ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

OFFLOADING DECISION WITH MOBILITY-AWARE FOR MOBILE EDGE COMPUTING IN 5G NETWORKS

M.Sc. THESIS

Saeid JAHANDAR BONAB

Department of Electronic and Communication Engineering

Telecommunication Engineering Programme

MAY 2021

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

OFFLOADING DECISION WITH MOBILITY-AWARE FOR MOBILE EDGE COMPUTING IN 5G NETWORKS

M.Sc. THESIS

Saeid JAHANDAR BONAB
(504181331)

Department of Electronic and Communication Engineering

Telecommunication Engineering Programme

Thesis Advisor: Prof. Dr. Mustafa ERGEN

MAY 2021

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

5G ŞEBEKESINDE MOBİL KENAR BİLGİ İŞLEM İÇİN MOBİLİTE BİLİNCİ İLE AKTARMA KARARLARI

YÜKSEK LİSANS TEZİ

Saeid JAHANDAR BONAB
(504181331)

Elektronik ve Haberleşme Mühendisliği Anabilim Dalı

Haberleşme Mühendisliği Programı

Tez Danışmanı: Prof. Dr. Mustafa ERGEN

MAYIS 2021

Saeid JAHANDAR BONAB, **a M.Sc.** student of ITU **Graduate School** student ID 504181331, successfully defended the **thesis** entitled "**OFFLOADING DECISION WITH MOBILITY-AWARE FOR MOBILE EDGE COMPUTING IN 5G NETWORKS**", which he/she prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor :	**Prof. Dr. Mustafa ERGEN** Istanbul Technical University	.....
Jury Members :	**Prof. Dr. Nihat KABAOĞLU** Istanbul Medeniyet University	.....
	Asst. Prof. Semiha TEDIK BAŞARAN Istanbul Technical University	.....
		.....

Date of Submission : 16 APRIL 2021
Date of Defense : 11 MAY 2021

To my family,

FOREWORD

This thesis is forwarded to all my teachers from whom I learned. Special appreciation goes to my advisor, Prof. Dr. Mustafa ERGEN for his advice and support. I would like to thank members of Istanbul Technical University Wireless Communication Research Group (ITU-WRG) for the pleasant atmosphere. I am grateful to Dr. Ibraheem SHAYEA for his supports.
I also acknowledge The Scientific and Technological Research Council of Turkey (TUBITAK) which supported this thesis under the project BIDEB-2232 118C276.
I am forever thankful and indebted to my family for their encouragement and sacrifice.

MAY 2021 Saeid JAHANDAR BONAB

TABLE OF CONTENTS

Page

ABBREVIATIONS

3GPP : Third Generation Partnership Project
5G : Fifth Generation of Mobile Networks
AMF : Access and Mobility Function
BS : Base Station
ETSI : European Telecommunications Standards Institute
Het-MEC : Heterogeneous MEC
HO : Handover
IoT : Internet of Things
ITU : International Telecommunication Union
MCC : Mobile Cloud Computing
MDP : Markov Decision Process
MEC : Mobile Edge Computing
QoE : Quality of Experience
RAN : Radio Access Network
RSU : Roadside Unit
SBA : Service Based Architecture
UDN : Ultra Dense Network
UE : User Equipment
VEC : Vehicular Edge Computing

SYMBOLS

c_m	: Computation intensity of each task
d_m	: Input data size
d_{sub}	: Subtask size
F_n	: Available CPU frequency on BS
$f_{m,n}$	: Available computation CPU for each task
$h_{m,n}$	: Channel gain from UE to BS
$\mathcal{N}_{\sigma^2}$	: Noise power
p_m	: UE transmission power
R	: Radius of the BS coverage area
t_m	: Computation deadline of each task
ω	: Channel bandwidth
τ_m^h	: One-time handover delay

LIST OF TABLES

Page

LIST OF FIGURES

Page

OFFLOADING DECISION WITH MOBILITY-AWARE FOR MOBILE EDGE COMPUTING IN 5G NETWORKS

SUMMARY

Mobile Edge Computing (MEC) is a key technology in the Fifth Generations (5G) of Mobile Networks. MEC optimizes communication and computation resources by hosting the application process close to the user equipment (UE) in network edges. This will efficiently enhance communication reliability and stability while reducing latency. The key characteristics of MEC are its ultra-low latency response and real-time applications in emerging 5G networks.

In 2017, the European Telecommunications Standards Institute (ETSI), changed the name mobile edge computing to multi-access edge computing to address non-cellular operators as well. As result, MEC servers can be deployed with Radio Access Network (RAN), Base stations (BSs), Wi-Fi access points, and fixed connections. The 3GPP includes MEC technology in the 5G network with the technical specification 3GPP TS 23.051.

MEC is not only a tool to improve the efficiency of current applications, but also is a key driver for new applications. For an instance, considering an autonomous vehicular network, MEC can be used to share information with roadside units (RSU), other connected cars, and pedestrians without involving any cloud servers.

Due to the low latency of MEC, it is particle to develop real-time artificial intelligence algorithms to enable autonomous vehicle function. Today, IoT devices and sensors collect big data and most have limited computation resources. Since these devices are all becoming smarter, they require storage, computation resources, and bandwidth, which all could be developed using MEC at the edge of the network.

In MEC, task offloading refers to sending computational tasks to the MEC server for processing. Once the MEC executes the received task, it is responsible for sending back the results to the appropriate user.

Considering multiple users within the coverage region of a MEC server, one of the main challenges is how to allocate the MEC resources appropriately. Once the computational resource is allocated for a user, it is going to decide whether to offload the task or execute it locally. Moreover, since there are multiple BSs for a single user, deciding what is the optimum BS for offloading is a research topic in many MEC-related articles. The most common solution is to develop a utility cost function and based on optimization methods, try to find the optimum solution. However, considering resource allocation, communication, computation, and user mobility leads us to analyze joint optimization problems. Most of these joint problems are non-convex and NP-hard to solve.

The second main challenge in MEC-enabled 5G networks is that MEC servers are distributed within the ultra-dense network. Hence, it is an issue to manage user

mobility within small-scale coverage of the MEC server. Most exciting studies in MEC are often neglecting the mobility of users and assume that they are constant due to complexity. However, in this research, the mobility of users is taken into account. Since the mobility of users within ultra-dense MEC coverage causes frequent handover, in our problem formulation the handover cost is included.

Handover acquires once the user leaves the coverage cell of the MEC server during the task processing. In order to support service continuity, the host MEC system should reallocate the task to the targeted MEC server. Therefore, handover results in additional costs which should be considered in the problem formulation.

In this study, our purpose is to consider user mobility while having optimum offloading decisions. During the task offloading if users move from the coverage region of a BS, it will cause handover. So, handover time cost has been considered in the optimization function. Although some related researches are also done focusing on mobility, handover cost, and task migration, the contribution of this research is to choose optimum parameters in optimization function and improve the performance evaluation using online algorithms.

In this study, it assumed that the upcoming future tasks are unknown and online task offloading decisions are considered. As a result, the proposed methods are not model-based and could be implemented in any mobility scenarios. Generally, two scenarios and one solution for each are considered. In the first one, called the online UE-BS algorithm, the users have both user-side and BS-side information. However, in the second scenario, BS-learning users only have user-side information.

In the first algorithm, since the BS information is available, it is possible to calculate the optimum BS for offloading. As a result, within this method, there would be no handover which makes the performance close to the optimal offline solution. However, due to the limited energy budget, we need to ensure that the users' total energy will not exceed its budget.

In the second algorithm, the BS-side information is not available. This means the channel gain available computing CPUs are unknown the users need to learn time and energy cost throughout the observation. Based on these observations the users find optimum BS. However, due to variance in observation, this might lead to sub-optimum BS selection. Selecting the sub-optimal BS would increase the total cost and would cause handover. Therefore, it is essential to choose optimum controlling parameters in the optimization function.

The simulations are conducted in MATLAB. In the result section, we compare our proposed algorithm with other related research studies. Also, to evaluate the performance it is compared with the optimum offline solution and two baseline scenarios. The results indicate that in case the BS-side information is known our proposed method would have near to optimum performance. Furthermore, for the scenario which is based on observation and learning, the re sults show the loss in performance due to additional costs such as handover time cost. Moreover, in terms of the energy budget, our proposed two methods have near to optimum solution comparing with a related article.

The remainder of this thesis is organized as follows. Chapter 1 provides the introduction on the subject, motivation of the study, and purpose of the thesis. Chapter 2 discusses the background on the initialization of MEC systems, The integration of

MEC with 5G networks, Mobility Management in MEC, and challenges. Moreover, the insight literature review is included. Chapter 3, provides the system model, problem formulation, and proposed online algorithms. The simulation setting and numerical results are included in chapter 4. Finally, chapter 5 presents the conclusion and recommendation of this study.

5G ŞEBEKESINDE MOBİL KENAR BİLGİ İŞLEM İÇİN MOBİLİTE BİLİNCİ İLE AKTARMA KARARLARI

ÖZET

Mobile Edge Computing (MEC), 5. nesil (5G) telsiz haberleşmede temel bir teknolojidir. MEC, uygulama sürecini ağ uçlarında kullanıcı ekipmanına (UE) yakın barındırarak iletişim ve hesaplama kaynaklarını optimize eder. Bu, gecikmeyi azaltırken iletişim güvenilirliğini ve kararlılığını verimli bir şekilde artıracaktır. MEC'nin temel özellikleri, çok düşük gecikmeli yanıtı ve yaygınlaşmaya başlayalan 5G ağlarındaki gerçek zamanlı uygulamalarıdır.

2017 yılında, hücresel olmayan operatörlere de hitap etmek için Avrupa Telekomünikasyon Standartları Enstitüsü (ETSI) tarafından mobil uç bilgi işlem terimi çoklu erişim uç bilgi işlem olarak değiştirildi. Sonuç olarak, MEC sunucuları radyo erişim ağı (RAN), baz istasyonları (BS'ler), Wi-Fi erişim noktaları ve sabit bağlantılar kullanıklarak yaygınlaştırılabilir. 3GPP TS 23.051 teknik özellikli 5G şebekesi de MEC teknolojisini içerir.

MEC, yalnızca mevcut uygulamaların verimliliğini artıran bir araç değil, aynı zamanda yeni uygulamalar için de kiritik bir rol oynamaktadır. Örneğin, otonom bir araç ağı düşünüldüğünde, MEC, herhangi bir bulut sunucusunu dahil etmeden yol kenarı birimleri (RSU), diğer bağlı arabalar ve yayalarla bilgi paylaşmak için kullanılabilir.

MEC sağladığı düşük gecikme sayesinde otonom araç fonksiyonlarının ihtiyacı olan gerçek zamanlı yapay zeka algoritmalarının geliştirilmesinde katkı sağlamaktadır. Günümüzde sınırlı hesaplama kapasitesine sahip olan IoT cihazları çok fazla veri toplamaktadır. Bu IoT cihazlar zamanla daha akıllı hale geldiği için depolama, bant genişliği ve işlemci gücüne ihtiyaç duymaktadırlar. Bu ihtiyaçları ağ uçlarında kullanılacak olan MEC teknolojisi ile karşılamak mümkündür.

MEC'de görev aktarımı (offloading), işlem için MEC sunucusuna hesaplama görevlerini göndermeyi ifade eder. MEC alınan görevi yerine getirdikten sonra sonuçları uygun kullanıcıya geri göndermekle sorumludur.

Bir MEC sunucusunun kapsama alanındaki birden fazla kullanıcıyı düşündüğümüzde, ana zorluklardan biri MEC kaynaklarının uygun şekilde nasıl paylaştırılacağıdır. Hesaplama kaynağı tahsis edildikten sonra, kullanıcı görevi aktarma veya yerel olarak yürütmeye karar verecektir. Ayrıca, tek bir kullanıcı için birden fazla BS olduğundan, görev aktarma için en uygun BS'nin hangisi olduğuna karar vermek, MEC ile ilgili birçok makalede araştırılan bir konudur. En yaygın çözüm, bir fayda maliyet fonksiyonu geliştirmek ve optimizasyon yöntemlerini kullanarak en uygun çözümü bulmaya çalışmaktır. Bununla birlikte, kaynak tahsisi, iletişim, hesaplama ve kullanıcı hareketliliği düşünüldüğünde, ortak optimizasyon problemlerini analiz etmemiz gerekmektedir. Bu ortak problemlerin çoğu konveks değildir ve (non-deterministic polynomial) NP-hard problemlerdir.

MEC özellikli 5G ağlarındaki diğer bir ana zorluk, MEC sunucularının çok yoğun şebekelerde dağıtılmasıdır. Bu nedenle, MEC sunucusunun küçük ölçekli kapsama alanı içinde kullanıcı hareketliliğini yönetmek bir sorundur. MEC'deki heyecan verici çalışmaların çoğu, kullanıcıların hareketliliğini ihmal etmekte ve karmaşıklık nedeniyle sabit olduklarını varsaymaktadır. Ancak bu araştırmada kullanıcıların hareketliliği dikkate alınmıştır. Kullanıcıların çok yoğun şebekelerde bulunan MEC kapsama alanı içindeki hareketliliği sık sık geçişe neden olduğundan, problem formülasyonumuza geçiş (handover) maliyeti dahil edilmiştir.

Geçiş (handover), kullanıcının aktif olduğu durumda kapsama alanında bulunduğu bir MEC hücresinden diğer bir MEC hücresine geçmesi olayına denir. Hizmet sürekliliğini desteklemek için, mevcutta hizmet veren MEC sunucusu görevini geçiş yapılacak olan MEC sunucusuna göndermelidir. Bu nedenle geçiş, problem formülasyonunda dikkate alınması gereken ek maliyetlerle sonuçlanır.

