MOBILE CLOUD COMPUTING FOR REDUCING POWER CONSUMPTION AND MINIMISING LATENCY

RAKAN AMEEN ALDMOUR

A Thesis in partial fulfillment of the requirements of Anglia Ruskin University for the degree of a Doctor of Philosophy

Submitted: January 2018

I would like to dedicate this research to

My wonderful family

&

My dear late friend

Thaer Wajeeh Alsuhaimat

who passed away on 02/01/2018.

May his soul rest in peace.

Ameen

'To live in the hearts of others is never to die'

Acknowledgments

I would like to send my highest gratitude to my supervisor Dr. Sufian Yousef for sharing his valuable expertise, encouragement, guidance and for his continuous support throughout the process of this research.

I am eternally grateful for the support and care of my parents and siblings, who have always been an endless source of strength, guidance, happiness and have inspired me to obtain my goals.

I would like to send special thanks to my valuable friend Dr. Bashar for his friendship, encouragement, motivation, enthusiasm and warmth. He had a great impact in my life and it has been a privilege to have his support throughout this part of the journey.

I offer sincere thanks to my friends; Dr Ahmed Sarhan, Mohammad Alradaideh, Hamad Aljazi, Khaldoun Almajali, Faris Al Badani, Syed Nooh, Mohammed Fakrudeen, Mohammad Wafeq , Ghaith Abu Alfalayeh, Doaa Al Jamal and Louise Newley, amongst others who have offered support, assistance and care throughout this process.

Lastly, I am thankful to the academic and administrative staff at Anglia Ruskin University for the guidance and support in my academic journey.

ANGLIA RUSKIN UNIVERSITY

ABSTRACT

FACULTY OF SCIENCE AND TECHNOLOGY
DOCTOR OF PHILOSOPHY

MOBILE CLOUD COMPUTING FOR REDUCING POWER CONSUMPTION AND MINIMISING LATENCY

RAKAN AMEEN ALDMOUR

JANUARY 2018

Mobile phones can be used for basic computations, whilst in contrast smart phones are used for intensive computations, videos, and social media. Processors have become faster, sensors are multi-functional, whereas battery lifetime has had minimal improvement. In view of rich media which involves huge computational complexity, there are greater challenges for mobile users in terms of battery, storage space and transmission time. Therefore, it is vital to send the intensive tasks from mobile to cloud with minimal delay. In this research, a strategy to either save a file on the mobile or offload it to the cloud has been designed through a comparative analysis of two main factors; offloading delay and the power consumption.

An efficient model of offloading process to estimate the battery consumption and the delay was established. Furthermore, a decision engine was designed to choose the most suitable mobile transmission protocol through a comparative study of each route. Additionally, a new model was proposed to compress the file during the 4G offloading technique, which decreased the file size resulting in less delay, enhanced quality of service and secured data integrity. Further innovations were the use of a secondary server when the first was busy and a back up server to prevent repetitive uploading of files.

The results have shown that the newly developed MECCA (Mobile Energy Cloud Computing Application) decreased the processing time by 30% while offloading the files. The use of the secondary server illustrated that the new features saved the mobile battery and reduced the processing time. Moreover, in Second Upload Round (SUR), the improvements in power usage reduction was noticeable whilst using the 4G connection. Overall, the results offer an insight into the offload possibility for different characteristics. The MECCA application was validated through comparative analysis of the main parameters to the developed analytical model which illustrated a match of 96% between the results.

Key Words: Cloud Computing, MECCA, Offloading, Battery Power, File Size and Mobile Protocols

Table of Contents

List of Figures

List of Tables

ABBREVIATION

CC: Cloud Computing

MCC: Mobile Cloud Computing

IT: Information Technology

GPS: Global Position System

GC: Google Cloud

OS: Operating System

GUI: Graphical User Interface

WLAN: Wireless Local Area Network

WPAN: Wireless Personal Area Network

LAN: Local Area Network

WNI: Wireless Network Interface

1G: First Generation

2G: Second Generation

3G: Third Generation

4G: Fourth Generation

5G: Fifth Generation

SMS: Short Message Service

EDGE: Enhanced Data to Global Evolution

GPRS: General Packet Radio Service

UMTS: Universal Mobile Telecommunications System

HSDPA: High-speed Downlink Packet Access

ABBREVIATION

ICT: Information and Communication Technology

IP: Internet Protocol

RAM: Random Access Memory

CPU: Central Process Unit

HDD: Hard Disk Drive

SaaS: Software as a Service

PaaS: Platform as a Service

IaaS: Infrastructure as a Service

EaaS: Everything as a Service

GSM: Global System for Mobile

AP: Access Point

MAUI: Mobile Assistance Using Infrastructure

LTE: Long Term Evolution

MACS: Mobile Augmentation Cloud Services

MCM: Mobile Cloud Middleware

VM: Virtual Machine

DE: Decision Engine

QoS: Quality of Service

IDE: Integrated Development Environment

CSV: Comma Separated Value

B: Bytes

MB: Megabytes

GB: Gigabytes

ABBREVIATION

TB: Terabytes

NI: Network Interface

SDC: Secure Digital Card

MECCA: Mobile Energy Cloud Application

DSL: Digital Subscriber Line

ADB: Android Debug Bridge

CMD: Command Prompt Command

USB: Universal Serial Bus

FUR: First Uploading Round

SUR: Second Uploading Round

MAh: Milliampere hour

mW: Milliwatts

CQI: Channel Quality Indicator

MCS: Modulation and Coding Scheme

TTI: Transmission Time Interval

SNR: Signal to noise ratio

SINR: Signal to Interference and Noise Ratio

SNDR: Signal to Noise and Distortion Ratio

TCP: Transfer Control Protocol

UE: User Equipment

RRC: Radio Resource Control

DCH: Dedicated Channel

FACH: Forward Access Channel

ABBREVIATION

AP: Access Point

QR: Quick Response

UMTS: Universal Mobile Telecommunications System

UTRAN: Universal Terrestrial Radio Access Network

CN: Core network

RNC: Radio Network Controller

FUR: First Upload Round

SUR: Second Upload Round

List of Appendices

Chapter 1 Introduction

1.1 Introduction

Mobile smartphones are becoming powerful and easy to use. In fact, they are taking the place instead of normal desktop computers for both work use and personal use. Smartphones hardware technology has improved to be able to perform intensive computing tasks rather than basic computation such as image processing, face recognition, heavy calculation, and gaming. These smartphone applications are becoming energy hungry, which means they use more power consumption due to the continuous use (Atre, et al., 2016).

The mobile battery is one of the biggest issues which impacts the development of smartphones due to several reasons:

- Smartphones are consuming energy much more than what the batteries capacity are able to provide, this is due to processors becoming more powerful and faster, screens are getting multi-functions and more colourful, and devices are provided with more sensors. On the other hand, there are only slight improvements in the mobile battery functions; the current battery lifetime, and the future trends are not predictable (Lauridsen, et al., 2013).
- Mobiles were only used for basic computations, such as messages and voice. Yet, smartphones are used for intensive computations, videos, and social media. As a result, more power consumption has shortened the battery life. The mobile battery is consumed quickly because of the heavy computations and communications (Moghadasi, et al., 2015).

Currently, the number of smart devices such as smartphones and tablets are growing noticeably, compared to the computing desktop. Smartphones are small computing devices that have been equipped with features such as camera, user interface, sensors, powerful processors, and device probabilities. These features enabled many beneficial applications which were unexpected earlier. Additionally, it is needed to make them more attractive to the customers. The Internet facilitates the connection with users in different ways such as emails and social networking. This helps users to handle work and share their pictures and video between one another. Nowadays, it is obvious smartphones are playing a crucial role in social networking. As a result, these smartphones are needed to manage and perform different kinds of functionalities, and most of them are known as heavy computing tasks.

Mobile devices have unique limitations such as:

- **Processing capabilities:** Some smartphones are unable to handle heavy tasks such as face recognition and gaming.
- **Limited battery energy:** although there were improvements in the smartphone circuits, the battery is still one of the big issues.
- **Memory capacity:** rich media needs a massive storage and smartphones require big storage to cope with massive data.

In the last seven years, these limitations have become prevalent among mobile clients, while the mobiles are becoming most popular because of their facilities and functionalities. Due to the powerful operating systems such as Apple iOS, Android, and Windows Mobile, smartphones are capable to perform advanced applications which are almost to the same standard as desktop computer applications. Every mobile device application executes some tasks, of which every task performs a particular computation on a given data (Altamimi, et al., 2015).

Mobile marketplaces recently are overloaded by offering a variety of popular applications. These kinds of applications increase the amounts of computation, communication, and memory resources of mobile devices. These include; gaming applications, photo, video editing and mobile client applications because they require heavy processing resources. Yet, mobile applications in these areas remain clearly inferior to desktops, in terms of performance and functionality.

1.2 Cloud-Assisted and Computation Offloading for Smartphones

Cloud-assisted processing promises to overcome applications limitations which are running on constrained resources devices, such as smartphones by letting them offload processing to more powerful remote devices. This idea of using the pool of powerful remote resources to extend smartphone devices capabilities was introduced in the early 2000s (Balan, et al., 2002). Moreover, different kinds of offloading systems have been introduced, such as; MAUI is proposed by Eduardo, et al., (2010), ThinkAir is proposed by Kosta, et al., (2012), and CloneCloud is presented by (Chun, et al., 2011). Systems such as these reduce application processing time and power consumption on the mobile devices in different ways such as; partitioning applications among local and remote device, and by processing intensive tasks on the remote side.

The limitations of smartphone devices as mentioned in several studies such as, Qi, et al., (2016), Altamimi, et al., (2015), Atre, et al., (2016), Jararweh, et al., (2014) and (Gao, et al., (2016), which are met by Cloud Computing (CC). The rapid evolution of cloud computing and data centres offer new opportunities for covering these limitations. New applications and hardware resources are presented as services through the Internet by several suppliers. The cost-effective and massive scalable resources are accessed on demand, to achieve higher reliability, efficiency, and the most important issue is to use unlimited hardware resources. These resources help to remove infrastructure barriers of mobile applications.

CC offers to mobile users virtual services with limitless computing resources from its database Gong, et al., (2010). Based on these observations, smartphones are eligible to send any computing task to the cloud by using remote task execution which is offered by different vendors such as Google cloud and can get results with less power consumption and less delay compared to the performing tasks locally on the smartphone itself (Ma, et al., 2013). CC is a new pattern for allowing limitless, appropriate needed on-demand entree to common space of massive computing resources with minimum effort from end users side. The end user uses cloud services without managing, maintaining, or controlling the cloud infrastructure. On the contrary, cloud computing kept doing some functions such as managing, maintaining, and operating the cloud to let the cloud services work in a professional, effortless, and considerate way for the end users. Zhang, et al., (2010) Stated that cloud database is a huge hard disk storage, high-speed network, and largest of servers. The importance of having these components, is to make the system more robust and high functioning, which ultimately results in more efficient processing.

1.3 Smartphones and Cloud Integration

Mobile phones can upload a text file to cloud server and ask to encode the file into a suitable format fitting the mobile phone capability with less power consumption and less processing time than doing the encoding on the mobile devices.

With the enhancement in the Hardware and Software in mobile devices, that helped in integrating mobile with cloud computing. This creates a new model called Mobile Cloud Computing (MCC), and this has become a potential technology technique for the mobile services users. The new Mobile Cloud Computing (MCC) paradigm incorporates the technology of CC with the mobile environment and enters the Information Technology (IT) world since 2009 (Dev & Baishnab, 2014). The report from Innovative Business Leader Research (IBLR) about the amount of MCC clients, has predicted to increase from 42.8 million in 2008 to 998 million nearly (19% of all mobile clients) in 2014. This shows the significance of CC for the users and how they depend on it in their daily life such as; E-mail, Global Position System (GPS), and Social media.

The advantages of cloud computing are now wide-open to mobile clients. Smartphone devices are capable to use remote infrastructures, as well as exchange data with different cloud providers. This has opened new chances for reducing application processing time and power consumption of smartphone devices by doing computation on cloud-assisted execution.

One of the common methods to reduce the power consumption and less delay in smartphone devices is offloading technique. Mobile devices can exploit advantages of the powerful resources infrastructure by migrating processing to remote servers such as Google Cloud (GC). Many studies such as Jararweh, et al., (2014), (Masiperiyannan, et al., (2014) and Qian & Andresen, (2015) have discussed this technique, in different areas, which includes making offloading decisions, partitioning and scheduling tasks.

Task offloading technique is one of the most efficient solutions to improve many of the smartphones device limitations, particularly in the energy limitation (Kumar & Lu, 2010). The recent version of smartphones have features which support the connection to the Internet and use remote computing resources. These types of smartphones are provided by cloud computing, which is accessible anytime and anywhere. Cloud-assisted smartphones help to overcome execution performance limitations for the applications which are running on poor resources by giving them access to more powerful remote resources. This means faster execution and more efficient energy which is more beneficial for the mobile users.

1.4 Research Questions

The research questions will focus on the following criteria:

1. What is needed for the bespoke mobile application to process tasks in high performance?

   To process tasks to a high standard, a pool of resources are needed to handle heavy processing tasks. So the question leads to the creation of the mobile cloud application to investigate the possibility of processing heavy tasks. Testing using the application and offloading tasks in a variety of scenarios with specified parameters to judge the performance.

2. How does the end user save mobile battery and processing time?

   In view of the rich media involving huge computational complexity, it is a big challenge for mobile users in terms of battery and time. The question includes the investigation of the possibility to send the intensive tasks from mobile to cloud. In this way, a mobile application needs to develop, design, and analyse the computation cost, considering all parts such as; cloud and network interfaces. The behaviour of data running on the mobile device based on a set of parameters, such as data type and size. These parameters give an indicator to check the file size before sending it by deploying decision engine. Thus meaning the application could fulfil certain requirements, for example, saving the mobile battery by deciding where to process the file. Different conditions are needed to offload the file, the elements will be considered by both, the deployed application and the core server before processing.

