User Association and Behavioral Characterization during Task Offloading at the Edge

Firdose Saeik*, John Violos*, Aris Leivadeas *, Marios Avgeris†, Dimitrios Spatharakis†, Dimitrios Dechouniotis†
* Department of Software and IT Engineering, École de technologie supérieure, Montreal, Canada
Email: firdose.saeik.1@ens.etsmtl.ca, violos@mail.ntua.gr, aris.leivadeas@etsmtl.ca
† Department of Electrical and Computer Engineering, National Technical University of Athens, Greece
Email: {mavgeris, dspatharakis, ddechou}@netmode.ntua.gr

***Abstract*—Current developments in computer vision and networking have made immersive applications, such as Augmented Reality (AR), Virtual Reality (VR), and Mixed Reality (MR), more affordable. As the driving force behind these types of applications is the high Quality of Service (QoS), more and more studies concentrate on offloading the application tasks to more powerful computing infrastructures without impairing the immersive user experience. This generates the problem of task offloading, defined as the transfer of resource-intensive computational tasks from a local device to an external resource-rich platform such as Cloud and/or Edge computing. Task offloading can be deemed extremely beneficial for low latency applications, however introducing several challenges in terms of task scheduling and allocation. These challenges are usually tackled via traditional optimization algorithms that can output at the same time which segments to offload and to which site (e.g. an Edge or Cloud server). These algorithms usually leverage basic input information such as task size, available computational and communication resources, etc. Going a step beyond, in this work, we propose a novel model that is able to blend the user association information through Social Network Analysis metrics and especially node centrality during the task offloading decision in an Edge infrastructure. Our results show that our approach can reduce the communication delay towards increasing the user experience.**

***Index Terms*—Edge Computing, task offloading, immersive application, quality of service**

I. Introduction

In the context of ubiquitous computing and the user-centered networking background, mobile and personal devices play a catalytic role. Networking and data sharing are expected to seamlessly take place among different mobile devices by taking advantage of their increasing capabilities (i.e., processing, storage, built-in wireless communication technologies, etc.).

Even though the mobile devices' computational capabilities increase, so does the computational requirements of new and emerging applications, such as immersive applications. Thus, lately, to increase the computing efficiency of mobile and personal devices while saving battery power, it is important to move computationally intensive tasks from these devices to a more powerful infrastructure. Traditionally, Cloud had been the de-facto platform for such solutions. Nonetheless, the centralized and remote data centers of the Cloud often impose restrictive delay communications not suitable for delay-critical applications. To this end, the recent trends of Edge Computing have enabled end devices to access the necessary computational resources at the edge of the network minimizing the communication delay.

In particular, end-devices are now capable of offloading their resource-intensive tasks to a nearby Edge device and minimize the overall execution time without adding excessive communication paths towards a distant Cloud infrastructure. This combination of mobile and edge resources can improve application performance by transferring computationally intensive tasks to edge devices. This approach, called task offloading at the edge, is practically a resource allocation and scheduling optimization problem exploring which user's tasks can be offloaded at the edge and how they can be allocated in the available physical infrastructure.

However, task offloading is a rather complex process and can be greatly affected by a number of different factors such as the application partitioning to several tasks, the offloading decision, the distributed task execution and so on [1]–[3]. To address these challenges, usually, multi-objective optimization algorithms are proposed to minimize the delay and/or maximize the lifetime of the battery-powered mobile devices without considering if one device's offloading decision may affect or benefit another device.

This can be extremely important for interactive applications, including immersive applications, where the users are expected to be highly correlated and often consuming the same content. Hence, task offloading could consider the frequent contacts and interactions among the users' devices as stemmed by their behavior.

In reality, the analysis of users' behaviour and their spatial relations to define better interactions (i.e. by recognizing repetitive patterns) has been a field of study in the context of computer science for many years. Such analysis can investigate how users are associated and how they behave according to the services being used. Accordingly, users' interactions and users' behaviour analysis are important for user engagements and a stepping stone to associate the users' behaviour to the perceived Quality of Service (QoS) and Quality of Experience (QoE).

