SEAMLESS COMMUNICATION TECHNIQUES IN MOBILE CLOUD COMPUTING: A SURVEY

Pagoui Lagabka Constant[1], Ahyoung Lee[1], Kun Suo[1], Donghyun Kim[2]
[1]Department of Computer Science, Kennesaw State University, Marietta GA 30060, USA, [2]Department of Computer Science, Georgia State University Atlanta GA 30303, USA

NOTE: Corresponding author: Ahyoung Lee, alee146@kennesaw.edu

Abstract – *Achieving seamless communication and smooth service provision between the cloud and end user's mobile device is one of the main challenges existing in mobile cloud environments. Mobile Cloud Computing (MCC) allows cloud environments to mitigate resource limitation problems for mobile devices. The most popular mobile devices such as smartphones, autonomous vehicles, drones, and other smart electronic equipment are in constant motion and frequently change their point of connection (base station or edge) to mobile computing networks. In these situations of mobility, the data being transmitted, and the services being provided to the device should not be interrupted as the proper function of the device depends on these services. Applications that rely heavily on data and services stored in the cloud environment should be available even when the device has moved from one pole to another. Various existing generic surveys emphasize important solutions to some of the challenges faced in MCC. Different solutions were proposed to achieve seamless communication in MCC, presenting the taxonomy of the interworking and mobility techniques and their possibilities. However, they have not provided a clear evaluation of MCC techniques for achieving seamless communication and service provision, and have not taken into consideration current technological advances such as 5G, femtocell, etc. In this paper, we provide a survey of the different solutions proposed to achieve seamless communication in MCC by taking current technological advances into account. Furthermore, some shortcomings associated with the presented methods are outlined, along with the current issues and research challenges faced in MCC. However, for the purposes of data protection and security, previously proposed schemes already achieve the goal of protecting users' attribute privacy and they have the same access policy; some can even achieve full security, but they are just limited in decryption efficiency.*

Keywords – 5G, cloud computing, handover, LTE, mobile cloud computing, mobile cloud environment, mobile networking.

1. INTRODUCTION

Mobile Cloud Computing (MCC) is built based on concepts of cloud computing and mobile computing as the combination of cloud computing technologies with mobile devices to bring rich computational resources and services to mobile users, network operators, and cloud computing providers. The exponential growth and expansion of mobile networks with an approximate 12.3 billion of interconnected mobile devices throughout the world [1] demand more than just conventional network capabilities to operate flawlessly. There is an obvious and urgent need to maintain seamless communication between the user equipment and the edge cloud while the user is mobile and guaranteeing the user's data security and privacy.

Also, with the deployment of fifth generation, 5G, wireless networks, which is expected to lead the mobile networking technology and the proliferation of IoT devices, most of the mobile network's parameters have changed as follows: bandwidth has increased, throughput has gotten better, latency has been reduced, and a wider range coverage is now provided. With these changes comes a range of challenges, such as adapting existing solutions and techniques for maintaining seamless communication, to the new MCC environment and providing high availability and reliability from a rich volume of computational resources.

Several techniques to provide seamless communication in MCC environments have been introduced over the years. And then, there are various existing generic surveys that emphasize the importance of MCC, the different approaches to achieve seamless communication in MCC, and the challenges and the taxonomy for the classification of the interworking and mobility techniques. However, they have not provided a clear evaluation of MCC techniques for achieving seamless communication and service provision, and have not taken into consideration current technological advances.

Although current research advances in networking have provided a plethora of solutions, achieving seamless communication between the user equipment (UE) and MCC environments remains one of the main challenges in MCC. Most mobile devices such as smart phones, autonomous vehicles, drones, and other smart electronic equipment are in constant motion and may lose connectivity to the edge cloud for a short amount of time changing their point of connection (base station or edge). In such situations, the data being transmitted, and the services being provided to the device should not be interrupted as the proper function of the device depends on these services.

Unfortunately, seamlessly executing computation-intensive applications in MCC environments is still complex. Several factors must be taken into consideration in order to achieve efficient and seamless communication in MCC; besides, the security and privacy of data must be maintained as well. The main contribution of this study is to present an evaluation of the existing techniques for seamless communication in MCC, a classification of the different solutions, and future directions for research by taking into account current technological advances.

The rest of this paper is structured as follows. Section 2 presents background information including an architecture overview, performance metrics and resource management of MCC. Section 3 describes the related work and Section 4 outlines existing techniques and solutions for achieving seamless communication. In Section 5, we discuss the challenges and future research directions; then, the conclusion is given in Section 6 of the paper.

2. BACKGROUND INFORMATION

2.1 Overview of mobile cloud computing architecture

A mobile cloud computing environment consists of a cloud server structure and mobile network structure. Mobile UEs communicate with the cloud server through a mobile network. Mobile device data and services are migrated to the edge cloud to improve the performance of the real-time intensive mobile applications. For the cloud to achieve smooth service provision, a very well defined MCC architecture is needed. Furthermore, achieving seamless communication in MCC environments requires an architecture that allows effortless computation offloading of data and computation-intensive tasks on the edge servers.

The MCC stakeholders ecosystem involves different partakers such as mobile users, network operators, Internet Service Providers (ISP), application services and Cloud Service Providers (CSP). These partakers are all interconnected through several networks from the edge to the cloud's data centers. Mobile users are the consumers that represent the mobile terminal of the cloud; network operators and ISP provide network infrastructure and data services to access the cloud environment, that is the Infrastructure as a Service (IaaS) part of the cloud. Application developers and CSP offer a software licensing and delivery model in which users purchase their software licenses on a subscription basis and use the software on the platform. Such business model is known as Software as a Service (SaaS) [2].

MCC Networking: There are two ways through which mobile devices can access cloud services, either via a Mobile Network (MN) or Access Points (AP).

- *Mobile Network (MN)*: It provides a connection between the mobile device and the cloud environment through base stations or satellites. It has evolved from the GSM (2G Global System for mobile communications) that uses circuit-switched with fixed slots allocated for transmission over the air, to 3G universal mobile telecommunication systems, 4G Long Term Evolution (LTE) and 5G core network [3]. The mobile network architecture consists of UE or a mobile device, radio access network, core network, inter-network and radio channel. In [4], UEs can be connected with multiple links through mobile networks and/or satellite; if a satellite module is not integrated into the UE, then external satellite communication devices are used. MNs are linked to the Internet, they also provide Internet access to their users. Thus, the UE receives cloud services through the Internet.
- *Access Point (AP)*: APs are edge devices that are connected to Internet service providers and provide an Internet connectivity to mobile devices through Wi-Fi. Once mobile devices are connected to the Internet, they can access cloud services. APs are commonly used to access a cloud as they provide a Wi-Fi-based connection, which has lower latency compared to MN.

From Hoang T. Dinh et al. [2], the communication between the UE and cloud environment in network systems prior to 5G and LTE is done as follows. The UE's requests and information are transmitted to the processors of the data centers through the edges that are connected to servers responding to mobile network services' requests. Services such as authentication, authorization, specific bandwidth and pay-as-you-go-based Internet services are provided to mobile users by mobile service operators or Internet service providers. Requests are then delivered to the cloud through the Internet; these requests are processed by controllers in the cloud to provide mobile users the appropriate services.

Fig. 1 illustrates the difference between cloud computing communications as a traditional cloud computing architecture in Fig. 1(a) and as a hierarchical 5G-enabled MEC architecture in Fig. 1(b). The traditional cloud computing paradigm faces substantial challenges, such as great communication overhead or long latency, due to the limited computational capability of IoT devices and geographically remote servers from the cloud, which is hard to satisfy the requirement of delay-sensitive tasks or resource-constrained IoT sensing devices. To solve those problems in traditional cloud computing with mobile users, the edge cloud was proposed as an extension of cloud computing. In this environment, the edge computing network was designed with cellular and other mobile devices, which enables computation and communication resources to be dispersed to the edge network closer to the end users, to provide efficient and low-latency services.

Moreover, 5G mobile networks present slightly different architecture and functionality, such that edge devices including base stations and wireless access points provide rich computation and storage resources that are sufficient to enable ubiquitous mobile computing [4]. 5G systems support *communications, computing, control and content*

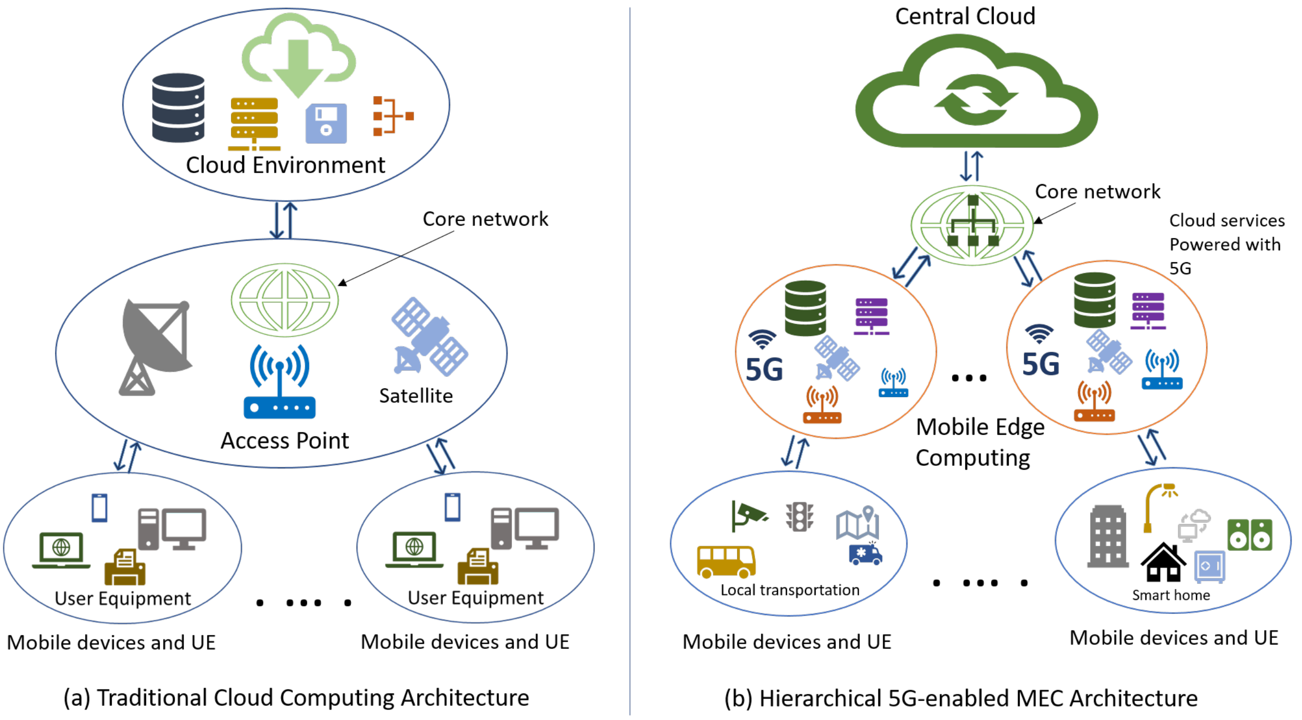


Fig. 1 – Overview of cloud communications: (a) A traditional cloud computing architecture provides access to resources between remote servers in the locally central cloud and the end users. (b) A hierarchical 5G-enabled MEC architecture provides access to resources to the end users from remote serves both in the locally central cloud and more than one mobile-edge clouds.

delivery for real-time and computation-intensive applications with Virtual Reality (VR) and Ultra-High Definition (UHD) video features.

