Computation offloading in beyond 5G/6G networks with edge computing

Citation for published version (APA):
Stan, C., Rommel, S., Vegas Olmos, J. J., & Tafur Monroy, I. (2023). Computation offloading in beyond 5G/6G networks with edge computing: implications and challenges. In R. Mehmood, V. Alves, I. Praça, J. Wikarek, J. Parra-Domínguez, R. Loukanova, I. de Miguel, T. Pinto, R. Nunes, & M. Ricca (Eds.), *Distributed Computing and Artificial Intelligence, Special Sessions I, 20th International Conference* (pp. 473-479). (Lecture Notes in Networks and Systems, LNNS; Vol. 741). Springer. https://doi.org/10.1007/978-3-031-38318-2_47

DOI:
10.1007/978-3-031-38318-2_47

Document status and date:
Published: 01/07/2023

Document Version:
Accepted manuscript including changes made at the peer-review stage

Computation offloading in beyond 5G/6G networks with edge computing: implications and challenges*

Catalina Stan[1] [0009−0000−2072−2279], Simon Rommel[1] [0000−0001−8279−8180], Juan José Vegas Olmos[2] [0000−0002−6796−1602], and Idelfonso Tafur Monroy[1] [0000−0002−2935−7682]

[1] Department of Electrical Engineering, Eindhoven University of Technology, Eindhoven, The Netherlands
{c.i.stan, s.rommel, i.tafur.monroy}@tue.nl

[2] NVIDIA Corporation, Ofer Industrial Park, Yokneam, Israel
juanj@nvidia.com

Abstract. The emerging beyond 5G/6G networks come with novel, latency-sensitive and computation-intensive applications that require enhanced network performance and infrastructure to meet the expected quality of experience for end users. To cope with this challenge, computation offloading leverages the benefits of multi-access edge computing to migrate the application tasks requiring additional computing resources for reduced execution delay. Although the benefits of introducing offloading mechanisms into the network might be straightforward, the implementation is not trivial due to various communication and computation trade-offs that must be made to obtain optimal offloading decisions. In this paper, we provide an overview of computation offloading with highlight on the networking perspective by looking at different offloading decisions, current research efforts, as well as the challenges that may be encountered while building an efficient and robust offloading mechanism. In addition, we provide our view on the evolution of computation offloading in 6G networks to support novel applications through enriched infrastructure and powerful artificial intelligence techniques.

Keywords: Beyond 5G · 6G · edge computing · computation offloading.

1 Introduction

The integration of edge computing into the existing networks aims to bring additional computing, storage and networking resources to support the emerging 5G/6G applications. In contrast to the traditional network architectures that mainly rely on the centralized mobile cloud computing (MCC) to handle complex computational tasks, edge computing introduces an intermediate layer, i.e.,

* This work was supported by the EC H2020 MSCA ITN-ETN IoTalentum (grant no. 953442), ECSEL JU BRAINE (grant no. 876967) and EC KDT JU CLEVER (grant no. 101097560) projects.

the edge layer, into the current infrastructure where the computing capabilities are distributed due to the dynamic deployment of the edge nodes [13]. In the mobile networks context, the edge computing paradigm is often referred to as multi-access edge computing (MEC) [3], as introduced by the European Telecommunications Standards Institute (ETSI), and it will be used throughout the rest of this paper. With this promising solution, the computing capabilities are brought closer to the end users, therefore reducing the distance between the source and destination of the data. Given this context, offloading intensive processing to edge nodes through computation offloading has been employed increasingly at a time when the latency requirements coming from applications such as augmented reality/virtual reality (AR/VR), autonomous driving or interactive gaming are more demanding with each generation of mobile networks. Although computation offloading is not a new technique, with its origins dating back to the early 2000s [7], it has been studied during the MCC environment deployment and more recently in the MEC environment in order to alleviate network congestion and improve quality of experience (QoE) for end users.

This paper gives an overview of the advantages of computation offloading by comparing MCC with MEC, describes different offloading decisions and their trade-offs, discusses the main challenges that arise during an offloading procedure, as well as presenting our view on how offloading will fit into the future 6G networks. The remainder of this paper is organized as follows: section 2 presents an overview of computation offloading, section 3 provides a closer look at the different challenges faced in offloading, section 4 presents our view on offloading in 6G networks, and section 5 summarizes and concludes the paper.