In this paper, inspired by the possible intersection of recent trends in task offloading at the Edge with the user association behavior in the context of an immersive application, we aim

978-1-6654-4505-4/21/$31.00 ©2021 IEEE

to propose a novel user-association and behavioral characterization based task offloading heuristic. Specifically, this approach's main goal is to consider typical user association aspects (i.e. interaction intensity) during the task offloading to minimize the end-to-end communication delay.

To the best of our knowledge, no model for users' association behaviour during task offloading exists yet. Accordingly, this paper presents a first attempt at investigating the end users' behaviour who offload tasks towards maximizing the delay performance achieved. At the same time, we aim to respond to questions that can impact the performance of task offloading, such as i) which users interact with each other and ii) which access points should be used as edge nodes during task offloading.

The rest of the paper is structured as follows. Section II provides a brief literature review. Section III presents the user association models considered in our work. The proposed task offloading solution is found in Section IV. Following, Section V provides the evaluation of our solution in a simulated environment. Finally, Section VI concludes the paper.

II. Related Work

Some user association approaches, which have recently attracted much attention, emphasize on reducing the latency and/or energy, and optimizing resource allocation in a cellular communication context. More specifically, in terms of reducing the latency, [4] aims to study the trade-off between maximizing the energy efficiency and minimizing both the wireless latency and the interference level. Similarly, the authors in [5] focus to minimize latency in a small cell base stations (SBS) scenario by adopting an appropriate caching strategy under Spatio-temporal traffic demands. In [6], authors present user-centric backhauling, which exploits the diversity of the radio and backhaul networks and that of the users QoE expectations and maximize system-centric and user-centric performance. In [7], a theoretical and practical framework is developed for BS energy savings that encompasses dynamic BS operation and user association during latency minimization. In [9], the joint optimization of content placement, user association, and unmanned aerial vehicles' positions is studied, aiming at minimizing the total transmit power of UAVs while satisfying the requirement of user experience.

Interestingly, the above works try to investigate the impact of user association or user behaviour. However, they are not specifically addressing the offloading mechanisms. Accordingly, some interesting studies examine offloading traffic in reducing the latency, bandwidth, and transmission cost [11]–[14]. A simple approach to reduce the bandwidth in the context of device-to-device communication (D2D) is presented in [11] where network traffic is expressed as a transmission cost that needs to be minimized. Similarly, network traffic can be minimized through an intelligent radio spectrum allocation proposed in [12]. Specifically, the authors present a novel spectrum sharing paradigm called inter-operator proximal spectrum sharing (IOPSS). A base station (BS) intelligently offloads users to the neighboring BSs based on spectral proximity to enhance the users' QoE and spectral resource utilization. A similar approach is proposed in [13], where a base station (BS) intelligently offloads users to neighbouring BSs based on spectral proximity while also leveraging the predictability of the user's mobility through a novel spectrum sharing paradigm (memory-based content-aware hybrid scheme). Another simple way to reduce the overall network traffic and thus increase the users' QoE is to decrease the downloading delay of the application addressed in [14].

Regarding task offloading for interactive applications such as VR, AR, the academic and industrial community identified that distant cloud resources are not always suitable. Thus, computational resources closer to users need to be leveraged. Hence, for each computational request, the requested tasks may be computed at various network locations, either locally on the VR/AR device, on the Edge of the network, or globally on the Cloud. Additional computational and delivery/delay costs can be incurred, depending on the location where the tasks are executed [1]–[3], [11].

