and the rest which is performed locally should equal to one and this is shown in constraint (16).

Hence, it is hard to find a closed solution for the partial offloading problem from mobile FNs to F-APs in the defined optimization problem. Thus, in the following section we propose a suboptimal solution.

III. The Partial Offloading Techniques

The approach we are going to propose, sees the offloading problem as separation of the data and control plane and it is mainly composed of two parts. The data plane section is considered to be through the FN to F-AP communication. In this stage we estimate the portion of a task to be offloaded to reduce the data loss and delay. On the other hand, the control plane section is applied through the D2D communication where the set of waiting time of the F-APs, depending on the number of tasks in their queue, is shared among FNs to assist the estimation of portion of offloaded task. Fig. 1 clearly depicts the movement of an FN to right while taking advantage of D2D communication, control plane, for having an estimation for partial offloading with the F-AP, the data plane.

A. Data Plane

In order to respect the second condition defined in (11), corresponding to avoid data loss, we introduce here a data plane optimization model able to estimate the amount of data to be offloaded. In this work, we assume that a task is lost when the jth F-AP is not able to send the result of the offloaded task to the FN before it moves out of its coverage. To make sure that a task is not lost we need to find the portion of the task that can be offloaded making sure the FN receives the result back before moving out of the coverage of the F-AP to which it has offloaded the task. However, the transmission time of the result and the waiting time of the task in the queue of the F-AP should also be considered for estimation of this portion. To avoid data loss, the distance

traversed by an FN should be less than the distance to remain in the coverage when the F-AP is used for offloading as shown in the second condition in (11). To find the portion of the lth task which can be offloaded considering the offloading time and the velocity, exploiting (9), (10) and (6), we can rewrite the second condition in (11), corresponding to the absence of data loss, as:

$$v_i \left(\alpha_{off}^l \frac{L_{s_l}}{r_{ij}} + T_{w_j}^l + \alpha_{off}^l \frac{O_l}{\eta_{comp_j}} + \alpha_{off}^l \frac{L_{r_l}}{r_{ij}} \right) \leq K_\tau^{i,j} \tag{18}$$

which shows that the condition for avoiding data loss depends, among others, on the α_{off}^l parameter. This brings us to:

$$\alpha_{off}^l \leq \frac{K_\tau^{i,j} \cdot \eta_{comp_j} \cdot r_{ij} - T_{w_j}^l \cdot v_i \cdot \eta_{comp_j} \cdot r_{ij}}{O_l \cdot v_i \cdot r_{ij} + L_{r_l} \cdot v_i \cdot \eta_{comp_j} + L_{s_l} \cdot v_i \cdot \eta_{comp_j}} \tag{19}$$

The above condition allows to minimize the data loss condition by setting an upper limit on the amount of data to be offloaded.

To further optimize the delay, in case the condition in (19) leads to a complete offloading, we propose a refinement optimization in which an FN avoids offloading the whole task thus reducing its idle time. To this aim, the delay minimization can be obtained by putting the amount of time needed for offloading equal to the amount of time needed for the local computation, i.e.,

$$T_{off,i}^l(\alpha_{off}^l) = T_{loc,i}^l(\alpha_{off}^l) \tag{20}$$

By resorting to (6) and (7), it is possible to rewrite the above condition as:

$$\alpha_{off}^l \frac{L_{s_l}}{r_{ij}} + T_{w_j}^l + \alpha_{off}^l \frac{O_l}{\eta_{comp_j}} + \alpha_{off}^l \frac{L_{r_l}}{r_{ij}} = \alpha_{loc}^l \frac{O_l}{\eta_{comp_i}} \tag{21}$$

which leads to:

$$\alpha_{off}^l = \frac{O_l r_{ij} \eta_{comp_j} - T_{w_j}^l \eta_{comp_i} r_{ij} \eta_{comp_j}}{\eta_{comp_j} \left(L_{s_l} \eta_{comp_i} + O_l r_{ij} + L_{r_l} \eta_{comp_i} \right) + O_l r_{ij} \eta_{comp_i}} \tag{22}$$

