# Journal of Technology Innovation and Society

E-ISSN: Pending
P-ISSN: Pending

## Publisher Information

**Publisher**:

**Institute of Advanced Technology and Green Innovation LLC (INATGI)**

**Location**: 2807 N Parham Rd, Ste 320, Henrico, VA 23294, United States

**Official Websites**:

- www.inatgi.com
- www.inatgi.in

**Publishing Frequency**:

Biannual (Two issues per year)

**Contact Details**:

- Phone: +1 (757) 407-6474
- Email: editorial@inatgi.com/editorial@inatgi.in

**Publishing Statement**

The ***Journal of Technology Innovation and Society*** (***JTIS***) is published by the Institute of Advanced Technology and Green Innovation LLC (INATGI) as an open access academic journal. All content is freely available without charge to the user. Users are allowed to read, download, copy, distribute, print, search, or link to the full texts of the articles in this journal without asking prior permission from the publisher or the author, in accordance with the principles of the Budapest Open Access Initiative (BOAI).

**DOI Registration**

All published articles are assigned to a **Digital Object Identifier** (**DOI**) under the prefix **10.63646**, ensuring permanent and citable access.

**Legal Statement**

# Journal of Technology Innovation and Society

Volume 2 Issue 1
July 2024
Virginia, USA

Institute of Advanced Technology and Green Innovation

E-ISSN: Pending
P-ISSN: Pending

**Table of Contents**

# 5G MEC Offloading: Two QoE-based Strategies

Mingyu Zhang[1], Yixin Zhang[2, *]

[1]The Cybersecurity Center, Florida Memorial University, Miami, US 33054
[2]The Fu Foundation School of Engineering and Applied Science, Columbia University, New York City, US 10027
[*]Email: Yixin.z@columbia.edu

## Abstract

5G MEC provides computing resources as services by 5G communication technologies. But, compared with conventional centralized cloud, MEC is a distributed resource with dynamic changes and conflicts between UEs (user equipment) and MEC servers. How to allocate MEC resources to meet users' requirements and to maintain QoE is an important issue in IT and across business disciplines. Offloading is an appropriate way to distribute MEC resources to achieve high levels of utilization and effectiveness. This study proposes two mechanisms that can work under complex conditions, such as multiple UEs' offloading requests, low latency, low energy consumption, and task consistency.

**Keywords:** Mobile edge computing (MEC), offloading, 5G MEC, MEC deployment scheme, resource allocation, Quality of Experience (QoE)

**Article History**:
Received December 27, 2023
Revised March 02, 2024
Accepted March 31, 2024
Available Online April 12, 2024


See: https://inatgi.in/index.php/jtis/index for more information. https://doi.org/10.63646/WRUC9372

# 5G MEC Offloading: Two QoE-based Strategies

## 1. Introduction

The development of mobile communication technologies and the adaptation of intelligent devices, various network services and applications are quickly emerging, and end-users are increasingly demanding high network performance, such as broad bandwidth, low latency, availability, reliability, and security. Although the processing power of the CPU (the central processing unit) of the new mobile device continues to strengthen, it still cannot process requests based on massive data in a limited time. In addition, the huge consumption of battery impedes the processing of applications by local servers. These issues affect the performance and the QoE (quality of experience) of the services on the UEs (user equipment). 5G MEC offloading has the potential to solve the problems and to provide high-performance services to end users (Mao, et al., 2017a; Guo, Liu & Zhang, 2018; Mach & Becvar, 2017).
MEC (Mobile edge computing) refers to the deployment of computing and storage resources at the edge of the mobile network, which provide IT services and cloud computing capabilities for mobile networks, and which provide end-users with low (no) latency and with high-performance service solutions. MEC is a critical factor improving the QoE of 5G network. As one of the key technologies in the MEC, task offloading refers to the transmission of part or all of the tasks of a UE with limited computing power to a cloud server through a network. Offloading allows the UE to extend functionality to the cloud and leverages the cloud's powerful computing power to expand its computing capacity, to decrease execution latency, to extend battery life, and to provide a high QoE for end users. The offloading technology consists of three aspects: a decision on computation offloading, the allocation of computing resources, and system implementation. In particular, a decision on computation offloading solves the problem of what resources the UE can offload, how to offload tasks, and how many resources need to be offloaded. Allocation of computing resources focuses on how to appropriately allocate resources that can be offloaded among entities (cloud, MEC, and UEs). System implementation deals with the implementation strategy for achieving offloading resources on the MEC platform (Abbas, et al., 2018; Mao, et al., 2017b; Pan & McElhannon, 2018).
5G MEC offers the advantages of scalability, flexibility, mobility, virtualization, low cost and no terminal limitations, and is widely used in data sensing, medical and health industries, social networking, multimedia searches, and many other fields. It significantly improves computing processing capability and the service quality in various fields, showing good development prospects and promising social benefits. The MEC offloading can transmit files to the MEC server that is closest to the UE. It not only reduces the network workload, but it solves the problems of energy consumption and transmission costs. Offloading also helps in the development of emerging technologies with zero latency. For example, autonomous vehicles need to sense road conditions, obstacles, and the driving information of surrounding vehicles in real time. Faster transmission and more accurate analysis of the related massive data and calculation can be realized through 5G MEC offloading (Wang, C. et al., 2017a; Liu, et al., 2016; Tao, et al., 2017).
The paper is outlined as follows. Section II depicts an overview of current research on MEC offloading. The two offloading mechanisms are described and explained in Section III. Section IV addresses research challenges and future trends toward 5G MEC offloading. The conclusion is in Section V.