2 Overview of computation offloading

In the computation offloading process, tasks coming from computation-intensive applications are offloaded to remote servers such as MEC or MCC servers for faster computing. To meet the expected QoE, which often translates to low latency in terms of network response [2], offloading processing to MEC rather than MCC has become a preferred solution due to shorter distance between the end users and edge nodes. In addition, this decentralized data processing method reduces the number of data flows transmitted in the network backhaul, therefore saving network bandwidth [7]. In the case of latency-tolerant applications, offloading computing to MCC is more appropriate, increasing the availability of resources at MEC for the latency-sensitive applications since edge resources are limited compared to cloud servers in terms of storage and computing capabilities. The aforementioned MEC and MCC aspects are summarized in Table 1 and are the basis of understanding the benefits of computation offloading to different elements of the network.

2.1 Computation offloading mechanisms

When building a computation offloading algorithm, one key aspect is to analyze the type of applications involved in the offloading process since it provides information about whether an application task can be partitioned or not and, as a

Table 1. Comparison between cloud and edge computing [9] [10].

Criteria	**Cloud computing**	**Edge computing**
Deployment	Centralized	Distributed
Configuration and planning	Sophisticated	Lightweight
Distance to end user	Large	Small
Latency	High	Low
Backhaul usage	Frequent	Infrequent
Computational capabilities	High	Limited
Storage capacity	High	Limited

consequence, if it can be included in the transmission queue for edge processing. With this in mind, the output of an offloading algorithm may result in one of the following decisions [9]:

- Local execution where application tasks are executed on the device.
- Full offloading to MEC where the entire computation is migrated at the edge.
- Partial offloading where the application is partitioned in multiple tasks that can be executed part on the device and part on the MEC host.

In addition to the above offloading decisions, tasks can be sent to the cloud for processing, but such solution may come at the expense of lower QoE. In Figure 1, the three offloading decisions are depicted in the context of 5G networks where the wireless and optical network (front-, mid- and backhaul) are the main components of the communication path when transmitting tasks to MEC/MCC.

Making optimal offloading decisions in time-varying 5G networks is not trivial, therefore various trade-offs must be made in the process. Some of these trade-offs may include the following network parameters: wireless and optical communication time, execution and queuing time at MEC/MCC, energy consumption for communication and computation, MEC load or MEC placement. Numerous efforts have been dedicated to addressing the optimal offloading problem in edge computing taking into account the implications of making these trade-offs between two or more network parameters. In the rest of this subsection, we will be looking at some of them.

In [8], computation offloading is placed in a vehicle edge computing network where tasks can be computed locally, on a fixed edge server or a vehicular edge server. The offloading decision is made using deep reinforcement learning (DRL), also including allocation for the computing and communication resources in the action space, while considering the delay of the computation task. In [16], a computation offloading scheme for dependent Internet of Things (IoT) applications is proposed, where the target is to obtain the offloading decision (local or edge). In the scheme, the decision is made with DRL by investigating the dependency between tasks, as well as the overhead, i.e., latency and energy consumption. In [17], an intelligent ultradense edge computing framework is proposed to solve the offloading decision problem together with resource allocation and service caching placement. DRL was employed to tackle the combined optimization problems,

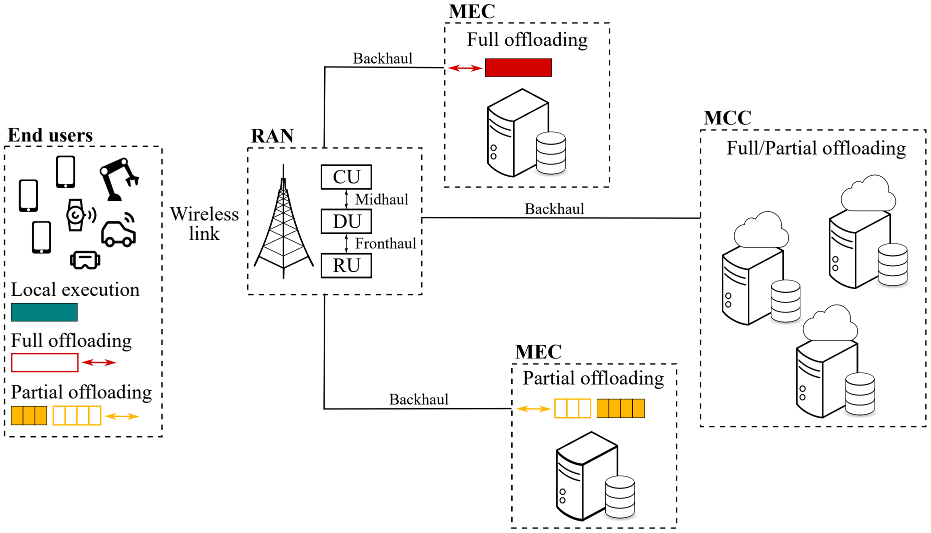

Fig. 1. Computation offloading decisions (local execution, full offloading and partial offloading) in 5G network with MEC and MCC.

