Optimal Task Processing and Energy Consumption Using Intelligent Offloading in Mobile Edge Computing

https://doi.org/10.3991/ijim.v16i20.34373

Sara Maftah[1(✉)], Mohamed El Ghmary[2], Hamid El Bouabidi[1], Mohamed Amnai[1], Ali Ouacha[3]
[1] Ibn Tofaïl University, Kenitra, Morocco
[2] Sidi Mohamed Ben Abdellah University, Fez, Morocco
[3] Mohammed V University, Rabat, Morocco
sara.maftah@uit.ac.ma

Abstract—The appearance of Edge Computing with the possibility to bring powerful computation servers near the mobile device is a major stepping stone towards better user experience and resource consumption optimization. Due to the Internet of Things invasion that led to the constant demand for communication and computation resources, many issues were imposed in order to deliver a seamless service within an optimized cost of time and energy, since most of the applications nowadays require real response time and rely on a limited battery resource. Therefore, Mobile Edge Computing is the new reliable paradigm in terms of communication and computation consumption by the mobile devices. Mobile Edge Computing rely on computation offloading to surpass cloud-based technologies issues and break the limitations of mobile devices such as computing, storage and battery resources. However, computation offloading is not always the optimal choice to adopt, which makes the offloading decision a crucial part in which many parameters should be taken in consideration such as delegating the heavy tasks to the appropriate machine within the network by migrating the high-resource node to an edge server and lend these capabilities to the low-resources one. In this paper, we use an Edge Computing simulator to see how network delay can impact the delivery of a certain result, we also experiment computation offloading using a two-tier with Edge Orchestration architecture, which turns out to be efficient in terms of processing time.

Keywords—Mobile Edge Computing, computation offloading, resource optimization, energy efficiency

1 Introduction and literature

Cloud computing is the first technology that enabled acquiring, on demand, a convenient set of configurable computing resources such as storage, servers, applications and services. It is used to relief the end user devices from the burden of heavy calculations. However, we are living an era where every device is connected [1], Internet of Things (IoT) invaded and as a result to their limited capabilities, the need for storage, communication and most importantly computation resources is growing [2]. According to statistics found by [3], the number of connected devices in the world will reach approximately 75 billion by 2025. Although Cloud Computing offers an increased efficiency, performance and payload, this rapid evolution imposed one more time, constraints to which we had to adopt a more sophisticated and adapted technology in terms of real-time response while consuming the least of the device's battery, especially since the proliferation of these devices had enhanced and improved our daily lives in every aspect, and it even proved its efficiency during a global humanitarian hardship like the Covid-19 pandemic [4]. Motivated by the need to deploy the 5G cellular network and the requirements to such constraints, Mobile Edge Computing (MEC) is considered to be an extension of the Cloud Computing services [5] and a key enabler to the emerging fifth generation of Mobile Networks (5G) [6] in which mobile resources can be optimized by hosting intensive computing tasks and process its large data within the radio access network (RAN) before sending it to the cloud. The main challenge consists in the growing number of heterogeneous devices, which resulted a huge increase in terms of the data produced by the users. These devices are characterized by their limited computation capacities; therefore, the traditional Cloud services will no longer satisfy the user's needs in terms of context-aware and low-latency computing and processing. Meanwhile, Edge Computing can significantly improve not only the quality of experience from an internet user point of view but with the emergence of such computing paradigm, the industrial automation evolved into an Industrial Internet of Things (IIoT) by relying on a two-tier computing architecture that comprises local and edge computing [7]. Mobile Edge computing supports the different platforms by reducing data transmission cost from devices to the cloud, which leads to relieving the pressure of network bandwidth and data centers, enhancing the response speed of user request services and improving the users' authority in terms of data privacy [8]. Therefore, computation offloading, the main feature of Mobile Edge Computing, was introduced to support the interconnection of resources-limited devices with the internet. This technique enables the device to offload part of the computation to a nearby remote server in order to help with task processing and prolong its battery lifespan as well. Processing a set of tasks can be done either by implementing an easy mode called Binary offloading, in which the entire load of tasks is processed either locally on the mobile device, or sent to a nearby Mobile Edge Computing server. Or by adopting a partial offloading mode, where the tasks are partitioned into several parts and according to their dependency and various computing requirements, some of the said tasks get to be processed locally and others are offloaded according to a certain decision-making strategy [9]. When it comes to modeling a computation offloading strategy, not only the decision that matters, but the server selection, wireless resources and channel allocation, transmission power setting and computation resources allocation, are taken in consideration in order to achieve the desired results such as latency minimization, computation and energy efficiency [10]. Many existing offloading methods were examined on a single device environment to achieve energy efficiency using algorithms based on Simulated Annealing [11] [12].