B. Control Plane

In order to perform the estimation of α_{off}^l in (19) and (22), it is needed to have knowledge of some quantities related to both F-APs and FNs. However, if some of them could be considered as known or easy to be found, the waiting time $T_{w_j}^l$ is unknown and, moreover, it is time variant, depending on the F-AP queue load. To this aim a control plane approach is considered herein by exploiting the D2D links among FNs for exchanging useful data for estimating the delay suffered by each task in each F-AP processing queue. To this aim we suppose that when an FN receives the result of its offloaded task, it keeps the record of the amount of time the task has waited in the queue of the that specific F-AP and also the time instant this information has been updated, corresponding to τ. The set of waiting time updated at the time instant τ at different F-APs of the ith FN is shown as:

$$\mathcal{B}_i = \left\{ T_{w_j}(\tau) \right\}, j = 1, \ldots, M \tag{23}$$

This set is including the latest received information about the waiting time of the F-APs by each FN. In the proposed idea as two FNs are approaching, they update their set of waiting time by comparing the time in which the corresponding F-AP has been updated in order to store only the most recent value. If the sender's updating time is more recent, the information about the waiting time of that F-AP will be updated in the recipient FN's waiting time set. This corresponds to say that the information in the buffer of each FN, defined in (23) can be rewritten as:

$$\mathcal{B}_i = \left\{ T_{w_j}(\bar{\tau}) | \bar{\tau} = \max_k(\tau_k), d_{ik} \leq d_{D2D} \right\}, j = 1, \ldots, M \tag{24}$$

where $\bar{\tau}$ is the maximum updating time instant, i.e., the most recent time instant, among all the k approaching FNs that are in the D2D coverage area of the ith FN, that is equal to d_{D2D}. Based on the exchanged information about the F-APs status among different FNs it is possible to resort to two control plane-based optimization algorithms that are called D2D communication approach and Time Aware- D2D (TA-D2D) approach, where the first is based on estimating (19), while the second is based on estimating (22), both based on the exchanged information in the control layer. The control layer information about the waiting time in each F-AP is then used as an input for the data plane for estimating the optimal α_{off}^l value for minimizing the data loss and the delay.

IV. Numerical Results

In this section, the numerical results obtained through computer simulations are presented. We consider to have a two layer scenario where in the first layer there are D2D communications among FNs to update their waiting list and on the second layer the F-APs perform the computation of the tasks sent from the FNs.

The computer simulations are performed in Matlab; the information regarding the parameters are shown in Tab. I. The computer simulations are carried out in terms of average task delay and data loss, defined as:

- Average Task Delay: The average time spent for offloading or for doing the local computation (See (8)).
- Data Loss: Number of unsuccessful receptions by FNs, due to moving out of the coverage of an F-AP, over total number of generated tasks (See (11)).

In this section we will compare the performance of the D2D and TA-D2D approaches with two benchmarks. First, an Intelligent Aided approach in which FNs are aware of the exact value of the waiting time. Second, a No Knowledge approach in which FNs do not have any information about the waiting time of F-APs.

We have compared the performance of these four approaches in terms of delay and data loss first for different number

TABLE I
SIMULATION PARAMETERS

Parameter	Value
Dimension	300m x 50m
Task size (L_s)	5 MB
Task result size (L_r)	1 MB
Path loss ($L(d_{ij})$)	140.7+36.7*log10(d) dB [11]
Bandwidth (B_{ij})	10 MHz
Thermal noise (N_T)	-174 dBm/Hz [11]
FN to FN coverage range	15 m
F-AP coverage radius (R_j)	70 m
Task Operation (O_l)	50G
FN Flops (η_{comp_i})	15G FLOPS
F-AP Flops (η_{comp_j})	150G FLOPS
Velocity (v_i)	[1-4] m/s

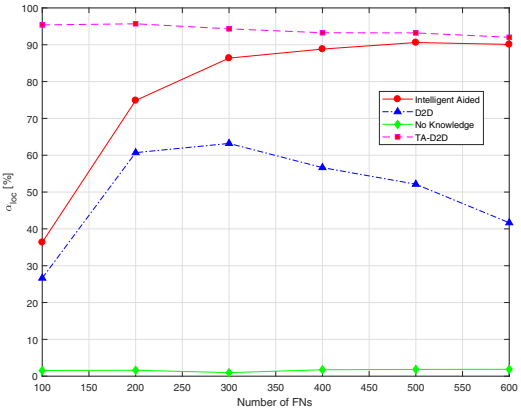

Fig. 2. Local Computation Percentage for a variable number of FNs by considering a task generation probability equal to 0.1

of FNs, by considering a generation task probability equal to 0.1, and then different task generation probabilities, while the number of FNs has been set to 300. Moreover, we have shown how the value of α_{loc} changes over time.

