Towards Joint Cell Selection and Task Offloading in Cellular IoT Systems with Edge Computing

Edgar Adrian Esquivel-Mendiola, Sergio Pérez-Picazo, Hiram Galeana-Zapién

Centro de Investigación y de Estudios Avanzados del I.P.N. (Cinvestav)
Cinvestav Tamaulipas
Ciudad Victoria, México
email: {edgar.esquivel,sergio.perez.picazo,hiram.galeana}@cinvestav.mx

Abstract—**Mobile Edge Computing (MEC) in cellular networks aims to bring computational capabilities close to end-users to reduce the latency of applications on the Internet of Things (IoT). This is particularly crucial to computation-intensive IoT broadband applications (e.g., video analytics, augmented reality, etc.) demanding a data processing task to be performed within a given time threshold. In this regard, the task offloading problem has been investigated in the literature in order to achieve an appropriate trade-off between energy and latency. However, there is a need for the joint design of task offloading mechanisms and cell selection algorithm as a mean to select the most appropriate to each device in order to meet delay requirements and fulfill resource constraints at the MEC server site. In this paper, we present the foreseen framework to tackle such a challenging problem.**

***Keywords*–Delay-sensitive applications, internet of things, mobile edge computing, offloading.**

I. Introduction

The Internet of Things (IoT) is the most recent evolution of the Internet technologies and services. In this novel paradigm, everyday objects will be equipped with communication, computing and sensing capabilities to collect data from their environment. The analysis of the collected data is expected to enable knowledge-based decisions and to produce value-added services in domains like healthcare, manufacturing, logistics, etc. Broadly speaking, IoT applications in such domains could be categorized according to the communication requirements (i.e., data throughput, data volume, latency, etc.), or in terms of the IoT device capabilities (low-cost fixed/mobile sensor, smartphone, wearable, etc.). In this context, the fifth generation (5G) mobile communication system will play a key role in the IoT ecosystem by providing wireless connectivity to IoT *mobile* devices dispersed over large areas and demanding stringent Quality of Service (QoS) levels. For instance, the 5G network is expected to support a major IoT market segment related to next-generation broadband and critical services having low delay requirements and heavy computational-resource needs. Towards providing flexible support of these IoT applications' categories, enabling cellular technologies, such as 5G new radio (NR), Long-Term Evolution (LTE) for Machine Type Communication (MTC), a.k.a. LTE-M, and Narrowband IoT (NB-IoT) have been proposed. Additionally, in order to reduce service latencies, the Mobile Edge Computing (MEC) aims to bring storage and computational resources close to end devices in the Radio Access Network (RAN) [1]. This implies the deployment of MEC servers co-located in the cell site with the Base Station (BS). This allows reducing the round-trip latency for applications that offload data from terminals to the network for data processing purposes and wait for the outcomes. Such a data processing is preferred to be performed at the network edge in order to avoid sending big amounts of traffic from the network edge through the backhaul towards more distant central computing resources (i.e., mobile cloud computing paradigm). In this sense, one of the main challenges is that MEC servers are generally known to be constrained by the amount of computation resources, implying that they might be easily overloaded due to intensive computation requests from IoT broadband applications. Another important resource constraint in the RAN is the available radio resources.

Under such a distributed computing environment, the task offloading problem has been subject of study in the literature aiming to determine if a given task is computed locally (at the mobile device) or at the network edge. However, as intensive computation tasks are energy-consuming, on-device computation approaches severely affect the lifetime of battery-limited devices. In order to address these challenges, different computation task models for full or partial data offloading have been explored in the literature [2], where some degrees of freedom could be allowed by considering soft deadline requirements (i.e., portions of data to be computed after a time threshold). The task offloading problem aims to achieve a trade-off between energy efficiency at mobile devices and end-to-end delay of applications. This latter aspect involves the appropriate modeling of delay components for data processing (at the mobile device or network edge node) and data transmission over wireless channel. At this regard, the task offloading problem it is commonly analyzed under the assumption that each mobile device is already assigned to a MEC server according to a cell selection criterion aimed to optimize radio link efficiency (e.g., minimum path loss, max-SINR, etc.). Nevertheless, such cell selection approaches neglect other aspects that could greatly influence the obtained Quality of Service (QoS), such as selecting an overloaded MEC server or a BS with exhausted radio resources.