3. What are the factors that affect the power consumption and processing time?

   There will be different considerations of different network interfaces, and file sizes, as it is expected that these elements will have some impact on the parameters measured.

1.5 Research Objectives

In the previous section, challenges are highlighted related to a particular set of parameters. In the research question mentioned in section 1.4, the list of objectives will be addressed in the following section. These objectives provide a baseline for the research:

1. Review the behaviour of mobile application, network, cloud, network interface, in terms of send and receive files between mobile and cloud, in order to specify the main current challenges for smartphones performance. And Experimentally investigate how mobile cloud computing works to support the mobile users and review the impact of using different file size, in terms of power consumption and processing time.
2. Demonstrate the impact of managing Internet connectivity in the power consumption and processing time and test the impact of compressing the file and offload it to the cloud to reduce the processing time.
3. Investigate the benefit of save the files on the cloud to prevent the continuation of re-uploading the files.
4. Develop a mathematical model to validate the experimental results.

1.6 Knowledge Gap

A number of research studies have been discussed by different authors in terms of power consumption and processing time as mentioned in chapter 1 section 1.3. These studies and their solutions will be considered in the literature review chapter and briefly summarised. There are several achievements by other researchers in the field of offloading technique. However, this section will also address the gap in knowledge of various studies, as follows:

The reduction of energy consumption is a big issue and it has been attracting effort for many researchers such as Fitzek & Widmer, (2011) and Rodriguez, et al., (2010). Nowadays, heavy and complex computation tasks need a powerful and scalable computing environment, this means that there are obstacles to process the files or run applications on both normal smartphones and server. Qian & Andresen, (2015) stated in their study, that they served mobile device tasks on the standard server to reduce their energy consumption by offloading tasks to the server. However, testing offloading needs to be done in the cloud, because it provides a stronger computing environment and flexible scalability.

Short time usage (limited battery) is one of the most recognised obstacles in smartphones resources (Qian & Andresen, 2015). These limitations cause constraints when executing the intensive tasks computations on the smartphone itself. In addition, if the heavy task is performed on the smartphone it increases the power consumption and the processing time. During recent years, the improvements in software, hardware, and communications are noticeable and have shown significant progress. On the other hand, the improvements in battery life have not kept pace compared to the others parts of the smartphones developments.

However, smartphones are constrained by batteries which have a limited amount of power, this can have significant effects on performing the tasks. The main concern for smartphones users is the limited battery life of their devices. Due to the significance of this issue, academic researchers and industry professionals have been working on it, starting from hardware stage to the application stage. Some of the proposed techniques and methodologies such as; smart batteries, efficient operating system, powerful application, Graphical User Interface (GUI), and tasks offloading have facilitated the manufacturing of more sophisticated mobile smartphones (Naik, 2010)

Justino and Buyya (2014), proposed a mobile cloud architecture to improve the battery life and processing time, however, in the summary of their research they concluded that more network interfaces are required to check the computation cost in relation to battery level and delay time. Another architecture was proposed by Magurawalage, et al., (2014) in regards to energy consumption and response time whilst using Wi-Fi and 3G to offload tasks. One of the limitations of this research is they did not use 4G technology even though it was launched before they had completed their experimentations.

According to Shiraz, et al., (2015) computation offloading time needs to be reduced by reducing the file size which will ultimately decrease the processing time. Qian & Andresen, (2015) designed a dynamic system to decrease the power consumption during processing. However, they omitted to consider the implication that file size would have on the execution. Furthermore, similar to Magurawalage, et al., (2014) they did not consider the impact of the 4G technology.

Zhou, et al., (2015) proposed a model for offloading decision in terms of network condition and the results show 70% of the execution cost is reduced. In addition, it is a model to estimate the cost of cloud services. However, they also stated that handover for offloading decision making between servers is still an obstacle that requires further study.

In the industry of mobile communications, battery consumption is still a prevalent issue. With the increased use of multimedia functions and applications, there is a significant impact on the battery. In addition, this impact on the battery is considerably higher than the additional benefits that have been developed on mobile devices, creating an excessive gap between the capability of the mobile device and efficiency. Therefore, battery capacity has become one of the most important elements and has a high impact on computationally intensive tasks during the development process due to the impact on the end user. Qi, et al., (2016). In contrast, desktop computers are extremely powerful in regards to power efficiency and their ability to handle tasks, also they are cost-effective (Rawashdeh, et al., 2016).

1.7 Contributions

At the end of this research, the contributions to knowledge are envisioned to be as follows:

1- Estimate the power consumption and the processing time for the smartphone whilst processing tasks locally on the smartphone and offloading the tasks to the cloud server. This method is conducted by registering the start time and battery level. Once the tasks are completed the duration and battery level are then read.

2- Estimate the power consumption and the processing time for the smartphone to execute tasks by offloading to cloud while using Wireless Local Area Network (WLAN), Third Generation (3G), and Fourth Generation (4G), then the power consumption and the processing time for a single file which has been offloaded to the cloud which will be checked from the device, the readings for initial time, and battery levels are recorded. However, in this model, there is a decision engine to choose one of the network interfaces, because for each type there is a specific amount of power consumption. So it is based on the type of choosing the medium for offloading.

3- A new model is proposed to reduce the delay whilst processing the file. This model compresses the file then offloads it using the 4G technique, which leads to a decreased file size, this will help to enhance the quality of service, such as less delay and data integrity.

4- A new technique to add a secondary cloud server to serve the end users whilst the primary server is busy.

5- Back up from the uploaded files on the server after processing, to use it in the in the next round of processing.

1.8 Thesis Structure

In chapter 2, the scope of this research will be set by a detailed review of background literature and keywords related to the project will be investigated and reviewed. Since the proposed mobile application involves many functions, it requires a comprehensive method and architecture to handle all of the important entities which will be discussed in chapter 3. In chapter 4 there will be a review of issues related to mobile battery, processing time, cloud, and mobile application. In chapter 5 offloading through network interfaces using compressed files will be discussed in detail. The final experiment in chapter 6 will focus on processing file with multiple servers and the ability to save the files on servers to prevent the duplication of re-offloading the file.

Chapter 2 Background and Literature Review

This chapter will discuss the background and literature review on the smartphones, cloud computing, network Interfaces, and offloading technique. This chapter will be divided into four parts as follows. The first part (2.1), contains on a brief history about smartphones and the importance for them in the people daily life these days. The second part (2.2) presents an overview about the cloud computing, and its services benefits. The network interfaces were described in more details in the part (2.3), in part (2.4) a brief for mobile cloud computing is discussed. Finally, offloading technique, frameworks, and review the literature whilst using the offloading especially to overcome smartphone limitations in part (2.5).

2.1 Smartphones

In this section, an overview of the significance of smartphones will be discussed.

2.1.1 History of Smartphone over the Time

Smart devices have been launched as voice device and it was during the 80s which are defined as the first generation. Then the second generation created at the early 90s. This generation reformed the whole mobile systems. The second generation succeeded in deploying digital communication inside mobile system. In this type, data service was launched as a Short Message Service (SMS). In the year 2000, data services were updated by engaging General Packet Radio Service (GPRS), and it was considered as a 2.5 generation. Then Enhanced Data to Global Evolution (EDGE) was introduced as the last arrangement which was termed as 2.75 generation and was aimed to serve rural areas (Bruce, 2013).

With the achievement of the second generation, more requests were arising on network capacity, bandwidth, and data rate. These requests moved forward to the third generation 3G. 3G was created into two systems; the first system is defined as Universal Mobile Telecommunications System (UMTS) and CMDA2000. Later on, a second system is defined as High-speed Downlink Packet Access (HSDPA). Today, the achievement of 3G is noteworthy because of the increasing of the traffic between %300 to %700 yearly (Gartner Group, 2011). On the counterpart to mobile system development, new wireless technologies such as Wireless Personal Area Network (WPAN) and wireless Local Area Network (WLAN) were introduced into smartphone devices. For instance, WLAN service is deployed in many regions like campuses, airports, homes, and hotels. The main reason for these new technologies is their mutual match to the smartphone devices because they afford greater data rate and less cost comparing with the cost from mobile system data. So today a lot of places deliver free entry to their WLAN.

WLAN was created as a spreading to Local Area Network (LAN) and it is not part of mobile system evolution. The main difference between wireless system and mobile system is the coverage range. The coverage range in the mobile communication system is a few of kilometres while the coverage in WLAN is a nearly hundreds of meters, (Cisco, 2012).

The development of the mobile devices advancements has assumed a part in the developing of the mobile phones. Additionally to the growth in the communication technology, the major effects were the improvements in the software and hardware of mobile devices. The development in the mobile phone hardware adds supporting feature to mobile devices such as, GPS. As well as a drive to lighter and smaller mobile device, more features and capabilities were introduced. On the other hand, the developments in the mobile software helped users in different types of Internet services such as e-mailing, gaming, office applications, and web browsing.

The merging of the mobile communication systems such as mobile computing with multimedia broadcasting, and the Internet, has structured the new pervasive Information and Communication Technology (ICT) system. This is obvious nowadays and will be prominent in the future. The Fourth Generation (4G) mobile device communication is the best stage toward this merging. The 4G guide is to achieve all-IP design where this design reduces the expansible, accommodating, and infrastructure costs for upcoming services on the Internet broadcasting and the computing. Additionally, the all- Internet Protocol (IP) design supports a smooth connectively amongst heterogeneous networks termed real Internet mobility (Bonomi, 2010). The 4G generation combines all worldwide networks and all terminal kinds in one mobile environment (Khan , et al., 2009). In this development, the mobile devices shift from voice-centric to IP centric devices (Akan & Edemen, 2010).

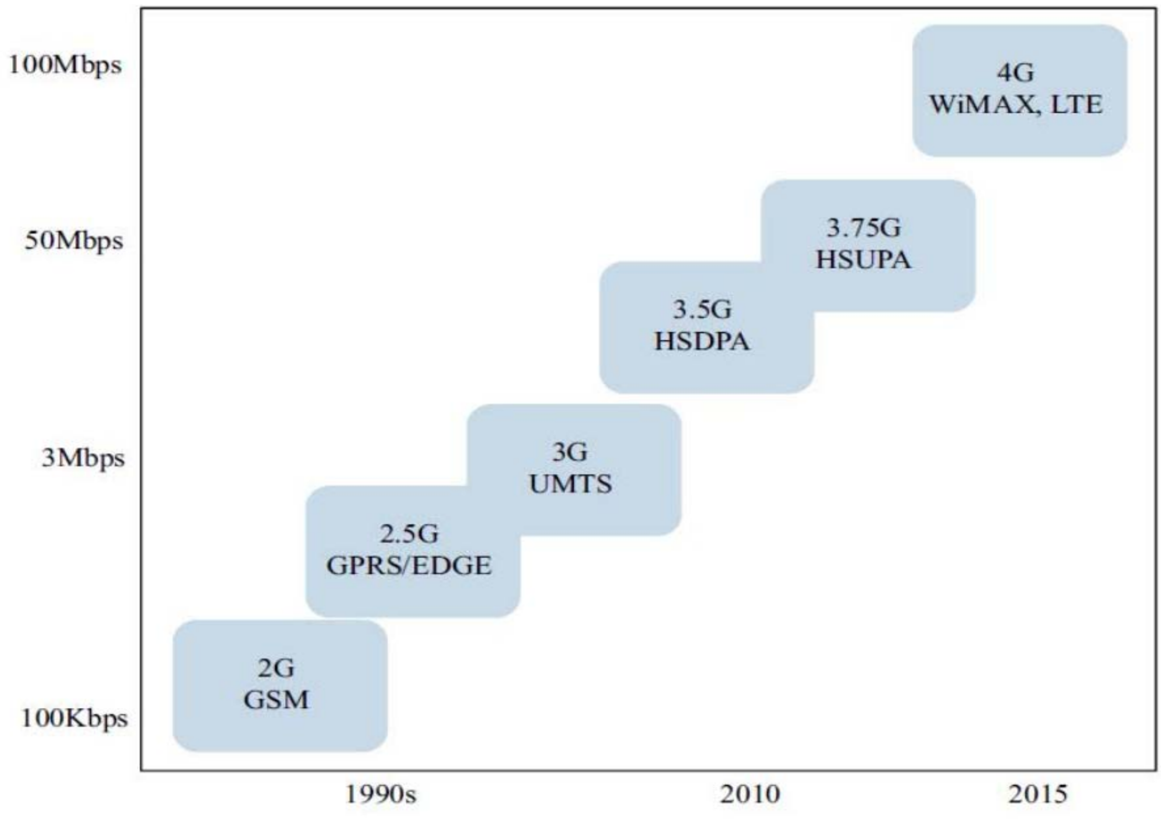


Figure 2. 1 Mobile Technology Evolution

The general view of mobile systems developments is shown in figure 2.1, (Akan & Edemen, 2010).

2.1.2 Smartphone Definition

According to Oxford dictionary 2017, Smartphone is defined as:

A mobile phone that executes a lot of functions similar to the computer, commonly having a touchscreen interface, Internet access, and a different types of operating system which enable to run and download applications.

As well as, the mobile device includes an intelligent functions for instance face recognition.

2.1.3 Smartphone as an Origin of Data

A smartphone fits an important origin of data in the recent ICT. User use smartphone to make and edit files, audio, record video, and capture images by using microphone and camera. Additionally, smartphones are capable of affording their locations information by using GPS technology. Nowadays, smartphones include many kinds of sensors: accelerator, digital compass, ambient light sensor, three-axis gyroscope, and proximity sensor. According to Lane , et al., (2010) each sensor is an origin of data and this helps smartphones capabilities and functionalities.

The software and hardware of smartphone offer the ability to create a massive amount of data in different kinds of format. On the other hand, smartphones are capable of creating data over their capabilities. For instance, the user can capture pictures but cannot handle on them such as reformatting or face recognition. In a few situations smartphones are capable of doing some enhancements, but at the same time, they need a concentrated processing which leads to drain their battery.

