Service Load Balancing in Fog-based 5G Radio Access Networks

Jofina Jijin, Boon-Chong Seet, Peter Han Joo Chong, and Hazim Jarrah
Department of Electrical and Electronic Engineering
Auckland University of Technology
Auckland 1010, New Zealand
{jofina.jijin, boon-chong.seet, peter.chong, hjarrah}@aut.ac.nz

***Abstract*—Fog-based radio access networks (F-RAN) are poised to play a significant role in future 5th generation (5G) cellular networks by harnessing the distributed resources of collaborative edge devices to deliver localized RAN services to the end-users. Through ingestion and processing of end-user tasks close to their sources, F-RAN has potential to meet the stringent latency and bandwidth requirements of 5G services and applications. Due to the multitude of edge devices with different resource capacities that can be selected by the fog access point (FAP) as service nodes to process an end-user task, an important issue is deciding which service node(s) should be assigned to process what tasks from end-users in the F-RAN. The tasks here refer to user processing tasks that would have normally performed by the FAP or remotely by the cloud. In this paper, we introduce the concept of virtual FAPs (v-FAPs) formed by a number of local devices such as WiFi access points, femtocell base-stations and the more resource-rich end-user devices under the coverage and management of the FAP. This paper focuses on addressing the task assignment problem by proposing a *service load balancing* algorithm for the v-FAPs. We model the user tasks as a task graph and the service nodes as an edgeless service graph. We then formulate an optimization problem to find the optimal assignment of tasks with the objective of balancing loads at the service nodes.**

Keywords—fog computing; radio access network; task assignment; load balancing; 5G.

I. Introduction

The future 5th Generation (5G) cellular networks will not only focus on providing high-speed and reliable human communication services, but also support communications between billons of smart objects or 'things' in the coming era of the Internet of Things (IoT) [1]. To achieve these goals, centralized radio access networks (C-RAN) was introduced where data received by base stations (BSs) are transmitted over fiber links to a central unit for processing on specialized hardware [2]. Recently, the concept of Cloud-RAN was proposed to replace the specialized hardware in C-RAN with commodity cloud-computing platform to allow for more flexible splitting and allocation of RAN functionalities between radio access points (RAPs) and cloud, depending on the available cloud resources. However, Cloud-RAN has following challenges [3, 4]: i) constrained backhaul capacity; ii) load concentration on centralized base band unit (BBU) pool; and iii) ultra low-latency requirements of 5G.

Fog-based Radio Access Network (F-RAN) is a promising candidate to tackle the aforementioned challenges by harnessing distributed resources of collaborative edge devices to deliver localized RAN services to end-users [5]. Its predominant philosophy is to make full use of local radio signal processing, cooperative radio resource management (CRRM) and distributed storage capability in edge devices [6]. Through ingestion and processing of end-user tasks close to their sources, F-RAN has potential to meet the stringent latency and bandwidth requirements of 5G services and applications. Fog may be considered as a more general concept of mobile edge computing (MEC) [7,8].

The F-RAN comprises mainly of geo-distributed fog access points (FAP) that may serve end-users directly or through other supporting edge devices acting as service nodes [9]. FAPs can be implemented as dedicated fog servers, or fog-enabled remote radio heads (RRHs) or

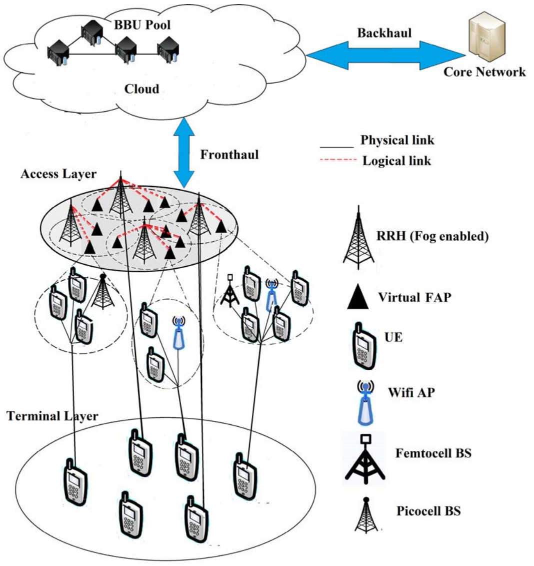


Fig. 1. F-RAN architecture

978-1-5386-3531-5/17/$31.00 ©2017 IEEE

macrocell base stations. The service nodes can be WiFi access points, femtocell base stations, or the more resource-rich end-user devices such as tablets and the high-end smartphones that can be incentivized to lease their processing resources and collaboratively serve other end-users. In this paper, we introduce the concept of virtual FAPs (v-FAPs) formed by a number of such local devices under the coverage and management of the FAPs. Fig. 1 shows the F-RAN architecture with the proposed v-FAPs at the access layer.