In this paper, we describe our work in progress where the aim is to develop an integral framework to jointly optimize the offloading decisions and the cell selection. Specifically, we formulate an optimization problem aimed to determine the most appropriate serving cell to each device attending

Copyright (c) IARIA, 2020. ISBN: 978-1-61208-819-8

to: a) radio channel bandwidth in the cell, and b) computing capacity of the MEC server, and c) energy constraints of mobile devices. We aim to determine the cell selection solution that provides the minimum the system performance delay in MEC-based IoT scenarios. Such a framework involves a number of challenges. First, the algorithms to be developed should work in distributed computing environments, implying that each edge node should perform resource allocation decisions independently based on local information. To this end, distributed iterative algorithms based on a pricing-based scheme (where the assignment process behaves as a bidding process that iteratively allocate edge resources to end users) are considered as a potential algorithmic solution. Second, realizing low-latency and energy-efficient in MEC scenarios demand joint radio and computational management schemes. It is well known that wireless channel conditions (path loss, interference, etc.) affects the amount of energy consumption required for data offloading. That is, poor channel conditions are likely to lead to low achievable data rate at the air interface, implying and increased energy consumption at the mobile device as well as transmission latency.

Attending to the aforementioned arguments, this paper presents a work in progress that aims at investigating integrated task offloading and cell selection approaches able to, for instance, select a given cell with favorable channel conditions to perform data offloading. The rest of the paper is organized as follows. Section II presents the related work. Section III presents the foreseen technical approach. Section IV details the envisioned contributions and also concludes the paper.

II. State of the Art

The task offloading problem has been studied in the literature with the aim of determine the appropriated site to perform the task processing according to the conditions observed in the system (i.e., energy constraints, latency requirements, channel conditions, computing capabilities). Some of the proposed solutions are based on greedy solutions, heuristics and well-known approaches from the literature. Several research works have tackled the task offloading problem in MEC-like system deployments. The most relevant approaches proposed so far are summarized in Table I, focusing on key aspects such as energy and latency requirements, type of task offloading (full or partial), algorithm type. We also highlight that the vast majority of existing approaches does not include the cell selection problem (i.e., tasks offloading is tackled assumed for a given cell selection solution). In what follows, we provide a brief analysis of the related work.

Huang *et al.* [3] proposed an algorithm to dynamically perform the task offloading process taking into account the wireless status. This approach is based on the Lyapunov optimization algorithm, which aims to improve the energy consumption [4], while satisfies the execution time required for mobile application. However, the authors do not consider the support of IoT devices in the cellular system. Similarly, Zhang *et al.* [5] proposed a mechanism to minimize the energy consumption during task offloading with MEC, although this work provides a support of multiple devices under a 5G heterogeneous network, it does not consider the main IoT scenarios for task offloading and assume that the MEC servers are not constrained. The results demonstrated that energy consumption can differ with the number of mobile devices, so there is not a lineal relation between them. Yu *et al.* [6] proposed an algorithm for task offloading dynamically. The main objective of this approach is to minimize the cost generated by the network through the development of supervised deep learning model. In contrast to other proposals, this implementation is modeled as a classification problem to search an alternative for task offloading process considering the network conditions. Similarly, Chen *et al.* [7] proposed an adaptative algorithm for task offloading at the same time it considers the capabilities MEC paradigm offer. The algorithm dynamically decide when to perform the task offloading process based on the network status. According to the authors, this work was evaluated with two real world applications: license plates recognition and voice recognition. As a result, the algorithm reduces the energy consumption without considering the capacity restrictions of the MEC server.

As shown in Table I, a drawback of existing approaches is that are proposed for centralized solutions where based on the complete information of the system determines the offloading of the tasks. Furthermore, most of the studied works do not consider the joint cell selection and delay minimization problem, instead they assume a previous assignment, which is mainly based on greedy solutions. In this sense, our envisioned solution described in the next section aims to design a joint cell selection and delay minimization algorithm, which operating in a distributed way at each BS determines the appropriated BS to offload the task according to delay and energy constraints.

III. Foreseen Research Work

This research aims to investigate novel joint cell selection and task offloading solutions to provide enhanced support of delay-sensitive services demanding computation-intensive capabilities. This section presents the foreseen technical approach towards this challenging problem. Namely, we present the system model and the corresponding optimization problem that we are targeting.