Finally, only a few studies investigate the user association or user behavior with the task offloading problem at the Edge [15], [16]. Specifically, in [16] a joint computation offloading and user association problem is formulated for minimizing the energy consumption of mobile users and edge servers. The authors proposed a novel multi-user association scheme that takes both computation task size and delay requirement into consideration. Specifically, a mobile user chooses a base station that provides the data rate that satisfies the delay constraint instead of the base station with the maximum data rate. However, delay minimization has not been considered in the user association decision. In [15] the authors propose a heuristic offloading decision algorithm (HODA), which is semi-distributed and jointly optimizes the offloading decision, and communication and computation resources to maximize the system utility. The system utility metric is a QoE measure based on task completion time and energy consumption of a mobile device. Particular emphasis is given on mobile devices' variability in capabilities and user preferences on the user QoE.

Similarly, with [15], our work tries to create a link between the users, the user experience, and the task offloading problem. However, going a step beyond, we try to find how the association between the users and their interactions can affect the user experience (expressed in terms of latency) during the task offloading problem. This encounter between Edge and user interaction (expressed through social theory) is expected to be beneficial. It could facilitate the task offloading mechanism, the data exchange, the optimization of the network and user resources, and new applications.

III. User Association

One of the proposed solutions novelties is its interaction and information-centric nature since this seems to be the basic foundation of applications able to distribute information in edge networks. Hence, the proposed solution will investigate the balance between user association aspects and node interaction when offloading resource-intensive computational tasks.

To duly model this user association, appropriate metrics should be defined as described below.

A. User association metrics

User association has been extensively used in cellular networks, and various models have been proposed, leveraging mobile phone and usage-related features. For example, mobile phone features could be used to predict personality traits, i.e., in physical encounters (people's co-presence) and interactions in the context of the cellular network. The co-presence is defined as the potential for nodes to be in the same community. In our view, this information does not suffice to model nearness to express the association/similarity between users. Nearness is a metric that can be decisive in reducing latency, when users offload common tasks to the same edge location. Usually, the spatial property resulting from a relatively small distance [17] has been used in exploring nearness contextualization [18] via the use of short or medium-range wireless technology (i.e. Bluetooth, Wi-Fi). In contrast, in our study, the nearness definition can be represented by the inter-contact distribution, expressed by the node degree metric, widely used in social network analysis (SNA) [20], [21].

Specifically, for the task offloading problem, user association's inclusion is of considerable importance for offloading as it directly affects the rate of communication data and requests made to offload. Unlike user association schemes in conventional heterogeneous networks [7]–[9], the user association scheme in task offloading at the edge should also take into account both the size of the computation data in terms of million instructions per set (MIPS) and the delay requirements of applications. Based on this, we define the following user association metric to be used in the context of task offloading.

Interaction Intensity: The interaction intensity can be perceived differently in terms of the network point of view and application point of view. From the network perspective, interaction intensity can be considered as the interconnection between the nodes. From the application perspective, interaction intensity can be modeled based on the time the application transfers data (e.g. average inter-contact/contact duration) to the associated base station. In the particular study we model the interaction intensity from the network point of view.

B. User Association through SNA

In the above sub-section we have defined the interaction intensity as the main metric to express the user association. In this sub-section, we try to formally model it by resorting to the SNA theory, and specifically to social communities.

A social community is naturally formed according to social relations among people, and it defines groups of individuals sharing the same social interests or behaviours [19]. In networks, communities may represent real social groupings by location, interests or background, and different communities are usually interested in different mobile contents [25]. Thus, detecting this community information can help improve data transmission efficiency among distributed and intermittently connected users.

Towards this end, a useful SNA metric that can help correlate the interaction intensity model with the community is the centrality. The term centrality evaluates the relative auxiliary significance of a node within a community [20], [22]. Some devices/people are more popular and interact with more devices/people than others, and thus they act as communication hubs in a community. Thus, a central node tends to have a higher proximity-encounter possibility and interaction frequency with the nearby devices. There are several ways to measure centrality; the most widely used are Freeman's degree, closeness, and betweenness measures [20], [23]. Closeness centrality is a metric for assessing whether a user is close to other users in a social network and thus able to communicate quickly with them. Betweenness centrality is a metric to check whether a specific user is an important node that lies in a high proportion of paths between other social networks users. Degree centrality indicates whether someone in a social network is involved in a large number of interactions.