Moreover, researchers have shown interest to such promising research area and the related contributions have increased in the recent years according to [13], especially with the growing number of devices and applications that require real time response. Therefore, the need for experiment results to endorse the research advancement persists, and in order to do so, researchers rely on simulation tools to run their suggested algorithms such as the extended versions of CloudSim [14] which is a tool used to simulate Cloud Computing environments.

In this paper, we will identify the purpose of our study, which is computation offloading in Mobile Edge Computing by getting familiar with the concepts and aspects related to such computing paradigm at first, in order to take a step towards experimenting using an extended version of the simulation tool CloudSim that will endorse our work in terms of processing time optimization. After introducing the theme and its state of art, the remainder of the paper is structured as follows. Section 2 represents the evolution of Mobile Edge Computing, to elaborate afterwards the concept of computation offloading and its modeling. A computation offloading experiment is conducted in Section 3 to discuss some aspects of the obtained results. Finally, Section 5 is a conclusion of this paper that leads to future work.

2 Mobile Edge Computing

2.1 Evolution of Mobile Edge Computing

Since 2006, the traditional computing scenario relied on Cloud Computing [15], which was an efficient paradigm and an important milestone that evolved rapidly in the last few decades. Nevertheless, the basic Cloud Computing architecture could not satisfy the growing demands of IoT in terms of latency, time and energy efficiency.

The generated data each year compared to the global data center traffic is huge, and it indicates that data sources for big data are also evolving to a wide range of edge devices instead of relying on the existing Cloud Computing which is becoming unable to analyze the data generated by the massively distributed computing power. Hence, due to the growing number of connected smart devices and the massive generated data, Edge Computing is becoming more popular with its four layered architecture (IoT device, Edge, Cloud and the communication infrastructure) [16]. The key enabler to such enhancements is the distributed architecture of edge computing and its proximity to the end user [17], which allows data to be processed at the edge of the network. However, the cloud resources will still be needed for their processing and storage capacities, as well as their high availability. In other words, in order to deliver a seamless service that combines a backbone network alleviation, an agile service response and powerful cloud backup, both the cloud and edge computing paradigms are needed, considering that edge computing is actually an extension of Cloud Computing [15], [18], [19].

Mobile Edge Computing servers are placed then on the mobile network, which enabled the deployment of the 5G cellular network, and has proven to be efficient for healthcare application [20], augmented reality and compute-intensive applications [21], [22]. Mobile Edge Computing is known for its location awareness, support for mobility, fast response time, real time interaction and can be deployed in heterogeneous environments.

Prior to Mobile Edge computing, which is a technology deployed at the edge of the network, offers low latency, high scalability, elastic services, context-awareness and support for mobility, Mobile Cloud Computing offered mobility support, high latency, average scalability and elastic services based on a centralized architecture and deployed at the network core. Cloud Computing on the other hand on the early days had enabled the same benefits as Mobile Cloud Computing, using the same architecture and environment, however there was no support for mobility [16].

2.2 Computation offloading

Computation offloading is a technique where mobile devices delegate storage and computation to remote powerful entities for time and energy optimization purposes, it was first experimented through the Mobile Cloud Computing paradigm to be extended afterwards to Mobile Edge Computing [15]. Moreover, in the recent years, Artificial Intelligence technologies has enabled through optimization abilities the efficient use of computation offloading to transfer heavy computation tasks from resource limited mobile devices to edge or cloud [23].

Computation offloading is divided into two distinct modes:

- Binary offloading (Coarse-grained): This mode is easier to implement, and suitable for simple independent tasks where the processing of the computation load is either done by the mobile device itself or sent to a more powerful computation resource.
- Partial offloading (Fine-grained): In this case, the tasks are partitioned into several parts, and according to the dependency of these parts, some of them gets to be processed locally on the mobile device itself, and others are offloaded into nearby resources accordingly with a certain decision-making strategy.

Knowing that Edge Computing relies on a distributed and decentralized computing architecture, time and energy costs are divided on both the end and the edge device and the cloud as well in case the offloading system model includes the cloud part. These costs must be calculated, and studied under different offloading mechanisms to obtain the most optimized results.

In order to measure the level of user satisfaction, the task processing delay is considered to be a more critical metric. In this paper, we will be focusing on task processing delays, locally and remotely on the edge.