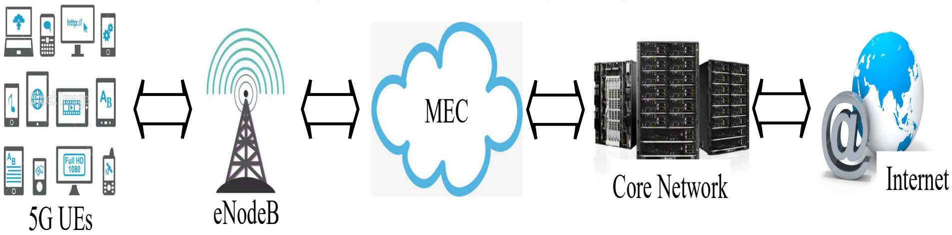
## 2. Current Research

The UE offloads partial of the computing task to the mobile edge cloud server deployed by the network operator near the base station by communicating with the nearby deployed base station (eNodeB). High-speed transmission of fiber enables rapid cross-response and low-latency connections. The MEC provides the UE with a wide range of catching, calculating and services.

 https://doi.org/10.63646/WRUC9372

Upon the deployment of 5G MEC, it can provide session and business continuity, QoE and pricing, and support for MEC local networks. The MEC is also flexible in the specific deployment modes of the 5G network: centralization and distribution. Centralized deployment supports enhanced gateway capabilities with UE, and the distributed deployment allocates services in different locations. This hierarchical placement of resources in the network makes network management more flexible and dynamic (Muhammad, et al., 2018; Shi, et al., 2016; Zheng, et al., 2016). The following figure (Figure 1) illustrates a complete network of 5G MEC that includes 5G UEs (end users), eNodeB (the edge base station), the MEC server, the core network, and the Internet. eNodeB is the connection bridge between the UEs and the MEC servers. The edge computing server is deployed in the wireless access network, which greatly reduces the distance from the UE. Due to the reduced transmission distance, the task migration of the 5G MEC no longer needs to go through the long backhaul link and the core network, thereby reducing the delay overhead. On the other hand, since the computing capability of the edge server is greater than that of the mobile device, the task processing time is greatly shortened.

Figure 1. 5G MEC Typology



In an offloading process, the UE typically consists of a code parser, a system parser, and a decision engine. The execution of an offloading decision is divided into three steps. First, the code parser determines what can be offloaded, and the specific task offloading depends on the application type and the code and data partitioning. Then, the system parser is responsible for monitoring various parameters, such as the available bandwidth, the size of the offloading file, transmission costs, and energy consumption. Finally, the decision engine decides whether to offload, based on the evaluations. The offloading decision is mainly divided into three categories: the low latency offloading mechanism, the energy efficiency offloading mechanism, and the balancing latency and consumption offloading mechanism. For example, shortening the latency of offloading process, decreasing the energy consumption and costs, and mitigating the execution failure of offloading process. Game theory algorithms and the MDP (Markov Decision Process) are popular models used to estimate potential offloading mechanisms (Zhou, et al., 2019; Chen, et al., 2013).

The decision to migrate tasks is the result of many factors, such as end-user, networks, mobile devices, servers and applications. End-users need to consider offloading costs and QoE. Network mainly focuses on Wi-Fi and 5G. Processing capability, memory and storage are factors that affect mobile devices. Whether the edge server has enough resources to fulfill the offloading tasks and the environment in which the application is running appropriately directly affects the decision of offloading. For the application itself, the higher the computational complexity of the application and the smaller the amount of data transferred, the greater the likelihood of offloading of the tasks.

Among the extant offloading strategies, some are energy-oriented decision methods, some are corresponding time and energy decision methods, some are overall mobile terminals as migration targets, and some are converting mobile terminal applications into multiple partitions that are the smallest unit of offloading.

### 2.1 Low Latency Offloading Mechanism

If a task is executed locally, the elapsed time is the time during which the application performs the task. If the task is offloaded to the MEC, the time spent will include three parts: the time at which the file to be offloaded is transferred to the MEC, the time of the task required to receive file from the MEC, and the time of getting the file back from the MEC to the UE. Therefore, the delay caused by offloading the computing tasks to the MEC affects the QoE. Although a model of considering multiple UEs leads to NP-hardness, a low latency offloading strategy is still a practical approach to the allocation of resources in the MEC to multiple UEs. In order to ensure QoE, many studies, aimed at reducing delays, involve different optimization algorithms and application scenarios. Examples include constructing a model of NP-hardness to resolve offloading for multiple UEs, or the Lyapunov optimal dynamic offloading mechanism associated with energy harvesting technologies. Low latency is a key factor that affects offloading energy consumption and user's QoE. Service with low latency is more attractive to end-users, although they may pay extra for the energy consumption and other costs (Kwak, et al., 2015; Mao, Zhang & Letaief, 2016; Kao, et al., 2017; Wang, C. et al., 2017b).

### 2.2Energy Efficiency Offloading Mechanism

The energy consumed for offload tasks to the MEC server consists of two aspects. One is the transmission energy that transfers the offloading file to the MEC, and the other is the energy consumed by the data returned to the UEs. One study used the TDMA (time-division multiple access) system to divide time slots. In each time slot, the UE offloads its file to the MEC based on channel quality and local energy consumption. The optimal offloading strategy was proposed for each UE. If the UE has a priority above a given threshold, the UE completely offloads the computing task to the MEC. Conversely, if the UE has a lower priority than the threshold, only some of the tasks are offloaded to satisfy the delay constraint. For UEs that do not meet the application latency constraints, tasks will be implemented locally. Furthermore, the TDMA mechanism has been extended to the OFDMA-based offloading scheme (orthogonal frequency-division multiple access); this can reduce energy consumption by 90% (You & Huang, 2016; You, et al., 2016). Another study (Zhang, et al., 2016) attempted to minimize offloading transmission and radio energy consumption by optimization algorithm with latency constraints. Offloading in small-cell network is also worth of examining from task and transmission perspectives (Yang, et al., 2018; Munoz, P-Iserte & Vidal, 2015). Cloud computing is scalable and adaptable, it is the potential to extend the mobile edge computing offloading mechanisms to other cloud-related systems, such as multi-access edge computing and cloud of things (El Haber, Nguyen & Assi, 2019; Nan, et al., 2017).