Although F-RAN is promising, its performance may be limited by the service nodes that constitute the v-FAPs, which are less resourceful compared to the FAPs or the core cloud. Overloaded service nodes may become single points of failure in the system. Therefore, an important issue is deciding which service node(s) should be assigned to process what tasks from end-users in the F-RAN. The tasks here refer to user processing tasks that would have normally performed by the FAP or remotely by the cloud.

As shown in Fig. 1, end-users (referred to as client nodes hereinafter) requests for help from its nearby FAP, which in turn dynamically forms a v-FAP from a set of service nodes in the client's locality. The FAP decides the set of service nodes to serve the client and the workload assignment to each of the service nodes based on their resource availability. We consider heterogeneous service nodes having different resource capacities (thus different costs of utilizing the resources). Three resource types: computation, storage, and communication resources, can be considered when assigning tasks from client node to each service node.

This paper focuses on addressing the task assignment problem by first modeling user tasks as a task graph and service nodes as an edgeless service graph. We then formulate an optimization problem to find the optimal assignment of tasks with objective of balancing loads at service nodes. The rest of the paper is organized as follows. Section II reviews the related works. Section III formulates the problem and presents the proposed service load balancing scheme. Numerical results are discussed in Section IV. Finally, Section V concludes the paper with some suggestions for future work.

II. Related Works

The authors in [10] focused on achieving ultra-low latency in F-RAN and proposed an algorithm to determine the optimal number of F-RAN nodes (small- and macro-cell BSs) and the amount of resources for a given distributed computing scenario. In [11], related works on computation offloading in MEC by user equipment (UE) were discussed. Various strategies for UEs to offload execution of their mobile applications to MEC nodes (small- and macro-cell BSs as in [10]) were explored with most focusing on enhancing UE's energy efficiency while reducing the execution delay of offloaded applications [12]. Little has been considered about the energy consumption or varying load at the side of MEC. To avoid using the remote cloud while enhancing the user's quality of experience (QoE), the formation of a femto-cloud (a coalition of femtocell access points) for collaborative processing is investigated in [13]. A cooperative game approach to forming the femto-cloud is proposed such that the available computation resources are maximally exploited while participating femtocell access points are monetarily rewarded in a fair manner.

In [14], the authors proposed an algorithm for selecting small-cell BSs in a small-cell cloud (similar to femto-cloud in [13]) to process offloaded applications from UEs. The algorithm takes into consideration of both UE's requirements and the status of small-cell BSs in order to achieve high user quality of service (QoS) while maintaining relatively balanced computation load among the small-cell BSs. The placement of decomposable application components onto physical MEC nodes is investigated in [15]. The user application and physical nodes are modelled as graphs whose nodes and edges represent the computation and communication resource entities, respectively. Several algorithms for placing the application to physical graphs in different scenarios are proposed with the aim of balancing the load and minimizing the sum resource utilization at the physical nodes.

The above existing works are mainly focused on offloading or placement of application computation to base-station type nodes in MEC, unlike our work in this paper, which addresses the RAN task assignment problem to a virtual group of co-located edge devices, including user equipment (UE), in F-RAN.

III. Problem Formulation and Proposed Scheme

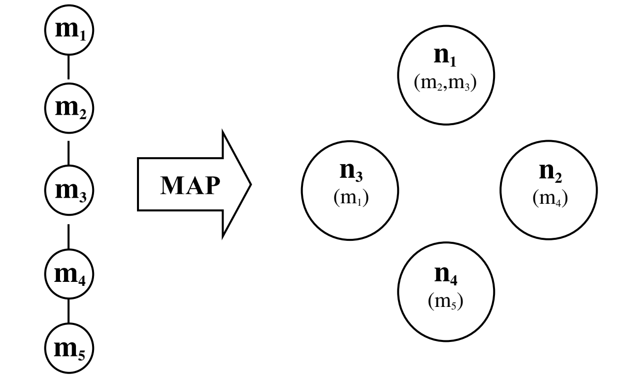
In the following, we firstly define the terminologies and objective function used in this paper. The proposed service load balancing algorithm and a greedy algorithm to which the proposed is benchmarked against are then presented.