2.1.4 Smartphone Purposes

The purposes of smartphones are broadly varying from stock marketing to kids gaming and elector campaigns. Social media network applications can work in the best way but need the strength of smartphone capability. These features are essential. Moreover, smartphones find their way in the business sector such as stock market, creative products, and banking. This happened because smartphone suits the demands which a lot of people who require mobility, and processing on the same operated device. Examples of smartphones abilities are holding soft media such as travel e-ticket, e-print, web news, online gaming, and e-shipping. All of these abilities impose the need for the smartphones to be a widespread mobile devices (Wang, et al., 2017).

2.1.5 Computation and Smartphone Limitations

The mobile devices consist of a combination of both software and hardware. The Software part consists of the applications and the operating systems. The hardware part consists of Random Access Memory (RAM), Central Process Unit (CPU), Hard Disk Drive (HDD), and Wireless Network Interface (WNI). The power is consumed in smartphones by hardware parts, which are being run using the operating system that based on the applications requirements. As a result, the consumption of power is based on the application type, because every service delivered needs a specific hardware such as RAM and CPU.

With the improvements in smartphones capabilities, mobile devices can access a massive amount of Internet materials. The new sites and social media networks take around 80 % of websites. (I. Union, 2010). Additionally, many of web access activities are originated from smartphones. A number of requests from smartphones are higher than the number of the requests from desktop computers and this even happened in during the year 2008 (The I. Union, 2010). This is expectable because of the enhancements in wireless speed, as well as the smartphone, is crucial for users (Vriendt, et al., 2002). A related study done by Nielson organisation (2009) figured out that Australian people stay more time using the fast Internet. Some people switch to watch online media, for instance, YouTube instead of TV (Dillistone, et al., 2010). The use of the smartphone for Internet contents such as social media and the news was raised in a very fast way in the latest years according to Gartner Group's report (Gartner Group, 2011). This report debates the percentage for the rich media or multimedia, such as audio, video, and images. This lead to multimedia on the Internet has been developed comparing to textual contents.

Altamimi, et al., (2015) Stated that there are unique limitations in mobile devices which have become more prominent for the mobile users. In addition, the enhancement of functions and facilities used by smartphones effect and drain mobile resources which has a direct impact on mobile users. Furthermore, mobile operating systems such as Apple iOS, Android, and Windows Mobile have the capability to execute advanced applications in line with computers functions. To some extent, each smartphone has capabilities to handle specific tasks, and this indicates that the smartphone has limited resources.

Smartphone battery remains one of the notorious concern for mobile users. This is noticeable through rich applications graphics and multimedia, which cause a higher burden on the consuming battery. Moreover, although there are issues with limited storage capability the major concern to users relate to the battery drainage caused by various complex computation that end users are interested (Qi, et al., 2016).

2.2 Cloud Computing

In this section, an overview of the cloud computing (CC) and services provided by CC and some of the meaning of the cloud computing are presented.

2.2.1 Overview on Cloud Computing

CC is a new pattern for allowing limitless, appropriate needed on-demand entree to common space of massive computing resources with minimum effort from end users side. The end user uses cloud services without managing, maintaining, or controlling the cloud infrastructure. On the contrary, cloud computing kept doing some functions such as manage, maintain, and operate the cloud to let the cloud services work in a professional, effortless, and kind way for the end user. Zhang, et al., (2010) defined cloud database as huge collections of hard disk storage, high-speed network, and large banks of servers. The Internet is the medium to deliver the cloud services. Cloud computing is created on a physical database that deals with the infrastructure of the cloud. In the computing technique. Dillon, et al., (2010) integrated the distributed computing, utility computing, and virtual computing that all allow the data centre to provide the cloud services and to power cloud functionalities. The combination of these technologies forms cloud computing. Nonetheless, the term cloud computing is new because of the way it obtains the data using computing resources on demand and the liability of the service usage for pay-as-you-go basis.

CC is a new computing model that always keeps improving and spreading. It is empowered by parallel computing, hardware, and web services. CC creates massive changes in ICT. CC defined as a new pattern for allowing limitless, appropriate needed on-demand entree to common space of massive computing resources such as storages, servers, network, services, and applications. These resources can be released and provisioned with minimum effort or service from end users perspective. There are some of the samples for CC structure such as Google Applications, Amazon. CC supports companies to improve the IT services, and develop several applications to get unlimited scalability on requiring services for the IT structure, as well to increase the profits. The Clients of CC could be single users, users in other Clouds, enterprises, and organisations.

CC has massive resources and rich infrastructure, which developed over data centres and networks of servers. Storage space and computing power on a cloud are “limitless” compared with smartphone devices. These features make CC a great complementary technology part to overweight the smartphones limitations in terms of computation. However, the accessibility of the cloud to the end user’s is one of the advantages of this technique.

The CC has different types of patterns: Private, Public, Hybrid, or Community. In Private Implementation models, the cloud infrastructure can be designed for the user which can be a company or organisation. It could be operated, owned, controlled and managed by users or by the third party. In public models, the infrastructure is designed for public users and operated and owned by cloud service providers. In the hybrid model, the infrastructure is designed by merging two or more types of cloud services models.

The last improvements in mobile computing technique have modified user favourites for the computer. The last article form Juniper Research Group mentioned that the enterprise market and customers for cloud-based is anticipated to rise by $9 billion in 2014 (Holman, 2011), because of using MCC. The amount of communication and computing machines are being changed by smartphone devices, such as Internet browsing, digital cameras, and PDAs.

2.2.2 Cloud Computing Definition

CC is a powerful technology to execute massive-scale and heavy computing over the equipped infrastructure. This technology reduces the needs of Software, dedicated space, and expensive infrastructure. (Hashem, et al., 2014)

National Institute of Standards and Technology (NIST) identifies it as a paradigm for convenient, enabling ubiquitous, on-demand access to a massive of configurable computing resources such as services, servers, storage applications, and networks. These resources require slight management effort or some of the collaboration from the service provider. (Mell & Grance, 2011).

Cisco defines CC as IT services and resources which were released from the implicit infrastructure at scale and provides basis on-demand.

CC is a powerful pool of resources as services through the Internet, and this helps to offer information and communication technology to users everywhere and anytime (Zhan, et al., 2015).

2.2.3 Cloud Computing Services Models

The basics of CC are virtualization of the resources and scalability of the services in a proficient way. As many end users can use the same computation resources in a scalable method, in addition there is a high availability. This means that cloud computing has to abide by the Quality of Experience (QOE) and Quality of Service (QoS) of the services to the clients. The availability defined as the cloud is available on demand or when the users request for the service. The scalability defined as the cloud can adjust to up or down the resources as needed to meet the user's requests (Gong, et al., 2010).

According to the services model which provided by the cloud, the researcher has devised different kinds of services model and titled them. The levels of cloud service differ in line with the end users to use the services. The following CC services are:

i. **Software as a Service (SaaS):** is one of the cloud services which offers software applications to the end clients, In this paradigm, end users get access to the cloud resources through the application layer. The applications are proceeding on the provider's cloud which will be accessed by the end users via an easy to use interface. This means it is not required to have software licensing, operating systems, storage, installation and support hardware infrastructure. So the end users will not have any concern about the mentioned issues. To take the benefits from this paradigm, the end users can gain this type of service by using the web browser (Yang, et al., 2015).

ii. **Platform as a Service (Paas):** In this paradigm, end users have a slight control over the services and infrastructure of the cloud. It is close to SaaS apart from the ability of the end users to use the cloud as a platform to create, host, run applications within the rules for the cloud service provider. End users can control and manage the deployed applications. However, there is no ability to control and manage operation system, storage, and the underlying cloud infrastructure (Yang, et al., 2016).

iii. **Infrastructure as a Service (IaaS):** Is a model give the end users to have the full managing of the usages and to get full access to the cloud resources. One of the vital issues in cloud resources is that users can use the pool of resources on demand, such as operating systems, network, and storage. As mentioned in Paas model, there is no control for users over cloud infrastructure, but they can custom the storage and operating system based on their requirements for the deployed application in terms of users. One of the popular examples of IaaS is Amazon, it offers processing ability and storage in the cloud to the users based on their requirements and demands (Duan, et al., 2012).

iv. **Everything as a Service (XaaS):** in this cloud service shows the whole or some of the previous services such as Microsoft Windows Azure, Google Doc, IBM Cloud, Salesforce, and Amazon. A hierarchy figure 2.2 below illustrates the cloud services levels:

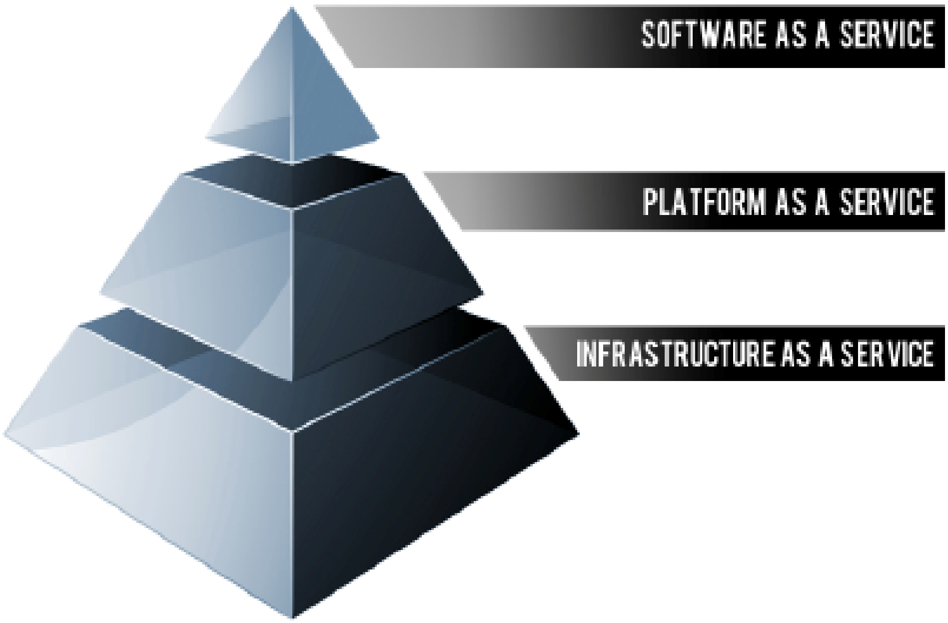


Figure 2. 2 Cloud services levels (Huber , et al., 2012)

To illustrate the services and products which are offer by cloud services, table 2.1 shows them with examples for each service.

Table 2. 1 Cloud types and services

Cloud Service Name	Service Available	Examples
SaaS	Clients Apps Office Productivity Virtual Desktop	Cisco Google Doc IBM Cloud
IaaS	Virtual Servers VLAN Networks	Amazon Microsoft Live Mesh GoGrid
PaaS	Database Apps Server File Sharing	Microsoft Azure Amazon SunCloud

2.3 Overview for Network Interfaces: Wi-Fi, 3G, 4G, and 5G

To gain Internet access from a mobile device, Wi-Fi and cellular technology are required. Cellular technology has been improving noticeably from last 20 years. The cellular technology evolution generations started from First Generation (1G), Second Generation (2G), 3G, and 4G to upcoming Fifth Generation (5G). In the 1980's, global Platforms launched Global System for Mobile (GSM). Additionally, recommend telecommunications companies to offer a good cost and the availability. There are limitations for each upcoming technology, and the improvement becomes main issue to emerge in the new technology. The recent technologies were designed in terms of the efficiency and speed capabilities for of cellular networks, this helped the formation of creating a new model (Kothari, et al., 2017).

Mobile networks are playing a crucial role to determine the efficiency of smartphone user application and offloading technique. Mobile user application which often interacts with the backend service provider diminishes the performance because of the limitiation of the mobile networks in terms of the bandwidth and latency. Offloading technique depends on the mobile network interface to support remote execution such as deliver or send the requests when they completed. The following subsections provide insight about network interfaces types.

2.3.1 Wireless Fidelity

Wi-Fi is the most common technology, it can be located indoors. It is based on the IEEE 802.11 standard. Wi-Fi is designed in two models, ad-hoc network model and infrastructure models. In ad-hoc mode, the way of communication is performed directly between devices which are provided by Wi-Fi equipment. According to the infrastructure model, device equipment's in the local network connect to Access Point (AP) or router to communicate together or get access to the internet (Sathiaseelan & Crowcroft, 2013).

Wi-Fi enables the direct connection between multiple devices and aids to share or transfer the data among them. Actually, one device is required to adjust the Wi-Fi directly to take the role of an access point and send the data to the device or broadcast it to a group of devices. This means, all the advantages of Wi-Fi communication such as range and data rate are maintained (Antonopoulos, et al., 2014).

The Wi-Fi traffic offloading is becoming more attractive because of the upcoming ultra-dense cellular networks. Nevertheless, the offloading decision through Wi-Fi and Wi-Fi Access Points (WAP), must be selected cautiously so as to not affect the users from their offloading experience perspective. Offloading using Wi-Fi is employed to transmit the data traffic which is initially targeted for the cellular network. More recently, Wi-Fi has been recognised for a few beneficial factors, the first is that is able to use an unlicensed frequency band within the network and does not overlap with other frequencies. Other reasons include; most mobile devices have the capability to use Wi-Fi, it is cheap to manage, most public places have accessible access points and it is seen as more efficient for energy. Thus meaning, mobile developers see this as a positive resolution to network congestion (Fakhfakh & Hamouda, 2017).

Wi-Fi offloading, which is happening when the mobile users use the Wi-Fi connection for sending and receiving the data, has been getting a lot of interest as one of the efficient practical solutions, with low financial burden in practice. For the network operator, mobile users can create Access Points (APs) easily and then install Wi-Fi AP with minimum cost. In practice, the operators offer Wi-Fi services by using hot-spot (Lee, et al., 2014).

2.3.2 Third Generation System (3G)

3G is created to offer faster and simpler wireless communication. The new development in data speed and high quality helps to support and offer new services such as; video calling, video streaming, and high speed in browsing. This leads to the creation of new technology such as 3G, with a high speed of around 144 kbps to 2mbps or greater. The standard for 3G is W-CDMA. As mentioned earlier, for each generation there are limitations. In this generation, power consumption is very high, as is the cost of using the service (Kothari, et al., 2017).