### 2.3 Balancing Offloading Mechanism

When performing complex computing tasks, such as face recognition systems or real-time video systems, both energy consumption and delay can affect the QoE; focusing only on one point cannot achieve UEs' requirements. So, knowing how to achieve a leverage between energy consumption and latency when performing offloading tasks is crucial to build a balanced strategy to more efficiently allocate resources in a virtuous cycle.

The following parameters are considered in the offloading process as a trade-off analysis: the total amount of data to be processed, the computing power of the UE and the MEC, the channel state between the UE and the SCeNB (the intermediate base station connecting the UE and the MEC), the energy consumption of the UE, waiting time, and executing time, and other related costs (Kao, et al., 2017; You & Huang, 2016). Studies seek to construct the offloading algorithms through different combinations to optimize offloading processes. For example, You, et al. (2016) used Gurobi optimizer to analyze the offloading problem in the static environment and extend the condition to dynamic phenomenon.

## 3. 5G MEC Offloading Mechanisms

Due to the diversity of offloading factors and metrics, it is not easy to design an appropriate offloading strategy and objective assessment of offloading performance. In this section, two offloading strategies will be introduced and generalized: optimal offloading and the consistency offloading mechanisms. The optimal strategy analyzes various costs and consumption of the complete offloading process and seeks the optimal mechanism for each of the UEs to accomplish overall high utilization. The consistency strategy solves the issue of VM migration from one MEC to another, in order to maintain both task consistency and a high QoE with low latency. Both of the mechanisms are capable of handling malicious requests from multiple UEs (Ndikumana, et al., 2017).

### 3.1 Optimal Offloading Mechanism

The first approach is an optimal offloading mechanism that adopts game theory to allocate limited edge resources to multiple UEs in order to benefit both end-users for maximum benefit and providers for the highest resource utilization. The route between the UEs and the edge servers is random. This leads to different transmission energy consumption, channel interference, and overall costs. If the edge center near the base station receives too many requests and uses the wireless channel to schedule resources, it can cause severe mutual interference. Hence, it is important to examine whether a request needs to be offloaded and how to appropriately allocate resources.

Model

Suppose there are N (= {1, 2, …, N}) UEs. Each one has a task that requires computing resources and can be selectively sent to close base stations (eNodeBs). eNodeB connected to the edge server can process the UE's request. The provider determines the communication speed between the UE and the eNodeB, such as file transmission channel, the uplink transmission rate, and the downlink transmission rate. There are M (= {1, 2, …, M}) wireless channels between the UEs and the eNodeBs.$\mathrm{O_n}$ is the offloading strategy of user n. $\mathrm{O_n} = 0$ means that the UE will run the task locally, $\mathrm{O_n} > 0$ means that the UE will offload the task to the edge. $\mathrm{O_{1-N}}$ (= $(\mathrm{O_1}, \mathrm{O_2}, \dots, \mathrm{O_N})$) represents the offloading strategies of all UEs.

If the total cost of MEC is less than or equal to the total cost of local computing, it is defined as effective cloud calculation, as in the following equation

$$C_n^E(O_{1-N}) \leq C_n^L \tag{1-1}$$

$\mathrm{C_n^L}$ represents the total cost of local computing, and $\mathrm{C_n^E(O_{1-N})}$ represents the total cost of MEC. For UEs, the optimal approach assists the UE in rationally selecting an offloading strategy with a high QoE. For providers, the optimal approach achieves higher resource utilization and revenues. The above function can be described as an optimization problem. It is used to calculate the maximum number of UEs that can obtain MEC resources among all of the UEs. The mathematical expressions are shown in equation (1-2). $\mathrm{R_O}$ is 1 when A is true, and 0 otherwise.

$$\begin{gathered} max \sum_{n \in N} R_{\{O_n > 0\}} \\ C_n^E(O_{1-N}) \leq C_n^L \\ R_O \in \{0, 1, \dots, \mathrm{M}\} \end{gathered} \tag{1-2}$$

This is an NP-Hardness issue, similar to the biggest packaging problem. Assuming another complete system cost is $\mathrm{I_n}$, the target problem is transformed into a combinatorial optimization problem in a multidimensional discrete space environment, i.e., minimizing $\mathrm{I_n}$. As shown in equation (1-3)

$$\begin{gathered} min \sum_{n \in N} I_n \\ R_O \in \{0, 1, \dots, \mathrm{M}\} \end{gathered} \tag{1-3}$$