Bu çalışmada amacımız, optimum görev aktarma kararlarını alırken kullanıcı hareketliliğini göz önünde bulundurmaktır. Görev aktarma sırasında, eğer kullanıcılar bir BS'nin kapsama bölgesinden uzaklaşırsa, bu durum geçişe neden olur. Dolayısıyla, optimizasyon fonksiyonunda geçiş süresi maliyeti dikkate alınmıştır. Literatürde hareketlilik, geçiş maliyeti ve görev geçişine odaklanan bazı çalışmalar olmasına rağmen, bu tezin katkısı optimizasyon işlevinde optimum parametreleri seçmek ve çevrimiçi algoritmalar kullanarak performans değerlendirmesini iyileştirmektir.

Bu çalışmada, gelecekteki görevlerin bilinmediği varsayılmış ve çevrimiçi görev aktarma kararları dikkate alınmıştır. Sonuç olarak, önerilen yöntemler model tabanlı değildir ve herhangi bir hareketlilik senaryosunda uygulanabilir. Genel olarak, iki senaryo ve her bir senaryo için bir çözüm ele alındı. Çevrimiçi UE-BS algoritması olarak adlandırılan ilk senaryoda, kullanıcılar hem kullanıcı tarafı hem de BS tarafı bilgisine sahiptir. Bununla birlikte, BS-Learning adlı ikinci senaryoda, kullanıcılar yalnızca kullanıcı tarafı bilgilerine sahiptir.

İlk algoritmada, BS bilgisi mevcut olduğundan, aktarma için optimum BS'yi hesaplamak mümkündür. Sonuç olarak, bu yöntemde, performansı optimum çevrimdışı çözüme yaklaştıran bir geçiş (handover) olmayacaktır. Ancak, sınırlı enerji bütçesi nedeniyle, kullanıcıların toplam enerji bütçelerini aşmamasını sağlamamız gerekmektedir.

İkinci algoritmada, BS-tarafı bilgisi mevcut değildir. Bu, kanal kazancının mevcut olduğu bilgi işlem CPU'larının bilinmediği ve kullanıcıların gözlem boyunca zaman ve enerji maliyetini öğrenmesi gerektiği anlamına gelir. Bu gözlemlere dayanarak kullanıcılar optimum BS'yi bulur. Bununla birlikte, gözlemdeki farklılıklar optimum olmayan BS seçimine yol açabilir. Optimum olmayan BS'nin seçilmesi toplam maliyeti artıracak ve geçişe (handover) sebeb olacaktır. Bu nedenle, optimizasyon fonksiyonunda optimum kontrol parametrelerinin seçilmesi önemlidir.

Bu tezde sunulan simülasyonlar MATLAB'da yapılmıştır. Sonuç bölümünde, önerdiğimiz algoritmayı diğer ilgili araştırma çalışmaları ile karşılaştırmaktayız. Ayrıca, performansı değerlendirmek için optimum çevrimdışı çözüm ve iki temel senaryo ile de karşılaştırılmaktadır. Sonuçlar, BS tarafı bilgisinin bilinmesi durumunda önerilen yöntemimizin optimum performansa yakın olacağını göstermektedir. Ayrıca gözlem ve öğrenmeye dayalı senaryo için sonuçlar, geçiş süresi maliyeti gibi ek maliyetler nedeniyle performans kaybını göstermektedir. Ayrıca, enerji bütçesi

açısından, önerilen iki yöntemimiz, ilgili bir makaleye kıyasla optimum çözüme yakındır.

Bu tezin geri kalanı aşağıdaki şekilde düzenlenmiştir. Bölüm 1'de konuya giriş yapılmış, çalışmanın motivasyonu ve tezin amacı verilmektedir. Bölüm 2'de, MEC sistemlerinin başlatılmasına ilişkin arka plan, MEC'in 5G ağları ile entegrasyonu, MEC'de mobilite yönetimi ve zorlukları tartışmaktadır. Bunlara ek olarak, literatür taraması verilmiştir. Bölüm 3'te, sistem modeli, problem formülasyonu ve önerilen çevrimiçi algoritmalar sağlar. Simülasyon parametreleri ve sayısal sonuçlar 4. bölümde yer almaktadır. Son olarak, Bölüm 5'te bu çalışmanın sonucu ve tavsiyeler sunulmaktadır.

1. INTRODUCTION

1.1 Motivation

Of late, the Cisco white paper has indicated that the overall annual data growth will be more than 25% between 2018 and 2023. The number of connected devices will also increase from 18.4 billion in 2018 to 29.3 billion by 2023. However, most connected devices have limited computation and storage resources to fulfil new application requirements [1]. The Internet of Things (IoT) devices, wearable gadgets, vehicles and mobile phones are all becoming smarter, requiring computation-intensive, latency-aware and bandwidth-rich resources. It is difficult to support these application requirements in limited local resources. Therefore, to overcome storage and computation limitations, Mobile Cloud Computing (MCC) was proposed [2]. By offloading computational tasks to a centralised MCC, we are able to achieve remotely unlimited storage and computation resources in centralised clouds. However, this causes considerable issues. First, MCC is not useful for widely-distributed users due to the centralisation of all services. Second, the vast distances between User Equipment (UE) and MCC clouds can cause tremendous latency. Third, concentrating data in a central cloud causes privacy and security concerns by making them vulnerable to attacks during data offloading [3].

In MCC, large amounts of data will be transmitted through networks, resulting in network congestion and bandwidth constraints. With the consideration of MCC limitations, the European Telecommunications Standards Institute (ETSI) has introduced Multi-access Edge Computing (MEC) as a key technology for fifth generation mobile networks (5G) [4]. *The aim of MEC is to unite the telecommunication and IT cloud services to provide cloud-computing capabilities within radio access networks in the close vicinity of mobile users* [5].

1.2 Purpose of Thesis

Managing mobility is challenging since MEC servers are highly densified in 5G and 6G networks. User mobility can cause frequent handovers within ultra-dense and small-scale MEC server coverage [6]. During task offloading where users move across the coverage area of various MEC servers, the UEs may also receive their computation results. Therefore, we need to design an optimum offloading solution. In MEC technology, having optimum offloading decisions will affect the system's performance. Most existing studies have assumed that users are stationary during task offloading and therefore, neglected the mobility of UEs due to the complexity of optimisation problems [7]. In practice however, the mobility of users should be considered, especially for high-speed vehicular applications. The purpose of the thesis is to manage the mobility of UEs with MEC systems.

To offload computational tasks into the targeted Base Station (BS), it is common to make decisions based on signal quality or distance between UEs and BS when considering high-speed vehicles [8]. However, this causes frequent handovers due to short-range data transmission and high mobility. Frequent handovers not only cause transmission interruption, but also increase task migration delays among MEC servers. Once the vehicle leaves the serving MEC host during task computation, the task migration is required between the MEC host and the targeted MEC server. Therefore, one main challenge in the MEC framework is choosing the optimum MEC server based on user movement [9]. In this study, the aim is not only to select optimum MEC servers but also is to choose optimum parameters to avoid frequent handovers.

Low latency is a key factor in the 5G and 6G networks. However, handover could result in large time cost due to service migration, signaling overheads, and handover measurement process. Moreover, comparing with the 5G networks, 6G networks require higher frequency bandwidth and ultra-low latency which makes it important to consider the handover cost of MEC systems in the deployment of 6G networks. As a result, some studies take handover time cost in the problem formulation [10]. The main contribution of the thesis is to consider handover cost in the problem formulation alongside transmission and computation delays as well as energy consumption.

1.3 Hypothesis

The hypothesis and contributions of the thesis are:

It is assumed that the limited energy budget is assigned to each user. To control energy consumption and minimize time cost simultaneously, in chapter 3, the problem is formulated as a trade-off between the time cost and energy cost. Later on, in the simulation, we investigate the effect of controlling parameters and show how to choose optimal parameters considering both energy consumption and delay.

It is also assumed that future upcoming tasks for MEC systems are unknown and users have mobility. Hence, in chapter 3, the online task offloading decisions is introduced. To tackle the mobility effect, two scenarios are considered in which, one of them has handover cost. Later on, in chapter 4, the simulation results illustrate our proposed algorithms have performance close to optimum offline solutions.

2. LITERATURE REVIEW

2.1 Background

Most UEs are resource-limited due to communication and computation requirements. Although new and improved hardware are present, UEs cannot provide the required resource for their application. IoT devices that generate massive data require appropriate computational and storage resources [11]. One solution is to transmit computation-sensitive tasks to centralised clouds. However, this results in massive delays, network congestion and bandwidth requirement. In 2009, Satyanarayanan et al. proposed a cluster of computers, named cloudlets, to be distributed in network edges [12]. Similar to the Wi-Fi connection, cloudlets cannot access users due to long distances. Moreover, cloudlets can only use one Wi-Fi or cellular network simultaneously. Later in 2012, Cisco introduced fog computing which is a cloud that extends to the edge of the network from the network core. The main drawback of fog nodes is that they rely on central clouds to manage them. Additionally, Cloud Radio Access Network (C-RAN) requires massive information exchange between distributed Remote Radio Heads (RRHs) and the centralised Baseband Unit (BBU) [13]. Therefore, in 2014, MEC was initiated to support C-RAN and bring computation and communication resources into the network edges [5].

2.1.1 Mobile edge computing

Cloud computing is an on-demand access to data storage and computing power that mainly refers to data centres available to customers over the Internet. Cloudlets were defined in [12] as reliable and resourceful clusters of computers that are available and close to mobile users, allowing them to connect to the Internet. The drawback of cloudlets is that they are only available at short distances and users must interchange between cloud and cloudlets in case cloudlets are unavailable.

Fog computing was introduced as an extension of the cloud from the core to the edge of the network. Fog nodes are capable of computing and processing collected data from

users at network edges, resulting in reduced latency and network bottlenecks [14]. Fog computing nodes are available in large numbers and are broadly distributed. The main distinction between fog computing and cloud computing is that cloud provides higher computing resources at high power in a centralised unit, whereas fog nodes have a moderate computational resource with low power consumption and are geographically distributed. Unlike centralised clouds, fog nodes require the support of cloud centres to manage themselves. Both cloudlets and fog computing nodes are segregated within the mobile network and are generally owned by private companies [13]. Therefore, it is difficult to provide mobile users with higher Quality of Experience (QoE).

Mobile computing enables UEs to transmit data using a wireless channel without requiring a fixed physical link between the UE and the infrastructure. The term *mobile cloud computing (MCC)* refers to a combination of cloud computing, mobile computing and wireless communication. MCC provides computation and storage resources away from mobile users [15].

Edge computing is a technology that allows computation to be conducted at network edges. Edge is defined as anything between UEs and the data centre or cloud. Therefore, edge computing brings computation close to data sources. Edge computing focuses more on the UE side, while fog computing concentrates on infrastructure. However, the definitions may be used interchangeably [16]. The advantages of edge computing are as follows:

- Decreases end-to-end latency via caching, service localisation, etc.
- Increases bandwidth for multimedia, Unmanned Aerial Vehicles (UAV), surveillance and video monitoring
- Provides computation-rich resources at the network edge
- Enables real-time edge intelligence and edge AI
- Increases flexibility and scalability to create on-demand virtual networks using network slicing

In 2014, *Mobile Edge Computing (MEC)* was initiated in the ETSI Mobile Edge Computing Industry Specification Group (MEC ISG). *MEC provides IT and cloud*

computing capabilities within Radio Access Network (RAN) in close proximity of users [5]. Edge computing simply brings processing and intelligence closer to users, which is commonly employed in IoT. MEC also provides RAN in the network edge, offering a service environment that results in lower latency, real-time access to RAN information and high bandwidth. Although cloud access radio network (C-RAN) reduces costs and power consumption, C-RAN requires massive radio information exchange [17].

As MEC research progressed, the term 'mobile edge computing' excludes the several access points constructing the network edge. In 2017, ETSI changed the name 'mobile edge computing' to 'multi-access edge computing' [4] to better reflect non-cellular operators' requirements as well as combine MEC with HetNet such as cellular networks, Wi-Fi and fixed connection [13]. Therefore, MEC servers can be deployed with RAN or other network elements such as BSs (e.g., 4G, 5G and 6G), Wi-Fi access points and fixed connections, as shown in Figure 2.1.

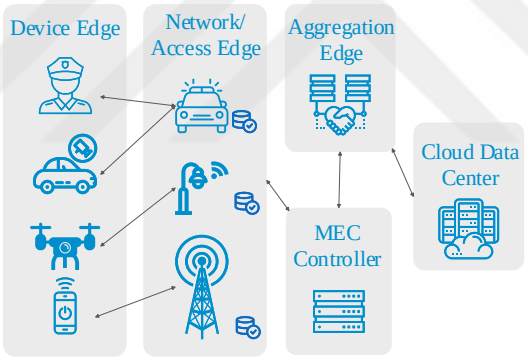


Figure 2.1 : From cloud to the network edges.

2.1.2 Integration of MEC in the 5G network

MEC plays an important role as a Key Performance Indicator (KPI) for the 5G network due to low latency and bandwidth efficiency. The 5G network is a Service Based Architecture (SBA) where services interact between various network functions, network virtualisation and software-defined networks [4]. The 3GPP adds MEC technology in the 5G network with technical specifications; 3GPP TS 23.051 [18]. Accordingly, the 3GPP 5G system allows the MEC system to interact in routing traffic and control policies.