A. System Model

The system model is illustrated in Figure 1. We consider a cellular deployment with N cell sites and M IoT devices in the service area. In each cell site, a MEC server is assumed to be co-located with the BS equipment. Accordingly, the following resource constraints per site are considered: a) radio channel bandwidth used in the BS, and b) computing capacity of the MEC server. In addition, in line with LTE/4G and NR/5G radio interfaces, we assume OFDMA as an access method in the air interface so that the total system bandwidth W is divided into K resource blocks according to a frequency reuse pattern. Therefore, the amount of radio resources is defined in terms of the K_j resource blocks assigned to each BS $j \in N$. Finally, it is worth clarifying that backhaul resources are not considered in the present modelling as all tasks are assumed to be processed either at the mobile device or the network edge. In other words, in collaborative approaches between edge and cloud computing resources, the described system model could be easily extended by assuming that each cell site is connected to the network core and central clouds by means of a backhaul network with finite link bandwidth at each BS.

Additionally, we assume that each IoT device i have specific delay and computation requirements to process task A_i, which cannot be partitioned and should be processed as

TABLE I. SUMMARY OF RELATED WORK.

Author	Year	Cell Selection	Algorithm	Task Offloading Type	Saved Energy	Reduce Latency
Xiang *et al.* [8]	2019	No	Fragment algorithm for data processing	Full	No	Yes
Ning *et al.* [9]	2019	No	Hybrid offloading algorithm	Full	No	No
Sun *et al.* [10]	2019	No	Hybrid offloading algorithm	Partial	No	Yes
Chen *et al.* [7]	2019	No	Adaptative offloading algorithm	Full	Yes	Yes
Sun *et al.* [11]	2018	No	Greedy algorithm	Full	Yes	Yes
Wu *et al.* [12]	2018	No	Offloading algorithm based on environment identification	Partial	Yes	No
Yu *et al.* [6]	2017	No	Heuristic	Partial	Yes	No
Deng *et al.* [13]	2016	No	Adaptative offloading algorithm for multiple users	Full	No	No
Zhang *et al.* [5]	2016	No	Efficient computing algorithm	Partial	Yes	No
Huang *et al.* [3]	2012	No	Dynamic data offloading algorithm in IoT devices	Partial	Yes	Yes

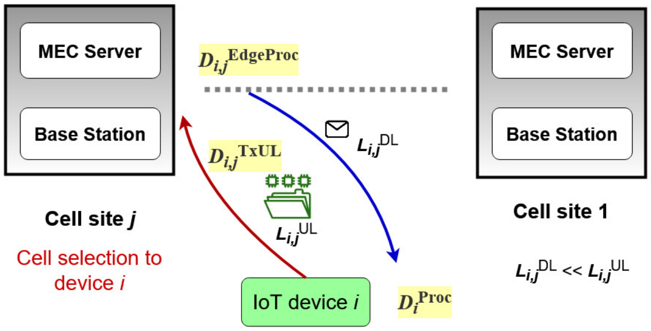

Figure 1. System Model.

a whole either at the mobile device or edge node collocated with the cell j serving the device (e.g., video stream analysis [2]). Each computation task is modeled using a three-field notation $A_i(L_{i,j}^{\text{UL}}, \gamma_{i,j}, D_i^{\text{req}})$, where $L_{i,j}^{\text{UL}}$ is the input data file (in bits) to be transferred through the uplink wireless channel to the selected edge node j, $\gamma_{i,j}$ denotes the workload (CPU cycles/bit) for processing one-bit data, and D_i^{req} is the hard deadline imposed by the application to further process the file and receive the corresponding response. In this sense, as illustrated in Figure 1, the total delay experienced by a file from a given device can be expressed as:

$$D_{ij} = \varphi_i D_i^{\text{proc}} + [(1-\varphi_i)(D_{i,j}^{\text{TxUL}} + D_{i,j}^{\text{EdgeProc}} + D_{i,j}^{\text{TxDL}})] \quad (1)$$

where $\varphi_i \in \{0,1\}$ is a variable that is equal to 1 if the task is processed locally at the device i, or 0 if it is offloaded to the mobile edge for processing purposes. Moreover, the terms $D_{i,j}^{\text{TxUL}}$ and $D_{i,j}^{\text{TxDL}}$ denote the transmission delays in the uplink and downlink, respectively, which can be derived based on the corresponding transmission rates and the length of the data to be offloaded. Notice that $L_{ij}^{\text{DL}} << L_{ij}^{\text{UL}}$ could be assumed due that the response from the edge server to the mobile device is smaller in size than the data offloaded to the MEC server. Furthermore, D_i^{proc} and $D_{i,j}^{\text{EdgeProc}}$ are the delay observed if the task is processed at the mobile device or MEC server, respectively. Following the formulation presented in [14], we define the processing delay D_i^{proc} and $D_{i,j}^{\text{EdgeProc}}$ as follows:

