



HAL
open science

Resource Allocation in Vehicular Fog Computing for an Optimal Use of EVs Electric Vehicles Energy

Habtamu Birhanie

► **To cite this version:**

Habtamu Birhanie. Resource Allocation in Vehicular Fog Computing for an Optimal Use of EVs Electric Vehicles Energy. Artificial Intelligence [cs.AI]. Université Bourgogne Franche-Comté, 2019. English. NNT : 2019UBFCK042 . tel-02482379

HAL Id: tel-02482379

<https://theses.hal.science/tel-02482379>

Submitted on 18 Feb 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



**THESE DE DOCTORAT DE L'ETABLISSEMENT UNIVERSITE BOURGOGNE FRANCHE-COMTE
PREPAREE A L'UNIVERSITE DE FRANCHE-COMTE**

Ecole doctorale n° 85936

Sciences Pour l'Ingénieur et Microtechniques

Doctorat d'Informatique

Par

HABTAMU MOHAMMED BIRHANIE

**Resource Allocation in Vehicular Fog Computing for an Optimal Use of
Electric Vehicles Energy**

Unité de Recherche :

Département de Recherche en Ingénierie des Véhicules pour l'Environnement - DRIVE

Thèses présentée et soutenue à Nevers, le 25 Octobre 2019

Composition du Jury :

MARCELO DIAS DE AMORIM	DR CNRS, Université Pierre et Marie Curie, France	Président de jury
RAMI LANGAR	Professeur, Université, Paris-Est Marne-la-Vallée, France	Examineur
KHALED BOUSSETTA	MCF- HDR, Université de Paris 13, France	Rapporteur
LYNDA MOKDAD	Professeur, Université of Paris 12, France	Rapporteur
SIDI-MOHAMMED SENOUCI	Professeur, Université de Bourgogne, France	Directeur de thèse
AHEMEDIN MOHAMMED	Assistant Professor, Wollo Université, Ethiopia	Codirecteur de thèse
EI-HASSANE AGLZIM	MCF-HDR, Université de Bourgogne, France	Encadrant
MOMAMED AYOUB MESSOUS	Ingénieur de recherche, Université de Bourgogne, France	Invité

ACKNOWLEDGMENT

I would like to acknowledge all the people who have helped and supported me for the success of this thesis.

First and for most, I would like to express my heartfelt gratitude to my doctoral supervisor, Professor Sidi-Mohammed SENOUCI, for his patience, kind, continuous support, and encouragement throughout my PhD study years. His guidance and continuous support over these years are priceless to me. It has been a great honor and pleasure for me to do my PhD under his supervision.

I extend heartfelt thanks to my co-advisors: Dr. El-Hassane AGLZIM, Dr. Mohamed-Ayoub MESSOUS, Dr. Ahmedin Mohammed AHMED for your guidance and supervision of this thesis. Your valuable support and expertise greatly help for the success of this thesis.

I am also grateful to my jury committee members, Lynda MOKDAD, Khaled BOUSSETTA, Rami LANGAR and Marcelo DIAS DE AMORIM for accepting to participate for my PhD defense. Thank you for your precious time, your interest, and your valuable feedback and suggestions to improve this dissertation work.

Last but not least, my special thanks go to all the team members of DRIVE Laboratory, and staffs of ISAT for greeting me with empathy during the last three years.

CONTENTS

1. General Introduction	1
1.1 Context and Motivation	1
1.2 Methodology and Contribution.....	3
1.3 Organization of the Thesis	4
2. Overview on Cloud, Edge Computing and Electric Vehicles	7
2.1 Introduction.....	7
2.2 Overview on Cloud Computing	8
2.2.1 Essential Characteristics.....	9
2.2.2 Service Models	10
2.2.3 Deployment Models	11
2.3 Computation at the Edge.....	12
2.3.1 Mobile Edge Computing (MEC).....	13
2.3.2 Fog Computing.....	15
2.3.3 Vehicular Fog Computing (VFC).....	16
2.3.4 Summary	17
2.4 Overview of Electric Vehicles	20
2.4.1 Electric Vehicles: Categories	20
2.4.2 Electric Mobility: Evolution, Subsidy, and Incentives.....	28
2.4.3 Summary	31
2.5 Conclusion	33
3. Resource Allocation in the Cloud and Fog Computing: State of the Art.....	35
3.1 Introduction.....	35
3.2 Resource Allocation in the Cloud	36
3.3 Resource Allocation in Fog Computing	41
3.4 Resource Allocation on Vehicular Fog Computing (VFC)	43
3.5 Discussion	45
3.6 Conclusion	46
4. Efficient Battery Management for Electric Vehicles	49
4.1 Introduction.....	49

4.2	Problem Statement and Description.....	51
4.3	Overview of Electric Vehicles Battery Technologies.....	51
4.4	Description of the Proposed Solution on EVs Battery Management.....	58
4.4.1	EV Battery Details.....	58
4.4.2	Battery Simulink Model	60
4.4.3	Proposed Solution.....	61
4.4.4	Performance Evaluation	63
4.5	Conclusion	67
5.	MDP-based Resource Allocation towards a VFC	69
5.1	Introduction.....	69
5.2	System Architecture.....	71
5.3	MDP Based Resource Allocation	73
5.4	Experimental Evaluation and Analysis.....	80
5.4.1	Simulation Scenarios	81
5.4.2	Results and Discussions	84
5.5	Conclusion	91
6	A Stochastic Theoretical Game Approach for Resource Allocation in VFC.....	93
6.1	Introduction.....	93
6.2	System Model	94
6.3	Stochastic Resource Allocation Game.....	100
6.4	Experimental Results and Discussions	104
6.4.1	Simulation Scenarios.....	104
6.4.2	Results and Discussions	105
6.5	Conclusion	110
7	Conclusion and Perspectives	111
7.1	Conclusion	111
7.2	Perspectives.....	112
8	Publications.....	115

GENERAL INTRODUCTION

1.1 CONTEXT AND MOTIVATION

Besides the high computation and battery storage capabilities the Electric Vehicles (EVs) own, recent ones are equipped, with different networking and communication interfaces. In fact, the technological advances in automotive sector and battery industry have led to the proliferation and the wide spread of EVs, especially in city centers. However, according to [1], more than 95% of vehicles travel is less than 100 miles daily, which on average represents a 34 KWh of energy consumed for a Tesla Model S whose battery capacity is 85 KWh with ~40% energy consumption. Thus, if we keep 20% of EVs energy reserved, we can get up to 40% of energy for other tasks, and the remaining 40% for daily travel. Hence, with the right incentive mechanisms for the owners of the parked EVs, the latter may be used to deliver other services and tasks. One of the most promising potential ideas that we identified in this thesis is to use EVs as computing devices for nearby mobile users. This will make a better usage of the underused resources of electric vehicles for computation service.

In another context, mobile devices, such as smart phones, tablet, notebooks are extensively used for our day-to-day activities as a convenient tool for communication, entertainment, social networking, etc. This popularity makes users to expect to run powerful applications such as interactive gaming, virtual reality, face recognition and natural language processing [2]. However, these mobile devices have usually limited capabilities in terms of computation power, battery lifetime, storage space and even available bandwidth. In order to address these limitations and to continue supporting the ever-increasing applications demands, service providers are usually willing to make use of powerful servers offered through the cloud. However, the recent development is to push the storage and processing capabilities to the edge of the access network closer to their end users, which introduces the fog computing concept.

In fog computing, services are hosted at the edge of the network such as set-top-boxes and access points. The deployed infrastructure of fog computing allowed users to perform their computation demands closer to the sources. As shown in the Figure 1.1, smart devices including vehicles, video cameras, and mobile devices are considered as users of the fog computing framework. Thus, the users' requests are served in the fog layer instead of sending them to the cloud. However, cloud computing service is always available for performing computation tasks that require higher computation resources.

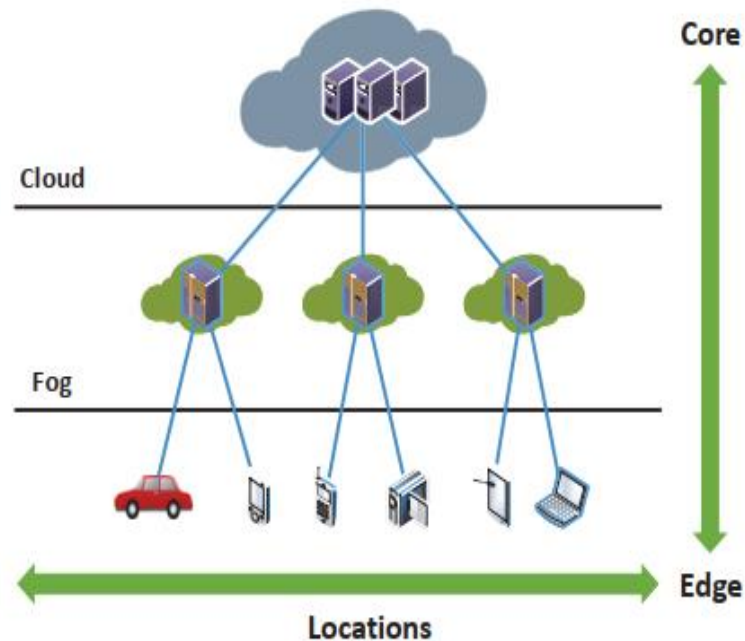


Figure 1.1 - Fog Computing Framework [3].

As presented above, the technological advancement and introduction of EVs with higher computation and energy storage capacity motivate us to propose to use EVs resources for serving local mobile users demand. Figure 1.2 shows the proposed architecture that uses aggregated EVs abundant resources as vehicular fog nodes to serve mobile devices' requests. Therefore, the main aim of this thesis is resource allocation in vehicular fog computing paradigm. Hence, the challenge in this scenario is to optimally use EVs resources for serving local mobile users' demands by keeping sufficient energy for mobility.

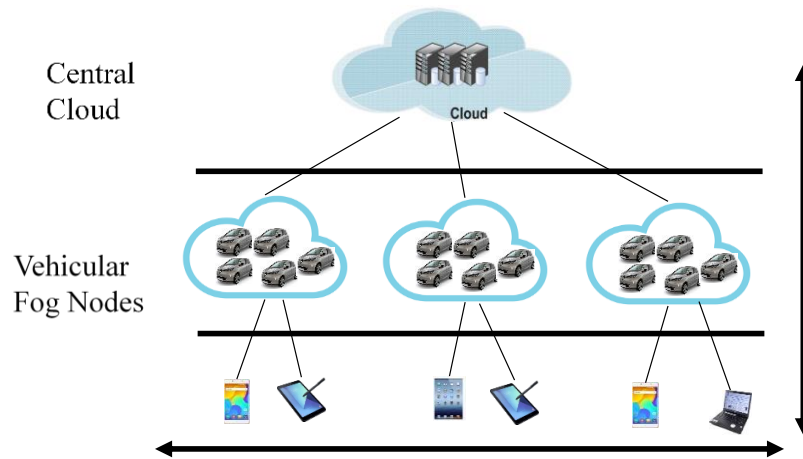


Figure 1.2 - Vehicular Fog Computing Framework.

This chapter represents the general introduction of this thesis. In section 1.2, we present our main contribution and methodology, while in section 1.3, we describe the organization of this thesis.

1.2 METHODOLOGY AND CONTRIBUTION

In the context presented on the previous section, the aim of this thesis is to use parked EVs resources to serve resource constrained mobile application demands. Thus, the contribution of our research is more focused to propose a solution to support computation and energy constrained mobile devices to offload their resource intensive tasks into the nearby parked EVs. Hence, in the proposed system, EVs are considered as vehicular fog nodes that promote from users to computation service providers. However, users' demands which require higher computation demand than EVs fog nodes will be redirected to the cloud.

To deal with abundant vehicles resources for communication, computation and storage, some works were proposed where parked vehicles act as a service delivery platform to stakeholders outside the car, such as pedestrians or moving vehicles [3]. In addition, a VFC enabled offloading scheme is proposed for real time traffic management in smart cities, where parked and moving vehicles act as a fog layer [4]. However, researchers do not consider the abundant resources of parked EVs to support the ever-increasing smart mobiles computation demands. On one hand, applications such as augmented reality and video processing transcends the resources of mobile devices. An alternative to alleviate this problem is to offload the computation demands of smart devices to other devices that have capabilities to compute on their behave, for instance central cloud. On the other hand, EVs with abundant resources parked for long period of time. Thus, by aggregating, these plentiful resources can have a great

potential to serve as fog nodes in vehicular fog computing paradigm. However, little attention is given for the context of using resources of EVs as fog nodes. Therefore, in our first contribution, we propose to show the potential of parked EVs resource to serve local mobile demands while keeping minimum resources for vehicle mobility, in consideration with State Of Charge and State Of Health level of the battery. In literatures, SOH is considered as a feature that determines the energy storage capacity of EVs battery. Hence, we consider EVs with higher SOH and higher SOC (above a threshold) to be selected for computation.

Then, we propose a new solution of parked EVs resource allocation to serve user's demands. We model the problem as Markov Decision Process (MDP) system that aims to optimize the use EVs energy for both computation and moving vehicles. The novelty of this solution is the fact that we take into consideration the dynamic parked EVs resources availability.

Finally, we propose to consider the dynamics of both the users' computation demands and the parked EVs resources and formulate the problem as a stochastic game. Hence, we propose a multiuser computation offloading algorithm, in which mobile users are active or inactive and vehicular fog nodes resources used for computation offloading vary randomly.

1.3 ORGANIZATION OF THE THESIS

Excluding the introduction and conclusion, this work is composed by five chapters in the following manner:

Chapter 2: surveys the most prominent literature works related with cloud, edge, and electric vehicles. It is composed of four sections. At first, we provide an overview about cloud computing characteristics, service, and deployment models. In the second section, we present a comprehensive survey on edge computing paradigms, while focusing on Mobile Edge Computing (MEC), Fog Computing (FC) and Vehicular Fog Computing (VFC). These paradigms are emerged as a solution to address the problems of central cloud computing. Finally, a detailed overview about electric vehicles, their categories, subsidies, and incentive mechanisms for encouraging electric vehicles buyers are presented.

Chapter 3: provides a review on the state-of-the-art related to resource allocation in the cloud, fog, and vehicular fog computing paradigms. Hence, this chapter is composed of three sections. In the first section, we give an overview of resource allocation in the cloud.

Second, we review the resource allocation approach in fog computing paradigm. Finally, the third part is dedicated to present the resource allocation approaches in vehicular fog computing paradigm.

Chapter 4: In this chapter, an efficient battery management for electric vehicles representing the first contribution of this thesis is presented. Since, using EVs battery frequently increases the charging/discharging cycle that have an impact on the battery lifetime, we propose in this chapter, to tackle the problem of using EVs battery for computation in consideration with their SOC (State Of Charge) and SOH (State Of Health) values. The SOC represents the amount or level of charge of EVs battery in relative to its total capacity; while, SOH is the ratio of the current capacity to the initial rated capacity provided by the manufacturer. Hence, in order to reduce the impact on the performance of EVs battery, we consider their SOC and SOH values.

Chapter 5: In this chapter, we propose a new resource allocation strategy based on Markov Decision Process (MDP) in a vehicular fog computing paradigm for optimal use EVs resource. Thus, we start by introducing the vehicular fog computing system architecture of our proposal. Next, we provide a detailed overview about MDP based resource allocation concepts. Then, we formulate the resource allocation problem using MDP system.

Chapter 6: In this chapter, a stochastic theoretical game approach for resource allocation in vehicular fog computing is presented. Thus, we present a generalized approach of resource allocation in VFC paradigm, wherein the amount of resource in the parking lot and computation demand of users' mobile are dynamic. Thus, we present at first the system model of the proposed approach. Then, we present a novel computation offloading decision formulated as a stochastic game with a set of different strategies (local computing, offloading to vehicular fog computing, and offloading to central cloud). Compared to the previous chapter, the innovative formulation of the problem focuses on considering both the dynamic environment of mobile users and the available resources of parked EVs. In fact, the MDP system is a one player game and considers resource allocation scheme from EVs perspective. In this chapter, we use a stochastic theoretical game with mobile device users as a set of players and dynamic available resources in VFC.

OVERVIEW ON CLOUD, EDGE COMPUTING AND ELECTRIC VEHICLES

2.1 INTRODUCTION

Mobile devices are becoming widely used in today's routine. Their growing scale and their wide use have made a great issue of power consumption and computation constraints. Particularly, with the wide adoption of Internet of Things (IoT), we expect not only the rapid growth of connected devices but also the need for real time computation. According to Cisco [5], the number of connected devices will be estimated to 50 Billion by 2020. These connected devices generate massive amount of data that require to be processed. However, devices such as mobile devices have constraints to meet the user's computation demands. In order to solve this issue, several researchers propose to perform the computation tasks by offloading them to powerful devices [6]. Therefore, at the beginning, cloud-based applications have been introduced, in which most of the data storage, analysis and decision making were sent and done on powerful datacenters in the cloud [7]. However, these powerful servers are located far from the origin of the computation, which could create latency on the delivery of computation results. Therefore, when the amount of data and the number of connected devices increase, moving the computation demands from the origin to a distant cloud is not efficient. Thus, a new mechanism to drag the computation capabilities of cloud services into the edge of the network and hence closer to the origin was introduced as a solution. Hence, in this chapter of the thesis, we present the detailed overview of these two paradigms of computation: Cloud and Edge.

In another context, EVs have gained greater attention in the context of growing global warming and global fossil oil crises. Moreover, International organizations and governments give different policy direction for the deployment and wide usage of EVs [8]. Furthermore, these EVs have onboard communication, data storage and computation capabilities. In recent years,

besides mobility purposes, these EVs are used as part of the future Intelligent Transportation System (ITS) [9]. For instance, in order to get the advantage of location awareness and low latency services, VFC enabled offloading scheme is formulated by leveraging the moving and parked vehicles as fog nodes for making real time traffic management system for smart cities [4]. In [4], therefore, vehicles are used as a solution for an ever-increasing computation demands, so that they can be used as a fog node for serving nearby devices. The authors in [10] proposed to investigate the feasibility and challenges of applying VFC for real-time crowd sourcing videos. Thus, they propose to turn commercial fleets into vehicular nodes so that these nodes gather and process videos within the ranges. Their experimental test proved the efficiency of VFC based crowdsourcing in real-world based traffic network scenarios in terms of the network latency, packet loss ratio and throughput. In this thesis, different from other researchers, we consider using parked EVs resource to serve constrained local mobile application demands.

Throughout this chapter of the thesis, we attempt to provide a comprehensive overview of the state of the art related to cloud and edge computing, and electric vehicles. Thus, this chapter is set out in the following manner: in the first part, we introduce an overview about cloud computing characteristics and features. In the second part, we present different concepts related to computation at edge of the network. Basically, we categorize the computation at the edge of the network into three paradigms: mobile edge computing, fog computing and vehicular fog computing. Finally, we give the details related to electric vehicles technology, namely their categories, incentive and subsidy policies, and deployment evolutions.

2.2 OVERVIEW ON CLOUD COMPUTING

In the past decade, cloud computing has got much attention. It is a model for providing different services in cost effective and pay-per-use way. According to US National Institute of Standard and Technology (NIST) [11], cloud computing is defined as “*a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction*”. This cloud computing model, is composed of five essential characteristics, three service models, and four

deployment models [11], as summarized in Figure 2.1. All these parts are described in the following subsections.

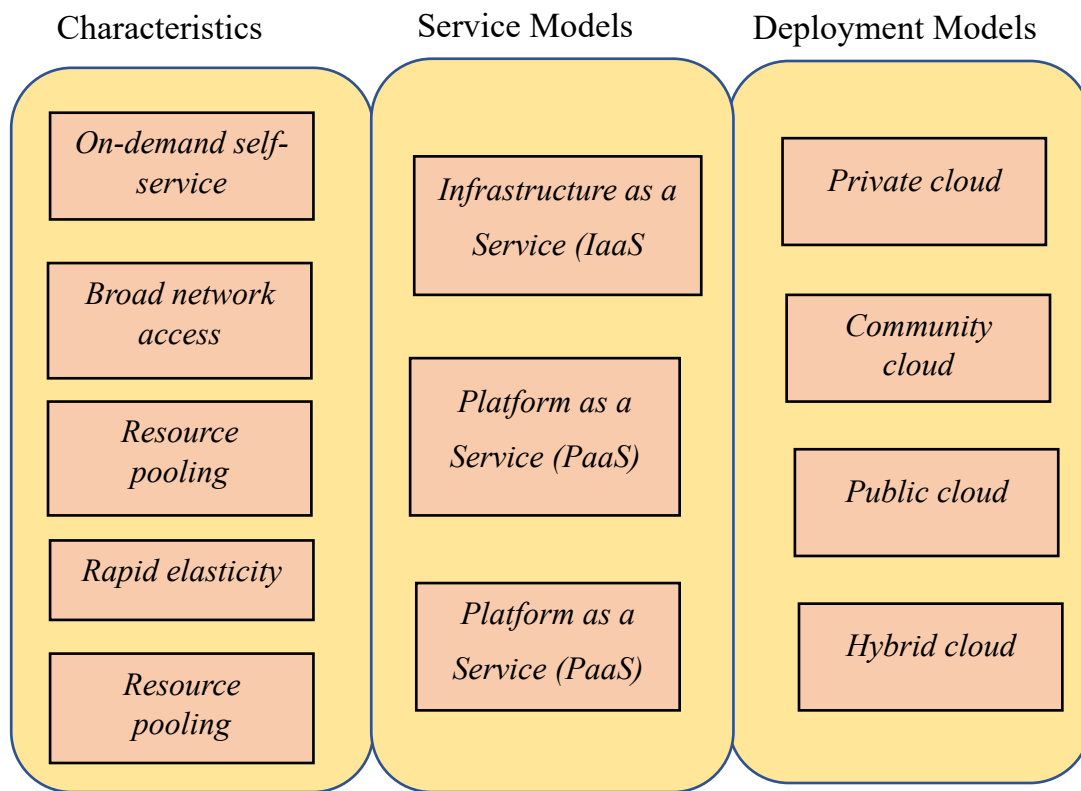


Figure 2.1 - Cloud: Characteristics, Service and Deployment Models.

2.2.1 ESSENTIAL CHARACTERISTICS

In the special publication of NIST [11], the five key characteristics that makes cloud computing different from other technologies are defined. These will be reviewed and presented as follows:

- **On-demand Self-Service:** A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service provider. It is an automated resource provisioning for cloud service users.
- **Broad Network Access:** It provides capabilities to access available resources over the network (remotely) through standard mechanisms that promote the use of heterogeneous thin or thick client platforms (e.g., mobile phones, tablets, laptops, and workstations).
- **Resource Pooling:** The provider's computing resources are pooled to serve multiple consumers using a multitenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand independent from their physical location. Thus, location independence in that the customer generally has no control or knowledge over the exact location. However, consumers may be able to specify their

location at a higher level of abstraction (e.g., country, state, or datacenter). Examples of resources include storage, processing, memory, and network bandwidth.

- **Rapid Elasticity:** Cloud resources can be elastically provisioned and released, in some cases automatically, to scale rapidly outward and inward commensurate with demand. For the cloud service users, resource provisioning often appears to be unlimited and can be appropriated in any quantity at any time.
- **Measured Service:** Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts). Thus, resource usage can be monitored, controlled, and reported, providing transparency for both the provider and consumer of the utilized service in order to enable the pay-as-per use model.

2.2.2 SERVICE MODELS

Classic cloud service models can be categorized into three types: Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). The following is the details the three service models:

- **Software as a Service (SaaS):** The capability provided to the consumer is to use the provider's applications running on a cloud infrastructure. The applications are accessible from various client devices through either a thin client interface, such as a web browser (e.g., web-based email), or a program interface. The consumers do not manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited user specific application configuration settings. The most popular SaaS applications are Google Documents¹ and Google Apps².
- **Platform as a Service (PaaS):** The capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages, libraries, services, and tools supported by the provider. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, but has control over the deployed

¹ Google documents. <https://docs.google.com/>,

² Google apps. <http://www.google.com/intx/fr/enterprise/apps/>

applications and possibly configuration settings for the application-hosting environment. The most popular cloud platforms are Microsoft Azure Services³ and Google App Engine⁴.

- **Infrastructure as a Service (IaaS):** The capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, and deployed applications; and possibly limited control of select networking components (e.g., host firewalls). Consequently, in this scenario the cloud service provider's control responsibility is lower than PaaS and SaaS. An example of a popular IaaS Cloud is Amazon EC2⁵. In [11], the details about the service dependent activities of the two parties are presented and summarized in Table 2.1.

Service Models	Consumer Activities	Provider Activities
SaaS	Uses application/service for business process operations.	Installs, manages, maintains, and supports the software application on the cloud infrastructure.
PaaS	Develops, tests, deploys, and manages applications hosted in a cloud system.	Provisions and manages cloud infrastructure and middleware for the platform consumers; provides development, deployment, and administration tools to perform consumers.
IaaS	Creates/installs, manages, and monitors services for IT infrastructure operations	Provisions and manages the physical processing, storage, networking, and the hosting environment and cloud infrastructure for IaaS consumers.

Table 2.1 - Cloud Consumer and Cloud Provider activities on Cloud Service Model.

2.2.3 DEPLOYMENT MODELS

Another important aspect of cloud computing is cloud deployment based on their variation in physical distribution and ownership. Regardless of their services, cloud services can be categorized into four models as presented in the following:

³ Microsoft azure services. www.azure.microsoft.com/.

⁴ Google app engine. <https://cloud.google.com/appengine/>,

⁵ Amazon EC2. <http://aws.amazon.com/ec2>.

- **Private Cloud:** The cloud infrastructure is provisioned for exclusive use by a private organization comprising multiple consumers (e.g., business units). It may be owned, managed, and operated by the organization. It is not available for public use. This model gives a high-level control over the use of cloud resources.
- **Community Cloud:** The cloud infrastructure is provisioned for exclusive use by a specific community composed by multiple users that have shared concerns (e.g., mission, security requirements, policy, and compliance considerations). It may be owned, managed, and operated by one or more of the organizations in the community, a third party, or some combination of them, and it may exist on or off premises.
- **Public Cloud:** The cloud infrastructure is provisioned for open use by the general public. It may be owned, managed, and operated by a business, academic, or government organization, or some combination of them. It refers to large infrastructure and big resource that aims to provide variety of services for the public through Internet service. It exists on the premises of the cloud provider.
- **Hybrid Cloud:** As its name already indicates this cloud deployment model works as a composition of two or more distinct cloud infrastructures such as private, community, or public cloud that remain unique entities, but are bound together by standardized or proprietary technology for enabling data and application portability (e.g., cloud bursting for load balancing between clouds).

2.3 COMPUTATION AT THE EDGE

The continuing growth of mobile traffic is mainly driven by streaming video, messaging, and peer to peer applications. The growth in mobile traffic is set to increase dramatically as enterprises extend their business processes to smart mobile devices and machine-to-machine solutions. Hence, computation at the edge of the network represents a key enabler to many mission-critical scenarios, from smarter traffic to video analytics. Nowadays, the key transformation has been the capabilities to run IT based servers at network edge, applying the concepts of cloud computing [12]. In addition, billions of IoT devices are expected to generate and communicate with the cloud ecosystem, which will create an expected delay and network jitter [5]. In order to solve this problem, a new computation approach called edge computing was recently introduced with the aim to provide cloud services at the edge of the network. Thus, in the following we detailed the edge computing paradigms namely mobile edge computing, fog computing and vehicular fog computing.

2.3.1 MOBILE EDGE COMPUTING (MEC)

Mobile edge computing (MEC) is the way of moving computing capabilities closer to mobile users. The term MEC was first used to describe the execution of tasks at the edge of the network in 2013, when Nokia Siemens and IBM network runs applications within mobile base station⁶. At the beginning, this concept was a local scope that didn't consider application migration, interoperability and others [13]. Its current features are acquired in 2014, when the ETSI (European Telecommunications Standards Institute) launched the Industry Specification Group (ISG) for Mobile-Edge Computing [14]. According to this specification, MEC aims to “*provide an IT service environment and cloud-computing capabilities at the edge of the mobile network*”. ESTI aims to create an “*open ecosystem*”, where MEC service providers can deploy their applications across MEC platforms. After the standardization process will finalized, telecom companies is in control of this services in their infrastructure [14]. There are different benefits of deploying MEC servers at the edge of the network infrastructures like 5G such as low latency, high bandwidth, and access to radio network information and location awareness [13].

For implementing MEC services, it is required to use virtualized servers in numerous locations at the edge of the mobile network. Some deployment locations considered by the MEC ISG are LTE/5G base stations (eNodeB), 3G Radio Network Controllers (RNC), or multi-Radio Access Technology (3G/LTE/WLAN) cell aggregation sites which can be located indoors or outdoors [13].

On the edge computing technical white paper proposed by ETSI, the edge architecture is designed to provide different services such as video analytics, location services, augmented reality, optimized local content distribution and data caching. As shown in Figure 2.2, edge computing architecture comprises of the end devices in which the computation originated, an edge server that is a powerful computer/server located at the edge of RAN (Radio Access Network) to provide computation service for end devices and the cloud server for storage, analysis and decision making. In addition, several researchers anticipated to use edge computing architecture for different scenarios. The authors in [15], for instance, proposed an intelligent and accurate resource management for internet of things (IoT) devices using artificial intelligent (AI) mechanism. In fact, AI has become the center of attention specially

⁶ <http://www-03.ibm.com/press/us/en/pressrelease/40490.wss>,

for industrial applications. Thus, coordination of AI and edge devices will remarkably enhance the range of sensing and computational speed of IoT based devices in industries. Therefore, this approach improves the efficiency of energy usage and delay time of artificial intelligence driven applications. On another research, authors in [16], proposed to use web service and service oriented architecture to provide location service and in [17] proposed a proactive caching strategy for large amount of data using mobile devices by providing location and mobile trajectories.

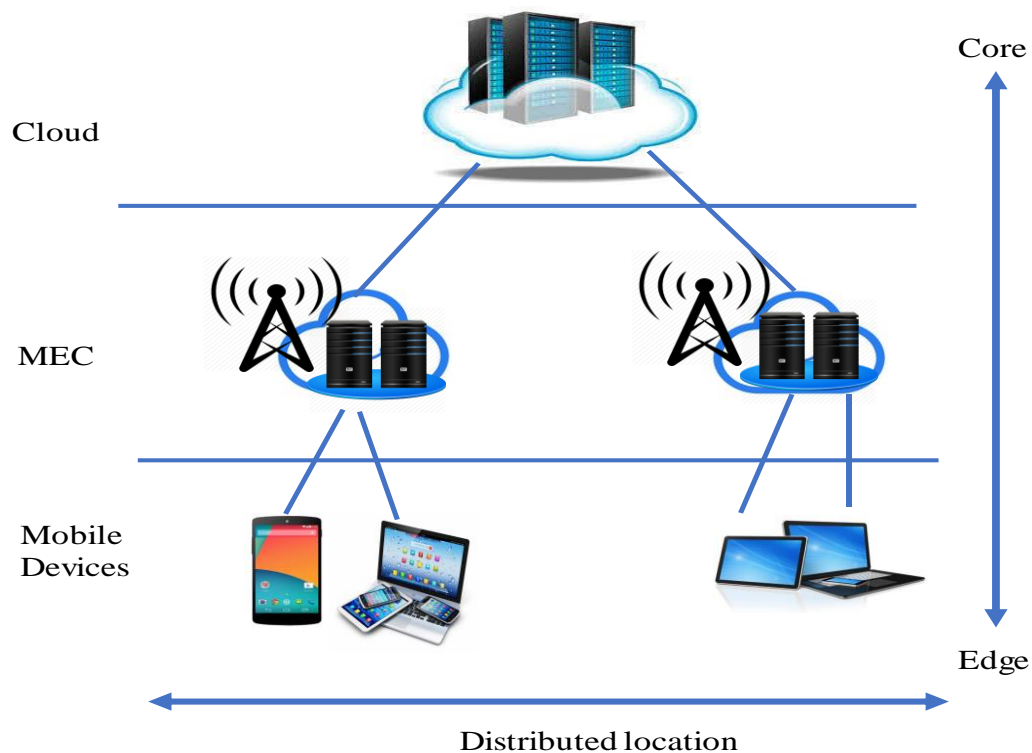


Figure 2.2 - The Basic Edge Computing Architecture [2]

2.3.2 FOG COMPUTING

Fog computing, a term was coined by Cisco, was introduced to extend the computing capability from the center to the edge of the network [18]. This concept was initially proposed by Cisco to enable applications on billions of connected devices, to run at the edge of the network. Thus, fog computing is defined as a highly virtualized platform that provides computation, storage, and networking services between end devices and traditional Cloud Computing Data Centers [19]. Thus, the user's computation demand is served at their vicinity rather than performing it at distant cloud. In addition, fog computing is primarily designed for applications that need real-time processing with low latency. Subsequently, the proposed architecture is improved in order to make the fog nodes able to collaborate with each other. This supports the development of three tier architecture: user devices, fog nodes and central cloud [20]. The authors in [21], presented a three tier fog computing architecture comprising: cloud, resource at the edge of the network and end user devices or sensors as shown in Figure 2.3. Thus, user's computation demands can be served at the end devices such as Wi-Fi access point, set-top-box, gateways,

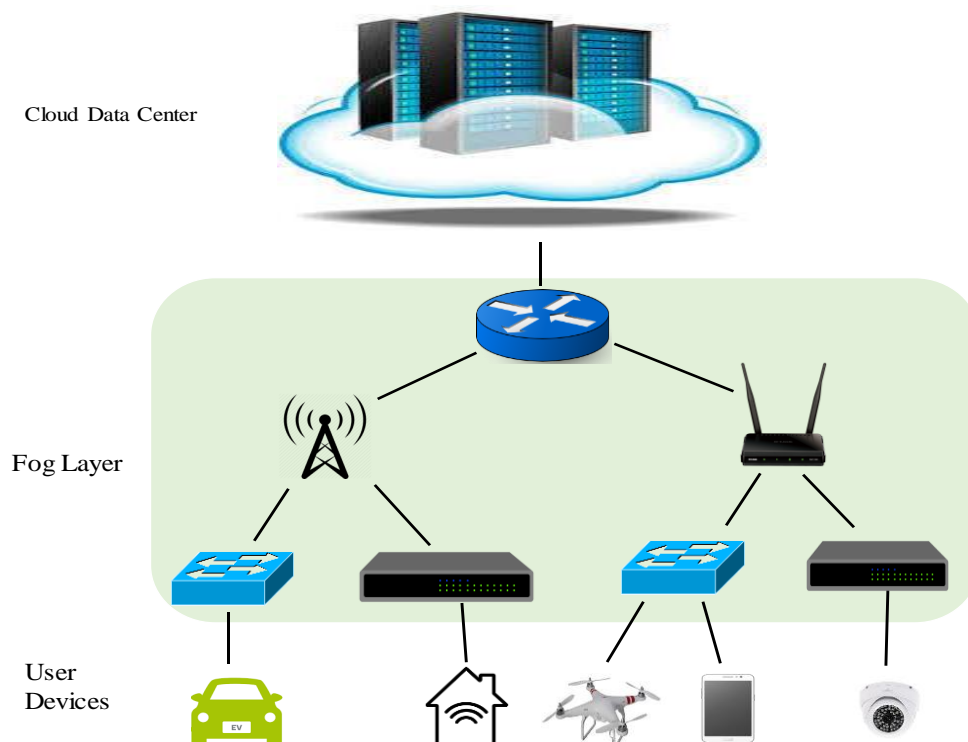


Figure 2.3 - Fog Computing Model [19]

routers etc. Eventually, this would reduce the amount of data movement from the source to distance cloud that reduces communication latencies.

In the literature, some applications are proposed as fog computing application. Authors in [22], for instance, proposed an emergency alert service called Emergency Help Alert Mobile Cloud (E-HAMC) to provide notification about disaster and emergency incidents at the time of disaster. On [23] authors proposed a fog computing based smart parking architecture to improve smart parking in real time. Likewise, authors in [24] proposed to use fog computing architecture and software defined network to improve the practical implementation and constraints of vehicular ad hoc network.

2.3.3 VEHICULAR FOG COMPUTING (VFC)

The other paradigm to use vehicles as part of the cloud ecosystem is Vehicular Fog Computing (VFC). In VFC the resources of both moving and parked vehicles are aggregated to perform the computation service. In order to assist huge data transmission and at real time, vehicular network needs an efficient architecture. The concept of VFC is proposed to install fog nodes on certain connected vehicles in order to deliver computing and communication services where and when it is needed.

The concept of VFC is presented in [25], for utilizing both parked and moving vehicles as part of communication and computation infrastructures. Moreover, the authors carried out a quantitative analysis to show fog capacity, together with the relationship among communication connectivity, capability, and vehicle mobility. However, it was purely a feasibility analysis. The authors in [26] proposed a VFC as three tier architecture: mobile devices, vehicular fog nodes and cloud servers as shown in Figure 2.4.

Vehicles as a fog node works in collaboration with fog computing platform located at the edge of the network. The fog computing platform provides discovery and networking service for vehicular fog nodes. Furthermore, fog nodes do have different components such as communication and computation. From the communication viewpoint, a vehicular fog node is equipped with multiple network interfaces. It can be connected to Internet through a cellular network, or a WLAN (e.g. 802.11p-based Dedicated Short Range Communications - DSRC), or LTE network-assisted Device-to-Device (D2D) communications [27]. Whereas, from the computing perspectives, the resources of vehicular fog node serve the demands of aggregated sensors data. In addition, these fog nodes have also the capability to host different applications of constrained nearby devices.