- Local processing: When an end user mobile device processes its computation load locally, the computational time depends on its computing resources. This metric is generally obtained using the task's data size and the processing speed of the mobile device.
- Edge Processing: Mobile devices are constrained by their computing resources, which are often not enough to process heavy tasks. Therefore, computation offloading comes in useful with the possibility to delegate the tasks to the edge, thus in this case, additionally to the processing time on the Mobile Edge Computing

server, a transmission delay is also calculated, it represents both time cost for uploading the input data and downloading the output results, the latter is rarely taken in consideration by researchers assuming that the obtained output data size is insignificant when compared to the input data size [11], [24]–[27].

2.3 Computation offloading modeling

The process of computation offloading consists on transferring and delegating computing tasks from resource limited devices to nearby powerful servers and retrieve the obtained results afterwards. This technique is considered to be promising in terms of coping with the growing demands for real time processing and low-latency responses, it is also considered to be one of the critical methods of edge computing, according to [28], computation offloading is divided into two main aspects, which are the offloading decision and resource allocation.

Offloading decision: The computation offloading decision is based on the performance maximization aspect as well as energy consumption minimization, in this regard, the possibility of offloading the task, partially or fully, is the first metric to be considered.

There are many metrics on which we can rely to make computation offloading decisions and model an optimized system, such as using bandwidth predictions. According to [29], the offloading decision is based on three predictions: the predicted time required to execute the task locally, the predicted time required to execute the computation remotely and the communication time which represents the time required to transmit the input and gather the output.

In our study, we will be comparing the local processing time of a certain task with its total cost if offloaded. Response time in case the task is offloaded into a Mobile Edge Computing server involves its processing time on the edge server as well as a transmission delay, which will be calculated using the bandwidth and the data size of the said task.

Latency, as we know, is what urged researchers and engineers to deploy this new paradigm, Mobile Edge Computing, therefore, bandwidth is an important metric when it comes to the offloading decision. However, there are many parameters to be considered to deliver user satisfaction and provide better quality of experience. For example, network condition, data usage, battery consumption and processing delay are used to determine an optimal offloading strategy, and are also used to manage different functions of Edge Computing using Deep Learning [18].

Figure 1 is an illustration of the general IoT-Edge-Cloud topology, in which wireless devices and sensors such as smartphones, drones, vehicles, healthcare applications, etc. represent the IoT layer. Meanwhile, in the Radio Access Network, where the base station is situated, we find MEC servers that provide computing and storage resources near the network edge without the need for the cloud unless these resources are no longer able to satisfy the user's demands. In this case, the cloud layer will be in charge to provide the needed computing and storage services.

Computation offloading is a feature where the IoT device can send workload to the MEC servers, or the Cloud servers, depending on how heavy are the task's requirements. The uplink input represents sending tasks to be processed either on the MEC servers, or the cloud servers, meanwhile, the downlink output is the response and the results of the processed tasks.

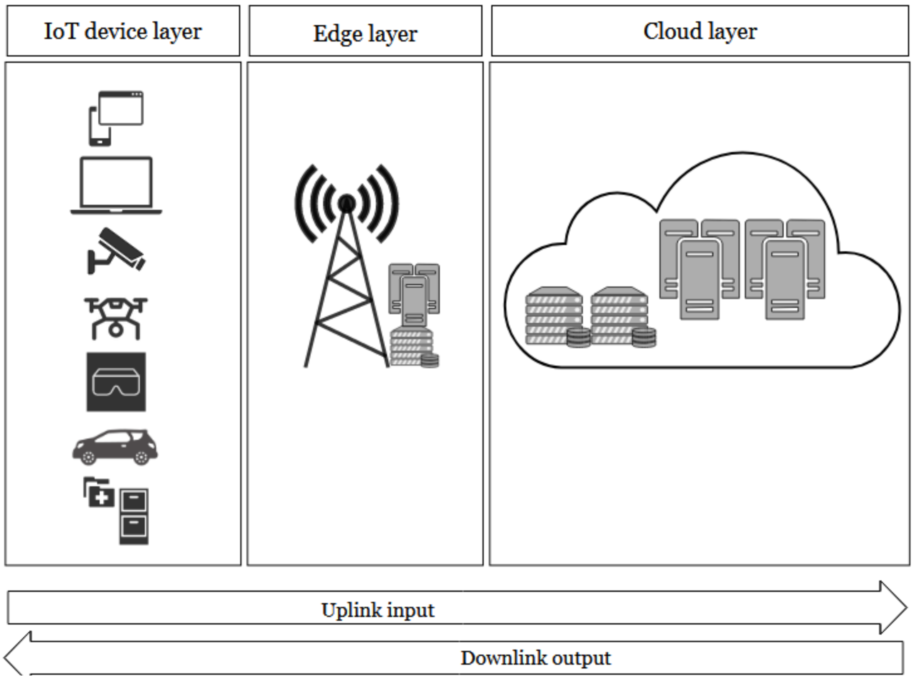

Fig. 1. IoT-Edge-Cloud topology

Task partitioning: The process of computation offloading arises the challenge of partitioning tasks efficiently [30], One of the main efficiency factors is the decision of which part of the task should be processed locally and which part should be offloaded to a remote, more powerful device. This process consists on decomposing a task into multiple sub-tasks, based on their dependency.