A. Definitions

Task graph: The task to be processed is abstracted as a graph in which nodes represent the tasks and edges between the nodes represent the relationships, e.g. execution sequence, of the tasks. Each node $m \in M$ in the task graph is associated with parameter $r_{m,k}$ that represents the demand for each resource type $k \in K$ by task m. The types of resources may include but not limited to computation, storage, and communication resources.

Service graph: The service nodes that may be selected to serve a client are also represented as nodes in an edgeless graph. Each node $n \in N$ in the graph is associated with parameter $u_{n,k}$ that represents the unit cost of using k-type resource of n^{th} service node for a given task, and is defined to be inversely proportional to the resource capacity, i.e. smaller the capacity of a resource, higher the cost of its use, and vice-versa.

Mapping: A mapping defines a specific pattern by which a client's tasks are assigned to the service nodes. Fig. 2 shows an example of mapping tasks from a client to available service nodes.



Task graph of a client node

Service graph with mapped tasks from client node

Fig. 2. Example of mapping tasks to service nodes with *M*=5 and *N*=4

B. *Formulation*

For a particular client, the cost of using a j^{th} mapping from a set of all possible mappings π for assigning *M* tasks to *N* service nodes, each with *K* types of resources needed for task execution, is given by:

$$c_j = \sum_{n=1}^{N} v_{j,n} \qquad (1)$$

where $v_{j,n} = \sum_{k=1}^{K} w_{n,k}$ is the cost of using n^{th} service node in j^{th} mapping, $w_{n,k} = u_{n,k}\, D_{n,k}$ is the cost of using resource *k* in *n*, and $D_{n,k}$ is the total demand for resource *k* by a set of tasks $M_n \subseteq M$ assigned to *n*. Denote $R_{n,k} = \{r_{1,k},..,r_{|Mn|,k}\}$ as the set of demands for resource *k* by the tasks in M_n. Then $D_{n,k} = \sum R_{n,k}$ if M_n is non-empty, or zero otherwise.

Objective function: The optimization objective function of the proposed algorithm is defined as:

$$\min_{1..J} \max_{1..N} \{v_{j,n}\} \qquad (2)$$

where *J* is cardinality of set π. Equation (2) determines the mapping which has the lowest maximum cost service node.

On the other hand, the objective function of a conventional greedy scheme that simply determines the mapping which has the minimum total cost is defined as:

$$\min_{1..J} \{c_{i,j}\} \qquad (3)$$

The pseudo-code of the proposed and greedy schemes are shown in Algorithm 1, and Algorithm 2, respectively. The proposed algorithm not only considers the cost of resource utilization but also load balancing whereas the greedy algorithm emphasizes mainly on minimizing the overall cost of resource utilization. Both the proposed and greedy algorithms require an optimization problem to be solved as a subroutine, for each client's tasks assignment.

Algorithm 1. Proposed scheme

```
for $j = 1$ to $J$ do  // for each mapping
    for $n = 1$ to $N$ do  // for each service node in a mapping
        for $k = 1$ to $K$ do  // for each resource in a service node
            $D_{n,k} = \sum R_{n,k}$  // find total demand for resource k in n^th service node
            $w_{n,k} = u_{n,k}\, D_{n,k}$  // find cost of using resource k in n^th service node
        end for
        $v_{j,n} = \sum_{k=1}^{K} w_{n,k}$  // find cost of using n^th service node in j^th mapping
    end for
end for
$mapping \leftarrow \min_{1..J} \max_{1..N} \{v_{j,n}\}$  // select mapping which has the lowest
return mapping  // maximum cost service node
```

Algorithm 2. Greedy scheme

```
for $j = 1$ to $J$ do  // for each mapping
    for $n = 1$ to $N$ do  // for each service node in a mapping
        for $k = 1$ to $K$ do  // for each resource in a service node
            $D_{n,k} = \sum R_{n,k}$  // find total demand for resource k in n^th service node
            $w_{n,k} = u_{n,k}\, D_{n,k}$  // find cost of using resource k in n^th service node
        end for
        $v_{j,n} = \sum_{k=1}^{K} w_{n,k}$  // find cost of using n^th service node in j^th mapping
    end for
    $c_j = \sum_{n=1}^{N} v_{j,n}$  // find total cost of using j^th mapping
end for
$mapping \leftarrow \min_{1..J} \{c_{i,j}\}$  // select mapping which has the minimum total cost
return mapping
```

IV. Evaluation

In this paper, we focus on balancing service loads in F-RAN by enhancing the overall resource utilization of the service nodes through more effective task assignments. Thus, a key performance metric is the standard deviation (SD) of the normalized load of service nodes. We expressed the normalized load of a service node as a percentage of its total resource capacity consumed for processing its assigned tasks.