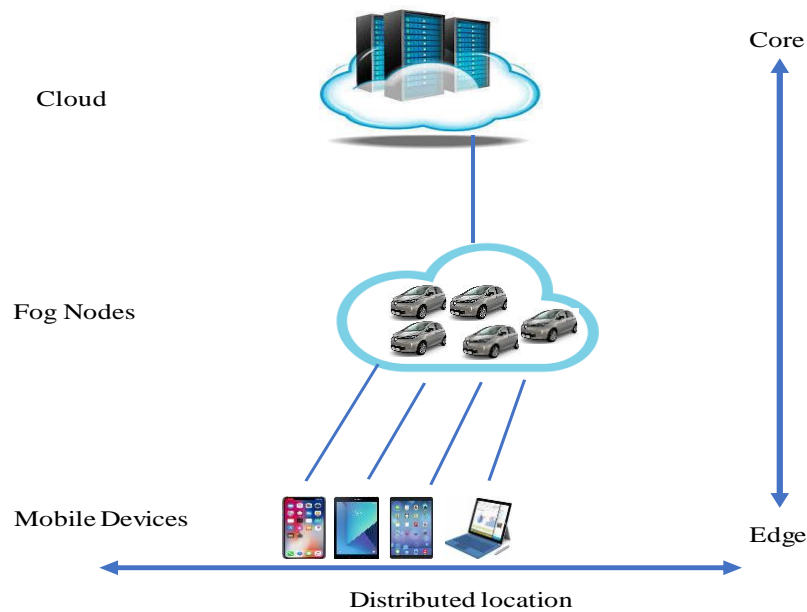


Figure 2.4 - Conceptual Scenario of Vehicular Fog Computing [26].

Moreover, VFC is still emerging stages compared to cloud computing, but there are a variety of areas (use cases) that have been identified as potential scenarios for VFC in both academia and industrial groups as presented in the following:

- In [4], the VFC-enabled offloading scheme is formulated as an optimization problem by leveraging moving and parked vehicles as fog nodes. Specifically, three layers of the architecture are proposed to dynamically cooperate with each other for network load balancing. This helps to manage traffic in a distributed and real time manner for citywide traffic management.
- Furthermore, the concept of using vehicles as an infrastructure have opened a new outlook on different vehicular applications, for instance driving safety, augmented reality, infotainment services, and video streaming [28].
- Thus, video streaming application is one of essential component in Internet of Things (IoT) [29] and involves smart phones. The internet of vehicles is used by various application for video streaming. Therefore, VFC architecture plays an important role for different application that requires high computational processing demands.

2.3.4 SUMMARY

Throughout the previous subsections, we present the edge computing paradigms: Mobile Edge Computing, Fog computing and VFC. Even though, we summarize the details of the most related architectures related with vehicular network, there are also other related scenarios to

the previous paradigms. For instance, authors in [30] define a vehicular cloud networks architecture as a solution to exploit the underutilized resources of vehicles by making vehicles as part of the cloud system to meet the requirements of vehicular ad hoc network applications and services, mainly for safety applications which require cooperation between different vehicles. In addition, central cloud is available to provide resources whenever required. In another example, authors in [31] propose to use the unused resources of connected vehicles to meet service requirements of intelligent transport system. The main objective of this work is to use advantages of cloud computing and applies to vehicular network. They proposed architecture with three layers: the central cloud datacenters, the vehicular cloud that can be regarded as on the computing capabilities providers besides the central cloud and the vehicles equipment's that have computation, communication and storage capabilities as a user and service provider in the proposed architecture.

Table 2.2 provides a compressive overview of the selected edge paradigms presented in the previous subsections in comparison with cloud computing [13]. They all are designed to provide services for users with some type of infrastructures such as fog node, MEC server, etc., which is easily accessible through broadband networks like fiber optic, local wireless communications, or high-speed mobile networks. Besides, all paradigms have their own responsibilities to take into consideration to monitor about the use of different resources. However, the entities in charge are different depending on the edge paradigm. One of the similarities between all paradigms is mobility. In reality, most of the computation services are done locally, it is necessary to take the devices mobility into consideration. Nevertheless, each paradigm makes use of different approaches to help users mobility: from mobility management entities located at a higher level in the network hierarchy, to mechanisms that provide support for the migration of virtual machines [13]. On one hand, edge paradigms work in a sense as an extension of the cloud services and complement its services, which enables the development of multi tired architecture. On the other hand, the items of the edge paradigm can also work in a decentralized and distributed way, that enable to make decisions autonomously and collaborate each other without completely depending on a central network [13]. Since, edge servers are located at the end of the network, latency, and jitter are low compared to the central cloud.

Features	Cloud	MEC	Fog Computing	VFC
Availability	High			
Geographically distributed	N/A	Yes		
Location computing	Far from users	At proximity to users		
Architecture	Centralized	N-tire, Decentralized and Distributed		
Ultra-low latency/Jitter/	Average	Low		
Communication	Constraint in bandwidth	Real time		
Mobility support	N/A	Yes		
Ownership	Private entities	Private entities and individuals		
Deployment	Network core	Network Edge	Near Edge, Edge	Edge

Table 2.2 - Comparison between Cloud, MEC, Fog and VFC.

The other most important benefit of edge computing is location of computing and geographical distribution of services. In edge paradigm, the computation is done with proximity to the users compared with cloud and the edge devices are geographically distributed.

Even though these edge paradigms have the same goal, they are different on how they want to fulfil their goal. For instance, MEC limits the deployment of its servers to edge of mobile infrastructure such as 5G. As for fog nodes are installed in user managed resources such as routers, switches, gateways, access points etc. Whereas, for VFC is designed to utilize vehicles as communication and computation devices. This difference on the deployment of computing nodes influences who can become a service provider. For instance, MEC services are provided by telecom operators, since they are the telecommunication and mobile infrastructure network. On the contrary, any user can introduce its own fog nodes in its computing ecosystem. Besides, in the case of VFC, unused resources of vehicles are aggregated to give services for nearby users. Therefore, vehicles with abundant resources are used for creating VFC architecture.

In order to create an ecosystem with standardized and open mechanisms for edge computing applications, a number of standardization organizations, consortiums, and initiatives have been

joining forces. The IETF⁷, ETSI⁸, Open Edge Computing Initiative⁹, Industrial Internet Consortium and OpenFog Consortiums¹⁰ can be counted as examples of such initiative attempts.

Thus, in this thesis, we propose to use parked EVs with abundant resources as fog nodes, wherein these resources are used to serve constrained mobile devices demand. In the next section, we give an overview about electric vehicles.

2.4 OVERVIEW OF ELECTRIC VEHICLES

EVs are introduced as a solution to solve the problem of environmental pollution, fossil oil depletion and others. In recent years, the advancement in automotive industry and battery technology helps to increase their popularity. Hence, in this section, we present different EV categories and the deployment of EVs followed by the incentives designed to increase the production and usage of EVs.

2.4.1 ELECTRIC VEHICLES: CATEGORIES

EVs are categorized into different groups based on different features. In this subsection we present two categories of EVs: based on power source and based on design or architecture.

2.4.1.1 VEHICLES CATEGORIES BASED ON POWER SOURCE

An electric vehicle is an automobile that uses one or more electric motors for mobility using energy stored in rechargeable batteries. According to their characteristics, EVs are usually grouped under the following categories: Hybrid Electric Vehicles (HEVs), Plug-in Electric Vehicles (PEVs), including Battery Electric Vehicle (BEVs), and Fuel-Cell Electric Vehicles (FCEVs), as shown in Figure 2.5. All these classes are described in the following.

⁷ <https://www.ietf.org/>

⁸ <https://www.etsi.org/>

⁹ <https://www.openedgecomputing.org/>

¹⁰ <https://www.iiconsortium.org/index.htm>

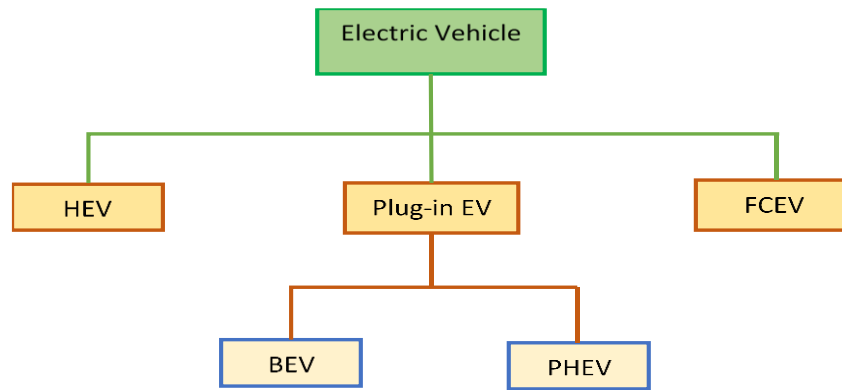


Figure 2.5 - Classification of Vehicles [32].

a) HYBRID ELECTRIC VEHICLES (HEVs)

The HEVs were developed to overcome the shortcomings of both Internal Combustion Engine (ICE) and battery powered electric vehicles. The HEV uses ICE to convert energy from the gasoline or diesel to mechanical energy, which is used to drive electric motor [32]. Thus, the onboard electric motor(s) serves as a device to optimize the efficiency of the ICE as well as to recover the energy during breaking or coasting of the vehicles. Therefore, HEVs primary energy source entirely comes from gasoline and regenerating breaking. There is no means to charge the battery of HEVs from external power source (e.g. power grid). Moreover, HEVs have also certain limitations, increased cost due to the introduction of electric motor(s), energy storage system and power converters. The first mass produced HEV was Toyota Prius, launched in Japan in 1997. Currently, Toyota Camry hybrid, Honda Accord, Toyota Avalon, Toyota Prius, Kia Optima, Chevrolet Bolt and Ford Fusion are the common hybrid electric vehicles.

b) PLUG-IN ELECTRIC VEHICLES (PEVs)

PEVs are a category of vehicles that includes both battery electric vehicles and plug-in hybrid electric vehicles. These categories of EVs are presented as follows.

i) BATTERY ELECTRIC VEHICLES (BEVs)

BEVs satisfy these two conditions: an electric motor powered by a battery replaces the Internal Combustion Engine Vehicle (ICEV) and the gasoline tank, and the vehicle is plugged to a charging spot when it is not in use [33][34]. Their batteries are mainly recharged through a plugin system connected to an external source of power usually grid or recharged through

power regenerative breakings. Since they only have an electric drive system, BEVs are considered as a solution to tackle the global energy crisis and global warming. Nevertheless, this group of EVs have several limitations including high initial cost, short driving range and reduced passenger and cargo space [32]. The first passenger electric car is produced by General Motors EV1 in mid-1990's. Nowadays, Tesla Model S series and BMW i3, Nissan Leaf and Hyundai are the most widely used BEVs in the world. In addition, reduction on the charging time increases the popularity of EVs. Thus, Table 2.4 shows the vehicle type with their battery capacity and charging time (refueling) [35].

EV Type	Battery	Range	Charging Time
Chevrolet Volt	16kWh, Li-manganese/NMC, liquid cooled, 181kg	64km	10h at 115V AC, 15A 4h at 230V AC, 15A
Mitsubishi iMiEV	16kWh; 88 cells, 4-cell modules; Li-ion; 150kg; 330V	128km	3h at 115V AC 15A 1.5h at 230V AC 15A
Nissan Leaf	30kWh; Li-manganese, 192 cells; air cooled; 272kg	250km	8h at 230V AC, 15A 4h at 230V AC, 30A
Tesla S	70 and 90kWh, 18650 NCA cells of 3.4Ah; liquid cooled; 90kWh pack has 7,616 cells; 540kg	424km	9h with 10kW charger; 120kW Supercharger, 80% charge in 30 min
BMW i3	22kWh (18.8kWh usable), LMO/NMC, 22kWh (18.8kWh usable), LMO/NMC,	130- 160km	4h at 230VAC, 30A; 9h with 10kW charger; 50kW Supercharger; 80% in 30 min
Smart Fortwo ED	16.5kWh; 18650 Li-ion	136km	8h at 115VAC, 15A; 3.5h at 230VAC, 15A

Table 2.3 - Battery Capacity and Technologies.

ii) PLUG-IN HYBRID ELECTRIC VEHICLES (PHEVs)

PHEVs works on batteries as a source of energy which are recharged by plugging into the power grid, even though they run mostly on batteries, they are also equipped with an internal combustion engine [36]. Therefore, the combustion unit is used for charging the battery and/or replacing the electric drive system when the battery state of charge becomes low and higher energy is required to drive.

c) FUEL-CELL ELECTRIC VEHICLES (FCEVs)

FCEVs uses fuel cells to generate electricity from oxygen or compressed hydrogen [32]. The power generated through this way is used to drive the vehicle or is stored in the EVs energy storage device, for instance battery pack or ultracapacitors. Since FCEVs generates electricity from chemical reaction, they do not need to burn fuel. Therefore, these categories of EVs are considered environmentally friendly and not pollutant. Although, they do have several advantages, they also have limitations, for instance an increased cost of fuel cells, storage of hydrogen production and transportation of hydrogen, and fuel cells lifecycle [32]. These vehicles are still in development phase, and the fueling stations are limited [36].

2.4.1.2 HYBRID VEHICLE CATEGORIES BASED ON ARCHITECTURES

Hybrid vehicles architecture contains different main components such as Electric Motor (EM), battery, convertor, Internal Combustion Engine (ICE), fuel tank, and controller board [37]. Therefore, different variations exist that integrate these components in different ways, sizes, and order. HEVs architecture contains different main components such as electric motor (EM), battery, convertor, internal combustion Engine (ICE), fuel tank, and controller board [37].

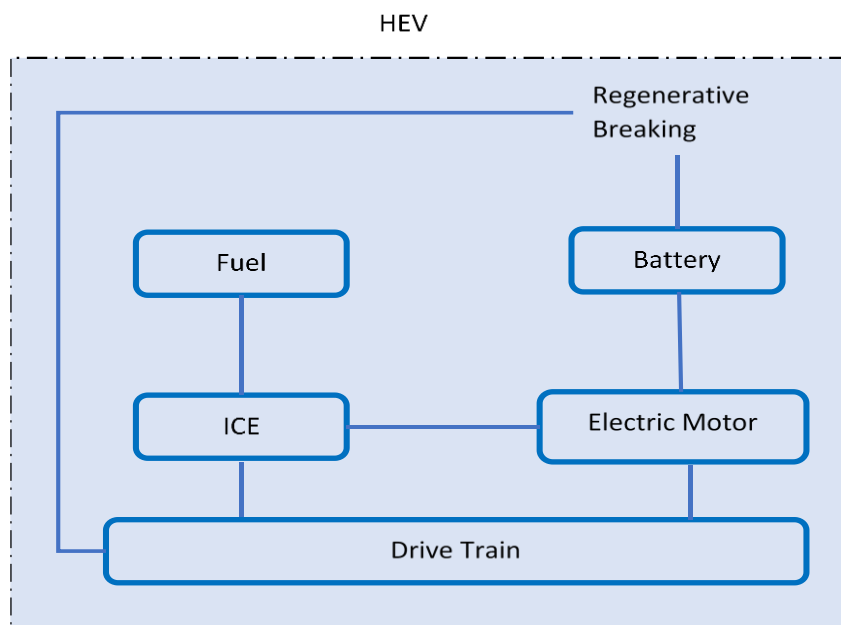


Figure 2.6 – Components of HEV

Therefore, different variations exist that integrate these components in different ways, sizes, and order. As shown in Figure 2.6, the HEV's architecture is classified based on different drivetrain configurations into four classes [38] that are described in the following: series, parallel, combined, and complex hybrid.

a) SERIES HYBRID

In series HEVs, the ICE mechanical output is converted into electricity using a generator as an additional unit for range extender. Thus, the wheels are driven by electric motor that is also operated by ICE. The ICE is switched on only to charge the battery or to feed the main motor when the battery state of charge has fallen below a lower cut off [39]. In series hybrid architecture, as shown in Figure 2.7, the generator and the battery are directly connected to the electric motor that drives the vehicle. The ICE is coupled with the generator to charge the battery and to provide power for the electric motor. Since the motor is not directly connect to the drive axle, the transmission doesn't require a clutch. In addition, the decoupling between the engine and the driving wheel adds flexibility to locate the ICE generator set. The drawback of this architecture is a need to have larger battery pack and generator required. Therefore, this configuration is more efficient at lower speeds and appropriate for urban driving. Chevrolet Volt is an example that deployed series hybrid architecture [40].

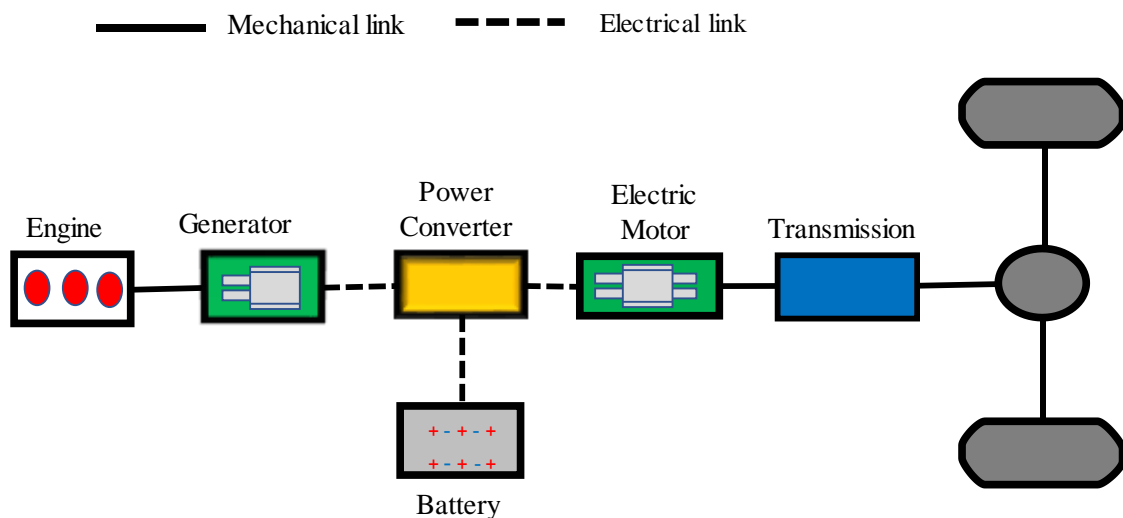


Figure 2.7 - Architecture of Series HEV.

b) PARALLEL HYBRID

Different from series hybrid, as shown in Figure 2.8, parallel hybrid works with the ICE and electric motor to provide energy for the main drive [39]. Since the ICE and electric motor are coupled with the transmission, the propulsion power may be supplied by ICE alone, with electric motor, or by both. Thus, this architecture provides an independent use of ICE and electric motor with the use of two clutches. As an advantage in parallel hybrid, HEVs require

only ICE and the electric motor [32]. In addition, it significantly needs a smaller ICE and engine motor to get the same performance, than that achieved with series hybrid. However, ICE in parallel hybrid works with different variants that increase the fuel consumption. Thus, this architecture is appropriate for high speeds and highway driving. Honda insight is an example that uses this architecture [40].

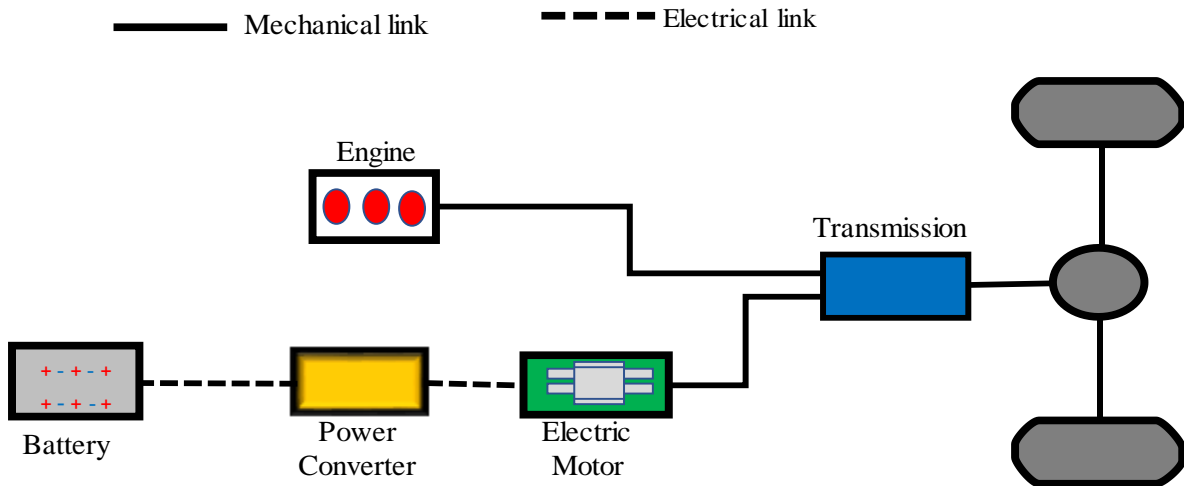


Figure 2.8 - Architecture of Parallel HEV.

c) COMBINED HYBRID

Combined hybrid, also called series-parallel hybrid architecture, that have the flexibility of working in either series or parallel [41]. It is designed to take the advantage of both series and parallel HEV architectures. As it is shown in Figure 2.9, it integrates an additional mechanical link compared with series hybrid, and an additional generator compared with parallel hybrid. Thus, combined hybrid architecture is developed by taking the advantages of both serial and parallel architectures. In comparison with the latter two architectures, the former is complicated

and more costly. However, with the advancement in the control and automotive technologies, some modern HEVs prefer to adopt this system [32].

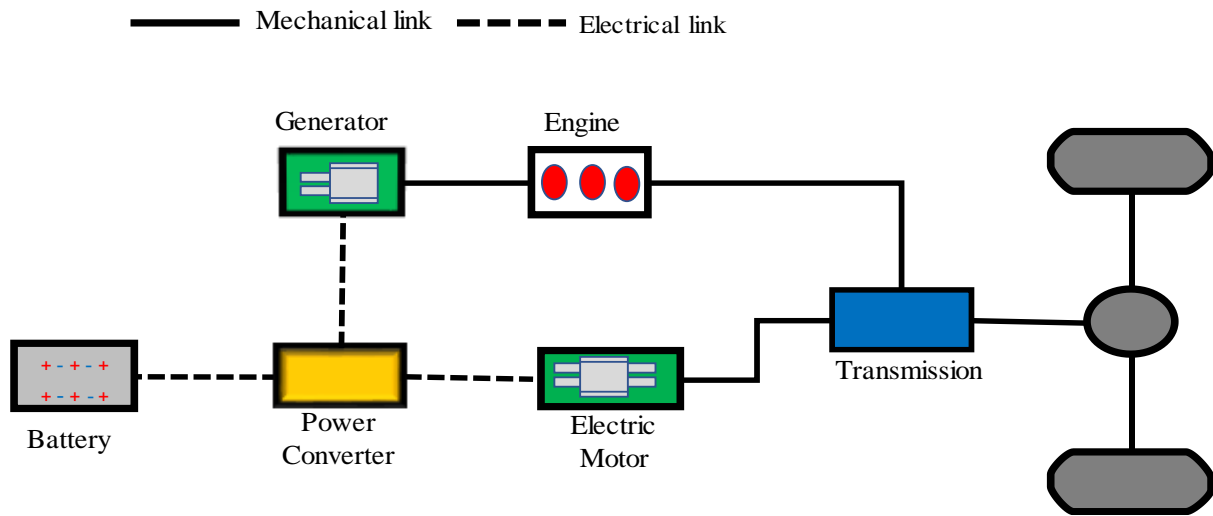


Figure 2.9 - Architecture of Combined HEV.

d) COMPLEX HYBRID

As its name indicates, complex hybrid architecture features complex arrangement and cannot be classified under the above three categories. As it is shown in Figure 2.10, it seems a similar architecture with combined hybrid with the exception of integrating power converters connected to motors and generators. Therefore, this topology is efficient in power handling, more controllable and reliable [32]. However, its drawback is the cost and the complexity to

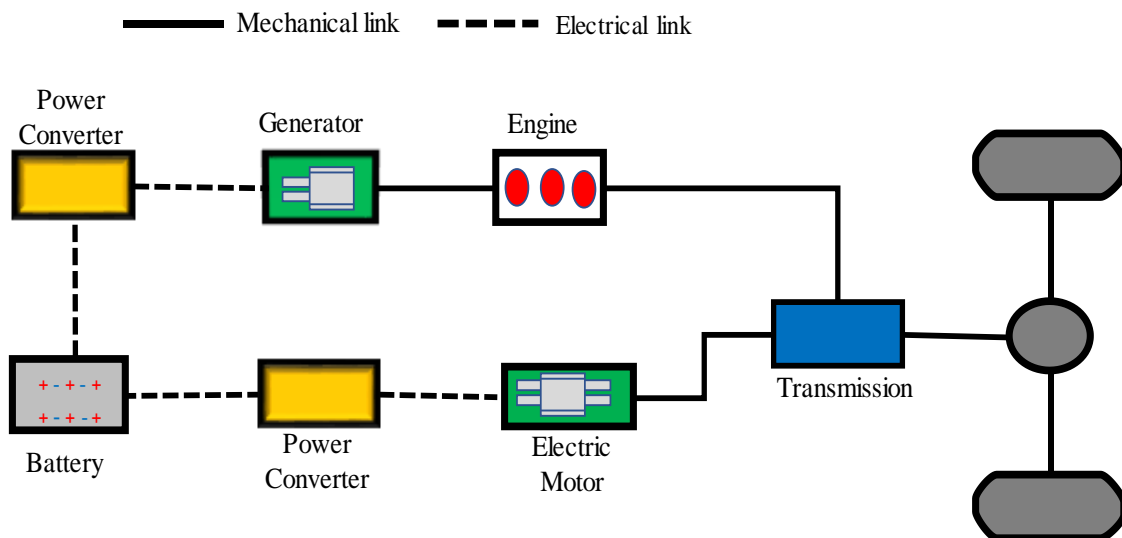


Figure 2.10 - Architecture of Complex HEV.

configure. Furthermore, the size and placement of different components (electric motor, ICE, battery, power converter etc.) becomes more complicated [40]. In addition, development and testing of control program are highly challenging and complex. Nevertheless, HEVs like Toyota Prius adopts this architecture [40].

2.4.1.3 BATTERY ELECTRIC VEHICLES ARCHITECTURE

Battery Electric Vehicles (EVs) come as a solution for energy crisis and global warming since they have zero oil consumption and zero emission and use battery to store electric energy for powering the motor [41]. Thus, BEVs have different components such as battery, power converter, electric motor and transmission driveshaft that work together for moving the vehicle. Similar with HEVs, power convertor uses the stored energy in the battery to drive electric motor. The electric motor in turn moves the transmission driveshaft that drives the wheels by means of a fixed or changeable gear and a power splitting differential gear. Depending on the motor type, rating, battery voltage, energy and power density, the power converter includes a DC-DC and motor driver. Thus, for regenerating energy, power converter should control the power flow in both directions, from the battery to the motor and vice-versa. It also requires having a battery, mainly lithium ion battery, for storing energy and this battery is mainly recharged by plug in system connected to electricity, for instance electric grid. Unlike HEVs, BEVs use power stored in their batteries. These vehicles do not incorporate the ICE motor and generator. Therefore, their range of drive depends on the battery size. The structure of BEVs is straight forward compared to that of HEVs drivetrains. A schematic of BEVs architecture is shown in Figure 2.11. Since, the battery is the only source of energy for BEVs, both the academia and the industry are working to improve the storage capacity and efficiency.

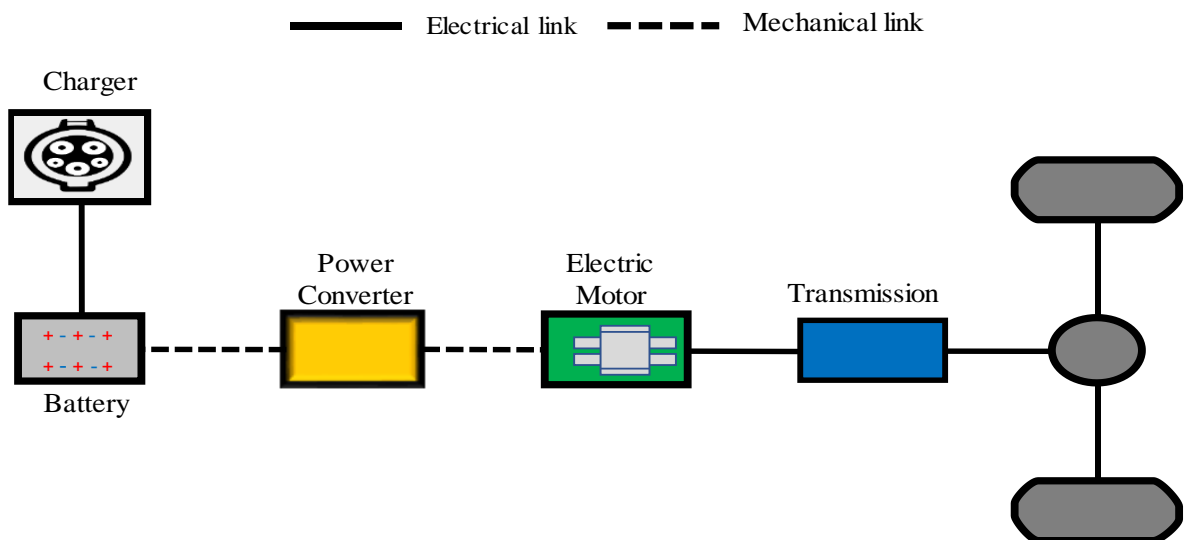


Figure 2.11 - Battery Electric Vehicle System.

However, a very high initial cost, short driving cost and charging time is considered as limitations for BEVs [42].

The basic requirement in EVs or HEVs is the use battery to store energy to supply electric energy which will be converted into mechanical energy in the electric motor for propulsion.

2.4.2 ELECTRIC MOBILITY: EVOLUTION, SUBSIDY, AND INCENTIVES

Electric based mobility is growing rapidly. Policies play an important role in the development of electric mobility. Thus, according to International Energy Agency (IEA), different policies are designed that have major impact on the deployment and development of electric mobility[8]. Thus, incentives are designed coupled with regulatory measures that boost the value of proposition of EVs; for instance, waivers to access restriction, lower toll, or parking fees. Another policy is designed to provide economic incentives, specifically to bridge the cost gap between EVs and ICE vehicles as well as to spur the early development of EV charging facilities. Moreover, tax cut is also used as a subsidiary to encourage electric mobility. Examples of countries exempting BEVs and PHEVs from circulation/ownership taxes are summarized in the list below[43]:

- In China, EVs are exempted from circulation/ownership taxes.
- BEVs weighing less than 2 tons are exempted from annual circulation tax in Denmark.
- In France, BEVs and some PHEVs are exempted from annual taxation for company cars.
- In Germany, BEVs and PHEVs are exempt from circulation tax for a period of ten years from the date of their first registration.
- In the Netherlands, zero-emission cars are exempt from paying road taxes. In 2015, this also applied to PHEVs emitting less than 50 g CO₂/km. As of 2016, they are subject to charges that are only 50% of the road tax for a regular car.
- Japan has EV exemptions from annual tonnage tax and reductions for automobile tax.
- EVs are exempted from road taxes (based on CO₂ emissions) and part of company car taxes in Sweden.
- In the United Kingdom, the excise duties starting from the second year of purchase are based on the CO₂ emissions per kilometer ratings: BEVs and some PHEVs are

exempt (this is also the case for the first year, due to purchase incentives). BEVs are also exempt from company car taxes.

- In the United States, states apply annual fees and provide tax exemptions for EVs.
- In Norway, survey results show that financial incentives such as Value-Added Tax (VAT) and vehicle registration tax exemptions, free access to toll roads and circulation tax rebates[44].

Additionally, subsidy grants are also given for the purchase of EVs. For instance, in China the National Electric Vehicle Subsidy Program grants subsidies for the purchase of electric cars. However, the level of subsidies depends on the vehicles range in kilometer, energy efficiency in kWh and battery back energy density in watt per kilogram (Wh/kg) [20]. Thus, higher energy density and efficiency with long travel range receives higher subsidy grant. Furthermore, different countries (E.g. Japan, United Kingdom,) also designs different subsidy grant schemes based on their policies and strategies.

Furthermore, some reports [20] also confirm that subsidiary and incentive policies influence on the total ownership gap between an ICE and electric car. Therefore, these and other subsidiaries, incentives and policies coupled with technological advancement, development in battery chemistry and expansion of production in the manufacturing platform and automotive industries enables the global deployment of electric mobility more popular.

Furthermore, the policy improvement certainly has a strong impact in the evolution of BEVs and PHEVs. Moreover, an improvement in battery technology has also influence on the EVs deployment. According to the Global EV outlook 2018 report [20], the global stock of electric passenger cars reaches 3.1 million in 2017 an increase of 57% from previous year (2016) as shown in the Figure 2.12 (a). This growth is similar to the growth rate of 60% 2015 and 2016. Moreover, about two-third of EVs sales in 2017 are BEVs, though the share of PHEVs is increasing in some countries. As it is shown in the same Figure, around 40% of the global EVs fleet is in China where the number of EVs passes 1 Million in 2017, while the number of EVs in European Union and United States each accounted a quarter of the global EVs share.

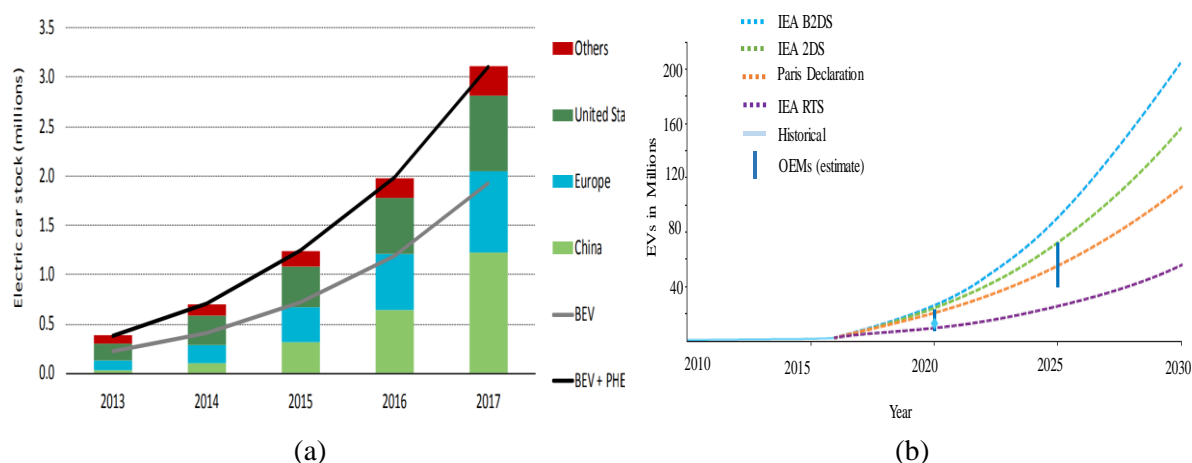


Figure 2.12 - (a) Passenger Electric Car Stock in Major Regions and (b) Deployment Scenarios for the Stock of Electric Cars to 2030 [20].

Additionally, accelerating the deployment of EVs during the 2020-30 decade will be essential for reaching the global EVs deployment rates compatible with clean mobility and decarbonization initiatives. For instance, the Paris Declaration on Electro-Mobility and Climate change and Call Action, announced at COP21-CMP11, express the ambition of exceeding the global threshold 400 Million electric two wheelers and 100 Million Electric Vehicles by 2030¹¹. As part of this agreement, governments are taking initiatives to and set EVs deployment targets as part of their clean energy and mobility plan. As it is shown in Figure 2.12 (b), the Paris Declarations forecasts the number global EVs a bit exceeds 100 Million. Whereas, the International Energy Agency (IEA) forecasts the number of EVs with three scenarios. On the first scenario, IEA RTS (International Reference Technology) that support policies announced or under construction forecast global EVs deployment of between of 40 and 70 Million by the year 2025 and between 120 and 160 Million by 2030. The IEA has also a scenario 2DS (2⁰ Celsius) aimed with a 50% probability of limiting expected global temperature increases to 2⁰ Celsius this century forecasts 160 Million number of global EVs. However, IEA's third scenario called B2SD (Beyond 2⁰ Celsius) that scenario is consistent with a 50% chance of limiting average future temperatures increases to 1.75°C compared to the pre-industrial era forecasts global EVs number beyond 200 Million in 2030.

There is considerable uncertain and much wider gap between different parties about the estimation on the number of EVs in the next 10 to 15 years. Nevertheless, the global number of market share is increasing in higher number. Besides, with the advancement of technology,

¹¹ <https://unfccc.int/media/521376/paris-electro-mobility-declaration.pdf>

EVs roaming in the streets of city centers are equipped with communication, data storage and other capabilities in order to make them communicate with each other and other roadside units for smart mobility.