Mobile Edge Computing: Mobile Edge Computing (MEC), also referred to as multi-access edge computing, is a standard that defines a network architecture in which cloud computing capabilities and services are enabled at the edge of the mobile network. When cloud services are provided closer to the mobile UE, latency and network congestion are reduced and the applications running on UE perform better. The design of mobile edge computation networks is conceptualized by taking into consideration the aftermath of both communication and computation. Edge cloud servers are implemented directly at base stations using a generic computing platform for allowing the execution of applications closer to the end-user equipment; they act as cache servers as well as transcoding servers with a given storage capacity and computing abilities [5]. The more detailed important roles of MEC architecture in 5G networking systems are described in [6] and its main services can be summarized as shown below.

- *Storage:* Since the storage capacity of UE is limited, the edge cloud handles a large amount of delay sensitive data generated by UEs in a real-time manner as accessing cloud computing systems directly increases latency.

- *Computation Offloading:* Computational tasks and processes requested by UEs are offloaded from the UE to the edge cloud; MEC integrates computing systems that provide on-site computation and information processing, which help to reduce latency and achieve real-time responses from the cloud. Computation offloading provides computation solutions to data intensive applications that require high computational processes. The following parameters are considered when performing tasks or data offloading: the transmission status between the UE and its edge server and the current edge server load status [7].

Tasks in MCC eligible for offloading can be classified into two categories: *computation-intensive* and *data-intensive*. Computation-intensive tasks are the type of tasks that need heavy computations with relatively fewer amounts of data transfers. The offloading decision of these tasks depends on the amount of required computations. Data-intensive tasks are the type of tasks that need a large amount of data transfers. These offloading tasks to the MCC environment are vital for the performance of the applications, and the offloading decision of these tasks heavily depends on the network bandwidth, since a network with lower bandwidth will increase latency and waste the UE energy [8].

This can be observed in the following mathematical relation, by considering a wireless access base-station s, which can be either a Wi-Fi access point, femtocell, or macrocell in cellular networks.

In [9], the uplink data rate $R_i(a)$ for computation offloading of mobile device user i is defined for the computation offloading decision $a_i \in$ {0,1}, where $a_i = 1$ if user i chooses to offload the computation to the cloud, otherwise $a_i = 0$ if user i decides to computer its task locally on the mobile device. Hence, the uplink data rate $R_i(a)$ is

$$R_i(a) = Wlog_2(1 + \frac{P_i H_{i,s}}{w_i + \sum_{m \in N \setminus \{i\}: a_m = 1} P_m H_{m,s}}), \tag{1}$$

where W is the channel bandwidth, P_i is the user's transmission channel, $H_{i,s}$ is the channel gain between the mobile device user i and the base station s, and w_i is the background interference power.

Besides, the maximum rate of channel capacity C in an Additive White Gaussian noise (AWGN) channel is defined in [10] as:

$$C = B\, log_2\Big(1 + \frac{P}{N_0 B}\Big), \tag{2}$$

where B, P and N_0 are the channel bandwidth, transmit power, and power spectral density of the noise, respectively. Thus, providing enough bandwidth for data-intensive tasks is vital in order to minimize energy consumption and latency in MCC networks.

- *Data Analysis:* Data gathered from UEs can be processed and analyzed at the edge level to extract essential information. This reduces the latency of sending and receiving data to the cloud for analysis.
- *Security:* Edge computing enhances cloud environment's security at the edge of the networks through micro-service management, hardware-assisted, caching systems, Software Defined Networking (SDN) and the use of machine-learning-based techniques. Several techniques have been proposed to protect vulnerable systems against various attacks such as Distributed Denial of Service attacks (DDoS), wireless jamming, spoofing and man-in-the-middle attacks.

  Security solutions that apply reinforcement learning techniques to provide secure offloading to the edge nodes were proposed by [11]. A deep learning-based physical layer authentication that uses spatial heterogeneity of wireless channels was proposed by [12], and their techniques distinguish multi-users such as legitimate edge nodes from attackers and malicious nodes without a test threshold.

In order to improve performance of mobile cloud computing, edge computing can be enabled in 5G networks through SDN, network function virtualization, massive MIMO, dynamic radio technologies access, D2D Communication, etc. However, the resources and services provided by the edge cloud are limited and can only support a finite number of devices.

Layers of the MCC Architecture: MCC architecture includes five main layers: the *application layer*, the *perception layer*, the *network infrastructure layer*, the *Internet communication layer*, and the *computation layer* [13]. The application layer correlates different mobile applications; it demands high computational power and is responsible for delivering end user resource-demanding services. The perception layer handles the physical connection with mobile devices; it relies on the network infrastructure layer to establish a smooth connection to access more computation and cloud applications services. The network infrastructure layer corresponds to the layer that handles the configuration of the physical mobile network. Besides, it serves as a connection gateway from the perception layer to the computation layer and represents the cloudlet infrastructure, which is used as an edge's link between UE and the cloud environment. The Internet layer coordinates the interconnectivity and communication of the mobile devices and the Internet; it plays the role of the link using Transmission Control Protocol (TCP), User Datagram Protocol (UDP) and Internet Protocol (IP) suite to connect mobile devices to the cloud environment. The computation layer is associated with the computation phase for offloaded mobile tasks - it includes massive storage resources, servers and task offloading managers and is in charge of decision making and data analysis/other real-time services provided to the UE. An illustration of the five-layers architecture for MCC is presented in Fig. 2.

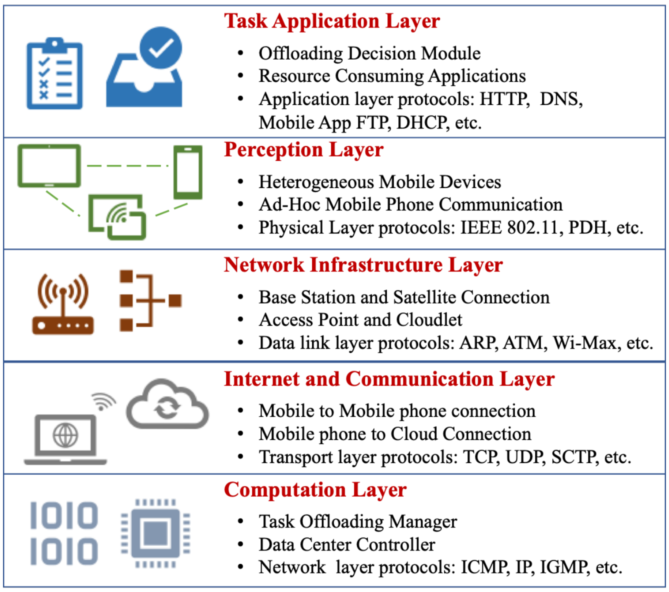

Task Application Layer
• Offloading Decision Module
• Resource Consuming Applications
• Application layer protocols: HTTP, DNS, Mobile App FTP, DHCP, etc.
Perception Layer
• Heterogeneous Mobile Devices
• Ad-Hoc Mobile Phone Communication
• Physical Layer protocols: IEEE 802.11, PDH, etc.
Network Infrastructure Layer
• Base Station and Satellite Connection
• Access Point and Cloudlet
• Data link layer protocols: ARP, ATM, Wi-Max, etc.
Internet and Communication Layer
• Mobile to Mobile phone connection
• Mobile phone to Cloud Connection
• Transport layer protocols: TCP, UDP, SCTP, etc.
Computation Layer
1010
1010
• Task Offloading Manager
• Data Center Controller
• Network layer protocols: ICMP, IP, IGMP, etc.

Fig. 2 – The layered architecture of mobile cloud computing.

2.2 Performance metrics of MCC

Several parameters should be considered when evaluating the performance of MCC. Performance metrics are determined based on cloud service providers and end users' needs. The end user needs seamless network connectivity, reliable and uninterrupted service provision while moving around, and faster responses from the cloud servers. On the other end, the cloud service provider needs to meet user's requirements while it provides fast and reliable services and reduces the overall cost of servers and infrastructures. Therefore, the following metrics are major determinant factors of MCC environment performance.

Latency: Latency is the cause of delays noticed by end users. There are several techniques to reduce latency from the UE point of view, among which we have displayed local animations, background loading to hide latency or pre-fetching and parallel connections on multiple threads. Latency occurs as well within the MCC environment, delays can arise anywhere from the edge to the data centers. Besides, there are techniques for reducing the average end-to-end delay in the MCC environment including machine learning and adaptive priorities based on when the request was initiated. The average latency of a device i to upload its computation task to a base station j is defined by [14]

$$L_{j,i}^{tran,d} = \frac{L_{j,i} * T}{R_{j,i} * \tau_{j,i}}, \tag{3}$$

where $L_{j,i}$ is the input data-size (in bits) for processing the computation task of the $i-th$ device, T is the length of one Time-Division Multiple Access (TDMA) frame, $R_{j,i}$ is the expected channel capacity, and $\tau_{j,i}$ is the time slot resource for each device.

Energy Consumption: Network energy consumption in UEs is mainly observed during task offloading, task execution or computation. If an edge cloud, associated with the base station to which the UE is connected, executes its UE's task, then the computation energy consumption is proportional to the changed capacity of the edge cloud. If the central cloud executes the task, then the consumed energy can be defined by the energy consumption of the cloud which the edge cloud is associated to[15]. The energy consumed by a node i is defined by [16]

$$E_{c(i)} = N_T * A + N_R * B, \tag{4}$$

where $E_{c(i)}$ is the absorbed energy by the node i after a given time, N_T and N_R are the number of transmitted and received packets, respectively, A and B are constant factors based on the energy model.

