Handling Service Allocation in Combined Fog-Cloud Scenarios

V.B.C.Souza [1] [2], W.Ramírez [2], X.Masip-Bruin [2], E.Marín-Tordera [2], G.Ren [3], G.Tashakor [2]
[1] Informatics Department (DPI), Universidade Federal de Viçosa (UFV), Viçosa, Brazil
[2] Advanced Network Architectures Lab (CRAAX), Technical University of Catalunya (UPC), Barcelona, Spain
[3] IBM Almaden Research Center, USA
Email: vitorbs@dpi.ufv.br;{wramirez, xmasip, eva, tashakor }@ac.upc.edu; gren@us.ibm.com

Abstract— **The recent technological advances related to computing, storage, cloud, networking and the unstoppable deployment of end-user devices, are all coining the so-called Internet of Things (IoT). IoT embraces a wide set of heterogeneous services in highly impacting societal sectors, such as Healthcare, Smart Transportation or Media delivery, all of them posing a diverse set of requirements, including real time response, low latency, or high capacity. In order to properly address such diverse set of requirements, the combined use of Cloud and Fog computing turns up as an emerging trend. Indeed, Fog provides low delay for services demanding real time response, constrained to support low capacity queries, whereas Cloud provides high capacity at the cost of a higher latency. It is with no doubt that a new strategy is required to ease the combined operation of cloud and fog infrastructures in IoT scenarios, also referred to as Combined Fog-Cloud (CFC), in terms of service execution performance metrics. To that end, in this paper, we introduce and formulate the QoS-aware service allocation problem for CFC architectures as an integer optimization problem, whose solution minimizes the latency experienced by the services while guaranteeing the fulfillment of the capacity requirements.**

Keywords—Internet of Things; Combined Fog-Cloud; IoT Service Allocation

I. INTRODUCTION

The recently defined Internet of Things (IoT) paradigm is mainly founded by the concept of a massive heterogeneous set of devices demanding anywhere, anytime, anyhow connectivity [1] to support a large set of enriched services and applications, strongly impacting on the society. Indeed, well known scenarios, such as Smart Cities, Smart Transportation or Smart Home, can be listed as reference sources of IoT future applications. Simultaneously, Cloud Computing has been positioned as the key enabler for IoT applications [2]. This is motivated by the huge capacity and processing resources of Cloud commodities, properly matching the required IoT services demands, including for example highly demanding services, such as media delivery or Data Center (DC) backup solutions.

Unfortunately, Cloud Computing faces substantial yet unsolved challenges, some of them, mostly centered on large response time, security concerns and global mobility support. These issues are mainly motivated by the large distance separating the end-user device requesting the service and the Cloud. To overcome these issues, a new architecture referred to as Fog Computing [3] has been recently proposed. Fog Computing inherits the main concepts of Cloud Computing, but move them to the edge of the network. The main goal is to bring computing resources closer to end-user devices, hence enforcing locality, which imposes low response time, low network load and less security concerns. Despite Fog commodities do not have the massive computing capacities inherent to Cloud, Fog can give a new breed to services requiring real time processing (e.g., Healthcare or Augmented Reality). Indeed, the goal of Fog Computing is not to compete with Cloud. Several studies position a scenario where Fog commodities are placed at the edge of the network and the cloud infrastructure is closer to the conventional network backbone [4], both operating together [5]. In this paper, we refer to this integration as Combined Fog-Cloud (CFC).

It is with no doubt that more research efforts are required to distill the challenges related to the actions supporting the allocation of IoT services in CFC scenarios. To fill this gap, this paper introduces and formulates the Service Allocation (SA) problem in CFC scenarios. The main goal of the SA problem is to minimize the latency experienced by a service to reach out to the resources meeting the service requirements.

The rest of this paper is organized as follows. Section II introduces the related work in the Fog Computing arena. Then, Section III discusses in a comprehensive manner the topology model for the CFC architecture and also distills the SA problem formulation. Section IV provides the numerical results for the latency in the CFC architecture. Finally, Section V provides the final conclusions and suggests avenues for future work.

II. RELATED WORK

The Research on the Cloud and Fog fields is still immature. There are no standards nor a widely accepted position related to what defines the overall Fog-Cloud architecture. However, compared to Fog Computing, Cloud Computing is well positioned in network research. Indeed, key studies describing the needs and the structure of data, management and control plane schemes for Cloud can be found in [4, 6, 7]. Related studies focusing on control actions needed for service allocation can be found in [8, 9]. Authors target the allocation of resources for clouds in dynamic scenarios [10]. Other studies, either present a methodology for service allocation in Cloud scenarios [11], or consider service allocation in mobile Cloud scenarios [12].