while federated learning was introduced for end user privacy preservation in the model training phase. A DRL-based online offloading method is proposed in [5], where the problem is divided into two sub-problems: offloading decision solved with DRL and resource allocation in terms of time required for energy harvesting and offloading, solved with one-dimensional bi-section search. Model-free DRL is used in [14] to solve the offloading challenge by considering non-divisible and delay-sensitive tasks as well as the load level dynamics of the edge nodes. The decision to offload a task and the selection of which edge node to offload it to was indicated in the DRL action space. A multilevel vehicular edge-cloud computing network is considered in [6] where both computation offloading and resource allocation are included in the optimization problem. Reinforcement learning was employed to solve the problem by minimizing the vehicle's consumption in terms of communication and computation time, as well as energy consumption. Similar network parameters were used in [4], where the aim was to maximize the long-term system utility in an edge computing assisted power IoT scenario.

3 Challenges in computation offloading

Offloading comes with a wide set of challenges that may be considered while building a computation offloading framework. Examples of such challenges are described as follows:

- The type of offloading decision, presented in Section 2, is the center of the algorithm and it is usually the first problem to be addressed.

- In partial offloading, the dependability between application tasks is difficult to approach and implement due to the execution order (one task may require the output of another task) and software/hardware restrictions (some tasks may require local execution, such as the camera input) [10].
- Definition of the network model and input parameters – it refers to the choices regarding the offloading environment (e.g., single or multi-user, static or mobile users, single or multi-MEC, MEC placement) and its parameters: communication (e.g., transmit power, available bandwidth) and computation capabilities (e.g., number of CPUs).
- Resource allocation for both communication (e.g., radio resources) and computation (e.g., number of cores per task) to achieve the desired offloading trade-off in terms of execution delay, transmission delay, energy consumption, etc.
- Management of user mobility as well as edge mobility (e.g., when deploying vehicle edge computing, unmanned aerial vehicles (UAVs)) since network parameters are time-varying and service continuity may be difficult to guarantee [9].

The examples above are proof that making optimal offloading decisions, especially in dynamic and dense 5G networks, is challenging, therefore employing advanced artificial intelligence (AI) techniques is a natural step to solve this problem, as seen lately in the literature [12].

4 6G vision for computation offloading

The range of applications is expanding with each generation of mobile networks, each application requiring its specific network infrastructure model, techniques and minimum performance indicators in order to provide the expected QoE. Offloading to edge has been one of the techniques used for improving the performance of latency-sensitive 5G applications such as AR/VR, face recognition, gaming, video analytics that require enhanced computing capabilities. In future 6G networks more applications are expected to emerge, requiring AI on both application and networking sides of the deployment. Therefore, AI-enabled computation offloading is expected to become one of the key drivers in providing real-time user experience for future applications including telepresence (mixed reality co-design, merged reality game/work), AI-assisted vehicle-to-everything (V2X), to name but a few [15]. With the increasing number of end devices, far edge computing can become more relevant in future 6G networks since the resources are placed even closer to the end users [1], creating a computing continuum spanning across the network. As a consequence, the latency can be reduced even more and the interaction and data sharing with external parties is avoided for privacy reasons, but coming at the expense of complex offloading mechanisms due to the increased number of computing nodes and offloading possibilities. In addition, 6G architectures are set to employ distributed AI mechanisms such as federated learning to preserve user privacy and reduce security risks by decoupling the model training from the need to access the raw training data generated and stored on end user devices [11].

5 Conclusion

This paper presented an overview of computation offloading starting with a comparison between MEC and MCC and highlighting the benefits of these computing resources. Then, different offloading decisions were introduced, followed by a short description of current computation offloading works and the challenges faced when building an offloading framework. Finally, we described our vision on the evolution of computation offloading in future 6G networks by looking at novel applications, enhanced computing resources through far edge, and federated learning as an AI training model solution for user data protection.