2.4.3 SUMMARY

Throughout the previous subsections, we detailed different categories of EVs based on their category in using energy stored in rechargeable batteries. One of the greatest challenges for the extensive adoption of EVs and HEVs are the challenges related to battery technology and capacity. Table 2.4 provides a comprehensive comparison between the main features of the different categories of EVs presented in the previous subsections. As presented in this table, energy system is a major issue to compare electric, hybrid, and ICE vehicles. In addition, electric and hybrid vehicles provide real fuel economy and gas emission reduction compared to ICE vehicles. Since it uses two motor types of motor engines to drive, HEVs system architecture is more complex than conventional thermal ICE vehicles.

	BEVs	HEVs	FCEVs
Propulsion	<ul style="list-style-type: none"> • Electric motor drives 	<ul style="list-style-type: none"> • Electric motor drives • IC 	<ul style="list-style-type: none"> • Electric motor drives
Energy system	<ul style="list-style-type: none"> • Battery • Ultracapacitor 	<ul style="list-style-type: none"> • Battery • Ultracapacitor • ICE generating unit 	<ul style="list-style-type: none"> • Fuel cells • Need battery/ ultracapacitor to enhance power density for starting
Energy source and Infrastructure	<ul style="list-style-type: none"> • Electric grid charging facilities 	Gasoline stations Electric grid charging facilities	<ul style="list-style-type: none"> • Hydrogen • Hydrogen production and transportation infrastructure
Characteristics	<ul style="list-style-type: none"> • Zero emission • High energy difficulty • Independence on crude oil • Relatively short range • High initial cost • Commercially available 	<ul style="list-style-type: none"> • Very low emission • Higher fuel economy as compared with ICE vehicles • Long driving range • Dependence on crude oil • Higher cost as compared with ICE vehicles • The increase in fuel economy and reduce in emission depending on the power level of motor and battery as well as driving cycle • Commercially available 	<ul style="list-style-type: none"> • Zero emission or ultra-low emission • High energy efficiency • Independence on crude oil (if not using gasoline to produce hydrogen) • Satisfied driving range high cost • Under development
Major Issues	<ul style="list-style-type: none"> • Battery and battery management • Charging facilities • Cost 	<ul style="list-style-type: none"> • Multiple energy sources control, optimization, and management. • Battery sizing and management 	<ul style="list-style-type: none"> • Fuel cell cost, cycle life and reliability • Hydrogen infrastructure

Table 2.4 - Characteristics of BEVs, HEVs, and FCVs [31].

As it is shown in Table 2.5, the BEVs battery storage capacity is higher than PHEVs. However, PHEVs have internal combustion engine powered by fuel, which gives an alternative in comparison with BEVs. The major problem of BEVs is related to the battery storage capacity that makes them not to go far from their home or charging stations for charging their batteries.

However, nowadays, there is a great development in improving the capacity of the battery that in turn increase the amount of travel distance.

EV Model	EV Type	Battery Capacity[kWh]	Range
Chevrolet Volt	PEV	16kWh	64km
Smart Fortwo ED	EV	16.5kWh	136km
BMW i3	EV	22kWh	130km
Tesla S	EV	70 and 90kWh	424km
Nissan Leaf	EV	30kWh	250km
Mitsubishi iMiEV	EV	16kWh	128km

Table 2.5 - EV model, Type, and Battery Capacity of PEV [34].

Thus, the number of EVs roaming on the streets is increasing and expected to increase in the coming decades. These EVs are coming with higher computational and battery storage capacities. However, these EVs are parked for 90% of the time in the parking lot. Thus, these resources of EVs in the parking lot are unused. Therefore, in this thesis, we propose to use the resources of EVs for computation to serve constraint mobile users' demands.

2.5 CONCLUSION

This chapter presents an overview about the main concepts that are related to this thesis: cloud, edge computing, and electric vehicles. In the first part of this chapter, we presented the general characteristics and requirements of cloud-based solutions. After that, we introduced the edge-based computations paradigms: mobile edge computing, fog computing and vehicular fog computing. Then we present a short summary and comparison between them.

In the second part of this chapter, we focused on electric vehicles. We showed the three different EVs categories and their characteristics: HEVs, PEVs and ECEVs. A short and brief comparison is drawn in order to highlight the main features and differences between these three categories. In addition, we also presented about the different subsidy and incentives designed to encourage the deployment of EVs throughout the world, and we have discussed the global EVs deployment status and future forecast.

Thus, these EVs are becoming intelligent and having an increasing battery storage capacity. Since, EVs parked for longer period in the parking lot, they didn't use their resources fully. Therefore, these unused resources are wasted. Hence, in this thesis we propose to exploit parked EVs as a fog node used as computation servers. Thus, in this chapter we provided an

overview on necessary background and foundation for thesis work. We also detailed about the computation paradigms at the edge of the network. Yet, different applications are implemented using VFC architecture. There are issues to be covered in using VFC for computing local mobile users demand. Hence, in this thesis we tackle the problem of resource allocation in VFC for resource constrained mobile devices.

However, in VFC resources are typically constrained and dynamic compared with cloud computing. This makes resource allocation a real challenge that needs to be addressed. Therefore, in the next chapter, we explore in more detail the state of the art on the problem of resource allocation in cloud and fog computing. It also provides a comprehensive review on resource allocation approach, as well as some of the techniques used for VFC based resource allocation. Then, we discuss the related gaps and introduce our contribution in the area.

RESOURCE ALLOCATION IN THE CLOUD AND FOG COMPUTING: STATE OF THE ART

3.1 INTRODUCTION

As presented in the previous chapter, cloud computing is a computing paradigm where a large pool of resources is connected in private, public, community or hybrid network. Thus, it provides on demand access on the basis of pay-as-per-use model with a determined quality of service to the users. However, cloud servers are located in distance location from the origin of the computation demand. These create a jitter and delay for performing delay sensitive application.

Specially, in the emerging era of Internet of Things (IoT), the exponential growth of smart devices with constrained resources leads to a growing need for efficient resource allocation mechanisms. Earlier, cloud computing was used as an efficient approach for storing and computing huge amount of data generated from such devices. Nowadays, cloud computing is unable to serve billions of smart devices requests efficiently due to bandwidth limitations, latency, etc. Therefore, in order to solve the limitations of cloud computing, a new approach is introduced called fog computing. In fog computing the computation is done closer to the users.

Thus, as presented in chapter 2, fog and vehicular fog computing (VFC) paradigms were proposed to push the computing capability of cloud service to the edge of the network. In fact, the advancement in the automotive industry and in Information and Communication Technology (ICT) enables the introduction of vehicles with computing, communication, and storage capabilities. Thus, these vehicles own the capabilities to serve local users demand. Hence, constrained mobile devices could offload their tasks to nearby vehicles that serve as fog nodes and when no resource is available, they are redirected to the cloud. Therefore, in this

chapter of the manuscript, we aim to provide the state of the art on resource allocation strategies on cloud, fog computing and vehicular fog computing. Thus, in section 3.2, we present the resource allocation concept: the strategies of resource allocation mechanisms in cloud environment. In section 3.3, we present the resource allocation in the fog computing paradigm. In section 3.4, we present the resource allocation in vehicular fog computing environment. Then, in section 3.5, we detailed discussion on resource allocation. Finally, section 3.6 concludes this chapter.

3.2 RESOURCE ALLOCATION IN THE CLOUD

As presented in the chapter 2 of this manuscript, we detailed about cloud computing characteristics, service, and deployment models. Thus, cloud servers have significant resource for computation and storage. These resources are dynamically allocated for users as per their demands. Hence, the objective of this section is to review the existing researches and provide an overview about cloud computing resource allocation and different algorithms. Therefore, in cloud computing, resource allocation is a task that takes into account many computing conditions such as datacenter management, operating systems, energy usage, etc. In addition, resource allocation deals with the division of available resources among different cloud user's applications in an efficient and effective way. Therefore, cloud computing delivers various services and resources are leased for cloud users based on their dynamic needs. Thus, the goal of resource allocation in cloud computing is to allocate cloud resources that meet the demands of the cloud users.

As shown in Figure 3.1, cloud computing platform consists of two parties: the consumers and cloud service providers [45]. On the one hand, from cloud user's viewpoint, resource allocation must be achieved with a lower cost with minimum time. On the other hand, cloud provider aims to deliver cloud services that meet users' requirements with lower cost. However, since the available resources are limited with resource diversity, locality restrictions, dynamic nature of resource requests, and environmental necessities, there is a need for an efficient and dynamic Resource Allocation Strategy (RAS) that is suitable for cloud environments in order to provide all requested resources on time. Moreover, cloud service resource allocation includes both physical and virtual resources on the cloud servers such as CPU cores, storage devices, and network bandwidth sliced and shared between virtual machines potentially running different

application workloads etc. [45]. Therefore, in such complex cloud environment scenarios, resource allocation is the main concern.

There are two major processes of cloud computing resource allocation: static and dynamic resource allocation. With the static approach, the idea is to assign fixed resources to all cloud users or applications. Hence, cloud users know the amount of resource allocated for them, so that they should monitor application's peak load request. The main limitation of this technique is under or over utilization of resources with regard to the applications workload. However, dynamic resource allocation provides cloud resources when cloud user or application is requested. This approach helps to avoid over or under resources utilization. There are different resource allocation algorithms and techniques in the literature for allocating cloud resources.

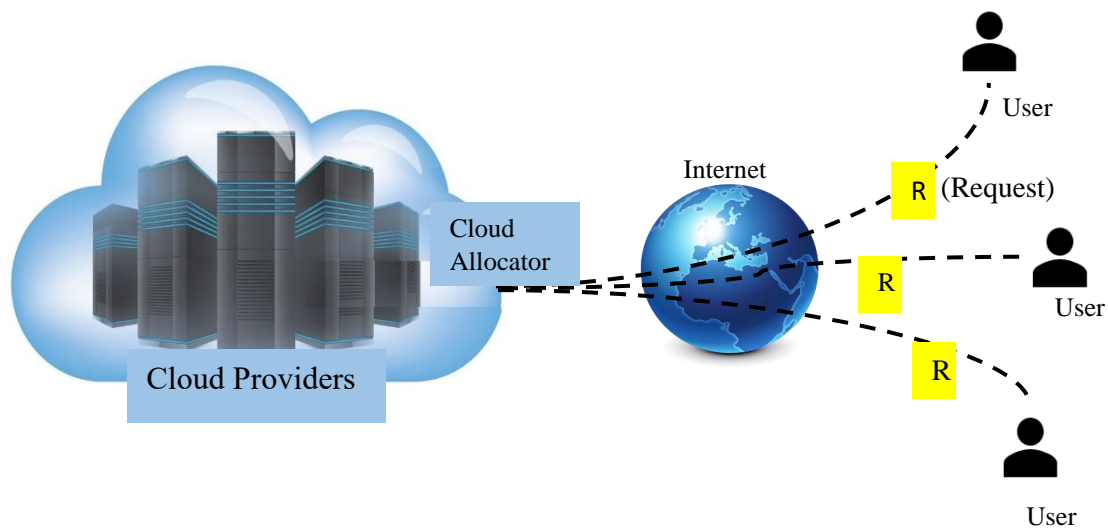


Figure 3.1 - Resource Allocation in Cloud Computing.

Furthermore, a resource allocation strategy in cloud computing aims on granting that the physical and/or virtual resources are delivered appropriately to end users [46]. There are a number of researches done on resource allocation in cloud computing platform. Thus, cloud resources are allocated considering different parameters, for instance, Quality of Service (QoS), Service Level Agreement (SLA), energy and power consumption, etc. Moreover, the resources allocation is done based on various scheduling algorithms.

Therefore, in order to enhance cloud service and increase the cloud service performance and efficiency, the resource allocation should be done optimally. An optimal resource allocation strategy should avoid the following conditions [47]:

- **Resource contention:** this situation arises when multiple applications try to access the same resource at the same time.

- **Scarcity of resources:** this occurs when the application request is high and there are limited resources.
- **Resource fragmentation:** this situation happens when the resources are isolated. There will be enough resources but not able to allocate to the needed application.
- **Over provisioning:** this arises when the application gets surplus resources than the demanded one to fit the QoS requirements.
- **Under provisioning:** this occurs when application is assigned with fewer numbers of resources than the demand requested to fit the QoS requirements.

Thus, Figure 3.2 shows different algorithms that aim to increase the QoS in cloud computing resource allocation strategies such as virtual machine migration, game theoretic, bio-inspired, priority based and service level agreement methods. In the following, we address some of the most commonly used strategies related to cloud resource allocation:

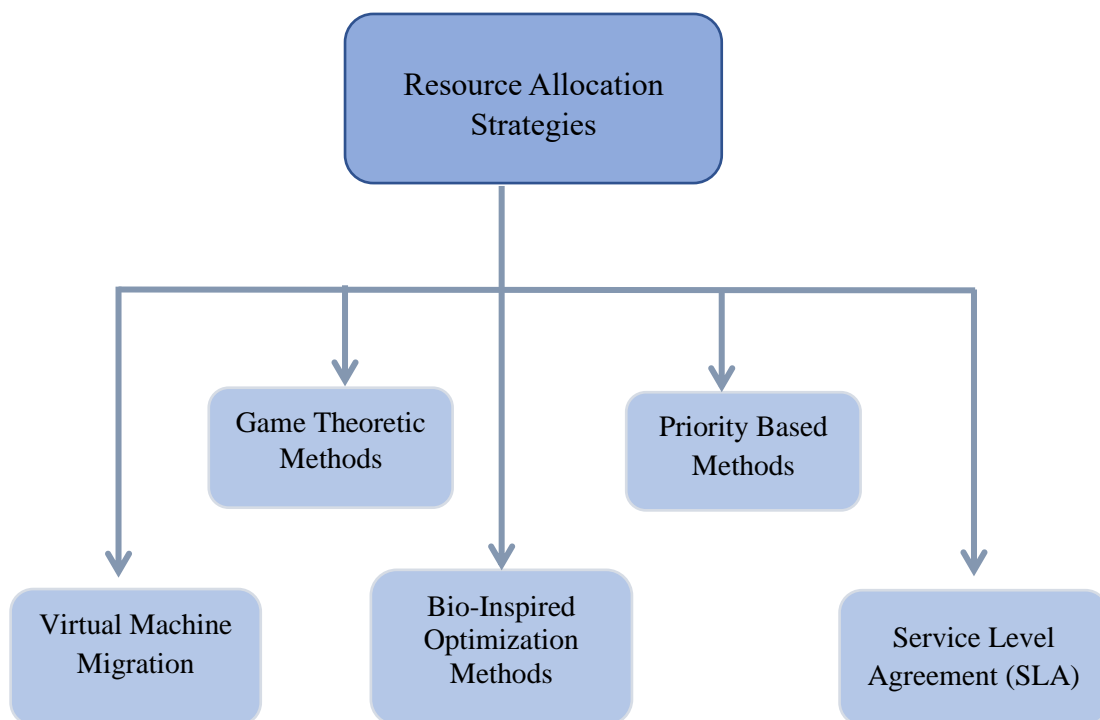


Figure 3.2 - Resource Allocation Strategies in Cloud Computing.

- **Virtual Machine:** virtualization is creating a virtual image of the physical device such as storage device, operating system, or other processing component of a system. Thus, in cloud computing, cloud resources are divided/sliced into one or more execution environments [48]. For instance, a single server can be mapped into multiple logical partitions. In virtualized environment, logical resources are created, increased, reduced, moved, or migrate in order to meet dynamic users demand. Thus, virtual machine

migration provides incredible advantages, for instance, load balancing, server solidification, online maintenance and proactive adaptation to non-critical failure, etc. [46]. Some of the authors use virtual machine for resource allocation in cloud service. For instance, in [49] uses virtual machine to dynamically allocate multimedia cloud resource based on application's QoS demands and support energy and cost savings by optimizing the number of servers in use. In addition, in [50] the authors propose dynamic migration optimization algorithm based on trusted computing technology to put forward secure virtual machine migration method in cloud environment.

- **Bio-inspired Optimization Methods:** Load balancing is the major activity in cloud computing environment in order to protect under or over provisioning of cloud resources. Thus, researches are done by using bio-inspired optimization techniques for managing the usage of virtual machines in cloud servers. For instance, in [51] the authors propose a particle swarm optimization load balancing algorithm in order to design an efficient scheduling algorithm which addresses different constraints such as heterogeneity, reliability, delay, etc. with low response time. They proved that the efficient utilization of resources without under/over utilization of resources. Likewise, the authors in [52] proposed to use a firefly algorithm as a bio-inspired technique in order to solve the problem of load balancing that reduces energy consumption in cloud servers. Thus, with this technique, they showed that the energy consumption parameters are more effectively optimized using the selected algorithm.

- **Priority Based Methods:** scheduling of tasks and resources by adopting an efficient task scheduling approach increases the efficiency of the cloud servers and QoS for users. Thus, scheduling tasks in cloud environment is done based on the selection of best suitable resources for tasks execution based on the users' QoS parameters. In addition, applications are becoming complex and require to leverage the computing power of the cloud for parallel computing [53]. Thus, it is important to manage resources of cloud environment to serve these complex applications. Therefore, the authors in [53] use a priority-based algorithm for consolidating parallel workloads in cloud environment. They use virtualization technique for partitioning the computing capacity of each node into two tires, the foreground and background virtual machine tires. To such purpose, they adopted and proved a scheduling algorithm to make efficient use of the two-tire virtual machines responsiveness. In addition, in [54] authors

proposed an improved priority based job scheduling algorithm in cloud computing environment. Thus, improved priority-based job scheduling algorithm uses iterative method to find priority of jobs and resources and also finds priority of jobs to achieve better performance. Finally, the proposed algorithm has better consistency than priority-based job scheduling algorithm and prioritized round robin algorithm.

- **Service Level Agreement (SLA):** SLA is a contractual agreement that specifies the QoS between cloud service provider and users about services that are delivered to users. Initially, SLA were used in networking but currently it is widely used by telecommunication service providers and cloud computing service providers [55]. Thus, SLA algorithm is mainly proposed to software as a service provider who wants to reduce the infrastructure cost and SLA violations. SLA helps to monitor the dynamic change of users' request and creating mapping between the users' requests with the available cloud resources. System performance is measured from both users and service providers point of view [46]. From customers' perspective, observed how many SLAs are violated and from cloud service providers' perspective, observed how much the cost is reduced.

- **Game Theoretic Methods:** resources in cloud environment seems infinite, but in reality they are limited and cloud servers that have limited capacity to serve applications and data [56]. Thus, cloud users are competing with each other in order to have all the necessary resources for applications, data storage and other services. In order to make fair resource allocation between users, researchers use game theoretic based resource allocation in cloud environment. The idea is to design and implement mechanisms that allow users to share in a transparent way among different users. For instance, in [56] propose an approach for resource allocation in a cloud computing environment where different users share the resources of several cloud providers. Their results guarantee a better availability of resources in the cloud, and fair allocation of available resources among cloud users. In [57], authors propose a non-cooperative game theory mechanism in an incomplete information environment to compare and select resource providers based on their best bid price.

As presented on the previous section, resource allocation is one of the most important tasks in cloud computing environment. Furthermore, it consists of identifying and assigning cloud resources to each incoming user request in order to meet their QoS preferences. A comprehensive survey of optimization techniques used for energy aware resource allocation in cloud computing is reviewed in [58]. Most of the reviewed papers focused on minimizing the consumption of energy in the cloud data centers. However, some of the authors tried to address an issue related to the performance and resource utilization in the cloud platform [58]. Therefore, the main issue in cloud resource allocation is to reduce energy consumption while maintaining QoS.

3.3 RESOURCE ALLOCATION IN FOG COMPUTING

As presented in chapter 2 of this manuscript, fog computing is a new paradigm of computing that brings cloud computing capabilities to the edge of the network in order to better support time dependent, location aware, massive scale and latency sensitive applications. In addition, fog computing provides resource constrained, possessing limited computation capabilities and limited battery storage capacity that make impractical to run sophisticated applications on them. Thus, fog computing requires fog nodes at the edge of the network to speed up data processing. Fog devices are the devices which have additional capability of storing and processing the data from the end devices such as router, set-top-box, etc. However, the management of these additional computing and storage devices requires new approaches. Thus, researches are done focusing on fog-based system to be properly analyzed for effective management of fog servers. Moreover, there is cooperation between the cloud and fog computing environment in which tasks that require higher computation demand will be transferred to the cloud. Hence, resource allocation in fog computing is done on multi-tier approach as shown in the Figure 3.2. As it is shown on the same Figure, when the user moves from the area coverage of one fog node, the connection is transferred and established with other fog node.

Resource allocation, as detailed on the previous section, is also one of the major issues in fog computing paradigm. Managing resources from smart devices is a complex task that requires novel mechanisms and techniques to solve this problem. Thus, different researches are proposed to investigate resource allocation that involves fog nodes.

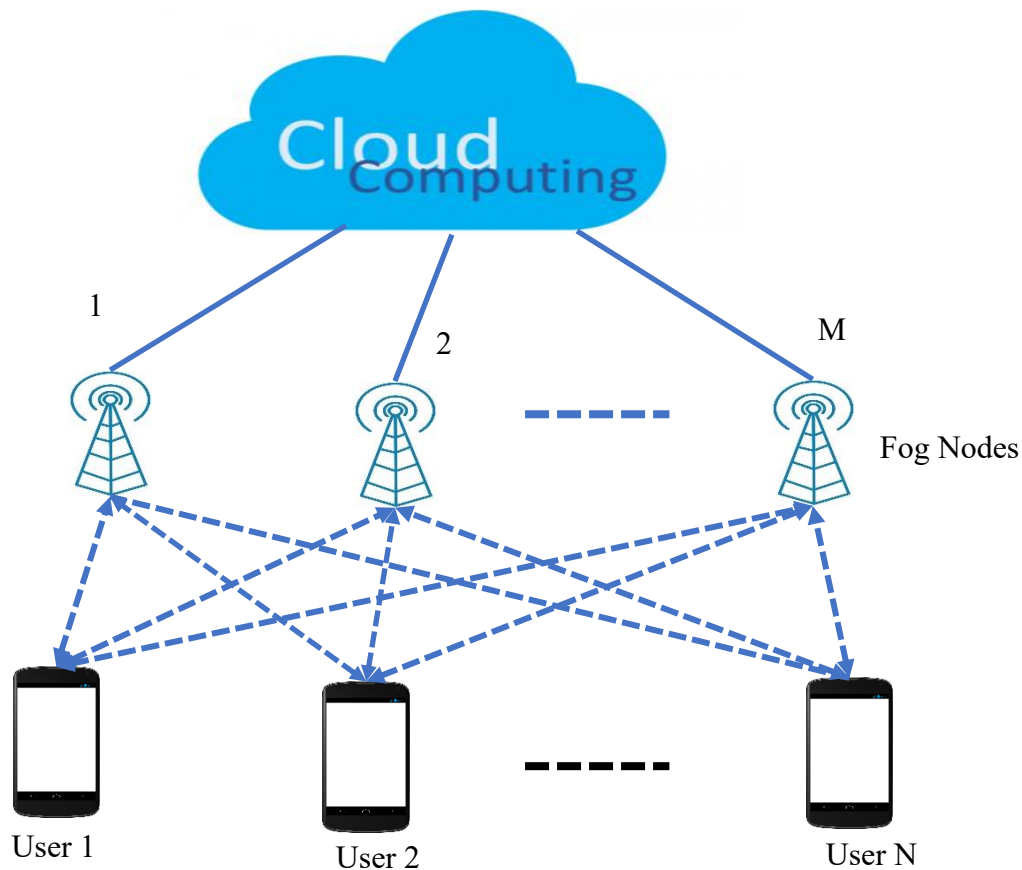


Figure 3.3 - Resource Allocation in Fog Computing.

Authors in [59], for example, proposed a coupling resource management based on fog computing in smart city system. In order to overcome the problem, they propose a fog computing model and extended Hungarian algorithm to manage the coupling resource.

Compared to cloud computing, the fog computing does not have enough computing and storage capacity. Thus, efficient resource allocation is an important issue in fog computing. Therefore, different works were done on resource allocation in fog computing paradigm. Moreover, due to its unique features, the resource allocation task considers the following two issues:

- **Mobility:** due to the dynamic nature, the users and fog nodes may join or leave a fog layer arbitrary. Therefore, dynamic load balancing and resource allocation to mobile users and devices are very challenging tasks due to time variant resource availability in fog layer [60]. Therefore, how the load balancing is done considering delay, power consumption, and bandwidth resource sharing should be studied. In addition, it is also

important to understand the mobility pattern of end users since it can help for task assignment and resource management in fog computing.

- **Virtualization:** small scale data centers at the edge of the network can benefit in different features, for instance, better QoS and lower delay. In addition, virtualization also helps to support and improve different service requirements and objectives of users' computation demands [60].

Therefore, different researches were realized on resource allocation in fog computing. Some of the related researches on resource allocation and scheduling in fog computing is summarized and presented in [61]. Thus, most of the researches focus on resource allocation and others on energy efficiency and load balancing in fog computing environment. For instance, in [62] authors proposed a fog community based approach for the problem of initial placement and continuous migration of fog services in order to provide follow-me experience for the users. Their proposal is an adaptation of fog principles of previously proposed community-based resource management framework for cloud computing. Finally, their experimental result proved the reliability of the proposed strategy in fog services. In addition, the authors also showed that the initial placement of the fog virtual machines could be done in an optimal way from low to medium and partially heavy loaded infrastructures.

3.4 RESOURCE ALLOCATION ON VEHICULAR FOG COMPUTING (VFC)

As we presented in the previous chapter, VFC has a huge abundant computational potential capacity by utilizing parked vehicles as a computing infrastructure. In particular, for applications that require higher computational capacities such as resource constrained mobile devices, the VFC infrastructure plays an important role in speeding up the computation task and reducing delay. However, resource allocation and management in VFC is an important concern in order to utilize the full advantage of available VFC resources. The resource utilization is focused to manage the available resource and minimize the cost for computation such as how the user and network operator interact and how the computational tasks are managed and distributed among numerous nearby vehicular resources [63]. Since, VFC is formed by aggregating an abundant resource of vehicles, an efficient cooperation and integration of those vehicles should be designed. However, when the VFC does not have enough resources (overloaded or there are no vehicular nodes), the workload can be offloaded to the next layer (cloud).

Some approaches are proposed to show how VFC resources are used for computation purpose. For instance, in [64] the authors present mobile users resource management in VFC, that aims to resolve the admission control problem for applications with different QoS requirements and dynamic vehicle resources. To solve this problem, a theoretical game approach is formulated and new scheduling algorithm for dynamic resource allocation is presented. Their experimental results achieve efficient QoS and scale well as the system size increases and keeps minimal energy consumption level for the vehicles mobility.

In [65], the authors propose to use public service vehicles such as buses as fog servers to perform delay sensitive tasks. However, the availability of resources and users demand arrival are dynamic overtime. Therefore, a semi-markov decision process (SMDP) based offloading policy is proposed to describe the features of the available resource and application aware delay requirements. Thus, the formulated SMDP aimed at maximizing the long term expected reward of VFC system and meet application aware delay requirements. Their simulation results show that the proposed offloading policy can obtain a significant expected reward performance. Authors in [66] propose to use VFC as a solution to relieve the problem of overload happen on base station and reduce delay of processing requests at peak-times. Their aim is to offload the computing demands from the base station to vehicular fog nodes by taking advantages of the unused resources of nearby vehicles. However, the wide area deployment of VFC still have challenges such as lack of incentive and task assignment mechanisms. Therefore, to address these challenges, at first efficient incentive mechanism based on contract theoretical modeling, then the pricing-based stable matching algorithm was used for task assignment problem into a two-sided matching problem between vehicles and user equipment's (UEs) is proposed. Their simulation results prove that the proposed incentive approach archives a social welfare that is close to optimal performance, while the proposed task scheme is able to achieve network delay close to optimal with a much lower complexity.

Based on the above literatures, we observe that resource allocation in vehicular fog computing paradigm is mainly designed to reduce the computation time and enhance QoS for local mobile devices. As stated previously, electric vehicles, as a solution for the problem global warming and energy crises, are becoming popular and in the coming decades widely used in the world. In addition, these EVs are coming with computation, storage, and communication capabilities. However, EVs with such capabilities spent most of their time parked on the parking. Thus, in this thesis, we propose to use these abundant resources of EVs for serving constrained mobile

users demand by considering the available resource on parked EVs. We focus essentially to use the abundant resources EVs with optimization techniques and models that minimize the efficiency of computation delay and energy consumption of nearby mobile devices.

3.5 DISCUSSION

Mobile devices are facing computational and storage problems due to the development of resource intensive applications. In recent years, the computational capacity and other resources of mobile devices are improved. However, they are still facing a great challenge in running rapidly growing application demands. Energy management and delay optimization is the crucial problem of mobile devices. This motivates most of the researchers to focus on managing energy and time required to achieve a given task. Therefore, the solution for such problem is to offload their computational demand to powerful devices. Computation offloading takes the advantage of available resource in cloud/fog to overcome the limitations of mobile devices [67]. Such solutions, as presented in chapter 2 of this manuscript, have significantly supports the computational capabilities of constrained devices.

However, the response times may be significantly determined by the number of offloading devices. Thus, due to network bandwidth limitation, when a lot of devices attempt to offload their computational task, network congestion is generated which leads to extra delay [68]. Moreover, offloading large data sizes that needs to be processed to distant server greatly affects data transmission time and energy. Hence, in this case, it might be better to process locally.

As presented in the previous sections, different algorithms for instance game theory is used for resource allocation task. Initially, game theory was developed for solving problems in economics, but nowadays it has been used to solve problems in different areas such as political science, psychology, logic, and computer science [69]. For instance, in order to grip the current and future challenges of wireless and communication network and an increasing popularity of mobile devices, game theory is used as the key tool [70]. This is mainly due to the need to add rules and procedures for decision making purpose for future generation wireless and communication networks.

In the other way, Markov Decision Model (MDP) is also used to model offloading task of constrained devices. For instance, in [71] authors presented an offloading problem as MDP and derive a near-optimal offloading policy. In order to handle the dynamic arrival of tasks, they use stochastic method. Likewise, in [72] authors propose a hybrid approach using MDP for better prediction of next most suitable offloading server among multiple available servers.

In this thesis, we focus on using the resources of parked EVs for computation. The resource constrained mobile devices offload their tasks to the nearby EVs using different offloading strategies. Hence, we develop and evaluate an optimization algorithm for resource allocation to accept or reject users demand in EVs fog nodes using MDP by keeping sufficient energy for mobility.

We also extend the MDP based resource allocation approach, in order to incorporate the delay and consider dynamic aspects of both the users and available resources of parked EVs. We provide a new model of resource allocation approach, called stochastic theoretic game model for vehicular fog computing. The proposed solution is able to incorporate both the dynamic number of users, resources in VFC, time and energy constraints.

3.6 CONCLUSION

This chapter provides a comprehensive summary for the state of the art about resource allocation on cloud, fog, and vehicular fog computing environment. At first, we have presented the essential information about the resource allocation task in cloud computing. Then, we describe the widely used strategies for resource allocation in cloud environment and discussed about the energy efficient resource allocation algorithms.

Furthermore, we have presented the resource allocation in fog computing paradigm. Then, we have provided an overview about the resource allocation approaches in vehicular fog computing environment, in which resources of vehicles are used to serve constrained devices. Finally, we discuss about offloading computation intensive tasks and resource allocation of constrained mobile devices is presented.

In vehicular fog computing the resources of parked or moving vehicles could be used for computation, communication, or storage purpose. Yet, using vehicles for serving users demand is a challenging and new scenario must be considered. Therefore, different from other research's, in the next chapter we present to use electric vehicles resource to serve constrained mobile devices while keeping minimal energy level for vehicles mobility.

Hence, we introduce in the next chapter the battery management system for electric vehicles which is our first solution to use electric vehicles resources for serving constrained mobile devices computation demand.

EFFICIENT BATTERY MANAGEMENT FOR ELECTRIC VEHICLES

4.1 INTRODUCTION

As presented in the previous chapters, EVs attracted widespread interest for having unique features such as environmentally friendly, low emission, quiet operation, etc. Since the 1990s, technological advances in the field of batteries, particularly for portable devices such as laptops and mobile phones, have encouraged electric vehicle (EV) manufacturers to improve the performance of their batteries [73]. The advancements in battery technologies have been key factor for the growing success of the EVs. Hence, a continued transition to electric drive will surely generate a great efficiency in the battery production.

Batteries play a very significant role in different devices. For instance, consumer electronic devices, electric vehicles, and standby capabilities in emergency backup systems use batteries as a main source of energy [74]. Nowadays, many industrial applications use chemical batteries as a main source to store energy. Specially, EVs use chemical batteries as their primary energy storage mechanism that helps to store the necessary required higher energy for driving and also for other consuming tasks that are used in the current and upcoming EVs.

Furthermore, as we have presented in chapter 2 of this manuscript, EVs are getting technological advancements with an improvement of the battery efficiency and storage capacity. As a result, the number of EVs roaming the roads is increasing, especially in city centers. On the one hand, these EVs are equipped with advanced communication, computation and data storage technologies spend most of their time in parking lots. On the other hand, mobile devices with constrained energy are massively produced. In order to support these constrained mobile devices, we propose in this chapter to use the abundant EVs resources available in the parking lot for computation. However, using EVs battery for computation, in addition to their mobility, may have an impact on EVs battery lifetime. Therefore, EVs used

for computation purpose should be selected with the battery health and energy level among the parked vehicles. Moreover, the amount of energy used for computation will be limited to the available energy above the threshold. In this context, EVs owner will have an energy for their mobility and mobile users will use of surplus battery of parked EVs. Hence, it is important to propose mechanisms to manage this energy for serving local mobile users demands.

Using EVs energy frequently increases the charge/discharge cycle that have an impact on the SOH (State of Health) status. Therefore, in this chapter, we tackle the problem of using EVs battery for computation in consideration with their SOC (State of charge) and SOH values. The former represents a measure of the battery's ability to store and deliver electrical energy, compared with a new battery, where 100% represents that the battery is fully charged and 0% represents the battery is fully discharged. Whereas, the latter is defined as the ratio of maximum charge capacity of an aged battery to maximum charge capacity when the battery was new, before any cycling [75]. Therefore, SOH is an important battery performance measurement tool in EVs, as it leads to maximize the utilized energy from the battery. The main difference on SOH and SOC estimation is that the change in SOH estimation happens slowly due to the irreversible physical and chemical changes that takes place due to the usage of battery, whereas, SOC is changing with charging and discharging cycles.

However, there are different factors that adversely affect the state of EV's battery: overheating, deep charge, overcharging or high voltage, to mention some of them [76][77]. These factors also have a profound effect on the life of the battery. Therefore, to improve the life span of the battery and solve the greater safety issues of batteries (e.g. Li-ion batteries), EVs have a system that regulates and controls the overall activities of the battery, called Battery Management System (BMS), including charging and discharging as well as monitoring the overall temperature of the battery system [78]. Thus, BMS monitors the system to ensure the efficient use of the battery; a continuous watch on its parameters such as SOH and SOC are good experience. In this chapter, we propose to consider these two parameters for EVs resource allocation.

This chapter is set out in the following manner, in section 4.2, we present the problem statement and description. In section 4.3, we give a background information on EVs battery technologies. Then, in section 4.4, we present the description of the proposed solution. Finally, we present discussions and concluding remarks, in section 4.5 and 4.6, respectively.

4.2 PROBLEM STATEMENT AND DESCRIPTION

EVs battery system is a key component and strongly influences the driving performance and distance covered per charge cycle [79]. In every application where batteries are deployed, the state of the battery is critical to ensure that the required energy is available [75]. As we presented in the previous chapters, EVs are coming with higher battery storage and equipped with computation and communication capabilities. However, these EVs with abundant resources are parked for extended period of time. Therefore, the main idea in this thesis is to use this abundant resource of EVs available in the parking lot. However, using EVs frequently increases the battery charging frequency that has an impact not only on the state of health status but also on the battery lifetime. In addition, we assume parking lots that do not have charging infrastructure. Thus, the main problem is to use available EVs energy for computation and mobility.

Therefore, the main problem to be resolved, in this chapter is: how to use surplus EVs resource for computation while keeping some amount of energy for vehicles mobility in consideration with its battery health status and charge level? Thereby, we built this chapter to answer this fundamental question. We consider the current status of SOH and SOC values before using EVs for computation. Therefore, we can categorize EVs battery with their SOH condition and SOC levels. In this way, we guarantee that EVs with lower SOH condition and SOC levels will not be used for computation. Since, we are using EVs energy to serve time and energy constrained mobile applications, we chose EVs battery with higher SOH values in order to serve them. However, serving the users demand should consider the battery level and state of health condition.

In the next section, a general overview about different EVs battery technologies with their storage capacity and different EVs battery modeling methods are detailed.

4.3 OVERVIEW OF ELECTRIC VEHICLES BATTERY TECHNOLOGIES

EVs are widely used and their deployment is forecasted to increase in coming years. They are categorized into different groups that have different characteristics presented in chapter 2 of this manuscript. Batteries are used as one of the main components for these EVs. Thus, in this section we detail different electric vehicles battery technologies, followed by their comparison.