Network resources: The main drawback of Cloud Computing is the channel bandwidth, it restricts cloud servers from handling computing tasks since their computing power is often sufficient, but in an idle state due to slow network transmission. A thing that urged the use of Mobile Edge Computing servers to bring these resources closer to the end device and focus on enhancing the network part between the mobile device and the edge server. The authors in [31] proposed a Mobile Edge Computing framework in which network virtualization is implemented by virtualizing radio resources along with computation and storage resources. Network slicing is a form of network virtualization, this technique is one of the main features in 5G cellular network, it provides multiple logical networks to operate on a shared physical network infrastructure. In a multi-user Mobile Edge Computing model based on Time Division Multiple Access (TDMA) and Orthogonal Frequency Division Multiple Access (OFDMA), resource allocation was studied by [32] to minimize energy consumption.

Computing resource allocation: The previous computing paradigms relied on virtualization to satisfy the end user by enabling the use of computation and storage resources on demand. Mobile Edge Computing also relies on two main strategies of virtualization: Virtual machines which are better at isolating, and containers that provide easier deployment.

3 Offloading simulation using EdgeCloudSim

Many simulation tools are available allowing researchers to obtain a real-world experiment result. Modeling and simulation of Cloud Computing infrastructures and services are enabled by an extensible framework called CloudSim [14] which offers many functionalities such as modeling a large scale of Cloud Computing data centers, virtualized server hosts, application containers and energy aware computational resources, as well as a dynamic insertion of simulation inputs and personalized policies for resource allocation purposes. With the appearance of other computing paradigms due to the several constraints while using Cloud Computing, mainly Fog Computing and Edge Computing, new extensions of CloudSim such as EdgeCloudSim [33] and iFogSim [34] were deployed to provide support for mobility, orchestration and resource management so that it can be efficiently used for these computing paradigms.

An experiment was conducted using EdgeCloudSim to visualize the important network delay and the advantage of opting for a system model that relies on both local processing and computation offloading to a Mobile Edge Computing server in terms of energy consumption.

Knowing that the obtained output data size is insignificant when compared to the input data size, Figure 2 highlights the average network delay, divided into an upload and download delay, when a certain task is offloaded.

In this experiment, we used a Two-tier with Edge offloading mechanism that depends on comparing the required capacity to process a task and the available resources at that moment. We accumulate afterwards the time cost throughout the experiment, which resulted a total response time cost of 1272.5716, 544.3834 and 1055.8426 seconds adopting respectively the Mobile (local processing), Edge (remote processing) and Hybrid (partial offloading) scenario. It is obvious that response time when the task is processed in the edge server is better due to the available computing resources, meanwhile, in the hybrid scenario, the mobile device depends mainly on its resources until they are saturated, however it offloads at the same time the tasks which percentage usage was predicted to be demanding which explains the high time consumption at first (local processing and transmission delays in case of the computation offloading).

Meanwhile, Figure 3 is a graphical representation of simulating the processing of 428 independent tasks, which is done sequentially, in terms of response time using both binary and partial offloading.