Accordingly, in this paper, we focus on the user association intensity model expressed as the centrality (degree distribution). Specifically, the node that shares the most links (common content) to other nodes will be the most central and appropriate to act as an offloading point (e.g. edge node).

IV. Task Offloading Algorithm

In this section, we model our task offloading algorithm based on the user association scheme presented above and emphasizing on the social awareness aspects such as node centrality for immersive and interactive services in the context of future mobile networks and services.

A. System model

In this subsection, we first introduce the system model, while providing the necessary terminology. Finally, we mathematical formulate the interaction intensity.

Let $G = (V, E)$ be an undirected graph where V represents the set of devices (end users) and E denotes the set of undirected links. Each link characterizes the contextual nearness of devices in accordance with the network point's of view interaction intensity. In other words, two nodes that are connected consume the same content. Existing works focused on link partitions [24] and link communities [25] to create the necessary social communities. In this work, we follow both a link and a node perspective as shown below, when calculating the interaction intensity.

Regarding the Edge infrastructure, it is modeled as a directed Graph:

$$G^E = (H, L) \tag{1}$$

where H is the set of host nodes (Edge nodes) and L the network connections between the nodes. Each Edge node $h \in H$ is characterized by a vector of capacities **U(h)** such that **U(h)** $= (CPU, RAM, Storage)$

Edge nodes are interconnected via a set of links L, where each link $l_{i,j} \in L$ is characterized by a latency value

$LAT(l_{i,j})$ between source node i and destination node j, with $LAT(l_{i,i}) = 0$ and a bandwidth value $BW(l_{i,j})$, with $BW(l_{i,i}) = \infty$. In the particular work, we assume that the Edge nodes are connected according to a full mesh topology.

To mathematical formulate the interaction intensity, we first calculate the centrality per end-user $v \in V$, as follows :

$$Centrality_v = \sum_{e(s,d) \in E, s=v} |e(s,d)| \quad (2)$$

The above Equation, goes through all the links/interactions in the graph G between two end users $s, d \in V$ and finds the number of links (degree), that an end user v has (i.e. when $s = v$.

Based on the centrality, we can now model the interaction intensity as follows:

$$InteractionIntensity = Centrality_v / |E| \quad (3)$$

Specifically, the interaction intensity for an end user v is the fraction of the centrality degree of the same end user v divided by the total number of interactions $|E|$ between all the end users V, where $|E|$ is the cardinality of the link vector E in graph G.

B. Algorithm

The goal of the algorithm 1 is to leverage the user behaviour when offloading tasks towards increasing the delay performance. The ultimate goal is to select the most popular nodes between the end-devices to act as edge nodes and thus to model our edge network.

The algorithm starts by getting as an input the users' interactions (i.e. which users share the same content or field of view) and will output the edge nodes that will be used to execute the offloaded tasks, along with the total execution time.

As a first step, the graph G is generated that contains the end user devices and their interactions. Based on the information provided by the graph, the centrality can be extracted, that will help calculate the interaction intensity, conforming to Eq. 3. Following, we sort the end devices according to their interaction intensity (i.e. popularity) and we select the $|H|$ most significant nodes. In the particular work, we assume that 25% of the nodes will act as edge nodes (e.g. $|H| = 0.25 * |V|$). These edge nodes will formulate our edge infrastructure $G^E = (H, L)$, where the tasks would be offloaded. In particular, each edge node will act as an SBS that will execute the tasks generated from its associated users.

As a next step, we start the simulation by generating the tasks from the users. Tasks are associated with the following information, sender s, receiver r, and time stamp t. If the sender and receiver correspond to the same SBS and share the same content, we calculate directly the task execution time. In particular, the execution time is calculated based on the control theory model proposed in [27]. If the sender and receiver correspond to a different SBS, we need to take into consideration a delay penalty that includes the delay of transferring the task through the edge infrastructure to the right Edge node.