Bandwidth Utilization: Bandwidth is the measure of the capacity of a channel to transfer data in a network. The wider or greater the bandwidth, the greater the amount of data that can be transferred and the number of users that can be handled by the network. Therefore, it is vital to maintain high bandwidth in order to achieve seamless communication in MCC networks. The available bandwidth of a channel i is defined by [10]

$$B_i = \frac{b_i(t)}{\sum_{i=1}^{I} b_i(t)}(B_{tot} - \beta), \tag{5}$$

where b_i is the bid vector given by the gateway, B_{tot} is the total maximum bandwidth in the Cloud Service Provider (CSP), β is the reserved bid for the CSP, and I the total number of gateways.

Reliability: It is the probability that a mobile device will perform as intended, so that its functions are satisfactorily executed for a given period of time under specified operating conditions in MCC. Thus, the reliability of a MCC setting is defined by the equation below:

$$P = \sum_{i=0}^{K} \pi * (1 - P(N+i-K, M, i)), \tag{6}$$

where

$$P(N+i-K, M, i) = \frac{C_i^M}{C_{N+i-K}^M}, \tag{7}$$

with

$$C_i^M = \binom{i}{M} = \frac{i!}{M!(i-M)!} \tag{8}$$

and

$$C_{N+i-K}^M = \binom{N+i-K}{M} = \frac{(N+i-K)!}{M!(N+i-K-M)!} \tag{9}$$

for $i > M$ and 0 otherwise. Here, N is the number of available paths, M is the number of actually used paths, K is the maximum number of failure paths, and i the number of failed paths [17].

Service Availability: It refers to the state of being used or obtained, such that MCC availability is directly proportional to its number of active edges and BS. It is essential for every MCC systems and mandatory for cloud service providers. Besides, it is actually one of the key factors to procure seamless data exchange in MCC environments, thus there can be interruptions of services or flow of data if the system is not 100% available.

Quality of Service (QoS): It is the measurement of the response of a system to different requirements, standards, and objectives expected by end users. Thus, it denotes the level of performance, reliability, and availability offered by a system. Moreover, QoS is sometimes associated with Quality of Experience (QoE), which is defined by techniques such as Mean Opinion Score (MOS), Net Promoter Score (NPS) or Standard deviation of Opinion Scores(SOS).

Security: Most of mobile devices contain end-user personal information such as pictures, a list of contacts, frequent locations, payment information, etc., which are targeted by attackers. Unfortunately, most mobile devices are unprotected and vulnerable due to their limited resources in terms of computation and storage, so that they cannot run powerful protection systems.

2.3 Resource management and allocation in mobile cloud computing

The cloud model consists of three service models according to the NIST cloud computing reference architecture [18]: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). Computing resources provided by the cloud systems are brought closer to end users through architectural-based and implementation-based techniques, such that these resources include computation power, storage, network infrastructure, data partitioning and scaling, security, and location through IaaS. Other resources are managed by PaaS such as computing platform, which includes the operating system, programming language execution environment, database, and web server. Applications and software are also provided as resources through the SaaS branch. From these cloud computing services, cloud computing resources can be classified in two categories: *data center resources* including storage, network bandwidth and available servers, and *computing resources* provided directly to mobile devices. Data centers comprise multiple branches for resource allocation and optimization, while computer resources include super computers, clusters, virtual machines, and operation system disk images. Therefore, we can define a resource allocation as the process of allocating available cloud resources to various mobile applications running in the cloud environment. It is performed with the objective of minimizing the costs of executing tasks and performing data exchange [19].

3. RELATED WORK

Several survey research papers have been published in the domain of MCC regarding the overall architectures and technical approaches to reduce latency, improve security, and provide reliable services and seamless communication to end users.

For instance, Othman et al. [4] proposed a generic survey of MCC application models and its different aspects including MCC architectures and its offloading decision affecting entities. The article also presents a comparison of MCC application models based on criteria such as bandwidth utilization, privacy, latency, generality, security, programming abstraction, scalability, complexity, and execution resource. Furthermore, the survey categorizes MCC application models by performance-based, energy-based, constraint-based, and multi-objective applications. A. Gani et al. [20] presented a review on mobility techniques for seamless connectivity in MCC to highlight issues and challenges faced when providing computational cloud services in MCC environments. Also, they discussed a comparison and classification of different seamless connectivity schemes in MCC. The mobility techniques were classified based on a connectivity approach, interworking method, mobility operation, network topology, and inter-working architecture. However, from the study, it should be noted that the use of a similar strategy and development idea of inter-operational and mobility techniques to overcome the challenges faced in intensive distributed mobile computing networks can be considered an efficient solution for achieving seamless connectivity.

MCC issues and research directions were discussed by Shon et al.[21], and different cloud computing systems and models were reviewed as well. They discussed MCC issues such as smartphone data slinging, access control and identity management, risk of multiple cloud tenants and security threats associated with authentication and authorization, and emergence of cloud standards and certifications. However, they didn't mention performance criteria that can be used to evaluate and highlight the MCC performance issues, and the scope of the study was not well defined as a broad overview of cloud computing.

A survey on data offloading techniques in cellular networks proposed by Rebecchi et al. [22] unveiled different offloading techniques and the principal requirements to embed data offloading abilities in mobile networks. Hence, the survey categorized existing offloading techniques based on their requirements in terms of assurance of content delivery, summarized existing works, described general architecture to enable mobile data offloading, and discussed open research issues. In order to achieve a state where there are adequate delivery conditions, there are two main offloading approaches considered in the survey: latency-free offloading in which every packet has a strict delivery latency constraint defined by the application, and impeded offloading where the reception of contents may be delayed on purpose up to a certain point in time.

Ejaz Ahmed et al. [23] proposed a comprehensive survey on seamless application execution frameworks in MCC in which they revealed state-of-the-art approaches proposed in order to achieve seamless execution of MCC applications. Hence, the survey classified the frameworks based on their implementation locations, and types such as cloud-based, cloudlet-based and hybrid, and presented categorical approaches that are used by the frameworks to achieve seamless execution of MCC applications.

M. Chiang et al. [24] presented an overview of research opportunities in the area of fog and IoT focusing on the network context of IoT in a survey. The survey discussed a group of different challenges in the field of IoT and barriers that are found when trying to overcome these challenges using computational resources. The challenges listed in the area of IoT include constraints and limitations of factors such as network bandwidth, latency, seamless services provision with intermittent connection to the cloud environment, resources constrained devices and IoT security challenges. They also highlighted the potential and challenges of a fog data plane and control plane such as the control, interface, configuration, and management of networks.

Yuyi Mao et al. [25] presented a comprehensive survey of the state-of-the-art mobile edge computing focusing on its communication perspective including joint radio and computational resource management. Hence, the proposed work equally highlighted interesting topics on MEC such as deployment of edge cloud systems, cache-enabled MEC, management of mobility for edge cloud, and privacy aware edge cloud. In addition a mobile computing platform for 5G is presented, as well as the comparison between MCC and MEC, MEC computation and communication models, resource management in MEC and a list of issues and research directions.

Issues and challenges encountered in MCC computation offloading have been highlighted by Akherfi et al. [26]. Hence, they presented state-of-the-art data offloading techniques, computation offloading methods, and an analysis of the techniques along with their principal issues. They additionally explored the major parameters on which the frameworks are based and implemented such as offloading method and grade of partitioning. In addition, the MCC computation offloading was defined as the task of sending computation intensive application components to a remote server, which handles and executes the computational tasks. Different offloading approaches, framework mechanisms and classes were also presented along with an insightful comparison of the frameworks for computational offloading. Some of the approaches that were presented in the paper use static offloading unlike others that utilize dynamic offloading. However, all the techniques were aimed to improve the potentialities of mobile devices by saving energy, reducing response time, or minimizing the execution cost.

In the same perspective, Shakarami et al. [27] proposed a survey of the stochastic-based computation offloading approaches in MCC environments including a taxon- omy of the techniques categorized into three fields, which are Markov process, Markov chain, and Hidden Markov models. The article defined Markov chain as a mathematical tool to model a transition from one state to another based on specific probabilistic rules. In addition, the survey highlighted a comparison of the Markov chain offloading mechanisms and open is- sues and challenges associated with different approaches.

4. EXISTING TECHNIQUES AND SOLUTIONS

The most common infrastructural solutions and techniques proposed by scholars to achieve seamless service provision in mobile cloud computing include fog, edge computing, handover techniques and femtocell technologies. Several algorithms have been proposed to determine the optimal edge computing point, reduce latency of data traffic, improve data offloading speed and enhance security in the MCC environment. Studies on MCC systems highlight that researchers in the field of mobile com- puting, software engineering, cloud computing, and arti- ficial intelligence have successfully utilized MCC architec- tural models and infrastructures to improve the perfor- mance of MCC systems through its software.

4.1 Fog architecture-based solutions for seamless handover

Wan et al. [28] proposed a novel fog computing archi- tecture that includes new schemes and techniques for symmetric inter-file coded cache placement, handling the inter-SBS communication phase, and a new asymmetric and optimal cache placement that performs file sub packaging according to the network structure.

Fog computing is known as a favorable architecture for computing and resource management that provides cloud services closer to end users, that is at the network edge. It includes both the data plane and control plane and aids applications in the area of IoT, in 5G systems and artificial intelligence. Fog computing reduces the need for specialized applications deployed just for the cloud, endpoints or edge devices, by enabling the same application to run anywhere and allowing applications from different suppliers to run on the same hardware without interference [29].

Luan et al. [30] outlined the main features of fog computing and described its concept, architecture and design goals in an article. Fog computing is an architecture that enables the deployment of virtualized cloud-like devices closer to mobile users. Edge computing is a distributed computing paradigm that enables cloud services closer to the location where it is needed, it extends cloud abilities at the edge of the computing network to execute high-demanding computational tasks and save a very significant amount of data at the surroundings of user equipment [6]. Communication between fog nodes is optimized through the handover process. Handover (HO) is a process of passing on an ongoing data session or service from one base station within the core network into another base station; it is a cross-layer concept to support user mobility.

A fog-aided architecture for seamless handover was proposed by [31]; the proposed architecture includes a general integration of all types of mobile devices and networks and assists Vehicle-to-Everything (V2X) distributed applications by responding to their latency minimization related needs, data privacy and security critical network related requirements. The proposed architecture is leveraged on 5G architecture, along with SDN and NFV to achieve proactive, context-aware, and secure handover mechanisms. Hence, fog-enabled architecture and SDN-enabled architecture have been combined in this approach; the authors assumed that connected vehicles are fog devices with distributed intelligence since vehicles mobility can be predicted and they are computing resource rich, they are thus equipped with satellite and terrestrial communication capabilities. For the SDN enabled architecture part, distributed controllers are implemented in fog-enabled vehicles and base stations to ensure fast and efficient handover. Handover decision should be kept local thus it important to implement the proposed SDN architecture in fog enabled devices that only respond to events taking place in their vicinity.