On the other hand, the seed contribution on Fog computing is the one presented at [3], where authors provide insights related to the main requirements and characteristics of Fog computing platforms in IoT scenarios, such as latency and energy consumption. Nevertheless, authors do not provide any evaluation results of Fog platforms regarding the presented requirements. Other related studies, such as [13], discuss the potential benefits of reliability features in Fog computing, though there is no methodology proposed for reliability evaluation. Authors in [14] focus on the energy consumption optimization as a key requirement in Fog Computing and present extensive evaluation results related to energy consumption in Fog Computing.

Different than existing contributions, this paper deals with the allocation of services in Combined Fog-Cloud scenarios (CFC). To this end, we propose an ILP model for the optimization of latency in planning scenarios. Although the work done in [15] aims to model the service allocation problem in planning scenarios through convex optimization techniques, the work we introduce in this paper present a different approach to the problem, highlighting the following:

- The latency added by the allocation of services in Fog is not neglected. Despite of the Fog's proximity to the end-users, the intrinsic latency of mobile applications may have negative effects on the allocated services allocated [16, 17].
- The proposed Fog-Cloud topology follows a four-tier layer design.

III. System Model and Problem Formulation

In this section, we first introduce the topology model of the Combined Fog-Cloud (CFC) and second we discuss the mathematical formulation of the SA problem.

A. System Model

The envisioned CFC scenario embraces both Fog and Cloud servers, which are hierarchically distributed in a vertical manner in four different layers, see Fig. 1. In our model, the hierarchy of a layer is determined by: capacity, vicinity, and reachability to end-users. For instance, we consider that highly mobile nodes, such as a vehicle traveling along a road, have low reachability to end-users. This is because they are in constant movement and it is hard to predict the total time they will be on the scope of end-users.

- The first tier is composed by end-user devices connected by means of distinct access technologies, such as WiFi or 4G. These end-user devices may both request and offer resources to the CFC model.
- The second tier is the Fog first layer and is the one closer to end-users. It is commonly connected to the end-users by means of a single-hop Wireless access connection, and it is composed by Fog servers with low capacity, which aggregates available resources on the underlying layer and guarantees a low-delay access to nearby users through a pool of virtualized resources. An example of this layer is the concept of vehicular clouds. Vehicular or Road clouds are mobile scenarios formed by vehicular networks providing Cloud Computing features [18].

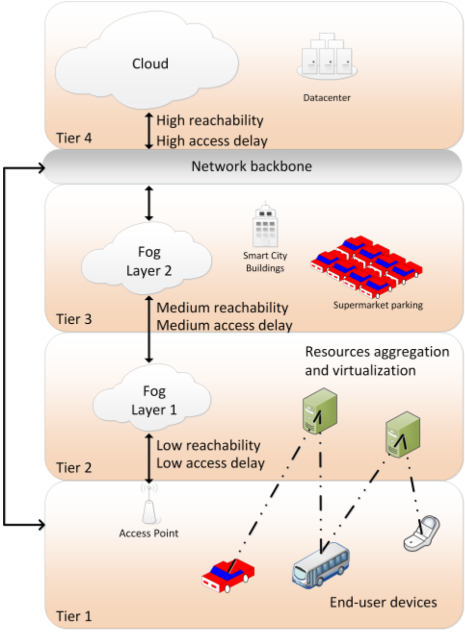


Fig. 1. Combined Fog-Cloud architecture with four tiers.

- The third tier of the proposed topology is composed by a second layer of Fog servers embracing nomadic as well as fixed nodes. This layer aims at the resource aggregation in a neighborhood wide area, enabling a collaborative sharing with medium capacity and latency. For instance, fog premises in public buildings or vehicles in a parking lot may share their resources, enabling the creation of a micro datacenter able to offer medium access latency.
- The fourth tier consists in Cloud servers, accessible through the network backbone, providing almost unlimited resources at the cost of a high latency.

In the following section, we describe the mathematical formulation of the presented model also introducing the most notorious characteristics of the resource allocation problem in a simple scenario.

B. Service Allocation Problem

The deployment of architectures based on the IoT concept faces plenty of difficulties. In this paper, we focus on the resource selection and services allocation considering a simplified use case, assuming: i) all service demands are coming from nodes directly connected to the first Fog layer, see Fig. 1; ii) services are categorized into two distinct service types in terms of the amount of required resources, as described later; iii) all services require the same type of resource, e.g., CPU, and; iv) the total capacity of each node is represented as the total amount of available slots, i.e., the slot is the measurement unit used to represent the minimum resource allocation.