The evolution of increasing the numbers and types of smartphones forces the 3G network to the limit. In addition, the capacities are becoming difficult and expensive for mobiles network operators, and this could affect the adequate provision and access for the subscribers (Ristanovic, et al., 2011).

In this type of network, the cellular interface contains three states as shown below:

- **Dedicated Channel (DCH).**
- **Forward Access Channel (FACH).**
- **IDLE Mode.**

There is no network activity whilst the radio interface is in the ILDE mode which means very low levels of power consumption. However, when the smartphones need to send or receive packets, the interface switches to the DCH state through establishment of a signal connection and then dedicates the transmission channel. The process in DHC needs a control message exchange between the backbone network and smartphones. Until this stage, the radio interface stays in the DHC state whilst sending and receiving data. When the time is expired the radio interface will move to the FACH (Geng, et al., 2015).

2.3.3 Fourth Generation System

4G system is introduced to cover the previous generation limitations, offering Internet access at much faster speed and less power consumption. The theoretical data rate for 4G is nearly 1 Gbps comparing to 3G is 100 Mbps. However, the update is still yet to turn up (Nossenson, 2009). The 4G system will finish the replacement and will lead to safe and secure Internet Protocol (IP). Additionally, it will also allow users to use different kinds of services such as; data, voice, and multimedia, anytime, anywhere, with a QoS. The limitations for the 4G system are, the customer needs to purchase new equipment to launch 4G network because the old devices do not support 4G, it is not available everywhere, finally, it has many bugs (Kothari, et al., 2017).

The massive development in the mobile devices industry, which is mostly helped to the wide using of multimedia website, and social network such as Facebook, Twitter, has led to introduce 4G communications network. The 4G is created to increase the network capacity and offer a high data rates. Moreover, the rich data such as; game, music, and video is one of the major reasons for this vital revolution. In addition, 4G helped in the important matters for the network operators in terms of Quality of service, cell overloading, and resource management. This shows the need for the efficient traffic offloading techniques (Antonopoulos, et al., 2014).

2.3.4 The Upcoming Fifth Generation System (5G)

5G generation of mobile cellular called as 5G, which is the updated generation of the old existing 4G. 5G uses mimo to optimise frequency reuses over 4G, which means more mobile customers can use it at the same time and in the same area with high speed and unlimited download monthly (Yanikomeroglu, 2012). This generation technology is expected to offer a greater capacity of the radio which will help to meet the high demand of bandwidth from user requests. (Fakhfakh & Hamouda, 2017).

5G Wireless network will support increases capacity up to 1000-fold, which is increasing in the capacity compared to the current networks. This means, 100 billion devices are expecting to connect, as well as around 7.6 billion mobile subscribers because of the revolution in the smartphones, sensors, and electronic tablet with speed up to 10Gb/s for the single user (Andrews, et al., 2014).

5G (IMT2020) is promising to support ina different variety of categories (Shafi, et al., 2017) as shown below:

1- Ultra-reliable and Low Latency Communication (URLLC): this means stringent requirements for availability, latency, and reliability. There are a lot of examples are tactile such as; vehicle-to-everything (V2X), intelligent transport systems, smart grids, public protection, Internet applications, wireless control of industrial manufacturing, remote medical surgery (Simsek, et al., 2016).

2- Enhanced Mobile Broadband (eMBB): in fact this category is similar to what is available nowadays but with an increasingly seamless experience for the user through enhancement in the performance. This scenario covers a variety of cases such as; hotspot and wide area coverage. For hotspot, a very high traffic capacity is required to support high user density. For wide area category, high mobility and seamless coverage are desired in addition with much enhancement in users data rates. Therefore it is important that the needed for user data rate is much higher than the other wide area coverage category.

3- Massive Machine Type Communications (mMTC): this category consists of a massive number of devices with high or low volume in terms of the non-delay-sensitive data. Additionally, the devices should have a much-extended battery life and at the same time low cost.

Table 2. 2 Comparison of Cellular Network

Generation	Year	Data Rate	Use
1G	1981	2 Kbps	Analog Voice
2G	1992	64 Kbps	Digital Voice
3G	2001	2 Mbps	Mobile Broadband
4G	2010	100 Mbps	Mobile Broadband
5G	2020	10 Gbps	Fiber Broadband

The next generation of mobile Network is 5G, and it will completely launched in around 2020 to meet customers’ requirements (Yanikomeroglu, 2012).

2.4 Mobile Cloud Computing

This section summarises the previous part in relation to the valuable opportunities which are enabled through cloud computing to smartphone devices. In addition, there will be the discussion about using the mobile device with the enhancement in the Hardware and Software in mobile devices. This helps in emerging mobile with cloud computing to create a new model which researchers call Mobile Cloud Computing (MCC), which becomes a potential technology technique for the mobile service users. Researchers have managed to merge cloud structure with smartphone devices, in order to control the limitations of smartphone resources to run suitable applications (Atre, et al., 2016).

The new MCC paradigm incorporates the technology of CC with the mobile environment and enters the IT world since 2009 (Dev & Baishnab, 2014). The report from Innovative Business Leader Research (IBLR) about the amount of MCC clients, is predicted to increase from 42.8 million in 2008 to 998 million nearly (19% of all mobile clients) in 2014. This means the significant of CC for the users and how they depend on it in their daily life such as; E-mail, GPS, and Social media.

One of the reasons that MCC has become widely used is because of the complex architecture which aids the execution of processing to overcome the software and hardware mobile device limitations. So the heavy tasks can be handled on the cloud server. Cloud process the files and sends the results to the smartphone devices (Wang, et al., 2017).

2.4.1 Mobile Cloud Computing Structure and Components

MCC is a new technology that aims to provide an intelligent platform and is rich of applications which can be delivered between the user's fingertips in an efficient way. The processing over the cloud without the intervention and use of the mobile device is called offloading and leads to a more efficient process. This is the key to integration between smartphones and cloud servers. Previous studies such as Mobile Assistance Using Infrastructure (MAUI) which is proposed by Eduardo, et al.,(2010), and ThinkAir proposed by Kosta, et al., (2012), and CloneCloud presented by Chun, et al., (2011), computation offloading is the main technology in MCC and it has grown to improve the end user's experience of mobile applications. Figure 2.3 below shows the structure of mobile cloud computing:

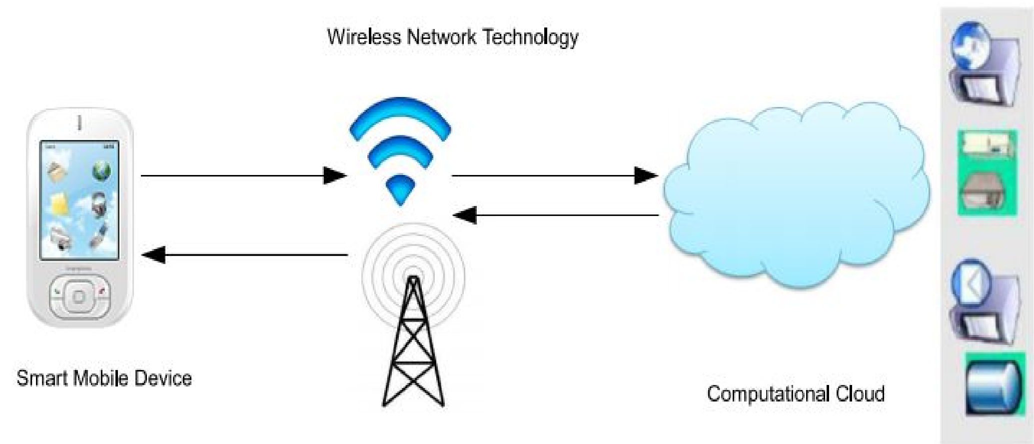


Figure 2. 3 Mobile Cloud Computing Model (Shiraz, et al., 2013)

2.5 Offloading

In this section, an overview of offloading technique will be presented.

2.5.1 Offloading Definition

In the early years of the 1970s, the offloading system was used at its early development stages. It was first used among servers to find an optimum load of balancing. Offloading can be described in many ways, for instance Zhou, et al., (2015) has defined offloading as the method that can improve the proficiency, quality, or even the performance of some calculation tasks by assigning the servers with an enormous powerful computation to deal with heavyweight data rather than the local servers. Memory, energy consumption, processing, storage, as well as the processing time are examples of the computation factors, where more than one of these factors can be used at the same time (Zhou, et al., 2015).

The new CC resources, support the limitations of the problems which faced mobile computing, meanwhile the cloud resources are scalable and are accessible anywhere and anytime in the world. As mentioned in section 2.2.3, there is a massive number of cloud services such as Amazon, Apple ICloud, and Windows Azure. These cloud providers, they use their platforms to offer different kinds of services. To exploit these services, extensive research has been studied on cloud computing and mobile networks. Since both of them are quite dynamic in terms of the infrastructure and services. To emerge CC with mobile devices, cloud offloading is one of the distribution computing systems which support to leverage resources involving mobile devices.

2.5.2 Offloading Techniques

Offloading techniques as described in the previous section depends on the definition which has been debated, the communication between local and remote machines is the medium required in order for the offloading to be executed. Hence, that is why the communication is an essential key for the offloading tasks. However, these communications require local processing and this would result in too much energy consumption. Network interface together with the processor are the biggest pressure on power consumption thus offloading need to be aware of the total consumptions and communications cost. There are many benefits of offloading such as to enhance the performance and save energy ((Flores & Srirama, 2014) and (Jararweh, et al., 2014)). To exploit these benefits and make the offloading decision more beneficial, there are different factors which should be taken into account such as file size and network interface.

Offloading is a simple mechanism for processing files, using powerful resources. It can trigger two valuable factors such as, the availability of resources and the benefits the user will gain Figure 2.4 shows the basic workflow for the offloading process.

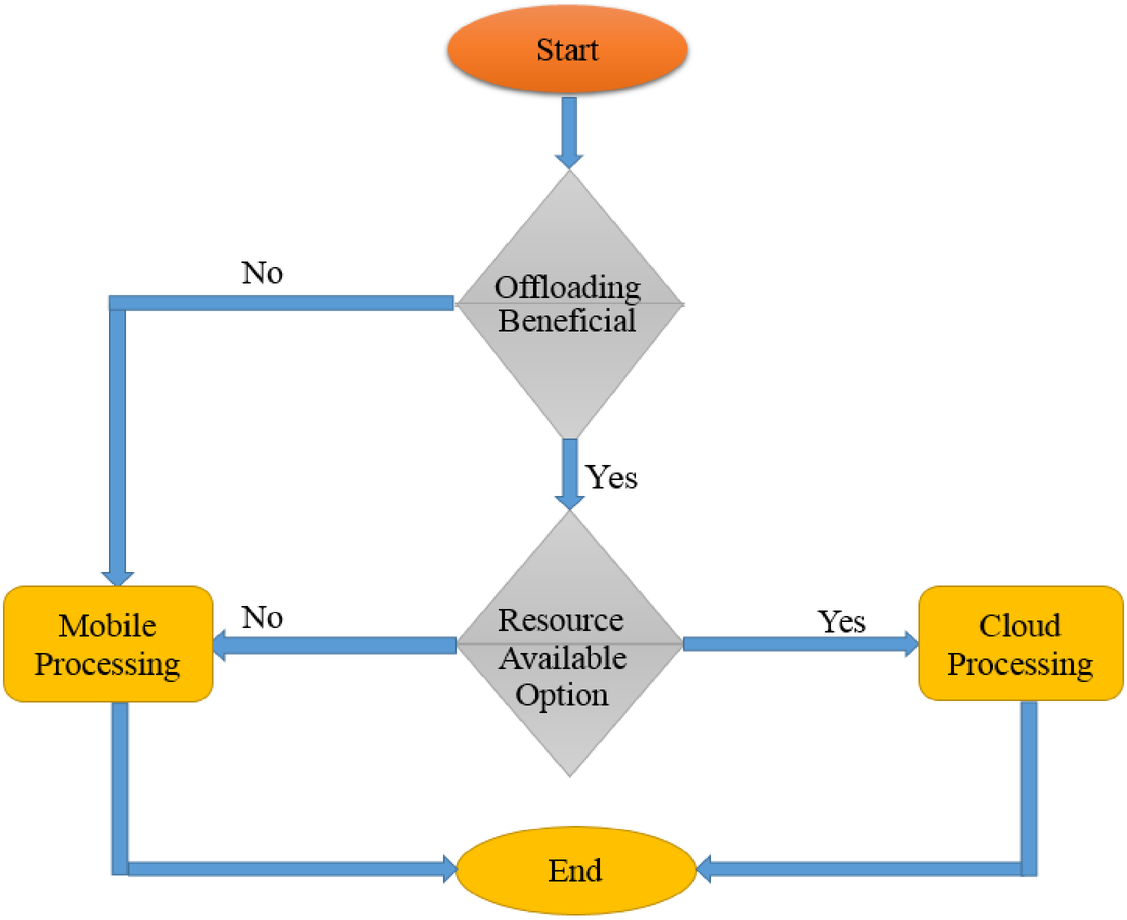


Figure 2. 4 Offloading process

Offloading has many motivations and advantages that make it an essential to be used as Kumar, et al., (2013), has proposed. To begin with, offloading minimises the required time of processing and works towards raising the response for the processing. In agreement of this, Wolski, et al., (2008), and Gao, et al (2012) have proposed offloading decision algorithms based on the bandwidth with an objective to support the offloading with a code in order to select the optimum wireless medium. In addition, recent studies have shown superior results when using a powerful server rather than restricted power servers, hence offloading can rise up the performance of an application. Moreover, offloading was initially used to balance the load between the used servers. Finally, on the word of Stephen, et al., (2012) offloading can be used as a technique to preserve the energy consumptions and minimise the latency which can be observed clearly from the smartphones devices.