We implemented the proposed and greedy schemes in MatLab and evaluated their performance under varying number of service nodes and service tasks. Table 1 summarizes the simulation parameters and values used.

TABLE I. Simulation Parameters

Parameter	**Value(s)**
Number of service tasks (*M*)	1, 3, 5, 7, 9
Number of service nodes (*N*)	2, 4, 6, 8, 10
Number of resource types (*K*)	3
Number of mappings (*J*)	$\frac{(N+M-1)!}{M!\,(N-1)!}$
Number of simulation runs	100

We considered a heterogeneous real-world like scenario where each service node has different available resource capacity, and each client task has different resource demand from the service nodes. We simulated these heterogeneous quantities as random numbers drawn from a bounded uniform distribution.

We further considered two cases. In the first case, the FAP (e.g. fog-enabled RRH) which manages the v-FAPs under its coverage is assumed to have full knowledge of the resource capacity of individual service nodes that constitute each v-FAP. Thus, the FAP can ensure that only a service node whose resource capacity meets the resource demand of a task will be assigned with the task. However, this may require non-trivial amount of information exchanges between the FAP and service nodes, which will only increase with the number of service nodes and number of resource types considered. Therefore, we investigate a second case where FAP does not have such information and evaluate the potential for *mapping failure*, which we defined as an event when one or more service nodes of a mapping chosen by the proposed or greedy algorithm based on cost considerations are unable to perform their assigned tasks due to insufficient resources.

Firstly, we analyze the results of the first case. Fig. 3 shows the impact of the number of service nodes *N* in a v-FAP on the SD of both schemes under two service loads: *M*=3 and *M*=7 tasks. The result shows that the SD of the proposed scheme is consistently lower than that of greedy scheme, which reflects a better load balanced performance. This can be attributed to the proposed scheme using mappings that have the lowest maximum cost service node, which tends to spread the total load across more service nodes. It is also observed that for both schemes, the SD reduces as *N* increases. This is expected as more service nodes are available to share the total processing load, which in turn reduces the load on any particular service node. Furthermore, as *M* increases from 3 to 7, the SD of the greedy scheme increases more significantly than the proposed scheme. This is because the greedy scheme always prefers to use lower cost nodes, which are loaded more heavily as the total processing load increases.

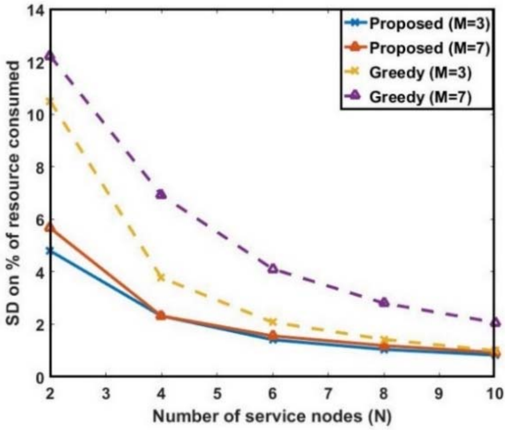


Fig. 3. Impact of service nodes on standard deviation of resource consumed with full knowledge of service node resource capacity.

Fig. 4 shows the impact of the number of service tasks *M* from client on the SD of both schemes for a given number of service nodes: *N*=4, and *N*=8. The result shows that the SD of both schemes increase with *M*, but the rate of increase of the proposed scheme is much slower due to its more even spreading of the load among the service nodes. Similar to the result in Fig. 3, it is also observed that the performance of both schemes are better when *N* is increased from 4 to 8.

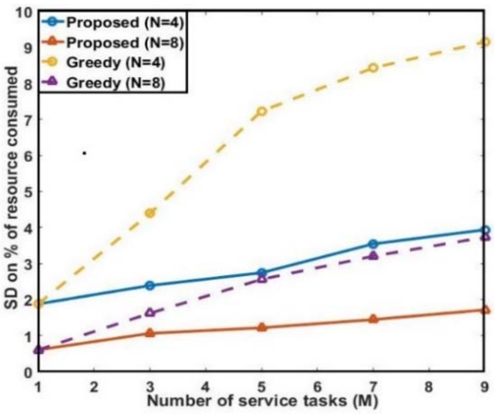


Fig. 4. Impact of service tasks on standard deviation of resource consumed with full knowledge of service node resource capacity.