Recently, in [4], the 3GPP clarified the deployment of MEC in 5G, as shown in Figure 2.2. The 5G SBA is shown on the left, while the MEC system architecture is on the right. The network functions are defined as follows:

1. *Access and Mobility Function (AMF)*: The AMF is responsible for mobility management as a centralised function in the 5G network. The AMF handles the RAN control plane, Non-Access Stratum (NAS) procedure, registration, reachability and connection management.

2. *Session Management Function (SMF)*: SMF is responsible for session management, IP allocation, Dynamic Host Configuration Protocol (DHCP) service, control of User Plane Function (UPF) and traffic configuration.

3. *Network Slice Selection Function (NSFF)*: Allocates network slices for AMF in order to provide them to users.

4. *Network Repository Function (NRF)*: All network functions are registered in NRF, while all services are registered locally in MEC. To interact with these functions, an established authentication should be present between them. The lists of available services are located in NRF. Some services are accessible via the Network Exposure Function (NEF).

5. *Unified Data Management (UDM)*: The Unified Data Management (UDM) handles user-related services such as user subscription, identification, authentication and access management as well as registers user network functions (AMF and SMF).

6. *Policy Control Function (PCF)*:): Policies and rules are managed in PCF which unifies policies and provides control plan function.

7. *Network Exposure Function (NEF)*: The NEF acts as a centralised point to service exposure as well as provides authentication for outside access requests.

8. *Authentication Server Function (AUSF)*: The AUSF establishes the authentication procedure.

9. *User Plane Function (UPF)*: From the MEC point, the UPF can be considered as distributed data planes. The UPF is responsible for data plane control and configures traffic rules.

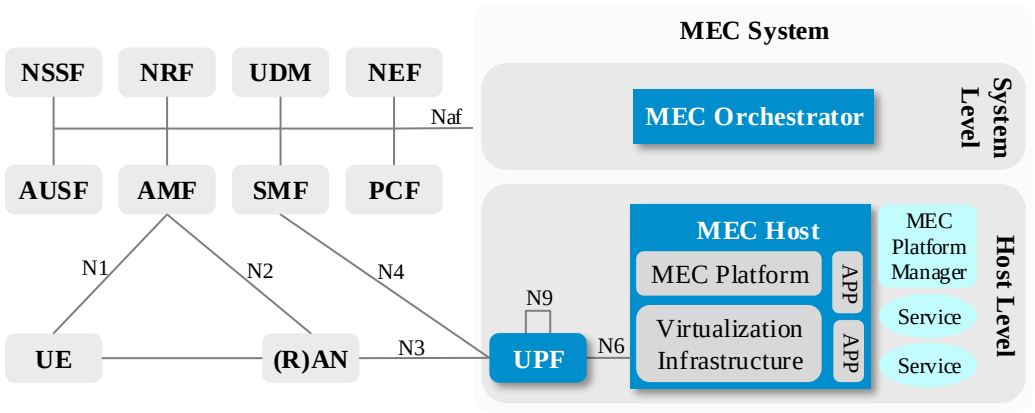


Figure 2.2 : Integrated MEC deployment in 5G network.

The MEC architecture is either system-level or host-level. The MEC orchestration (MECO) is the main part of the MEC system level which stores information of MEC host, resources, services and topology of the MEC system. MECO also selects hosts for application instantiation, triggering and relocating. The host is responsible for platform management (PM) and the virtualisation infrastructure manager (VIM). The PM manages application life cycles, controls the application rules and requirements, executes fault reports and controls acknowledgements from VIM. The VIM is responsible for virtualised resource allocation, providing virtual infrastructure, monitoring application performance and provisioning MEC applications. Lastly, the MEC host constitutes PM and VIM.

Developers must operationalise several standards in integrated MEC and 5G/6G networks for MEC orchestration, as shown in Figure 2.3.

2.1.3 Mobility management in MEC

Mobility management is an important factor in MEC systems since UE mobility may shift an associated network entity into a different MEC host. The requirements for mobility in MEC systems are as follows [19]:

- Service continuity
- Application mobility (VM); i.e., reallocation of applications
- Mobility of UE information for a specific application

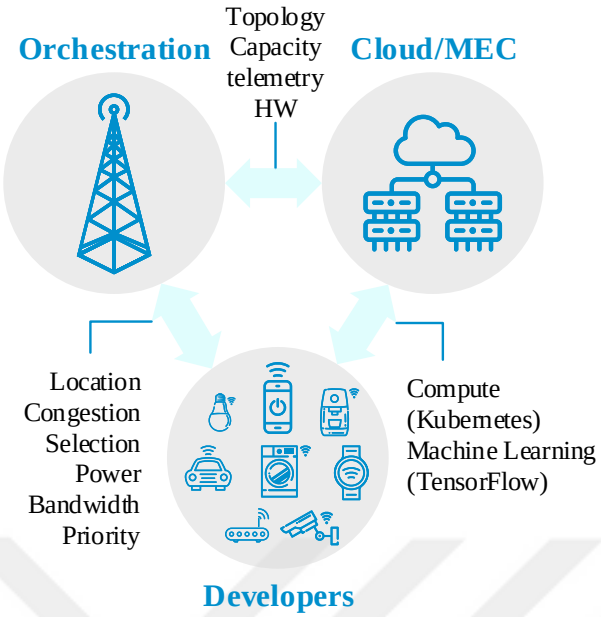


Figure 2.3 : Standards which must be operationalised in integrated MEC and 5G/6G networks.

Handover takes place once the UE crosses the coverage cell of the MEC server. Four scenarios can cause handover to occur during task offloading in MEC systems: no handover, data transmission, task computation, and when sending the result to UEs.

During UE movement within a mobile network, the MEC host serving the UE can be changed, causing handover. In order to support the handover, MEC systems should reallocate the application instance into a target MEC host. Therefore, to improve the QoE, reallocation failure must be reduced. The reallocation failure occurs due to three reasons: too late reallocation, too early reallocation and wrong MEC host reallocation [19]. As a result, efficient handover prediction is crucial in MEC.

The mobility of UEs takes place during two scenarios: intra-MEC host mobility and Inter-MEC host mobility [19]. In the first scenario, the UE moves inside the coverage of the MEC host within the underlying network. Therefore, MEC systems do not require reallocation of the service. In the second scenario, UEs move out of the MEC host coverage into another MEC server coverage which may cause service interruption. In order to provide service continuity, the MEC server is required to hand over the UE's service to the targeted MEC host.

The AMF function is responsible for managing mobility and access of the device through interfaces between endpoints and the core network in association with other network functions, as shown in Figure 2.4.

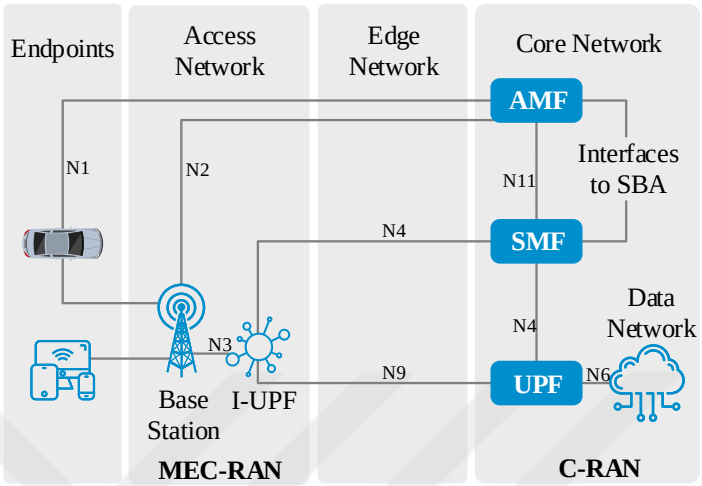


Figure 2.4 : Mobility management in MEC-based 5G network.

2.1.4 Mobility challenges in MEC

In MEC, handovers may occur during data transmission or the computation phase. Once handover is acquired, the application data of the host MEC server will transfer to the new MEC server via a core network, resulting in latency. One of the key issues in MEC systems is controlling handover while having an optimum offloading decision. Some studies had investigated the handover prediction based on the UE's trajectory to handle handovers [20, 21]. However, managing mobility and handovers in ultra-dense MEC networks with higher UE mobilities are challenging due to frequent handovers, large signalling overheads and the ping-pong effect. Moreover, the handover cost should be jointly analysed while considering offloading decisions, energy constraints and resource allocation. Table 2.1 provides a summary of mobility management challenges in MEC.

Table 2.1 : Summary of Mobility Management Challenges in MEC.

References	Topic	Challenges
[19]	Pre-configuration	- Pre-allocation: enables MEC systems to reduce end-to-end delay during high mobility - Reallocation group: To shares application information among MEC server for handover scenarios
[21, 22]	HO prediction	- Essential to choose an optimum server sequentially as vehicles move - Study the handover in urban areas due to higher uncertainty
[23, 24]	Densification	- Manage the mobility of users in an Ultra Dense Network (UDN) - Small-scale coverage results: frequent handovers, HO process power consumption, and radio resource constraint - Real-time information is required for long-term optimisation

2.1.4.1 Densification

Densification is a key parameter for MEC-enabled 5G and 6G networks. Hence, it is challenging to manage the mobility of users in an Ultra Dense Network (UDN). MEC servers provide small-scale coverage, therefore, it is required to use multiple distributed MEC servers. User mobility within MEC coverage causes frequent handovers and service distributions [6]. Studies that focused on Het-MEC had found that one of the main challenges in dense Het-MEC systems is mobility management [23]. Firstly, users may face frequent handovers while moving across a small cell MEC server, resulting in service interruptions and large overhead sizes. Secondly, in order to select targeted MEC hosts, the user must perform handover and processing measurements. However, considering ultra-dense Het-MEC systems, the handover process will lead to power consumption and radio resource constraints. Thirdly, conventional handover decisions are mainly based on signal quality levels between users and the BS. The upcoming system information such as channel status, computation resources, task arrivals and user trajectory must all be considered for long-term optimisation [24]. Some of the challenges of distributed computing over Het-MEC are as follows:

- Determining how to dynamically distribute computing tasks over multiple edge nodes
- Deciding where to run the computation task and what metrics should be considered
- Establishing which metrics are required to select nodes for offloading
- Defining application-awareness in dynamic computing and network access algorithms

2.1.4.2 Handover prediction and pre-allocation

Most common handover controllers decide based on the nearest distance or signal quality level. Considering the vehicle's high mobility in vehicular edge computing systems, managing handover is challenging due to frequent handovers and MEC server exchanges. Frequent handovers cause communication failures and higher delays as a result of task migration. Therefore, it is essential to sequentially choose an optimum server as vehicles move [22]. One solution is to predict handover based on the vehicle's

trajectory. In [21], handover decision is based on vehicle trajectory in a highway scenario. However, it may be challenging to study handover in urban areas due to higher uncertainty [22]. Therefore, to determine the optimal worst-case policy for practical scenarios, the behaviour of drivers should be considered as random.

As shown in Fig. 2.5, if UEs are considered to be highly mobile, such as vehicles, failure may occur as a result of too late reallocation. However, if the UE's trajectory data is available, then MEC systems can proactively predict handover decisions. Therefore, MEC systems are able to select optimum MEC hosts with smooth reallocation, allowing UEs to have higher QoE. In this example, the MEC host reallocates the application state information in advance to the target MEC host before leaving the optimal current MEC host. Pre-allocation enables MEC systems to reduce end-to-end delays and reallocation delays during high mobility handovers [19].

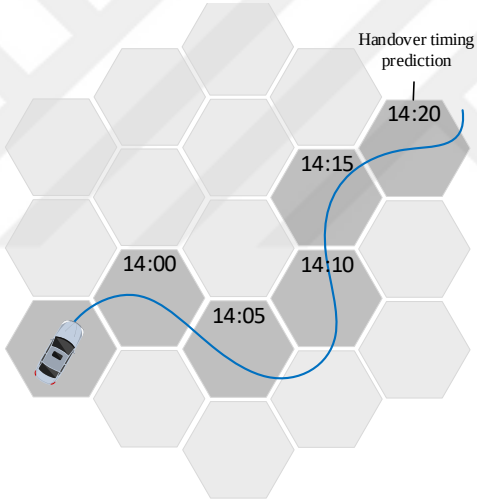


Figure 2.5 : Handover timing prediction in MEC-based vehicular network.

2.1.4.3 Preconfigured reallocation group

Ultra-low latency is a key requirement in 5G and 6G networks. The reallocation process between MEC servers during handover may also cause higher delays. Therefore, the MEC system must minimise handover latency by optimising the reallocation process [19]. One method to optimise handover latency is to preconfigure a group of MEC servers, called the "reallocation group", as shown in Fig. 2.6. The reallocation group is configured based on users' QoE and MEC server's physical location. All MEC servers within a group can share application information so when handover occurs, the MEC servers can establish communication setup based

on preconfigured reallocation in the group. When a UE moves in a group, the MEC server has information about the targeted MEC host. As a result, MEC systems can pre-allocate the application in advance.

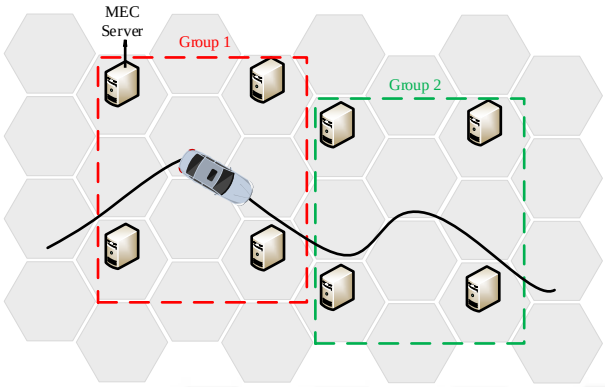


Figure 2.6 : Preconfigured Reallocation Group.