Kapsalis et al. [32] presented a cooperative model, a fog architecture where the tasks to be completed by the nodes are characterized by their computational characteristics and are assigned to the appropriate host subsequently. The model consists of different layers including a hub layer, device layer, fog layer, and cloud layer. The Device layer includes actual physical devices that have small computational power and low storage, the hub layer contains gateways and is in charge of creating fog messages and forwarding them to the fog layer acting as mediator, the fog layer includes the computing edges that function in collaborative way to execute tasks and the cloud layer provides a guaranteed execution environment to the tasks. The proposed solution allows fog networks to be optimal in executing time critical tasks. It integrates into edge computing architectures, the communication between devices in edge networks via the MQTT messaging protocol and the inclusion of nearby access points or mobile edges in a collaborative way for specific types of tasks, to allow better efficiency, coverage and QoS. However, this solution omitted to take into account some cases where the expected participation of some edge devices cannot be guaranteed.

To enhance the scalability of fog-computing and augment its computational power and storage power in mobile cloud computing, Sookhak et al. [33] proposed a fog architecture-based solution called Fog Vehicular Computing (FVC). In this solution, it is suggested that a pool of parked smart vehicles can be used as a source of computing resources, referred to as FVC zone. The maximum capacity of an FVC zone is determined from the predicted need of computational power and resources in the area. The FVC architecture has three main layers, the policy management layer, the application and services layer and the abstraction layer. The application and services layer is responsible for providing real-time applications to end users according to collected data from the deployed sensors in the inertial navigation system; it provides services such as information and entertainment as a service, network as a service, storage as a service and entertainment as a service. The policy management layer allocates appropriate computation and storage resources to different tasks, deals with issues such as monitoring the system state dynamically, and includes policy, fog, and vehicular cloud. The abstraction layer protects the security and privacy of data; It conceals the FVC heterogeneous platform and reveals a monotonic interface for monitoring, delivering, and maintaining the physical resources, such as memory, processor unit, and networking. FVC's architecture's decision process includes a decision manager that computes the completion time of a task and assigns the task to the required sub-layer which also includes a resource manager. This solution relies entirely on the supposition that there will always be a gathering of smart vehicles with enough computational power and resources to operate as fog devices in high-traffic area; however, that is not always the case. Currently, the ratio of smart vehicles to the regular ones is not significant, thus a group of parked vehicles might not be a considerable source of computational resources.

Besides, Bruschi et al. [34] also proposed a framework that leverages fog computing, SDN and NFV capabilities to respond to the necessity of bringing services to the edges and make them more accessible to users to reduce latency during service provision, and reinforce the personalization of services. The proposed framework operates by considering three main stakeholders including CSPs, telecommunication operators, and end users; it includes several functional blocks and interfaces to allow future cloud applications to perform efficiently and provide more than standard services, and enable end users and telecommunication operators to benefit by providing application services. Its architecture leverages tools such as OpenVolcano, which manages functionalities of the data plane and control plane associated with real-time analytics, an external controller that provides decisions on the long-term.

To add additional support to user mobility, allow service differentiation and help applications achieve seamless service provision in the MEC environment, Bruschi et al. [35] presented a policy regarding virtual object clustering and migration; the proposed policy takes into consideration end users proximity, and involves a parameter of the subscription-based proximity ranging to enable service differentiation between users. The authors considered a network of fog-hosted virtual objects with a variety of proximity distances and requirements where an individual user belongs to a given set of virtual objects. User proximity is computed and classified in different levels according to the different requirements and subscription-based parameters; virtual objects clustering is performed according to their inter-affinities, which are classified in different levels, and are merged based on the maximum path lengths and proximity levels; after the merging of different clusters, C clusters and their corresponding minimum proximity requirements are obtained. The next step involves cluster migration, in which quality of service is maintained while the end user moves from one location to another as some of the proximity requirements are no longer met; thus, migrations are performed based on user's previous and new locations, access point time, and shortest path length from the device. This solution is however limited due to the fact that multiple affinity levels and computational power and capabilities of different access points and data centers were not considered.

Santa J. et al. [36] proposed a framework called MIGRATE to provide an efficient and seamless handover of application services to mobile devices and support UEs' operations. MIGRATE leverages MEC's capabilities and the dynamic creation of virtual mobile devices to perform data processing and caching given the limited capabilities of physical mobile devices, and allow mobile devices to maintain MEC services while moving to a new network domain with virtualisation capabilities. To provide MEC services closer to mobile devices, the authors considered edge virtualization domains, in which mobile devices are deployed and whose data is updated using a local access to a cluster-based database. Then, the services are deployed in the cloud domain as virtual functions, and the devices continuously pass on data to the platform thanks to an SDN switch that is used as an entry point to the wired network. The migration of services from one access point to another is instantiated when the switch detects a packet coming from the same mobile device address to a new port; when this happens, the switch reports it to the SDN controller, which either re-routes the traffic towards the initial mobile device or requests the preparation of a new virtualization domain to host a new virtual mobile device that inherits the behavior and characteristics of the initial one. After that, the SDN waits for a notification confirming the completion of the action to establish a new route in the switch and send data through the new virtual mobile device. This solution can be further extended to reduce the latency of service migration and use an SDN multi-controller solution.

4.2 Edge server and base station placement solutions

To solve the edge server placement problem in MCC, Wang et al. [37] proposed a solution that uses mixed integer programming to determine the optimal placement location of the edge. The problem was first formulated as a multi-objective restraint optimization that incorporates edge servers in some appropriate locations to stabilize the workloads of edge servers and minimize the access latency. In the article, the authors considered a network G with a set $s_1, s_2, s_3, \dots, s_k$ of K edge servers to be placed in K optimal places; the edge server executes all the tasks assigned to the base stations, that is, the amount of requests performed by mobile users at each base station $b_j \in B$. The locations must be chosen in such a way that the workloads are balanced, and the access delay reduced, taking into consideration the following constraints: No two edge servers share the same base station, each base station is co-located with an edge server which will execute all mobile user task requests from the base station. The weighting sum method was adopted in this solution to change the problem of edge server placement into a signal objective optimization problem with a Pareto optimal solution which is obtained by transforming the multi-objective into a single optimization problem. However, in this solution, the authors assumed edge servers are homogeneous, that is, they have the same computational resources power, and it is not not the case in practical environments; thus, this approach is limited to an environment with a homogeneous setting.

Lee, Daewon, et al. [38] proposed an MCC proxy-based architecture to improve link performance between mobile hosts and an algorithm to optimize bandwidth usage. The proxy-based algorithm includes three parts, which are denoted as initial part, proxy election part, and sub-proxy election part. In the article, the network congestion problem is solved by improving the link performance using proxy as a cache server. The proxy server is selected based on four parameters, which include the type of host, the state of the host, the hardware performance of the host and the available amount of concurrent connections. These four parameters are constantly checked by the proxy manager to perform proxy selection. Also, information from network layer 3 is used to select the optimal access network. The following information is required to find the appropriate access network: the state of the network, the hop count to the selected proxy, the highest capacity of the network, the expected network load and the location or the depth of network hierarchy.

A cooperative edge caching approach to reduce delays in clustered mobile networks by optimizing content placement, small cell base stations, and bandwidth allocation in large-scale user-centric mobile networks based on the stochastic network information, was proposed by [29]. The proposed solution solves two problems, the problem of content placement and that of small cell base stations clustering. The article considered a homogeneous mobile network with edge caching, where content is partially or completely stored at each small cell base stations after being coded into segments, the user is served by a cluster of candidate base stations after raising a content request. The mobile device seeks coded segments from candidate base stations in increasing order of transmission distance, if the requested content is cached. Also, in case where the segments obtained from the caches of all candidate base stations are not sufficient to decode the segments, the closest base station will fetch the rest of the bases from remote servers via backhaul, and send them to the user through wireless transmission. If the requested content is not stored in cache, the nearest SBS will fetch the whole content from remote servers.

Guo et al. [39] proposed a solution to the edge placement problem in order to optimally allocate workload to edge clouds and minimize communication latency between the edge server and mobile devices. The proposed approach is based on K-means and mixed-integer quadratic programming; to solve this problem, the authors considered a mobile edge network including several base stations and potential locations for edge clouds placement and represented it by a graph $G = V \cup S, E$ where V represents the base stations, S is the set of potential locations for edge clouds placement and E is the connection between two base stations. The steps to solve the edge placement problem are established according to the minimum communication latency between two base stations and the minimum workload of each edge cloud. The scheme takes as input a set of base stations and edge clouds, and returns the optimal locations of edge clouds. It first finds out if there is an edge located at a given location, if a base station is allocated to a given edge cloud and if the base station is associated with an edge cloud; then, it defines a fitness function such that the edge placement problem is transformed into a single objective optimization problem by using a weighted sum method. This problem is solved by selecting the locations with minimal communication delay using K-Means algorithm and simplifying the workload allocation problem using a mixed integer quadratic programming algorithm, and then solving it using the Boolean Quadric Polytope cutting plane method. The proposed approach is however not the most efficient; change of workload size during the allocation is not taken into consideration, which makes the solution less reliable.

4.3 Energy consumption and latency minimization during data offloading

A system that minimizes execution latency during the migration of a mobile web worker from mobile device to an edge server and provides its seamless offloading was proposed by Jeong, Hyuk-Jin, et al. [40]. In the system, the intact web app that has computation-intensive codes executed in a web browser, is run by a mobile client. When accessible edge servers are detected by the client, the mobile web worker manager is responsible for finding the best server to process the worker, which reduces the delay between the time at which a request is sent by the main thread to the worker and the time at which a result is received from the worker. Thus, the HTML5 web worker is migrated across the cloud, the client, and the edge, and keeps the offloading states while the mobile client switches its objective server. Web snapshots are used to move web workers by the system, by a script written in JavaScript to restore the run-time state of a web worker when this one is executed. The authors also highlighted issues of generating a snapshot code that restores both JavaScript objects and native data such as web assembly functions and built-in objects.