Next, we analyze the results of the second case. As in Fig. 3, Fig. 5 shows the impact of *N* on the SD of both schemes for *M*=3 and *M*=7, but under the scenario that FAP has no knowledge about the resource capacity of individual service nodes, which may be more practical for implementation. The result shows that in comparison with the greedy scheme, the SD of the proposed scheme increases only marginally even when the number of service tasks is increased from 3 to 7, which reflects its resiliency in maintaining a load balanced performance. The other performance trends remain similar to those in Fig. 3.

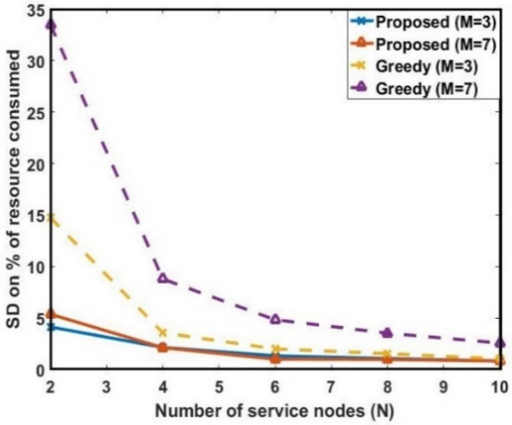


Fig. 5. Impact of service nodes on standard deviation of resource consumed with no knowledge of service node resource capacity.

Similarly, Fig. 6 shows the impact of *M* on the SD of both schemes given *N*=4, and *N*=8. Generally, there is no significant difference in performance of the proposed scheme, while SD for greedy scheme is higher than that in Fig. 4, particularly with low number of service nodes, e.g. *N*=4. The other performance trends remain similar to those in Fig. 4.

Table II shows a comparison of mapping failure rates between the proposed and greedy schemes. As defined earlier, the mapping chosen by a scheme fails when the resource demand of a task exceeds the resource capacity of the service node assigned by the mapping to process the task.

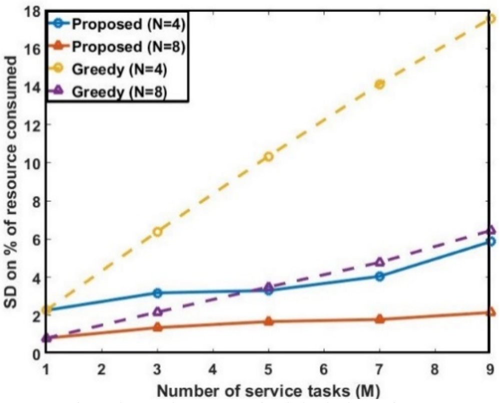


Fig. 6. Impact of service tasks on standard deviation of resource consumed with no knowledge of service node resource capacity.

TABLE II. FAILURE RATE COMPARISON

Scheme	Mapping Failure Rate (%)					
	No. of Service Nodes (*N*) (with *M*=5)			No. of Service Tasks (*M*) (with *N*=6)		
	N=2	*N*=6	*N* =8	*M*=1	*M*=5	*M*=9
Proposed	2	0	0	0	0	6
Greedy	56	41	39	0	41	93

The result shows that the failure rate of the proposed scheme is consistently lower than the greedy scheme for all considered scenarios (comprising of 5 different combinations of *N* and *M*). The proposed and greedy scheme suffers a failure rate of up to 6%, and 93%, respectively. For both schemes, the trends of the failure rate are consistent with those of SD in Figs. 5 and 6, i.e. the high failure rate occurs under the same circumstances (defined by *N* and *M*) where SD is high, and vice-versa. The low failure rate of the proposed scheme is encouraging, particularly given the fact that it was achieved without the need for FAP to know the resource capacity of individual service nodes.

Finally, Table III compares both the failure rate and SD between the proposed and greedy schemes as the number of service tasks (*M*) is further increased for a given number of service nodes. The results show that the failure rate of the greedy scheme reaches 100% when the number of service tasks increases to 15, while it takes a larger load of 25 service tasks for the proposed scheme to suffer a similar failure. The proposed scheme also widens its SD performance gap over its greedy counterpart, demonstrating its better scalability.