The current Network State Information (NSI) as well as the IT resources information are considered as a static graph, which restricts the amount of nodes only to both the accessible resources in a given instant and the total slots available in each node, as shown in Fig. 2. We consider an IoT resource

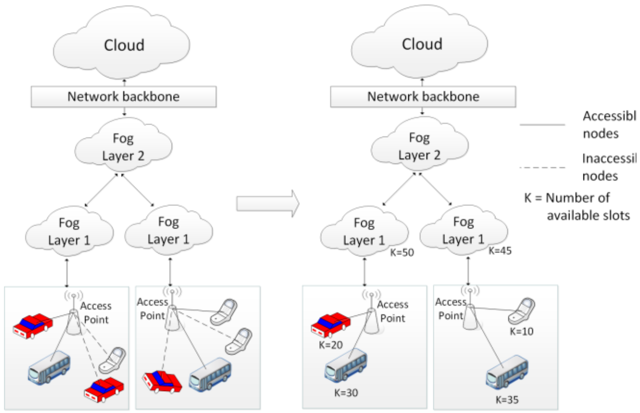


Fig. 2. Used NSI according to reachability of IoT user-nodes.

accessible when a connection to this resource may be set regardless of the number of available slots. It is worth mentioning that, as aggregation points, the Fog layer 1 nodes capacity is precisely the sum of individual capacities of each accessible end-user device in the underlying layer, as showed in Fig. 2. Moreover, the distribution of a service in slots of distinct resources (including resources in distinct layers) is also permitted.

Leveraging the presented topology representation, we propose to model the Service Allocation problem as an ILP problem. In order to accomplish that, we highlight two main goals, first to obtain a low latency on mobile services, and second, the need to provide the highly demanding requirements of existing but also future IoT services. Considering this, in the modeled ILP problem, our main objective is to minimize the total delay for services requesting resources. In the model, the delay D observed by the set of services S is minimized according to (1). For the sake of understanding, Table 1 defines the metric values used in this paper.

$$\text{min: } \sum_{i=0}^{|S|} D_i \tag{1}$$

Furthermore, we consider two distinct types of services in terms of required computational capacity. These services are based on the so-called mice and elephants effect, where the mice represent a big amount of services with low requirements, i.e., a few number of slots, whilst the elephants represent a small number of services requiring a large number of slots. Regardless of total service requirements, we consider a service is allocated after successful allocation of all required slots. Thus, the objective function must comply with the constraint showed in (2).

$$\sum_{r=0}^{|R|} \sum_{k=0}^{|K_r|} Y_{i,r,k} = U_i, \forall i \in S \tag{2}$$

where $Y_{i,r,k}$ is an integer linear variable defined as:

$$Y_{i,r,k} = \begin{cases} 1, & \text{if service } i \text{ is allocated in slot } k \text{ of resource } r \\ 0, & \text{otherwise} \end{cases}$$

In order to cope with resources and slots limitations, the model must also observe (3) and (4). Equation (3) is a capacity

TABLE I. SYMBOLS DEFINITION

Symbol	Definition
S	Set of services requiring IT resources to be executed
R	Set of accessible IT resources offered by IoT devices
K_r	Set of slots of a specific resource *r*, i.e, respective node capacity
U_i	Requirements of service *i* in terms of number of slots
N_r	Slot allocation time for a resource *r*
D_i	Total delay of resources allocation for service *i*

constraint, which avoids the allocation of more slots than the total available for each resource, whereas Equation (4) avoids the same slot of a resource to be used by more than one service simultaneously. It is worth mentioning that the model meets the described topology, where two sort of Fog nodes with distinct capacities, i.e., distinct number of slots are considered according to the layer they are located, as well as one Cloud node able to run all services in S, i.e., with unlimited capacity in the problem scope.

$$\sum_{k=0}^{|K_r|} \sum_{i=0}^{|S|} Y_{i,r,k} \leq K_r, \forall r \in R \tag{3}$$

$$\sum_{i=0}^{|S|} Y_{i,r,k} \leq 1, \forall r \in R, \forall k \in K_r \tag{4}$$