$$D_i^{\text{proc}} = \frac{L_{i,j}^{UL}\gamma_{i,j}}{C_i^{\text{Device}}} \quad (2)$$

where C_i^{Device} denotes the computing capacity (CPU cycles/sec) of the mobile device. Similarly, the processing delay at the MEC server can be computed as follows:

$$D_{ij}^{\text{EdgeProc}} = \frac{L_{i,j}^{UL}\gamma_{i,j}}{C_{i,j}^{\text{Edge}}} \quad (3)$$

where $C_{i,j}^{\text{Edge}}$ denotes the amount of computing resources (CPU cycles/sec.) assigned by the edge node j to process the task of device i. Furthermore, we can estimate the energy consumption for the task processing at the mobile device as follows [15]:

$$E_i = (L_i^{\text{UL}}\gamma_i)f_i \quad (4)$$

where f_i denotes the required energy to process one bit at the mobile device. The residual energy could provide an appropriate hint to decide if a task is offloaded or computed locally. In addition, Equation (4) could be extended to include the energy consumption required to transmit the task to the BS node.

B. Problem Formulation

Let $b_{i,j} = 1$ the variable denoting whether or not device i is associated with BS computing node j to offload tasks. The joint cell selection and task offloading is that of determining the most appropriate assignment $B = \{b_{i,j}\}$ in order to minimize total delay of all devices while satisfying the computing resources at each node j as well as the energy constraint of IoT devices. The problem formulation can be written as follows:

$$\min \quad \sum_{j=1}^{N}\sum_{i=1}^{M} D_{ij} b_{i,j} \tag{5}$$

$$\text{s.t.} \quad \sum_{i=1}^{M} C_{i,j}^{\text{Edge}} b_{i,j} \leq 1,\ j = 1,\ldots,N \tag{6}$$

$$D_{ij} \leq D_i^{\text{req}}, i = 1,\ldots,M \tag{7}$$

$$\sum_{j=1}^{N} b_{i,j} \leq 1,\ i = 1,\ldots,M \tag{8}$$

$$b_{i,j} \in \{0,1\} \tag{9}$$

where (5) is the objective function defined in terms of delay experienced by the application. In the case of constraint (6), for each connected user node j will allocate an amount of computing resources denoted as $C_{i,j}^{\text{Edge}}$, so the allocation of a total of devices to an edge node should not exceed the maximum available computing capacity. In the constraint (7), the delay $D_{i,j}$ refers to the sum of various delay components considering the processing time (in MEC node j or locally on device i) and transmission time on the wireless interface. In either case, the total delay must satisfy the latency requirement of the application.

The problem (5)-(9) is a combinatorial optimization problem due to the binary variable $b_{i,j}$, so that solving the problem with exact algorithms may be difficult even for a small number of N and M. We aim to refine such a problem in order to make it more tractable, e.g., reducing the number of constraints and to model the association process as a message passing based on pricing values of BSs. In that way, dual decomposition theory could be applied to design a distributed approach to solve the above problem. More specifically, the above formulated problem could be refined depending on the implementation and validation of the envisioned algorithms. In order to narrow down the joint cell selection and task offloading problem, we consider the following Research Questions (RQ):

- *RQ1:* ¿Is it possible to design a cell selection criteria to steer device associations based on radio/computation conditions at the MEC servers and delay requirements of applications?
- *RQ2:* ¿How to design an efficient distributed cell selection algorithm that operating with network partial state information could find the optimal assignment of communication and computation resources in order to minimize the system delay?
- *RQ3:* ¿How to properly model a decision making mechanism to determine if a task should be processed locally, at the MEC server or a partial offloading?

We aim to conduct Montecarlo simulations to evaluate the performance of our joint cell selection and task offloading approach. Moreover, two task offloading approaches from the literature will be used for benchmarking purposes. The evaluation will be carried out in terms of the total average delay achieved by the algorithms when offloading the tasks to the edge server or computed locally at the mobile device. On the one hand, we are interested in analyzing the suitability of assigning mobile device i to edge server j in order to meet the delay requirement imposed by task A_i. In this sense, we aim to demonstrate that a cell selection procedure that accounts for computing capacity of edge servers and delay requirements from mobile devices, lead to better performance in terms of overall system delay. Hence, a set of simulations varying the available system bandwidth and the delay requirements from applications are considered in the experimental analysis. On the other hand, we want to evaluate the computing delay conditions associated to the task processing under the variation of the availability of computing capabilities at the devices and the MEC servers. At this regard, we aim to compare the proposed solution with related works from the literature that are based on both computing and joint computing-communication constraints.