2.2 Related Studies

During the last few years, significant research have been conducted on MEC from various aspects such as architecture, challenges, application use cases, security and privacy. Many have also addressed the integration of MEC technology with 5G and heterogeneous networks. The surveys in [13, 25] present the fundamentals of MEC technology, integration with the 5G network and the MEC architecture. The recent challenges, application use cases and security issues were also addressed. The authors in [7] had reviewed MEC models and discussed joint optimisation and resource allocation. In [26, 27], comprehensive surveys were conducted on ML use cases in MEC systems. The challenges and future research directions such as FL were also investigated. The survey in [9] focused on VEC framework including advantages, key enablers, challenges and applications in VEC. The authors in [28] highlighted service migration in MEC. Table 2.2 provides a summary of existing surveys on MEC and mobility management.

2.2.1 Mobility-aware offloading decision optimisation in MEC

Ultra-low latency is a key factor in MEC-enabled 5G and 6G networks. High mobility of users, especially in a vehicular network, causes frequent handovers and significant delays. The task offloading from the user to the MEC and the task

Table 2.2 : Summary of existing surveys on MEC and mobility management.

Reference	Contribution
[3]	- Review major challenges: security and privacy
[13, 25]	- Survey on MEC architecture and computation offloading - Review of the fundamentals of MEC, technology integration and state-of-the-art - Survey of fog computing: fundamental, network applications and challenges
[7]	- Survey on MEC models. Review of joint communication and computation resource allocation
[26, 27]	- Comprehensive survey on the use of ML in MEC systems: challenges and future perspectives - Federated Learning in Mobile Edge Networks: fundamentals, challenges and existing solutions
[9]	- Survey of state-of-the-art research on VEC: key enablers, advantages, challenges and applications
[28]	- Survey on service migration in mobile edge computing

migration among MEC servers also result in delays. To minimise the handover cost and optimise the offloading process in the MEC-based network, researchers jointly analysed the problem by considering mobility, communication, computation, handover, and backhaul costs. One of the challenges of joint optimisation problems is that they are mostly non-convex and NP-hard. In this section, we investigate the related studies and the proposed solutions to joint mobility management and task offloading optimisations. Table 2.3 provides a summary of related studies on mobility-aware offloading decision optimisation in MEC.

Table 2.3 : Summary of related studies on mobility-aware offloading decision optimisation in MEC.

References	Topic	Proposed frameworks	Contributions
[22]	Task offloading	Robust time-aware MDP-based offloading	- Task offloading decision was formulated as MDP to minimise delay considering handover, migration, communication and computation. - Addressed the uncertainty of transition probabilities
[21]	Joint optimisation	Boundless Simulation Area	- Mobility-aware computation offloading was proposed in MEC-based
[29]	Proactive network association	Mobility-aware anticipatory network association	- Event-triggered delay model was formulated for mobility-aware anticipatory network association, taking future possible handovers into account. - Based on the MDP and Lyapunov optimisation, a two-stage online decision algorithm for proactive network association was innovated.
[20]	Task offloading	Heuristic mobility-aware offloading algorithm	- Mobility-aware multi-User offloading optimisation for MEC
[30]	Handover management	A randomised algorithm	- Existing work focused on cloudlet placement and user-to-cloudlet association problem
[24]	Ultra-dense network	User-centric energy-aware mobility management scheme	- Minimised average delay subjected to communication, computation, and handover under the limited energy budget of users
[31]	Fog computing	Parallel offloading scheme for a vehicular fog computing	- Offloading delay, and handover cost are considered as performance metrics. - The RSU considered the targeted node based on vehicle mobility and dynamic computation resources.

In [20], Zhan et al. investigated the offloading decision problem by considering multiple users where all are served by a single base station. The mobility of UEs was also considered in the optimisation problem. The offloading decision was based on the optimum utility cost defined as a trade-off between task latency and energy consumption. The offloading decision problem was shown as NP-hard. As a result, the article proposed the heuristic mobility aware offloading algorithm (HMAOA) to obtain the best offloading scheme. The proposed method first transforms the optimisation problem into multiple local optimisation problems. Next, each local optimisation problem is decomposed into two subproblems. The first subproblem is resource allocation which is decomposed into a convex computation allocation problem to be solved using a numerical method. The second subproblem is the offloading decision which is decomposed into a Nonlinear Integer Programming (NLIP) and solved by a partial order-based heuristic approach. The simulation results showed an enhancement in user experience over local execution. The outcomes further proved that the offloading performance can achieve near to optimum solution (99.5%). Although, handover was considered while allocating the resource, future studies should assess the handover cost, delay and available resources of the host MEC server and the targeted MEC server. It was assumed that the UE's trajectory can be accurately estimated within a short period. This can be considered in future research.

2.2.1.1 Optimization of high-speed vehicular edge computing systems

Latency is a critical requirement in a vehicular network. Frequent real-time information updates cause significant delays due to the high mobility of vehicles. Task migration among VEC servers also results in delays. Unfortunately, traditional MEC-based offloading methods are inefficient in VEC systems due to high dynamic characteristics and invalidity at a short time. As an alternative to the deterministic optimisation problem, the sequential decision problem must be considered. In [22], the authors proposed task offloading for VEC by taking into account handover and migration in high mobility vehicular networks. In this paper, the task offloading decision is formulated as a Markov Decision Process (MDP) to minimise delay by considering handover, migration, communication and computation. The robust time-aware MDP-based offloading (RTMDP) was proposed to tackle random driver behaviour, complex path and inaccurate sample data over a sequential time slot.

To address the uncertainty of transition probabilities, the authors used an optimal worst-case policy. The results indicate that the performance of the RTMDP method under both certain and uncertain probabilities can achieve better delay. For task offloading problems, future work should consider how to decrease the effect of uncertainty in transition probabilities.

For vehicle random mobility and frequent handovers during task offloading in the MEC-based vehicular network, joint optimisation is required under the random mobility model. In [21], the authors proposed mobility-aware computation offloading in MEC-based vehicular networks. The study assumed that the mobility of vehicles is a random process and considered the possibility of handover throughout the task offloading process. Therefore, handover, multiple rates and backhaul costs were jointly optimised. The article proposed an optimisation model based on the average total costs. The costs include both task time intake and task energy consumption. To model the vehicles' mobility, the Boundless Simulation Area (BSA) model was used. According to the mobility model, the vehicles' mobility can be considered as a Markov chain. Once the handover occurs during data transmission, the backhaul link engages in transferring computational tasks from the host MEC to the targeted MEC server. To evaluate the performance, the system was designed as a highway section with two Roadside Units (RSUs). The results revealed that by increasing data size, the expected costs obtained from computation and communication will also increase. The initial position of the vehicle further influences the total cost of offloading. The simulation results indicate that the cost was reduced by up to 17% in comparison to the baseline, which is a deterministic velocity model. The vehicles' trajectory was assumed to be unchanged in this study. However, future studies should analyse the mobility model in uncertain environments. The task size was also assumed to be limited. The offloading time was less than the vehicle's moving time within two cells. Therefore, more than one handover will not be present.

The fog node selection is challenging due to the dynamic topology of high mobility and uncertain computation resources. In [31], Xie et al. developed a parallel offloading scheme for a vehicular fog computing system. Offloading delay and handover cost were considered as metrics of offloading performance. The proposed method selected target fog nodes by the RSU. The RSU considers the targeted node according to vehicle

mobility and dynamic computation resources. The fog node selection was designed based on the Hidden Markov model and Markov chain theories. Simulation results indicate that the proposed scheme can reduce the service time as well as the handover cost. For future research, the proposed method can be extended for multiple mobile users while considering energy constraints.

2.2.1.2 optimization of distributed computing networks in IoT systems

Ultra-low latency is a key factor in IoT. One solution to achieve ultra-low latency in MEC systems is a proactive network association. In [29], an online proactive network association was proposed under a distributed computing network to minimise the total delay of handover, transmission and computation with limited energy consumption. The article first modelled a mobility-aware anticipatory network association based on event triggers. The mechanism also considers possible handovers. The integrated MDP and Lyapunov optimisations were developed for a two-stage online decision algorithm for a single machine without knowing the probability distribution for the random event scenario. For multiple mobile machines, a non-linear transformation method was used for general scenarios in an online association decision algorithm. The dataset for mobile machine trajectories was taken from GeoLife. The simulation results revealed enhanced performance, especially in higher handover costs and dense MEC servers, resulting in lower average delay.

2.2.1.3 Optimization of MEC systems in ultra-dense networks

Mobility management is challenging while considering RAN due to the co-provisioning of radio access and computing services of MEC servers in BS. Mobility management has been studied in the LTE network, however, UDN brings new problems such as frequent handovers and ping-pong effects. In [24], Sun et al. developed a novel user-centric energy-aware mobility management (EEM) scheme to optimise the delay and energy consumption constraints of users. The system model considers both the radio handover and migration cost. The paper applied Lyapunov optimisation and multi-armed bandit using an online learning method without future system information. The user-centric mobility management algorithm was proposed to minimise average delay subjected to communication, computation and handover under users' limited energy budget. The simulation results indicate

that the proposed algorithm can achieve near to optimum delay cost under energy consumption constraints. When the learning time is large, frequent handovers further increase the handover regret, resulting in high delay. Therefore, the learning time must be carefully selected. Future research should consider high mobility scenarios by examining cooperative task computing for multiple BSs.

2.2.1.4 Handover minimization using region partitioning

To minimise the handover in MEC-based networks, the authors in [30] proposed a region partitioning approach. Most works had focused on cloudlet placement and offloading problems, assuming that cloudlets have fixed capacities. However, the article indicated that the number of service handovers between the MEC servers impacts the QoE and service costs. A mobility-aware randomised partition algorithm (MAPA) was further proposed to minimise the number of possible handovers by dividing MEC coverage regions into disjointed clusters. The article evaluated the proposed algorithm for both random and real traces. The results indicate that the proposed MAPA algorithm can find sub-optimal partitions and reduce the total handover. For future works, and to increase the accuracy of user traces, the integrated deep learning prediction method can be used to anticipate the users' trajectory to design a fast adaptive algorithm.

2.2.2 Machine learning based mobility-aware offloading decision in MEC

Recently, the International Telecommunication Union (ITU) has published framework standards for handling data to enable ML in future networks, including 5G. MEC can be used as a source, collector, pre-processor, model and distributor [32]. Accordingly, MEC can pre-process collected data and execute models before sending them to the centralised cloud for further processes. ML can also provide numerous advantages in MEC-enabled 5G and 6G networks [33]. Since mobile data are exponentially increasing, and big data with low validity duration are heterogeneous from any source, the ML application in wireless communication will be highly beneficial. Conventional methods in handover decision optimisations and task offloading are insufficient when performing in an offline, iterative or heuristic manner. ML can be used for time-varying and dynamic systems to noticeably improve system performance over time. ML can also provide distributed learning by using multiple MEC servers. Finally, ML

can jointly optimise offloading decisions and mobility management. The following research had focused on the optimisation of handover cost and offloading decision using AI-based approaches. The related studies are summarised in Table 2.4.

Table 2.4 : Summary of related studies on machine learning-based mobility aware offloading decision in MEC.

References	Topic	Proposed frameworks	Contributions
[34]	Task scheduling	DRL offloading scheduling in VEC	- Minimised the long-term cost including latency and energy consumption by scheduling to wait in the queue - In the MDP model, the vehicle's mobility, handover, and dynamics of the queue and wireless transmission were all considered.
[35]	Handover optimisation	ML approach in vehicular fog computing	- ML-based fog location and cost predictor were purposed to optimise the transition of connected vehicles between fog nodes.
[36]	Mobility management	RL approach for mobility management in UDN	- An online RL was proposed to optimise handover decisions by predicting user movement trajectory and periodic characteristics of the number of users.
[37]	Edge on wheels distributed learning	Decentralised edge architecture OMNIBUS	- Enabled a continuous distribution of computational capacity by exploiting moving vehicles as storage and computation resources
[38]	Edge autonomous energy management	RL-based droplet framework	- Droplets learn energy consumption statistics of the devices - Indicated that the mobile device cloud is an alternative for edge cloud

In [34], Zhan et al. proposed offloading scheduling by considering handover decisions. The Deep Reinforcement Learning (DRL) offloading scheduling was proposed for VEC. The method minimised the long-term cost in terms of a trade-off between task latency and energy consumption. To achieve minimum long-term costs, the tasks are scheduled to wait in the queue. However, vehicle movement causes frequent handovers and dynamic wireless environments. Therefore, the optimum solution considers where and when to schedule the task for offloading. To solve the optimisation problem, the solution was modelled based on MDP, and the model was calculated using the DRL algorithm. In the MDP model, the vehicle's mobility, handover, dynamics of the queue and wireless transmission were all considered. The DRL approach was designed based on the proximal policy optimisation (PPO) algorithm. The results indicate that the proposed method can achieve the lowest long-term cost. Moreover, the DRL approach is capable of solving complex decision-making problems. To evaluate the proposed method, the author compared it with six known baseline algorithms and modelled the system in a highway scenario. According to recent research, DRL is considered to be an alternative method for optimising resource allocation, energy consumption, channel state information, latency and bandwidth usage. In order to jointly optimise various parameters, DRL should be investigated in future studies.