Offloading technique is categorised into two types: dynamic offloading, and static (Sharifi, et al., 2012). According to dynamic offloading, it occurs at runtime, and offloads to a pool of powerful resources, because the local resources are not available and start to offload when the required resources are insufficient. For example, when (MAUI) Eduardo, et al., (2010), they used dynamic offloading to reduce energy consumption for smartphones. In this way, they annotate a particular server to process the file. Inside the mobile application, there is a profiler to assess the current situation for the smartphone and to make a decision for offloading.

Offloading computation method and data storage to a remote powerful server is an effective way for different reasons such as to reduce local overhead, and this concerns a number of issues, which are, network delay, power consumption, and device capacity (Li, et al., 2015).

The offloading technique has been invented for multiple reasons such as improving performance, saving energy, and load balancing. The offloading mechanism is classified into three major styles based on the kind of remote as shown below:

- The first approach where they are offloaded to local, this means it could be on the same network or close to it, where the mobile existing (Wolski, et al., 2008). The mobile sends the intensive computation to the cloud, request to execute the given tasks.
- The second approach offloads to a web proxy, while the proxy is working as an intermediate between mobile and web server (Kelenyi & Nurminen, 2010).

- The last approach is to offload to the cloud where the cloud offers a pool of resources, such as storage and processing to a mobile device (Kumar, et al., 2013).

Offloading technique is from the hottest and attractive topics for many researchers. Cloud could be used for multiple different types of tasks to help smartphones to save the energy. Therefore, it is important to estimate the energy which is consumed in task offloading to make task offloading decision. Kumar & Lu, (2010) proposed an energy model when mobile devices perform computation and communication. The decision is made based on the energy cost of computation for a given task and mobile communication. The proposed model shows the impact of the communication on offloading, and explains that the offloading is useful especially in a heavy computation. Static offloading divide codes to parts, then decides which part should be offloaded to execute it. Balan, et al., (2007) checked the codes for each application and the functions for each procedure, after that the value can be seen when offloading is executed. The decision is based on raising static utility function for each application regarding to delay and resource usage.

2.5.3 Offloading Frameworks

The offloading problem has been studied by researchers who proposed some of the solutions regarding offloading objectives, approaches, architectures, and applications model, following studies illustrate them:

Jararweh, et al., (2014) proposed a possible solution to increase the limitations of mobile computing, such as; storage capacity, battery life, and processing power, by sending tasks to cloud computing then return the results to mobile devices. This helps to reduce the time for performing intensive tasks and needed power. However, joining smartphones to the cloud suffers from the power consumption when transmission and high network latency especially when using Long Term Evolution (LTE) or 3G network interfaces. They introduced cloudlet-based to decrease the power consumption and network delay for the mutual applications, for example, multimedia applications. The results indicate the new paradigm aids to save the power as well as decrease the communication latency for the smartphones.

CUCKOO is a module which used an Android application that has the capability of taking a decision whether to execute part of the task on the mobile or send it to the server. The CUCKOO module was developed by Kemp, et al., (2012), and was a perfect fit with an innovative tool to perform locally or remotely. Wireless medium, memory, and device type are some examples of the local resources of the smartphones which have a huge impact on processing offloading decisions.

Kovachev & Klamma, (2012) proposed a model for offloading called Mobile Augmentation Cloud Services (MACS) which permits Android applications to perform over the cloud. The model is based on Android platform and study based application allocation and decides which parts of the applications to perform locally and remotely over the cloud. The offloading decision is created as an enhancement problem regarding the cloud parameters and the device, such as the connection between the mobile and the cloud, and the remaining energy for the device. In Cuckoo, the framework allows only a static partitioning, whereas MACS permits a dynamic application splitting.

Flores & Srirama, (2014) proposed a Mobile Cloud Middleware (MCM) model, which helps to delegate mobile tasks and dynamically assigned to the cloud infrastructure. The testbed shows that mobile cloud middleware improves the effectiveness of the quality of service for smartphones and service them in real time responses in mobile cloud applications. MCM as well enhances the orchestration and incorporation of mobile tasks allocated with a minimal data transfer.

Hung, et al., (2011) proposed a framework called MCC which helps to process the interoperability issues and takes the intensive processes from mobile devices in ease method. The results from using mobile cloud middleware show improvement to the QoS for devices and helps to preserve response time for mobile cloud applications.

Satyanarayanan, et al., (2009) offered a new system technique to handle the suspended cloudlet which is an intermediate between the mobile device and cloud. The new system called Virtual Machine (VM) will serve the mobile once the cloudlet is down. This technique will start to transfer the precise functions from damaged cloudlet to core server. The prototype in this new system includes on Mobile is represented by Nokia N810, Cloudlet is represented by desktop running Ubuntu Linux.

Jararweh, et al., (2014) offered a paradigm called a Cloudlet-based MCC system which has the purpose to decrease the delay and the power consumption while using MCC. The proposed Cloudlet model merged with mobile cloud computing concept to propose a new framework for the MCC concept. The new framework decreases the communication latency and the power consumption for the smartphones when the mobile device offloads the tasks while keeping the good quality of services.

Fesehaye, et al., (2012) proposed a design cloudlet and service architecture. The experiments focus on different kinds of mobile applications such as; video streaming, collaborative chatting, and file editing. The results by using NS2 simulation show hoe to get a high performance using cloudlets around the cloud in terms of the throughput and delay in the transferring for the system, it depends on the number of hops between cloudlets when transfer data. So there is a better performance when the number of hops are not more than two. They proposed an intelligent decision to support whichever gives the least delay will then be used.

Existing mobile cloud system uses a technique as presented in figure 2.5 (Fesehaye, et al., 2012):

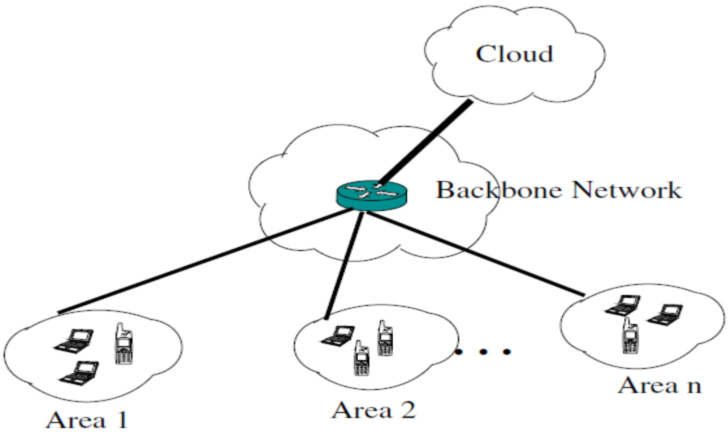


Figure 2. 5 Current mobile network model

Furthermore, they studied cloudlets as networks that cover a particular zone to provide IMCA model basically creating a proximal community network. In their study, they assess the effect of cloudlet network with Wi-Fi, for instance, high bandwidth wireless on IMCA. For this influence, they designed a network service architecture for a new IMAC model. There are two ways to send the data:

- Firstly in the cloudlet-based scenario, the smartphone users use the cloudlet to transmit data.
- Secondly cloud-based approach, mobile users use the core server to transmit data.

In the first scenario, cloudlets work as a temporary service machine. Cloudlets update the status for the mobile nodes at the final stage in every communication. For instance, regarding editing the file; the file is downloaded to the closest cloudlet for the smartphone nodes, then users do edit once editing is finished, the new editing and status of the mobile node are updating to the cloud.

In general, every mobile device uses the backbone network which is a part of computer structure that helps to connect to the core cloud as well as to connect nodes with each other in the same zone or different zone. However, these approaches could cause a high latency as well as the energy consumption for the mobile using 3G/4G than Wi-Fi. Bandwidth weight in 3G networks topology is higher than Wi-Fi (Balasubramanian, et al., 2010).

Cloudlets have up to now offered to support smartphone clients when they joined to cloudlets directly connected to them in different services such as processing and storage (Satyanarayanan, et al., 2009).

2.6 Summary of Energy Preserving Techniques for Smartphones

Mobile device energy is one of the most important issues in mobile resources. Many techniques and methods have been offered to reduce mobile device power consumption such as power sleep mode, redesign communication protocols, efficient operating system, smart batteries, and power schedule, efficient GUI.

Despite the convenience offered by the compactness of smartphone devices, there are many limitations in terms of its hardware, especially its battery. In contrast to other components, the speed of development in enhancing the energy of smartphone devices batteries has been slow (Gao, et al., 2016).

Shu, et al., (2013) proposed a new model known as e-Time, which stipulates that data transmission is essential with mobile devices. However, it is unreliable and unpredictable through several factors, such as; wireless connections, user mobility, and bandwidth. Consequently, these factors increase the power consumption of the mobile. Regarding to the results, energy usage will be increased through the bad connection. A novel energy-efficient data communication strategy between cloud and mobile proposed called e-Time. The new model used data if there is a bad connection because it will make a delay in data transmission and increase the power consumption. So, in this case, there is a tradeoff decision. The results show 20%-35% energy saved by the e-Time model which is proposed by (Shu, et al., 2013).

Masiperiyannan1, et al., (2014) indicates in their study that to save energy using computation offloading is based on three issues, a number of computation to be executed, and number of data to be transmitted, and wireless bandwidth. Furthermore, they mentioned the current research focuses on deciding whether to offload the heavy tasks by forecasting the relationships between these three issues.

Altamimi & Naik, (2014) developed a practical Decision Engine (DE) technique and the engine is accurate in the offloading decision. The new engine is capable to transfer from mobile devices to cloud computing, to recognise the Energy- as- a- Service technique. The DE developed on Android mobile, and Amazon ES2 and S3 clouds to do some of the investigations to determine the cost of system factors used by the DE.

Smartphones are essential in daily life because they have powerful resources that offer users to perform several kinds of tasks. MCC helps to perform that in less time and with less effort. However, smartphones have limited resources such as memory and battery. MCC has powerful resources which help to avoid such limitations in smartphones, especially in battery issue. A new technique proposed to reduce power consumption also to enhance the computations was introduced (Bahwaireth, et al., 2015).

Smartphones energy cost comparison for upload and download of large files such as a video from and to different multimedia cloud computing was proposed (Altamimi, et al., 2012). WIFI and 3G network interface together with HTTP internet protocol was used in this system. This system works to estimate the energy weight for the similar file used on the mobile device. The result has shown an improvement using MCC to preserve the energy of the smartphone device between '30% to 70%'.

Kumar & Lu (2010) proposed an energy model when mobile device performs computation and communication. The decision is made based on the energy cost of computation for a given task and mobile communication. The proposed model shows the impact of the communication on offloading, and explain that the offloading is useful especially in a heavy computation. In addition, they considered most of the applications used from a mobile device, for instance, voice recording and games as a high energy consumption. For this issue, they have proposed the cloud technique to avoid such high consumption

Mobile devices and cloudlet providers need an intermediate layer to help in taking decisions on where to offload the data, either to the cloudlet or the clone. Therefore, a new decision-making algorithm in the intermediate layer was proposed by Magurawalage, et al (2014). This algorithm has taken the power consumption into consideration to enhance the MCC performance, as well as other tasks to be considered such as responding to time and data caching procedure to the cloudlets.

Shiraz, et al.,(2015) have carried out some analysis for computational frameworks with regards to the MCC offloading. This research has investigated the recent studies of the latest computational offloading mechanisms considering the limited resource-intensive. This research has concluded that data sent for offloading with 13241.1 KB can consume approximately 31.6% extra energy, and 39% extra time.

Shiraz, et al., (2015) proposed an Energy-Efficient Computational Offloading model (EECOF) for processing heavy computation using MCC. The model addresses the issue of additional energy consumption in computational offloading by employing a distributed design and employing the lightweight procedure for reducing the overhead of runtime component offloading.

Albasir, et al., (2013) focused on their study to check the energy cost whilst web browsing. They figured out smartphones consume energy if the web page contains advertisements. Based on this result they proposed a client-server algorithm to save energy by controlling the web browsing contents. This means the server side should cope with the web page contents based on the request from the smartphone. The requests include network interface type and battery level.

JADE is a new system developed by Qian & Andresen (2015). This system works on improving the smartphone applications performance as well as reducing the energy consumption. It designed to reduce the workload by providing runtime technique to offer services for different categories such as; device profiling, computation offloading, program profile, and wireless communication. They have been able to prove that their system was capable to decrease the energy consumption to approximately 39%. The mobile device which offloads the tasks is called the client and the device that gets and performs the tasks is known as the server. Regarding to the mobile applications include transferable tasks, if the task executed on the client it is called local execution, otherwise, it is called remote execution. The JADE system was designed to be able to handle the offloading tasks within the Android device themselves and also between Android operating systems and non-Android operating systems. JADE is a multi-level offloading system which means that JADE is capable of offering an optimal method to transmit the loads to the server and depends on the performance and power consumption for optimum.

Altamimi, et al., (2015) proposed a new model to save the energy consumption for the smartphones. This system aims to have an assessment of the energy cost for the offloading tasks while using WLAN, 3G, and 4G. They have used five different brands of mobile phones as a way to validate their system, where they were able to show some result on how accurate their system can estimate the needed energy for offloading tasks.

Justino & Buyya, (2014) proposed an interface to support the allocation for the intensive tasks from smartphone applications which are deployed on Android platform to the CC by Aneka Cloud Application. ANEKA interface is a new mechanism on the mobile client library. They used the application performance time and the mobile battery life as two different mobile applications to validate their proposed mechanism. This mechanism works to minimise the process which can involve the sending and serialising of messages, desterilising and gathering the messages' responses for the communication that occurs in a mobile device to cloud.

2.7 Discussion of Related Studies

MCC is a good choice to extend the abilities of resources for smartphones because the new computing model has fostered a lot of applications to perform over the cloud. Offloading tasks from smartphones to the cloud is one of the methods used to enhance the computing for mobile devices and to extend the battery life. To make the offloading technique valuable, it is vital to estimate power consumption and processing time whilst completing the communication activities. So it is simply that if the offloading task uses less power consumption and time than processing tasks on the mobile itself, then the task should offload to the cloud.