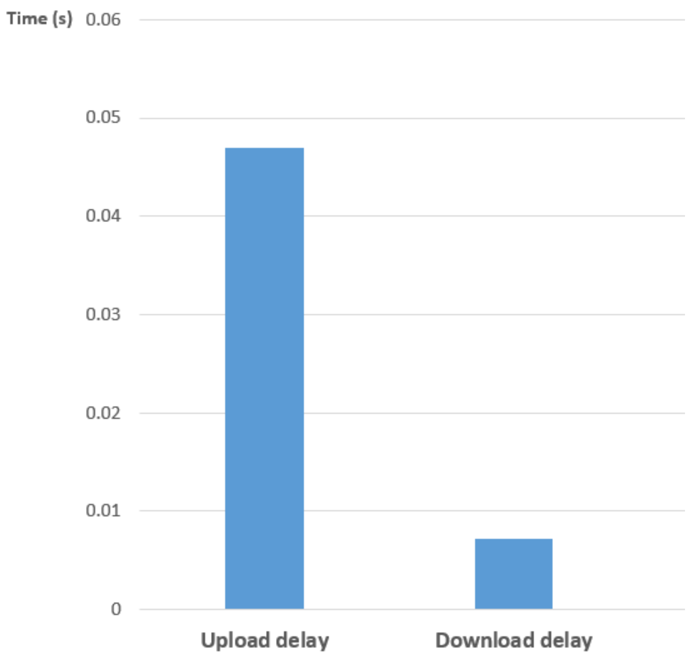

Fig. 2. Network delay while offloading a task

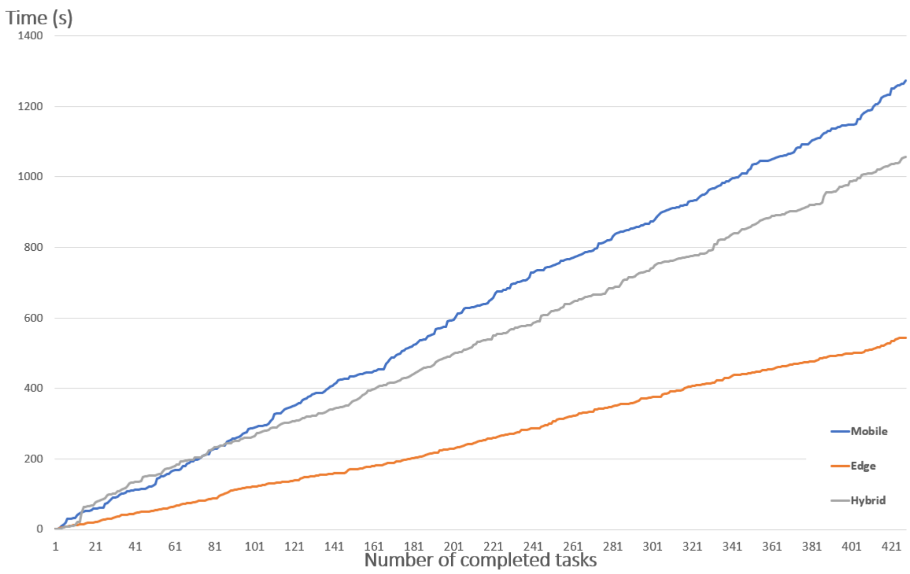

Fig. 3. Total processing time of 428 tasks using a Two-tier with Edge Orchestrator architecture

Regarding the energy consumed by the IoT device in case of a Hybrid scenario, which is the sum of:

- The energy consumed while processing the task locally.
- The energy consumed while uploading the input data and downloading the output result, which represent the transmission energy cost.
- The energy consumed while the IoT device is on idle, on a connected-standby mode to receive data.

Figure 4 is a comparison between energy cost when implementing each scenario: Local processing (Mobile), full offloading (Edge) and partial offloading (Hybrid).

As a result, the energy consumed by the mobile device was indeed reduced in case computation is offloaded.

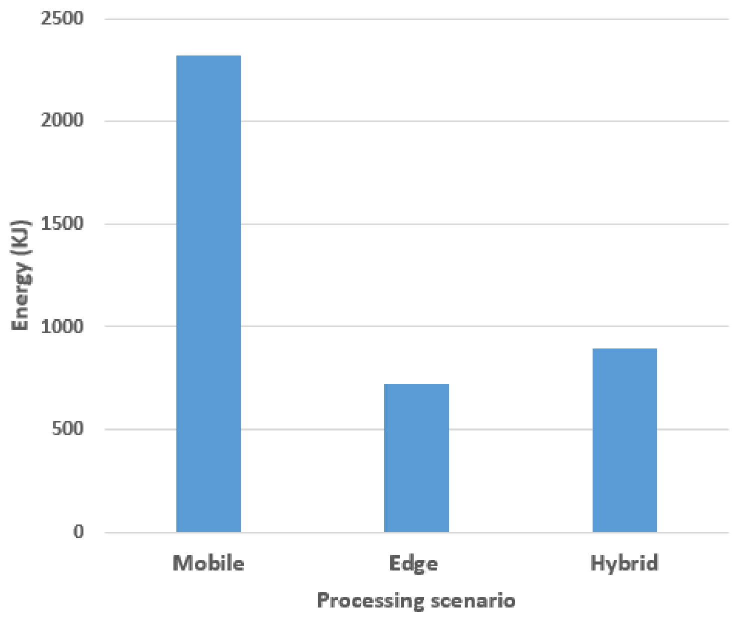

Fig. 4. Total energy consumption when processing 428 tasks

4 Conclusion and future work

Mobile Edge Computing is considered to have a great potential and a key enabler for the 5G networks, especially with the computation offloading feature. In this paper, we highlighted the aspects related to Mobile Edge Computing and the different key enablers for its successful deployment. A first experiment using EdgeCloudSim was conducted to endorse the importance of such paradigm in optimizing the task's processing time, in which the adopted computation offloading strategy depends on comparing the required capacity to process a task and the available resources at that moment.