The general expression is:

$$\Gamma_{MEC} = (\mathrm{N}, \{O_n\}_{n \in N}, \{I_n\}_{n \in N}) \tag{1-4}$$

On the condition of

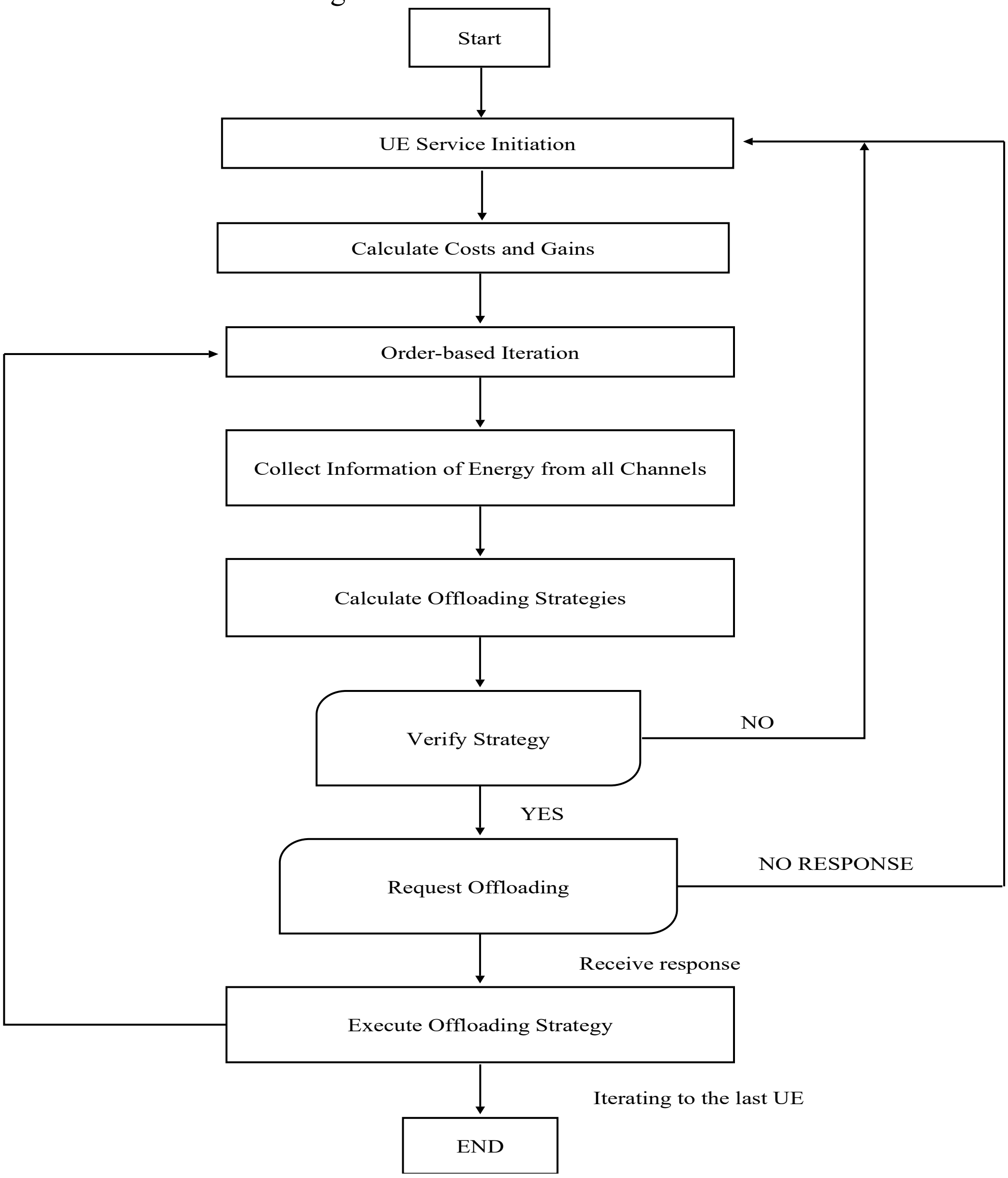
$$I_n(O_n^*, O_{N-n}^*) \leq I_n(O_n, O_{N-n}^*) \tag{1-5}$$

where $O^*$ (= $(O_1^*, O_2^*, \ldots, O_N^*)$) is the set of all potential UEs' offloading strategies when MEC offloading is under the Nash Equilibrium.

Offloading Process

This section will depict the complete iterative processes in detail (Figure 2). In total, the process consists of ten steps, from beginning to end. The iteration will be terminated until all of the requested tasks are examined and assigned for potential optimal offloading strategies.

Figure 2. The Iteration Process

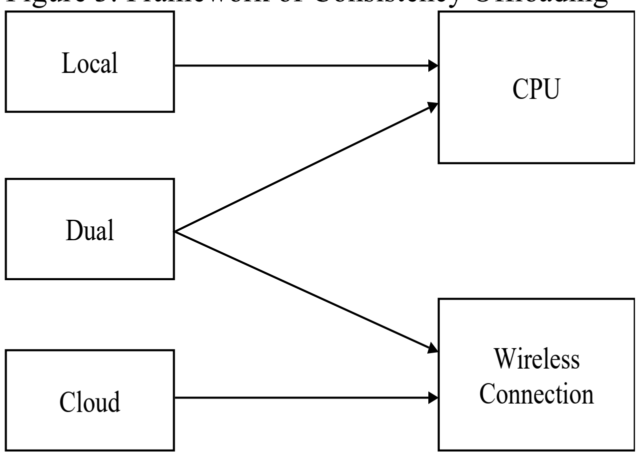


First is the initiation of every UE locally.
Second is the calculation of channel gain and the transmission of each UE based on the random distance between the UEs and the BS.
Third is the ranking of UE based on its power; the high channel power that the UE preferentially selects in the iteration process.
Fourth is the transferring of energy on selected channels and the collecting of energy information from all channels.
Fifth is the calculation of all of the potential offloading strategies and the examination of the related costs and consumption.
Sixth is verifying the strategy. If an offloading strategy is better than the local implementation, continue to the next step; otherwise, it will go back to the initiation for the next UE's decision making.
Seventh, the UE sends an offloading request to the related servers.
Eighth, if the UE receives a response from the servers, it will implement an offloading strategy.
Ninth, if the UE doesn't receive a response from servers, the local implementation will continue. The iteration will go back to the initiation for the next UE.
Tenth, the complete iteration will proceed from the first UE until the last, and then it will end.
This approach is effective if all of the users are in a static environment; in that case, the UEs won't change the offloading decisions during the whole process. However, in practice, 5G users move frequently, and tasks relate to multiple MEC servers. In this circumstance, it is necessary to consider the VM migration, in order to maintain the continuity and the consistency of services.