References

1. 5GPPP Architecture Working Group: The 6G Architecture Landscape (February 2023). https://doi.org/10.5281/zenodo.7313232
2. Bhattacharya, A., De, P.: A survey of adaptation techniques in computation offloading. J. Netw. Comput. Appl. **78**, 97–115 (2017). https://doi.org/https://doi.org/10.1016/j.jnca.2016.10.023
3. ETSI GS MEC 003: Multi-access Edge Computing (MEC); Framework and Reference Architecture (March 2022), v3.1.1
4. Hu, J., Li, Y., Zhao, G., Xu, B., Ni, Y., Zhao, H.: Deep Reinforcement Learning for Task Offloading in Edge Computing Assisted Power IoT. IEEE Access **9**, 93892–93901 (2021). https://doi.org/10.1109/ACCESS.2021.3092381
5. Huang, L., Bi, S., Zhang, Y.J.A.: Deep Reinforcement Learning for Online Computation Offloading in Wireless Powered Mobile-Edge Computing Networks. IEEE Trans. Mob. Comput. **19**(11), 2581–2593 (2020). https://doi.org/10.1109/TMC.2019.2928811
6. Khayyat, M., Elgendy, I.A., Muthanna, A., Alshahrani, A.S., Alharbi, S., Koucheryavy, A.: Advanced Deep Learning-Based Computational Offloading for Multilevel Vehicular Edge-Cloud Computing Networks. IEEE Access **8**, 137052–137062 (2020). https://doi.org/10.1109/ACCESS.2020.3011705
7. Lin, L., Liao, X., Jin, H., Li, P.: Computation Offloading Toward Edge Computing. Proceedings of the IEEE **107**(8), 1584–1607 (2019). https://doi.org/10.1109/JPROC.2019.2922285
8. Liu, Y., Yu, H., Xie, S., Zhang, Y.: Deep Reinforcement Learning for Offloading and Resource Allocation in Vehicle Edge Computing and Networks. IEEE Trans. Veh. Technol. **68**(11), 11158–11168 (2019). https://doi.org/10.1109/TVT.2019.2935450
9. Mach, P., Becvar, Z.: Mobile Edge Computing: A Survey on Architecture and Computation Offloading. IEEE Commun. Surv. Tutor. **19**(3), 1628–1656 (2017). https://doi.org/10.1109/COMST.2017.2682318
10. Mao, Y., You, C., Zhang, J., Huang, K., Letaief, K.B.: A Survey on Mobile Edge Computing: The Communication Perspective. IEEE Commun. Surv. Tutor. **19**(4), 2322–2358 (2017). https://doi.org/10.1109/COMST.2017.2745201
11. McMahan, H.B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data (2023)

12. Shakarami, A., Ghobaei-Arani, M., Shahidinejad, A.: A survey on the computation offloading approaches in mobile edge computing: A machine learning-based perspective. Computer Networks **182**, 107496 (2020). https://doi.org/https://doi.org/10.1016/j.comnet.2020.107496
13. Shi, W., Cao, J., Zhang, Q., Li, Y., Xu, L.: Edge Computing: Vision and Challenges. IEEE Internet Things J. **3**(5), 637–646 (2016). https://doi.org/10.1109/JIOT.2016.2579198
14. Tang, M., Wong, V.W.: Deep Reinforcement Learning for Task Offloading in Mobile Edge Computing Systems. IEEE Trans. Mob. Comput. **21**(6), 1985–1997 (2022). https://doi.org/10.1109/TMC.2020.3036871
15. The 5G Infrastructure Association: European Vision for the 6G Network Ecosystem (June 2021). https://doi.org/10.5281/zenodo.5007671
16. Xiao, H., Xu, C., Ma, Y., Yang, S., Zhong, L., Muntean, G.M.: Edge Intelligence: A Computational Task Offloading Scheme for Dependent IoT Application. IEEE Trans. Wirel. Commun. **21**(9), 7222–7237 (2022). https://doi.org/10.1109/TWC.2022.3156905
17. Yu, S., Chen, X., Zhou, Z., Gong, X., Wu, D.: When Deep Reinforcement Learning Meets Federated Learning: Intelligent Multitimescale Resource Management for Multiaccess Edge Computing in 5G Ultradense Network. IEEE Internet Things J. **8**(4), 2238–2251 (2021). https://doi.org/10.1109/JIOT.2020.3026589