In [35], Memon et al. proposed the ML approach for handover optimisation in vehicular fog computing. This paper presented an ML-based fog location and cost predictor to optimise the transition of connected vehicles between fog nodes. In order to determine the optimum fog node, the fog location predictor was used. The cost predictor was also applied to calculate the cost of using a particular fog. The ML approach employed a three-layer feed-forward neural network for fog predictor. The results revealed a 99.2% accuracy. To calculate the cost associated with the requested service, the integrated Recurrent Neural Network (RNN) and the long short-term memory (LSTM) cells were used. The proposed method utilised a predictive system to minimise service interruption during handover between fog nodes. The simulation was conducted in JAMScript using a real-world dataset of vehicle movements. The results indicate that the neural network is less expensive in transferring copies of the trained network to the device for local use as compared to instance-based learning algorithms such as K-nearest neighbours (KNN).

In [36], Zhang et al. proposed a reinforcement approach for mobility management in ultra-dense edge computing. MEC is a promising technology for delay-sensitive applications. In order to reduce delays, the authors minimised delays with handover cost as a penalty term in the task offloading process. This paper proposed an online reinforcement learning to optimise handover decisions by predicting user movement trajectory and the periodic characteristics of the number of users. The proposed MDP-based optimal decision (MBOD) handover scheme first constructs the MDP state by predicting user trajectory. Next, it optimises the MDP's total reward to obtain optimal wireless access node. The mobility management function obtains the handover policy to determine the MEC servers during user movements and achieves optimal handover decisions. The experiments were conducted in the Java language. The results revealed that the proposed method can reduce the average delay of task computation and handover rate compared to conventional handover schemes.

Using multi-MEC servers enables the distribution of ML computation into multiple servers. In [37], the authors proposed a distributed edge computing architecture called OMNIBUS which uses vehicles for computation and storage purposes. The proposed OMNIBUS solution has developed a predictive method to learn vehicular movements and timing in order to offload tasks into the targeted vehicular MEC server. Another contribution was introducing parallel learning, optimisation and caching while the vehicle is moving. The goal for using distributed learning is to reduce training time for large scale ML.

Energy consumption for battery-powered IoT devices is a challenging issue, especially for IoT devices collaborating with edge devices. One reason is that both the IoT device and the edge device move while users request services. Another reason is the constant disruption caused due to drained device battery. Although task offloading and execution are the main concerns in MEC systems, energy maintenance in edge deployment should also be considered. In [38], the authors proposed a Reinforcement Learning (RL) based droplet framework for the autonomous energy management of the edge. The droplets in the framework learn the energy consumption statistics of devices to form reliable resources. The RL framework proceeds based on agents without human interactions, making it autonomous. It was further indicated that mobile devices can be alternatives to the cloud edge due to their mobility based on user

movements. The results obtained from deploying a gradient-based approach compared to two different state-of-the-art approaches exhibited a 10% increase in the rewards earned from energy reduction.

3. MOBILITY-AWARE OFFLOADING DECISION FOR MEC

3.1 System Model

Consider a MEC network, consisting of user equipment (UEs) and N base stations (BSs) equipped with a MEC server, where the BSs are distributed on a finite two-dimensional (2-D) regular grid network each with a supporting radius of R as shown in Figure 4.1. Let $\mathcal{N} = \{1, 2, ..., N\}$ denote the index set of MEC servers and l_n denote the location of BS n equipped with a MEC server. The mobile UEs create a total M task for offloading into MEC servers denoted by the set $\mathcal{M} = \{1, 2, ..., M\}$. Let l_m denote the location of task m created by a user while having mobility. Due to densified deployment of the BSs, it is assumed that each task m can be served by multiple BSs. The set of admissible BSs to the location l_m can be represented as:

$$\mathcal{A}(m) = \{n | \parallel l_n - l_m \parallel \leq R, \forall n \in \mathcal{N}\} \tag{3.1}$$

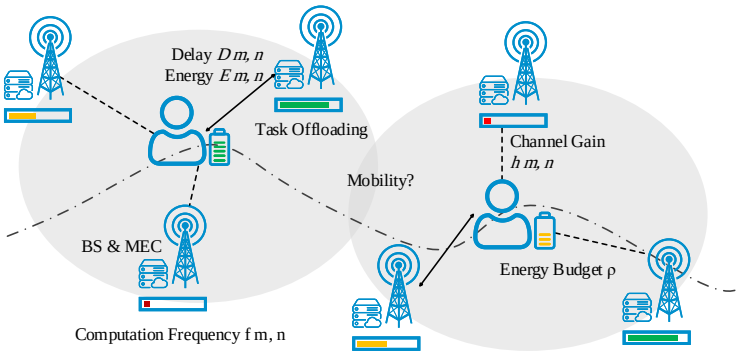


Figure 3.1 : System model of mobility-aware task offloading in MEC enabled 5G network.

In this study, the users decide which candidate BSs $\mathcal{A}(m) \in \mathcal{N}$ are appropriate to offload computational tasks based on utility cost. Also, it is assumed that only a BS is responsible for computing each task m. Therefore, for each task m, users will choose one BS station from $\mathcal{A}(m)$. Moreover, due to large task sizes, each task m

could be divided into many subtasks. Lastly, to manage the mobility of the users, an online learning approach is considered. Therefore, any random mobility model could be applicable. To illustrate the system model, it is divided into three sub-models as follows.

3.1.1 Computation task model

Suppose that a UE generates task m to offload into a MEC server. The computational tasks can be parametrized as a triplet in below:

$$\chi \triangleq [d_m, c_m, t_m] \tag{3.2}$$

Where d_m specifies the total data size of task m (in bits), the computation intensity c_m refers to the number of required CPUs to accomplish computing one-bit data of a task and t_m indicates the computation deadline time allowed for executing the task m (in seconds). Since the size of the computed result is generally small, it is omitted from the equation. Furthermore, considering large input data size, d_m can be divided into k_m number of equal-sized subtaks. Therefore, $d_m = k_m d_{sub}$, where d_{sub} denotes each subtask data size.

After offloading the computation task, the MEC server can allocate CPU for task processing. Let F(n) denote the total computing rate of MEC server n, which indicates the maximum available CPU cycle frequency of the MEC server. The computation frequency $f_{m,n}$, denoted as CPU frequency that MEC server n allocates to task m. Accordingly, the time cost for executing a subtask of m by MEC server n is given as:

$$t^e_{m,n} = d_{sub} \times \frac{c_m}{f_{m,n}} \tag{3.3}$$

3.1.2 Network model

In this study, it is assumed that channel bandwidth ω is equally allocated among the tasks. According to [39], the data transmission rate from UE with task m to the BS n is obtained by

$$r_{m,n} = \omega \times \log_2(1 + \frac{p_m h_{m,n}}{\mathcal{N}_{\sigma^2}}) \tag{3.4}$$

Where $\mathcal{N}_{\sigma^2}$ represents the noise power, p_m is the transmission power of UE m, and $h_{m,n}$ denotes the wireless channel gain from UE with task m to BS n at location l_m, respectively. Also, it is assumed that during task offloading the users are constant and since each task m could be divided into multiples subtasks, the channel gain $h_{m,n}$ would remain constant. The wireless channel gain $h_{m,n}$ is described as

$$h_{m,n}[dB] = 127 + 30 \times \log d[Km] \tag{3.5}$$

Where $d = \| l_n - l_m \|$ refers to the distance between task m at location l_m and BS n (in Km).

Accordingly, the time and energy cost of offloading a subtask of m to BS n is calculated, respectively

$$t^o_{m,n} = \frac{d_{sub}}{r_{m,n}} \tag{3.6a}$$

$$e^o_{m,n} = p_m t^o_{m,n} \tag{3.6b}$$

3.1.3 Mobility model

Due to the large data size of tasks, it could be divided into multiple subtasks. Hence, each subtask can be executed in different BSs. Changing the BSs while computing subtasks of m result in handover delay cost. Let τ^h_m denote handover delay of one-time BSs switch. Considering the multiple subtasks, the sequence of BS denoted by $\eta_m = \{\eta^1, \eta^2, ..., \eta^{k_m}\}$. According to the [24], the handover delay for all subtask of m can be calculated by

$$t^h_{m,n} = \tau^h_m \sum_{k=2}^{k_m} \mathbb{H}\{x\} \tag{3.7a}$$

$$\mathbb{H}\{x\} = \begin{cases} 1 & \eta^k \neq \eta^{k-1}, \quad \eta^k \in \mathscr{A}(m) \\ 0 & \text{otherwise} \end{cases} \tag{3.7b}$$

3.2 Problem Formulation

In the mobile system, the UE experience for offloading decisions is determined by both latency and energy budget. Therefore, we need to design a utility function as a trade-off

between the time and energy budget to make the decision which BS is appropriate to offload and when to perform handover due to the UEs' mobility. As a result, the total time cost for task m can be calculated as the sum of execution time, task offloading time, and handover time as below

$$T_{m,\eta}^{tot} = \sum_{k=1}^{k_m} (t_{m,\eta}^{e} + t_{m,\eta}^{o}) + t_{m,\eta}^{h} \tag{3.8}$$

The total time cost could be written as

$$T_{m,\eta}^{tot} = \sum_{k=1}^{k_m} (t_{m,\eta}^{e} + t_{m,\eta}^{o} + \tau_m^h \mathbb{H}\{x\}) \tag{3.9}$$

Since in this study we only consider task offloading energy cost of UEs, the total energy consumption of UE for offloading task m is calculated by

$$E_{m,\eta}^{tot} = \sum_{k=1}^{k_m} e_{m,\eta}^{o} \tag{3.10}$$

Considering UEs' mobility while task offloading, the decision-making process cannot predict the UE trajectory after computation deadline time t_m. Therefore, the total time cost is constrained to be less than t_m denoted by

$$T_{m,\eta}^{tot} \leq t_m, \forall m \in M \tag{3.11}$$

Eventually, due to UE's limited energy budget, the sum of the energy cost of M tasks is constrained to be less than ρ, the total UE energy capacity, i.e.,

$$\sum_{m=1}^{M} E_{m,\eta}^{tot} \leq \rho \tag{3.12}$$

As mentioned before in equation 3.1, all candidate BS η for task m should be in coverage area within a radius of R, which is denoted by

$$\eta_m^k \in \mathscr{A}(m), \forall k \in \{1, 2, ..., k_m\} \tag{3.13}$$

Our target is to find the optimal offloading decision to minimize the total time cost within a limited energy budget for all tasks M. The problem is formulated as

$$\textbf{GP:} \min_{\eta} \frac{1}{M} \sum_{m=1}^{M} T_{m,\eta}^{tot} \tag{3.14a}$$

$$s.t.: (3.11) - (3.13) \tag{3.14b}$$

where $\eta \in \{\eta_1, \eta_2, ..., \eta_M\}$ is the variable vectors to be optimized. Constraint 3.11 ensures to execute the task on time (subject to the computation deadline time t_m). Constraint 3.12 guarantees the energy consumption to be below the energy budget ρ. Constraint 3.13 states all the candidate BSs that could serve task m.

Due to the non-linearity of the optimization problem 3.14 and complexity of other variables, the problem **GP** is a Mixed-integer Nonlinear Programming (MINLP) problem [40]. As a result, an online algorithm is proposed to address UEs' trajectory during task offloading.

3.3 Online Task Offloading Decision Algorithm

In this section, based on Lyapunov optimization, we present two mobility-aware online task offloading decisions for MEC without the knowledge of future tasks. Later on, we will compare the proposed online algorithms with the offline optimal solution.

3.3.1 Mobility-aware UE-BS algorithm

In this algorithm, the UEs have both UE-side state information and BS-side information simultaneously. As mention before, the BS remains constant during the offloading of one task. Therefore, having both side state information helps UE to select the optimal BS for offloading and avoid any handover. As a result, within the UE-BS scenario, all subtasks of m will be served by a single optimum BS η_m^{opt}. The main challenge in solving **GP** in an online algorithm is that without having future task information $m+1, m+2, \ldots, M$, we will spend the limited energy budget for current tasks and nothing might remain for the upcoming tasks. To overcome this issue, the solution is to define an energy queue and store the used energy budget. Therefore, by exceeding a specific amount of energy budget for task m the future tasks will be

allocated to another candidate BS $\eta \in \mathscr{A}$. According to [24], the energy queue is obtained by

$$\beta_{m+1} = \max\{\beta_m + E^{tot}_{m,\eta^{opt}_m} - \rho/M, 0\} \tag{3.15}$$

Where $\beta_0 = 0$. Eventually, by using Lyapunov optimization we can define the optimization problem as the trade-off between time cost and energy cost of task offloading denoted by

$$Z^{tot}_m = \alpha T^{tot}_{m,n} + \beta_m E^{tot}_{m,n} \tag{3.16}$$

Where α is a control parameter to adjust the trade-off between energy and time cost. As mentioned before, since we have both side information in the UE-BS scenario and we stick on the appropriate BS for the whole time of processing the task m, there would be no handover and the total time cost can be reduced to $T^{tot}_{m,\eta} = \sum_{k=1}^{k_m}(t^e_{m,\eta} + t^o_{m,\eta})$. Moreover, since all the subtask will be served by a single optimum BS η^{opt}_m, the optimization function in 3.16, can be simplified as

$$z_m = \alpha(t^e_{m,\eta^{opt}_m} + t^o_{m,\eta^{opt}_m}) + \beta_m e^o_{m,\eta^{opt}_m} \tag{3.17}$$

To minimize the cost z_m in equation 3.17 for each task m, the online UE-BS algorithm is shown in Algorithm 1.