### 3.2 Task Consistency Offloading Mechanism

By processing offloading technology, VM (virtual machine) migration can ensure the continuity and consistency of services. VM migration means that VMs running on the current node will migrate to another more suitable node(s) with higher utilization and lower costs (Wang, et al., 2015). Much research has focused on the offloading strategies from one node to another node. But, in practice, this phenomenon often occurs: VMs need to migrate to multiple nodes in order to achieve the optimal performance of QoE. Thus, the conditions of consistency offloading are complicated and dynamic. In this study, we present a generalized model, which can assist practitioners or scholars to implement offloading mechanisms through various perspectives (Kwak, et al., 2015).

Figure 3. Framework of Consistency Offloading



In 5G MEC offloading, tasks can be divided into three categories: 1) tasks that are executed locally in UE, 2) tasks that are executed in the cloud, and 3) tasks that can be executed either in local UE or in the cloud (Dual). Since it aims at optimizing resource utilization with the tolerated delay, this solution is designed to determine whether the VMs (virtual machine) need to migrate, and how to allocate limited resources after VM migration. It uses MDP (the Markov Decision Process) to evaluate which way is better to maintain task consistency and to guarantee QoE with potential low costs (Wang, et al., 2016; Ding, Fan & Poor, 2019;). According to Figure 3, three factors that directly impact the final offloading strategy: processing costs, processing channel, and processing order.

<u>Processing Costs</u>

Low latency applications, such as IoT (Internet of Things) and the connected terminals, require high reliability and low end-to-end delay (millisecond) communications. To support low latency, virtual machines and data source are offloaded as users move from one MEC to another. The offloading process may have a negative impact on latency. Hence, it is possible to consider a high-speed path with a small delay in the backhaul link, meanwhile, the transmitted file needs to be compressed, and the virtual machine recovery process needs to be simplified. In general, transmission, energy, waiting, and executing costs are the major factors that impact on the overall utilization in 5G MEC offloading (Min, et al., 2019; Xu, He &Li, 2014; Lu & Xu, 2018; Lyu, et al., 2018; Kherraf, et al., 2019; Zhang, et al., 2017).

<u>Processing Channel</u>

In 5G MEC offloading, there are two major channels. If tasks are implemented locally, the processing channel is the CPU. If tasks are implemented in the MEC, the wireless transmission channel is the cellular network or the WLAN (wireless local area network). In addition, some tasks may be allocated to a combined channel, such as CPU and WLAN, CPU and cellular, and cellular and WLAN. Based on the MDP, the particular channel with minimum costs and optimal utilization of the resources will be the offloading strategy (Mach & Becvar, 2017; Mao, et al., 2017b).

<u>Processing Order</u>

In the 5G MEC offloading process, there exist tasks with malicious requirements that come from different UEs. How to effectively fulfill tasks in the MEC, by certain orders, that impact on the overall performance of services and resource utilization is the question under consideration. For sequential tasks, the optimal offloading strategy is suitable for processing tasks. For concurrent tasks, load balancing heuristics are used to offload tasks to the MEC. For the partial offloading model, the influence of inter-task dependencies is proposed, and the polynomial time algorithm is used to solve the optimal solution of offloading (Guo, Liu & Zhang, 2018; Kao, et al., 2017; Ding, Fan & Poor, 2019).
The following model is a general MDP that examines the minimum costs and optimal utilization. The goal is to minimize the total costs of a specific task:

$$min \sum \gamma^t C \tag{2-1}$$

where $\gamma^t$ $(0 \leq \gamma < 1)$ is the discount, and C represents all possible costs during the offloading process. For example, transmission costs, energy consumption, waiting time, service time, etc. As MDP, we have two main algorithms: policy ($\pi(s)$) and value ($V(s)$).

$$\pi(s) = arg \min_{a} \{ \sum_{s'} P(s' \mid s, a)[C(s' \mid s, a) + \gamma V(s')]\} \tag{2-2}$$

$$V(s) = \sum_{s'} P_{\pi(s)}(s, s')[C_{\pi(s)}(s, s') + \gamma V_{s'}] \tag{2-3}$$

There are the state transition P, cost function C, and the probability ($P_{\pi(s)}(s, s')$) that action $\pi(s)$ in state s will lead to state $s'$ for an MDP. We seek the policy ($\pi(s)$) that minimizes the discounted costs. Specifically, a represents an action, and s and $s'$ represent certain states. The value $V(s)$ is the actual value of action a in state s. During the iterative process, these two steps are repeated until there is no chance to calculate the strategy with the minimum cost.

## 4. Challenges and Future Trends

In 5G networks, according to business needs, MEC can be flexibly layered to maximize resources utilization and to reduce the computational costs and energy consumption with low latency. However, the offloading of 5G MEC still faces problems, such as mobility management, security and interference control.

### 4.1Mobility Management

In mobility management, in order to complete the offloading of the corresponding tasks, it is necessary to consider low latency and path prediction, in order to achieve high QoE communication.