Overall, from the previous studies Altamimi, et al., (2012), Justino & Buyya, (2014), Qian & Andresen, (2015), Kemp, et al., (2012), and Shiraz & Gani, (2014), it is noted that smartphones have limited resources, and this prevents them from providing an efficient processing system. In addition, the general consensus of the researchers is that battery power needs to be extended which means a reduction in power consumption is required. Different ways were proposed to reduce the power consumption and processing time, such as cloud computing, cloudlet, and virtual machine as this work has been completed by Shu, et al., (2013), Masiperiyannan, et al., (2014), Qian & Andresen, (2015), and Fesehaye, et al., (2012).

JADE is a new system developed by Qian & Andresen, (2015), one of the major areas focused on in this research was the categorisation of data by the operating system such as; Android and non-Android. This system works on improving the smartphone applications performance as well as reducing the energy consumption. They have been able to prove that their system was capable to decrease the energy consumption to approximately 39%. The JADE system was designed to be able to handle the offloading tasks within the Android device themselves and also between the Android operating system and non-Android operating system. JADE is a multi-level offloading which means that JADE is capable of offering an optimal method to transmit the loads to the optimal server depends on the performance and the power consumption.

The author who proposed JADE system he intended to extend his work by enhancing the model to include cloud platform. Also, he proposed the addition of a fast wireless link to the experiments such as 4G, for computation offloading. Furthermore, he planned in the future to investigate how cloud could benefit the mobile application.

Since it is not crucial to be concerned about the operating system of the phone there is no need to categorise data as Android and non-Android. Instead, the data can simply be classified as text (message and some stored documents) or multimedia (image or videos that are stored on phone or SD card). For example, if Mac or Windows phones are used then it is possible to use the same Gmail service platform easily through it. Also, if the mobile users save some documents then they can save it to Windows phone or Mac. There should be no concern raised about OS because the cloud already has facilities to keep the data compatible on VM or server OS. As a mobile user, there are no requirements to consider the OS compatibility as they are able to use the cloud services. In addition, in their study, the processing was completed by using a normal server to reduce the power consumption, however, tests such as these need to be completed by powerful and scalable servers such as a cloud server.

As discussed in 2.6, Altamimi, et al., (2012) introduced a new system which illustrated that MCC reduced power consumption significantly, although the author experimented on 3G and Wi-Fi, and concluded that there were limitations to the testing as the experiments were not run on newer protocols (4G). The implication of this is that there is insufficient knowledge of the impact of these factors when using a 4G network connection.

Another proposed model called CUCKOO which is created by Kemp, et al., (2012) was used by researchers to check whether to complete processing locally or remotely, in this context they deployed a dummy remote implementation. This highlighted some issues with the cloud, however, it was not a realistic scenario. Therefore, it would be appropriate to suggest that future work should be based on real cloud examples, to provide more accurate and reliable outcomes in terms of delay and power consumption when transferring files.

Additionally, in this model, they attempt to save mobile battery power by offloading to a powerful server. Within the model there is a Quick Response (QR) code address for each server, which is provided to the mobile user to scan with their smartphone camera, this enables them to connect to the chosen server address. Although this is an innovative idea there are some barriers that the mobile user may experience. For instance, if the mobile user has some issues with their camera function or a low battery they will be unable to use this function. In addition, this may also increase the power consumption used. It would be more advantageous if the mobile user is not involved in the connection between mobile and server. A preferred method would be for the mobile application to automatically choose an appropriate network interface.

Some offloading frameworks such as Active Service Migration (ASM) which is proposed by Shiraz & Gani, (2014) they completed the experiments using a virtual mobile device to delegate the computational task of the mobile application to offload it to the cloud server. This could cause additional pressure on the cloud server such as management and virtual device deployment as mentioned in Satyanarayanan, et al., (2009). It would be suitable if the mobile devices used are real and not virtual, also they should be connected to remote services in the cloud server which have powerful resources.

Areas of previous research that have been highlighted to require improvement in this research are:

- JADE system – categorised by the operating system and normal server was used
- Cuckoo System – dummy mobiles were used and the QR code is required to gain access to the server.
- ASM and MCE– virtual mobile devices were used.
- Altamimi 2012 – 4G network technology was not used.

The aforementioned points will be addressed in the experiments in chapters 4, 5 and 6.

Chapter 3 Research Methodology

3.1 Introduction

This chapter will present an overview of the methods used in the experiments. Tools and equipment will be discussed in detail. Including mobile, cloud, and network interfaces. The purpose of this chapter is to discuss and explain the research methodology of the whole research project and how the objectives will be achieved. Within this chapter, there will be justification of the method and the technological tools used to obtain the results based on the discussion of the literature review.

3.2 Research Quantitative Methodology

Research is a systematic and logical approach to investigate a certain subject to gain knowledge or understanding (Kumar, 2008). There are three fundamental elements of research; it must use a logical process to investigate the aims of the research, it should use a combination of thought, planning, and action and lastly that it should look at different approaches to the outcome or results of the research (DePoy & Gitlin, 2016).. Quantitative research is known as a positivist approach which analyses numerical data through locked measurements or methods.

Birley & Moreland, (1998) discussed the foundation of the study that uses a quantitative technique of research. Burns (2000) states that within a piece of quantitative research there should be a testing of a hypothesis, experimental tests during which different variables will be controlled to test the outcome, concise definitions of operational tasks that are carried out to prevent confusion and lastly a replication of any tests to ensure the results are reliable.

Moreover, it is suggested that researchers that use the positivist paradigm work diligently to forecast the outcome of their research based on current theories and models in line with cause and effect scenarios (Saunders, et al., 2009). Mostly, the positivist approach seen in quantitative research looks for connections and correlations within the statistical data that is produced and then compares and contrasts the data to see the impact that each variable has on the measured parameters (Monsen & Horn, 2008).

3.3 Methodology Approach of this Research

The methodology that will be used during the process of this research adopts a quantitative scientific approach of experimentation which coincides with the computer science research perspective and therefore aims to support the hypothesis through an evidenced-based approach. The experimental method is acknowledged to be strongly linked to a positivist approach, which is reductionist in nature. Although it is multifaceted, so it only permits a small number of parameters to be varied whilst controlling most factors (Bryman & Bell, (2011) Punch, (2005))

It is widely acknowledged that the outcome of experiments should be reliable, replicable and reproducible, Freitas & Jameson, (2012). Reproducibility is described as a longitudinal principle used to enable detailed experiments to be performed using a specific system or application and can be comparatively analysed on alternative systems or applications (Papadimitriou, et al., 2012). This will be seen during the progress of the experimental process of this research through the varied use of mobile devices and applications.

The offloading method to cloud computing requires more assessment with appreciation for the services which are offered to mobile devices, in this research the services are related to the duration of execution time and energy saving. Actually, saving mobile devices energy is essential for all services that are provided by cloud (Itani, et al., 2011). To examine whether or not mobile devices save energy with minimum delay by using the offloading technique, a large number of experiments were conducted on a real cloud. As discussed before some of the processing such as heavy calculation, reading and analysis of large excel files, face recognition, video, and gaming drain, mobile devices battery as well as the delay in processing time if it is executed locally on the mobile devices. On the other hand, the files exchange with the real cloud over the Internet, which means that sending, executions of tasks and receiving the results, consumes mobiles energy.

The following experiments in this research will investigate and measure the energy costs as well as processing time whilst offloading on various mobile devices.

3.4 Android Experiment Design

The three experiments will be carried out on Android platform. Android is defined as an open source platform and it is the official Integrated Development Environment (IDE) for the Android operating system, which supports several devices with different hardware abilities, features and sizes. Android applications are normally developed using Java programming language, C and C++. Android source code and SDKs are available for public, whereas, iOS is not open source. International Data Corporation (IDC, 2016) reports that Android has the major share of the global market, with almost 9 in 10 mobile devices running Android operation system. Grønli, et al., (2010) conclude that the Android platform provides a more efficient processing working environment due to stronger emulator support in comparison to other platforms development environment. These were major factors for the planning of the experiments as a high percentage of end users are using it, it is a stronger environment and therefore, this research will be more beneficial to a larger consumer basis.

As seen in previous research by Qian & Andresen, (2015) Android has been used to conduct several experiments related to mobile cloud computing. However, as discussed in chapter two section 2.7 there was a weakness in using the method of processing files without consideration of available resources of the cloud server.

3.5 Mobile Cloud Application

To process files on the mobile device and interact with the cloud server, there are two types of processing. The following section explains them in more details:

• **Deploy mobile applications:** These types of applications are developed using the mobile platform and supported by programming languages such as IOS, and Android, this is discussed in section 3.4. This type of application is similar to the application which will be proposed in this research to process the files. After developing the application, it should be installed on the smartphone to start activities such as, send or receive requests to and from the cloud server.

• **Embedded browser applications:** These kinds of applications are already built in the smartphone and developed by using standard web development languages such as JavaScript and HTML.

A mobile cloud application is an application that includes a set of main functionalities which could be executed independently on either a smartphone device or on a cloud server. This means the application is eligible to execute on both sides. The offload decision is regularly taken based on different parameters, in this research the decision is taken in terms of battery power consumption and processing time. One of the most important issues in the mobile applications is to act as the interface of the system and it should be easy to use and understand.

MCC enhances the smartphone application as a varied range of possible mobile cloud applications. MCC offers the mobile application with massive computational power such as real-time application response, image processing, and complex calculation, additionally, connects the multiple users to the same application in the same time and the sends results if required.

3.6 Research Design

According to the objectives which are set out in the introduction chapter section 1.4. The main objective is to investigate the impact on processing time and power consumption whilst using the mobile cloud computing. As discussed in the introduction, Lauridsen, et al., (2013) stated that these are the biggest challenges in mobile usage. Therefore, this will be analysed through comparative experiments between the cloud server and mobile device by using a developed application which will be illustrated in chapter 4, 5, and 6. Furthermore, as discussed by Wang et al., (2016) merging the mobile with the cloud will facilitate the processing for the mobiles in a more streamlined way and reduce the battery consumption. Using this new technology and through measurement of the two parameters; battery level and processing time, the expectation is that the files will be handled and processed more efficiently, ultimately supporting the mobile users. The experiments will be conducted using various different file sizes by increasing gradually and this will give a broader observation of the changes in the scenarios.

The experiments carried out in this study will also fulfil the criteria of quantitative research outset by Burns (2000). This hypothesis will be tested through concise experimentation using set parameters to verify the outcome of each experiment scenario fulfilling the aforementioned criterion explained by Freitas & Jameson, (2012) and Papadimitriou, et al., (2012).

The aim of this research is to establish whether it is more viable to process files using the cloud. These experiments will be completed using the created application on an Android mobile device and using Comma Separated Value (CSV) files of various sizes (1 Mb, 3 Mb, 5 Mb, 7Mb, and 9 Mb) in experiment one and file size 1 Mb, 3 Mb, 5Mb, 10Mb 15 Mb, 20 Mb, and 40Mb in experiments 2 and 3. For each main experiment there will be a specific number of scenarios that will be repeated to ensure that an average result can be obtained for both processing time and power consumption. The details of the methodology of each individual experiment will be explained further within chapters 4, 5 and 6. At the end of each experiment there will be a full analysis of the results and a conclusion is established from the analysis of these results. This will allow for a decision pathway to be selected and for more concise and direct research to be completed, enabling a very clear and structured evaluation of the hypothesis of this research.

3.7 Experimental Setup Tools

In the experiments, the mobile application will be deployed using Android Studio, because the program it helps to build the high quality of mobile applications for Android devices and tablets.

Recently, the high percentage of smartphones are based on the Android operating system, such as, Samsung, HTC, and LG mobile device. For these reasons, the experiments were conducted using Android Studio.

Various mobile devices with high specifications will be used to allow the results to be analysed and reflected across the wider spectrum of devices. Samsung S4, HTC10, Samsung S7, and Samsung tab4, all run by using Android operation system. In this case, the mobile represents a client side.

In contrast, as stated by Google Cloud Platform (2017) the server side is represented by the Google cloud platform which is used to perform the tasks for the following different reasons:

1. Server-less: new model helps in different parts:

I. Data processing
II. Database Management
III. Infrastructure Cost Reductions
IV. Operation Cost Reduction

2. Future-proof infrastructure: due to the following reasons

I. Custom machine types: Configures the right combination of virtual CPU and memory for the workload.

II. Sub-second Archive Store: data is available and provides a high throughput for the prompt restoration of the data.
III. Global Network of Data Centre.
IV. Faster Boot Times: Serves the application capacity very rapidly.

3. Seriously powerful data: due to

I. Context-rich media: use huge amounts of data to deliver context-rich experience
II. Machine intelligence: deep learning

Google platform provides SaaS and Paas via Google Applications, as they were discussed in more details in chapter 2. The application engine provides the architecture which helps Google application and promises transparent scalability by offering a financial model pay –per- use, as well as offering a data storage. The Google cloud platform server specifications are; high memory machine type with 8 virtual CPUs and 52 GB of memory, 8 virtual CPU, 52GB memory, and 64TB HDD.

Until this stage, client and server parts are known. To get access to the cloud from the mobile device when deciding to send the tasks. The internet connection needs to be established and the only way to create it, is by using different type of network interfaces such as, WLAN, 3G, and 4G technologies which will be explained in section 3.8. The new generation of smartphones are equipped with an environment to run the previous technologies.