To reduce energy consumption and latency in fog computing architecture, Quang Duy La et al. [41] proposed an approach that uses device-driven and human-driven intelligence as key enablers; it performs adaptive low latency Medium Access Control (MAC)-layer scheduling among sensor devices, and detects user behaviors, by applying machine learning techniques. The authors equally developed an algorithm to perform efficient offloading decision in the presence of multiple fog nodes. Achieving device-driven intelligence refers to equipping devices with smarter functionalities such as sensing, computing, storage, smart data processing, networking services and communication; human-driven intelligence associates human domain data with network-domain decisions that will benefit the network [41].

The article presents two case studies, user-behavior-driven healthcare monitoring and device-driven adaptive task offloading. The first case study involves using a machine learning technique-based health monitoring module to create a non-complex ML model that detects human activities driving the sampling of an adaptive sensor and scheduling scheme of MAC using some data and accelerometer sensors. The second case study depicts an environment involving an end user with N independent tasks, where each task has the possibility to be offloaded to a computer processor of any of the available fog nodes or processed locally by the end user's computer processor; for each task, the user must decide the appropriate CPU to be used to process it with the objective to reduce delay and energy consumption. The energy consumption and latency minimization problem is a mixed-integer nonlinear programming that is solved by first transforming the problem into a corresponding uniform Quadratic Constrained Quadratic Programming (QCQP), dropping the rank-one limitation, which makes the QCQP problem SemiDefinite Programming (SDP) convex and can be cleared up using the interior point method, and then constructing a number of reasonable solutions based on Gaussian randomization, and finally choosing the solution, which minimizes the objective function over all solutions. The shortcoming associated with this solution is the fact that intelligence in fog computing is still in its infancy and the assumptions made are not realistic yet.

Amir Erfan Eshratifar et al. [42] introduced BottleNeck, a new deep learning architecture to reduce the workload size to be sent from the UE to the cloud, along with a training method to compensate for the potential accuracy loss that arises during the compression of the workload before its transmission to the cloud. BottleNeck is basically an auto-encoder in which the agent handles the responsibility of learning a compact representation of the features in a transitional layer. It is a novel partitioning method that initializes a bottleneck in a neural network using the suggested BottleNeck unit. Spatial, channel-wise reduction units and compressor units are used in its architecture on the mobile device to generate a compact representation of the tensor that is transmitted to the cloud. BottleNeck's algorithm comprises three steps, which include training, profiling and selection. For a given number of locations in the network, BottleNeck is placed on an arbitrary selected layer. Different architectures associated with degrees of dimensionality reduction are trained along the channel of spatial dimensions; then, the best partitioning that minimizes the device's energy consumption is chosen. The problem with this architecture, however, is the lack of accuracy in loss, it does not accurately provide the amount of data loss.

To minimize the consumption of energy during task offloading and computation under both the main and edge processing delay limitations, Xianyan Hu et al. [15] proposed a computing architecture that comprises both hybrid edge and central cloud, one macrocell with a Macro-Base-Station (MBS) and several small cells each with a small base station, and a continual algorithm to find a solution to the combinatorial mixed-integer and non-convex optimization problems. In this solution, the authors considered the delay of synchronizing end user's tasks, the end users' tasks are assumed to already be synchronized in the problem formulation. To guarantee the quality of services provided by the edge clouds, the edge processing latency constraints require the corresponding latency to not exceed a targeted threshold. Furthermore, to further reduce the complexity of solving the optimization problem for reducing the total energy consumption of the network during task offloading and computation, massive multiple-input multiple-output technology is applied at the multi-antenna macro-base-station.

An energy-efficient architecture based on service provision was proposed by Hani et al.[43] to improve the quality of service of the handover process in MCC. The proposed architecture implies four layers, the media connectivity layer, the application layer, the Internet protocol multimedia subsystem (IPMS) layer, and the communication layer, and was implemented in C++. The Media Connectivity Layer (MCL) is responsible for connectivity and media related operations and services, it includes the Media Resource Agent (MRA) and Media Resource Function Controller (MRFC). The application layer connects to the IPMS layer to assure data communication and to the cloud computing servers as an enterprise server. The IPMS layer is responsible for offering services such as web browsing, video streaming, videoconferencing, email, the Internet, and handles the registration process used to obtain users' location. This layer also integrates a Call Session Control Function (CSCF) to associate the users identity to the IP address; the function has three parts known as Proxy-CSCF, serving CSCF, and interrogating CSCF. The communication layer carries the data and binds the media layer to the IPMS layer; besides, it includes a Media Gateway Controller Function (MGCF), Media Resource Function Controller (MRFC) and Breakout Gateway Control Function (BGCF). In addition, it includes an energy-efficient detection model to ascertain the energy of nodes when initiating the handoff process. The energy consumed during the handover process is proportional to the distance between the mobile device and its access point and the time required to complete the handover process. Thus, minimizing the distance between the UE and the handover time reduces energy consumption. Also, when the access point is changed, the re-registration and reattachment process necessitate additional energy and the previous energy consumed has to be taken into account in the calculation of the total energy consumed. The proposed architecture is more suitable for mobile phones when initiating the handover process in a cloud computing environment and has not been assessed for potential vulnerabilities yet.

Ren et al. [44] proposed an efficient technique to find the optimal resource allocation solution that minimizes latency in a multi-user mobile edge computation offloading system by developing a sub-gradient algorithm. In this solution, the authors first determined data segmentation methods by considering N mobile devices $\{1, 2, 3, \cdots, N\}$ and a base station BS that links the devices to the cloud, the CPU and edge cloud compression capacity of the CPU V_n^d and V^c respectively, and the compression capacity of each device V_n^c with the following constraint $\sum_{i=1}^{N} V_n^c <= V^c$. Two compression models were first considered, the Multi-Access Model where one time slot is divided into several portions, reducing the data rate of each portion; and Partial Compression Offloading Model where each file can be partitioned in two parts with one part compressed locally and the other in the edge cloud. The proposed algorithm, which is based on the sub-gradient method for similar non-differential convex problems, finds the optimal resource to be allocated with the aim to reduce the weighted sum latency of the compression in all devices.

With the goal of reducing the delay of handling tasks execution and tasks failure of data partitioned based applications, Nguyen et al. [45] proposed a fuzzy based logic mobile edge orchestrator to segment tasks from UEs and associate them to the appropriate edge servers. The proposed framework gets as input the network and resources information such as bandwidth, size of the task being processed, the characteristic of the edge server's virtual machine being used, and the latency sensitivity associated with each task. It also involves a fuzzification step where membership functions are set accordingly to transform the inputs into fuzzy values, and a defuzzification step where fuzzy values are transformed to normal values. The strategy to divide the execution of tasks includes the fact that the orchestrator determines if the task has to be collaboratively processed by the edge and cloud servers or the edge server alone by computing and choosing the environment with the smaller fuzzy values of input parameters, and crisp output value; if the crisp output value is greater than the threshold, the task is executed by the cloud server alone.

4.4 Latency minimization through load balancing and offloading

To reduce high data traffic in edge networks, Zhao et al. [46] proposed a solution based on their Enumeration based Optimal Placement Algorithm (EOPA) and Divide-and-Conquer based Near Optimal Placement Algorithm (DCNOPA) to efficiently distribute virtual machine replica copies (VRCs) of applications to the edge network. In this solution, a graph $G(V, E)$ is used to model the physical edge network. The V in the graph is defined as $\{V = \{v, v = 1, 2, ..., \mid V \mid\}$ and represents the set of edge servers, and E the set of connections between edge servers with the assumption that all mobile users are assigned in the edge network randomly. Also, the assignment of virtual machines is done according to the following constraints: each virtual machine replica of an application can only be associated with one edge server, similarly, each edge server only holds one virtual machine replica of an application. The optimal placement algorithm finds the placement $S' = \{l_{u,s}, \forall u \in V, \forall s \in S\}$ among all potential placements of k VRCs, to obtain a reduced data traffic for each request by considering all potential placement cases for k VRCs, and computing the average data traffic for each placement case. The divide-and-conquer based near optimal placement algorithm divides all edge servers into k clusters and deploys only one VRC for each cluster, thus reducing the original problem of finding k VMs to a problem of determining an efficient placement for one virtual machine replica in each cluster, which reduces its complexity considerably.

Mobile data offloading schemes based on a Finite Horizon Markov Decision Process (FHMDP) to reduce the communication cost for delivering mobile data with different latency sensitivities through several wireless networks were proposed by Dongqing and al.[47], where FHMDP plans data offloading decisions at each decision epoch. In the model, mobile data is initially delivered to one or more device through cellular and Wi-Fi networks. The data being sent from the cloud environment is divided into a sequence of data units, which are predetermined by the mobile network operator. Also, the access point station that carries a copy of the data can transmit it to the user using D2D communication. The approach was embedded in a hybrid offloading algorithm that can support different delay requirements with lower computational complexity. The algorithm computes the optimal policy through three phases: initialization, planning and offloading. The expected number of mobile access points in different locations is calculated in the initialization phase and is used to indicate the availability of D2D action in the planning phase, the offloading action at each decision epoch is determined in the last phase.

Aral, Atakan, et al. [48] proposed an algorithm for distributed data dissemination and replicas across IaaS; it relies on dynamic creation and withdrawal of replicas guided by continuous monitoring of data requests coming from edge nodes of the underlying network. The proposed algorithm uses geographical location of data during the distribution process resulting from the plethora of ordinary data requests that stem from the clients within surroundings. The cost of storing replicas as well as expected delay improvement to make a migration or duplication decision to one of the neighboring nodes is evaluated through the algorithm, which also enables users to handle the balance between cost minimization and delay optimization. Also, a replica discovery method where the important nodes are identified and notified of replica creations or removals is provided by the proposed work. The algorithm is complemented with a replica discovery method where concerned nodes are notified of nearby replicas. On the other hand, experimental results show that communication overhead and miscommunication errors caused by replica placement and discovery are not significant, which is not always true. Also, the proposed solution is not appropriate for real-time systems, which require real-time performance guarantee.