In Fig. 2, behavior of α^l_{loc} is shown when the number of FNs increases. As seen, in No Knowledge approach FNs are always offloading the whole task due to the lack of knowledge about waiting time. However, in Intelligent Aided approach for small number of FNs higher portion of tasks are offloaded while for higher number of FNs by having the knowledge of the increase of traffic in F-APs queue, most are performed locally. On the other hand, D2D approach offloads more comparing with TA-D2D approach because of the FNs are not constrained in minimizing the delay. Moreover, D2D offloads a higher portion comparing with Intelligent Aided approach due to the fact that Intelligent Aided scheme has a perfect knowledge about the queue of the F-APs, and this up to date information, exchanged among FNs, takes some time to spread among all the FNs.

Fig. 3 depicts the delay for different number of FNs. If FNs offload more, the waiting time increases and as a result the delay is expected to increase. As seen the TA-D2D approach performs closer to the Intelligent Aided approach. When number of FNs is low, the traffic is not high and as a

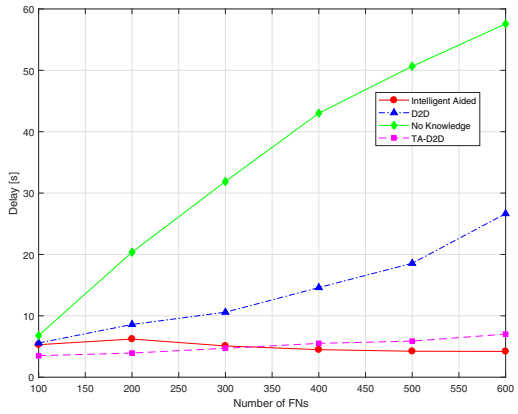

Fig. 3. Average Task Delay for a variable number of FNs by considering a task generation probability equal to 0.1

result higher portion of tasks are performed locally in TA-D2D scenario compared with the Intelligent Aided scenario leading to a lower delay. However, FNs in TA-D2D offload more over time and as a result delay raises a little. As depicted in the figure, D2D approach has a higher delay comparing with TA-D2D scenario because of the higher portion of tasks offloaded, and by an increase in number of FNs and time, traffic at the queue of the APs increases and this leads to more time for processing the tasks. As expected, No Knowledge approach offloads the highest amount of task and this leads to a higher delay over time.

Data loss is depicted in Fig. 4. As seen, the Intelligent Aided scenario, in which FNs are aware of the waiting time, is able to estimate the offloading portion precisely which never leads to a data loss. On the other hand, No Knowledge scenario has the highest amount of data loss because it offloads tasks regardless of the long queue of the F-APs. Moreover, when the waiting time is updated among FNs with D2D communication, FNs are able to better estimate the offloading portion and as a result data loss is low. As expected, D2D approach has a higher data loss compared with TA-D2D because the average value of α^l_{loc} for D2D is lower and this means in TA-D2D approach, FNs will perform more portion of the task locally, comparing with D2D approach, even though they can offload the whole task leading to a lower data loss.

Fig. 5 shows the delay in terms of different task generation rate. As seen, in low traffic situation, because all scenarios have low α^l_{loc}, more offloading is performed and the results are sent back quickly due to the low waiting time, however TA-D2D performs more locally and it leads to a higher delay for smaller task generation probability. On the other hand, by generating more tasks and having a higher waiting time, TA-D2D performs better due to the amount of local computation.

Fig. 6 depicts the impact of higher traffic on data loss. By an increase in task generation rate, Intelligent Aided approach has still zero percent of data loss due to the awareness about the waiting time and preciseness in estimating α^l_{loc}. However,

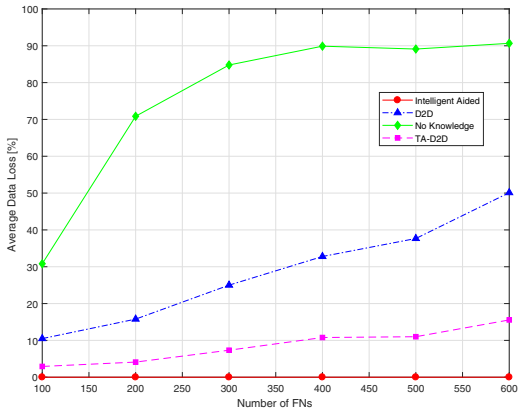

Fig. 4. Data Loss for a variable number of FNs by considering a task generation probability equal to 0.1