Many challenges still remain, such as a deep study about channel and computing resources allocation, hence, this paper is an initiation that needs more research and experiments.

5 References

[1] M. Stoyanova, Y. Nikoloudakis, S. Panagiotakis, E. Pallis, and E. K. Markakis, "A survey on the internet of things (iot) forensics: challenges, approaches, and open issues," IEEE Communications Surveys & Tutorials, vol. 22, no. 2, pp. 1191–1221, 2020. https://doi.org/10.1109/COMST.2019.2962586

[2] T. Alam, "Cloud-based iot applications and their roles in smart cities," Smart Cities, vol. 4, no. 3, pp. 1196–1219, 2021. https://doi.org/10.3390/smartcities4030064

[3] T. Alam, "A reliable communication framework and its use in internet of things (iot)," CSEIT1835111— Received, vol. 10, pp. 450–456, 2018.

[4] R. P. Singh, M. Javaid, A. Haleem, and R. Suman, "Internet of things (iot) applications to fight against covid-19 pandemic," Diabetes & Metabolic Syndrome: Clinical Research & Reviews, vol. 14, no. 4, pp. 521–524, 2020. https://doi.org/10.1016/j.dsx.2020.04.041

[5] Y. Siriwardhana, P. Porambage, M. Liyanage, and M. Ylianttila, "A survey on mobile augmented reality with 5g mobile edge computing: architectures, applications, and technical aspects," IEEE Communications Surveys & Tutorials, vol. 23, no. 2, pp. 1160–1192, 2021. https://doi.org/10.1109/COMST.2021.3061981

[6] Q.-V. Pham, F. Fang, V. N. Ha, M. J. Piran, M. Le, L. B. Le, W.-J. Hwang, and Z. Ding, "A survey of multi-access edge computing in 5g and beyond: Fundamentals, technology integration, and state-of-the-art," IEEE Access, vol. 8, pp. 116 974–117 017, 2020. https://doi.org/10.1109/ACCESS.2020.3001277

[7] Y. Ma, C. Lu, B. Sinopoli, and S. Zeng, "Exploring edge computing for multitier industrial control," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 39, no. 11, pp. 3506– 3518, 2020. https://doi.org/10.1109/TCAD.2020.3012648

[8] H. Ning, Y. Li, F. Shi, and L. T. Yang, "Heterogeneous edge computing open platforms and tools for internet of things," Future Generation Computer Systems, vol. 106, pp. 67–76, 2020. https://doi.org/10.1016/j.future.2019.12.036

[9] H. Lin, S. Zeadally, Z. Chen, H. Labiod, and L. Wang, "A survey on computation offloading modeling for edge computing," Journal of Network and Computer Applications, vol. 169, p. 102781, 2020. https://doi.org/10.1016/j.jnca.2020.102781

[10] A. Shakarami, M. Ghobaei-Arani, M. Masdari, and M. Hosseinzadeh, "A survey on the computation offloading approaches in mobile edge/cloud computing environment: a stochastic-based perspective," Journal of Grid Computing, vol. 18, no. 4, pp. 639–671, 2020. https://doi.org/10.1007/s10723-020-09530-2

[11] H. Li, "Multi-task offloading and resource allocation for energy-efficiency in mobile edge computing," International Journal of Computer Techniques, vol. 5, no. 1, pp. 5–13, 2018.

[12] M. El Ghmary, Y. Hmimz, T. Chanyour, and M. O. C. Malki, "Energy and processing time efficiency for an optimal offloading in a mobile edge computing node," International Journal of Communication Networks and Information Security, vol. 12, no. 3, pp. 389–393, 2020.

[13] B. Wang, C. Wang, W. Huang, Y. Song, and X. Qin, "A survey and taxonomy on task offloading for edge-cloud computing," IEEE Access, vol. 8, pp. 186 080–186 101, 2020. https://doi.org/10.1109/ACCESS.2020.3029649