A task scheduling algorithm for MCC based on a heuristic ant colony optimization algorithm was proposed by Wang et al. [49], taking into consideration four types of time constrained tasks, adapting to several MCC elements such as Cloudlet, mobile device cloud and incorporating a variety of objectives including efficient load balancing, minimization of energy consumption, and improvement of reliability and profit. The proposed algorithm is embedded in a system that involves a task tracker, which is responsible for gathering resource consumption and offloaded tasks information and using the algorithm to determine which task should be executed on a given service provider. It considers four phases or models for the resolution of the task scheduling problem. The *task graph* model involves a set of interactive tasks represented by a graph $G = (V, L)$ with V representing tasks nodes and L the relationships between them, with a flow that includes tree structure, independent node, regular mesh structure and linear chain topology. The *communication model* incorporates the channel state determined by the channel gain and classified as good or bad depending on a given threshold, the communication delay defined as the ratio of the length of each task over the channel state. The *execution model* includes mobile execution phase that considers the computational resource consumption and execution time of each task defined by the computing capacity of the device and the task length, and completion time phase that sums up the different execution delays of the task. The *task scheduling model* considers reduction of resource consumption and profit maximization for users. The algorithm is divided into three parts, known as *task selection*, which selects each task to be executed based on the relative pheromone ratio, *service provider selection*, which is responsible for selecting the provider that should execute the selected task based on the load ratio of each provider, and *task scheduling*, which initializes main parameters, and uses the precedent parts of the algorithm to find the best ob- jective function for each task. The main issue associated with this solution is the fact that it is a static algorithm and it is suitable for batch scheduling only.

Li et al. [50] proposed a computation partitioning technique to improve the performance of big data en- vironments in MCC. In this technique, two computation partitions are considered; the first one is responsible for monitoring the changes in resources including bandwidth, CPU, etc., while the other partition deter- mines the computation location where a given task should be executed. The main goal of this methodology is to enhance the application performance on UE by improving the computation partitioning decision. For this, it is important to find an efficient way to solve the single-frame execution time problem, then establish the partitioning scheme for multi-frame execution, as the single-frame execution alone is inefficient. These calculations depend on the network bandwidth and the changes in the environment of the system. The model includes three types of tasks that are local, transferring and cloud tasks. A graph is used to represent the tasks' data flow, and the adjacency matrix of the graph is used to perform task selection. For the single-frame task execution problem, the efficient partitioning scheme is determined by a Genetic Algorithm (GA) due to its strong search capabilities. Additional optimization and adjustments are performed to settle the total execution time of multi-frame data. However, for this solution to be effective, data-frames congestion, instability of data during transfer, and limitation of resources should be considered.

4.5 Solutions for data security and privacy

Qiu, Tie, et al. [51] proposed SIGMM, a machine learning algorithm for spammer identification in industrial MCC. The framework makes use of data, where each user node is classified into one class in the construction process of the model, the data includes the relationship with other users, user's identification, the time-stamped post record, and the activity in the past three months. A Pearson correlation coefficient and Principal Component Analysis (PCA) were employed to characterize different features and model the parameters accurately. SIGMM fits the behavior data of regular users and spammers, in which the behavior data of ordinary users and spammers are mixed by random sampling. However, this solution is not suitable for large networks since the algorithm is based on binary classification, the types of users are varied and complex in large networks and thus more than two categories are required to classify the nodes accurately.

Xiao et al. [11] presented security solutions that apply reinforcement learning techniques to protect edges from spoofing, malware, jamming, and eavesdropping attacks that might occur during data offloading to edges nodes. The radiocommunication channels of edges nodes are vulnerable to attacks launched from the physical layer or Medium Access Control (MAC) layers during data offloading in MCC environment. Most of the solutions include the use of Q-learning to prevent attacks; the main reason is the fact that Q-learning-based security schemes do not require any prior knowledge of the network, they apply the iterative Bellman equation to update Q-values, and only use two parameters, which are the learning rate and the discount factor, to control their learning performance. Nonetheless, security schemes based on Q-learning require exploring all the possible states and pairs of actions before significantly changing the network policies, resulting in a slower reaction in case of an imminent attack.

Nguyen et al. [53] proposed a method based on deep learning to prevent and detect cyberattacks in MCC: a training dataset is used to train the neural networks of the framework that implements the technique offline, then, once the model is ready, it is integrated in the MCC environment to detect and prevent attacks online. The model involves two major phases which are feature analysis and learning process. Feature analysis includes the extraction and examination of abnormal attributes in the dataset to identify traits associated with malicious packets, and dimensionality reduction using the Principal Component Analysis (PCA) technique to remove irrelevant features or attributes that are not needed for the detection of attacks. The learning process comprises three types of layers including the input layer, some hidden layers and the output layer. The features are fed directly to the input layer; then, a Gaussian Binary Restricted Boltzmann Machine (GRBM) is used to convert them into binary codes, which are used in the hidden layers. A series of learning steps are performed to adjust the weights of each layer. However, only theoretical evaluation of the model was performed, even though high accuracy was obtained, the model was not evaluated in a practical and real time environment.

To improve the efficiency of encryption and decryption schemes in MCC and make them suitable for mobile devices, Zhang et al. [52] introduced a system architecture of anonymous attribute-based access control in mobile cloud computing, a decryption method called match-then-decrypt where a matching phase is added before the decryption phase. The technique involves a basic anonymous Cyphertext Policy - Attribute-Based Encryption (CP-ABE) construction and the procurement of security-enhanced extension using the reasonable Canetti–Halevi–Katz technique based on one-time signatures. In Canetti –Halevi –Katz transformation, a test can be made during the decryption process before completing it and the subsequent decryption is completed if and

Table 1 – Qualitative review of different solutions proposed to achieve seamless communication in MCC.

Proposed Approach	Analysis Summary		Shortcoming
Seamless handover & service provision [28, 29, 30, 31, 32]	Technique	Vehicular Computing, V2X, FVC, Edge Selection, etc.	• Mobile vehicles cannot act as fog-enabled devices yet.
	Description	• The proposed solutions are mainly based on fog-enabled vehicles, FVC, and base stations to ensure fast and efficient handover. • Proposed a fog computing platform that enables the allocation and management on the set of computational resources for executing effectively IoT tasks.	
Placement of Edge Servers & Base Stations [29, 37, 38, 39]	Technique	Mixed-integer quadratic programming, K-Means, Caching at edge, Proxy-based and Greedy content placement algorithm, etc.	•Limited to an environment with homogeneous setting. • Do not include load balancing management. • Expensive solutions since caching hardware must be integrated on each edge cloud.
	Description	• The proposed solutions are mainly based on mixed-integer quadratic programming and K-Means algorithms to compute optimal placement locations of edges such that the workloads are balanced and the access delay reduced. • The proxy server is selected based on four parameters, which include the type of host, the state of the host, the hardware performance of the host and the available amount of concurrent connections. • Proposed a homogeneous mobile network with edge caching where the mobile device fetches coded segments directly from candidate SBSs in ascending order of transmission distance, if the requested content is cached.	
Reducing Latency & Energy Consumption, Improvement of handover QoS [15, 40, 41, 42, 43, 44]	Technique	Web worker migration, machine learning, Gradient algorithm, Data segmentation Mixed-integer non linear programming, Gaussian randomization, Subgradient algorithm, etc.	• Solution is very limited. • Mainly suitable for web applications only. • Intelligence in fog computing is still in its infancy, and the assumptions made are not realistic yet. • Lack of accuracy in loss. • Latency in links is not the only major parameter to be considered.
	Description	• A hybrid edge and central cloud computing architecture was proposed, including one macro cell with a Macro Base Station (MBS) and multiple small cells each with an SBS, and an iterative algorithm used to solve the combinatorial mixed-integer and non-convex optimization problems. • Web worker migration techniques and machine learning were proposed to detect user's behaviors, find optimal servers and make efficient offloading decision. • Spatial and channel-wise reduction units were applied to create a compressed representation of the feature tensor which is transmitted to the cloud. • Energy consumed during the handover process can be reduced by computing the minimum distance between the UE and the handover BS. • Subgradient algorithm was applied to compute the minimum latency between links in an edge and perform resource allocation accordingly.	
Data Offloading & Load Balancing [46, 47, 48, 49]	Technique	Finite Horizon Markov Decision Process, Ant Colony Optimization, Divide-and-conquer based near optimal placement algorithms, etc.	• The solution only reduces the complexity of the problem. • Not appropriate for real time systems which require real time performance guarantee. • Static algorithm based and suitable for batch scheduling only.
	Description	• Optimal placement (EOPA) and divide-and-conquer based near optimal placement algorithms (DCNOPA) were proposed to efficiently distribute virtual machine replica copies (VRCs) of applications to the edge network to reduce high data traffic in edge networks.	

		• A Finite Markov Decision Process was proposed to minimize the communication cost for delivering mobile data with different delay sensitivities through multiple wireless networks and manage replicas by monitoring data requests. • A Task tracker approach was proposed that can gather resource consumption and offloaded tasks information and determines which task should be executed on a given service provider.	
Data Security & Privacy [11, 51, 52, 53, 54]	Technique	Machine Learning, cryptography, deep learning, Gaussian Binary Restricted Boltzmann Machine, etc.	• Not suitable for large networks since the algorithm is based on binary classification. • Requires exploring all the possibles states and pairs of actions. • Model not evaluated in practical and real time environment. • Not suitable for file sharing systems.
	Description	• Machine learning algorithms were applied for spammer identification in industrial MCC. • Reinforcement learning techniques, especially Q-learning, were applied to protect edges from spoofing, malware, jamming, and eavesdropping attacks that might occur during data offloading to edges nodes. • Deep learning-based solutions were applied to prevent and detect cyberattacks in MCC online. • Encryption and decryption techniques were proposed to achieve data security. • Match-then-decrypt technique was proposed in which a matching phase is added before the decryption phase, to improve encryption and decryption in security schemes. • A data encryption solution was proposed between mobile devices and private and public clouds environments.	

only if the test passes, which is more suitable for performing matching before decryption. The transformation is applied to obtain a chosen cyphertext (CCA2) secure extension. The whole architecture is embedded in the four algorithms of the anonymous attribute-based access control system known as *Setup*, *KeyGen*, *AnonEncrypt*, *AnonDecrypt*. The matching phase returns the symbol to terminate decryption with overwhelming probability, it ends with the initiation of the next decryption phase; the decryption phase returns the original message. The solution focuses on decryption because the full decryption cost linearly increases with the complexity of access policies.

With the same vision of enhancing data security in an MCC environment, Yang et al. [54] proposed an encryption scheme known as File Remotely keyed Encryption and Data Protection (FREDP)that performs data encryption between mobile devices and private and public cloud environments. In the proposed scheme, the computation resources of private clouds are used to remotely encrypt mobile devices data; however, the encryption key is not shared with the private cloud environment which performs data integrity verification; the encrypted data is encrypted by block then shared with the public cloud to store it. To enforce security in high traffic systems, the mobile devices and private clouds are assumed to have shared authentication key pairs and public keys of each other. User's sensitive information is first encrypted by the mobile device using the private cloud's public key, and sent to the private cloud, which decrypts it using its own private key and performs user's authentication. When the authentication is completed, the mobile device partially computes the cipher text of the remaining data block by block using the private cloud's public key and sends it to the private cloud, which completes the remainder of the encryption and decrypts it using its private key. A data fingerprint is generated for every metadata block, which is sent to the public cloud from the private cloud; the public cloud decrypts the message using its private key and performs data authentication, if the authentication is successful, the public cloud sends back an acknowledgment message. The proposed scheme enhances the confidentiality of the files and the security of the encryption key. However, with this method, only the data owner can access the file which is not suitable for file sharing systems. Table 1 gives a qualitative overview of different solutions with their proposed techniques and their shortcomings.