4.3.1 TYPES OF EV BATTERY TECHNOLOGIES

In this subsection, we introduced an overview on different EVs battery technologies. We first present about the lead acid battery technology. Followed by Nickel metal hybrid battery and sodium nickel chloride battery technologies. Finally, we present the widely used and promising battery technology in EVs is detailed (Lithium ion battery).

i. LEAD ACID BATTERY

Lead acid battery is one of the oldest battery technology used in electrical power system for more than a century [80]. Thus, this battery technology is matured, and its potential improvement is very low. In addition, the energy-to-weight and energy-to-volume ratio is very low. Therefore, this kind of battery does not have the capability to provide the required energy and power with limited space of EVs. As shown in Figure 4.1, this battery technology has low energy storage capacity compared with other technologies. Furthermore, this battery technology has also limitation of low cycle life and low energy density [80].

ii. NICKEL METAL HYDRIDE BATTERY (Ni-MH)

A nickel metal hydride battery, abbreviated NiMH, or Ni–MH, is a type of rechargeable battery. The chemical reaction at the positive electrode is similar to that of the nickel–cadmium cell (NiCd), with both using nickel oxide hydroxide (NiOOH). However, the negative electrodes use a hydrogen-absorbing alloy instead of cadmium. A NiMH battery can have two to three times the capacity of an equivalent size NiCd, and its energy density can approach that of a lithium-ion battery. The major advantage from a manufacturing point of view is the safety of NiMH compared to Li-ion batteries, and, so far, no incidents have been reported in the press [40]. Furthermore, NiMH batteries are preferred in industrial and consumer applications due to their design flexibility (e.g. ranging from 30 mAh to 250 Ah), environmental acceptability, low maintenance, high power and energy densities, cost and most importantly safety (in charge and discharge modes, especially at high voltages) [40]. This technology has been recently used in Hybrid Vehicles (such as Toyota Prius for instance) [81].

iii. SODIUM NICKEL CHLORIDE BATTERY

The sodium nickel chloride battery, also known as the “zebra” battery, is another high temperature battery system. Sodium nickel battery cell uses sodium and nickel chloride

electrodes [80]. It is considered as safe and low cost compared to Li-ion battery, with longer life cycle [81]. Furthermore, higher cell voltage and the ability to resist fully discharging or over charging without degrading its life expectancy [82]. Although, this technology is studied extensively because of its inexpensive material, higher specific energy, and power. The zebra battery has problems such as life-cycle, safety and hazards to the environment [83].

iv. LITHIUM-ION BATTERY

Lithium-ion, also referred as Li-ion, is a battery technology that already dominates the cell phone and laptop markets. It rapidly becomes a standard for HEVs and EVs because of its higher specific power and energy [83]. As shown in Figure 4.1, the Li-ion battery achieves higher specific energy and power in comparison with other battery technologies. A Li-ion battery stores more energy capacity than sodium nickel chloride battery [40]. Thus, Li-ion is a powerful candidate of energy storage for EVs with its long lifecycle, higher energy density and low self-discharging rate [84]. However, it suffers from major issues such cost, operational temperature range, environmental impact, and safety. In addition, material availability, for example Li, concerns have been raised [40]. Furthermore, flammable liquids limited life cycle and battery life decreased with high temperature are the limitations of this battery technology [84]. Therefore, lithium-ion systems must be maintained within specified range of operating temperature in order to avoid permanent cell damage and failure [80]. Li-ion battery is the most widely used technology in advanced electrified vehicles and developed to meet different specification, each with different chemical compositions. Thus, the Li-ion battery is the most feasible battery used to attain fair and effective transportation to sustainable global EVs development. However, this battery is expensive in comparison with other battery technologies. Despite that, Li-ion is still a preferred choice for most of hybrids and battery EV [76].

There are varieties of Li-ion battery chemistries, with several features and degree of maturity. The detailed advantages and disadvantages of the technology are presented in Table 4.1. More general state of the art of lithium ion batteries with their current and future status are discussed and presented in [85]. Among all different battery technologies, Lithium-ion (Li-ion) batteries are a preferred choice for EVs in the global market since it has high power rating, high energy density, and high cycle life [86].

Technology	Advantages	Disadvantages
Lithium Cobalt Oxide (LiCoO ₂)	Power and energy density	Safety, cost
Nickel Cobalt and Aluminum (NCA)	Power and energy density, calendar, and cycle life	Safety
Nickel Manganese Cobalt (NMC)	power and energy density, Cycle, and calendar life	Safety
Lithium Polymer (LiMnO ₄)	Power density	Calendar life
Lithium ion phosphate (LiFePO ₄)	Safety	Energy density, calendar life

Table 4.1 - Comparison of Lithium-ion Battery Technologies [81].

4.3.2 BATTERY ENERGY STORAGE CAPACITY

In this subsection, we describe the features of different battery technologies for EVs applications such as specific power/energy, storage capacity, etc. EV batteries are a long established means of storing electricity in the form of chemical energy [80]. They are electrochemical devices. Currently, the biggest challenge is the selection of appropriate battery type and battery size for EVs. For instance, plug-in HEVs provide higher battery storage compared to HEVs by incorporating a larger battery [77]. Therefore, this large battery reduces substantial amount of fuel for the engine that in turn substituted with electricity charged from power grid. The battery units must be sized in order to store more energy (kWh) and provide required amount power (kW) for driving the vehicle [77]. Also, battery should meet appropriate charge and discharge cycle and calendar life. With all battery chemistries, there is a tradeoff between the energy density and usable power density of the battery [87].

The amount of electrical energy is expressed either per unit of weight (Whkg^{-1}) or per unit of volume (wh^{-1}) that a battery is able to deliver. For instance, Figure 4.1 shows the volumetric energy density (Wh/kg) with respect to gravimetric energy density (Wh/kg) for different batteries categories. The goal of this Figure is to show how much energy is stored per kg of the battery pack. Hence, it can be seen that among other technologies, lithium-ion batteries perform better, this is due to the battery pack store more energy density and specific power into a smaller battery package. For example, the volume and weight savings (about 60%) over a Ni-MH battery means less weight and more space for comfort in the vehicles [88].

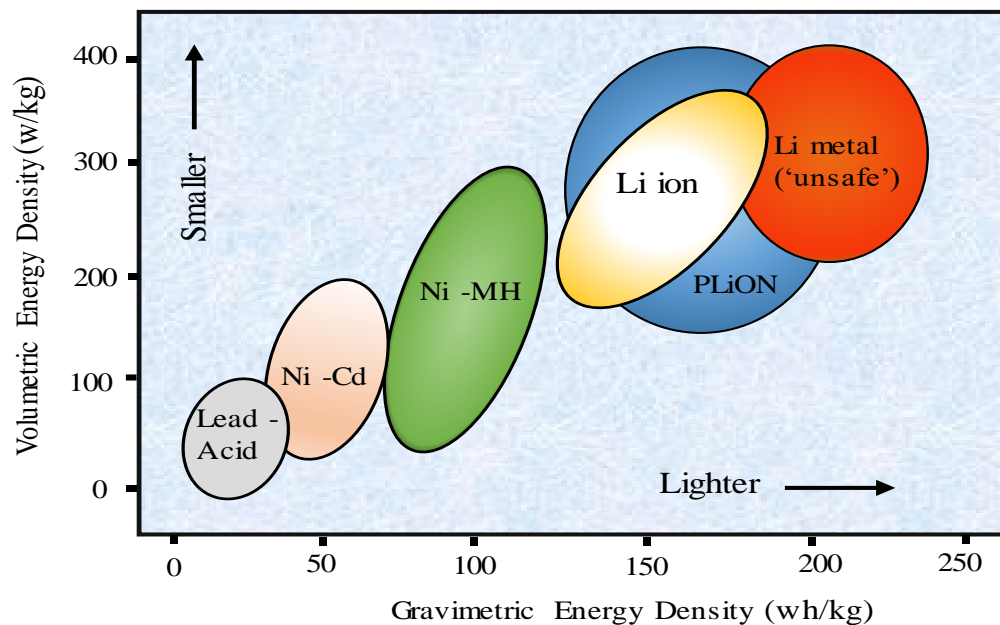


Figure 4.1 - Volumetric Energy Density vs. Gravimetric energy of Common Battery Technologies [90].

It is also important to show the relationship between the car range and battery capacity, since adding additional weight of battery reduces the efficiency of EVs on the road. Therefore, it is important to compare batteries according to their energy and power densities. Thus, the Ragon plot, in Figure 4.2, illustrates the range of specific power and specific energy for different battery technologies. It specifically shows the specific power vs. specific energy for various electrochemical storage batteries. The specific energy represents the total energy a battery can deliver in watt-hours per kilogram (Wh/kg) and specific energy is the battery's ability to deliver power in watts per kilogram (W/kg).

In general, this framework is used to compare easily the different batteries suitable for use in different EVs such as BEVs and HEVs [89]. Therefore, from the various battery technologies presented on Figure 4.2, the Li-ion battery has higher capacity to deliver energy for driving. However, the lead-acid battery is the lowest in delivering the required amount of energy for driving.

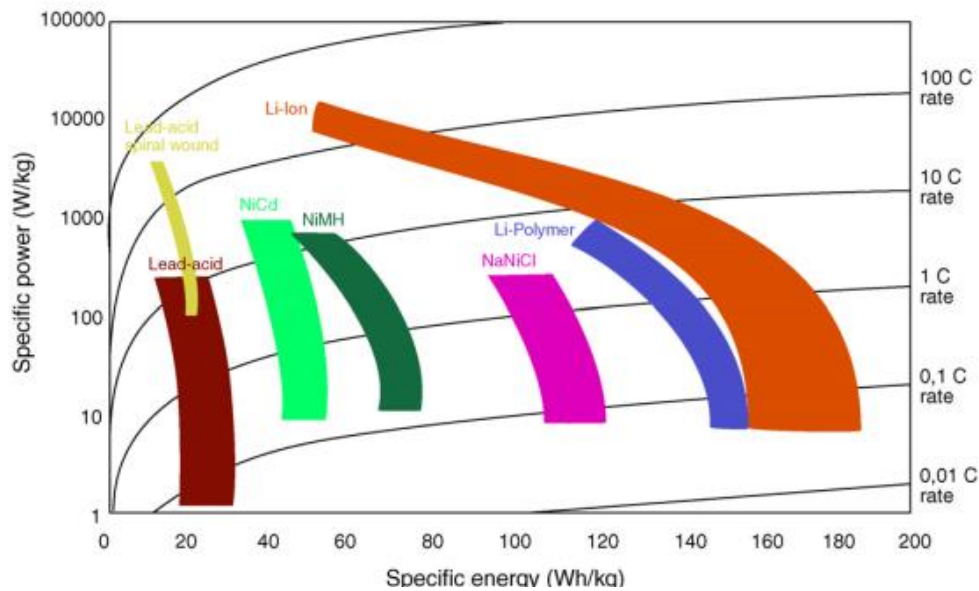


Figure 4.2 - Ragone Plot [91].

Earlier, the cost of battery is almost half the price of EVs that makes their operation in high cost [90]. Thus, extensive research is done to improve the efficiency and reduce its price. Therefore, the growth in deployment of EVs is accompanied by an increase in the battery production capacity. The growth in battery capacity is likely to have great implication on the reduction of battery price. Accordingly, the cost of EVs battery is dropped by 73% from 2010 to 2016 as shown in the Figure 4.3. It will also be expected to decrease up to 66% by 2030 and accordingly the EVs production will increase with lower production cost [91].

Although, the price of EV batteries is significantly decreasing, the lifetime of the battery is depending mostly on the usage profile. Furthermore, different battery models are used to measure and analyze the available EVs battery energy. Thus, in the following we want to provide an overview on how SOC and SOH of EVs battery are measured.

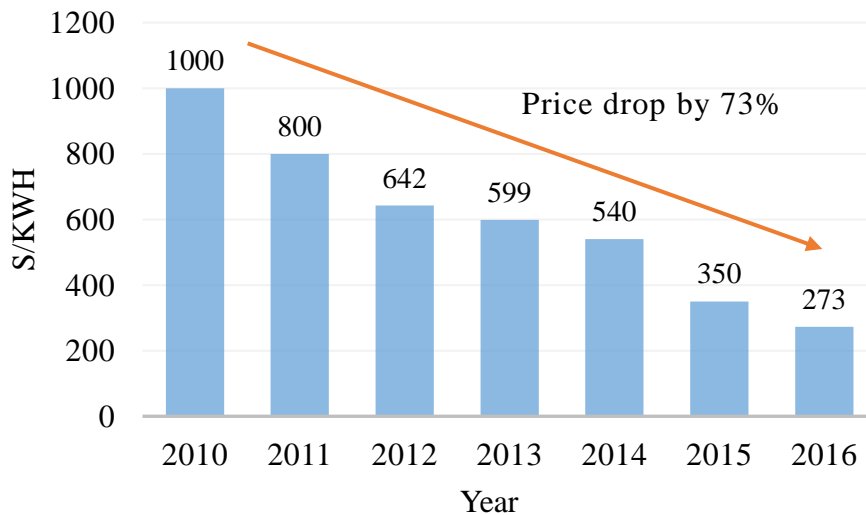


Figure 4.3 - Price Trend of a Lithium-ion Battery from 2010 to 2016 [92].

4.3.3 BATTERY MODELING METHODS

Over the years, various battery models are studied in the application of EVs and researchers have developed different battery models of different levels of accuracy and complexity. These models can be categorized into different groups. Thus, according to [92], they can be classified as electrochemical, mathematical, electrical, and polynomial.

The first battery model is an electrochemical model. This model is usually used to optimize the physical design aspects of the batteries, characterize the fundamental mechanisms of power generation, and relate battery design parameters with macroscopic (e.g., battery voltage and current) and microscopic (e.g., concentration distribution) information [93]. The main advantage of using an electrochemical battery model is that a highly accurate description of electrochemical processes within the battery can be obtained. However, many parameters related to electrochemistry are difficult to achieve in real-time applications. This model also requires days of simulation time, complex numerical algorithms, and battery-specific information that is difficult to obtain, because of the proprietary nature of the technology [94]. Furthermore, to solve complex numerical techniques are required, leading to high computational expense for on-board systems [95]. Hence, the electrochemical models are not suitable for vehicular applications, since the control system needs real-time data.

The second category of battery model is the mathematical model. This model uses empirical equations or models like stochastic approaches to analyze and predict system-level behavior for instance battery runtime, efficiency or capacity [93]. There are different types of Blackbox models that have also been proposed, such as neural network model [96], wavelet model [97]

and support vector machine model [84], in which no physical knowledge about the battery is required. However, mathematical models cannot offer an accurate information required to perform circuit simulation and optimization. In addition, most mathematical models only work for specific applications and provide in accurate results with error [92].

The third category of battery model is the electrical model. An electrical models uses a combination of voltage sources, resistors, and capacitors for co-design and co-simulation with other electrical circuits and systems [92]. Thus, electrical model accuracy lies between electrochemical and mathematical models. In comparison with other models, electrical models are more realistic, intuitive, and easy to handle. Moreover, it can be used for many electrical battery ranging from lead-acid to polymer Li-ion batteries [92].

Finally, the fourth category is polynomial model. This model represents the battery in form of simple expression containing state of charge (SOC), within the temperature for a certain range. Battery parameters such as the open-circuit voltage and all the internal impedances are a function of a fixed quantity (mostly state of charge). However, these models have limitations to provide information that are important in circuit simulation and optimization [93]. EVs can be configured to use one of the methods to know the status of SOC and SOH level. After that they will be used for computation or rejected, based on their level of SOC and SOH. Therefore, as we present above accurate real time SOC and SOH estimation improves the whole EVs energy management system. Thus, in this way we want to give an overview on how the EVs battery efficiency is measured with a focus on SOC and SOH are measured.

In the next section, we study and present the details about the description of the proposed solution on efficient EVs battery management.

4.4 DESCRIPTION OF THE PROPOSED SOLUTION ON EVs BATTERY MANAGEMENT

In this section, an overview about EVs battery detail and the proposed solution on the efficient battery management are discussed.

4.4.1 EV BATTERY DETAILS

As we have presented above, most of the EVs in the current market uses Li-ion batteries and it is forecasted to increase in the coming years. Thus, composition of the battery might slightly

different in different EV manufacturers, but Lithium ion batteries are composed of cells, modules, and packs. Battery cell is the basic unit of Li-ion battery that exerts energy by charging and discharging. Cells are designed to store high capacity per unit of volume in order to provide maximum performance in a restricted area of EVs and also, they are designed to have longer lifespan [95]. Besides, cells must endure shocks transmitted during the drive and possess high reliability & stability to the extent high and low temperatures. In order to provide this shield, cells are mounted on a module. Thus, the battery module is a battery pack that puts cells into frames by combining fixed number of cells for protecting them from shocks, heat and vibration while the vehicle drives [94].

Finally, the battery pack is the shape of a battery system installed to an EV. Moreover, this battery pack is holds of modules, and various control/protection systems including Battery Management Systems (BMS), a cooling system etc. [94]. Often, around 12 cells go into a conventional battery module for EVs and the capacity is within the 2~3kWh range are designed. For performing our simulation, we choose the two most popular EVs in the market, ZOE ZE40 and BMW i3 2018 electric vehicles.

For ZOE ZE40 EVs¹², as shown in Figure 4.4, it has $N=12$ battery modules in the battery pack. These modules are connected in series and each module has $M=16$ battery cells (C) that are connected in parallel. Each module has volt power of 3.7V (V) and the nominal capacity (N) of the battery is 63.35Ah. Thus, total capacity of each module gives the volt multiplied with the volt that is 44.4 volt (3.7volt * 12 module). Therefore, the total battery capacity of ZOE ZE40 battery is calculated as follows:

$$\text{Total_capacity} = C * N * V = 192 * 3.7 * 63.35 = 45,000 \text{wh} / 1000 \text{w} = 45 \text{kWh}$$

¹² <https://pushevs.com/2018/01/29/2018-nissan-leaf-battery-real-specs/>

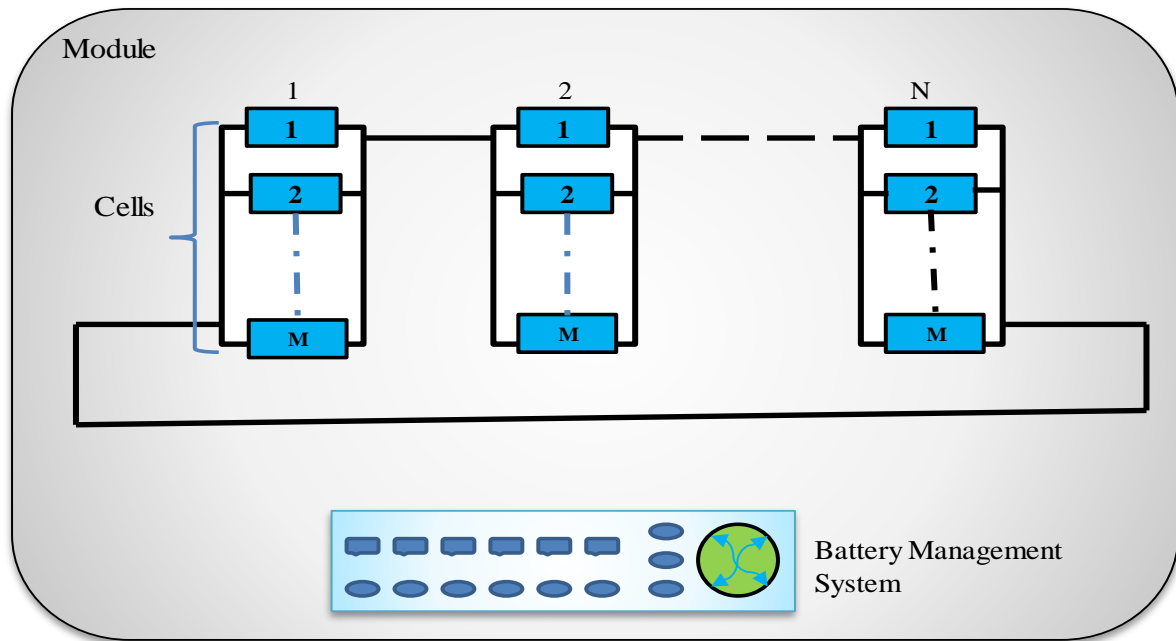


Figure 4.4 - EV Battery Pack Detail.

This gives the total capacity of ZOE XE40 EV, from which 41kWh is an available capacity for drive. Whereas, for BMW i3 2018 EV battery pack contains only $N=8$ series battery modules and on each module $M=12$ cells in total 96 cells (C) connected in parallel where each cell has 94Ah nominal capacity (N) and 3.68 V of nominal voltage (V) capacity. Hence, the total capacity of battery of BMW i3 2018 EV is computed as follows:

$$\text{Total_capacity} = C * N * V = 96 * 94\text{Ah} * 3.68\text{V} = 33.2\text{kWh}.$$

This gives the total capacity of battery size, from which 27.2kWh is usable energy available for drive. We modeled the battery of these to EVs using Matlab Simulink model. Thus, in the following section we present the MATLAB Simulink model of EV battery.

4.4.2 BATTERY SIMULINK MODEL

For the purpose of modeling, EV battery the MATLAB and its Simulink software tool are used. In the MATLAB graphical editor Simulink¹³, a generic model of Li-ion battery is developed according to shepherd's model. The Simulink toolbox is widely used simulation software in academia and industries for modeling and simulating dynamic systems. Simulink is a block diagram-based environment for simulating electronics and model-based design. It supports simulation, automatic code generation, continuous test, and verification of embedded systems.

¹³ <https://www.mathworks.com/help/physmod/sps/powersys/ref/battery.html>

The toolbox helps to incorporate algorithms in the model and exports simulation results into MATLAB workspace for further analysis.

The input parameters in the MATLAB Simulink model EV battery is the arrival of tasks with Poisson arrival rate and energy amount as a cycle. Thus, three different application demands are considered with different amount of energy demand. As presented in table 4.2, these

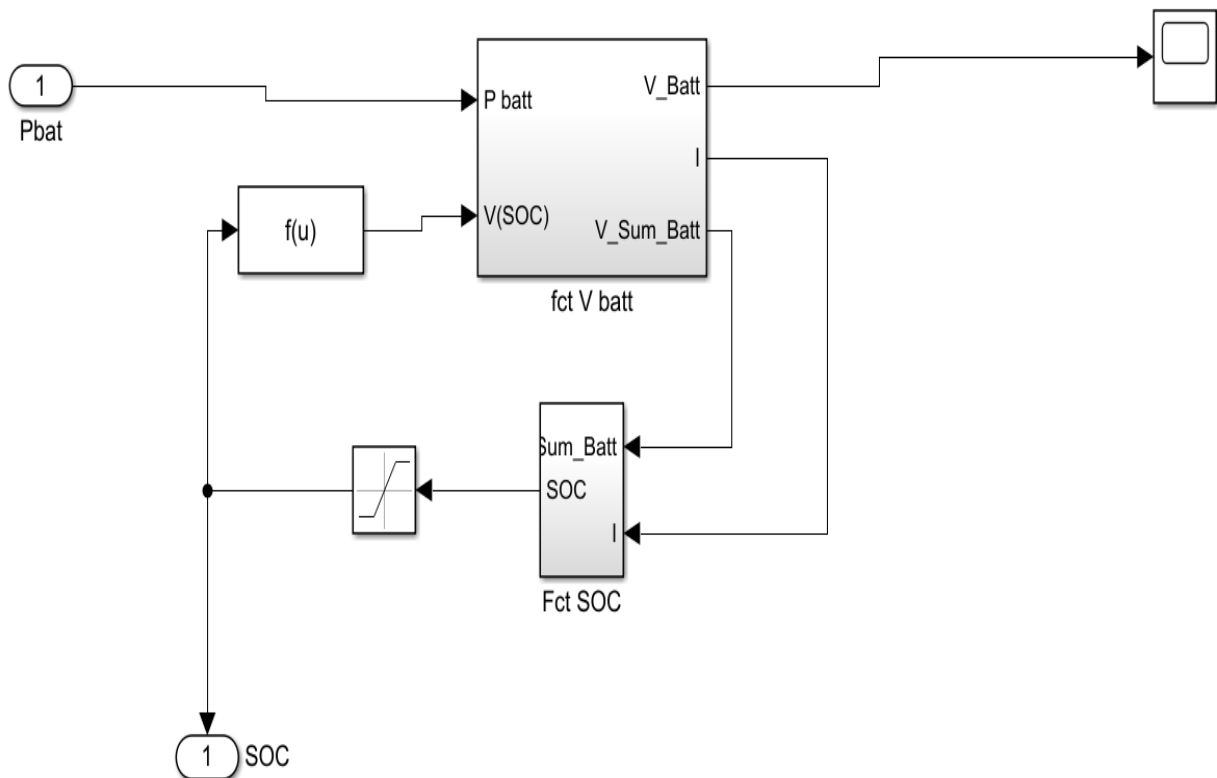


Figure 4.5 - Lithium ion Battery Simulink Model

applications have low, medium, and high energy consumption demand. In addition, these application demands are generated using Poisson distribution and created as a cycle in MATLAB workspace and transferred to Simulink model as it is shown in Figure 4.7. Finally, Matlab Simulink provides the SOC level after the computation tasks are performed. Since we consider the parked EVs for computation, the energy for acceleration, break or for other related not modeled as shown in the Figure 4.5.

4.4.3 PROPOSED SOLUTION

In this section, we present the proposed approach to choose EVs with healthier battery status and higher energy. As we presented in the previous section, EVs are designated for computation by following the steps shown in Figure 4.6. On one hand, the system accepts user's computation demand as a cycle. On the other hand, when EVs enters into the parking lot

they start to communicate about their battery SOC and SOH with the nearby fog controller. Then, as it is presented above, in the proposed algorithm, EVs are used to server users demand only when their battery SOH threshold is above δ . If there is no EV with battery SOH above the threshold value, the users demand is redirected to the cloud. Although, those EVs with SOH above threshold values will be selected, their SOC should be examined in order to use them for computation. Thus, we consider SOH value of EVs battery as healthy and in a good condition for performing additional tasks when the SOH value is above a certain threshold δ .

In addition, the parked EVs computing resource will be aggregated for creating stable environment and simplifying the VFC implementation. In addition, when the users demand arrives, the system ensures that there is sufficient resource or not.

Moreover, we assume that EVs arrive in the parking lot with full battery SOC level. We set threshold value for SOC as ϕ of EVs battery capacity. When the aggregated resource reaches the threshold value, the user's demand is redirected to the cloud server. As it is stated in the previous chapters, local users demand is severed only when there are enough resources in parking lot, otherwise it will be redirected to the cloud. In the next subsection, we present our evaluation results and discussions.

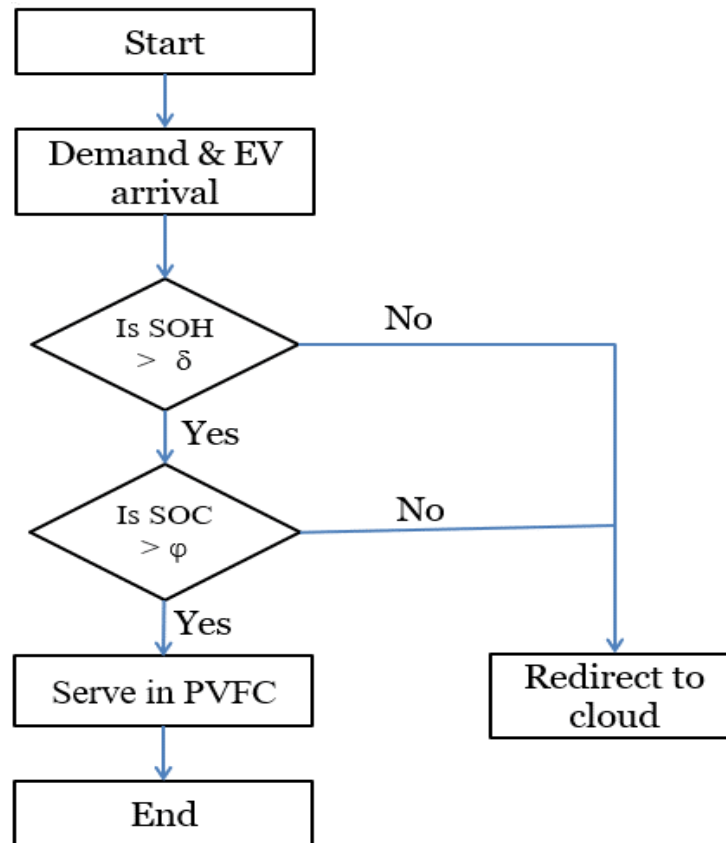


Figure 4.6 - Flow Chart to Serve Users Demand in EVs.

4.4.4 PERFORMANCE EVALUATION

In this section, we present and analyze the performance of different EV models to serve users' computation demand by changing the number of EVs in the parking lot. As we presented in the previous subsections, the EVs arrive to the parking lot with different SOH level status. We show the capacity of EVs to serve users' demand. In the following, we present the simulation assumptions followed by discussions of the main results.

a) SIMULATION ASSUMPTIONS

In order to evaluate the performance of the system, we used Matlab¹⁴ and its Simulink tool. In addition, we use the two popular electric vehicle models: EVs ZOE Z.E 40 2018 and BMW i3 2018 batteries for serving users demand. These two EVs have different battery energy storage capacities. In addition, we considered three types of application demands with different types of energy consumption in watt. Any of the users' demand is arrive into the system by following

¹⁴ <https://www.mathworks.com/products/matlab.html>

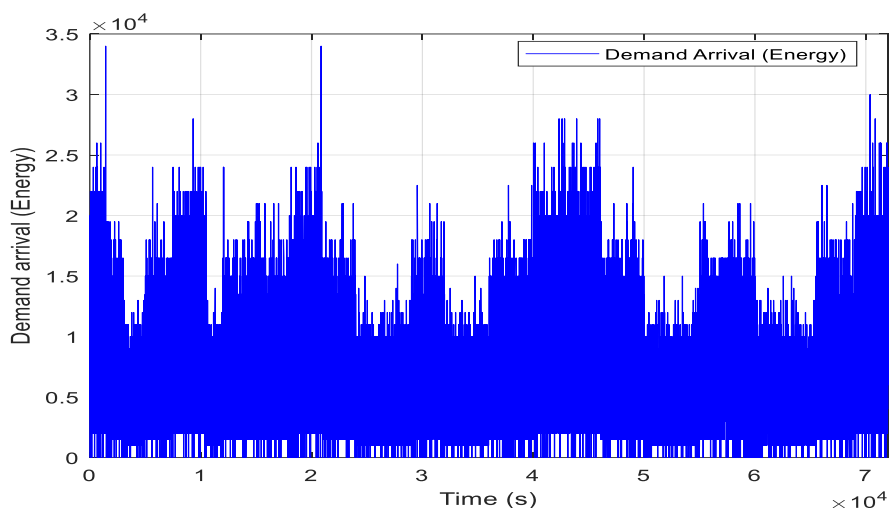


Figure 4.7 - Changes in arrival rate of users demands over time.

a Poisson process, total simulation time is 7200 seconds, as shown in Figure 4.7. Our performance considers the following procedures:

- We vary the number of EVs in the parking lot and we observe the impact on the availability of resources to serve the users demand.
- We consider different EV models with different battery capacities to observe the energy consumption evolution during the simulation time.

The set of simulation scenario and the main simulation parameters used in the simulation are presented in Table 4.2.

Simulation Scenarios	
ZOE ZE 2018 Model Battery Capacity (kWh)	41
BMW i3 2018 Model Battery Capacity (kWh)	33.2
EVs #	[1, 2, 3,4,5]
Simulation Parameters	
SOH min Values (%) - δ	90
SOC min Values (%) - φ	40
Simulation Time	7200 seconds
Application energy (unit of power)	{1000, 1500, 2000}

Table 4.2 - Main Simulation Parameters.

b) SIMULATION RESULTS AND ANALYSIS

As we have presented on the previous sections, all parked EVs in the parking lot are not used for serving users demand. Hence, we have to select EVs in consideration with their SOH and SOC level. Thus, at first, we show how EVs are selected according to their EVs state of health status and charge level. As it is shown in Figure 4.8, from EVs that arrive to the parking lot (see Figure 4.8a), those EVs with higher SOH status are selected (see Figure 4.8b). As we presented on the previous sections, EVs with higher SOH status have lower charging and discharging cycle. Hence, we choose EVs batteries used for computation where their SOH condition is above δ and the SOC level of EVs is above ϕ . For instance, every EV that enters to the parking lot with a higher SOC cannot be used for computation unless their SOH is above the threshold. In addition, EVs with higher SOH cannot be used for computation unless the SOC are above the threshold. Thus, Figure 4.8 shows the number of parked EVs and the selected vehicles to be used for computation. Figure 4.8(a) shows an example where there is 10 random parked EVs with different SOH status. Thus, based on our SOH value ($\delta=90\%$) 5 EVs are selected for serving users demand, see Figure 4.8(b).

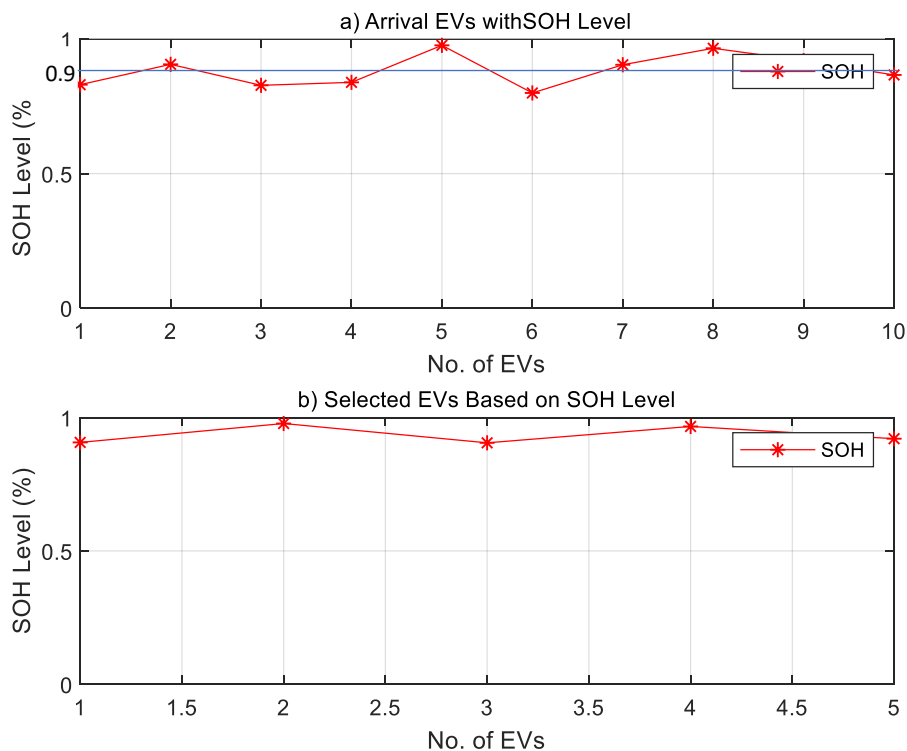


Figure 4.8 - (a) EVs arrival and (b) Selected EVs with higher SOH Level.

As mentioned before, we performed the simulations using two EV models with different battery capacities and we use the arrival of users' demand depicted in Figure 4.7 for the selected

EV models (BMW i3 2018 and ZOE ZE40 2018) that have different battery storage capacity. Figure 4.9 shows the SOC level of the selected EV models for serving users demand. Figure 4.9 (a), shows the battery consumption evolution of ZOE ZE 40 model EVs. Since the battery capacity is large, the SOC lasts longer compared to BMW i3 2018 EVs model.

Figure 4.9(b) shows the evolution of SOC consumption of BMW i3 2018 EVs model in serving users computation demand. We can observe that when the number of parked EVs is increasing, the overall SOC consumption level lasts longer.

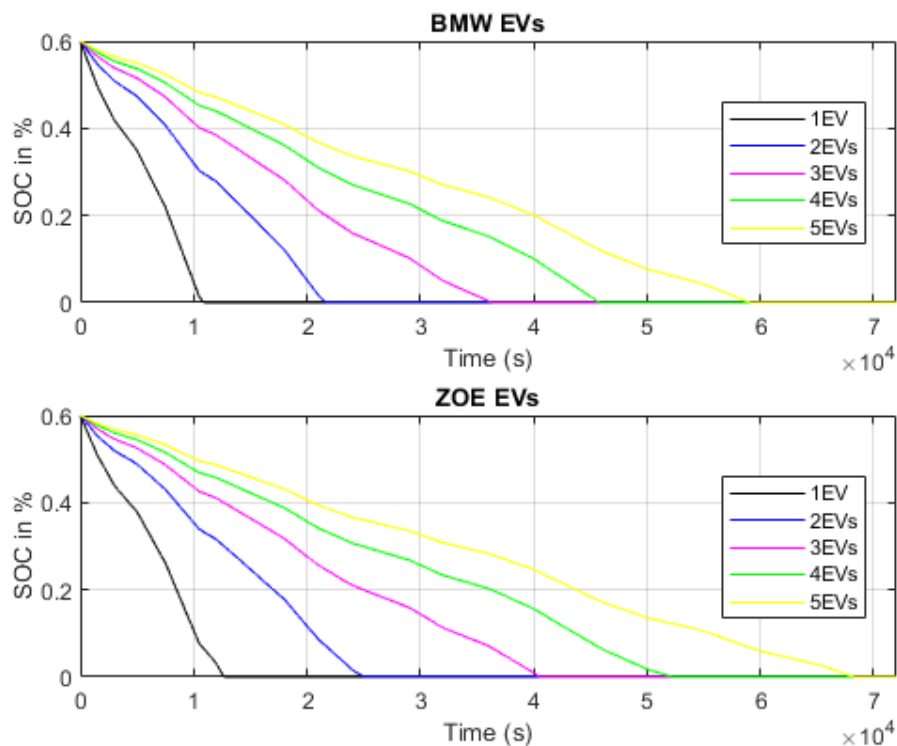


Figure 4.9 - SOC evolution of consumption over time for
(a) ZOE and (b) BMW models.