TABLE III FAILURE RATE AND STANDARD DEVIATION COMPARISON

M with *N*=6	Mapping Failure Rate (%)		Standard deviation (%)	
	Proposed	Greedy	Proposed	Greedy
15	34	100	4.11	16.89
20	76	100	4.74	22.21
25	99	100	6.21	27.99

V. CONCLUSION

In this paper, the service task assignment with load balancing has been studied under the context of fog-based 5G radio access networks. We first formulate an optimization problem for mapping a linear task graph to an edgeless service graph with the objective function to minimize the maximum resource costs. A service load balancing algorithm for the v-FAPs is then proposed. To our knowledge, no similar research has been done for F-RAN with v-FAPs. Numerical results show that the proposed scheme can achieve significantly more balanced loads amongst service nodes and lower failure rates due to overloading as compared to a greedy minimum cost scheme. As future work, we will study the formation protocol for v-FAPs and their roles in the optimal functional split between the cloud and fog in future 5G radio access networks.

REFERENCES

[1] F. Boccardi, R. W. Heath, A. Lozano, T. L. Marzetta, and P. Popovski, “Five disruptive technology directions for 5G”, IEEE Communication Magazine, vol. 52, no. 2, pp. 74–80, 2014.

[2] M. Peng, Y. Sun, X. Li, Z. Mao, and C. Wang, “Recent advances in cloud radio access network system architecture, key techniques, and issues”, IEEE Communication Surveys and Tutorials, vol. 18, no. 3, pp. 2282–2308, 2016.

[3] I. Chih-Lin *et al.*, “Recent progress on C-RAN centralization and cloudification,” IEEE Access, vol. 2, pp. 1030–1039, 2014.

[4] D. Wubben, P. Rost, J. Barlett, M. Lalam, V. Savin, M. Gorgogolione, A. Dekorsy and G. Fettweis, “Benefits and impact of cloud computing on 5G signal processing”, IEEE Signal Processing Magazine, vol. 31, no. 6, pp. 35-44, 2014.

[5] Y.-J. Ku *et al.*, "5G Radio Access Network Design with the Fog Paradigm: Confluence of Communications and Computing," IEEE Communications Magazine, vol. 55, pp. 46-52, 2017.

[6] M. Peng, S. Yan, K. Zhang, and C. Wang, “Fog computing based radio access networks: Issues and challenges”, IEEE Networks Magazine, vol. 30, no. 4, pp. 46–53, 2016.

[7] A. V. Dastjerdi, H. Gupta, R. N. Calheiros, S. K. Ghosh, and R. Buyya, “Fog computing principles architecture and applications”, Chapter 7 in Internet of Things: Principles and Paradigms (Eds. Buyya & Dastjerdi), Elsevier, Massachusetts, USA, 2016.

[8] Y. Shih et al., “Enabling low-latency applications in fog radio access networks,” IEEE Network, vol. 31, no. 1, pp. 52–58, 2017.

[9] S. Hung *et al.*, “Architecture harmonization between cloud radio access networks and fog networks,” IEEE Access, vol. 3, pp. 3019–34, 2015.

[10] T. C. Chiu, W. H Chung, A. C. Pang, Y. J. Yu and P. H. Yen, “Ultra-low latency service provision in 5G fog-radio access networks”, in Proc. IEEE 27th Annual International Symposium in Personal, Indoor, and Mobile Radio Communications, 2016.

[11] P. Mach, Z. Becvar, “Mobile-edge computing: a survey on architecture and computation offloading”, in-press IEEE Communications, Surveys and Tutorials, 2017

[12] Y. Zhang, H. Liu, L. Jiao, and X. Fu, “To offload or not to offload: an efficient code partition algorithm for mobile cloud computing”, in Proc. IEEE International Conference on Cloud Networking, 2012.

[13] S. M. S. Tanzil, O. N. Gharehshiran, and V. Krishnamurthy, “Femto-cloud formation: a coalition game-theoretic approach”, in Proc. IEEE Global Communications Conference, 2015.

[14] M. Vondra and Z. Becvar, “QoS-ensuring distribution of computation load among cloud-enabled small cells”, in Proc. IEEE International Conference on Cloud Networking, 2014.

[15] S. Wang, M. Zafer, and K. Leung, “Online placement of multicomponent applications in edge computing environments”, IEEE Access, vol. 5, pp. 2514–2533, 2017.