Low Latency

Low latency applications, such as IoT (Internet of Things) and Vehicular Network, require high reliability and low end-to-end delay (millisecond) communications. To support low latency, virtual machines and data source are offloaded as users move from one MEC zone to another. The offloading process may have a negative impact on application latency. MEC systems require offloading within a short time. Hence, it is possible to consider a high-speed path with a small delay in the backhaul link, meanwhile, the transmitted file needs to be compressed, and the virtual machine recovery process needs to be simplified (Wang, S. et al., 2017a; Zheng, et al., 2016; Munoz, P-Iserte & Vidal, 2015).

Path Prediction

The key to mobility management is the offloading of virtual machines and files. Traditional

MEC offloading only transfer computing tasks to another server when transfer is happening. Inappropriate data offloading will lead to high latency and will increase the MEC network load. The solution is to analyze user tracks when the MEC provides services to users, and to predict the next MEC that the user will arrive at, in order to transfer the data and resources to the new MEC, in advance. But this technology has two main challenges. The first is track prediction. An accurate prediction can achieve seamless switching between MEC servers and can reduce prefetch redundancy. It requires precise modeling and high-complexity machine learning techniques (Li, S. et al., 2018; Xu, Chen & Ren, 2017; Lu, 2019). The second challenge is how to access the data that needs to be delivered in advance. Inaccurate prediction will lead to unnecessary waste. Balancing the amount of data transmitted and the accuracy of the prediction is challenging. In addition, VM migration leads to a heavy burden on the backhaul link and to high latency. It is necessary to implement a technology that can quickly migrate VMs within a short time (e.g., milliseconds). Hence, if VMs can be migrated in advance, it will better solve both heavy load and high latency (Mao, et al., 2017a).

**4.2 Security Management**

Security is a technical challenge in cloud computing offloading. Since 5G MEC is a distributed system, single point is so weak that the attack of a single point may lead to a destruction of the entire system. Multi-tenant mode will cause malicious users to sneak into the network to exploit cloud platform vulnerabilities in order to attack the network. In addition, the open-source software is vulnerable to attacks as well. File that is offloaded to the cloud or to the edge can also be easily attacked or tampered with. Many of the security solutions originally used for cloud computing are no longer suitable for 5G MEC offloading. The security issues are distributed at various levels, including edge node security, network security, data application security, security situation awareness, security management and coordination, and identity management. Security issues in 5G MEC cannot be fully solved by conventional cloud resolution, due to its unique and complex distributed network. Therefore, it is important to design appropriate mechanisms to deal with malicious safety issues (Shibin & Kathrine, 2017; Xiao, et al., 2018; Roman, Lopez & Mambo, 2016; Xu, et al., 2019; Chaudhary, Kumar & Zeadally, 2017).

**4.3Interference Management**

Interference is also one of the key issues that need to be solved during the offloading process. If many applications of UEs are simultaneously offloaded to the MEC server, interference will occur. How to allocate resources with the guaranteed QoE while mitigating interference is a critical question in 5G MEC offloading. Interference management has multiple implementation methods and is closely related to resource management. The nature of interference is the conflicting use of resources. Unreasonable allocation of network resources is the root of interference. In a distributed 5G MEC network, the large quantity of offloading requests of UEs and the complex network environment can reduce the overall resource utilization. Effective resource allocation is an important means for the management of interference. On the one hand, it can increase network capacity by reasonable use of network resources. On the other hand, it can correct resource allocation strategies and can increase network capacity through interference management (Bu, Yu & Yanikomeroglu, 2015; Li, et al., 2015; Wang, S. et al., 2017b; Huang & Li, 2016).

## Conclusion

5G MEC is a cloud service platform that runs on the edge network. It can improve both service performance and user experience by deploying business processing and resource scheduling to the cloud service platform. 5G MEC offloads the services and capabilities originally in the cloud center to the edge network. By distributing computing, storage, communication resources at the edge, 5G MEC can effectively decrease network overload, can shorten service latency, can save energy consumption and other related costs, and can guarantee task consistency.

In this study, based on 5G MEC offloading, we present two potential robust and efficient offloading approaches to distribute services and to fulfill UE's requests: an optimal offloading strategy and a task consistency offloading strategy. It is a good attempt to examine and to allocate cloud resources and offload tasks from different aspects: low latency, energy efficiency, low costs and consumption, quick executing time, and waiting time.

## References

Abbas, N., Zhang, Y., Taherkordi, A., & Skeie, T. (2018). Mobile edge computing: A survey. IEEE Internet of Things Journal, 5(1), 450-465. DOI:10.1109/JIOT.2017.2750180.

Bu, S., Yu, F. R., & Yanikomeroglu, H. (2015). Interference-aware energy-efficient resource allocation for OFDMA-based heterogeneous networks with incomplete channel state information. IEEE Transactions on Vehicular Technology, 64(3), 1036-1050. DOI:10.1109/TVT.2014.2325823.

Chaudhary, R., Kumar, N., & Zeadally, S. (2017). Network service chaining in fog and cloud computing for the 5G environment: data management and security challenges. IEEE Communications Magazine, 55(11), 114-122. DOI:10.1109/MCOM.2017.1700102.

Chen, M., Liew, S. C., Shao, Z., & Kai, C. (2013). Markov approximation for combinatorial network optimization. IEEE transactions on information theory, 59(10), 6301-6327. DOI:10.1109/TIT.2013.2268923.

Ding, Z., Fan, P., & Poor, H. V. (2019). Impact of non-orthogonal multiple access on the offloading of mobile edge computing. IEEE Transactions on Communications, 67(1), 375-390. DOI:10.1109/TCOMM.2018.2870894.

El Haber, E., Nguyen, T. M., & Assi, C. (2019). Joint Optimization of Computational Cost and Devices Energy for Task Offloading in Multi-tier Edge-clouds. IEEE Transactions on Communications. DOI:10.1109.