Moreover, from Figure 4.9 we notice that ZOE ZE 40 EV model (Figure 4.9(a)) clearly outperforms in terms of serving local mobile users demand. This is because of the larger battery size of ZOE ZE 40 EV model compared BMW i3 2018 EVs.

As we said earlier in this chapter and also in the chapter 2 of this manuscript, EVs are becoming more popular especially in the city centers. Therefore, we notice that when the number of EVs is increasing in the parking lot, the amount of aggregated resources becomes high. Consequently, SOC consumption lasts longer which implies more available resources to serve

users' demand. Thus, resources in the parking area depend on the number of EVs with higher SOH and SOC levels. Otherwise, if all EVs in the parking lot didn't meet SOH and SOC requirements, no EV is selected that indicates there is no resource in EVs. Therefore, all users demand will be redirected into the cloud.

In the following, we examine the capacity of parked EVs resource to serve users demand in EVs. Here, the capacity of EVs to serve the users demand depends on the availability of resource in each EV. Therefore, the resources of every EV are aggregated that gives the amount of available resources in the parking lot. As we said earlier, when a greater number of EVs with higher SOH and SOC battery status enter into the parking lot, a greater amount of resources is available to serve users' demands. In this regard, Figure 4.10 shows the amount of available resources in accordance with the number of parked EVs. So, when the number of EVs with higher SOC and SOH battery status increases in the parking lot, additional resources will be available to serve users' demands. However, when we compare the two EV models, ZOE ZE 40 EV and BMW i3 2018, the former has higher amount of resources.

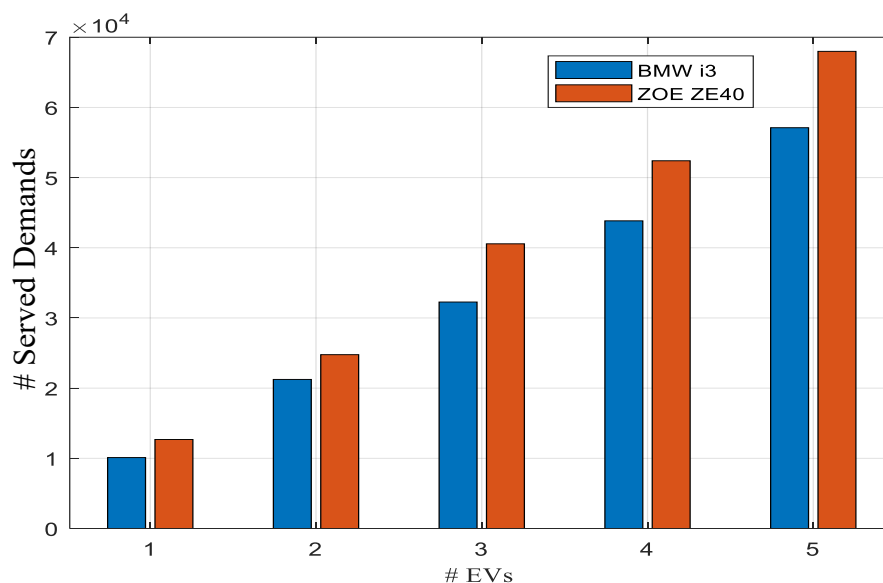


Figure 4.10 - Served Users Demands in the two EV models.

4.5 CONCLUSION

In order to solve the problem of constrained mobile devices, service providers design different solutions. One of the solutions is providing service near to the origin of the source. Nowadays in addition to high tech equipment's installed in EVs, the battery capacity and the battery technology is highly improved. Moreover, with the technology advancement, electric vehicles

are becoming intelligent and powerful to provide different services, for instance computation, storage, and communication. Therefore, one of the mechanisms proposed in this chapter is to evaluate the available resources of parked EVs for computing users demand. In this regard, the main contribution of this chapter is to efficiently manage EVs battery for serving users' demands in consideration with SOC and SOH status. We discussed about the available amount of resources in parked EVs by considering their battery SOC and SOH status. We consider SOC and SOH parameters since they describe the status of EV battery energy storage capacity. We consider two EV models, namely ZOE ZE40 2018 and BMW i3 2018 models, for comparing the availability of resources. In order to solve the problem of managing EVs battery efficiently based on the SOH and SOC values and perform the simulation, we use MATLAB's and its Simulink tool to model EVs battery. We proved to use EVs energy for computation by taking in consideration with SOC and SOH of the battery. Therefore, our solution proposes to use EVs battery with a determined thresholds level of SOC and SOH.

In the next chapter, we present further solution based on Markov Decision Process (MDP) approach where parked EVs resources are dynamic overtime and we aim to safeguard optimally the use of their resources.

MDP-BASED RESOURCE ALLOCATION TOWARDS A VFC

5.1 INTRODUCTION

As thoroughly discussed in the previous chapters, EVs besides their networking and communication interfaces, most recent vehicles are equipped with high computation and battery storage capabilities. Moreover, the incentive and subsidiary mechanisms are also designed by the governments to support the automotive industry and EV users. Since, EVs represent a solution for global warming and fossil energy crises, they are receiving an increased attention in recent years from government, international organization, academia, and research organizations. In recent years, EVs are coming with intelligent capabilities in regard to higher energy storage, computation, and communication capabilities. However, according to a survey [1], more than 95% of the time these vehicles are remained parked. Consequently, by designing an incentive mechanism for the owners of the parked vehicles, these abundant resources are used to provide services for nearby users.

Furthermore, portable devices, such as smart phones, tablet, notebooks are widely used for our day-to-day activities as a convenient tool for communication, entertainment, social networking etc. However, due to the size of the devices, mobile devices have usually constrained in terms of computation power, battery lifetime, storage space and even available bandwidth. Therefore, to solve these limitations of mobile devices, service providers are usually willing to make use of powerful servers offered through the cloud. Thus, constrained mobile devices offload their computation intensive task to the powerful server to perform the execution on their behalf. However, as we discussed in chapter 3, recently the trend is to push the storage and processing capabilities to the edge of the access network closer to their end users.

As discussed earlier in this manuscript, some recent bodies of research focus on introducing vehicles to the fog-computing paradigm for communication and service delivery. On the one

hand, there are unused resources on the parked EVs that can be used for serving local on-demand applications. Whereas, on the other hand, resources constrained mobile devices pose a significant challenge on executing resource intensive mobile users' applications. Therefore, in this chapter of the manuscript, we propose to integrate EVs resources available locally within vehicles as part of the fog computing infrastructure to serve mobile application demands. The resulting framework would constitute the new paradigm called Vehicular Fog Computing (VFC). Therefore, instead of sending users' computation requests to remote cloud servers, VFC handles the nearby users' demands using locally available vehicular resources. Furthermore, VFC distinguishes itself from other existing techniques with its proximity to end-users, dense geographical distribution and mobility support [63].

Also, parked EVs are considered as fog nodes to serve users demand. However, using EVs battery may reduce the performance and battery life. Hence, in the previous chapter, we introduce an efficient battery management system by considering the SOC and SOH of EVs battery. However, this time we want to tackle a decision to *Accept* or *Reject* users demand in fog nodes.

To this end, in this chapter, we propose an innovative system of VFC that employs EVs as a fog node for computing local device application demands. To the best of our knowledge, this work is the first to consider EVs energy to serve local mobile demands. Particularly, we consider the parked electric vehicles as fog nodes that would offer computing services to serve nearby users. Here, we address the problem of resource allocation for mobile applications with highly intensive computation tasks. The abundant unused resources of parked EVs are aggregated in order to serve the computation and storage demands of these mobile devices particularly constrained in terms of energy and processing capabilities. A Markov Decision Process is adopted to model this resource allocation problem and solve the underlying optimization problem using a dynamic programming.

The rest of this chapter is organized as follows. We first provide the details of the proposed system architecture in section 5.2. This is followed by the MDP based formulation of the underlying resource allocation problem, thorough a presentation of the corresponding solution and the proposed decision-making algorithm in section 5.3. Next, the performance evaluation

of the proposed resource allocation scheme is illustrated in section 5.4. Finally, conclusions and general discussions for possible future research directions are given in Section 5.5.

5.2 SYSTEM ARCHITECTURE

We consider for this study the synoptic architecture for a VFC as shown in Figure 5.1. Beside the central cloud, the proposed fog computing scheme comprises three different components: the fog controller, a set of parked EVs considered as fog nodes and finally mobile devices. Mobile devices have become very popular and run different kind of energy intensive, heterogeneous, and dynamic applications. On the one hand, due to the devices size, their computation, storage and energy capabilities are limited [98]. On the other hand, more powerful and resource hungry applications are emerging and attract great attention. Therefore, in order to get good application execution and to save energy for mobile devices, computation offloading is a promising option.

We consider that parked electric vehicles at parking lot are considered as fog servers. Thus, EVs are used as a computing infrastructure by aggregating the abundant resources of individual vehicle resources. Therefore, VFC offers this resource pool that contains a certain level of energy/computation in order to compute highly intensive computation tasks.

In the proposed system, each mobile application demand and electric vehicle arrival follows a Poisson process. In addition, electric vehicles' departure rate follows an exponential distribution. Thus, in this work, we propose to use part of the unused resources of parked EVs by nearby mobile devices for computation while making sure to not exhaust all the EVs energy.

The central cloud is a widely used approach of providing computing resources and services for resource constrained devices. It allows an on-demand infrastructure service from anywhere to extend the capabilities of mobile devices through offloading technics. Typically, data generated by mobile devices are transferred to geographically distant clouds to be processed and stored [21]. However, this computing model increases latency when billions of devices are connected to the Internet [99]. Thus, in this work, we propose to use cloud computing only when there is no enough resource in the parking lot. Nowadays, an alternative model that can solve this limitation is bringing the computation resources of cloud server near to the origin called fog computing.

The fog controller provides wireless access (Cellular or Wi-Fi) for mobile devices. It also affects computation demands to the nearby parked EVs (within its coverage range). Upon arrival of energy intensive mobile application computation requests, the controller decides whether to run it at the VFC or redirect the requests to the cloud according to the availability of energy within parked EVs. Cloud computing is used whenever there are no enough resources available to serve users' application demands using the parked EVs.

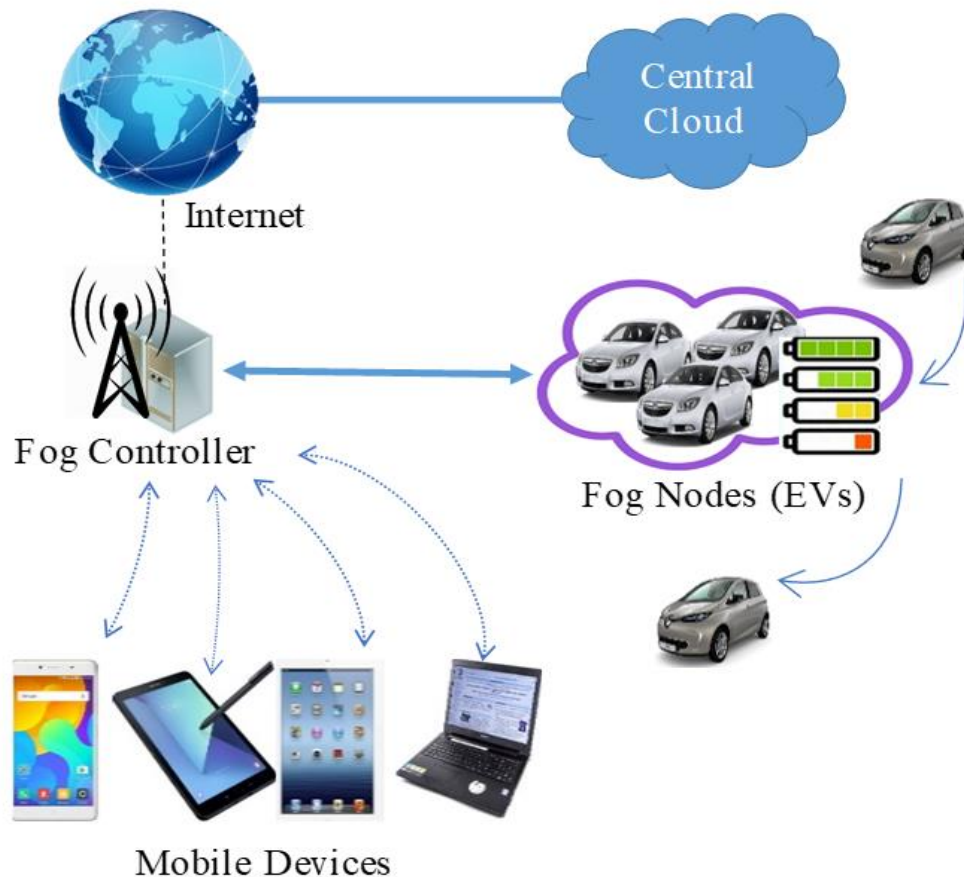


Figure 5.1 - Vehicular Fog Computing (VFC) Architecture.

In this chapter, we assume that there is no charging infrastructure in the parking lot that supplies electric energy for recharging EVs. Moreover, available energy in the parking lot depends mainly on the parking lot occupancy level and state of charge of the parked EVs. If higher energy level becomes available, additional computation requests from mobile device would be served. Therefore, the main objective in this chapter is to optimize the state of charge for parked EVs in order to better serve the demands of mobile applications. The underlying optimization

problem is formulated through a Markov Decision Process, where the details are provided in the following section.

5.3 MDP BASED RESOURCE ALLOCATION

Markov decision process (MDP) constitutes a mathematical framework that aims to model decision making problems, where a decision maker (in our case the fog controller) oversees the choice of the decision that would be taken in order to optimize the performance of its system. They are widely used in several disciplines, such as: automated control, robotics, intelligent vehicular and manufacturing. The goal of MDP in resource allocation problem is to find an optimal allocation of a fixed amount of resources using optimization techniques. In this work, the aim is to optimally use the available computation resources of parked EVs to serve local mobile applications demands. To resolve this issue, we formulate the problem as a Markov Decision Process. In the following, we recall the MDP concept before presenting the problem formulation and solution.

5.3.1 MARKOV DECISION PROCESSES CONCEPT

Markov decision process (MDP) constitutes a mathematical framework for dynamic controlling systems that evolves stochastically [100]. MDPs are used for modeling decision making. The systems resolved by MDP are observed at times $t=1, 2, \dots, n$ where n is called the time horizon. The decision maker takes a decision regarding the action to be done at each time. The target of the decision maker is to choose the actions that optimize the performance of the system. This section introduces our notation and assumptions and serves as a brief overview of the basic MDP results.

An MDP consists of a set of 5 elements $\langle S, A, P, R, \gamma \rangle$ as described below:

- S : is a finite set of the system states,
- A : is a finite set of actions to be chosen by the decision maker,
- $P: S \times A \times S \rightarrow [0, 1]$ defines the transition probability. The probability that the agent goes to state $s \in S$ upon execution of action $a \in A$ in state $s \in S$ is $P(s|S, a)$. We assume that, for any action, the corresponding transition matrix is stochastic $\sum_{s'} P(s'|S, a) = 1 \quad \forall s \in S, a \in A$.

- $R: S \times A \rightarrow \mathbb{R}$ defines reward function. The agent obtains a reward of $R(s, a)$ if it executes action $a \in A$ in state $s \in S$. We assume the rewards are bounded.
- γ : where $\gamma \in [0, 1)$ is the discount factor (a unit of reward as time $t + 1$ is worth the same to the decision maker as a reward of γ at time t), γ_t is the reward that the agent receives at time t , whose distribution depends on the policy π and initial distribution

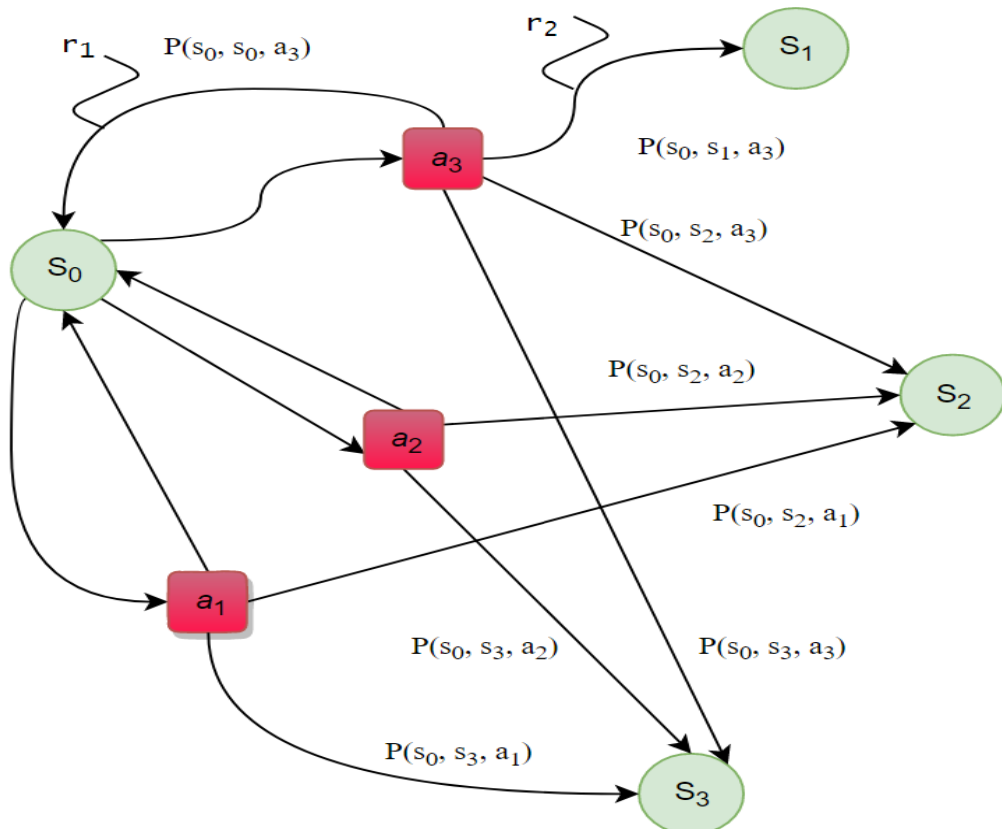


Figure 5.2 - Example of MDP System with Four States and Three Actions.

over the state space $s : S \rightarrow [0, 1]$. This definition is meant to capture the notion of present value in accounting: a dollar now is worth more than a dollar one-time unit later. The smaller γ is, the more we weight the present. For example, bank pay interest so that \$95 at year t can become \$100 at year $t + 1$. This suggests that having \$100 at year $t + 1$ is the same as having \$95 at year t ; i.e., we can assign $\gamma=0.95$ to capture this effect.

Figure 5.2 shows an example of MDP system with four states and three possible actions. At each iteration time $t_i \in \{t_1, t_2, t_3, t_4, \dots, t_n\}$, the system environment in the state $s_i \in \{s_1, s_2, s_3, s_4\}$, MDP offers an action $a_i \in \{a_1, a_2, a_3, a_4\}$ chosen by the decision maker which optimizes the decision makers total reward.

The probability $P(s_i, s_{i+1}, a_i)$ is that the system moves for the next state s_i to the state $s_{i+1} \in t_i$ by taking an action a_i where the chosen action maximizes the total reward. Moreover, next state s_{i+1} depends on the current state s_i and the selected decision makers action a_i . Therefore, MDP satisfies markov property, given an action a_i and the current state s_i , the next move is conditionally independent of all previous states and actions.

An MDP consists of an agent and an environment that the agent interacts with. This interaction happens over a sequence of discrete time step t , as it is shown in Figure 5.3. Thus, at each time step t the fog controller perceives the state of the environment and action a_i is chosen by the fog controller to perform, which is the MDP decision maker in our system architecture, then the system moves to the state s_{i+1} according to the evolution of the system's environment. The goal of the fog controller is to maximize the rewards received from its interaction with the environment. The dynamics of the environment are stationary, and the state signal must contain all the relevant information.

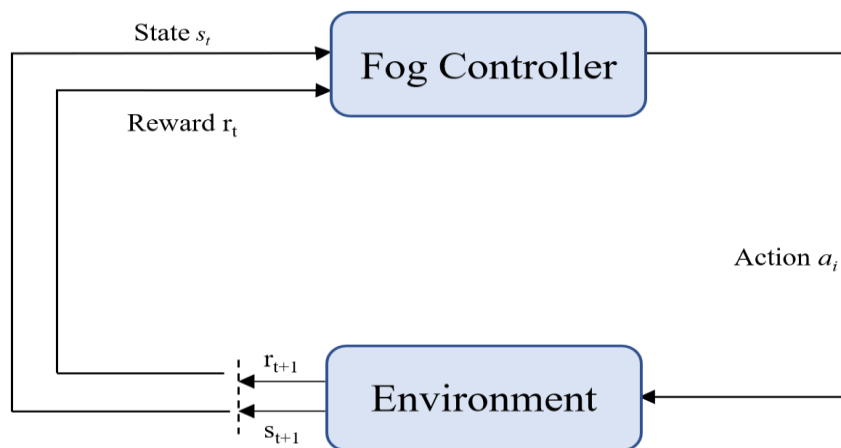


Figure 5.3 – Evolution of Markov Decision Process System.

Thus, in MDP, the goal is to find a policy that maximizes the total expected discounted reward over an infinite time horizon. In MDP [101], at each step, the decision maker observes the current state of the system and chooses an action according to its policy. There are several commonly used ways to find the optimal policy, and the central of all of them is the concept of policy iteration. A policy iteration is a method of solving the problem through iterative method for the objective of finding optimal policy. In optimal policy, for every state, there is no action that gets a higher sum of discounted future reward than the chosen action. Therefore, the optimal policy is the policy that achieves the highest value for every state:

The objective function to be maximized is defined in equation (5.5):

$$\text{Maximize} \left(\sum_{t=0}^{\infty} \gamma^t R(s, s_{t+1}, \gamma(s_t)) \right) \quad (5.5)$$

Subject

to:

$$\sum_{t=0}^{\infty} R((s = s_n)_t, s'_{t+1}, a_i) = 0$$

$$\forall s \in \{s_1, s_2, s_3, \dots, s_n\}, \quad \forall a_i \in \{a_1, a_2, a_3, \dots, a_n\}$$

Where $s_1, s_2, s_3, \dots, s_n$ denotes the state of the system and $a_1, a_2, a_3, \dots, a_n$ is the action selected that maximize the collected rewards.

For the agent to be able to maximize the reward collected from its interaction with the environment, it must evaluate the value of each state and implement a mapping from state to action at each time step. This mapping is the agent's policy denoted by $\pi(S)$. Therefore, to find the policy (mapping from states to actions) that maximizes the sum discounted rewards, the algorithm uses the following two equations, (5.6) and (5.7), to calculate the optimal policy for all the states S until no further changes take place (according to the discount factor):

$$V(S) = \sum_{S'} P_{\pi(S)}(S, S') (R_{\pi(S)}(S, S') + \gamma V(S')) \quad (5.6)$$

$$\pi(S) = \arg \max \{ \sum_{S'} P_a(S, S') (R_a(S, S') + \gamma V(S')) \} \quad (5.7)$$

Where $P_{\pi(S)}(S, S')$ is optimal policy transition probability from state S to S' if action a is chosen and $R_{\pi(S)}(S, S')$ is a reward obtained by moving from S to S' according to the policy.

Several techniques are available in the literature to solve MDPs problems: Linear Programming (LP) and Dynamic Programming (DP) [100]. Linear programming is used when the state space is finite. Contrary to the linear programming, the Dynamic programming has the advantage to offer the possibility to solve complex problems with wide state and action [102]. In this work, we use dynamic programming where the model and simulation assumptions are presented in the following subsections.

5.3.2 PROBLEM FORMULATION AND SOLUTION

In this section, our goal is to construct resource allocation policy that decides whether the users' demand should be admitted or rejected in VFC based on the available resources in the parking lot. We want to find a policy that maximizes expected rewards.

As indicated in the previous subsections, we formulate the problem as an MDP with the following five elements tuples $\langle S, A, P, R, \gamma \rangle$:

- States (S): we represent the states as the amount of aggregated energy available in parked EVs. The decision is made on the basis of these states. Notice, the more energy available in parked EVs, higher number of users' demand is served, and higher rewards will be collected. However, the amount of energy depends on the number of parked EVs with the amount state of charge. Given this consideration, we take state spaces S to consist of vectors of length n . Thus,
 - ✓ $S = S_0, S_1, S_2, \dots, S_n$ refers to the different aggregated energy status of the parked EVs:
 - S_0 : Full Energy,
 - S_n : Empty Energy,
 - S_i : Energy level with load level = $i\%$, $i \in \{1, 2, 3, \dots, n - 1\}$.
- Actions (A): given our representation of states, we need two actions: *Accept* or *Reject*. These actions have clear effect on the next move of the system state. Thus, $A = a_0, a_1, a_2, \dots, a_n$; where a_i refers to the action to be selected according to the transition probabilities:
 - ✓ a_0 : represents the action taken when no resources are available for computation. Therefore, the users' demand will not be served at VFC and will be redirected to the cloud.
 - ✓ a_i : the use of surplus energy for computation ($k\%$ of energy level for computation), $k \in \{1, 2, 3, \dots, n - 1\}$. This refers the use of energy for both computation and EVs travel.

- Transition probability (P): $P(S, S', a_i)$ is the probability of moving from the energy level S at $t = n$ to the energy level S' at $t = n + 1$ by doing the action a_i . $P(S, S', a)$ is determined by the arrival and departure of EVs with a certain SoC (State of Charge). Hence, $P(S, S', a)$ is related to the energy consumption of users' computation demand and the arrival of EVs in the next times $t = n + 1, n + 2, \dots$, etc. The availability of energy for computation for next time $t = n + 1, t = n + 2, \dots$ is related to the number of served users' demands and the arrival of new EVs in the parking area, as shown in Figure 5.4. Therefore, when EVs, with certain state of charge above the threshold, enters in the parking lot at time t it adds energy to the aggregated energy level as a result state change (from s to s') may happen at time $t+1$. Similarly, when EVs depart from the parking, there might be a change in the state of the system, depending on the amount of energy EVs have. Whereas, the departure or arrival of an EVs with minimum energy level will not impact the system's state. However, if there is no new arrival or departure of EVs, the system keeps using the available energy for computing users' demands that results in a state change.



Figure 5.4 - EVs Arrival and Departure in Parking lot.

- Rewards, $R(s, a)$ is an immediate reward collected by the decision maker if it performs action a in state s . In our study, we represent R as a positive integer $R \in \{0, 1, 2, 3 \dots n\}$, where $R_i = 0$ means that the fog controller does not assign users computation demand to EVs.
- γ : is the discount factor. Discount reward is computed as the reward overtime by some factor γ , with $0 < \gamma < 1$. That is, if R_t is the reward obtained at time k , we take the total reward to be $\sum_{t=0}^{\infty} \gamma^t R_t$.

- The controller: represents the fog controller, presented in the previous sections, which is responsible for aggregating and managing the resources of nearby EVs. It also takes decisions regarding the allocation users' application demand based on the available energy.

The MDP model aims to find the best policy $\pi(s)$, that takes the form: $\pi : S \rightarrow A$, that the controller has to choose regarding the use of the surplus power in the parking lot. The goal is to maximize the reward by using the energy of EVs for serving local mobile user's computation demands without affecting the energy used by for EVs mobility. In order to guarantee this, we define the energy level s_n (no energy for computation) as the state that score lowest reward. Therefore, at each time, the decision maker tries to choose other action that optimizes the total reward of the system than an action a_0 which leads to the battery status s_n .

In the next subsection, the detailed description of the proposed algorithm is presented.

5.3.3 MDP-BASED DECISION-MAKING ALGORITHM

We showed that the problem is formulated as a standard MDP problem and can be solved with Dynamic programming approach. According to Bellman[102], with dynamic programming optimal policy is computed with two vectors: the expected value function $V(s)$ with real values whereas a policy $\pi(s)$ contains an action of states. Therefore, as it is presented in the previous subsection, the decision maker objective is to find the optimal policy that maximizes the overall profit of the system. Hence, the decision maker follows policy iteration algorithm shown in algorithm 5.1. The algorithm iteratively performs two steps: value determination (see line 6 Algorithm 5.1) which calculates the utility of each state given the current policy, and policy improvement which updates the current policy (see line 7 Algorithm 5.1) if any improvement

Algorithm 5.1: Schematic Description of The Policy Iteration

Algorithm

```

1  Function Policy_Iteration (S, A, P,  $\gamma$ ) returns a policy
2  Input: S, A, P,  $\gamma$ 
3  Output:  $\pi(s)$  – optimal policy
4  Repeat
5      For each stat i do
6           $V(S) = \sum_{S'} P_{\pi(S)}(S, S')(R_{\pi(S)}(S, S') + \gamma V(S'))$ 
7           $\pi(S) = \arg \max \{ \sum_{S'} P_a(S, S')(R_a(S, S') + \gamma V(S')) \}$ 
8      End for
9  Until change = false
10 Return R

```

is possible. The algorithm terminates when the policy stabilizes. The detailed activity of the fog controller /decision maker is explained in Figure 5.5.

The decision-making approach we proposed is described through sequence diagram as shown in Figure 5.5. In order to decide to accept or reject computation request of mobile users, the fog controller adopts first come first serve scheduling discipline. Up on arrival of EVs into the parking lot, they sent their energy status information to fog controller. In addition, fog controller receives users' requests from mobile devices. Once the energy status of EVs and mobile applications energy demands received, an action that scores maximum reward is selected. Then, the tasks will be admitted into fog nodes. However, if there is no energy to serve the users demand in the parking, the controller rejects the users request and their computation request will be redirected to the cloud. To evaluate the performance of the proposed algorithm, we did simulation with different scenarios that presented in the following section.

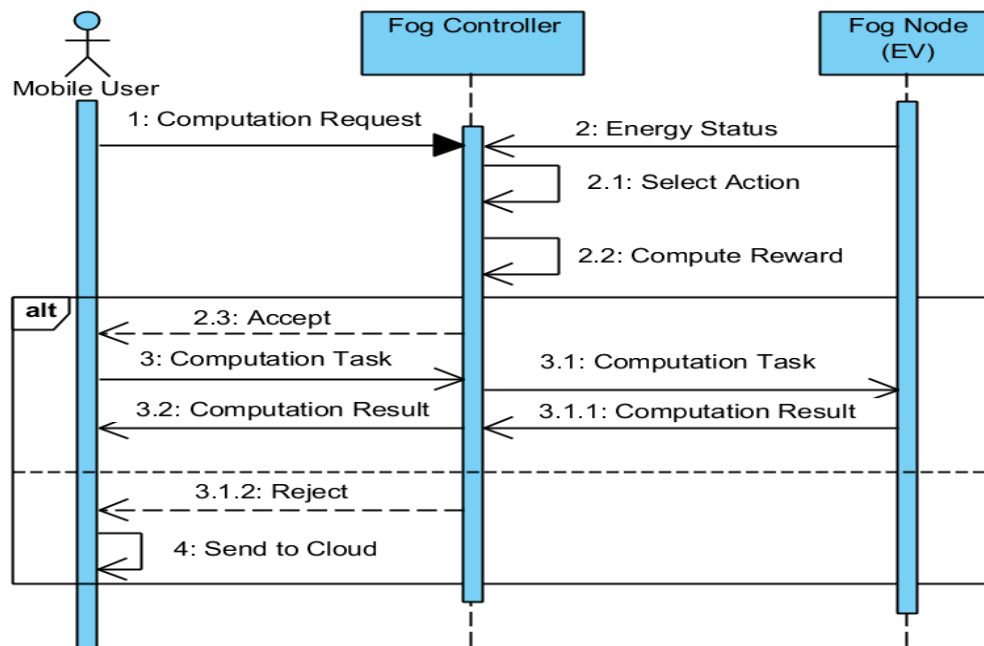


Figure 5.5 - Mobile Users' Resource Allocation Algorithm.

5.4 EXPERIMENTAL EVALUATION AND ANALYSIS

In this section, we simulate the proposed solution using Matlab MDP-toolbox. In the first subsection, we explain the simulation environment and metrics used for simulation. Then, we

devote the second subsection to highlight and discuss the different results on hand. Before detailing those subsections, let us present our detail MDP representation.

As we presented in section 5.3, the state space is modeled with the number of each parked EVs energy level. However, considering each parked EVs in the parking as a state will create larger and infinite state space. In order to simplify simulation process, we used an aggregated state space. The idea behind is that the resulting reduced state space forms small stationary finite state MDP, which can then be efficiently solved. Hence, we identified five possible different states to represent the occupancy rates, namely $S = \{\text{Full, High, Medium, Low, Empty}\}$ which represent the overall available energy in the parking: up to 100%, up to 75%, up to 50%, up to 25 and $\approx 0\%$ respectively. In addition, we also define four possible actions, namely $A = \{a_0, a_1, a_2, a_3\}$, which represent the action taken at each energy level. Thus, action a_0 is taken when there is 0% up to 25% energy is available in the parking. An action a_1 is selected when the available energy level is from 25% up to 50%, a_2 is selected when energy status is from 50% up to 75% and action a_3 is selected when the available energy is greater than 75%.

Furthermore, for each action a_i , we defined its corresponding transition probability matrix $P(S, S', a_i)$. The matrix is an $n - \text{by} - n$ matrix that comprises all the probabilities for moving from energy status S at time t to energy status S' at time $t + 1$ using action a_i . Finally, we use reward and global reward values for measuring the performance of the proposed model. The reward is collected at each decision slot whereas global reward is the sum for all rewards collected over simulation time. In order to study the performance of our approach, we calculate the reward of our MDP strategy (MDP_str) at each decision period throughout simulation time. Then, we compute the global reward for each case. Thus, higher global reward value shows the availability of higher energy gain in parked EVs. Whereas, lower global reward shows lower energy level in parked EVs.

5.4.1 SIMULATION SCENARIOS

The parked EVs computing resource will be aggregated for creating stable environment and simplifying the implementation of VFC. Thus, Figure 5.6 shows synoptic MDP graph with five states. It also shows the different transition probabilities between different states with action a_0 . In addition, the states refer to the battery status of our system, such that S_0 represents 100%

battery level and $S_1, S_2, S_3,$ and S_4 represents 75%, 50%, 25% and 0% battery level, respectively. Each state represents the aggregated energy level of parked EVs.

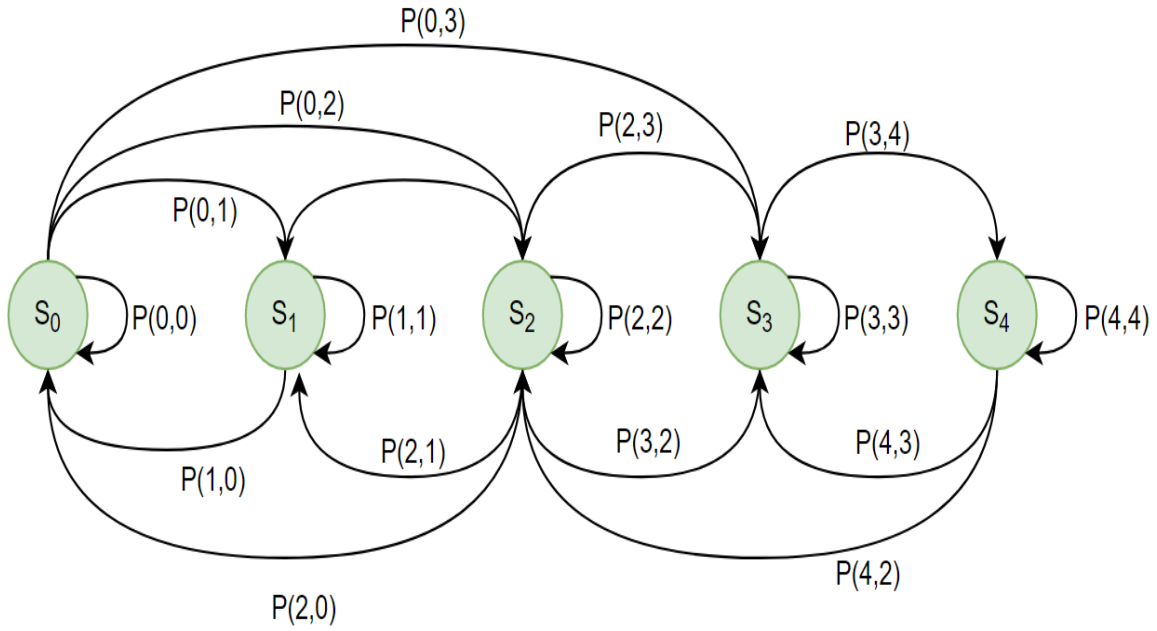


Figure 5.6 - Markov Chain Model with five states for EVs arrival.