Algorithm 1: User association model - Algorithm

Input: user interactions
Output: offloading decision and execution time
Pseudo code:

1) Create a graph $G = (V, E)$ representing the interactions between the end users
2) Calculate the centrality (degree) of each node $v \in V$ and extract their interaction intensity.
3) Sort nodes in V according to Eq. 3 to extract the interaction intensity
4) Select the most popular nodes $|H| = k * |V|$
5) Create a graph $G^E = (H, L)$
6) Generate tasks t
   a) if source and destination of the task are on the same SBS calculate the task execution time.
   b) if source and destination of the task are on different SBS add the delay overhead of transferring the task between the corresponding SBSs.

V. Performance Evaluation

In order to evaluate our proposed methodology we used the CloudSim Plus simulator [28]. CloudSim Plus is widely used from researchers for modeling, simulation, and experimentation of Cloud computing infrastructures and application services. CloudSim Plus is available on GitHub [1] with multiple useful examples. In our experiments we used typical CloudSim components such as Data Centers, Brokers, Virtual Machines (VM), and Cloudlets.

In particular, we simulated an edge infrastructure topology that consists of one Data Center (that can act as the main macro Base Station), four processing edge nodes (Small-cell Base Stations) and sixteen user devices that continuously generate tasks according to a uniform distribution. Both users and processing edge nodes are distributed in the area that covers the edge computing infrastructure. The simulated application is an augmented reality application with different Field of View (FoV) resolutions that provides a sense of immersion with high quality. The application of FoV supports rendering technologies that run on the processing edge nodes and enables end users to benefit from a high-quality immersive experience. The end users generate tasks that arrive to the edge of the network and at the same time users interact the one with the other in the virtual environment of the application.

The tasks produced by users that interact should be ideally offloaded in the same processing edge nodes. Application users that have significant interaction generate tasks with data and intercommunication dependencies. If these tasks are offloaded to the same processing edge nodes, they will be executed more efficiently compared to be offloaded to different processing

[1] https://github.com/manoelcampos/cloudsim-plus

TABLE I
EXPERIMENTAL EVALUATION.

Method	Makespan	Throughput	Average	Median	Std Dev	Skewness	Kurtosis	Tail Latency
RandomHostMIPS:1000	118.15	2900.77	0.371	0.481	0.129	-0.636	-1.171	0.481
SNA4HostMIPS:1000	118.14	2900.72	0.362	0.481	0.138	-0.520	-1.455	0.481
RandomHostMIPS:1000-3000	118.25	2898.16	0.483	0.494	0.215	0.312	-0.786	0.862
SNA4HostMIPS:1000-3000	118.24	2898.56	0.472	0.492	0.221	0.334	-0.861	0.863

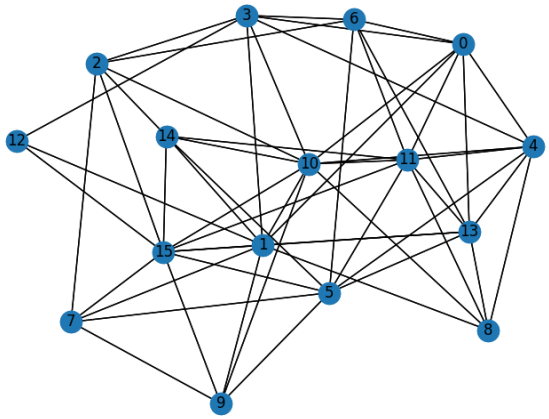

Fig. 1. End-user interaction with random network generator

edge nodes. Otherwise, there will be an additional overhead in communication between processing edge nodes, data transfer and execution time. In our experiments, we will see the improvements in the execution times of the tasks when we apply the offloading algorithm proposed in section IV.