Algorithm 1 Mobility-aware online UE-BS algorithm

Input: $\mathscr{A}(m)$, d_m, c_m, $f_{m,n}$, $h_{m,n}$ and α.

1: **if** $m = fJ+1, \forall f = 0,1,...,F-1$ **then**
2: $\beta_m \leftarrow 0$
3: **end if**
4: Choose η^*_m subject to 3.11, 3.13 by solving
$\min_{n\in\mathscr{A}(m)} \alpha(t^e_{m,\eta^{opt}_m} + t^o_{m,\eta^{opt}_m}) + \beta_m e^o_{m,\eta^{opt}_m}$
5: Update β_m according to 3.15.

3.3.2 Mobility-aware BS learning algorithm

In this algorithm, the UEs only have UE-side state information. Therefore, in order to make offloading decisions, the UEs are required to learn the BS state information.

Unlike the UE-BS algorithm which all UEs stick on a single and optimum BS for all subtask of m, the learning process causes suboptimal BS selection. Choosing suboptimal BS not only causes additional cost but also can result in handover during the learning process.

To learn optimal BS, one solution is to offload all subtask of m to every candidate BS $\eta_m \in \mathscr{A}(m)$ and observe the total energy and time costs based. Let $\tilde{t}_{m,n}$ and $\tilde{e}_{m,n}$ denote the observed time cost and energy cost, respectively. Therefore, based on equation 3.16, the observed optimization function is as

$$\tilde{z}_{m,n} = \alpha \tilde{t}_{m,n} + \beta_m \tilde{e}_{m,n} \tag{3.18}$$

Where $\tilde{z}_{m,n}$ is a noisy version of $z_{m,n}$ with a specified variance. The main challenge in the BS learning scenario is that UEs might offload their subtask into suboptimal BS due to the variance of $\tilde{z}$. Therefore, the UE tries to offload their task into all BS to learn the best optimum BS. However, offloading as many tasks is not practical and might cause frequent handover. The possible solution is to assign a stop parameter for the learning process. Therefore, the algorithm will only apply to the first k_s subtasks and the remaining subtask will be offloaded to pre-determined optimal BS. The second challenge is how to choose the stop learning parameter. The large k_s might result in frequent handovers and increase the cost whereas small k_s lead to selecting suboptimal BS. To minimize the cost $\tilde{z}_{m,n}$ in equation 3.18 for each task m, the online UE-BS algorithm is shown in Algorithm 2.

Algorithm 2 Mobility-aware online BS learning algorithm

Input: $\mathscr{A}(m)$, d_m, c_m and α.

if $m = fJ+1, \forall f = 0,1,...,F-1$ **then**
  $\beta_m \leftarrow 0$
end if
for $k = 1,...,|\mathscr{A}(m)|$ **do**
  connect o each BS $n \in \mathscr{A}(m)$ once.
  Update $\bar{z}_{m,n,k} = \alpha\tilde{t}_{m,n} + \beta_m\tilde{e}_{m,n}$.
  Update $\theta_{m,n,k} = 1$.
end for
for $|\mathscr{A}(m)|+1,...,k_s$ **do**
  Connect to $\eta_m^k = \arg\min_n\{\bar{z}_{m,\eta_m^k,k} - \beta\sqrt{\frac{2\ln k}{\theta_{m,n,k}}}\}$
  Observe $\tilde{t}_{m,\eta_m^k}$ and $\tilde{e}_{m,\eta_m^k}$.
  $\bar{z}_{m,\eta_m^k,k} \leftarrow \frac{\theta_{m,\eta_m^k,k}\bar{z}_{m,\eta_m^k,k} + \alpha\tilde{t}_{m,\eta_m^k} + \beta_m\tilde{e}_{m,\eta_m^k}}{\theta_{m,\eta_m^k,k}+1}$.
  $\theta_{m,\eta_m^k,k} \leftarrow \theta_{m,\eta_m^k,k} + 1$.
end for
for $k_s+1,...,k_m$ **do**
  Connect to $\eta_m^{k_s}$, $\forall k \in \{k_s+1, k_s+2, ..., k_m\}$
end for
Update β_m according to 3.15.

4. SIMULATION RESULTS

4.1 Simulation Settings

In this section, simulation experiments are provided to evaluate the proposed methods. Table 4.1 summarizes the main simulation parameters with their values. In our simulation, the coverage region of each MEC-equipped BS is considered to be a circle with a radius of 150m deployed on regular grid network within a area of 1000×1000 m^2, shown in Figure 4.1. The channel bandwidth of BS is set to be $\omega = 20$ MHz, and the total available CPU frequency for each BS is $F_n = 25$ GHz. The channel gain from UE to BS $h_{m,n}$ is modeled in equation 4.5 based on [41]. All the wireless communication parameters are set based on 3GPP specification [42].

Table 4.1 : Simulation parameters.

Parameters	Value
Radius of the BS coverage area R	150 m
Availbe CPU frequency on BS F_n	25 GHz
Channel bandwidth ω	20 MHz
Channel gain from UE to BS $h_{m,n}$	$127 + 30 \log d(km)$ dB
Subtask size	$d_{sub} = 0.62$ Mbits
Input data size of each task d_m	$[37.2, 74.4]$ Mbits
Computation intensity of each task c_m	$[500, 1000]$ cycles/bit
Total available computation CPU for each task m by BS n $f_{m,n}$	$[0, F_n]$
Computation deadline of each task t_m	150 ms
Noise power $\mathcal{N}_{\sigma^2}$	2×10^{-13} W
UE transmission power p_m	0.5 W
One-time handover delay τ_m^h	5 ms
Battery capacity	1kJ

According to the [24], one-second video size is set to $d_{sub} = 0.62$ Mbits. Let assume that each video is 60 second to 120 second long and $k_m \in \{60, \ldots .120\}$. According to the equation $\mathrm{d}_m = k_m \mathrm{d}_{sub}$, the input data size is uniformly distributed $d_m \sim U(37.2, 74.4)$ MBits. The same consideration holds for both computation intensity and available computation CPU which is uniformly distributed within $c_m \sim U(500, 1000)$

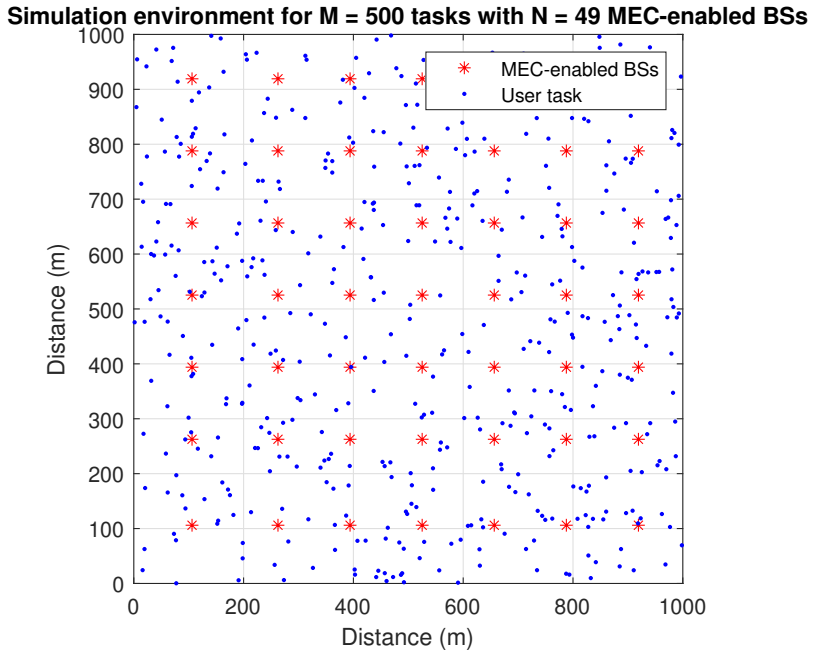


Figure 4.1 : Simulation environment.

cycles/bits and $f_{m,n} \sim U(0, F_n)$ GHz, respectively. To ensure that execution of each subtask is completed within its latency requirement t_m is set to 150 ms.

The number of tasks varies for different simulations. However, we considered a fixed number of BS $N = 49$. The transmit power of mobile UEs are $p_m = 0.5$ W and the noise power is set to $\mathcal{N}_{\sigma^2} = 2 \times 10^{-13}$ W. Also, one time handover is $\tau_m^h = 5$ ms and battery capacity is $J = 1$kJ.

4.2 Numerical Results

In this section, we compare our proposed two methods, UE-BS and BS-Learning algorithms, with the optimum offline solution and two common benchmark algorithms named Time Greedy and Energy Greedy. Moreover, to evaluate the performance of proposed methods, it is compared with the related study in [24]. The summary of the whole simulation steps is provided in Figure 4.2

Figure 4.3 compares Average time cost and Total energy cost for all algorithms over different task sizes $\mathscr{M}$. As seen in Figure 4.3a, both EMM-GSI and the proposed UE-BS algorithms have time cost near to optimum offline solution due to having BS-side information. However, EMM-LSI and the proposed BS-learning have additional costs as a result of sub-optimal BS selection and handover cost.

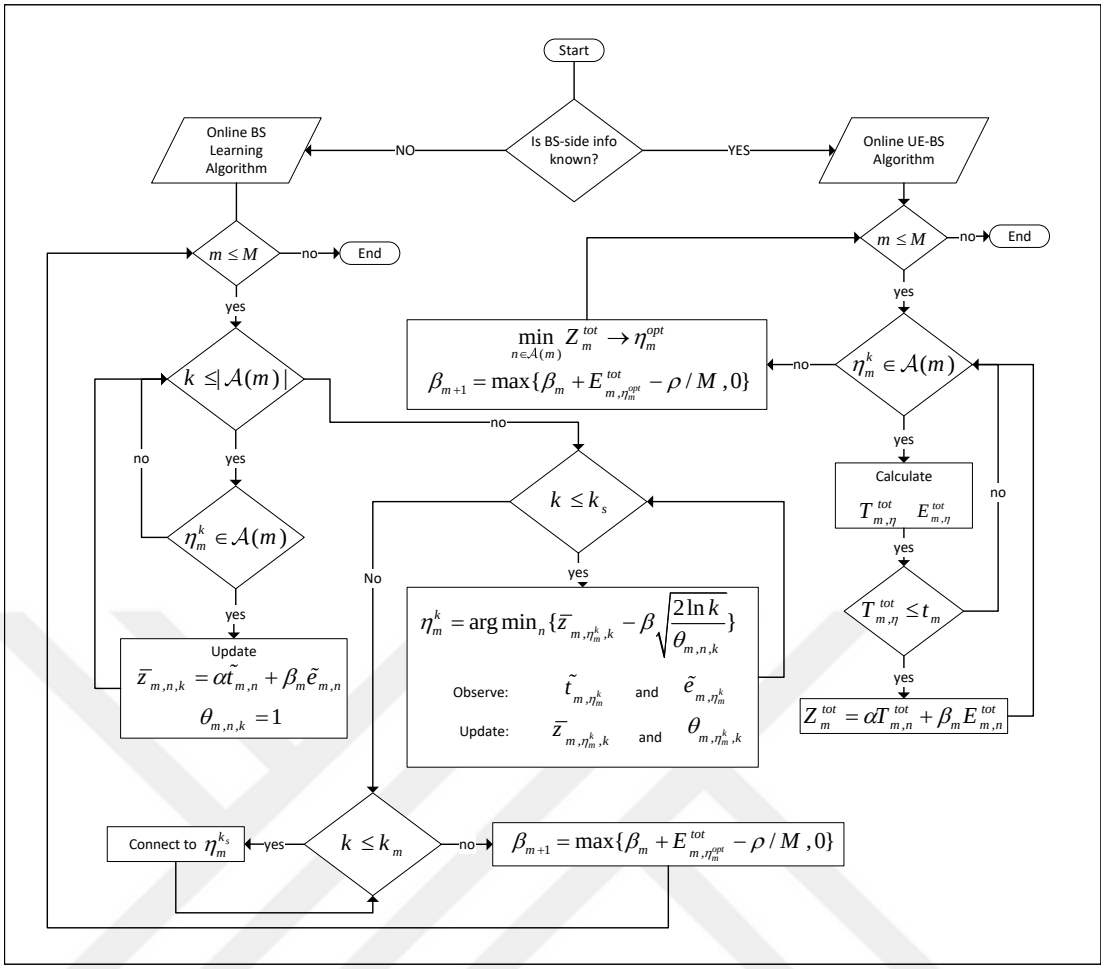
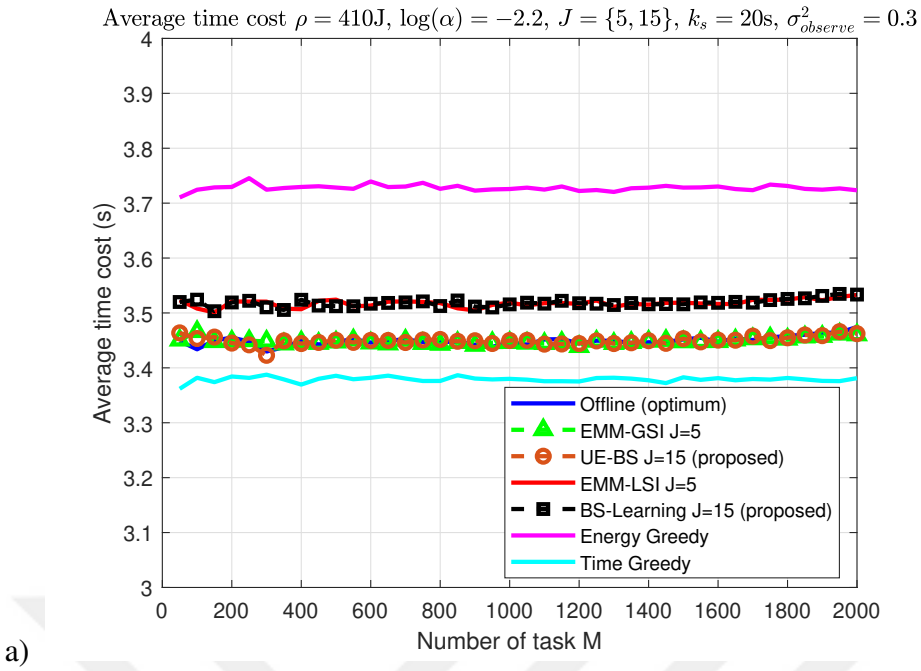


Figure 4.2 : Simulation steps.