In order to evaluate the performances of the proposed MDP model, we used Matlab and its MDP Toolbox [103] to implement our MDP_str scheme for resource allocation. Furthermore, in order to prove the effectiveness of our strategy, we compared the simulation results with *Fixed_str1*

Case 1	(C1: Poisson): $\lambda = 30\%$
Case 2	(C2: Poisson): $\lambda = 60\%$
Case 3	(C3: Poisson): $\lambda = 80\%$
Case 4	(C4: fixed): = 30%
Case 5	(C5: fixed): = 60%
Case 6	(C6: fixed): = 80%

Table 5.1 - Summary of Simulation Scenarios.

and *Fixed_str2* strategies. The later are defined as two different resource allocation algorithms that define fixed thresholds for the available energy in order to decide whether to serve users' demand or not. Specifically, *Fixed_str1* and *Fixed_str2* use respectively 30% and 60% of the available energy to process highly intensive computation tasks. However, no computation demands are served if the available energy level is less than the previously fixed thresholds.

As illustrated in Figure 5.7, we chose six different scenarios for the changes in occupancy level over a 24 hours period. Specifically, the first 3 scenarios, plotted in Figure 5.7 (a), consider 3 different Poisson distributions for occupancy level of EVs in a given parking lot. The corresponding values of parameter λ are: 30%; 60% and 80%. Whereas, in Figure 5.7 (b) plots the three other scenarios with fixed occupancy level over time, namely: 30%; 60% and 80%. Table 5.1 summarizes the used scenarios. Furthermore, the level of energy available in the parking lot is determined by the amount of energy offered by each EV. It is also influenced by the difference between arrival rate and departure rate of EVs over time. The following subsections present the result and discussions of the proposed solutions.

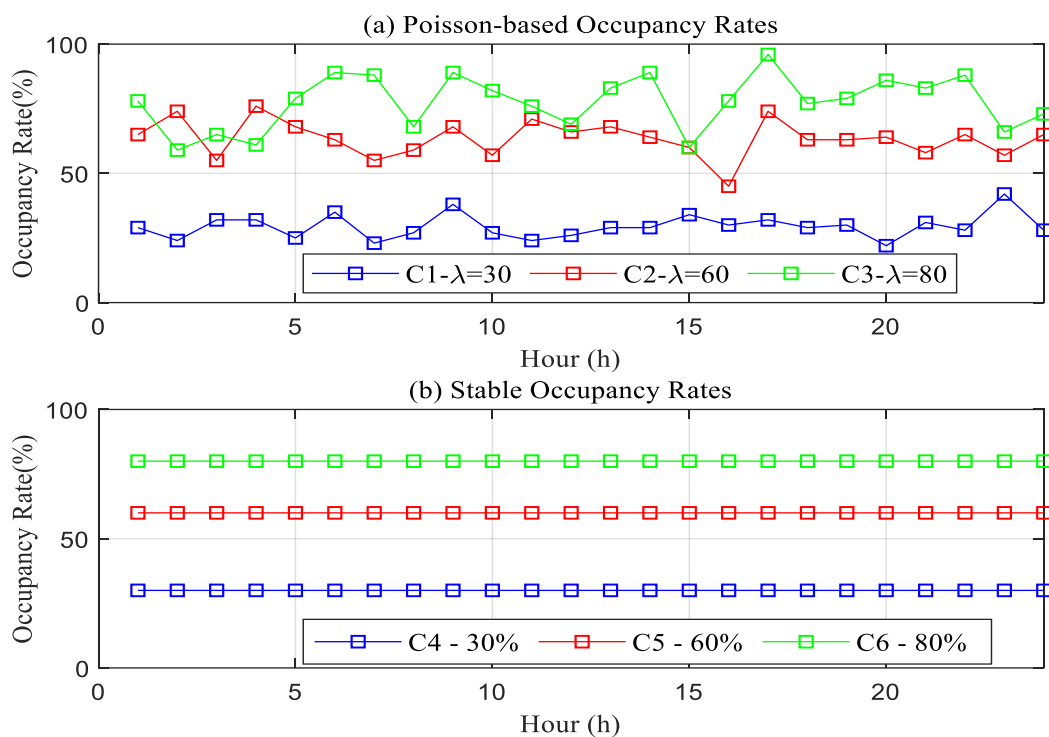
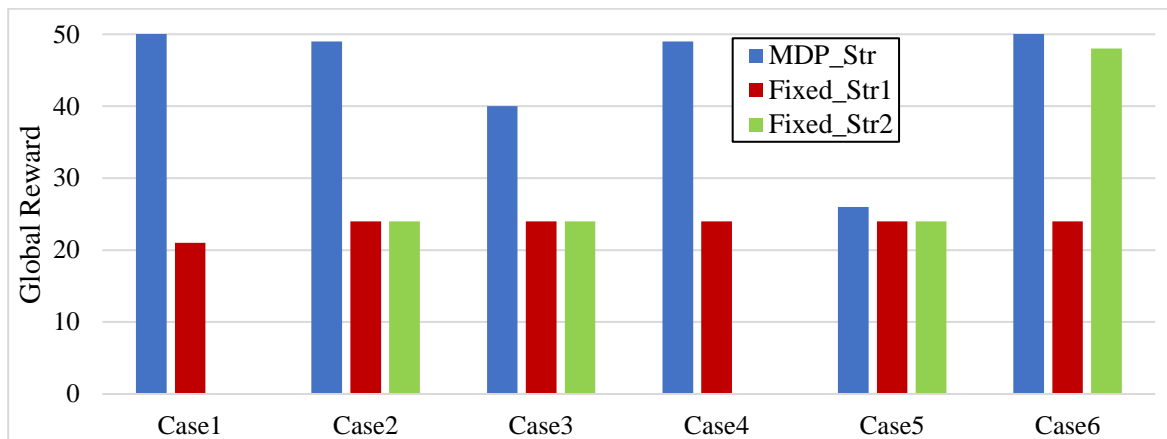


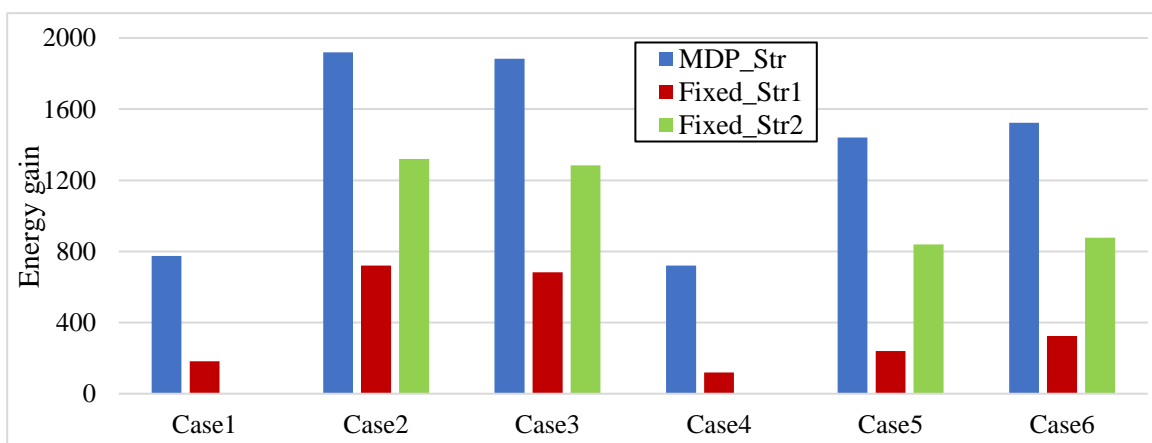
Figure 5.7 - Different simulation scenarios for the changes in occupancy rates over a 24 hours period.

5.4.2 RESULTS AND DISCUSSIONS

As mentioned before, the proposed system is evaluated with six different cases (shown in Figure 5.7). Figure 5.8 shows the simulation results achieved by each resource management strategy for the different scenarios. We can notice from Figure 5.8. (a) that our *MDP_str* always achieves higher scores in terms of global reward. Hence, *MDP_str* makes better use of the overall energy available in EVs in order to optimally serve a higher number of users' demands compared to the two other strategies. Specifically, the global reward is improved on average by 37.78% and 57.33% compared to *Fixed_str1* and *Fixed_str2*, respectively. Moreover, Figure 5.8. (b) shows the global energy gain obtained by the three resource allocation policies for the six different scenarios. Indeed, *MDP_str* scores higher energy gain compared to *Fixed_str1* and *Fixed_str2*. The *MDP_str* improves the energy gain by 65.40% and 54.41% on average compared to *Fixed_str1* and *Fixed_str2*, respectively.



(a) Global Rewards values for each Scenario



(b) Energy gain evaluation for each scenario

Figure 5.8 - Overall Performance Evaluation for the Different Simulation Scenarios.

Finally, in order to provide yet more extensive insights for the performance of the proposed scheme, we evaluated the different performance metrics over simulation time and specifically: (i) reward, (ii) global reward and (iii) energy level. Thus, in the following we perform additional simulations with different level of parking occupancy rate using the six cases shown in Figure 5.7.

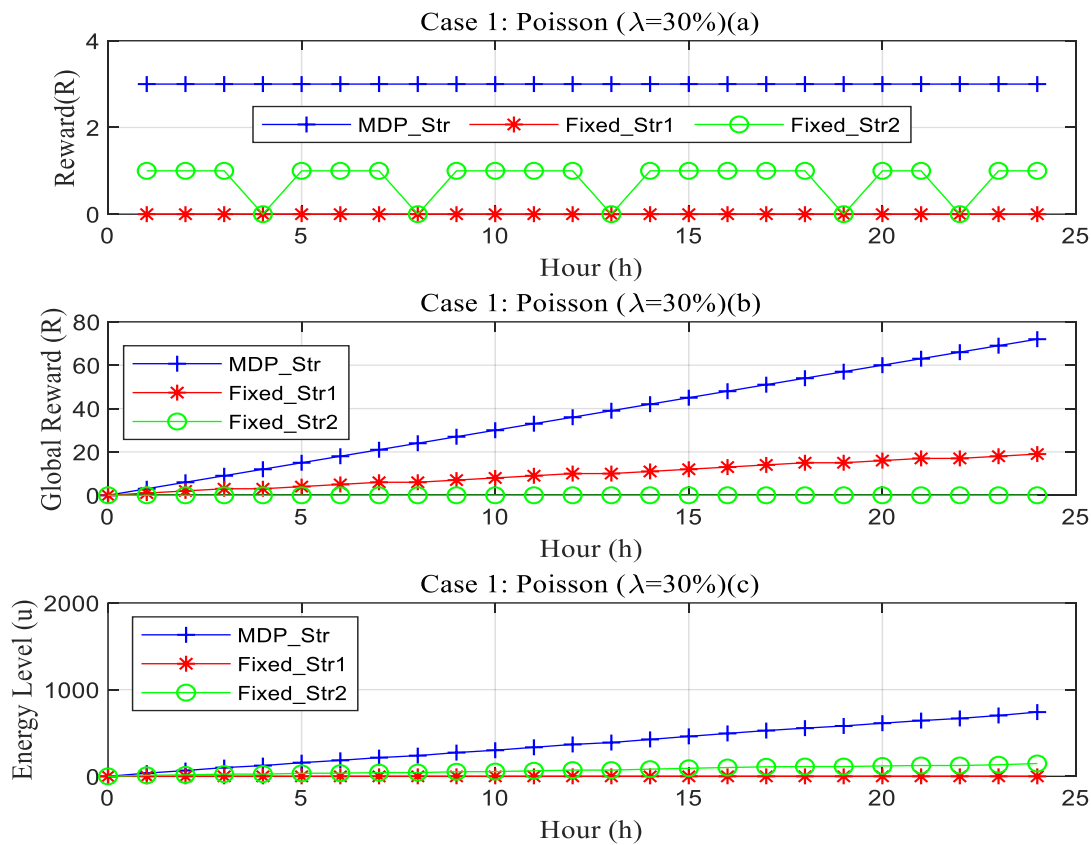


Figure 5.9 - Reward, Global Reward and Energy Gain for case 1.

On the first case, we consider the occupancy level of the parking lot follows a Poisson distribution with lower mean value of 30%. Figure 5.9 shows the amount of reward, global reward and the energy level obtained on using EVs resource for computation.

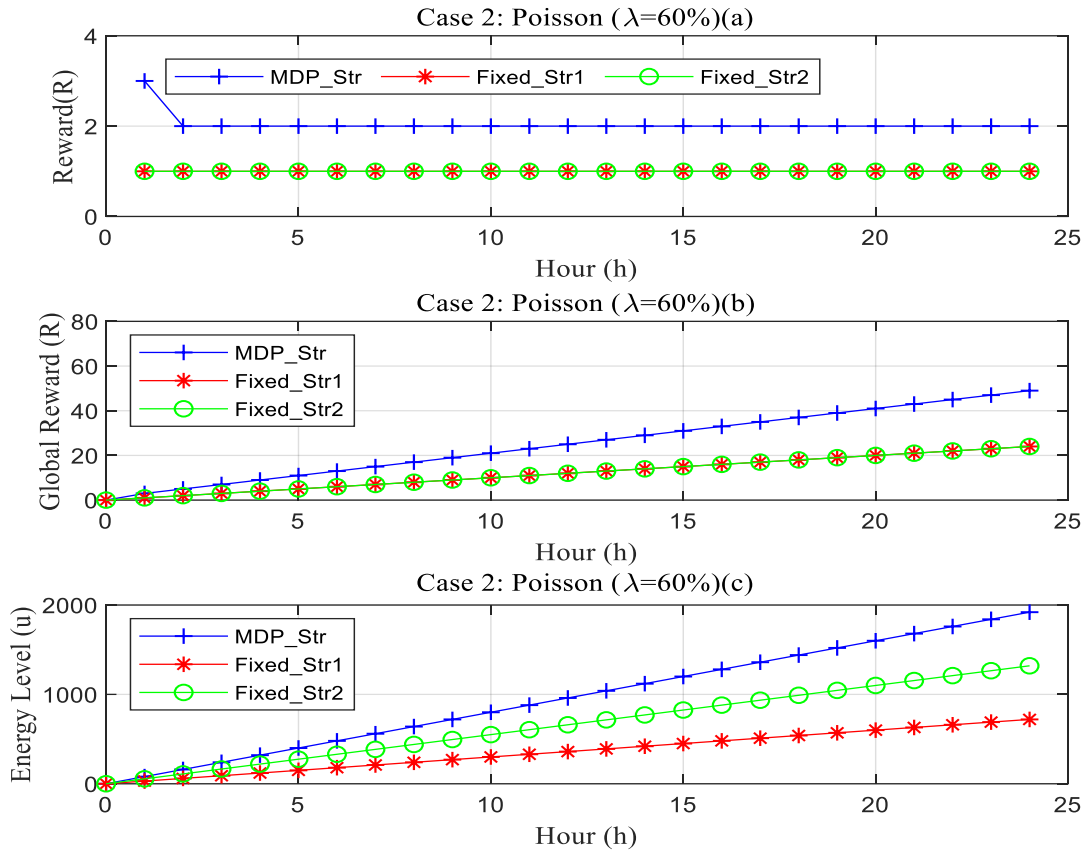


Figure 5.10 - Reward, Global Reward and Energy Gain for case 2.

Then, the occupancy rates in case 2 follows a Poisson distribution with a mean value $\lambda = 60\%$. Figure 5.10 shows the reward at each time and the global reward collected by the decision maker. In addition, the amount of energy gain also depicted as it is shown in Figure 5.10 (a), the *MDP_str* achieves higher values for rewards over time compared to *Fixed_str1* and *Fixed_str2*, which achieved comparable rewards. Consequently, in Figure 5.10 (b) the final overall reward value achieved by *MDP_str* was twice the value achieved by the other two policies. Moreover, *MDP_str* obtained respectively 62.5% and 31.2% better energy gains at time $t=24$ compared to *Fixed_str1* and *Fixed_str2*.

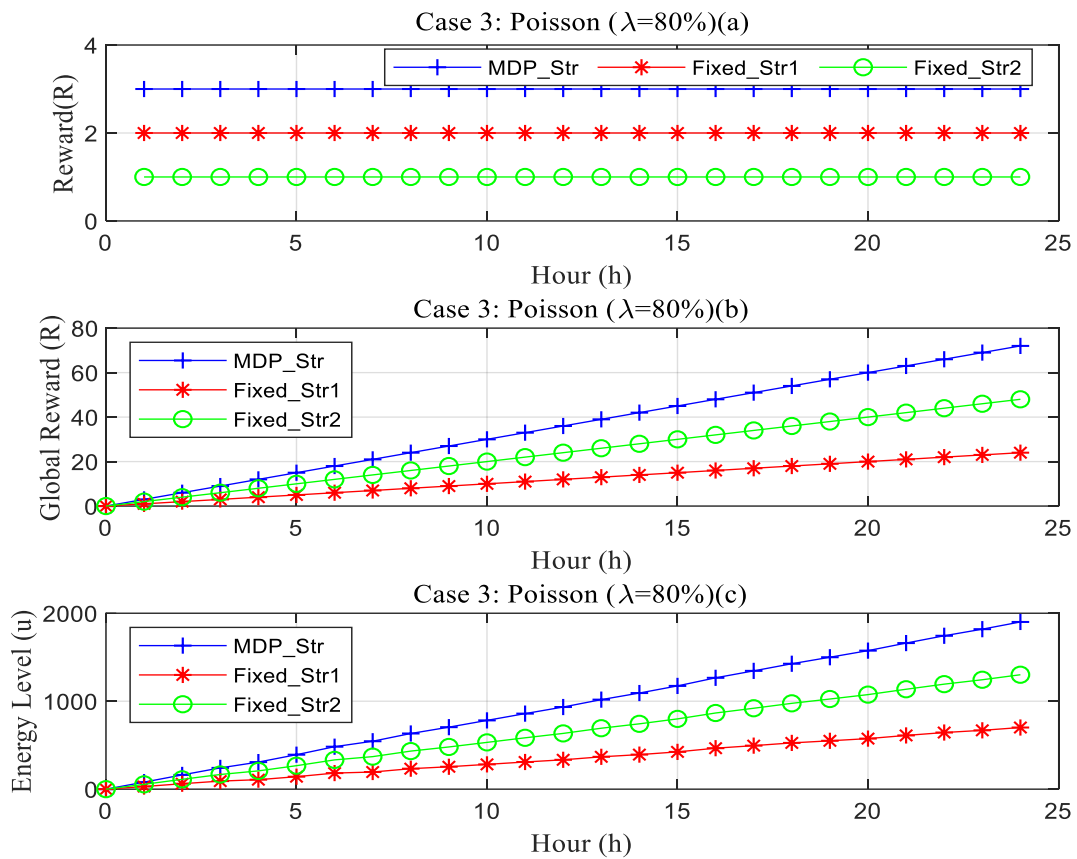


Figure 5.11 - Reward, Global Reward and Energy Gain for case 3.

In the case 3, when the occupancy rate follows a Poisson distribution with mean value of 80%. Figure 5.11 depicted the reward, global reward collected and the energy gain. As it shown in Figure 5.11(a) all strategies collect stable reward with different reward values for all strategies. However, the proposed, *MDP_Str*, strategy outperforms compared to other strategies.

Second, case 4 characterizes a stable energy level over time (30%). As depicted in Figure 5.12, *Fixed_str2* scored zero reward (section (a) & (b)) and no energy gain (section (c)) throughout simulation time. This is because the available energy in EVs was below the fixed threshold. However, *MDP_str* achieved better global reward and better energy gain compared to *Fixed_str1*.

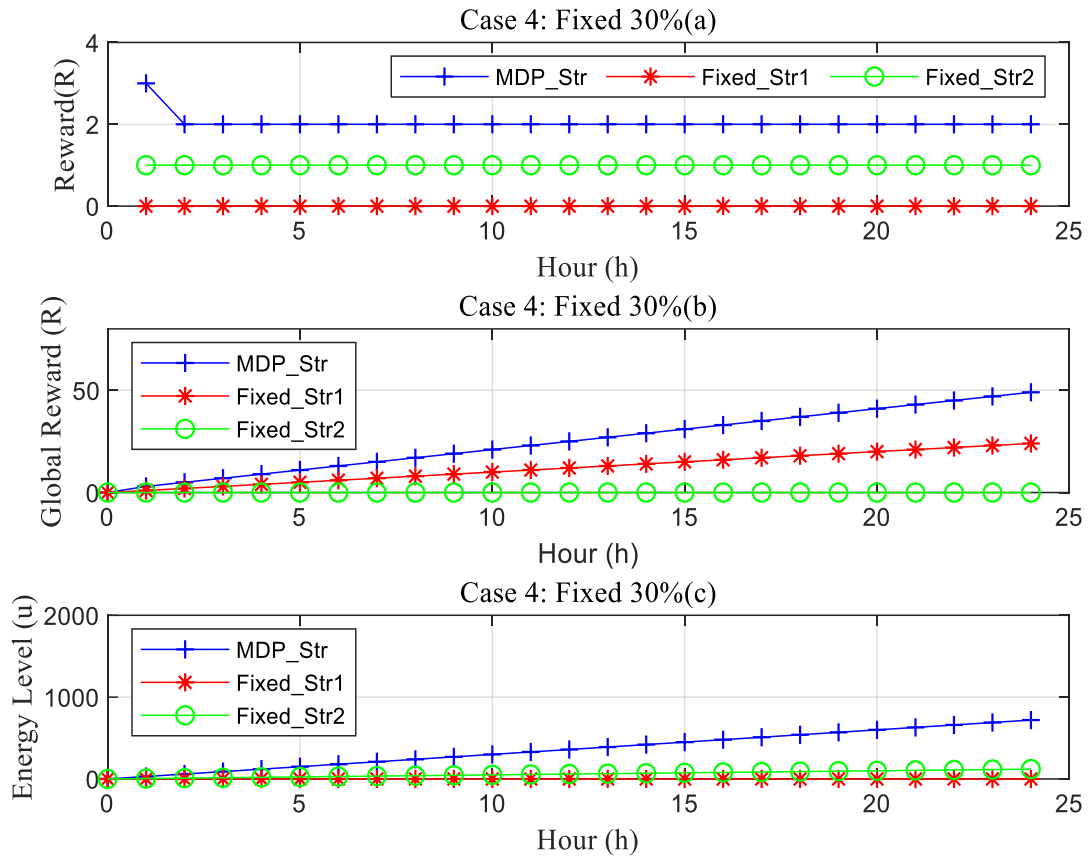


Figure 5.12 - Reward, Global Reward and Energy Gain for case 4.

In the fifth case, case 5 characterizes a stable energy level over time (60%). As depicted in Figure 5.13, all the three strategies scores stable reward with different reward values throughout the simulation time. This is because the available energy in EVs was above the fixed threshold for *Fixed_str1* and *Fixed_Str2*. However, *MDP_str* achieved better global reward and better energy gain compared to *other Fixed strategies*.

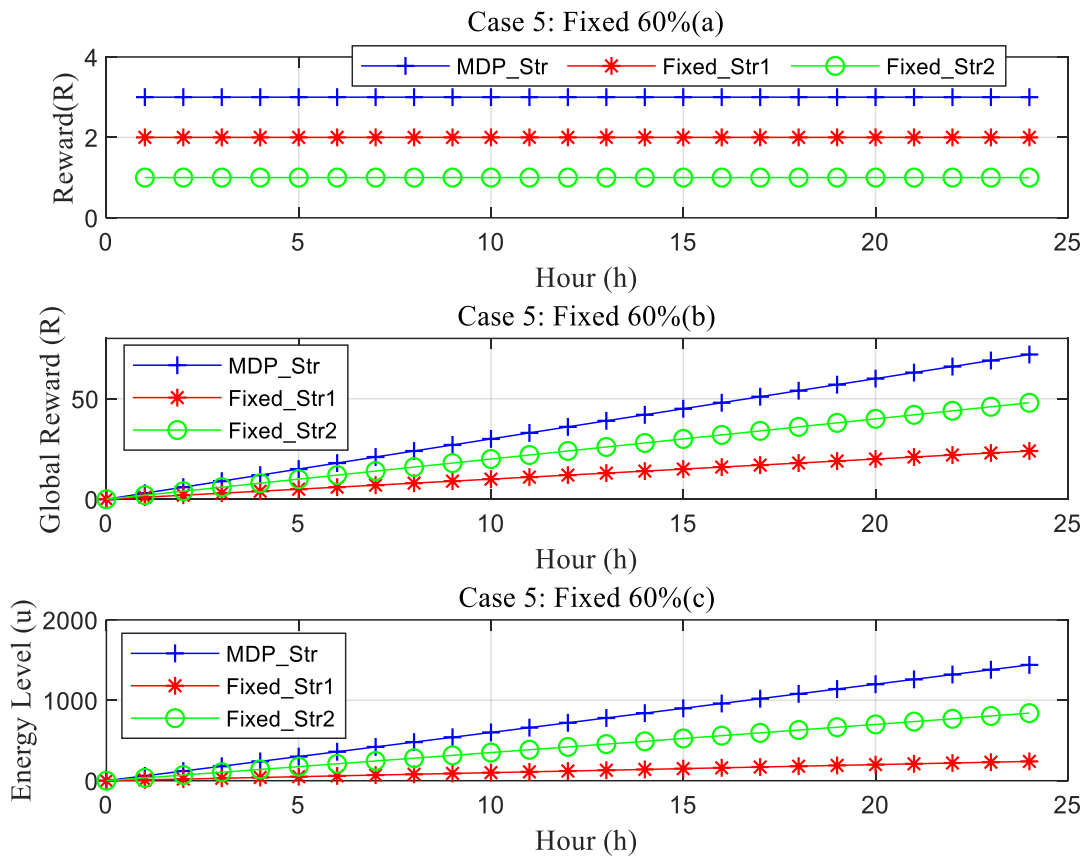


Figure 5.13 - Reward, Global Reward and Energy Gain for case 5.

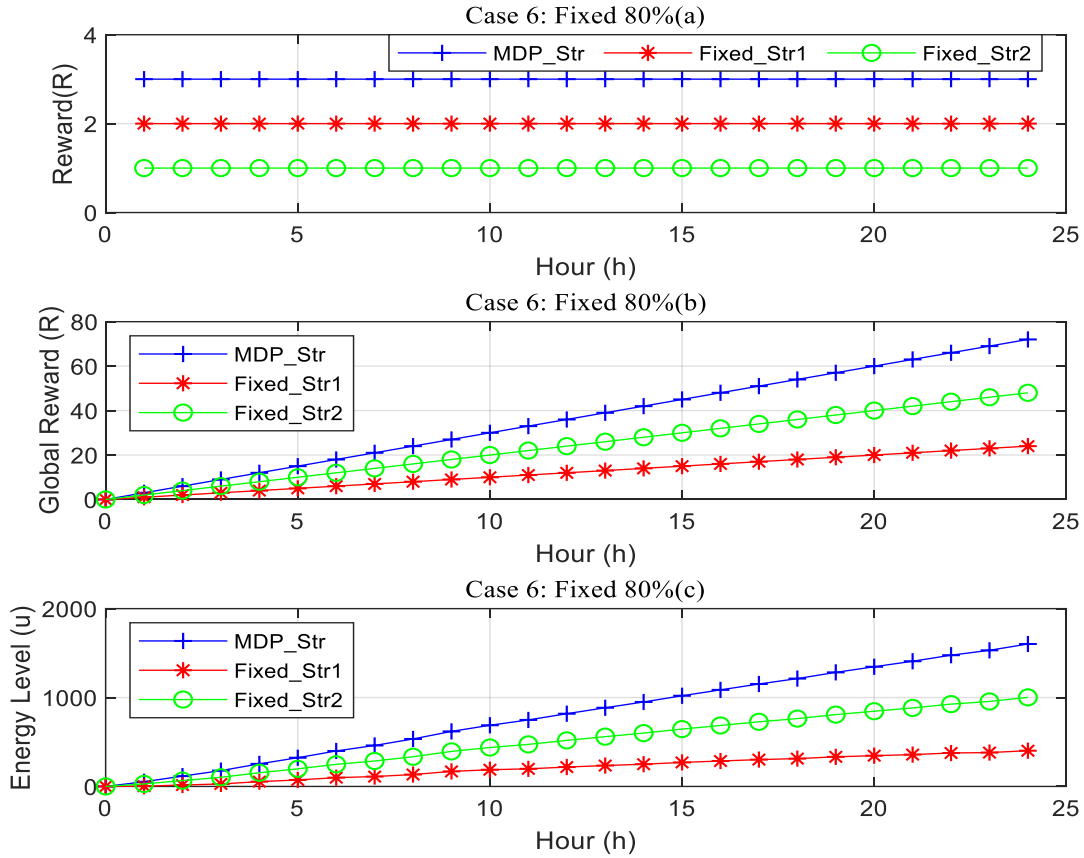


Figure 5.14 - Reward, Global Reward and Energy Gain for case 6.

Finally, in case 6, when the parking occupancy level is as high as 80 %, the results depicted in Figure 5.14 (a) show a rather stable reward value for the three policies with different reward values. However, *MDP_Str* always obtained higher rewards values than those of *Fixed_str1* and *Fixed_str2*. This would also have an influence on the global reward values depicted in Figure 5.14 (b), where *MDP_Str* outperforms once again the other models. Furthermore, the high occupancy levels in the parking lot would result to a better availability of energy resources and consequently serves higher number of users' demands.

Figure 5.14 (c) shows that, at time $t=24$, *MDP_str* has provided 78.7% and 42.4% better energy gain compared to *Fixed_str1* and *Fixed_str2*, respectively. Specifically, *Fixed_str1* and *Fixed_str2* offer respectively 324 and 877 units of energy, while *MDP_str* achieved 1524 units.

As a summary to the performance evaluation section, the analysis of the obtained simulation results has shown the effectiveness of our MDP-based solution for resource allocation. Moreover, we demonstrated that such scheme is suitable to optimize the usage of constrained resources, such as energy in parked EVs. *MDP_str* improves the overall energy used to

compute highly intensive computation tasks by 75% and 58% compared with *Fixed_str1* and *Fixed_str2*, respectively.

5.5 CONCLUSION

EVs popularity has been expanding rapidly due to different reasons like government incentive, higher energy storage and as a solution for global greenhouse gases emission, to mention some. Recently, these EVs become more technologically advanced with computing, communication, and storage capabilities. However, with this powerful potential, EVs are parked for very longer period in the parking lots. With proper incentive approach for the EVs owners, we can use them for computing local application demands.

In this chapter, we proposed a new approach to optimize the usage of the surplus of available energy within parked electric vehicles for the processing of highly intensive computation tasks. The later would be offloaded from nearby mobile devices with constrained energy through a VFC architecture. An optimal decision-making scheme is formulated using a Markov Decision Process in order to maximize the long-term reward of the system. In the proposed model, the decision to accept or reject users' demands to fog nodes (EVs) is done by fog controller based on the available resources in a parking lot. Simulation results show that the proposed strategy outperforms other fixed strategies in terms of rewards and better energy gain. In addition, the effectiveness of the proposed approach by improving the global reward value by 51% and scoring the energy gain of 66% compared to other models.

The proposed solution would help to improve the energy constraint of mobile devices by computing energy constrained mobile application in parked EVs. This solution helps to optimize the use of surplus energy stored in EVs and helps to solve the problem of mobile user constraints.

In the next chapter, we present further solution based on stochastic game resource allocation approach that aims to use surplus parked EVs energy for serving nearby mobile users' applications. Stochastic game resource allocation helps to model the dynamics of both the mobile computation demand and the availability of EVs resources. We also introduce energy and delay for modeling the resource allocation problem.

A STOCHASTIC THEORETICAL GAME APPROACH FOR RESOURCE ALLOCATION IN VFC

6.1 INTRODUCTION

As we presented in chapter 3 of this manuscript, EVs become more technologically advanced with computing, communication, and storage capabilities. On the one hand, with this powerful potential, EVs are parked for very longer period of time in the parking lots. On the other hand, mobile devices have usually limited capabilities in terms of computation power, battery lifetime, storage space and available bandwidth [104]. Thus, in order to enhance the energy constraint of mobile devices, different mechanisms based on the concept of computation offloading are widely used in the literatures. Using this mechanism, mobile users offload their computation task either in the neighboring fog nodes or in the distance cloud servers. This eventually helps to reduce the energy consumption of mobile devices.

Also, as discussed earlier in this manuscript, we consider parked EVs as fog nodes to form a VFC layer that would offer computing services for nearby users. Furthermore, in the previous chapter, we implemented MDP approach to make a decision whether to *Accept* or *Reject* the users demand in fog nodes. However, MDP is a one player game, we considered only the fog node aspects. Thus, in this chapter we tackle multiuser computation offloading for VFC under dynamic environment as a multi-player stochastic game.

To do so, unused resources of parked EVs is aggregated to serve the computation demand of mobile devices with constrained energy, limited storage, and processing capabilities. Moreover, the number of mobile users is dynamic over time. For such types of situations, the long-term system performance is more relevant. Therefore, we adopt a game theory approach for resource allocation in VFC. Game theory is a powerful tool for designing distributed mechanisms, so that the users of mobile devices can locally make decisions based on strategic interactions and

achieve a mutually satisfactory computation offloading solution. Thus, mobile users are defined as the players in our game with two possible states (active/inactive). Furthermore, three different strategies are possible for each player (local computing, offload to the cloud, or offload to VFC). Finally, utility function is defined as a weighted combination between energy consumption and time delay.

In this context, the main contributions of this chapter are summarized as follows: we formulate an innovative resource allocation problem in multiuser context based in a computation offloading scheme. The problem is formulated as a stochastic game in which mobile users, considered as players, compete for time varying available resources within VFC. We also implement a new compatible algorithm to resolve the problem and reach an equilibrium. Besides, a comprehensive simulation work is achieved in order to prove the effectiveness of the proposed solution. Hence, in this chapter, we study multi-player resource allocation problem, where mobile users as a player interacts with the environment in order to minimize their cost, which, unlike the previous chapter, is assumed to be stochastic. This appears in many resource allocation scenarios, with a particular instance mobile device resource allocation in VFC/cloud computing, as considered here.

The remainder of this chapter is organized as follows: Section 6.1 briefly presents the system model. Then, in Section 6.3 we present stochastic resource allocation game formulation, where we spectacle our game model, Nash Equilibrium analysis and stochastic game algorithm. In section 6.4, we detailed experimental evaluation and discussion. Finally, in Section 6.5, we expose after that the results of our proposed algorithm.

6.2 SYSTEM MODEL

We consider a set of mobile users $N = \{1, 2, \dots, N\}$ where each user, i , executes computationally intensive tasks and delay sensitive tasks while trying to preserve its energy and delay. Each computing task $T_i = (D_i, C_i)$, is characterized by the size of required input data D_i including program instructions to execute the tasks and the number of CPU cycle C_i required to perform the computation. In our model, we use aggregated resources of parked EVs resources as a vehicular fog node. Furthermore, we also introduce a fog controller through which users can offload their computation tasks to neighboring vehicular fog nodes or towards distant cloud servers. We use s_i to denote i 's computation strategy. Specifically, together with local computing, i can choose one elements of the set $S_i = \{0, 1, 2\}$, where 0 corresponds to local computing, 1 and 2 corresponds to VFC and cloud computing, respectively.

As stated in Chapter 5 (Section 5.2), we consider EVs with surplus energy as a fog node in which constrained local mobile users is served. However, when there is no resource in EVs users demand is forwarded to cloud servers. Although, we consider the same architecture as the previous chapter, we present here a novel stochastic game formulation approach for solving the problem.

Similar to many previous studies in mobile cloud computing and mobile networking [105], to enable manageable analysis and get useful insight, we consider a quasi-static scenario where the set of mobile device users are remain unchanged (e.g. the change may happen within several seconds); while it may change across different periods. This assumption works for different kind of applications for example face recognition, and natural language processing in which the size of computation offloaded data is smaller and hence computation offloading can be done in a smaller time scale than the time scale of users' mobility. We detail in the following the communication and computation models. Finally, the utility function of the proposed model is presented.

6.2.1 COMMUNICATION MODEL

We assume that each mobile user has a computationally intensive and delay sensitive computation task that needs to be executed. These tasks are computed locally, offloaded either to VFC or eventually to the cloud over fog controllers. The introduced fog controller is

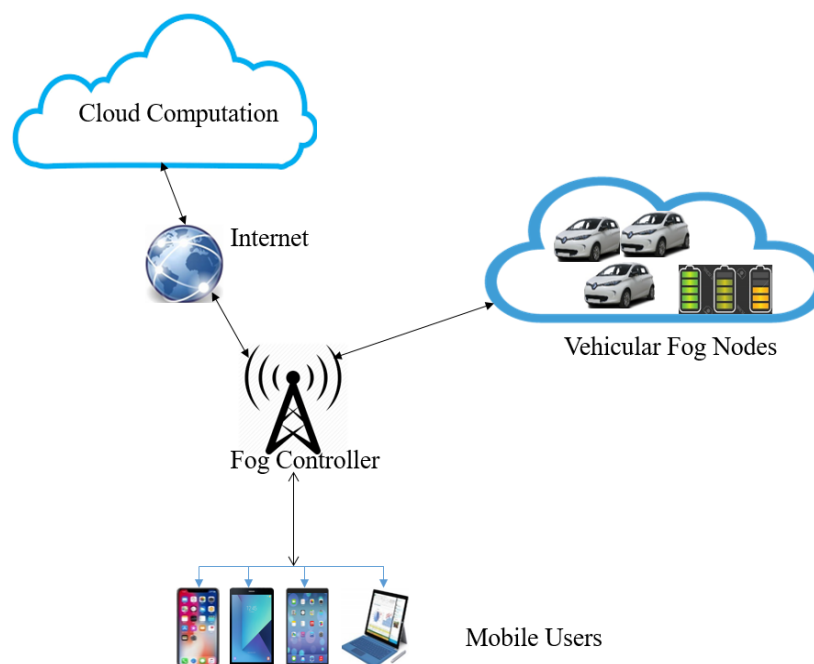


Figure 6.1 - Vehicular Fog Computing Architecture

considered to be able to communicate with vehicular fog nodes and a distant cloud server that manages the uplink/downlink communication of mobile users. Hence, the fog controller uses the wireless mediums deployed nearby (Wi-Fi for example) to communicate with vehicular fog nodes, whereas it uses cellular communication (3G/4G/LTE) to reach the distant cloud server. Therefore, *if user i chooses to offload its computation task to VFC or cloud, the conceivable data rate for each i can be computed, as in [11], by Eq. (6.1):*

$$R_{Data} = B \log_2 \left(1 + \frac{p_i g_{i,0}}{\sum_{j \in A \setminus \{i\}; s_j = s_i} p_j g_{j,0} + \sigma_0} \right) \quad (6.1)$$

Where B is the channel bandwidth, p_i is the transmit power of user i , $g_{i,0}$ is the channel gain from user i and σ_0 is the background noise power. Notice that, for communication model, we see that if too many mobile devices trying to offload their computation tasks at the same time via the same channel, a network congestion could occur. This would eventually lead to low data rates, which in turn significantly affects the overall network performance.