Every user in the simulated Edge infrastructure use the augmented reality application and generates multiple tasks. The interaction between users can be represented with the graph G as modeled in Section IV and an example is depicted in Figure 1. To generate the interaction between users, we used the Erdos–Renyi model via the Networkx tool.

We run four different task generation and offloading cases. In the first case named RandomHostMIPS:1000, the edge node selection happens randomly. In the second case named SNA4HostMIPS:1000, we use the SNA task offloading mechanism. In both cases we use a fixed amount of 1000 MIPS for the tasks. Next we use again the random task offloading mechanism and the SNA task offloading mechanism but with a variable tasks size in a range from 1000 to 3000 MIPS according to the FoV of each user. These cases are named RandomHostMIPS:1000-3000 and SNA4HostMIPS:1000-3000 respectively. During the experiments 342720 tasks are generated.

A. Evaluation Metrics

In our experimental setup we used multiple evaluation metrics. Some of these metrics have different interpretations in the distributed computing literature. Thus, we describe them in the context of our experiments. Makespan declares the total time taken by all infrastructure resources to complete the execution of all tasks during the experiment. Makespan depends on the infrastructure resources, the task offloading mechanism and the task generation process. Throughput declares the average number of tasks completed per second for all processing edge nodes.

The average, median and standard deviation of the execution times are the widely used statistical measurements of the execution times for the task. Average and median declare the middle values. Standard deviation declares how much the execution times of the tasks differ from the mean value. Two additional statistical measures are the skewness and kurtosis. Skewness indicates the symmetry of the time values and kurtosis indicates if the distribution of the time values is heavy-tailed or light-tailed. An additional evaluation metric is the tail latency. Tail latency is the 98th percentile and declares the smallest value of the 2% highest response times.

B. Results and Discussion

The experimental results in Table I show an improvement in the execution times using the user association model compared to a random task offloading. We see an improvement in the average execution times in both cases of constant and variable task sizes. We also see a small improvement in makespan and throughput. The improvement is small cause of the nature of task generation process meaning that the tasks are generated in a constant rate independent from the completion rate. The change in skewness means that the tail on the left side of the distribution of the execution times became longer or fatter and we have a transposition of execution times to smaller values. The SNA task offloading mechanism achieves greater values of negative kurtosis which means that the distribution of execution times becomes flatter. We expect, that for a larger network topology the benefits of the user association model will be even more evident, and thus this constitutes part of our future work.

VI. CONCLUSION

In this paper, we have presented an algorithm to study users' behaviour who offload tasks towards improving the delay performance. Particular emphasis is given on the social aspects of the communication expressed by the centrality metric (degree distribution). At the same time, we tried to address the questions that can impact task offloading performance, such as i) Which users interact with each other and ii) which access points should be used during task offloading. Specifically, we devised an algorithm to prioritize the nodes that share the most links to other nodes, expressed by the interaction intensity user association metric. These nodes are the most central and appropriate to act as an offloading point.

In the experimental evaluation of our proposed model we saw improvements with the SNA task offloading mechanism in all evaluation metrics. The reason that the improvements were small is the nature of our experiments and not the applicability of the proposed model. In our experiments the overheads of communication between different processing edge nodes were small and the rate of tasks that have communication and data dependencies in comparison with the total number of tasks was also small. This is a preliminary research work with the goal to confirm the proposed model surpass a random baseline. As a first step of our future work, we aim to test our model in a larger infrastructure, while taking into consideration energy constraints of the devices. Finally, we also aim to run additional experiments in a testbed and see the amount of improvements in the evaluation metrics with a real augmented reality application in an edge computing environment.

ACKNOWLEDGMENT

This work was supported in part by the CHIST-ERA-2018-DRUID-NET project "Edge Computing Resource Allocation for Dynamic Networks".