Finally, in order to obtain the delay for each service allocation, we compute the delay added by each slot allocation in distinct nodes, as shown in (5). Hence, the delay added by a node *r* is denoted by N_r, so that the delay added by nodes in the first Fog layer is lower than the delay added by nodes in the second Fog layer. Aligned to that, the delay added by the Cloud is higher than the delay added by the underlying Fog nodes.

$$\sum_{r=0}^{|R|} \sum_{k=0}^{|K_r|} Y_{i,r,k} \times N_r = D_i, \forall i \in S \tag{5}$$

One may notice that the results obtained by (5) take into account the delay for each allocated slot of a service. In order to compute the total delay for each service, we implement two distinct approaches. In the first approach (serial), when a service requires two or more slots from the same resource, we consider the delay of just one allocation for this resource. For instance, consider that the delay of the nodes of the Fog first layer is 1 time unit. In this sense, if a service requiring 3 slots is allocated on 2 distinct nodes in the first layer, e.g., 2 slots of resource *r1* and 1 slot of resource *r2*, then the final delay is $N_{r1} + N_{r2} = 2$ time units, where N_{r1} and N_{r2} are the delays to allocate resources *r1* and *r2* respectively.

On the other hand, the second approach (parallel) takes into account the possibility of parallel allocation observed on several IoT services. For instance, consider a service requiring 3 slots, which are distributed among Fog layer 1, Fog layer 2 and Cloud, respectively with delays 1, 2 and 10 time units. In this parallel approach, the final delay is 10 time units if all slots are successfully allocated, whilst the previous serial version faces a 13 time units delay. In the following section, we show the results obtained by the model, as well as a comparison between both approaches.

IV. Results

This section presents the results achieved by the presented model, obtained using PuLP [19] and Gurobi Optimizer [20]. Each presented value is an average of 30 executions of the implemented model. We first introduce the testbed scenario, and later we present and discuss the achieved results.

A. Evaluation Setup

For sake of simplicity, we assume that all Fog nodes in a specific layer can offer the same amount of slots, whilst the available Cloud capacity is always enough for the execution of all services in the simulation trials. Moreover, individual slots in distinct resources and layers are identical, that is, distinct slots may offer the same sort of IT resources. As a consequence, in this work, we distinct services only by the capacity (number of slots) required to fully allocate them. Hence, in the presented results we use the mice and elephants terminology, representing respectively, the large amount of services requiring few slots in opposite to the low amount of services requiring higher number of slots.

The values used to carry out the simulation driving the presented results are shown in Table 2. In our simulations, the total number of services ranged from 10 to 90, whereas the total capacity of the Cloud node was the sum of all services requirements.

B. Evaluation Results

Fig. 3 shows the experienced average delay for a service allocation versus the number of requested services. These results were split into two groups in order to facilitate the comparison of service allocation time when considering the total time for a service allocation as the sum of delays in a serial execution (first approach), and also when considering the delay as the maximum delay in a parallel execution (second approach). Further, we show a comparison for both types of services, mice and elephants. The obtained results show that mice and elephants delays are very similar in a parallel allocation approach, i.e., when slots from distinct resources and layers are allocated simultaneously. On the other hand, in a serial allocation approach, one should first notice that, as expected, all delays are higher in comparison to the parallel delay independently of the service type.

Moreover, it is also noticeable the fact that, in serial allocation, elephants delays are considerably higher than mice delays, in contrast to the parallel allocation approach. This may be explained by the fact that the distributed allocation of a service may increase the number of distinct resources used in the same layer of the model. Consequently, an elephant has a higher probability of being distributed in a higher number of distinct resources. Nevertheless, this distribution among resources on the same layer does not significantly increase the final delay of services allocation in the parallel approach. On the other hand, in a serial allocation, even when distributing among nodes on the same layer, the allocation of an elephant may result in a higher delay, due to the higher probability of using a larger amount of distinct nodes.

TABLE II. Simulation Parameters

Parameter	Value
Number of Fog layer 1 nodes	4
Number of Fog layer 2 nodes	2
Number of Cloud nodes	1
Total capacity of each Fog layer 1	20 slots
Total capacity of each Fog layer 2	100 slots
Delay of Fog layer 1 nodes	1 time unit
Delay of Fog layer 2 nodes	2 time units
Delay of Cloud node	10 time units
Percentage of mice	90% of number of services
Percentage of elephants	10% of number of services
Mice requirements	3 slots
Elephants requirements	30 slots