5. OPEN CHALLENGES AND ISSUES

Most of the challenges existing in the MCC environment today are associated with reducing latency, increasing bandwidth, achieving uninterrupted communication between a mobile device and the cloud environment with intermittent connectivity, assuring constant network availability and heterogeneity, providing data access efficiency and security and privacy during exchange of data, and overcoming constraints associated with limited sources of energy. These challenges can be classified in two categories, MCC infrastructural challenges and mobile devices related constrains.

- *Network Bandwidth*: Adequate network bandwidth is essential for achieving seamless communication in the MCC environment. Unfortunately, in the MCC environment, some wireless networks provide low bandwidth, intermittent signal availability or less reliable transmission, which cause critical issues resulting in quality of service degradation, additional latency and hangs in applications.

- *Network Availability*: To achieve seamless communication in an MCC environment, one should assure that there is a continuous Internet connectivity and exchange of data between the mobile device and the cloud. Unfortunately, network coverage is nonexistent in some areas due to a lack of infrastructure, or interference with other signal blockers. This causes major delays and applications' hangs since most of mobile applications need to be always linked to the cloud from any place to function properly.

- *Heterogeneity Management*: Heterogeneity is defined as the existence of various types of mobile devices, clouds and wireless networks with different hardware, architectures, infrastructure, and technologies. Available edge technologies expected to initiate and facilitate collaboration of heterogeneous computing devices toward unrestricted mobile computing are unfortunately limited, thus making heterogeneity in the MCC environment challenging for achieving seamless communication in MCC as variations of network and its related technologies affect the delivery of cloud services.

- *Latency Reduction*: Reducing latency in MCC networks is critical for achieving seamless service provision, latency constitutes a significant barrier that limits the solutions proposed in MCC and it is not a new problem in MCC. Several different types of solutions are proposed by scholars to reduce latency in MCC but it remains one of the major challenges in the area. The latency of a system in MCC is proportional to the processing time of computational operations, the downloading and offloading time and the rate at which operations are performed in the system. Long WANs are one of the causes of latency in MCC, offloading mobile intensive applications to distant cloud resources, for instance, creates a bottleneck in the network. To reduce interaction latency, proposals such as Cloudlet, MOMCC, and SAMI are proposed to create a proximate cloud to access nearby remote resources, but further advancement to achieve crisper response is required [55].

- *Energy Consumption*: A limited amount of energy in mobile devices is one of the issues encountered in MCC when considering intensive computational solutions to achieve seamless service provision. Developers and operators deployed applications and systems with the goal to conserve mobile battery power; however, resource and data intensive tasks offloaded from the mobile device to the cloud require more energy. Estimating energy efficiency of computation offloading is complex because of the heterogeneity of wireless technologies and infrastructures.

- *Mobility Management*: In the MCC environment, the ability of mobile devices to move around constitutes an obstacle for achieving seamless connectivity to the cloud. Most of the time, intermittent connectivity, unreliable services, degradation of quality of service, and disruption are caused by user equipment mobility. The solutions to this problem involve handover between service areas or base stations and location prediction as the device moves around, to route the data to the intended destination and re-synchronization of service provision if for some reason the device is disconnected from the network. However, handover also comes with other challenges such as handover failure and handover latency, which become important issues when dealing with wireless heterogeneity and intensive traffic applications in MCC; these issues depend as well on choosing the right network based on application requirement, user preferences, services offered and environment. Also, most handover techniques are defined in IP layers, which makes it challenging to properly define the role of each layer handover for achieving low latency.

- *Security and Privacy*: Protection of users' data remains one of the greatest concerns in MCC. Most mobile devices store users' confidential information such as medical records, banking information, location and other personal information, which is shared with cloud infrastructures as some applications use it to provide appropriate services. Storing private information in the cloud environment and accessing it through Internet services and wireless networks make it vulnerable to numerous cyberattackers. Identity provisioning and access management through different environments are a sample of the security keys, which manifest the necessity of secure intercommunication in MCC [55]. Thus, cloud service providers need to increase security by providing strong authentication and authorization methods. For instance, data migration or tasks offloading across multiple clouds should be highly secured as these communication processes involve personally identifiable data. However, adding protection such as data encryption to computational tasks shared on the network increases processing overhead, which causes latency.

6. CONCLUSION

Although MCC already mitigates the computational resources needed by mobile devices to provide services and run applications smoothly, the communication and service provision between the cloud environment the mobile device are sometimes not the best. In this survey, we presented an overview of the MCC environment and investigated different techniques recently proposed in the literature to achieve seamless communication in the MCC environment, by taking into consideration recent technological advances in networking and MCC in general such as the deployment of 5G systems and availability of new techniques in artificial intelligence. We provided a brief overview of solutions in Table 1 with their shortcomings. A lot of progress has been made in the MCC field, especially to achieve reliable and seamless communication between the cloud and the mobile device, but there are still some overwhelming challenges faced by the operators and service providers. Some of these challenges are presented in this paper with the hope that scholars are working to overcome them. We believe this paper will serve as a guide for future research works for achieving seamless data exchange and application offloading in MCC.

REFERENCES

[1] G. Forecast, "Cisco visual networking index: global mobile data traffic forecast update, 2017–2022," *Update*, vol. 2017, p. 2022, 2019.

[2] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, "A survey of mobile cloud computing: architecture, applications, and approaches," *Wireless communications and mobile computing*, vol. 13, no. 18, pp. 1587–1611, 2013.

[3] P. Rost, A. Banchs, I. Berberana, M. Breitbach, M. Doll, H. Droste, C. Mannweiler, M. A. Puente, K. Samdanis, and B. Sayadi, "Mobile network architecture evolution toward 5g," *IEEE Communications Magazine*, vol. 54, no. 5, pp. 84–91, 2016.

[4] M. Othman, S. A. Madani, S. U. Khan, *et al.*, "A survey of mobile cloud computing application models," *IEEE communications surveys & tutorials*, vol. 16, no. 1, pp. 393–413, 2013.

[5] 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.

[6] N. Hassan, K.-L. A. Yau, and C. Wu, "Edge computing in 5g: A review," *IEEE Access*, vol. 7, pp. 127276–127289, 2019.

[7] I. Alghamdi, C. Anagnostopoulos, and D. P. Pezaros, "Time-optimized task offloading decision making in mobile edge computing," in *2019 Wireless Days (WD)*, pp. 1–8, IEEE, 2019.

[8] M. E. Khoda, M. A. Razzaque, A. Almogren, M. M. Hassan, A. Alamri, and A. Alelaiwi, "Efficient computation offloading decision in mobile cloud computing over 5g network," *Mobile Networks and Applications*, vol. 21, no. 5, pp. 777–792, 2016.

[9] X. Chen, "Decentralized computation offloading game for mobile cloud computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 4, pp. 974–983, 2014.

[10] S. Misra, S. Das, M. Khatua, and M. S. Obaidat, "Qos-guaranteed bandwidth shifting and redistribution in mobile cloud environment," *IEEE Transactions on Cloud Computing*, vol. 2, no. 2, pp. 181–193, 2013.

[11] L. Xiao, X. Wan, C. Dai, X. Du, X. Chen, and M. Guizani, "Security in mobile edge caching with reinforcement learning," *IEEE Wireless Communications*, vol. 25, no. 3, pp. 116–122, 2018.

[12] R.-F. Liao, H. Wen, J. Wu, F. Pan, A. Xu, H. Song, F. Xie, Y. Jiang, and M. Cao, "Security enhancement for mobile edge computing through physical layer authentication," *IEEE Access*, vol. 7, pp. 116390–116401, 2019.

[13] A. Aliyu, A. H. Abdullah, O. Kaiwartya, M. Tayyab, and U. M. Joda, "Mobile cloud computing: layered architecture," *2018 Seventh ICT International Student Project Conference*, pp. 1–6, 2018.

[14] J. Ren, G. Yu, Y. He, and G. Y. Li, "Collaborative cloud and edge computing for latency minimization," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 5, pp. 5031–5044, 2019.

[15] X. Hu, L. Wang, K.-K. Wong, M. Tao, Y. Zhang, and Z. Zheng, "Edge and central cloud computing: A perfect pairing for high energy efficiency and low-latency," *IEEE Transactions on Wireless Communications*, vol. 19, no. 2, pp. 1070–1083, 2019.

[16] M. A. Mohammed and N. ȚĂPUȘ, "A novel approach of reducing energy consumption by utilizing enthalpy in mobile cloud computing," *Studies in Informatics and Control*, vol. 26, no. 4, pp. 425–434, 2017.

[17] S. Li, W. Sun, Y. Zhang, and H. Liu, "Reliability analysis for multipath communications in mobile cloud computing architectures," *Wireless Communications and Mobile Computing*, vol. 2018, 2018.

[18] R. B. Bohn, J. Messina, F. Liu, J. Tong, and J. Mao, "Nist cloud computing reference architecture," in *2011 IEEE World Congress on Services*, pp. 594–596, IEEE, 2011.

[19] A. Ionescu, "Resource management in mobile cloud computing," *Informatica Economica*, vol. 19, no. 1, p. 55, 2015.

[20] A. Gani, G. M. Nayeem, M. Shiraz, M. Sookhak, M. Whaiduzzaman, and S. Khan, "A review on interworking and mobility techniques for seamless connectivity in mobile cloud computing," *Journal of Network and Computer Applications*, vol. 43, pp. 84–102, 2014.

[21] T. Shon, J. Cho, K. Han, and H. Choi, "Toward advanced mobile cloud computing for the internet of things: Current issues and future direction," *Mobile Networks and Applications*, vol. 19, no. 3, pp. 404–413, 2014.

[22] F. Rebecchi, M. D. De Amorim, V. Conan, A. Passarella, R. Bruno, and M. Conti, "Data offloading techniques in cellular networks: A survey," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 2, pp. 580–603, 2014.

[23] E. Ahmed, A. Gani, M. K. Khan, R. Buyya, and S. U. Khan, "Seamless application execution in mobile cloud computing: Motivation, taxonomy, and open challenges," *Journal of Network and Computer Applications*, vol. 52, pp. 154–172, 2015.