REFERENCES

[1] X. Ma, Y. Zhao, L. Zhang, H. Wang and L. Peng, "When mobile terminals meet the cloud: computation offloading as the bridge," in IEEE Network, vol. 27, no. 5, pp. 28-33, September-October 2013, doi: 10.1109/MNET.2013.6616112.

[2] Jianyu Wang, Jianli Pan, Flavio Esposito, Prasad Calyam, Zhicheng Yang, and Prasant Mohapatra. 2019. Edge Cloud Offloading Algorithms: Issues, Methods, and Perspectives. ACM Comput. Surv. 52, 1, Article 2 (February 2019), 23 pages. DOI:https://doi.org/10.1145/3284387

[3] L. Lin, X. Liao, H. Jin and P. Li, "Computation Offloading Toward Edge Computing," in Proceedings of the IEEE, vol. 107, no. 8, pp. 1584-1607, Aug. 2019, doi: 10.1109/JPROC.2019.2922285.

[4] U. Challita, W. Saad, and C. Bettstetter, "Cellular-connected uavs over5g: Deep reinforcement learning for interference management," arXivpreprint arXiv:1801.05500, 2018.

[5] S. Tamoor-ul-Hassan, S. Samarakoon, M. Bennis, M. Latva-aho and C. S. Hong, "Learning-Based Caching in Cloud-Aided Wireless Networks," in IEEE Communications Letters, vol. 22, no. 1, pp. 137-140, Jan. 2018, doi: 10.1109/LCOMM.2017.2759270.

[6] M. Jaber, M. A. Imran, R. Tafazolli and A. Tukmanov, "A Multiple Attribute User-Centric Backhaul Provisioning Scheme Using Distributed SON," 2016 IEEE Global Communications Conference (GLOBECOM), Washington, DC, 2016, pp. 1-6, doi: 10.1109/GLOCOM.2016.7841518.

[7] K. Son, H. Kim, Y. Yi and B. Krishnamachari, "Base Station Operation and User Association Mechanisms for Energy-Delay Tradeoffs in Green Cellular Networks," in IEEE Journal on Selected Areas in Communications, vol. 29, no. 8, pp. 1525-1536, September 2011, doi: 10.1109/JSAC.2011.110903.

[8] D. Liu et al., "User Association in 5G Networks: A Survey and an Outlook," in IEEE Communications Surveys & Tutorials, vol. 18, no. 2, pp. 1018-1044, Secondquarter 2016, doi: 10.1109/COMST.2016.2516538.

[9] M. Chen, M. Mozaffari, W. Saad, C. Yin, M. Debbah and C. S. Hong, "Caching in the Sky: Proactive Deployment of Cache-Enabled Unmanned Aerial Vehicles for Optimized Quality-of-Experience," in IEEE Journal on Selected Areas in Communications, vol. 35, no. 5, pp. 1046-1061, May 2017, doi: 10.1109/JSAC.2017.2680898.

[10] M. Chen, W. Saad and C. Yin, "Virtual Reality Over Wireless Networks: Quality-of-Service Model and Learning-Based Resource Management," in IEEE Transactions on Communications, vol. 66, no. 11, pp. 5621-5635, Nov. 2018, doi: 10.1109/TCOMM.2018.2850303.

[11] W. Wang, R. Lan, J. Gu, A. Huang, H. Shan and Z. Zhang, "Edge Caching at Base Stations With Device-to-Device Offloading," in IEEE Access, vol. 5, pp. 6399-6410, 2017, doi: 10.1109/ACCESS.2017.2679198.

[12] M. Srinivasan, V. J. Kotagi and C. S. R. Murthy, "A Q-Learning Framework for User QoE Enhanced Self-Organizing Spectrally Efficient Network Using a Novel Inter-Operator Proximal Spectrum Sharing," in IEEE Journal on Selected Areas in Communications, vol. 34, no. 11, pp. 2887-2901, Nov. 2016, doi: 10.1109/JSAC.2016.2614952.