Guo, H., Liu, J., & Zhang, J. (2018). Computation Offloading for Multi-Access Mobile Edge Computing in Ultra-Dense Networks. IEEE Communications Magazine, 56(8), 14-19. DOI:10.1109/MCOM.2018.1701069.

Huang, G., & Li, J. (2016). Interference mitigation for femtocell networks via adaptive frequenc y reuse. IEEE Transactions on Vehicular Technology, 65(4), 2413-2423. DOI:10.1109/TVT.201 5.2418813.

Kao, Y. H., Krishnamachari, B., Ra, M. R., & Bai, F. (2017). Hermes: Latency optimal task assignment for resource-constrained mobile computing. IEEE Transactions on Mobile Computing, 16(11), 3056-3069. DOI:10.1109/TMC.2017.2679712.

Kherraf, N., Alameddine, H. A., Sharafeddine, S., Assi, C., & Ghrayeb, A. (2019). Optimized Pr ovisioning of Edge Computing Resources with Heterogeneous Workload in IoT Networks. IEEE Transactions on Network and Service Management. DOI:10.1109/TNSM.2019.2894955.

Kwak, J., Kim, Y., Lee, J., & Chong, S. (2015). DREAM: Dynamic resource and task allocation for energy minimization in mobile cloud systems. IEEE Journal on Selected Areas in Communications, 33(12), 2510-2523. DOI:10.1109/JSAC.2015.2478718.
Li, C., Zhang, J., Haenggi, M., & Letaief, K. B. (2015). User-centric intercell interference nulling for downlink small cell networks. IEEE Transactions on Communications, 63(4), 1419-1431. DOI:10.1109/TCOMM.2015.2402121.
Li, D. X., He, W., & Li, S. (2014). Internet of things in industries: A survey. IEEE Transactions on industrial informatics, 10(4), 2233-2243. DOI:10.1109/TII.2014.2300753.
Li, S., Zhang, N., Lin, S., Kong, L., Katangur, A., Khan, M. K., ... & Zhu, G. (2018). Joint admission control and resource allocation in edge computing for internet of things. IEEE Network, 32(1), 72-79. DOI:10.1109/MNET.2018.1700163.
Liu, J., Mao, Y., Zhang, J., & Letaief, K. B. (2016, July). Delay-optimal computation task scheduling for mobile-edge computing systems. In 2016 IEEE International Symposium on Information Theory (ISIT) (pp. 1451-1455). IEEE. DOI:10.1109/ISIT.2016.7541539.
Lu, Y. (2019). Artificial intelligence: a survey on evolution, models, applications and future trends. Journal of Management Analytics, 6(1), 1-29. DOI:10.1080/23270012.2019.1570365.
Lu, Y., & Li, D. X. (2018). Internet of Things (IoT) cybersecurity research: a review of current research topics. IEEE Internet of Things Journal. Early Access. DOI:10.1142/S2424862218500148.
Lyu, X., Tian, H., Jiang, L., Vinel, A., Maharjan, S., Gjessing, S., & Zhang, Y. (2018). Selective offloading in mobile edge computing for the green Internet of Things. IEEE Network, 32(1), 54-60. DOI:10.1109/MNET.2018.1700101.
Mach, P., & Becvar, Z. (2017). Mobile edge computing: A survey on architecture and computation offloading. IEEE Communications Surveys & Tutorials, 19(3), 1628-1656. DOI:10.1109/COMST.2017.2682318.
Mao, Y., You, C., Zhang, J., Huang, K., & Letaief, K. B. (2017). A survey on mobile edge computing: The communication perspective. IEEE Communications Surveys & Tutorials, 19(4), 2322-2358. DOI:10.1109/COMST.2017.2745201.
Mao, Y., Zhang, J., & Letaief, K. B. (2016). Dynamic computation offloading for mobile-edge computing with energy harvesting devices. IEEE Journal on Selected Areas in Communications, 34(12), 3590-3605. DOI:10.1109/JSAC.2016.2611964.
Mao, Y., Zhang, J., Song, S. H., & Letaief, K. B. (2017). Stochastic joint radio and computational resource management for multi-user mobile-edge computing systems. IEEE Transactions on Wireless Communications, 16(9), 5994-6009. DOI:10.1109/TWC.2017.2717986.
Min, M., Xiao, L., Chen, Y., Cheng, P., Wu, D., & Zhuang, W. (2019). Learning-based computation offloading for IoT devices with energy harvesting. IEEE Transactions on Vehicular Technology, 68(2), 1930-1941. DOI:10.1109/TVT.2018.2890685.
Munoz, O., Pascual-Iserte, A., & Vidal, J. (2015). Optimization of radio and computational resources for energy efficiency in latency-constrained application offloading. IEEE Transactions on Vehicular Technology, 64(10), 4738-4755. DOI:10.1109/TVT.2014.2372852.
Nan, Y., Li, W., Bao, W., Delicato, F. C., Pires, P. F., Dou, Y., & Zomaya, A. Y. (2017). Adaptive energy-aware computation offloading for cloud of things systems. IEEE Access, 5, 23947-23957. DOI:10.1109/ACCESS.2017.2766165.
Ndikumana, A., Ullah, S., LeAnh, T., Tran, N. H., & Hong, C. S. (2017, September). Collaborative cache allocation and computation offloading in mobile edge computing. In 2017