To start running the experiments it is crucially important to deploy the mobile Android application using Android Studio as mentioned in section 3.4. The deployed application should be installed on the mobile to use it in the scenarios. This application will contain two options regarding to the location processing; one is local processing, second is cloud processing. Additionally, there is a feature called Cloud Zipped, the purpose of it, is to compress the file before offloading it to the cloud to support and focus on the quality of service such as delay. Furthermore information will be supplied in chapter 5 section 5.9. The diagram 3.2 below shows the MCC structure and the deploying of the application then install it on the mobile:

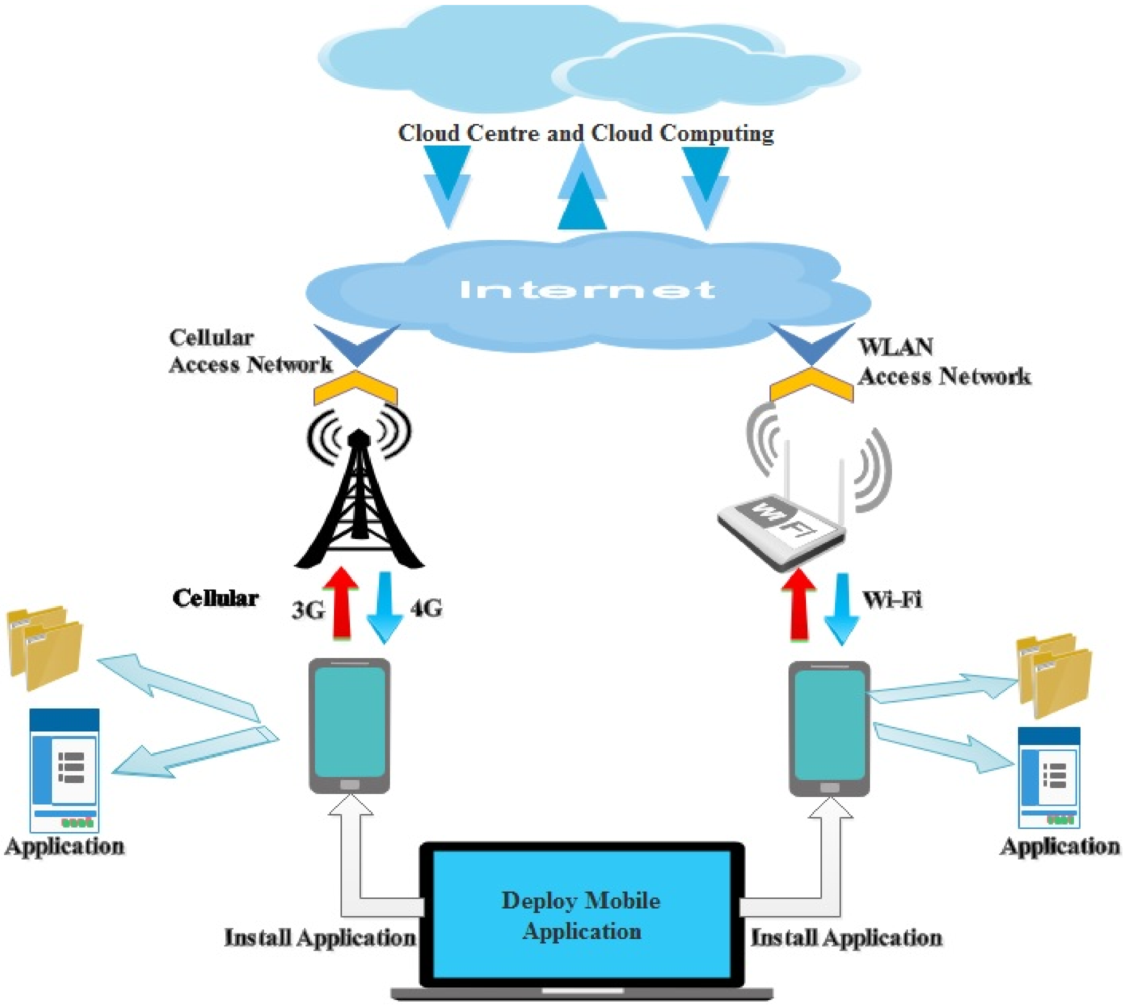


Figure 3. 1 Proposed system model architecture

3.8 Network Interfaces

The network interface is defined as a medium between two devices to help in creating the connection between them. After the connection is created, the devices start doing the activities such as sends and receive files between them. In this research, the activities are to offload and process the tasks from client's side (which are smartphone users) to Google cloud server. Once the results are ready, the server side sends the results will be sent back over the Internet to the clients. However, for each network interface there is a particular amount of power consumption and time to process files.

As mentioned in the previous section, the mobile device should create access to the Internet whilst it offloads tasks to Google cloud server by Network Interfaces (NI). NI is a component which contains the needed hardware physically to create a connection between the computing device and the public or private network. The network interface could be Ethernet or wireless such as WLAN, 3G, and 4G. Every smartphone is equipped with these technologies, so through these technologies it is possible to get the connection and start doing activities. In this research, the activities are to offload tasks to cloud computing and then record the results.

Xiao, et al., (2008) discussed the importance of mobile network interfaces on Nokia S60 and the effect on power consumption using YouTube as a case study. The results show that 3G technology consumes energy higher than WLAN, which is nearly 1.45 times. Further research by Albasir, et al., (2013) analysed the energy cost whilst web browsing based on the network type. Their research highlighted the importance of network interfaces on the energy consumption. These studies illustrated that it is vital to understand the network type before offloading tasks to the cloud server to save mobile energy and time.

The difference between the proposed mobile application architecture and the above-mentioned work is the deployed application which considers the offloading decision. It does this based on the file size by taking into account the network interface that is used during the offloading. This is due to the fact that each network interface has a different level of consuming power which has the impact on the mobile battery. This is mentioned in the previous chapter and published in earlier work (Aldmour, et al., 2016).

These are the types of network interfaces that will be used in this research:

1- **3G:** Third generation of the mobile broadband Internet. The average mobile broadband download speed is 6.1Mbit/s, and 1.6Mbit/s for upload.
2- **4G**: Fourth generation of the mobile broadband Internet. The average mobile broadband download speed is 15.1Mbit/s, and 12.4Mbit/s for upload.
3- **Wi-Fi**: the average speed in the whole scenarios is 20Mbit/s.

These will be discussed in relation to the experiments throughout chapters 4, 5, and 6.

3.9 File Attribute and File Size

Different kinds of files consume more power whilst processing, such as text, image, and video. In this research, the file type that will be used is text file format called CSV in different file sizes. This allows data to be saved in a tabular structure to support the processing. CSV files could be used in different application, such as Open Office Calc and Microsoft Excel spreadsheet. In the experiments, the same file type is used to have the same processing procedures for comparative analysis.

File size is the measure of space a file takes on a storage medium such as a computer hard drive and External memory File sizes can be measured in different units such as; bytes (B), Kilobytes (KB), Megabytes (MB), Gigabytes (GB) Terabytes (TB). In the following experiments presented in chapters 4, 5, and 6, the whole file sizes are in Megabytes (MB). The sizes are 1 Mb, 3 Mb, 5 Mb,9 MB, 10 Mb, 15 Mb, 20 Mb, and 40 Mb.

3.10 The Measured Parameters

Many parameters are important while processing files, especially when offloading over network interfaces. Based on the findings presented in chapter 2 section 2.7 power consumption and time are the notorious weaknesses in mobile devices. This insight provided a base to initiate the most important parameters for this research in terms of smartphones and their challenges. During the processing of files, whether it is locally or remotely, the quality of service and mobiles limited resources are very important factors in terms of the previous parameters.

In the experiments which will be taken place in chapter 4, 5, and 6, minimum delay and power consumption will be investigated. These will be measured in the following way:

1- **Battery Usage:** The amount of power which is consumed to read, offload, and executes files and get the results from the cloud server over the Internet, but in the local processing, the power consumption is only for reading and processing tasks, as it is conducted on the mobile, so no need for other activities such as send and receive. Battery usage is expressed in milliwatts (mW). To check the accurate power consumption, Zhang, et al., (2010) proposed an application called PowerTutor which is more efficient than other power models and it can be used on a broad range of mobile devices that are using Android platform. Additionally, supports to check the power including GPS, audio, CPU, cellular, and Wi-Fi communication components. PowerTutor was deemed the appropriate tool to support the measuring of the power consumption within this research. The PowerTutor is applied in the smartphones to be able to specify the mobile battery usage for each service in the smartphones. Another method used to measure the power consumption called Android Command will be discussed in detail in chapter 6.

2- **Processing Time:** is the time it required to complete the prescribed procedure. In the following experiments the procedures are offloading the file from mobile to Google cloud, and to get the results back from the cloud server over the Internet. The results time are measured in seconds (s). In this research, processing time will be measured through the use of the proposed model.

3.11 Mobile Android Application Handling Tasks

For processing heavy task computations, the offloading technique is an effective way to reduce the power consumption and to minimise delay. However, it requires further efforts and skills to develop mobile applications that included computation offloading capabilities. In this research Mobile Energy Cloud Computing Application (MECCA) will be designed to investigate the following reasons as discussed in the chapter 1 section 1.6:

- Processing time and mobile battery usage of the mobile devices will be estimated by recording the initial reading. After processing the second reading, will be taken which will illustrate the usage for both parameters, which will be tested in chapter 4, 5, and 6.
- Furthermore, according to processing on the Google cloud, a more in-depth investigation will be examined in chapter 5 through the comparative study of the impact on network interfaces such as WLAN, 3G, and 4G. A mobile application called PowerTutor, will be used to help in measuring the power

consumption before and after processing. For processing time a function will be used in the deployed application to record both readings.

- In addition, within the chapters, a decision engine will be introduced to support in offloading file based on the file size in terms of network medium, 3G, 4G, and Wi-Fi.
- The mobile application provides an easy-to-use mobile application for users to help users to extend battery life and minimise delay during processing the heavy computations based on the uploading technique.

3.12 Experimental Scenarios

There are different types of processing files scenarios, the first one is on the mobile device itself which means ‘locally’ processing. In the second scenario, the processing files will be on the cloud server, this means ‘remotely’ processing. For this type of processing there are different types of scenarios, for each cloud scenario, the connection needs to be established to get access to it. The scenarios will study the processing time and the power consumption for both locations. The following scenarios explain each procedure:

1- **Mobile Scenario**: In the mobile scenarios, the mobile device reads the file which is located in the mobile memory, and the task processing takes place locally on the mobile. This is the usual method where there is no Internet connection needed for offloading tasks. At the end, the mobile will get the results of processing the files, so the power consumption and processing time are known.

2- **Cloud scenario**: In the Google cloud scenario, where the task execution takes place on the cloud server, which means there will be an offloading task and

Internet connection will be required. The file is available locally on the mobile. In this case, the mobile has to establish the internet connection by using different types of network interfaces, either through cellular network or Wi-Fi to upload the selected file from the mobile device to the cloud. Then, the cloud starts processing the offloaded file. The processing time and the results of the processing file will be shown on the mobile device screen. As discussed in section 3.10, power tutor will be used to measure the mobile battery usage.

3.13 Offloading Decision Scenario on the Smartphones

As previously mentioned, offloading is a critical method due to the benefits which are provided to mobile devices. To clarify this issue, if a mobile should execute a task where the file is located on the mobile itself, there are two scenarios to perform the task: either perform the task locally on the mobile (S1) or send the task to the cloud (S2). Assume the mobile device consumes the power which is equal to Energy, E (S1) while the task is performed on the smartphone. Also, assume the mobile device spends power equal to E (S2) while offloading to the cloud, which includes reading and uploading the file then sending it to the cloud. In this issue, offloading technique is valuable only if E (S2) < E (S1). For offloading to be valuable, the power load of offloading for a particular task must be calculated to measure it with power level of performing the task on the mobile. From the smartphone perspective, the power consumed through task offloading is mostly because of the networking activities. The main issue in the experiments is creating models to check and estimate the cost of offloading which occurred by network connections activities.

3.14 Data Analysis Technique

In this research the data will be obtained from the deployed mobile application, through offloading the files which are available on the mobile device. Two types of data will be generated and then illustrate them in figures and tables. These results will be comparatively analysed with consideration to file size and the network interface, during processing files local and whilst offloading them to cloud.

4 Chapter 4 Experimental Evaluating the Offloading Power Consumption and the Delay Cost

In this chapter, the evaluation study was prepared to consider the possibility of the offloading method to save the mobile device battery. The methodology and experiments have been explained and at the end there is a discussion of the results for each scenario within the experiment. The results of the experiments were conducted for two parts: mobile and cloud. Furthermore, the results were analysed based on two parameters; the processing time and the power consumption to execute the file. The subsequent sections provide the details. The contributions of this chapter are published in (Aldmour, et al., 2016).

4.1 Motivation

The purpose of this chapter is to investigate the possibility of the offloading methods to save the smartphones batteries and to minimise offloading delay. The experiments were executed to examine whether a smartphone preserves battery by offloading tasks to the cloud computing using the mobile Android application which will be installed on the smartphone. The results will support researcher's insight about performing tasks on the node itself or by offloading to the core cloud, to know the processing time and the power consumption for each file. This is important for the examination of offloading method by the Android application. In fact, the experiments of executed files in this chapter could help the developers of CC to propose an efficient offloading paradigm to save energy for smartphones.

The processes and parameters which will be addressed in this chapter are detailed below:

- An intensive number of experiments will be conducted to check whether or not mobile devices save battery by using cloud computing. These experiments are performed by estimating the power and time costs for sending and receiving files to and from the CC, through WLAN connections. Then, performed using the same files on mobile devices in terms of power and time. The results will be compared with the previous experiments which were conducted on CC.
- Processing time for various file sizes will be calculated for the mobile device and cloud.
- Power consumption for processing files locally on the mobile and on the cloud will be measured.

The above-mentioned investigations will be considered for two scenarios related to the locations which are locally on the mobile device and on the cloud.

4.2 Experimental Tools

A comparison review of power consumption and processing time for computations of the files on mobile and on the cloud in this chapter.

In these experiments, the mobile application will be deployed as mentioned in chapter 3 in section 3.11. A new mobile device is used in the experiments because it has good specifications. Additionally, it was commonly used among the mobile users. The mobile device is Samsung Galaxy S4, Octa-core CPU, 2GB RAM, and Android OS, v4.2.2. As discussed in section 3.7 the Google cloud platform is used for computing and hosting the cloud side of testing. To commence processing the files, the smartphone should gain access through WLAN network to the cloud and then begin processing activities which are to send and receive.