[14] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. De Rose, and R. Buyya, "Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," Software: Practice and experience, vol. 41, no. 1, pp. 23–50, 2011. https://doi.org/10.1002/spe.995
[15] M. De Donno, K. Tange, and N. Dragoni, "Foundations and evolution of modern computing paradigms: Cloud, iot, edge, and fog," Ieee Access, vol. 7, pp. 150 936–150 948, 2019. https://doi.org/10.1109/ACCESS.2019.2947652
[16] L. U. Khan, I. Yaqoob, N. H. Tran, S. A. Kazmi, T. N. Dang, and C. S. Hong, "Edge-computing-enabled smart cities: A comprehensive survey," IEEE Internet of Things Journal, vol. 7, no. 10, pp. 10 200–10 232, 2020. https://doi.org/10.1109/JIOT.2020.2987070
[17] S. K. Datta and C. Bonnet, "An edge computing architecture integrating virtual iot devices," in 2017 IEEE 6th Global Conference on Consumer Electronics (GCCE). IEEE, 2017, pp. 1–3. https://doi.org/10.1109/GCCE.2017.8229253
[18] X. Wang, Y. Han, V. C. Leung, D. Niyato, X. Yan, and X. Chen, "Convergence of edge computing and deep learning: A comprehensive survey," IEEE Communications Surveys & Tutorials, vol. 22, no. 2, pp. 869–904, 2020. https://doi.org/10.1109/COMST.2020.2970550
[19] C. Jiang, X. Cheng, H. Gao, X. Zhou, and J. Wan, "Toward computation offloading in edge computing: A survey," IEEE Access, vol. 7, pp. 131 543–131 558, 2019. https://doi.org/10.1109/ACCESS.2019.2938660
[20] Z. Ning, P. Dong, X. Wang, X. Hu, L. Guo, B. Hu, Y. Guo, T. Qiu, and R. Y. Kwok, "Mobile edge computing enabled 5g health monitoring for internet of medical things: A decentralized game theoretic approach," IEEE Journal on Selected Areas in Communications, vol. 39, no. 2, pp. 463–478, 2020. https://doi.org/10.1109/JSAC.2020.3020645
[21] T. X. Tran, A. Hajisami, P. Pandey, and D. Pompili, "Collaborative mobile edge computing in 5g networks: New paradigms, scenarios, and challenges," IEEE Communications Magazine, vol. 55, no. 4, pp. 54–61, 2017. https://doi.org/10.1109/MCOM.2017.1600863
[22] P. Cong, J. Zhou, L. Li, K. Cao, T. Wei, and K. Li, "A survey of hierarchical energy optimization for mobile edge computing: A perspective from end devices to the cloud," ACM Computing Surveys (CSUR), vol. 53, no. 2, pp. 1–44, 2020. https://doi.org/10.1145/3378935
[23] S. Deng, H. Zhao, W. Fang, J. Yin, S. Dustdar, and A. Y. Zomaya, "Edge intelligence: The confluence of edge computing and artificial intelligence," IEEE Internet of Things Journal, vol. 7, no. 8, pp. 7457– 7469, 2020. https://doi.org/10.1109/JIOT.2020.2984887
[24] M. Huang, Q. Zhai, Y. Chen, S. Feng, and F. Shu, "Multi-objective whale optimization algorithm for computation offloading optimization in mobile edge computing," Sensors, vol. 21, no. 8, p. 2628, 2021. https://doi.org/10.3390/s21082628
[25] Y. Deng, Z. Chen, X. Yao, S. Hassan, and A. M. Ibrahim, "Parallel offloading in green and sustainable mobile edge computing for delay constrained iot system," IEEE Transactions on Vehicular Technology, vol. 68, no. 12, pp. 12 202–12 214, 2019. https://doi.org/10.1109/TVT.2019.2944926
[26] M. El Ghmary, Y. Hmimz, T. Chanyour, and M. O. C. Malki, "Time and resource constrained offloading with multi-task in a mobile edge computing node." International Journal of Electrical & Computer Engineering (2088-8708), vol. 10, no. 4, 2020. https://doi.org/10.11591/ijece.v10i4.pp3757-3766
[27] T. Chanyour, Y. Hmimz, M. El Ghmary, and M. O. C. MALKI, "Delay aware and user-adaptive offloading of computation-intensive applications with per-task delay in mobile edge computing networks," International Journal of Advanced Computer Science and Ap-