Fig. 4(a) shows the percentage of allocated slots in distinct layers of the CFC scenario versus the number of services requesting resources. The percentage of allocated slots in distinct layers for parallel and serial approaches presents the same behavior, in the sense that both cases prioritizes the usage of the nodes offering lowest delays, i.e. in the lowest layers. Thus, resources in Fog layer 1 are selected until the extinguishment of their available slots. This behavior is shown in Fig. 4(b), depicting the number of allocated slots in distinct layers versus the number of services. We show how the number of allocated slots in Fog layer 1 keeps growing up to reaching its 30 services capacity limit. For higher number of services the number of slots allocated to Fog layer 1 stays constant at 30, and services are allocated to Fog layer 2, that keeps growing up to its limit of 50 services. This strategy would scale up for scenarios with more fog layers. We can also observe that cloud resources are only used when there are no available slots on any fog layer.

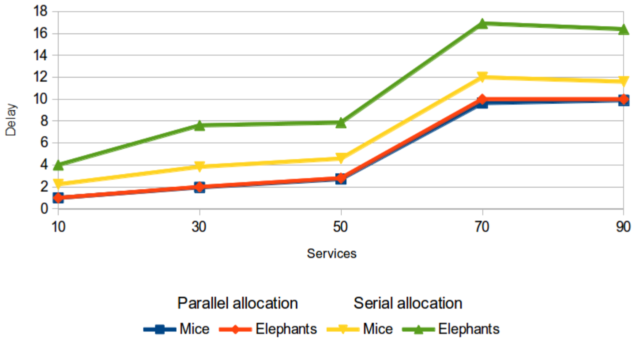
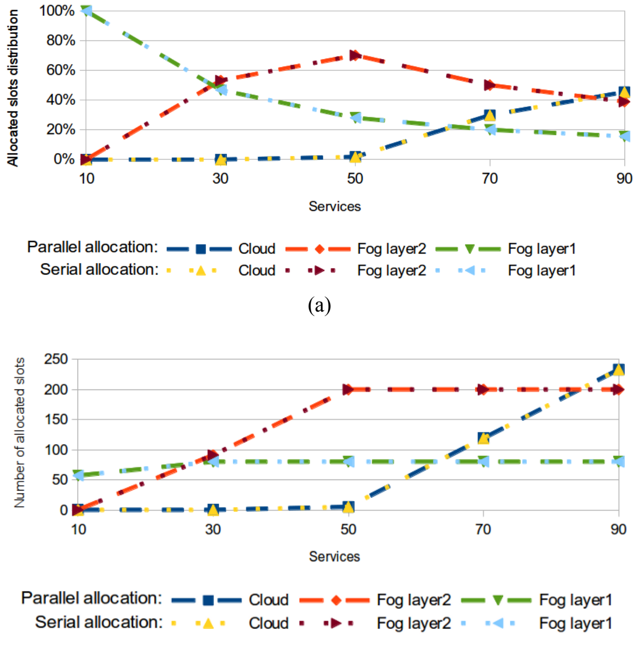


Fig. 3. Delay versus total services.



(a)

(b)

Fig. 4. Nodes allocation per CFC layer versus number of services.

Furthermore, the slots allocation on each layer, depicted in Fig. 4, explains the growth on the delay shown in Fig. 3 when increasing the number of services from 50 to 70. Once slots of Cloud nodes are selected for allocation due to the extinguishment of available resources in Fog layer 1 and 2, the services experience an increase in the average delay.

On the other hand, the average delay is not increased when more than 70 services are present due to the fact that we consider just one Cloud node, which means that there is no distribution among distinct Cloud nodes. As a consequence, the delay added by the Cloud is computed just once, regardless the number of allocated slots. Furthermore, one can observe a slight decrease on the average delay in the serial allocation when the number of services increases to 90. Indeed, this occurs because all added services are allocated completely in the Cloud due to the lack of resources in the Fog nodes. Consequently, the total delay is just the delay added by the Cloud server, i.e., 10 time units. Thus, the average delay starting from this point also tends to 10 time units.

V. Conclusion and Future Work

This work presented a novel Combined Fog-Cloud architecture consisting of a dual-layer Fog aiming to diminish the Cloud access delay in an IoT scenario. With this approach, a service may avail geographically distributed network elements in scenarios such as Smart Cities or Smart Transportation Systems, among others, diminishing the necessity of demanding further Cloud resources preventing high delays. The presented results prove the benefits of service distribution among multiple low-delay Fog nodes avoiding the high delay access on upper layers. Furthermore, the employment of a second Fog layer enables a low delay in a scenario with medium number of service requests, where there are not enough IoT resources available in a single hop wireless connection, but, in the other hand, there is not a demand of connecting to the Cloud. Future works include the categorization of slots in terms of the offered resources, as well as, the distinction among services taking into account their requirements according to each slot category, for instance its maximum allowed delay. Furthermore, we will study the impact on time overhead produced by the service distribution.