6.2.2 COMPUTATION MODEL

The proposed system uses the same vehicular fog computing system architecture as the previous chapter. However, in this chapter we want to study the dynamics of both the user's mobile computation demand and the availability of EVs resource in the parking lot. Thus, we consider that each mobile device has one or more computation and delay sensitive tasks. The device has to decide where to compute the task either locally, offload to VFC or offload to the cloud server. Thus, three choices are available and for each choice there are different utility function is assumed: local computing, computing on VFC and computing in the cloud. In the following, we introduce the time and energy consumption models in the case of local computing and computation offloading to VFC or cloud.

6.2.2.1 LOCAL COMPUTATION STRATEGY

In the case of local computing, the task has to be processed using local computing resources. In this case, there is no data to send data to the cloud or VFC through wireless media. Thus, the utility function is determined by the available computation capability of local devices, i.e., the computation capability and available energy. The first part is the related to the local execution time needed to perform its computation task that requires C_i amount of CPU cycles and can be expressed as:

$$T_i^0 = \frac{C_i}{F_i^0} \quad (6.2)$$

Where T_i^0 is execution time of the task and F_i^0 denotes the computational capability of mobile devices, and C_i is the requested computational demand of mobile devices.

In order to model the expected energy consumption of local computing, we consider that the energy consumption of local computing is proportional to data size; thus, we use e^L as a coefficient denoting the energy consumption per unit of data and D_i is an input data including program instructions. Thus, the energy function is expressed as:

$$E_i^0 = e^L D_i \quad (6.3)$$

6.2.2.2 COMPUTATION ON VFC STRATEGY

The first possible offloading alternative is to send the computation intensive tasks via fog controller and the processing is done at EVs fog nodes. In comparison with local computing, the delay for computing on VFC requires an extra overhead, this is due to an additional time required to transfer the data to fog nodes. Therefore, if an active user, i , chooses to offload its computation task to VFC in close proximity via fog controller. We denote $T_i^{vfc,off}$ as the communication time to offload data to the VFC and the corresponding energy, $E_i^{vfc,off}$ is the energy needed to offload to VFC. They are defined as follows:

$$T_i^{vfc,off} = \frac{D_i}{R_{Data}}, \text{ and } E_i^{vfc,off} = e_t^{vfc} D_i \quad (6.4)$$

Where R_{Data} is the data transmission rate and e_e^{vfc} the energy required to transmit one unit of data to VFC.

After data transmission, VFC spends energy ($E_i^{vfc,comp}$) with the corresponding computation time ($T_i^{vfc,comp}$) to finish mobile user i 's task. They are also defined as follows:

$$T_i^{vfc,comp} = \frac{C_i}{F_i^{vfc}}, \text{ and } E_i^{vfc,comp} = e_e^{vfc} C_i \quad (6.5)$$

Where e_e^{vfc} denotes energy coefficient to execute one CPU cycle, and F_i denotes computation demand for offloaded task.

Then, the total energy and time cost when the mobile device user, i , choose to perform its task into the nearby VFC (i.e., where strategy $s_i=2$) can be given as:

$$T_i^{vfc} = T_i^{vfc,off} + T_i^{vfc,comp}, \text{ and} \quad (6.6)$$

$$E_i^{vfc} = E_i^{vfc,off} + E_i^{vfc,comp}$$

Furthermore, F_i^{vfc} and $E_i^{vfc,comp}$ are limited by the computing and the energy capacity of VFC F^{VFC} and E^{VFC} as $\sum_i^N F_i^{VFC} \leq F^{VFC}$, and $\sum_i^N E_i^{VFC} \leq E^{VFC}$. If there is no resource or enough resource to compute in VFC, the users' computation demand will be redirected to the cloud. Therefore, the cost incurred to execute the task in the cloud is presented in the next subsection.

6.2.2.3 COMPUTATION ON THE CLOUD STRATEGY

As we presented in chapter 3 of this manuscript, cloud computing is a de facto model for internet-based applications. Specifically, data generated by user devices such as smartphones and other devices are all send to a distant cloud to compute or store their data. Cloud computing provides scalability, can compute the gigantic amount of data in short period of time, also supports user's mobility and high data storage service and other advantage for mobile users. Despite these advantages, there can be several limitations, such as latency and security that impact its popularity. Hence, mobile applications are greatly impacted by the latency of cloud computing; thus, the quality of service is degraded. Therefore, in our proposed system, mobile users offload their computation demand to the cloud only when there is no or enough resource in VFC.

Therefore, the mobile data is transferred, and the computing is done on the cloud server. In order to model the expected time and energy it takes to complete i 's task, we use similar model as in the case of computing in VFC, but in cloud servers where we suppose that there is no resource limitation. The total time required for computation, in cloud servers, is the sum of the time it takes to offload data and the time required to transfer to the cloud server. Therefore, the equation for execution time can be expressed as:

$$T_i^c = \frac{C_i}{F_i^c} + \frac{D_i}{R_{Data}} \quad (6.7)$$

Where F_i^c the computation capability of the cloud server and R_{Data} is the data transmission rate.

The energy cost required to execute the tasks in the cloud is expressed as follows:

$$E_i^c = e_t^c * D_i + D_i * e_e^c \quad (6.8)$$

Where e_t^c denotes energy consumption coefficient required to send one unit of data to cloud server and e_e^c denotes energy consumption coefficient required to compute one unit of data.

6.2.3 UTILITY FUNCTION

In the following, the utility function is presented. Due to the size of the mobile devices, battery storage capacity is limited. Therefore, smart management of this constrained resource is very decisive for extending mobile devices battery lifetime. Moreover, high computation tasks require significant amount of time and energy for completing tasks execution. Taking this in consideration, we model the cost of i as a linear combination of two performance metrics: time and energy. We use the following notations: T_i is the time it takes to finish the computation, E_i is the energy consumption, γ^T and γ^E are used to represent the weighting attribute to energy and time to finish the computation, respectively, where their sum is equal to 1. Here, the aim to use, γ^T and γ^E , in the utility function makes the game model adaptable to resource preferences in different applications. Accordingly, depending on the envisioned application or even the actual situation, different tasks can have different weighting parameters. For example, time sensitive applications would require higher value of γ^T for achieving lower response time, however, if the intention is on energy saving, at the time when energy shortage happens, γ^E will be given a higher value.

Using these notations, for the case of local computing, the cost of i is determined by the local computing time and the consumed energy per CPU cycle:

$$U_i^L = \gamma^T T_i^0 + \gamma^E E_i^0 \quad (6.9)$$

For the case of offloading the mobile users' computation demand to the VFC, the cost of the task is determined by the transmission time, the corresponding energy, and the computing energy with corresponding computing time in VFC. It is expressed as follows:

$$U_i^{VFC} = \gamma^T (T_i^{vfc}) + \gamma^E (E_i^{vfc}) \quad (6.10)$$

For the last case, i.e., offloading to the cloud, the computation cost of i is determined by the transmission time, the corresponding transmit energy, and the computing time in the cloud. It is expressed as follows:

$$U_i^c = \gamma^T (T_i^c) + \gamma^E (E_i^c) \quad (6.11)$$

From equation 6.9, 6.10 and 6.11, We can then express the expected cost of i in strategy profile s as:

Finally, from equation 6.12, we define the overall utility as:

$$U_i = U_i^L + U_i^{VFC} + U_i^c \quad (6.12)$$

$$U = \sum_{i \in N} U_i \quad (6.13)$$

Similar to previous works [106] [2] [108], we do not model the time needed to transmit the computation results from VFC or cloud server to its respective initiator. This is due to the fact that the size of the data resulting from computation task is considered very small compared with the originated data from the source device. This assumption works for a variety of applications like pattern recognition, face and speech recognition, the size of the result of the execution significantly smaller than input data D_i . In the following section, the stochastic game formulation, Nash equilibrium analysis and the proposed algorithm are presented.

6.3 STOCHASTIC RESOURCE ALLOCATION GAME

We tackle throughout this study about using EVs energy for computing local mobile users' energy constrained application. As mobile users are self-interested and selfish in offloading their computation tasks, we formulate a resource allocation game as a stochastic game and detail is presented in the following.

6.3.1 STOCHASTIC GAME FORMULATION

The main reason for using game theory is to simplify and analyze complicated decision making process realized by rational decision makers of conflicting objectives [108]. Moreover, as it is presented in chapter 3 of this manuscript, game theory is an extensively adopted mathematical tool to design and analyze computation offloading of multiple mobile users who are supposed to decide for achieving their own interests. Therefore, each mobile users' objective is to choose the best strategy that maximizes their own objectives. Specifically, each user is modeled as a rational game player that observes and reacts with other players strategy with best response manner. This interaction between the players helps to reach an equilibrium point (called Nash

Equilibrium), where no individual mobile user will change its strategy unilaterally [105]. In addition, by leveraging the intelligence of each individual mobile device game theory is useful for designing decentralized solution with low complexity.

The stochastic game we are proposed is defined with five tuples - $G: \langle N, S, A, P, U \rangle$, where N is the set of active mobile users, S is their states, A is the action space, P is the transition probability of mobile devices, and U is the utility function. The action a_i can be in the set $A_i = \{0. \text{ Compute locally, 1. Offload to VFC, and 2. Offload to Cloud}\}$. $u_i(s, a_i, a_{-i})_{i \in N}$ is the stage payoff function of player u_i , which is the overhead generated by i to carry out one of the three strategies cited above. u_{-i} is the other players strategy of all players excluding player i and a_{-i} is their associated strategies. $P: SA \rightarrow \Delta(S)$ is a measurable transition function where $\Delta(S)$ is the space of probability distribution over S . The game starts at an initial state s_1 , and is played as follows. At each stage $t \in N$, each player $i \in N$ chooses an action $a_i^t \in A_i(s^t)$, receives the stage payoff $u_i(s_t, a_t)$, where $a_t = (a_i^t)_{i \in N}$, and the game moves to a new state s^{t+1} that is chosen according to the probability distribution $P(\cdot | s^t, a^t)$.

The mobile user i utility (u_i) function is the expected sum of discounted stage payoff function for each user i is as follows:

$$\bar{u}_i = E(\gamma \sum u_i(s, a_i, a_{-i})) \quad (6.14)$$

Where γ = the discount factor.

Each mobile user independently adjusts its strategy to minimize its expected payoff function, which is expressed as:

$$\begin{aligned} \bar{u}_i^* &= \min (\varphi_i^\pi(s, a_i, a_{-i})) \\ &= \min (u_i + \gamma \sum_{s' \in \Omega} p(s' | s, a_i, a_{-i}) \sum \pi(s', a'_i)) \sum_{a'_i \in A_i, a'_{-i} \in A_{-i}} \pi(s', a'_i) \\ &\quad \pi(s'_i, a'_{-i}) \varphi_i^\pi(s', a'_i, a'_{-i}) \end{aligned} \quad (6.15)$$

Where

- \bar{u}_i^* denotes the optimal long-term utility of mobile user i from state S ,
- $\min (\varphi_i^\pi(s, a_i, a_{-i}))$: the long-term utility of player i playing a_i and other players applying a_{-i} ,
- $P(s' | s, a_i, a_{-i})$, the transition probability from state s to s' when player i takes action a_i and other player a_{-i} ,

- s is state space,
- a_i set of actions of player i .

By choosing a_i , user i can decide where to process its task to minimize its cost function u_i .

6.3.2 NASH EQUILIBRIUM (NE) ANALYSIS

In order to attain a stable convergence state in a stochastic game, all players need to reach a consensus, called Nash Equilibrium (NE). This equilibrium point is defined as a stable state of convergence, where each player cannot further improve its utility by unilaterally changing its strategy. For every player i and every strategy profile $\sigma = (\sigma_i)_{i \in N}$, we denote the strategy profile of all players, except player i , by $\sigma_{-i} = (\sigma_j)_{j \neq i}$.

Definition 1: let $\varepsilon \geq 0$. A profile of strategies σ is a T-stage ε -equilibrium if it is a λ -discounted ε -equilibrium if:

$$\gamma_i^\lambda(s_1, \sigma) \geq \gamma_i^\lambda(s_1, \sigma'_i, \sigma_{-i}) - \varepsilon, \forall s_1 \in S, \forall i \in N, \forall \sigma'_i \in \Sigma_i \quad (6.16)$$

The utility that corresponds to an ε -equilibrium, i.e. $\gamma^\lambda(s_1, \sigma)$, is called an ε -equilibrium at the initial state s_1 . According to Solan [109], when both state and action spaces are finite, λ -discounted ε -equilibrium exists.

Theorem 1: *Every stochastic game with finite state and action spaces has a λ -discounted equilibrium in stationary strategies [109].*

Proof: Let $M = \min_{s,a} |u_i(s; a)|$ be a bound on the absolute values of the payoffs. Set $X = \prod_{i \in N, s \in S} (\Delta(A_i(s)) \times [-M, M])$. A point $(x_{i,s}^A, x_{i,s}^U)_{i \in N, s \in S} \in X$ is a collection of one mixed action and one payoff to each player at every state. For every $u = (u_i)_{i \in N} \in [-M; M]^{N \times S}$ and every $s \in S$, we define a matrix game $G_s^\lambda(u)$ as follows:

The action spaces of each player i is $A_i(s)$:

The payoff to player i is:

$$\lambda u_i(s, a) + (1 - \lambda) \sum_{s' \in S} q(s'|s, a) u_i(s'). \quad (6.17)$$

We define a set-valued function $\varphi: X \rightarrow X$ as follows:

For every $i \in N$ and every $s \in S$, $\varphi_{i,s}^A$ is the set of all best responses of player i to the strategy vector $x_{-i,s} := (x_{j,s})_{j \neq i}$ in the game $G_s^\lambda(u)$. That is,

For every $i \in N$ and every $s \in S$, $\varphi_{i,s}^V(x, v)$ is the maximal payoff for player i in the game $G_s^\lambda(v)$, when the other players play x_{-i} :

The set-valued function φ has convex and non-empty values. It turns out that every fixed point of φ defines a λ -discounted equilibrium in stationary strategies. \square

$$\varphi_{i,s}^v(x, u) := \min_{y_{i,s} \in \Delta(A_i(s))} (\lambda r(s, y_{i,s}, x_{-i,s}) + (1 - \lambda) \times \sum_{s' \in S} q(s'|s, y_{i,s}, x_{-i,s}) u_{i,s'}). \quad (6.19)$$

$$\varphi_{i,s}^A(x, u) := \left\{ \operatorname{argmin}_{y_{i,s} \in \Delta(A_i(s))} \lambda r_i(s, y_{i,s}, x_{-i,s}) + (1 - \lambda) \sum_{s' \in S} q(s'|s, y_{i,s}, x_{-i,s}) u_{i,s'} \right\}. \quad (6.18)$$

In the next subsection, we present the proposed stochastic game algorithm.

6.3.3 STOCHASTIC GAME ALGORITHM

As presented in the previous subsection, we formulate the resource allocation problem as a stochastic game and showed the existence of Nash Equilibrium. In this subsection, we detailed the proposed stochastic game algorithm.

Therefore, the details of the different steps are illustrated in Algorithm 6.1. Specifically, each player i selects its strategy according to the strategy selection probability matrix. The latter has been initialized as a uniformly distributed probability function between the three possible strategies. Then, each player i updates its probability vector based on the possible outcome

Algorithm 6.1: Stochastic Game Algorithm	
1	Input: A set of computation tasks: $\{Ti = (Di, Ci)$.
	Output: strategy, Utility Value (U).
2	Initialization
3	• Each i sets its strategy as a uniform distribution
4	• $P_i(0) = \frac{1}{3}$; $b \in (0, 1)$ learning step.
	• $r_i(n) = 1 - \rho u_i(n)$ is an action-payoff.
5	Repeat:
6	• Each active mobile user j selects a strategy $a_i(n)$ according to its current strategy selection probability vector $p_j(n)$.
7	If $u_j(n) < u_j(n+1)$ then
8	• $u_i(n+1) = u_j(n)$
9	Else
10	• Update the strategy selection probability
11	• $P_{i,m}(n+1) = P_{i,m}(n) + br_i(n)(1 - P_{i,n}(n))$, $m = a_i(n)$
12	• $P_{i,m}(n+1) = P_{i,m}(n) - br_i(n) P_{i,m}(n)$, $m \neq a_i(n)$
13	End
14	Until NE reaches

provided by the action-reward values received from the dynamic environment. We denote the strategy selection probability vector for a mobile user i as $P_j = (P_{j0}, P_{j1}, P_{j2})$, where P_{jm} is the

probability to select strategy s_i on each iteration. The proposed algorithm works in an iterative way, where each mobile user independently chooses its offloading strategy based on the values of the probability matrix over the strategy space.

Each player would eventually receive an action reward from the dynamic environment. Thus, computation of tasks with lower cost receives higher reward, which will result to higher probability of selection. After continuously interacting with the dynamic environment, each mobile user will reach its optimal strategy with probability 1.

The proposed algorithm is characterized by the following features: i) offloading strategy selection is based on the probability distribution over strategy space, and ii) the probability space is updated according to the received reward from the dynamic environment. Moreover, the updating rules in the algorithm are only dependent on mobile users' action reward experiences (see Algorithm 6.1-line No. 4), this makes the decision-making process via the proposed solution fully distributed. Each mobile computes their utility and selects strategy with lower utility value experiences (see Algorithm 6.1 - line No. 8). Finally, the strategy selection probability is updated in line 11 and 12 of the proposed Algorithm 6.1.

In the next section, we present the experimental results and discussion of our proposed model.

6.4 EXPERIMENTAL RESULTS AND DISCUSSIONS

During this section, we evaluate the performance of resource allocation in VFC through simulation using Matlab. Our main purpose is to find the best possible solution for the resource allocation problem that minimizes computation overhead in terms of the overall energy consumption and time. Specifically, the overall overhead is equally dictated by energy consumption and time. Then, we detail the simulation scenarios followed by the given results and discuss them.

6.4.1 SIMULATION SCENARIOS

For conducting the simulation, we setup a scenario where a group of mobile users scattered in the coverage of the Fog controller. Each user i may randomly be active or inactive according to uniform distribution probability within $[0, 1]$. Different learning step-sizes for the algorithm

is considered ranging from 0.1 to 0.5. The main simulation parameters used in our simulation are summarized in Table 1.

Parameters	Values
Number of users (N)	{10, 15, 20, 25}
Coverage of Fog Controller	50
User's active probability	[0, 1]
Bandwidth of Channel (B)	70 Mbps
Local Computation power	0.8 GHz
VFC Computation power	15 GHz
Cloud Computation power	30 GHz
Weight of Computational Energy (μ_i^E)	[0.3,0.5,0.7]
Weight of Computation time (μ_i^T)	$1-\mu_i^E$
Data size (D_i)	5000KB

Table 6.1 - Main Simulation Parameters

In order to efficiently evaluate the performances of the proposed solution, different performance metrics have been considered:

- First, the impact of the number of mobile users on the performances of the overall system has been evaluated. This is used to show the scalability of the solution and the consistency of the obtained results when the number of mobile users increases.
- Also, the overall system wide cost, which represents the total computation overhead of the system, is evaluated. A smaller average value for the system-wide computational cost is more desirable.

6.4.2 RESULTS AND DISCUSSIONS

First, to evaluate the convergence of the proposed solution, we plot the strategy selection probabilities for 25 selected users with different learning step-sizes. Step-size influences the convergence time of the algorithm and has greater impact on the oscillation behavior of the graphs. As shown in Figure 6.2, with a smaller step-size ($b = 0.1$), the users select their strategy from uniform probability distribution. However, when it continues to use the proposed algorithm, they keep updating their strategy selection probability and did not converge for 100 unit of simulation time. Whereas, in Figure 6.3, we use a step-size b equal to 0.5. Initially, the

mobile user's selects their offloading strategy according to the probability distribution then the user's strategy selection probability is keep updating. Finally, after some iteration, the probability to choose offloading strategy to VFC converges to 1. This result shows that, after convergence, mobile users choose VFC to perform their computation tasks.

To evaluate the performance of our Algorithm for different number of mobile users, we plot two figures with learning step $b = \{0.1, 0.5\}$. Figure 6.4 shows the performance of our algorithm in terms of average system wide cost for different number of mobile users with learning step-size $b = 0.1$. In addition, we evaluated the impact of different number of mobile users when the step-size $b = 0.5$ (see in Figure 6.5) to show its impact on the average system wide cost for

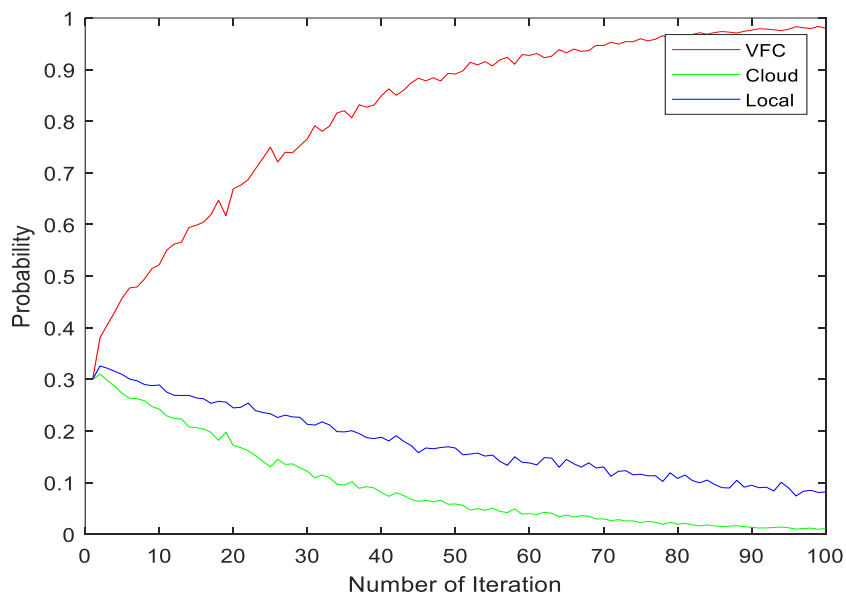


Figure 6.2 - Strategy selection probability when $b = 0.1$.

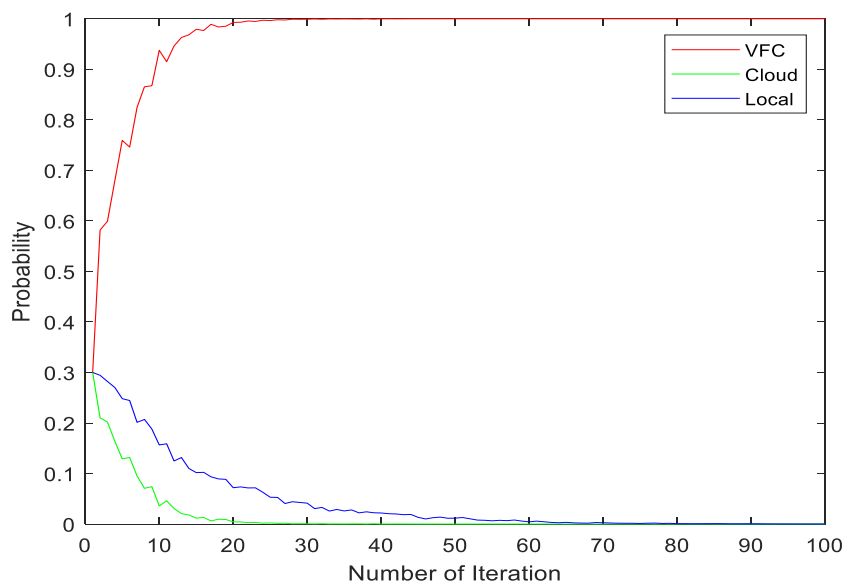


Figure 6.3 - Strategy selection probability when $b = 0.5$.

local, VFC and cloud offloading strategies. The consumed average system wide cost is lower when the step-size is larger ($b = 0.5$). This happens because the algorithm takes less time to reach convergence. In addition, the consumed average cost increases when the number of mobile users increases in a stable manner in both cases, VFC and cloud. Whereas, the average cost of local computation is high in both cases ($b = 0.1$ and $b = 0.5$) compared to VFC and cloud.

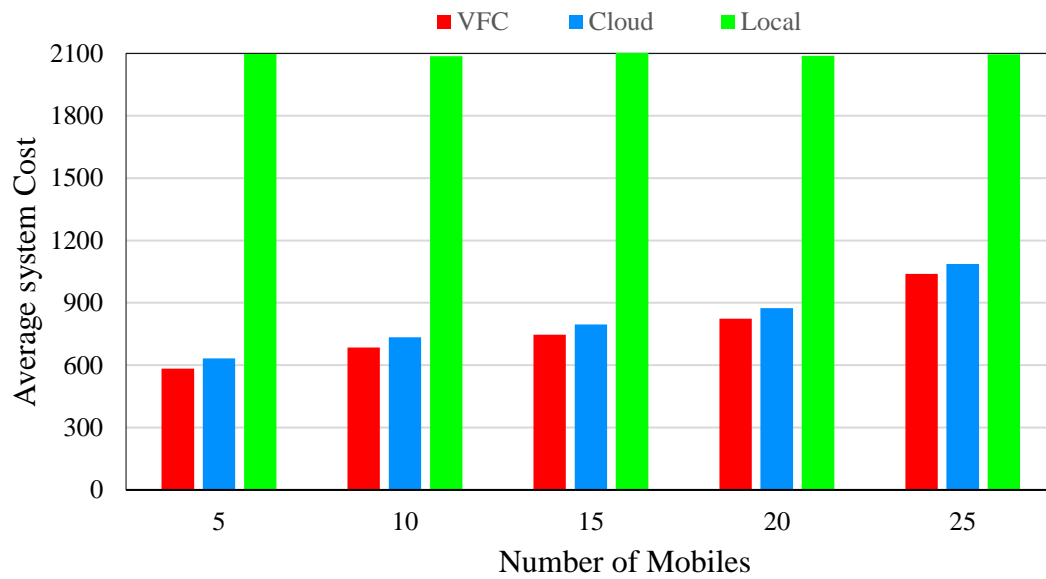


Figure 6.4 - Average system wide cost when $b = 0.1$.

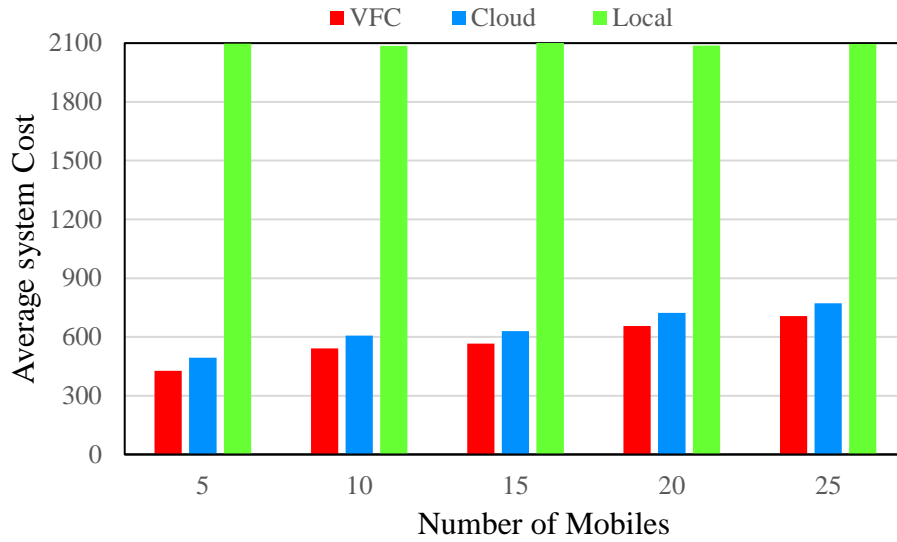


Figure 6.6 - Average system wide cost when $b = 0.5$.

Figure 6.6 shows the system wide cost for the two step-sizes. As it is shown, the cost of offloading tasks at $b = 0.1$ is lower than $b = 0.5$. This is because the convergence speed is slow when the value of b decreases and takes more time to converge. When the convergence speed requires more time, the system will increase the system wide cost.

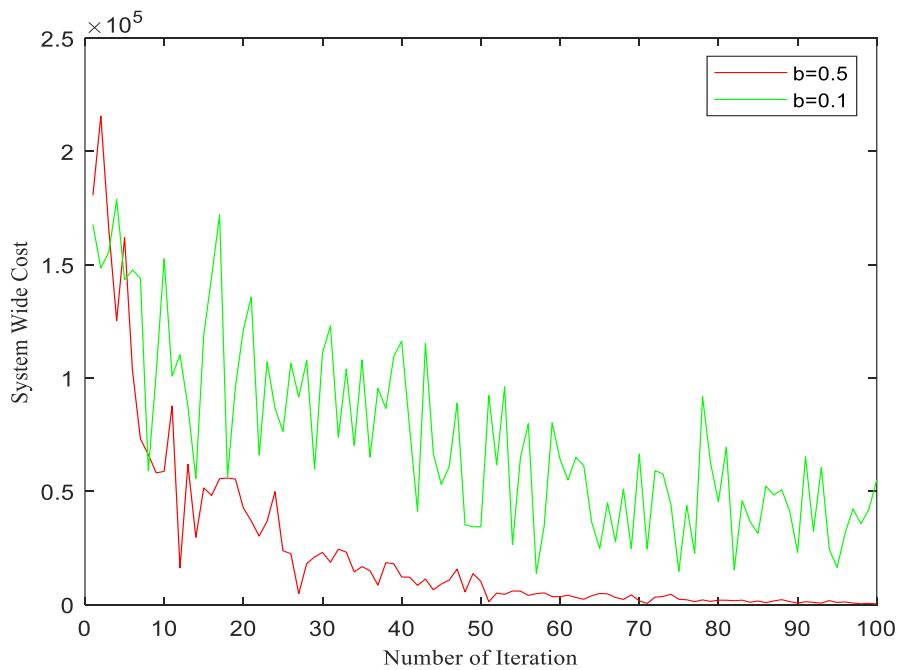


Figure 6.5 - System Wide Cost.

In addition, for more comprehensive simulation results, we consider the impact of changing data size and CPU cycle at the same time. As it is shown in Figure 6.7 and 6.8, we evaluate the average cost for different data size and CPU cycle, with learning step values of $b = \{0.1, 0.5\}$. From Figure 6.7, we can see that the average system wide cost of computing application demands in the cloud server is higher cost than the average cost computing in VFC (Figure 6.8). Whereas, both offloading strategies (i.e., VFC and cloud) performances are not significantly affected by the amount of different data size. In addition, we realize that the proposed resource allocation approach (i.e., VFC based resource allocation) minimizes the computation cost of local application demands.

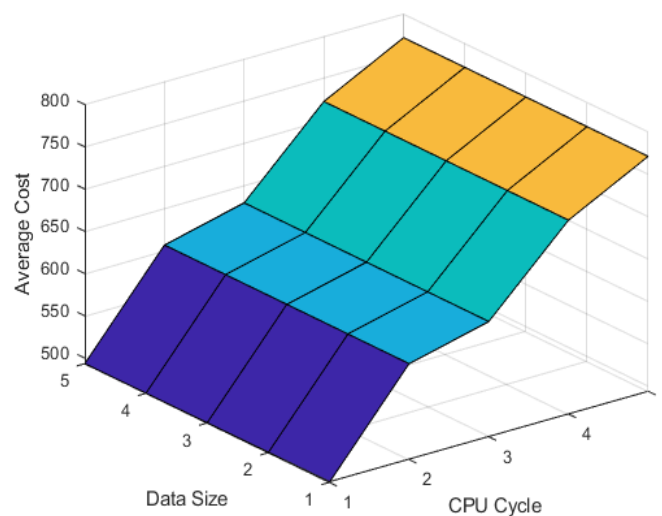


Figure 6.8 - Cloud Average system wide cost when $b=0.1$.

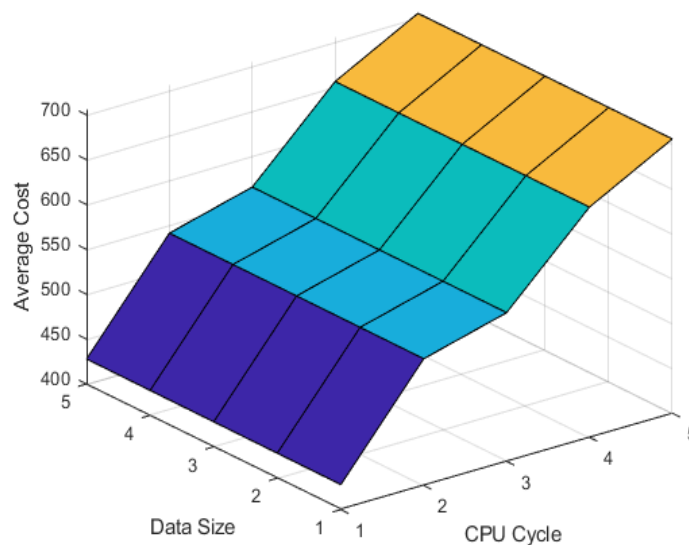


Figure 6.7 - VFC Average system wide cost when $b=0.5$.

6.5 CONCLUSION

This chapter presents a new approach to make better usage of underutilized resources in parked EVs in terms of energy, storage, and computation capabilities for local application demands. The key idea is to use these idle resources for serving highly intensive and energy hungry computation demands of nearby mobile users. Furthermore, the proposed scheme considers using aggregated resources of parked EVs as fog nodes, in order to extend the computation capabilities of constrained neighboring mobile devices. The amount of available resources in vehicular fog node and the number of active computation demands are highly dynamic.

Therefore, a stochastic theoretical game has been formulated in order to address this resource allocation problem, with the objective is to minimize the overall overhead in terms of energy and delay. Furthermore, a distributed algorithm has been implemented and tested through extensive simulation work. The obtained simulation results show the effectiveness of the proposed solution compared to classical models. In the next chapter we conclude this manuscript and we present some perspectives.

CONCLUSION AND PERSPECTIVES

This thesis focuses on the use of abundant resources of parked EVs for serving resource constrained mobile application demands. In this last chapter, we present conclusions about the work presented in this thesis and propose future perspectives for possible extensions. Therefore, we presented, on the first section the conclusion. Then, the perspectives are presented in the second section.

7.1 CONCLUSION

With the advancement of information and communication technology, electric vehicles are introduced with communication, computation, and storage capabilities. In addition, the advancement in battery technology improves the energy storage capacity of EVs battery that helps to improve energy storage capacity and extend the driving range of the vehicles. However, EVs with these capabilities and higher stored energy stay more than 90% of the time at the parking lot. Therefore, we observe that parked EVs have rich and underutilized resources for computing local users demand.

For these reasons, we have focused our attention during this thesis on using abundant EVs resource to serve constrained mobile application demands. Hence, in this thesis we have dealt with allocation of parked EVs resources for mobile users. Thus, the local mobile application demands are offloaded to the nearby parked EVs, leading to a new paradigm called vehicular fog computing architecture.

To do so, at first, we started to study and provide background information about cloud, edge computing, and electric vehicles. Hence, we have reviewed major concepts of cloud computing and different edge computing paradigms. In addition, we also presented detailed overview about electric vehicles. Here, our goal was to get a deeper understanding on the cloud computing and different paradigm of edge computing, in order to get position of the thesis in relation to existing research and models. Furthermore, we also reviewed the state of the art on resource allocation in cloud and fog computing paradigm. Therefore, we have reviewed

different algorithms used in cloud and fog computing architecture. Afterward, we have presented our different proposals.

In the first proposal of this thesis, we focused on efficient electric battery management for resource allocation, where EVs battery resource are used to serve users demand in consideration with the state of charge (SOC) and state of health (SOH) of the battery. Hence, only EVs batteries with higher value of state of charge and health have been used for serving users' demands. Thus, we use Matlab Simulink tool to model the battery of EVs and task profile is created as a workload cycle. Then, we have showed that EVs that serve users demands were selected according to their SOC and SOH levels. Therefore, EVs with lower SOH values are not used for computation service.