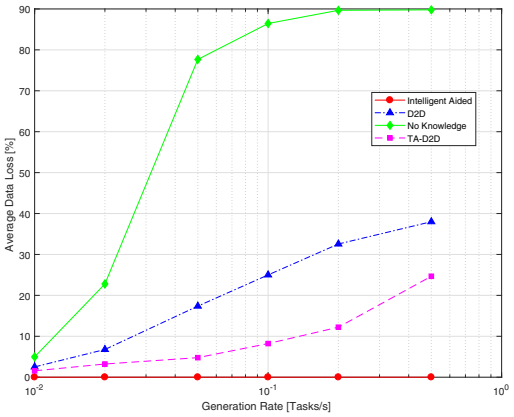

Fig. 6. Data Loss for a variable task generation probability by considering the number of FNs equal to 300

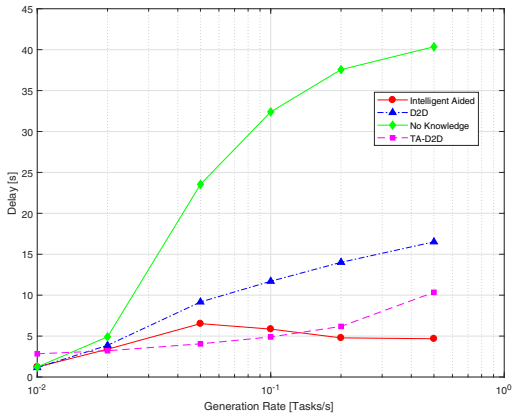

Fig. 5. Average Task Delay for a variable task generation probability by considering the number of FNs equal to 300

TA-D2D has a close performance to it at low generation rate and it increases over time. The D2D approach in which α_{loc}^{l} is estimated by the amount of waiting time, performance is a little worse than TA-D2D in which lower portion of tasks are offloaded.

The simulation results underscore the impact of separation of data and control plane and the exploitation of the D2D communication on the performance in terms of data loss and delay. It is proved that the knowledge about waiting time greatly impacts the delay and data loss. 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 data loss.

V. Conclusions

In this work, we studied the partial offloading in fog computing architecture. Best effort was made to estimate the right amount of task to offload in order to avoid high amount of delay and data loss. Deploying a D2D communication we tried to pass the waiting time information among FNs in order to better estimate the task offloading portion. We further proposed a method in which this portion is estimated in a way to have the lowest amount of idle time when a task is requested. Simulation results demonstrate that our proposed method has lower delay and data loss by benefiting from the D2D communication achieving results comparable with the ideal situation.

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] Z. Yin, F. R. Yu, S. Bu, and Z. Han, "Joint cloud and wireless networks operations in mobile cloud computing environments with telecom operator cloud," *IEEE Trans. Wireless Commun.*, vol. 14, no. 7, pp. 4020–4033, Jul. 2015.

[3] M. Zhanikeev, "A cloud visitation platform to facilitate cloud federation and fog computing," *Computer*, vol. 48, no. 5, pp. 80–83, May 2015.

[4] S. Yan, M. Peng, and W. Wang, "User access mode selection in fog computing based radio access networks," in *IEEE ICC 2016*, May 2016.

[5] 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.

[6] H. Kim and N. Feamster, "Improving network management with software defined networking," *IEEE Commun. Mag.*, vol. 51, no. 2, pp. 114–119, Feb. 2013.

[7] C. Wang, F. R. Yu, C. Liang, Q. Chen, and L. Tang, "Joint computation offloading and interference management in wireless cellular networks with mobile edge computing," *IEEE Trans. Veh. Technol.*, vol. 66, no. 8, pp. 7432–7445, Aug. 2017.

[8] J. Oueis, E. C. Strinati, and S. Barbarossa, "The fog balancing: Load distribution for small cell cloud computing," in *2015 IEEE 81st Vehicular Technology Conference (VTC Spring)*, May 2015.

[9] 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.

[10] 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.

[11] *Further advancements for E-UTRA physical layer aspects*, 3GPP TR 36.814, Rev. 9.0.0, Mar. 2010.