19th Asia-Pacific Network Operations and Management Symposium (APNOMS) (pp. 366-369). IEEE. DOI:10.1109/APNOMS.2017.8094149.
Pan, J., & McElhannon, J. (2018). Future edge cloud and edge computing for internet of things applications. IEEE Internet of Things Journal, 5(1), 439-449. DOI:10.1109/JIOT.2017.2767608.
Roman, R., Lopez, J., & Mambo, M. (2018). Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges. Future Generation Computer Systems, 78, 680-698. DOI:10.1016/j.future.2016.11.009.
Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2016). Edge computing: Vision and challenges. IEEE Internet of Things Journal, 3(5), 637-646. DOI:10.1109/JIOT.2016.2579198.
Shibin, D., & Kathrine, G. J. W. (2017, February). A comprehensive overview on secure offloading in mobile cloud computing. In 2017 4th International Conference on Electronics and Communication Systems (ICECS) (pp. 121-124). IEEE. DOI:10.1109/ECS.2017.8067851.
Tao, X., Ota, K., Dong, M., Qi, H., & Li, K. (2017). Performance guaranteed computation offloading for mobile-edge cloud computing. IEEE Wireless Communications Letters, 6(6), 774-777. DOI:10.1109/LWC.2017.2740927.
Truong-Huu, T., Tham, C. K., & Niyato, D. (2014, December). To offload or to wait: An opportunistic offloading algorithm for parallel tasks in a mobile cloud. In 2014 IEEE 6th International Conference on Cloud Computing Technology and Science (pp. 182-189). IEEE. DOI:10.1109/CloudCom.2014.33.
Wang, C., Liang, C., Yu, F. R., Chen, Q., & Tang, L. (2017a). Computation offloading and resource allocation in wireless cellular networks with mobile edge computing. IEEE Transactions on Wireless Communications, 16(8), 4924-4938. DOI:10.1109/TWC.2017.2703901.
Wang, C., Yu, F. R., Liang, C., Chen, Q., & Tang, L. (2017b). Joint computation offloading and interference management in wireless cellular networks with mobile edge computing. IEEE Transactions on Vehicular Technology, 66(8), 7432-7445. DOI:10.1109/TVT.2017.2672701.
Wang, S., Urgaonkar, R., Zafer, M., He, T., Chan, K., & Leung, K. K. (2015, May). Dynamic service migration in mobile edge-clouds. In 2015 IFIP Networking Conference (IFIP Networking) (pp. 1-9). IEEE. DOI:10.1109/IFIPNetworking.2015.7145316.
Wang, S., Zhang, X., Zhang, Y., Wang, L., Yang, J., & Wang, W. (2017a). A survey on mobile edge networks: Convergence of computing, caching and communications. IEEE Access, 5, 6757-6779. DOI:10.1109/ACCESS.2017.2685434.
Wang, S., Urgaonkar, R., He, T., Chan, K., Zafer, M., & Leung, K. K. (2017b). Dynamic service placement for mobile micro-clouds with predicted future costs. IEEE Transactions on Parallel and Distributed Systems, 28(4), 1002-1016. DOI:10.1109/TPDS.2016.2604814.
Xiao, L., Wan, X., Dai, C., Du, X., Chen, X., & Guizani, M. (2018). Security in mobile edge caching with reinforcement learning. IEEE Wireless Communications, 25(3), 116-122. DOI:10.1109/MWC.2018.1700291.
Xu, J., Chen, L., & Ren, S. (2017). Online learning for offloading and autoscaling in energy harvesting mobile edge computing. IEEE Transactions on Cognitive Communications and Networking, 3(3), 361-373. DOI:10.1109/TCCN.2017.2725277.
Xu, X., Xue, Y., Qi, L., Yuan, Y., Zhang, X., Umer, T., & Wan, S. (2019). An edge computing-enabled computation offloading method with privacy preservation for Internet of connected vehicles. Future Generation Computer Systems, 96, 89-100. DOI:10.1016/j.future.2019.01.012.

Yang, L., Zhang, H., Li, M., Guo, J., & Ji, H. (2018). Mobile edge computing empowered energy efficient task offloading in 5G. IEEE Transactions on Vehicular Technology, 67(7), 6398-6409. DOI:10.1109/TVT.2018.2799620.
You, C., & Huang, K. (2016, December). Multiuser resource allocation for mobile-edge computation offloading. In 2016 IEEE Global Communications Conference (GLOBECOM) (pp. 1-6). IEEE. DOI:10.1109/GLOCOM.2016.7842016.
You, C., Huang, K., Chae, H., & Kim, B. H. (2017). Energy-efficient resource allocation for mobile-edge computation offloading. IEEE Transactions on Wireless Communications, 16(3), 1397-1411. DOI:10.1109/TWC.2016.2633522.
Zhang, H., Guo, J., Yang, L., Li, X., & Ji, H. (2017, May). Computation offloading considering fronthaul and backhaul in small-cell networks integrated with MEC. In 2017 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS) (pp. 115-120). IEEE. DOI:10.1109/INFCOMW.2017.8116362.
Zhang, K., Mao, Y., Leng, S., Zhao, Q., Li, L., Peng, X., ... & Zhang, Y. (2016). Energy-efficient offloading for mobile edge computing in 5G heterogeneous networks. IEEE access, 4, 5896-5907. DOI:10.1109/ACCESS.2016.2597169.
Zheng, J., Cai, Y., Wu, Y., & Shen, X. S. (2016, July). Stochastic computation offloading game for mobile cloud computing. In 2016 IEEE/CIC International Conference on Communications in China (ICCC) (pp. 1-6). IEEE. DOI:10.1109/ICCChina.2016.7636777.
Zhou, W., Fang, W., Li, Y., Yuan, B., Li, Y., & Wang, T. (2019). Markov Approximation for Task Offloading and Computation Scaling in Mobile Edge Computing. Mobile Information Systems, 2019. DOI:10.1155/2019/8172698.