IV. Conclusion and Future Works

In order to complement the design of task offloading schemes in MEC scenarios, we studied the integration of wireless channel conditions into a joint formulation. In particular, we have described the design of a joint task offloading and cell selection schemes able to determine the serving cell (MEC server) to each mobile device taking into account device and network resource constraints and QoS requirements of applications. We have discussed that the design of task offloading solutions it is commonly tackled from a mobile computing perspective (i.e., reduce energy consumption, etc.), whereas few attention has been paid on how the cell selection procedure would severely impact the observed latency due to, for instance, the assignment of a device to an overloaded cell. Taking this into account, we presented a work in progress towards a novel task offloading and cell selection framework to drive the assignment decisions based on applications' requirements and the availability of radio and computing resources at the MEC servers. Finally, we described the experimental simulation to be used for validating our proposed approach.

References

[1] N. Hassan, S. Gillani, E. Ahmed, I. Yaqoob, and M. Imran, "The role of edge computing in internet of things," IEEE Communications Magazine, August 2018, pp. 1–6.

[2] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," IEEE Communications Surveys Tutorials, vol. 19, no. 4, 2017, pp. 2322–2358.

[3] D. Huang, P. Wang, and D. Niyato, "A dynamic offloading algorithm for mobile computing," IEEE Transactions on Wireless Communications, vol. 11, no. 6, 2012, pp. 1991–1995.

[4] S. Ou, K. Yang, and J. Zhang, "An effective offloading middleware for pervasive services on mobile devices," Pervasive and Mobile Computing, vol. 3, no. 4, 2007, pp. 362–385.

[5] K. Zhang et al., "Energy-efficient offloading for mobile edge computing in 5g heterogeneous networks," IEEE access, vol. 4, 2016, pp. 5896–5907.

[6] S. Yu, X. Wang, and R. Langar, "Computation offloading for mobile edge computing: A deep learning approach," in 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC). IEEE, 2017, pp. 1–6.

[7] X. Chen et al., "An adaptive offloading framework for android applications in mobile edge computing," Science China Information Sciences, vol. 62, no. 8, 2019, p. 82102.

[8] B. Xiang, J. Elias, F. Martignon, and E. Di Nitto, "Joint network slicing and mobile edge computing in 5g networks," in ICC 2019-2019 IEEE International Conference on Communications (ICC). IEEE, 2019, pp. 1–7.

[9] Z. Ning, X. Wang, J. J. Rodrigues, and F. Xia, "Joint computation offloading, power allocation, and channel assignment for 5g-enabled traffic management systems," IEEE Transactions on Industrial Informatics, vol. 15, no. 5, 2019, pp. 3058–3067.

[10] H. Sun, F. Zhou, and R. Q. Hu, "Joint offloading and computation energy efficiency maximization in a mobile edge computing system," IEEE Transactions on Vehicular Technology, vol. 68, no. 3, 2019, pp. 3052–3056.

[11] Y. Sun, Z. Hao, and Y. Zhang, "An efficient offloading scheme for mec system considering delay and energy consumption," Journal of Physics: Conference Series, vol. 960, 01 2018, p. 012002.

[12] H. Wu, Y. Sun, and K. Wolter, "Energy-efficient decision making for mobile cloud offloading," IEEE Transactions on Cloud Computing, vol. 8, no. 2, 2020, pp. 570–584.

[13] M. Deng, H. Tian, and X. Lyu, "Adaptive sequential offloading game for multi-cell mobile edge computing," in 2016 23rd international conference on telecommunications (ICT). IEEE, 2016, pp. 1–5.

[14] W. Almughalles, R. Chai, J. Lin, and A. Zubair, "Task execution latency minimization-based joint computation offloading and cell selection for mec-enabled hetnets," in 2019 28th Wireless and Optical Communications Conference (WOCC), 2019, pp. 1–5.

[15] Y. Pei, Z. Peng, W. Zhenling, and H. Wang, "Energy-efficient mobile edge computing: Three-tier computing under heterogeneous networks," Wireless Communications and Mobile Computing, vol. 2020, 04 2020, pp. 1–17.

ISBN: 978-1-61208-819-8