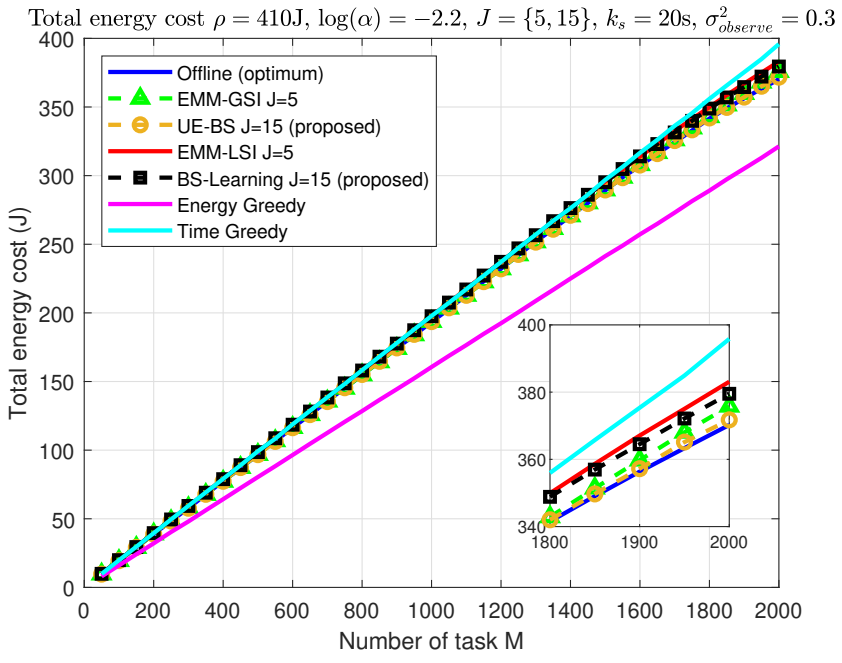
Moreover, for the higher number of task size in Figure 4.3b, the proposed methods have slightly better performance comparing with exciting studies and they have energy cost near to optimum offline solution. Furthermore, the proposed methods keep the energy consumption below the energy budget ρ while having near to low time cost.

Figure 4.4 shows the trade-off between average time cost and total energy cost for α from 10^{-4} to 10. The α is our controlling parameter in the optimization solution. Therefore, we need to ensure it is optimally selected. As seen in Figure 4.4 the interception points indicate the optimum α values.

In Figure 4.4a, the BS-side information is known, however, in Figure 4.4b, the proposed algorithm is going to learn the total cost throughout the observation. Nevertheless, the interception points are almost the same varies between $[-2.3, -2.1]$. For the convenience and to evaluate all the algorithms in the same condition, the controlling parameter is set to $\log(\alpha) = -2.2$



a)

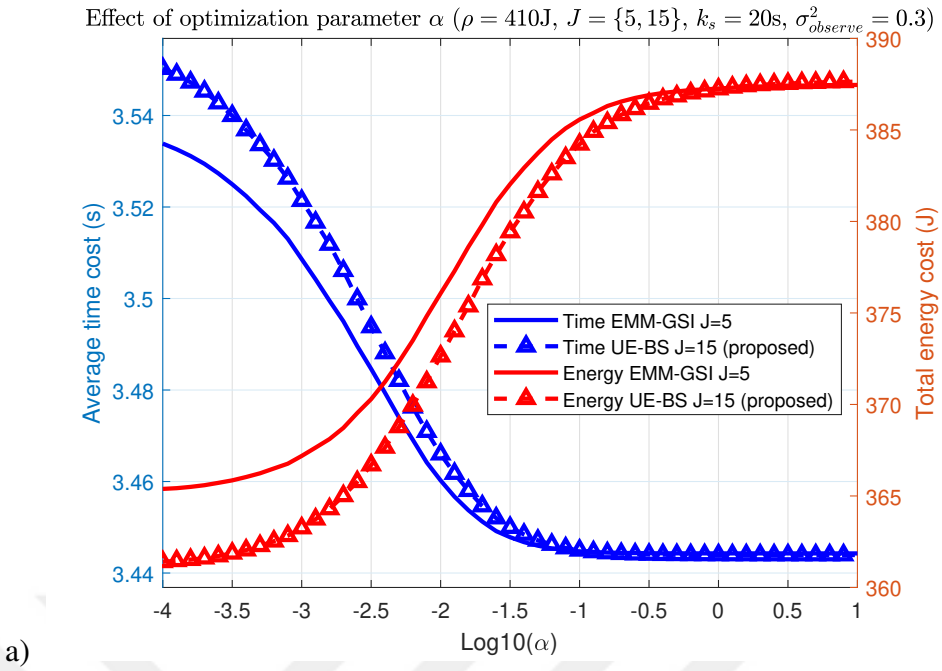


b)

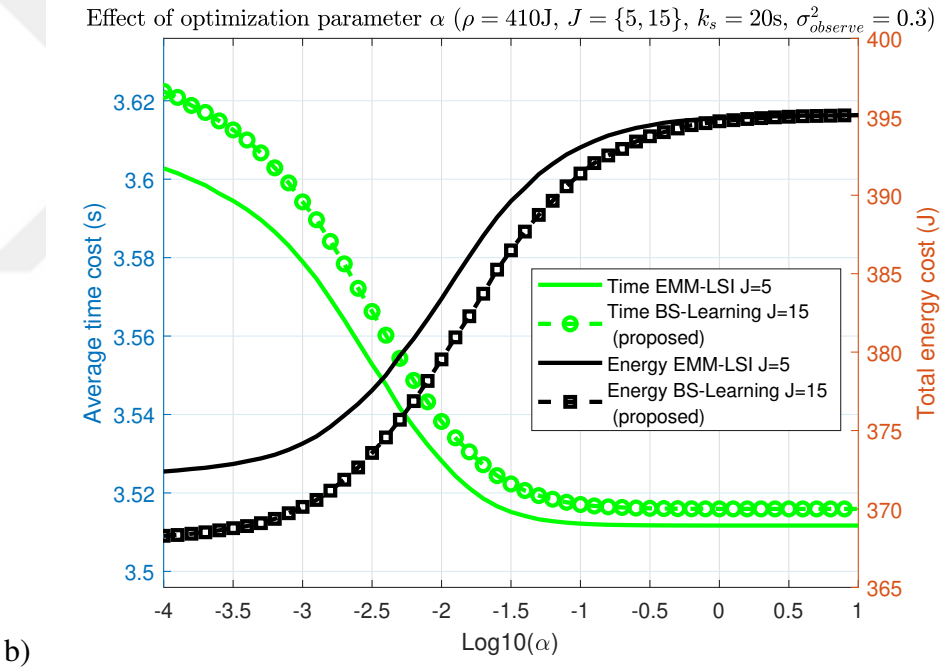
Figure 4.3 : Performance evaluation of Algorithms ($\rho = 410$J, $\log(\alpha) = -2.2$, $J = \{5, 15\}$, $k_s = 20$s, $\sigma^2_{observe} = 0.3$) a) Average time cost b) Total energy cost.

Moreover, due to the handover and learning process in Figure 4.4b, the time and energy costs are slightly higher than those in Figure 4.4a.

Figure 4.5 shows the performance of the algorithms for various energy budgets ρ. The figures indicate that by increasing the energy budget ρ, all the algorithms would reach



a)



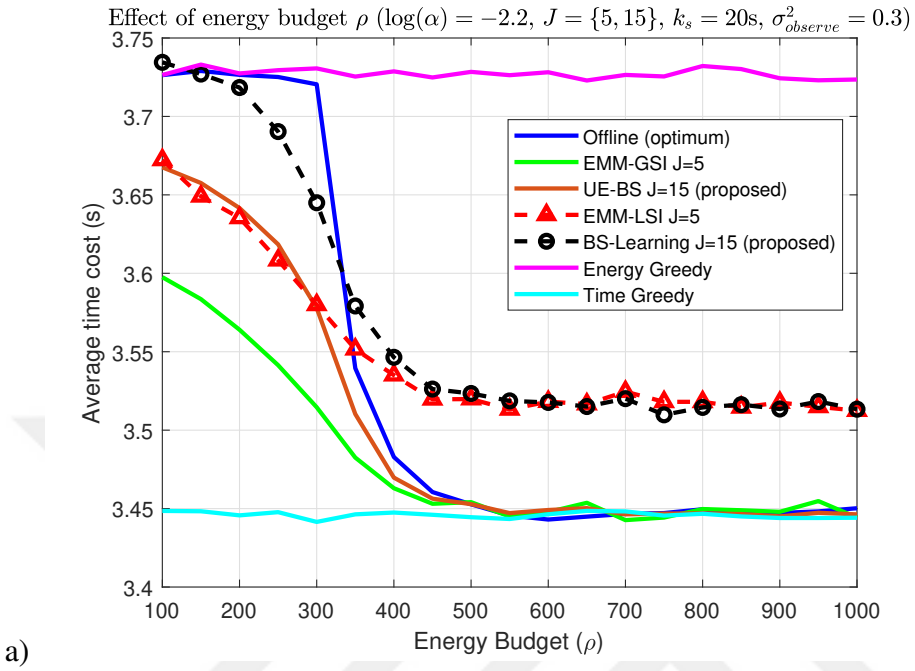
b)

Figure 4.4 : Effect of optimization parameter α ($\rho = 410$J, $J = \{5, 15\}$, $k_s = 20$s, $\sigma^2_{observe} = 0.3$) a) Online UE-BS Algorithm b) Online BS-Learning Algorithm.

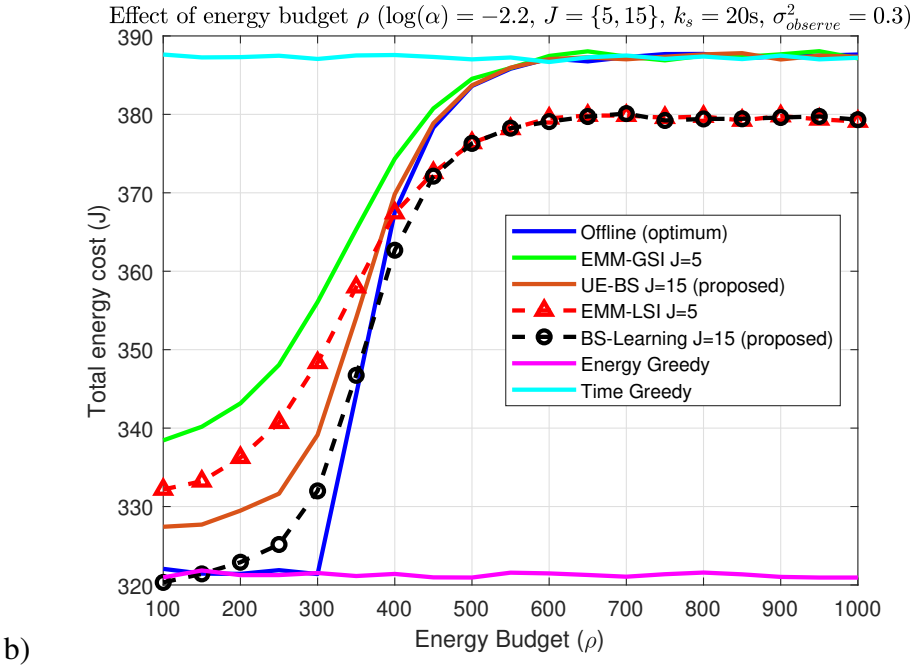
to performance of time greedy. This happens because the users have enough energy budget to neglect the cost and stick on the BS with the lowest average time cost.

Moreover, as seen in both figures in Figure 4.5, if the users do not have BS-side information, they would experience performance loss. Therefore, the EMM-LSI and proposed BS-Learning algorithms are not going to reach the optimum offline cost level.

The next important point in Figure 4.5 is the slope of the lines. The results indicate that our proposed algorithms are closed to optimum offline solution comparing with the ones in [24].



a)



b)

Figure 4.5 : Effect of energy budget ρ ($\log(\alpha) = -2.2$, $J = \{5, 15\}$, $k_s = 20$s, $\sigma^2_{observe} = 0.3$) a) Average time cost b) Total energy cost.

5. CONCLUSIONS AND RECOMMENDATIONS

To conclude the thesis, we conduct research on online task offloading decisions in MEC considering user mobility. Moreover, a limited energy budget is allocated for the users. In order to not exceeding the energy budget, the optimization function developed as a trade-off between time cost and energy cost. Since the future upcoming tasks are unknown, the online algorithms are considered for the solution.