[24] M. Chiang and T. Zhang, "Fog and iot: An overview of research opportunities," *IEEE Internet of things journal*, vol. 3, no. 6, pp. 854–864, 2016.

[25] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.

[26] K. Akherfi, M. Gerndt, and H. Harroud, "Mobile cloud computing for computation offloading: Issues and challenges," *Applied computing and informatics*, vol. 14, no. 1, pp. 1–16, 2018.

[27] 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*, pp. 1–33, 2020.

[28] K. Wan, D. Tuninetti, M. Ji, and G. Caire, "A novel cache-aided fog-ran architecture," in *2019 IEEE International Symposium on Information Theory (ISIT)*, pp. 2977–2981, IEEE, 2019.

[29] S. Zhang, P. He, K. Suto, P. Yang, L. Zhao, and X. Shen, "Cooperative edge caching in user-centric clustered mobile networks," *IEEE Transactions on Mobile Computing*, vol. 17, no. 8, pp. 1791–1805, 2017.

[30] T. H. Luan, L. Gao, Z. Li, Y. Xiang, G. Wei, and L. Sun, "Fog computing: Focusing on mobile users at the edge," *arXiv preprint arXiv:1502.01815*, 2015.

[31] M. R. Palattella, R. Soua, A. Khelil, and T. Engel, "Fog computing as the key for seamless connectivity handover in future vehicular networks," in *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, pp. 1996–2000, 2019.

[32] A. Kapsalis, P. Kasnesis, I. S. Venieris, D. I. Kaklamani, and C. Z. Patrikakis, "A cooperative fog approach for effective workload balancing," *IEEE Cloud Computing*, vol. 4, no. 2, pp. 36–45, 2017.

[33] M. Sookhak, F. R. Yu, Y. He, H. Talebian, N. Sohrabi Safa, N. Zhao, M. K. Khan, and N. Kumar, "Fog vehicular computing: Augmentation of fog computing using vehicular cloud computing," *IEEE Vehicular Technology Magazine*, vol. 12, no. 3, pp. 55–64, 2017.

[34] R. Bruschi, F. Davoli, P. Lago, A. Lombardo, C. Lombardo, C. Rametta, and G. Schembra, "An sdn/nfv platform for personal cloud services," *IEEE Transactions on Network and Service Management*, vol. 14, no. 4, pp. 1143–1156, 2017.

[35] R. Bruschi, F. Davoli, P. Lago, and J. F. Pajo, "Move with me: Scalably keeping virtual objects close to users on the move," in *2018 IEEE International Conference on Communications (ICC)*, pp. 1–6, 2018.

[36] J. Santa, J. Ortiz, P. J. Fernandez, M. Luis, C. Gomes, J. Oliveira, D. Gomes, R. Sanchez-Iborra, S. Sargento, and A. F. Skarmeta, "Migrate: Mobile device virtualisation through state transfer," *IEEE Access*, vol. 8, pp. 25848–25862, 2020.

[37] S. Wang, Y. Zhao, J. Xu, J. Yuan, and C.-H. Hsu, "Edge server placement in mobile edge computing," *Journal of Parallel and Distributed Computing*, vol. 127, pp. 160–168, 2019.

[38] D. Lee, H. Lee, D. Park, and Y.-S. Jeong, "Proxy based seamless connection management method in mobile cloud computing," *Cluster computing*, vol. 16, no. 4, pp. 733–744, 2013.

[39] Y. Guo, S. Wang, A. Zhou, J. Xu, J. Yuan, and C.-H. Hsu, "User allocation-aware edge cloud placement in mobile edge computing," *Software: Practice and Experience*, vol. 50, no. 5, pp. 489–502, 2020.

[40] H.-J. Jeong, C. H. Shin, K. Y. Shin, H.-J. Lee, and S.-M. Moon, "Seamless offloading of web app computations from mobile device to edge clouds via html5 web worker migration," in *Proceedings of the ACM Symposium on Cloud Computing*, pp. 38–49, 2019.

[41] Q. D. La, M. V. Ngo, T. Q. Dinh, T. Q. Quek, and H. Shin, "Enabling intelligence in fog computing to achieve energy and latency reduction," *Digital Communications and Networks*, vol. 5, no. 1, pp. 3–9, 2019.

[42] A. Erfan Eshratifar, A. Esmaili, and M. Pedram, "Bottlenet: A deep learning architecture for intelligent mobile cloud computing services," *arXiv e-prints*, pp. arXiv–1902, 2019.

[43] Q. B. Hani and J. P. Dichter, "Energy-efficient service-oriented architecture for mobile cloud handover," *Journal of Cloud Computing*, vol. 6, no. 1, pp. 1–13, 2017.

[44] J. Ren, G. Yu, Y. Cai, Y. He, and F. Qu, "Partial offloading for latency minimization in mobile-edge computing," in *GLOBECOM 2017-2017 IEEE Global Communications Conference*, pp. 1–6, IEEE, 2017.

[45] V. Nguyen, T. T. Khanh, T. Z. Oo, N. H. Tran, E.-N. Huh, and C. S. Hong, "Latency minimization in a fuzzy-based mobile edge orchestrator for iot applications," *IEEE Communications Letters*, vol. 25, no. 1, pp. 84–88, 2020.

[46] L. Zhao, J. Liu, Y. Shi, W. Sun, and H. Guo, "Optimal placement of virtual machines in mobile edge computing," in *GLOBECOM 2017-2017 IEEE Global Communications Conference*, pp. 1–6, IEEE, 2017.

[47] D. Liu, L. Khoukhi, and A. Hafid, "Prediction-based mobile data offloading in mobile cloud computing," *IEEE Transactions on Wireless Communications*, vol. 17, no. 7, pp. 4660–4673, 2018.

[48] A. Aral and T. Ovatman, "A decentralized replica placement algorithm for edge computing," *IEEE transactions on network and service management*, vol. 15, no. 2, pp. 516–529, 2018.

[49] T. Wang, X. Wei, C. Tang, and J. Fan, "Efficient multi-tasks scheduling algorithm in mobile cloud computing with time constraints," *Peer-to-Peer Networking and Applications*, vol. 11, no. 4, pp. 793–807, 2018.

[50] J. Li, L. Huang, Y. Zhou, S. He, and Z. Ming, "Computation partitioning for mobile cloud computing in a big data environment," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 4, pp. 2009–2018, 2017.

[51] T. Qiu, H. Wang, K. Li, H. Ning, A. K. Sangaiah, and B. Chen, "Sigmm: A novel machine learning algorithm for spammer identification in industrial mobile cloud computing," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 4, pp. 2349–2359, 2018.

[52] Y. Zhang, X. Chen, J. Li, D. S. Wong, H. Li, and I. You, "Ensuring attribute privacy protection and fast decryption for outsourced data security in mobile cloud computing," *Information Sciences*, vol. 379, pp. 42–61, 2017.

[53] K. K. Nguyen, D. T. Hoang, D. Niyato, P. Wang, D. Nguyen, and E. Dutkiewicz, "Cyberattack detection in mobile cloud computing: A deep learning approach," in *2018 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6, IEEE, 2018.

[54] L. Yang, Z. Han, Z. Huang, and J. Ma, "A remotely keyed file encryption scheme under mobile cloud computing," *Journal of Network and Computer Applications*, vol. 106, pp. 90–99, 2018.

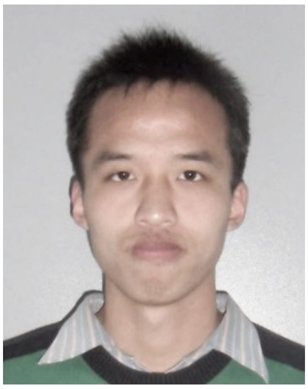
[55] Z. Sanaei, S. Abolfazli, A. Gani, and R. Buyya, "Heterogeneity in mobile cloud computing: taxonomy and open challenges," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 369–392, 2013.

AUTHORS

Pagoui Lagabka Constant received his bachelor degree in electrical engineering technology from Kennesaw State University, Georgia, USA, in 2017. He is currently pursuing the Master of Science degree in computer science at Kennesaw State University. He is also a Graduate Research Assistant, his research interests include artificial intelligence, mobile edge computing, and software defined networks.

Ahyoung Lee received her M.S., and Ph.D. degrees in computer science and engineering from the University of Colorado, Denver, in 2006 and 2011, respectively. She was a postdoctoral fellow at Georgia Institute of Technology in the BWN Lab under the supervision of Prof. Ian F. Akyildiz with a research project focused on Software Defined Networking (SDN) from 2013 to 2015. Currently, she is an Assistant Professor with the Department of Computer Science at the Kennesaw State University. Her current research interest focuses on modeling and analysis with applications in SDN, mobile wireless network, cyber-physical systems, sensor networks, future Internet architecture for improving Big Data centers, Internet of Things (IoT), and Internet-centric technologies in the cloud for network management.

Donghyun Kim received his Ph.D. degree in Computer Science from the University of Texas at Dallas, Richardson, TX, USA in May 2010. He received a Master of Science degree in Computer Science and Engineering from Hanyang University, South Korea in Feb. 2005 and B.S. degree in Electronic and Computer Engineering from Hanyang University, Ansan, South Korea in Feb. 2003. Dr. Kim is an Assistant Professor in the Department of Computer Science at Georgia State University (GSU), Atlanta, GA, USA since August 2020. Before joining GSU, Dr. Kim was an Assistant/Associate Professor (July 2016 - June 2020) in the Department of Computer Science at Kennesaw State University (KSU), Marietta, GA, USA and an Assistant Professor in the Department of Mathematics and Computer Science at North Carolina Central University (NCCU), Durham, NC, USA (August 2010 - June 2016). He is serving as an Associate Editor of several well-known journals including IEEE Access and PeerJ Computer Science, and has served as a Guest Editor of many prestigious journals, most notably Theoretical Computer Science, Journal of Combinatorial Optimization, and IEEE Transactions on Network Science and Engineering. He has also served as a TPC Chair for several international conferences, most recently IEEE IPCCC 2021. Dr. Kim is a Senior Member of IEEE and ACM.



Kun Suo received the BS degree in software engineering from the Nanjing University, China, in 2012, and Ph.D. degree from the University of Texas at Arlington in 2019. He is currently an Assistant Professor in the Department of Computer Science at Kennesaw State University. His research interests include the areas of cloud computing, virtualization, operating systems, Java virtual machines, and software defined networks. He is a member of the IEEE.