Acknowledgements

This work was supported by the Spanish Ministry of Science and Innovation under contract TEC2015-66220-R. Vitor B.C. Souza is supported by CAPES Foundation, Ministry of Education of Brazil, proc. no 11888/13-0.

References

[1] Perera, C.; Zaslavsky, A.; Christen, P. & Georgakopoulos, D. Context Aware Computing for The Internet of Things: A Survey *Communications Surveys Tutorials, IEEE,* 2014, *16*, 414-454

[2] Heilig, L. & Voss, S. A Scientometric Analysis of Cloud Computing Literature *Cloud Computing, IEEE Transactions on,* 2014, *2*, 266-278

[3] Bonomi, F.; Milito, R.; Zhu, J. & Addepalli, S. Fog Computing and Its Role in the Internet of Things *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing, ACM,* 2012, 13-16

[4] Aazam, M. & Huh, E.-N. Dynamic resource provisioning through Fog micro datacenter Pervasive Computing and Communication Workshops (PerCom Workshops), 2015 IEEE International Conference on, 2015

[5] Masip-Bruin, X. et al. Foggy clouds and cloudy fogs: a real need for coordinated management of fog-to-cloud (F2C) computing systems. IEEE Wireless Communications Magazine, June 2016

[6] Nahir, A.; Orda, A. & Raz, D. Resource allocation and management in Cloud Computing *Integrated Network Management (IM), 2015 IFIP/IEEE International Symposium on,* 2015, 1078-1084

[7] Donadio, P.; Fioccola, G.; Canonico, R. & Ventre, G. Network security for Hybrid Cloud *Euro Med Telco Conference (EMTC), 2014,* 2014, 1-6

[8] Parikh, S.A, Survey on cloud computing resource allocation techniques *Engineering, 2013 Nirma University International Conference on,* 2013

[9] Lin, W.; Peng, B.; Liang, C. & Liu, B. Novel Resource Allocation Model and Algorithms for Cloud Computing *Emerging Intelligent Data and Web Technologies, 2013 Fourth International Conference on,* 2013

[10] Dai, J.; Hu, B.; Zhu, L.; Han, H. & Liu, J. Research on dynamic resource allocation with cooperation strategy in cloud computing *System Science, Engineering Design and Manufacturing Informatization (ICSEM), 2012 3rd International Conference on,* 2012, *1*, 193-196

[11] Rogers, O. & Cliff, D. A financial brokerage model for cloud computing *Journal of Cloud Computing, Springer-Verlag,* 2012, *1*

[12] Kaewpuang, R.; Niyato, D.; Wang, P. & Hossain, E. A Framework for Cooperative Resource Management in Mobile Cloud Computing *Selected Areas in Communications, IEEE Journal on,* 2013, *31*

[13] Madsen, H., et al. Reliability in the utility computing era: Towards reliable Fog computing, *Systems, Signals and Image Processing (IWSSIP), 2013 20th International Conference on,* 2013, 43-46

[14] Al Faruque, M. & Vatanparvar, K. Energy Management-as-a-Service Over Fog Computing Platform *Internet of Things Journal, IEEE,* 2015

[15] Deng, R.; Lu, R.; Lai, C. & Luan, T. Towards power consumption-delay tradeoff by workload allocation in cloud-fog computing *Communications (ICC), 2015 IEEE International Conference on,* 2015

[16] Wang, X., et al., Energy and Delay Tradeoff for Application Offloading in Mobile Cloud Computing *Systems Journal, IEEE,* 2015, *PP*, 1-10

[17] Zhang, X. M.; Zhang, Y.; Yan, F. & Vasilakos, A. Interference-Based Topology Control Algorithm for Delay-Constrained Mobile Ad Hoc Networks *Mobile Computing, IEEE Transactions on,* 2015, *14*, 742-754

[18] Baby, D., et al. *(Eds.)* VCR: Vehicular Cloud for Road Side Scenarios *Advances in Computing and Information Technology, Springer Berlin Heidelberg,* 2013, *178*, 541-552

[19] Optimization With PuLP. Available: http://www.coin-or.org/PuLP/

[20] Gurobi Optimization. [Online]. Avaliable: http://www.gurobi.com/