Generally, two scenarios are studied. In the first scenario, both user-side and BS-side information are available. Therefore, the user would be able to calculate the optimum BS and stick on it for the whole task processing period. In the second scenario, the users have no access to BS information. Unlike the first scenario, the users could not be able to calculate the utility cost and take offloading decisions based on it. Therefore, the users offload limited tasks to each BS and observe the total time cost and energy consumption. Later, after learning the optimum BS, the remaining tasks would be offloaded to that BS. However, due to sub-optimum BS selection in the BS-learning algorithm, the BS might be changed during task offloading which results in additional costs including handover.

Simulation results indicate that our proposed two algorithms have slightly better performance comparing with the ones in the exciting article. Although, there is performance loss in the second scenario due to handover, however, the results are close to the optimal offline solution.

In this study, we allocate the computation resources as uniform distribution. For future studies, it could be better to consider resource allocation in the problem formulation as well. This requires solving NP-hard problems and non-convex. One solution is to use ML approaches. However, the application of ML approaches in MEC systems is challenging due to the high complexity of ML algorithms as well as high demands for computation resources. One solution is to divide learning computations into smaller tasks and distribute them among multiple MEC servers. However, there is the issue of determining the types of computations that can be divided as well as how to divide

them while considering MEC resources. Managing to merge the results obtained from different MEC servers for a specific task should also be assessed. Due to UE mobility and frequent handovers among MEC servers, it is quite challenging to integrate outputs from various subtasks into a single output while the UE's trajectory and location are dynamic.

REFERENCES

[1] **Tran, T.X.**, **Hajisami, A.**, **Pandey, P. and Pompili, D.** (2017). Collaborative Mobile Edge Computing in 5G Networks: New Paradigms, Scenarios, and Challenges, *IEEE Communications Magazine*, *55*(4), 54–61.

[2] **u. R. Khan, A.**, **Othman, M.**, **Madani, S.A. and Khan, S.U.** (2014). A Survey of Mobile Cloud Computing Application Models, *IEEE Communications Surveys Tutorials*, *16*(1), 393–413.

[3] **Mollah, M.B.**, **Azad, M.A. and Vasilakos, A.** (2017). Security and privacy challenges in mobile cloud computing: Survey and way ahead, *Journal of Network and Computer Applications*, *84*, 38 – 54.

[4] **Kekki, S.**, **Featherstone, W.**, **Fang, Y.**, **Kuure, P.**, **Li, A.**, **Ranjan, A.**, **Purkayastha, D.**, **Jiangping, F.**, **Frydman, D.**, **Verin, G.**, **Wen, K.W.**, **Kim, K.**, **Arora, R.**, **Odgers, A.**, **Contreras, L.M. and Scarpina, S.** (2018). MEC in 5G networks, *ETSI white paper*, *28*, 1–28.

[5] **Patel, M.**, **Naughton, B.**, **Chan, C.**, **Sprecher, N.**, **Abeta, S. and Neal, A.** (2014). Mobile-edge computing introductory technical white paper, *White paper, mobile-edge computing (MEC) industry initiative*, 1089–7801.

[6] **Sung, N.W.**, **Pham, N.**, **Huynh, T. and Hwang, W.** (2013). Predictive Association Control for Frequent Handover Avoidance in Femtocell Networks, *IEEE Communications Letters*, *17*(5), 924–927.

[7] **Mao, Y.**, **You, C.**, **Zhang, J.**, **Huang, K. and Letaief, K.B.** (2017). A Survey on Mobile Edge Computing: The Communication Perspective, *IEEE Communications Surveys Tutorials*, *19*(4), 2322–2358.

[8] **Song, H.**, **Fang, X. and Yan, L.** (2014). Handover Scheme for 5G C/U Plane Split Heterogeneous Network in High-Speed Railway, *IEEE Transactions on Vehicular Technology*, *63*(9), 4633–4646.

[9] **Liu, L.**, **Chen, C.**, **Pei, Q.**, **Maharjan, S. and Zhang, Y.** (2020). Vehicular edge computing and networking: A survey, *Mobile Networks and Applications*, 1–24.

[10] **Park, H.**, **Lee, Y.**, **Kim, T.**, **Kim, B. and Lee, J.** (2018). Handover Mechanism in NR for Ultra-Reliable Low-Latency Communications, *IEEE Network*, *32*(2), 41–47.

[11] **Lei, L.**, **Xu, H.**, **Xiong, X.**, **Zheng, K.**, **Xiang, W. and Wang, X.** (2019). Multiuser Resource Control With Deep Reinforcement Learning in IoT Edge Computing, *IEEE Internet of Things Journal*, *6*(6), 10119–10133.

[12] **Satyanarayanan, M.**, **Bahl, P.**, **Caceres, R. and Davies, N.** (2009). The Case for VM-Based Cloudlets in Mobile Computing, *IEEE Pervasive Computing*, *8*(4), 14–23.

[13] **Pham, Q.**, **Fang, F.**, **Ha, V.N.**, **Piran, M.J.**, **Le, M.**, **Le, L.B.**, **Hwang, W. and Ding, Z.** (2020). A Survey of Multi-Access Edge Computing in 5G and Beyond: Fundamentals, Technology Integration, and State-of-the-Art, *IEEE Access*, *8*, 116974–117017.

[14] **Gu, Y.**, **Chang, Z.**, **Pan, M.**, **Song, L. and Han, Z.** (2018). Joint Radio and Computational Resource Allocation in IoT Fog Computing, *IEEE Transactions on Vehicular Technology*, *67*(8), 7475–7484.

[15] **Dinh, H.T.**, **Lee, C.**, **Niyato, D. and Wang, P.** (2013). A survey of mobile cloud computing: architecture, applications, and approaches, *Wireless communications and mobile computing*, *13*(18), 1587–1611.

[16] **Shi, W.**, **Cao, J.**, **Zhang, Q.**, **Li, Y. and Xu, L.** (2016). Edge Computing: Vision and Challenges, *IEEE Internet of Things Journal*, *3*(5), 637–646.

[17] **Wang, X.**, **Ji, Y.**, **Zhang, J.**, **Bai, L. and Zhang, M.** (2020). Joint Optimization of Latency and Deployment Cost Over TDM-PON Based MEC-Enabled Cloud Radio Access Networks, *IEEE Access*, *8*, 681–696.

[18] (2018). 5G system architecture for the 5G system (3gpp ts 23.501 version 15.2.0 release 15), *European Telecommunications Standards Institute (ETSI)*.

[19] (2017). Mobile Edge Computing (MEC) End to End Mobility Aspects ETSI GR MEC 018 V1.1.1, *European Telecommunications Standards Institute*.

[20] **Zhan, W.**, **Luo, C.**, **Min, G.**, **Wang, C.**, **Zhu, Q. and Duan, H.** (2020). Mobility-Aware Multi-User Offloading Optimization for Mobile Edge Computing, *IEEE Transactions on Vehicular Technology*, *69*(3), 3341–3356.

[21] **Huy Hoang, V.**, **Ho, T.M. and Le, L.B.** (2020). Mobility-Aware Computation Offloading in MEC-Based Vehicular Wireless Networks, *IEEE Communications Letters*, *24*(2), 466–469.

[22] **Zhang, X.**, **Zhang, J.**, **Liu, Z.**, **Cui, Q.**, **Tao, X. and Wang, S.** (2020). MDP-Based Task Offloading for Vehicular Edge Computing Under Certain and Uncertain Transition Probabilities, *IEEE Transactions on Vehicular Technology*, *69*(3), 3296–3309.

[23] **Park, C. and Lee, J.** (2020). Mobile Edge Computing-Enabled Heterogeneous Networks, *IEEE Transactions on Wireless Communications*, 1–1.

[24] **Sun, Y.**, **Zhou, S. and Xu, J.** (2017). EMM: Energy-Aware Mobility Management for Mobile Edge Computing in Ultra Dense Networks, *IEEE Journal on Selected Areas in Communications*, *35*(11), 2637–2646.

[25] **Taleb, T.**, **Samdanis, K.**, **Mada, B.**, **Flinck, H.**, **Dutta, S. and Sabella, D.** (2017). On Multi-Access Edge Computing: A Survey of the Emerging 5G Network Edge Cloud Architecture and Orchestration, *IEEE Communications Surveys Tutorials*, *19*(3), 1657–1681.

[26] **Rodrigues, T.K.**, **Suto, K.**, **Nishiyama, H.**, **Liu, J. and Kato, N.** (2020). Machine Learning Meets Computation and Communication Control in Evolving Edge and Cloud: Challenges and Future Perspective, *IEEE Communications Surveys Tutorials*, *22*(1), 38–67.

[27] **Lim, W.Y.B.**, **Luong, N.C.**, **Hoang, D.T.**, **Jiao, Y.**, **Liang, Y.C.**, **Yang, Q.**, **Niyato, D. and Miao, C.** (2020). Federated Learning in Mobile Edge Networks: A Comprehensive Survey, *IEEE Communications Surveys Tutorials*, *22*(3), 2031–2063.

[28] **Wang, S.**, **Xu, J.**, **Zhang, N. and Liu, Y.** (2018). A Survey on Service Migration in Mobile Edge Computing, *IEEE Access*, *6*, 23511–23528.

[29] **Cui, Q.**, **Zhang, J.**, **Zhang, X.**, **Chen, K.**, **Tao, X. and Zhang, P.** (2020). Online Anticipatory Proactive Network Association in Mobile Edge Computing for IoT, *IEEE Transactions on Wireless Communications*, *19*(7), 4519–4534.

[30] **Guan, X.**, **Wan, X.**, **Ye, F. and Choi, B.** (2018). Handover Minimized Service Region Partition for Mobile Edge Computing in Wireless Metropolitan Area Networks, *2018 IEEE International Smart Cities Conference (ISC2)*, pp.1–6.

[31] **Xie, J.**, **Jia, Y.**, **Chen, Z.**, **Nan, Z. and Liang, L.** (2019). Efficient task completion for parallel offloading in vehicular fog computing, *China Communications*, *16*(11), 42–55.

[32] (2020). Y.3174: Framework for data handling to enable machine learning in future networks including IMT-2020, *The ITU Telecommunication Standardization Sector*.

[33] **Sun, W.**, **Zhang, H.**, **Wang, R. and Zhang, Y.** (2020). Reducing Offloading Latency for Digital Twin Edge Networks in 6G, *IEEE Transactions on Vehicular Technology*, *69*(10), 12240–12251.

[34] **Zhan, W.**, **Luo, C.**, **Wang, J.**, **Wang, C.**, **Min, G.**, **Duan, H. and Zhu, Q.** (2020). Deep-Reinforcement-Learning-Based Offloading Scheduling for Vehicular Edge Computing, *IEEE Internet of Things Journal*, *7*(6), 5449–5465.

[35] **Memon, S. and Maheswaran, M.** (2019). Using machine learning for handover optimization in vehicular fog computing, *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, pp.182–190.

[36] **Zhang, H.**, **Wang, R. and Liu, J.** (2019). Mobility Management for Ultra-Dense Edge Computing: A Reinforcement Learning Approach, *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*, pp.1–5.

[37] **Ergen, M.**, **Inan, F.**, **Ergen, O.**, **Shayea, I.**, **Tuysuz, M.F.**, **Azizan, A.**, **Ure, N.K. and Nekovee, M.** (2020). Edge on Wheels with OMNIBUS Networking in 6G Technology, *IEEE Access*, 1–1.

[38] **Balasubramanian, V.**, **Zaman, F.**, **Aloqaily, M.**, **Alrabaee, S.**, **Gorlatova, M. and Reisslein, M.** (2019). Reinforcing the Edge: Autonomous Energy Management for Mobile Device Clouds, *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp.44–49.

[39] **Goldsmith, A.** (2005). *Wireless communications*, Cambridge university press.

[40] **Guo, H.**, **Liu, J.**, **Zhang, J.**, **Sun, W. and Kato, N.** (2018). Mobile-Edge Computation Offloading for Ultradense IoT Networks, *IEEE Internet of Things Journal*, *5*(6), 4977–4988.

[41] **Niu, C.**, **Li, Y.**, **Hu, R.Q. and Ye, F.** (2017). Fast and Efficient Radio Resource Allocation in Dynamic Ultra-Dense Heterogeneous Networks, *IEEE Access*, *5*, 1911–1924.

[42] **Lopez-Perez, D.**, **Guvenc, I. and Chu, X.** (2012). Mobility management challenges in 3GPP heterogeneous networks, *IEEE Communications Magazine*, *50*(12), 70–78.

CURRICULUM VITAE

Name Surname : Saeid Jahandar Bonab

EDUCATION :

- **B.Sc.** : 2018, University of Tabriz, Faculty of Electrical and Computer Engineering, Department of Communications Engineering

PROFESSIONAL EXPERIENCE AND REWARDS:

- 2020-2021 Student researcher at the TÜBİTAK project BİDEB-2232 118C276.

PUBLICATIONS, PRESENTATIONS AND PATENTS ON THE THESIS:

- **Jahandar, S., Shayea, I., Ergen, M., Mohamad, H.** 2021. Handover Decision with Mobile Edge Computing in 6G Networks: A Survey. *IEEE Access*. (Under Revision)

F. M. SURNAME	OFFLOADING DECISION WITH MOBILITY-AWARE FOR MOBILE EDGE COMPUTING IN 5G NETWORKS	2020