Then, we presented our second proposal, using MDP model, to optimize the use of parked EVs resources for both serving users' demands and mobility. We have showed that our proposal increases the reward of the decision maker while keeping resources for vehicle mobility. However, MDP is a one player game in which we only model the dynamics from EVs perspective. Thus, in our third proposal, we presented a new approach of resource allocation to model both mobile devices and EVs as stochastic game. Thus, the resource allocation problem is formulated in order to minimize the overall overhead in terms of energy and delay. Thus, a distributed algorithm has been proposed and tested through simulation work. The obtained results confirm that the proposed VFC paradigm outperforms compared to computing in local and cloud servers.

7.2 PERSPECTIVES

This section is devoted to study new research perspectives and possible new direct applications on the contribution provided during the process of this thesis. In fact, in this thesis some issues have not been addressed and represent issues that will be addressed in the future. Thus, future works can be further extended to the following directions:

- In our proposal, EVs resources are used to serve all applications demands of mobile devices. On one hand, resources of EVs are limited and dynamic overtime. On the other hand, mobile applications are different in their types and quality of service (QoS). Therefore, as future work, it is appropriate to design a task scheduling mechanism that uses the applications priorities. Thus, application demands with lower

priority will wait until higher priority one is completed or redirected to cloud. However, higher priority tasks are served in VFC as they arrive.

- Resource prediction model: EVs arrival and departure are dynamic over time. To create a stable and manageable system, a predictive approach for future available resources is essential. Hence, as a future work, resource prediction on the parked EVs will be studied that maximizes the profit of services providers.
- Creating EVs battery profile: since using the same batteries for computation significantly reduce the battery lifetime. Thus, as a future work, all EVs batteries used for computation are registered and stored. Thus, a history of EVs used for computation service is kept. Therefore, EVs which were not used for computation service frequently will be selected for computation in the future.
- Developing an incentive and pricing model for EV owners: in our proposed model, EVs resources are used for computation with the permission and willingness of the owners. Due to the utilization of EVs resources for computation, the design and implementation of different incentive and pricing scheme should be studied. Therefore, the proposal and design of incentive approach should also be investigated.

➤ **International Conference**

- H. M. Birhanie, M. A. Messous, S.-M. Senouci, E.-H. Aglzim, and A. M. Ahmed, “MDP-Based Resource Allocation Scheme Towards a Vehicular Fog Computing with Energy Constraints,” *IEEE Glob. Commun. Conf.*, pp. 1–6, 2018.

➤ **National Conference**

- Habtamu M. Birhanie, Joelle Klaimi, Sidi-Mohammed Senouci, El-Hassane Aglzim, “Markov Decision Based Approach for Resource Allocation in Vehicular Fog Computing”, *FUTURMOB’2017*.
- Habtamu M. Birhanie, Sidi-Mohammed Senouci, El-Hassane Aglzim, Ahmedin M. Ahmed, “MDP-Based Resource Allocation Scheme Towards a Vehicular Fog Computing”. poster paper, *FUTURMOB’2018*.

➤ **Publications Under Review**

- H. M. Birhanie, M. A., Sidi-Mohammed Senouci, Mohammed Ayoub Messous, Amel Arfaoui, Ali Kies, “A Stochastic Theoretical Game Approach for Resource Allocation in Vehicular Fog Computing”, Accepted as poster paper to *CCNC 2020* Conference.
- H. M. Birhanie, M. A., Sidi-Mohammed Senouci, Mohammed Ayoub Messous, “Resource allocation for Vehicular Fog Computing: A Stochastic Theoretical Game Approach”, Journal paper in progress, will be submitted to *International Journal of Communication Systems – Wiley Online Library*

BIBLIOGRAPHY

- [1] F. H. Rahman, A. Yura, M. Iqbal, and S. H. S. Newaz, "Street Parked Vehicles Based Vehicular Fog Computing: TCP Throughput Evaluation and Future Research Direction," *2019 21st Int. Conf. Adv. Commun. Technol.*, pp. 26–31, 2019.
- [2] K. Kumar and Y.-H. Lu, "Cloud Computing Can Offline Computation Save Energy," *IEEE Comput.*, vol. 43, no. April, pp. 51–56, 2010.
- [3] R. Cogill *et al.*, "Parked cars as a service delivery platform," in *2014 International Conference on Connected Vehicles and Expo, ICCVE 2014 - Proceedings*, 2014, pp. 138–143.
- [4] Z. Ning, J. Huang, and X. Wang, "Vehicular Fog Computing: Enabling Real-time Traffic Management for Smart Cities," *IEEE Wirel. Commun.*, vol. 26, no. February, pp. 87–93, 2018.
- [5] D. Evans, "The Internet of Things - How the Next Evolution of the Internet is Changing Everything," *CISCO white Pap.*, no. 1, pp. 1–11, 2011.
- [6] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile Edge Computing: A Survey," *IEEE Internet of Things Journal*, vol. 5, no. 1. pp. 450–465, 2018.
- [7] B. Ravandi and I. Papapanagiotou, "A Self-Learning Scheduling in Cloud Software Defined Block Storage," *IEEE Int. Conf. Cloud Comput. CLOUD*, vol. 2017-June, pp. 415–422, 2017.
- [8] G. E. Outlook, "Global EV Outlook 2019 to Electric Mobility." 2019.
- [9] M. Khabazian, S. Aissa, and M. Mehmet-Ali, "Performance modeling of message dissemination in vehicular ad hoc networks with priority," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 1, pp. 61–71, 2011.
- [10] C. Zhu, G. Pastor, Y. Xiao, and A. Ylajaaski, "Vehicular Fog Computing for Video Crowdsourcing: Applications, Feasibility, and Challenges," *IEEE Commun. Mag.*, vol. 56, no. 10, pp. 58–63, 2018.
- [11] G. T. Mell Peter, "The NIST Definition of Cloud Computing Recommendations of the National Institute of Standards and Technology," *NIST Spec. Publ.*, vol. 145, pp. 1–7, 2011.
- [12] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile Edge Computing A key technology towards 5G," *ETSI White Pap.*, vol. 11, 2015.
- [13] R. Roman, J. Lopez, and M. Mambo, "Mobile edge computing, Fog et al.: A survey and analysis of security threats and challenges," *Futur. Gener. Comput. Syst.*, vol. 78, pp. 680–698, 2018.
- [14] A. Bezeichnung, "Mobile-Edge Computing – Introductory Technical White Paper," *ETSI White Pap.*, no. 1, pp. 1–8, 2014.
- [15] A. H. Sodhro, S. Pirbhulal, and V. H. C. De Albuquerque, "Artificial Intelligence Driven Mechanism for Edge Computing based Industrial Applications," *IEEE Trans. Ind. Informatics*, 2018.

- [16] E. Pencheva, "Location Service in Mobile Edge Computing," in *2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN)*, 2017, pp. 617–622.
- [17] X. Li and J. Wan, "Proactive caching for edge computing-enabled industrial mobile wireless networks," *Futur. Gener. Comput. Syst.*, vol. 89, pp. 89–97, 2018.
- [18] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the first edition of the MCC workshop on Mobile cloud computing - MCC '12*, 2012, no. August 2012, pp. 13–16.
- [19] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog Computing and Its Role in the Internet of Things Characterization of Fog Computing," 2012, pp. 13–15.
- [20] C. Mouradian, D. Naboulsi, S. Yangui, R. H. Glitho, M. J. Morrow, and A. Polakos, "A Comprehensive Survey on Fog Computing : State- of-the-art and Research Challenges," *IEEE Commun. Surv. Tutorials*, vol. 20, no. 1, pp. 416–464, 2017.
- [21] C.-H. Hong and B. Varghese, "Resource Management in Fog/Edge Computing: A Survey," 2018.
- [22] M. Aazam and S. Korea, "E-HAMC : Leveraging Fog Computing for Emergency Alert Service," in *EEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*, 2015, pp. 518–523.
- [23] C. Tang, X. Wei, C. Zhu, W. Chen, and J. J. P. C. Rodrigues, "Towards Smart Parking Based on Fog Computing," *IEEE Access*, vol. 6, no. 1, pp. 70172–70185, 2018.
- [24] N. B. Truong, G. M. Lee, A. M. Crépeau, and R. Cedex, "Software Defined Networking-based Vehicular Adhoc Network with Fog Computing," in *IFIP/IEEE International Symposium on Integrated Network Management (IM)*, 2015, pp. 1202–1207.
- [25] X. Hou, Y. Li, M. Chen, D. Wu, D. Jin, and S. Chen, "Vehicular Fog Computing: A Viewpoint of Vehicles as the Infrastructures," *IEEE Trans. Veh. Technol.*, vol. 65, no. 6, pp. 3860–3873, 2016.
- [26] Y. Xiao and C. Zhu, "Vehicular fog computing: Vision and challenges," in *2017 IEEE International Conference on Pervasive Computing and Communications Workshops, PerCom Workshops 2017*, 2017, pp. 6–9.
- [27] M. Bagaia, A. Ksentini, T. Taleb, R. Jantti, A. Chelli, and I. Balasingham, "An efficient D2D-based strategies for machine type communications in 5G mobile systems," *IEEE Wirel. Commun. Netw. Conf. WCNC*, vol. 2016-Septe, no. Wcnc, 2016.
- [28] S. Raza, S. Wang, M. Ahmed, and M. R. Anwar, "A Survey on Vehicular Edge Computing : Architecture , Applications , Technical Issues , and Future Directions," *Wirel. Commun. Mob. Comput.*, vol. Article ID, pp. 1–19, 2019.
- [29] Q. He, J. Liu, C. Wang, and B. Li, "Coping With Heterogeneous Video Contributors and Viewers in Crowdsourced Live Streaming: A Cloud-Based Approach," *IEEE Trans. Multimed.*, vol. 18, no. 5, pp. 916–928, 2016.
- [30] T. Mekki, I. Jabri, A. Rachedi, and M. ben Jemaa, "Vehicular cloud networks: Challenges, architectures, and future directions," *Veh. Commun.*, vol. 9, pp. 268–280, 2017.

- [31] K. Zheng, H. Meng, P. Chatzimisios, L. Lei, and X. Shen, "An SMDP-Based Resource Allocation in Vehicular Cloud Computing Systems," *IEEE Trans. Ind. Electron.*, vol. 62, no. 12, pp. 7920–7928, 2015.
- [32] C. C. Chan, "The state of the art of electric, hybrid, and fuel cell vehicles," *Proc. IEEE*, vol. 95, no. 4, pp. 704–718, 2007.
- [33] S. F. Tie and C. W. Tan, "A review of energy sources and energy management system in electric vehicles," *Renew. Sustain. Energy Rev.*, vol. 20, pp. 82–102, 2013.
- [34] M. Peng, L. Liu, and C. Jiang, "A review on the economic dispatch and risk management of the large-scale plug-in electric vehicles (PHEVs)-penetrated power systems," *Renew. Sustain. Energy Rev.*, vol. 16, no. 3, pp. 1508–1515, 2012.
- [35] D. T. Hoang, P. Wang, D. Niyato, and E. Hossain, "Charging and discharging of plug-in electric vehicles (PEVs) in vehicle-to-grid (V2G) systems: A cyber insurance-based model," *IEEE Access*, vol. 5, pp. 732–754, 2017.
- [36] R. G. Gago, S. F. Pinto, and J. F. Silva, "G2V and V2G electric vehicle charger for smart grids," in *IEEE 2nd International Smart Cities Conference: Improving the Citizens Quality of Life, ISC2 2016 - Proceedings*, 2016, pp. 1–6.
- [37] K. V. Singh, H. O. Bansal, and D. Singh, "A comprehensive review on hybrid electric vehicles: architectures and components," *J. Mod. Transp.*, vol. 27, no. 2, pp. 77–107, 2019.
- [38] H. S. Das, C. W. Tan, and A. H. M. Yatim, "Fuel cell hybrid electric vehicles: A review on power conditioning units and topologies," *Renew. Sustain. Energy Rev.*, vol. 76, no. February, pp. 268–291, 2017.
- [39] A. Babu and S. Ashok, "Algorithm for selection of motor and vehicle architecture for a plug-in hybrid electric vehicle," in *2012 Annual IEEE India Conference, INDICON 2012*, 2012, no. 1, pp. 875–878.
- [40] B. G. Pollet, I. Staffell, and J. L. Shang, "Current status of hybrid, battery and fuel cell electric vehicles: From electrochemistry to market prospects," *Electrochim. Acta*, vol. 84, pp. 235–249, 2012.
- [41] C. C. Chan, A. Bouscayrol, and K. Chen, "Electric, hybrid, and fuel-cell vehicles: Architectures and modeling," *IEEE Trans. Veh. Technol.*, vol. 59, no. 2, pp. 589–598, 2010.
- [42] K. Chen, A. Bouscayrol, A. Berthon, P. Delarue, D. Hissel, and R. Trigui, "Global Modeling of Different Vehicles," *IEEE Veh. Technol. Mag.*, no. June, pp. 80–89, 2009.
- [43] I. E. A. International and E. Agency, "Global EV Outlook 2016 Beyond one million electric cars." . <https://renewablesnow.com/news/ts-of-stationary-batteries-to-fall-by-up-to-66-by-2030-irena-586230/>, 2016.
- [44] International Energy Agency, "Global EV Outlook 2018. Towards cross-model electrification," *Electric Vehicles Initiative*. p. 143, 2018.
- [45] C. Ghribi, M. Hadji, and D. Zeghlache, "Energy efficient VM scheduling for cloud data centers: Exact allocation and migration algorithms," in *Proceedings - 13th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing, CCGrid 2013*, 2013, pp. 671–678.

- [46] N. Hamdy, "Resource Allocation Strategies in Cloud Computing : Overview," *Int. J. Comput. Appl.*, vol. 177, no. 4, pp. 18–22, 2017.
- [47] V. Vinothina, S. Lecturer, and R. Sridaran, "A Survey on Resource Allocation Strategies in Cloud Computing," vol. 3, no. 6, pp. 97–104, 2012.
- [48] B. P. Rimal, E. Choi, and I. Lumb, "A taxonomy and survey of cloud computing systems," in *NCM 2009 - 5th International Joint Conference on INC, IMS, and IDC*, 2009, pp. 44–51.
- [49] M. M. Hassan and A. Alamri, "Virtual Machine resource allocation for multimedia cloud: A Nash bargaining approach," in *Procedia Computer Science*, 2014, vol. 34, pp. 571–576.
- [50] L. Tang, "Dynamic migration optimization algorithm for virtual machine under trusted computing environment," *Proceedings - 2018 International Conference on Virtual Reality and Intelligent Systems, ICVRIS 2018*. IEEE, pp. 48–52, 2018.
- [51] S. Urathkal, M. Angalore, and I. Ndia, "A Novel Bio-Inspired Load Balancing of Virtual Machines in Cloud Environment," in *2014 IEEE International Conference on Cloud Computing in Emerging Markets (CCEM)*, 2014, pp. 1–4.
- [52] A. Goyal and N. S. Chahal, "Bio inspired approach for load balancing to reduce energy consumption in cloud data center," in *International Conference Communication, Control and Intelligent Systems, CCIS 2015*, 2016, pp. 406–410.
- [53] X. Liu, C. Wang, B. B. Zhou, J. Chen, T. Yang, and A. Y. Zomaya, "Priority-based Consolidation of Parallel Workloads in the Cloud," vol. 24, no. 9, pp. 1874–1883, 2013.
- [54] S. J. Patel and U. R. Bhoi, "Improved priority based job scheduling algorithm in cloud computing using iterative method," in *Proceedings - 2014 4th International Conference on Advances in Computing and Communications, ICACC 2014*, 2014, pp. 199–202.
- [55] L. Wu, S. K. Garg, and R. Buyya, "SLA-based resource allocation for software as a service provider (SaaS) in cloud computing environments," *Proc. - 11th IEEE/ACM Int. Symp. Clust. Cloud Grid Comput. CCGrid 2011*, pp. 195–204, 2011.
- [56] M. Jebalia, A. Ben Letaïfa, M. Hamdi, and S. Tabbane, "A Fair Resource Allocation Approach in Cloud Computing Environments," 2018, pp. 54–57.
- [57] A. Nezarat and G. Dastghaibifard, "A Game Theoretic Method for Resource Allocation in Scientific Cloud," *Int. J. Cloud Appl. Comput.*, vol. 6, no. 1, pp. 15–41, 2016.
- [58] D. A. Alboaneen, H. Tianfield, and Y. Zhang, "Metaheuristic approaches to virtual machine placement in cloud computing: A review," in *Proceedings - 15th International Symposium on Parallel and Distributed Computing, ISPDC 2016*, 2016, pp. 214–221.
- [59] T. Wang, Y. Liang, W. Jia, M. Arif, A. Liu, and M. Xie, "Journal of Network and Computer Applications Coupling resource management based on fog computing in smart city systems," *J. Netw. Comput. Appl.*, vol. 135, no. March, pp. 11–19, 2019.
- [60] M. Mukherjee, L. Shu, and D. Wang, "Survey of fog computing: Fundamental, network applications, and research challenges," *IEEE Commun. Surv. Tutorials*, vol. 20, no. 3, pp. 1826–1857, 2018.
- [61] R. K. Naha *et al.*, "Fog computing: Survey of trends, architectures, requirements, and

- research directions,” *IEEE Access*, vol. 6, pp. 47980–48009, 2018.
- [62] S. Filiposka, A. Mishev, and K. Gilly, “Community-based allocation and migration strategies for fog computing,” in *IEEE Wireless Communications and Networking Conference, WCNC*, 2018, vol. 2018-April, pp. 1–6.
- [63] X. Hou, Y. Li, M. Chen, D. Wu, D. Jin, and S. Chen, “Vehicular Fog Computing: A Viewpoint of Vehicles as the Infrastructures,” *IEEE Trans. Veh. Technol.*, vol. 65, no. 6, pp. 3860–3873, 2016.
- [64] J. Klaimi and S. Senouci, “Theoretical Game-based Resource Management For Mobile Users in a Vehicular Fog Computing Environment,” *2018 14th Int. Wirel. Commun. Mob. Comput. Conf.*, pp. 452–457, 1859.
- [65] Z. Wang, Z. Zhong, and M. Ni, “Application-Aware Offloading Policy Using SMDP in Vehicular Fog Computing Systems,” in *2018 IEEE International Conference on Communications Workshops (ICC Workshops)*, 2018, pp. 1–6.
- [66] Z. Zhou, P. Liu, J. Feng, Y. Zhang, S. Mumtaz, and J. Rodriguez, “Computation Resource Allocation and Task Assignment Optimization in Vehicular Fog Computing: A Contract-Matching Approach,” *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3113–3125, 2019.
- [67] S. Deshmukh and R. Shah, “Computation offloading frameworks in mobile cloud computing : A survey,” in *2016 IEEE International Conference on Current Trends in Advanced Computing, ICCTAC 2016*, 2016.
- [68] S. Guo, B. Xiao, Y. Yang, and Y. Yang, “Energy-efficient dynamic offloading and resource scheduling in mobile cloud computing,” in *Proceedings - IEEE INFOCOM*, 2016, vol. 2016-July.
- [69] G. Husain and R. Sridaran, “A survey of Game Based Strategis of resource Allocation in Cloud Computing,” in *International Conference on Computing for Sustainable Global Development*, 2016, pp. 3561–3566.
- [70] D. T. Hoang, X. Lu, D. Niyato, P. Wang, D. I. Kim, and Z. Han, “Applications of Repeated Games in Wireless Networks: A Survey,” *IEEE Commun. Surv. Tutorials*, vol. 17, no. 4, pp. 2102–2135, 2015.
- [71] E. Hyytia, “Offload (Only) the Right Jobs : Robust Offloading U sing the Markov Decision Processes,” in *proc. of IEEE WOWMoM*, 2015, pp. 1–9.
- [72] H. Zannat and S. Hossain, “A Hybrid Framework using Markov Decision Process for Mobile Code Offloading,” in *2016 19th International Conference on Computer and Information Technology (ICCIT)*, 2016, pp. 31–35.
- [73] I. E. A. International and E. Agency, “Global EV Outlook 2018.” <https://www.iea.org/gevo2018/>, 2018.
- [74] C. Forgez, K. El Kadri Benkara, G. Friedrich, and D. V. Do, “Impedance Observer for a Li-Ion Battery Using Kalman Filter,” *IEEE Trans. Veh. Technol.*, vol. 58, no. 8, pp. 3930–3937, 2009.
- [75] M. Coleman, W. G. Hurley, and C. K. Lee, “An improved battery characterization method using a two-pulse load test,” *IEEE Trans. Energy Convers.*, vol. 23, no. 2, pp. 708–713, 2008.

- [76] R. A. Hanifah, S. F. Toha, and S. Ahmad, "Electric Vehicle Battery Modelling and Performance Comparison in Relation to Range Anxiety," *Procedia Comput. Sci.*, vol. 76, no. Iris, pp. 250–256, 2015.
- [77] S. Amjad, S. Neelakrishnan, and R. Rudramoorthy, "Review of design considerations and technological challenges for successful development and deployment of plug-in hybrid electric vehicles," *Renew. Sustain. Energy Rev.*, vol. 14, no. 3, pp. 1104–1110, 2010.
- [78] M. A. Hannan, M. M. Hoque, A. Hussain, Y. Yusof, and P. J. Ker, "State-of-the-Art and Energy Management System of Lithium-Ion Batteries in Electric Vehicle Applications: Issues and Recommendations," *IEEE Access*, vol. 6, no. c, pp. 19362–19378, 2018.
- [79] L. Zheng, L. Zhang, J. Zhu, G. Wang, and J. Jiang, "Co-estimation of state-of-charge, capacity and resistance for lithium-ion batteries based on a high-fidelity electrochemical model," *Appl. Energy*, vol. 180, pp. 424–434, 2016.
- [80] J. Baker, "New technology and possible advances in energy storage," *Energy Policy*, vol. 36, no. 12, pp. 4368–4373, 2008.
- [81] A. Mahmoudzadeh Andwari, A. Pesiridis, S. Rajoo, R. Martinez-Botas, and V. Esfahanian, "A review of Battery Electric Vehicle technology and readiness levels," *Renew. Sustain. Energy Rev.*, vol. 78, no. February, pp. 414–430, 2017.
- [82] J. Dixon, I. Nakashima, E. F. Arcos, and M. Ortúzar, "Electric vehicle using a combination of ultracapacitors and ZEBRA battery," *IEEE Trans. Ind. Electron.*, vol. 57, no. 3, pp. 943–949, 2010.
- [83] L. Guzzella and A. Sciarretta, *Vehicle propulsion systems: Introduction to modeling and optimization*, Third Edit., vol. 9783642359. Springer, New York, 2013.
- [84] Z. Chen, M. Sun, X. Shu, R. Xiao, and J. Shen, "Online State of Health Estimation for Lithium-Ion Batteries Based on Support Vector Machine," *Appl. Sci.*, vol. 8, no. 6, p. 925, 2018.
- [85] A. Fotouhi, D. J. Auger, K. Propp, S. Longo, and M. Wild, "A review on electric vehicle battery modelling: From Lithium-ion toward Lithium-Sulphur," *Renew. Sustain. Energy Rev.*, vol. 56, pp. 1008–1021, 2016.
- [86] F. Saidani, F. X. Hutter, R. G. Scurtu, W. Braunwarth, and J. N. Burghartz, "Lithium-ion battery models: A comparative study and a model-based powerline communication," *Adv. Radio Sci.*, vol. 15, pp. 83–91, 2017.
- [87] A. F. Burke, "Batteries and ultracapacitors for electric, hybrid, and fuel cell vehicles," *Proc. IEEE*, vol. 95, no. 4, pp. 806–820, 2007.
- [88] M. Armand, "Issues and challenges facing rechargeable lithium batteries," vol. 414, no. November, pp. 359–367, 2001.
- [89] P. Van den Bossche, F. Vergels, J. Van Mierlo, J. Matheys, and W. Van Autenboer, "SUBAT: An assessment of sustainable battery technology," *J. Power Sources*, vol. 162, no. 2 SPEC. ISS., pp. 913–919, 2006.
- [90] Statista, "Battery Share of Large EV Costs 2030." <https://www.statista.com/statistics/797638/battery-share-of-large-electric-vehicle-cost>, 2018.

- [91] International Renewable Energy Agency (IRENA), “Costs of Stationary Batteries to Fall by up to 66% by 2030.” <https://renewablesnow.com/news/ts-of-stationary-batteries-to-fall-by-up-to-66-by-2030-irena-586230>, 2018.
- [92] M. Chen and G. A. Rincon-Mora, “Accurate Electrical Battery Model Capable of Predicting Runtime and I–V Performance,” *IEEE Trans. ENERGY Convers.*, vol. 21, no. 2, pp. 504–511, 2006.
- [93] S. Bhide and T. Shim, “Novel predictive electric Li-ion battery model incorporating thermal and rate factor effects,” *IEEE Trans. Veh. Technol.*, vol. 60, no. 3, pp. 819–829, 2011.
- [94] K. Liu, K. Li, Q. Peng, and C. Zhang, “A brief review on key technologies in the battery management system of electric vehicles,” *Frontiers of Mechanical Engineering*, vol. 14, no. 1, pp. 47–64.
- [95] C. Zhang, K. Li, S. McLoone, and Z. Yang, “Battery modelling methods for electric vehicles - A review,” in *2014 European Control Conference, ECC 2014*, 2014, pp. 2673–2678.
- [96] C. C. Chan, E. W. C. Lo, and S. Weixiang, “Available capacity computation model based on artificial neural network for lead-acid batteries in electric vehicles,” *J. Power Sources*, vol. 87, no. 1, pp. 201–204, 2000.
- [97] Y. Song and L. Gao, “Incremental battery model using wavelet-based neural networks,” *IEEE Trans. Components, Packag. Manuf. Technol.*, vol. 1, no. 7, pp. 1075–1081, 2011.
- [98] H. Zhang, Q. Zhang, and X. Du, “Toward vehicle-assisted cloud computing for smartphones,” *IEEE Trans. Veh. Technol.*, vol. 64, no. 12, pp. 5610–5618, 2015.
- [99] C. W. Paper, “Fog computing and the internet of things: Extend the cloud to where the things are,” *2015 [online]*. Available: https://www.cisco.com/c/dam/en_us/solutions/trends/iot/.../computing-overview.pdf.
- [100] E. A. Inria and S. C. France, “Constrained Markov Decision Processes,” London, U.K.: ChapmanHall/CRC, 1999.
- [101] D. Dolgov and E. Durfee, “Resource Allocation Among Agents with MDPs-Induced Preferences,” *J. Artif. Intell.*, no. 27, pp. 505–549, 2006.
- [102] R. Bellman, “Theory of Dynamic Programming.” Princeton University Press, 1957.
- [103] M. J. Cros, “Markov decision processes toolbox: available: <https://www.mathworks.com/matlabcentral/fileexchange/25786-markov-decision-processes-mdptoolbox> [online],” 2016.
- [104] M. Abu Alsheikh, D. T. Hoang, D. Niyato, H. P. Tan, and S. Lin, “Markov decision processes with applications in wireless sensor networks: A survey,” *IEEE Commun. Surv. Tutorials*, vol. 17, no. 3, pp. 1239–1267, 2015.
- [105] X. Chen, “Decentralized computation offloading game for mobile cloud computing,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 4, pp. 974–983, 2015.
- [106] J. Zheng, Y. Cai, Y. Wu, and X. S. Shen, “Dynamic Computation Offloading for Mobile Cloud Computing: A Stochastic Game-Theoretic Approach,” *IEEE Trans. Mob. Comput.*, vol. PP, no. c, p. 1, 2018.

- [107] S. Josilo and G. Dan, "Selfish Decentralized Computation Offloading for Mobile Cloud Computing in Dense Wireless Networks," *IEEE Trans. Mob. Comput.*, vol. 18, no. 1, pp. 207–220, 2019.
- [108] J. Zheng, Y. Wu, N. Zhang, H. Zhou, Y. Cai, and X. Shen, "Optimal Power Control in Ultra-Dense Small Cell Networks: A Game-Theoretic Approach," *IEEE Trans. Wirel. Commun.*, vol. 16, no. 7, pp. 4139–4150, 2017.
- [109] E. Solan, "Stochastic Games," New York; NY: Springer, New York, 2012.

LIST OF FIGURES

Figure 1.1 - Fog Computing Framework [3].	2
Figure 1.2 - Vehicular Fog Computing Framework.	3
Figure 2.1 - Cloud: Characteristics, Service and Deployment Models.	9
Figure 2.2 - The Basic Edge Computing Architecture [2].	14
Figure 2.3 - Fog Computing Model [19].	15
Figure 2.4 - Conceptual Scenario of Vehicular Fog Computing [26].	17
Figure 2.5 - Classification of Vehicles [32].	21
Figure 2.6 – Components of HEV	23
Figure 2.7 - Architecture of Series HEV.	24
Figure 2.8 - Architecture of Parallel HEV.	25
Figure 2.9 - Architecture of Combined HEV.	26
Figure 2.10 - Architecture of Complex HEV.	26
Figure 2.11 - Battery Electric Vehicle System.	27
Figure 2.12 - (a) Passenger Electric Car Stock in Major Regions and (b) Deployment Scenarios for the Stock of Electric Cars to 2030 [20].	30
Figure 3.1 - Resource Allocation in Cloud Computing.	37
Figure 3.2 - Resource Allocation Strategies in Cloud Computing.	38
Figure 3.3 - Resource Allocation in Fog Computing.	42
Figure 4.1 - Volumetric Energy Density vs. Gravimetric energy of Common Battery Technologies [90].	55
Figure 4.2 - Ragon Plot [91].	56
Figure 4.3 - Price Trend of a Lithium-ion Battery from 2010 to 2016 [92].	57
Figure 4.4 - EV Battery Pack Detail.	60
Figure 4.5 - Lithium ion Battery Simulink Model.	61
Figure 4.6 - Flow Chart to Serve Users Demand in EVs.	63
Figure 4.7 - Changes in arrival rate of users demands over time.	64
Figure 4.8 - (a) EVs arrival and (b) Selected EVs with higher SOH Level.	65
Figure 4.9 - SOC evolution of consumption over time for (a) ZOE and (b) BMW models.	66
Figure 4.10 - Served Users Demands in the two EV models.	67
Figure 5.1 - Vehicular Fog Computing (VFC) Architecture.	72
Figure 5.2 - Example of MDP System with Four States and Three Actions.	74
Figure 5.3 – Evolution of Markov Decision Process System.	75
Figure 5.4 - EVs Arrival and Departure in Parking lot.	78
Figure 5.5 - Mobile Users' Resource Allocation Algorithm.	80
Figure 5.6 - Markov Chain Model with five states for EVs arrival.	82
Figure 5.7 - Different simulation scenarios for the changes in occupancy rates over a 24 hours period.	83
Figure 5.8 - Overall Performance Evaluation for the Different Simulation Scenarios.	84
Figure 5.9 - Reward, Global Reward and Energy Gain for case 1.	85

Figure 5.10 - Reward, Global Reward and Energy Gain for case 2.	86
Figure 5.11 - Reward, Global Reward and Energy Gain for case 3.	87
Figure 5.12 - Reward, Global Reward and Energy Gain for case 4.	88
Figure 5.13 - Reward, Global Reward and Energy Gain for case 5.	89
Figure 5.14 - Reward, Global Reward and Energy Gain for case 6.	90
Figure 6.1 - Vehicular Fog Computing Architecture.....	95
Figure 6.2 - Strategy selection probability when $b = 0.5$	106
Figure 6.3 - Strategy selection probability when $b = 0.1$	106
Figure 6.4 - Average system wide cost when $b = 0.1$	107
Figure 6.5 - Average system wide cost when $b = 0.5$	108
Figure 6.6 - System Wide Cost.....	108
Figure 6.7 - VFC Average system wide cost when $b=0.5$	109
Figure 6.8 - Cloud Average system wide cost when $b=0.1$	109

LIST OF TABLES

Table 2.1 - Cloud Consumer and Cloud Provider activities on Cloud Service Model.....	11
Table 2.2 - Comparison Between Cloud, MEC, Fog and VFC.	19
Table 2.3 - Battery Capacity and Technologies.....	22
Table 2.4 - Characteristics of BEVs, HEVs, and FCVs [31].	32
Table 2.5 - EV model, Type, and Battery Capacity of PEV [34].	33
Table 4.1 - Comparison of Lithium-ion Battery Technologies [81].	54
Table 4.2 - Main Simulation Parameters.	64
Table 5.1 - Summary of Simulation Scenarios.	82
Table 6.1 - Main Simulation Parameters	105

Titre : Allocation des ressources dans l'informatique en brouillard véhiculaire pour une utilisation optimale des véhicules électriques

Mots clés : Informatique en brouillard véhiculaire, allocation de ressources, processus de décision Markovien, véhicule électrique

Résumé : Les progrès technologiques ont permis aux véhicules électriques d'avoir des capacités à la fois de calcul, de communication, de stockage et de perception. Néanmoins, la plupart du temps, ces véhicules électriques sont en stationnement, ce qui engendre une sous-utilisation de leurs capacités embarquées. Ainsi, une meilleure gestion et une mise en commun de ces ressources sous-utilisées deviennent fortement recommandées. Les ressources agrégées seraient utiles pour des applications de sécurité routière, des applications liées au confort ou pourraient même être utilisées en tant que centre de calcul distribué. En outre, les véhicules en stationnement pourraient également être utilisés comme plate-forme de fourniture de services. Par conséquent, l'utilisation de ressources abondantes agrégées pour le déploiement de différentes applications mobiles locales a conduit au développement du concept d'informatique en brouillard véhiculaire (en anglais, *Vehicular Fog Computing* - VFC). Grâce à ce dernier, les véhicules dans les aires de stationnement, les centres commerciaux ou les aéroports vont agir en tant que nœuds *fog*. Dans un autre contexte, les applications mobiles sont devenues de plus en plus populaires, complexes et gourmandes en ressources. Certaines applications mobiles nécessitent des capacités de calcul intensives et une consommation d'énergie élevée qui transcendent les capacités limitées des appareils mobiles. Tout au long de ce travail, nous abordons les verrous liés au déploiement d'un système VFC agrégeant les

ressources inutilisées des véhicules électriques en stationnement pour être utilisées comme nœuds *fogs* répondants aux demandes de calcul des utilisateurs mobiles à proximité. Notre travail commence par un état de l'art sur les véhicules électriques et l'allocation de ressources dans le système VFC. En outre, nous évaluons le potentiel des ressources agrégées dans les véhicules électriques pour répondre aux demandes d'applications d'utilisateurs mobiles locaux en prenant en compte l'état de santé de la batterie (en anglais, *State of Health* - SOH) et son état de charge (en anglais, *State of Charge* - SOC). Notre objectif est de choisir des VEs ayant un état de santé et de charge satisfaisants pour faire partie du VFC tout en permettant aux propriétaires de ces véhicules de disposer d'une quantité d'énergie suffisante pour leur mobilité. Nous abordons, par la suite, le problème d'allocation de ressources avec une nouvelle solution basée sur le processus de décision Markovien (en anglais, *Markov Decision Process* - MDP) qui vise à optimiser l'utilisation de l'énergie des véhicules électriques pour répondre à la fois à aux demandes de calcul et de mobilité des utilisateurs. Enfin, nous proposons une approche basée sur un jeu stochastique pour montrer la dynamique de la demande de calcul des utilisateurs mobiles et la disponibilité des ressources des véhicules électriques. Nous avons donc pu montrer dans cette thèse les avantages de l'utilisation des ressources sous-utilisées des véhicules électriques afin de répondre aux demandes de calcul des applications d'utilisateurs mobiles.

Title: Resource Allocation in Vehicular Fog Computing for an Optimal Use of Electric Vehicles Energy

Keywords: Vehicular Fog Computing, Resource Allocation, Stochastic Game, Markov Decision Process, Electric Vehicles.

Abstract: Technological advancements made it possible for Electric vehicles (EVs) to have onboard computation, communication, storage, and sensing capabilities. Nevertheless, most of the time these EVs spend their time in parking lots, which makes onboard devices cruelly underutilized. Thus, a better management and pooling these underutilized resources together would be strongly recommended. The new aggregated resources would be useful for traffic safety applications, comfort related applications or can be used as a distributed data center. Moreover, parked vehicles might also be used as a service delivery platform to serve users. Therefore, the use of aggregated abundant resources for the deployment of different local mobile applications leads to the development of a new architecture called Vehicular Fog Computing (VFC). Through VFC, abundant resources of vehicles in the parking area, on the mall or in the airport, can act as fog nodes. In another context, mobile applications have become more popular, complex and resource intensive. Some sophisticated embedded applications require intensive computation capabilities and high- energy consumption that transcend the limited capabilities of mobile devices. Throughout this work, we

tackle the problem of achieving an effective deployment of a VFC system by aggregating unused resources of parked EVs, which would be eventually used as fog nodes to serve nearby mobile users' computation demands. At first, we present a state of the art on EVs and resource allocation in VFC. In addition, we assess the potential of aggregated resources in EVs for serving local mobile users' applications demands by considering the battery State of Health (SOH) and State of Charge (SOC). Here, the objective is to choose EVs with a good condition of SOH and SOC so that owners secure tolerable amount of energy for mobility. Then, we address the problem of resource allocation scheme with a new solution based on Markov Decision Process (MDP) that aims to optimize the use of EVs energy for both computing users' demands and mobility. Hence, the novelty of this contribution is to take into consideration the amount of aggregated EVs resource for serving users' demands. Finally, we propose a stochastic theoretical game approach to show the dynamics of both mobile users' computation demands and the availability of EVs resources. We have therefore been able to show in this thesis the benefits of using the unused EVs resources for serving users' mobile applications demands.