plications, vol. 11, no. 1, pp. 726–733, 2020. https://doi.org/10.14569/IJACSA.2020.0110190
[28] Z. Li, X. Zhou, Y. Liu, C. Fan, and W. Wang, "A computation offloading model over collaborative cloud-edge networks with optimal transport theory," in 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom). IEEE, 2020, pp. 1006–1011. https://doi.org/10.1109/TrustCom50675.2020.00134
[29] R. Wolski, S. Gurun, C. Krintz, and D. Nurmi, "Using bandwidth data to make computation offloading decisions," in 2008 IEEE International Symposium on Parallel and Distributed Processing. IEEE, 2008, pp. 1–8. https://doi.org/10.1109/IPDPS.2008.4536215
[30] B. Varghese, N. Wang, S. Barbhuiya, P. Kilpatrick, and D. S. Nikolopoulos, "Challenges and opportunities in edge computing," in 2016 IEEE International Conference on Smart Cloud (SmartCloud). IEEE, 2016, pp. 20–26. https://doi.org/10.1109/SmartCloud.2016.18
[31] M. Liu, Y. Mao, S. Leng, and S. Mao, "Full-duplex aided user virtualization for mobile edge computing in 5g networks," IEEE Access, vol. 6, pp. 2996–3007, 2017. https://doi.org/10.1109/ACCESS.2017.2786662
[32] C. You, K. Huang, H. Chae, and B.-H. Kim, "Energy-efficient resource allocation for mobile-edge computation offloading," IEEE Transactions on Wireless Communications, vol. 16, no. 3, pp. 1397–1411, 2016. https://doi.org/10.1109/TWC.2016.2633522
[33] C. Sonmez, A. Ozgovde, and C. Ersoy, "Edgecloudsim: An environment for performance evaluation of edge computing systems," Transactions onEmerging Telecommunications Technologies, vol. 29, no. 11, p. e3493, 2018. https://doi.org/10.1002/ett.3493
[34] H. Gupta, A. Vahid Dastjerdi, S. K. Ghosh, and R. Buyya, "ifogsim: A toolkit for modeling and simulation of resource management techniques in the internet of things, edge and fog computing environments," Software: Practice and Experience, vol. 47, no. 9, pp. 1275–1296, 2017. https://doi.org/10.1002/spe.2509

6 Authors

Sara Maftah is a Phd student at the Faculty of Sciences, Ibn Tofaïl University, Kenitra, Morocco. Affiliated with the Research Laboratory in Computer Science and Telecommunications (LARIT), Team Networks and Telecommunications. She obtained a Research Master's degree in Cloud and High-Performance Computing from the National School for Computer Science and Systems Analysis (ENSIAS), Mohammed V University (UM5), Rabat, Morocco (Email: sara.maftah@uit.ac.ma).

Pr. Mohamed El Ghmary is a Professor of Computer Science at the Faculty of Science Dhar El Mahraz (FSDM), Sidi Mohamed Ben Abdellah University, Fez, Morocco. He is an associate member of the Intelligent Processing and Security of Systems (IPSS) team of Computer Science Department at the Faculty of Science, Mohamed V University (UM5), Rabat Morocco. He is also an associate member of Research Laboratory in Computer Science and Telecommunications (LARIT), Team Networks and Telecommunications, Faculty of Science, Ibn Tofail University, Kenitra, Morocco. His research interests are Mobile Edge Computing (MEC), Cloud Computing, Machine Learning, Deep Learning, Intelligent Systems and Optimization (Email: mohamed.elghmary@usmba.ac.ma).

Hamid El Bouabidi is a Phd student at the Faculty of Sciences, Ibn Tofaïl University, Kenitra, Morocco. Affiliated with the Research Laboratory in Computer Science

and Telecommunications (LARIT), Team Networks and Telecommunications. He obtained a Research Master's degree in Cloud and High-Performance Computing from the National School for Computer Science and Systems Analysis (ENSIAS), Mohammed V University (UM5), Rabat, Morocco (Email: elbouabidihamid2015@gmail.com).

Pr. Mohamed Amnai has been an Assistant at National School of Applied Sciences Khouribga, Settat University, Morocco. He joined, the Department of Computer Science and Mathematics, Faculty of Sciences Tofaïl University, Kenitra, Morocco, as an Associate Professor in 2018. He is also an associate member of Research Laboratory in Computer Science and Telecommunications (LARIT), Team Networks and Telecommunications Faculty of Science, Kenitra, Morocco. He is also an associate member of laboratory IPOSI National School of Applied Sciences, Hassan 1 University, Khouribga, Morocco (Email: mohamed.amnai@uit.ac.ma).

Pr. Ali Ouacha is a professor of computer science at the Faculty of Science of Rabat (FSR) Mohamed V University (UM5). He received his doctorate degree from the Mohammadia School of Engineering (EMI). Ouacha is a permanent member of the Intelligent Processing and Security of Systems (IPSS) team of Computer Science Department at the FSR. His research is currently focused on optimizing the performance of routing protocols in an Internet of Things (IoT), Mobile Edge Computing (MEC) and Mobile Ad-hoc Networks environment. He is the author and co-author of several publications in international journals and conferences. Ouacha is a member of the organizing committees of several scientific events (Conferences and Congresses). He is also a member of the Technical Program Committee (TPC) of several international conferences and journals (Email: a.ouacha@um5.ac.ma).

Article submitted 2022-08-01. Resubmitted 2022-09-25. Final acceptance 2022-09-25. Final version published as submitted by the authors.