[13] F. Pervez, M. Jaber, J. Qadir, S. Younis and M. A. Imran, "Memory-Based User-Centric Backhaul-Aware User Cell Association Scheme," in IEEE Access, vol. 6, pp. 39595-39605, 2018, doi: 10.1109/ACCESS.2018.2850752.

[14] S. M. S. Tanzil, W. Hoiles and V. Krishnamurthy, "Adaptive Scheme for Caching YouTube Content in a Cellular Network: Machine Learning Approach," in IEEE Access, vol. 5, pp. 5870-5881, 2017, doi: 10.1109/ACCESS.2017.2678990.

[15] X. Lyu, H. Tian, C. Sengul and P. Zhang, "Multiuser Joint Task Offloading and Resource Optimization in Proximate Clouds," in IEEE Transactions on Vehicular Technology, vol. 66, no. 4, pp. 3435-3447, April 2017, doi: 10.1109/TVT.2016.2593486.

[16] Y. Dai, D. Xu, S. Maharjan and Y. Zhang, "Joint Computation Offloading and User Association in Multi-Task Mobile Edge Computing," in IEEE Transactions on Vehicular Technology, vol. 67, no. 12, pp. 12313-12325, Dec. 2018, doi: 10.1109/TVT.2018.2876804.

[17] Herrlich, Horst. "A concept of nearness." General Topology and its applications 4.3 (1974): 191-212.

[18] R. Sofia, S. Firdose, L. A. Lopes, W. Moreira and P. Mendes, "NSense: A people-centric, non-intrusive opportunistic sensing tool for contextualizing nearness," 2016 IEEE 18th International Conference on e-Health Networking, Applications and Services (Healthcom), Munich, 2016, pp. 1-6, doi: 10.1109/HealthCom.2016.7749490.

[19] D. J. Watts and S. H. Strogatz, "Collective dynamics of small-world networks," nature, vol. 393, no. 6684, pp. 440–442, 1998.

[20] N. Kayastha, D. Niyato, P. Wang and E. Hossain, "Applications, Architectures, and Protocol Design Issues for Mobile Social Networks: A Survey," in Proceedings of the IEEE, vol. 99, no. 12, pp. 2130-2158, Dec. 2011, doi: 10.1109/JPROC.2011.2169033.

[21] A. Leivadeas, C. Papagianni, and S. Papavassiliou, "Socio-aware virtual network embedding", IEEE Network, vol. 26, no. 5, pp. 35-43, Oct. 2012.

[22] S. Aral and D. Walker, "Identifying influential and susceptible members of social networks," Science, vol. 337, no. 6092, pp. 337–341, 2012.

[23] Freeman, L. (1977). A Set of Measures of Centrality Based on Betweenness. Sociometry, 40(1), 35-41. doi:10.2307/3033543

[24] T. Evans and R. Lambiotte, "Line graphs, link partitions, and overlapping communities," Physical Review E, vol. 80, no. 1, p. 016105, 2009.

[25] Y.-Y. Ahn, J. P. Bagrow, and S. Lehmann, "Link communities reveal multi scale complexity in networks," Nature, vol. 466, no. 7307, pp. 761–764, 2010.

[26] Bollobás, Béla, and Andrew Thomason. "Random graphs of small order." North-Holland Mathematics Studies. Vol. 118. North-Holland, 1985. 47-97.

[27] M. Avgeris, D. Spatharakis, D. Dechouniotis, N. Kalatzis, I. Roussaki,and S. Papavassiliou, "Where there is fire there is smoke: a scalable edge computing framework for early fire detection," Sensors, vol. 19, no. 3,p. 639, 2019.

[28] M. C. Silva Filho, R. L. Oliveira, C. C. Monteiro, P. R. M. Inácio and M. M. Freire, "CloudSim Plus: A cloud computing simulation framework pursuing software engineering principles for improved modularity, extensibility and correctness," 2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), Lisbon, 2017, pp. 400-406, doi: 10.23919/INM.2017.7987304.