4.2.1 Experiment 1 File Attributes

To check the battery level and delay on the smartphones, a specific file type is needed which is applicable for different kinds of processing such as calculation of power and delay. As mentioned in the methodology chapter section 3.9 CSV file will be considered in this chapter. According to file size, this is discussed in the same section, but for this chapter, the chosen file sizes are; 1 Mb, 3 Mb, 5 Mb, 7 Mb, and 9 Mb.

4.2.2 Experiment 1 Parameters Measured

In relation to processing files, whether it is locally or remotely tested, quality of service and mobiles limited resources are very important issues during the computation of files. In this chapter, minimum delay and power consumption will be investigated. Further details about these parameters were described in chapter 3 section 3.10.

4.2.3 Experiment 1 Networks Interfaces

The medium between two devices is very important which helps to establish the Internet connection. In these experiments, Wi-Fi which is discussed in chapter 3 section 3.8 in more detail, will be used in these experiments to offload the files from the smartphones to the Google cloud and receive the results of processing back to the mobile device.

4.3 Proposed Offloading Application Architecture

To understand the offloading technique, the developed architecture application in figure 4.1, explains it in more details. The part which offloads tasks is called the client, and in this case, this part normally is the mobile device. The part which receives and performs the tasks is called server. The deployed mobile application contains remotable tasks to send them to cloud server. If the tasks are performed on the client part which means not offloaded tasks; this is called local execution. However, if the tasks are performed on cloud server side, this is called remote tasks.

To assist the offloading tasks, an offloading computation technique is required to do some important functions such as communication, profiling, and optimisation:

4.3.1 Communication

To offload files from client to server, the application should be capable to establish the connection between the mobile and the cloud which is essential for both. Otherwise, the connection adheres to the remote function if the signal is cut off.

4.3.2 Profiling

Full information is required from both sides. According to the cloud server, the server name and location should be known, this helps to make a correct offloading decision. The details from the mobile side are required, such as network interface and file size.

4.3.3 Optimisation

The deployed application should be able to estimate the power consumption and the processing time based on the network interface type and the file size.

4.4 Application Architecture

The process of the application is to select the file from the mobile gallery by pressing the 'browse' button, then the mobile user selects the file and reads it. In this case, the file is ready for the processing either locally or on the cloud. If the processing takes place locally on the mobile itself, the file starts performing, and then the results are obtained. If the location for processing is on the cloud, the client should create the connection by selecting Wi-Fi, and then select the required file from the mobile to the cloud. After that, the results will be measured, and then sent back to mobile over the internet, as mentioned earlier the results will be provided in terms of processing time and power consumption. Figure 4.1 below illustrates the application architecture:

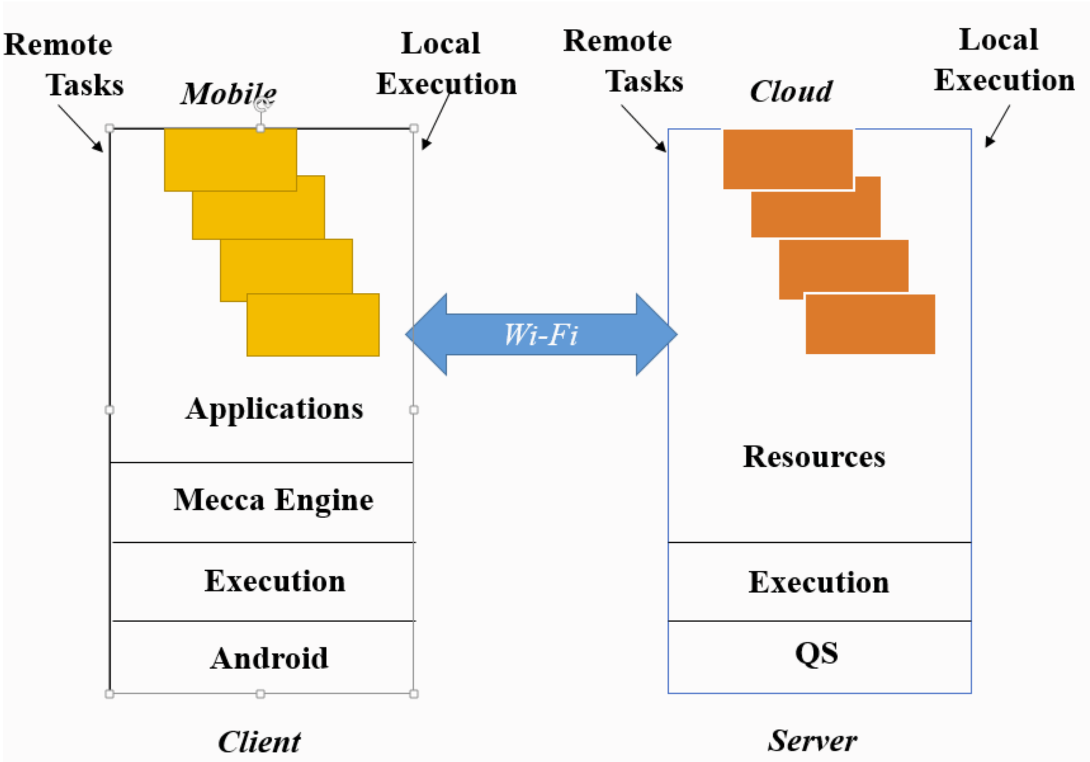


Figure 4. 1 Mecca application architecture enables offloading for smartphones

4.5 Mobile Application Development Frameworks Layers (MADFL)

The mobile applications development framework can help to understand the system architecture. MADFL's five layers consist of cloud, offloading, communication, data/information, applications, and presentation layers. The layered framework collects all of the elements together, such as software architecture, communication infrastructure and information which launches the mobile application and starts processing. In addition, it enables user interfaces and application programming to design a suitable mobile architecture.

Table 4. 1 The five layers of the mobile applications development framework

Presentation	 Mobile User Interface and Smartphone	Smartphones, Smart laptops
Application		Different kinds of applications, Calculation, Gaming, Face recognition
Data/Information		Text, Multimedia information
Connection		Medium to connect mobile to Internet. Wireless Area Network (WAN), Cellular 3G, 4G.
Goole Cloud Server		The place of providing services such as calculation, gaming, and GPS

4.5.1 Google Cloud

Cloud Platform contains a set of physical resources, such as computers, software, hard disk and virtual machines (VMs). These resources offer different kinds of services, such as computing, networking, storage and database. According to the offloading task which is under computing service, the mobile application will handle the tasks in it, by launching the application and offload the files from the database which are saved on the mobile memory card.

4.5.2 Communication

Mobile network resources provide the essential communication infrastructure for smartphone applications. The new smartphones are equipped with network resources such as Wi-Fi, Bluetooth, and cellular, for example, 3G and 4G. These resources help smartphones to run applications over the Internet. The deployed application offers the end user different options to process the file, which is processed locally or offloaded on the cloud using Wi-Fi.

4.5.3 Data and Information

Mobile applications could deal with varied types of data such as multimedia and text. The data is located in the mobile storage. The deployed application allows the end user to browse files and select the data which then starts processing it locally or on the cloud.

4.5.4 Applications

The application is a set of rules and processes developed based on the clients or enterprise needs. Additionally, it is designed to execute a set of coordinated tasks for the benefit of the end user, such as Safari, WhatsApp, and Skype. In this layer, this application is deployed based on different parameters which are power consumption, processing time, and file size to aid the end users to choose the most efficient method to save the battery, in addition, it will also consider the quickest processing technique.

4.5.5 Presentation

Presenting information on smartphones to the end user, and offers appropriate navigation system is the essential function in developing the mobile in this layer. The deployed application is agile which gives the end user the ability to select the location of processing and also to choose the medium connection such as 4G or Wi-Fi.

4.5.6 Communication Strategy

Accessing the Internet on the mobile phone is widely available nowadays, and it is very straightforward because mobile devices are already equipped with technology that helps to support this issue. This new technology is different among smartphone devices, such as some technology supports new network cellular generation for example 4G. In offloading scenarios, as stated the tasks could be sent to the core server either via Wi-Fi or cellular network data. According to processing files on the mobile, there is not any kind of communication required, because the processing would be local on mobile and there is no need to connect to the cloud server.

4.6 The Proposed Mobile Application Processing

The deployed mobile application is a set of functions and processes, which help to compute the files. In the proposed mobile application, two locations are considered, one performs tasks on the mobile itself without any kind of network connection. Second, it performs tasks on cloud connecting through Wi-Fi. Additionally, Figure 4.2 below shows the structure for the mobile application.

The deployed mobile application using Android Studio is installed on the mobile device to start performing tasks for both locations, which are mentioned in the previous paragraph. To perform tasks locally on the mobile device, in this scenario, the installed mobile application allows browsing files which are located locally on Secure Digital Card (SD card). The file which is required to be processed should be selected from the list. For the selected files, the processing commences, but before that readings for the battery level and time are recorded. Once the processing has finished, the new readings for the battery and the processing time are known and will be compared with the initial readings. At the end, the results are published on the mobile screen. At this point, the battery level and time for calculation of the selected file is independent of the network interface, which means that there was no network connection because the processing was local on the mobile.

From figure 4.2 the functions and the processes known as program transcript are installed on the mobile device, this is because the performing the tasks cannot be executed without it.

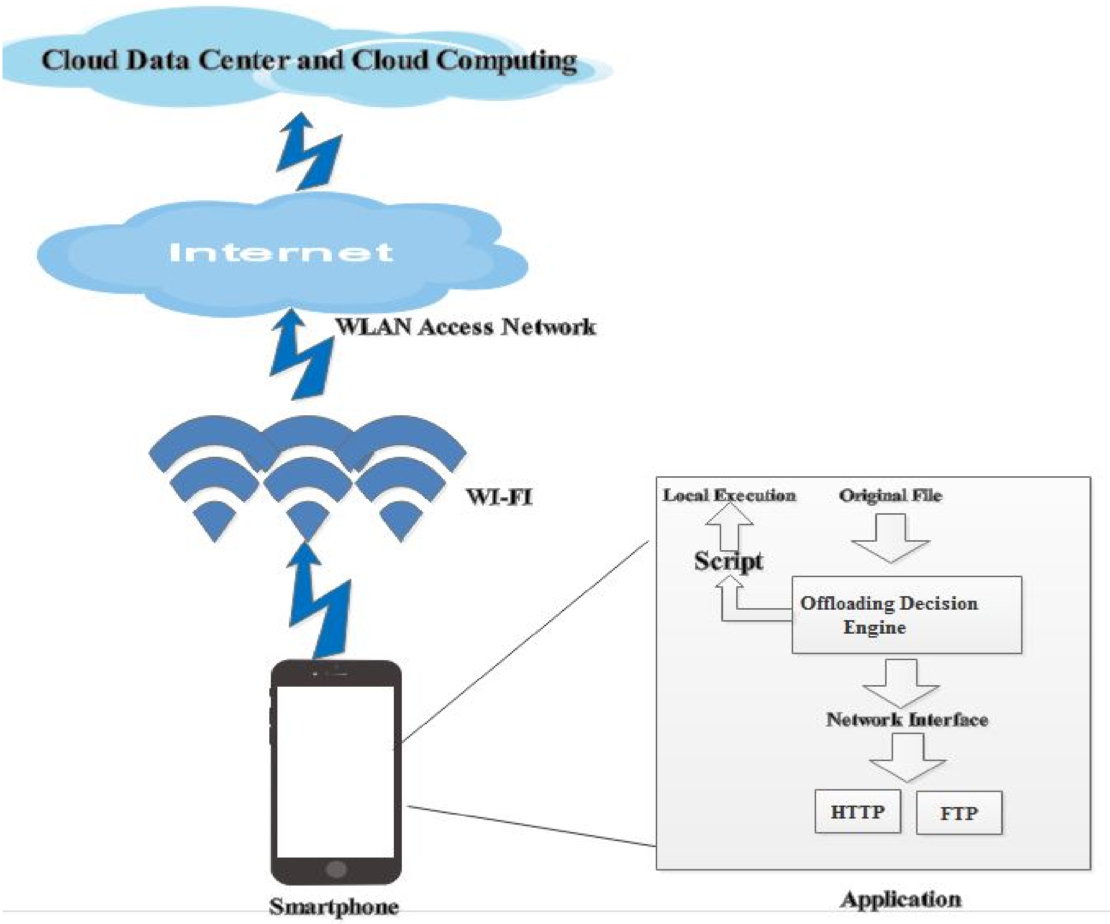


Figure 4. 2 Encoding scenarios where the original file on the mobile

4.7 Experimental Scenarios

In this section, ninety experiments were conducted to measure the actual processing time and power consumption by using a real smartphone and cloud to process the files. This means that the computations should be conducted on both sides. Table 4.2 shows the scenarios for processing files.

Table 4. 2 Local and offloading scenarios

Scenario	Data	Processing	Network Activities
S1	Mobile	Local	No network connections

S2	Cloud	Cloud	Send tasks, download tasks, and output using Wi-Fi

Regarding to the second scenario, the network connection is needed to offload file to cloud as shown in figure 4.2. In addition, one server is reserved exclusively for these experiments, so inside the program transcript the server is known. To gain access to the cloud server, the mobile device has to connect through different kinds of networks such as 3G, 4G, and Wi-Fi. In this chapter, Wi-Fi is used as a network connection between the mobile device and the cloud. Before commencing the processing, the mobile user is required to switch on Wi-Fi button and connect to the wireless network Wi-Fi. The mobile user launches the application which is installed on the mobile device. The Browse button in the application allows the mobile user to select the file from the list which is saved in the mobile Secure Digital Card (SD card), it is then sent to the cloud server to start performing tasks.

As seen in the previous scenario, the readings for the battery level and time are recorded before and after processing the file. Once the processing tasks are finished, the results are sent back from the cloud server over the Internet through Wi-Fi. Figure 4.3 below shows the mobile application processes flow.

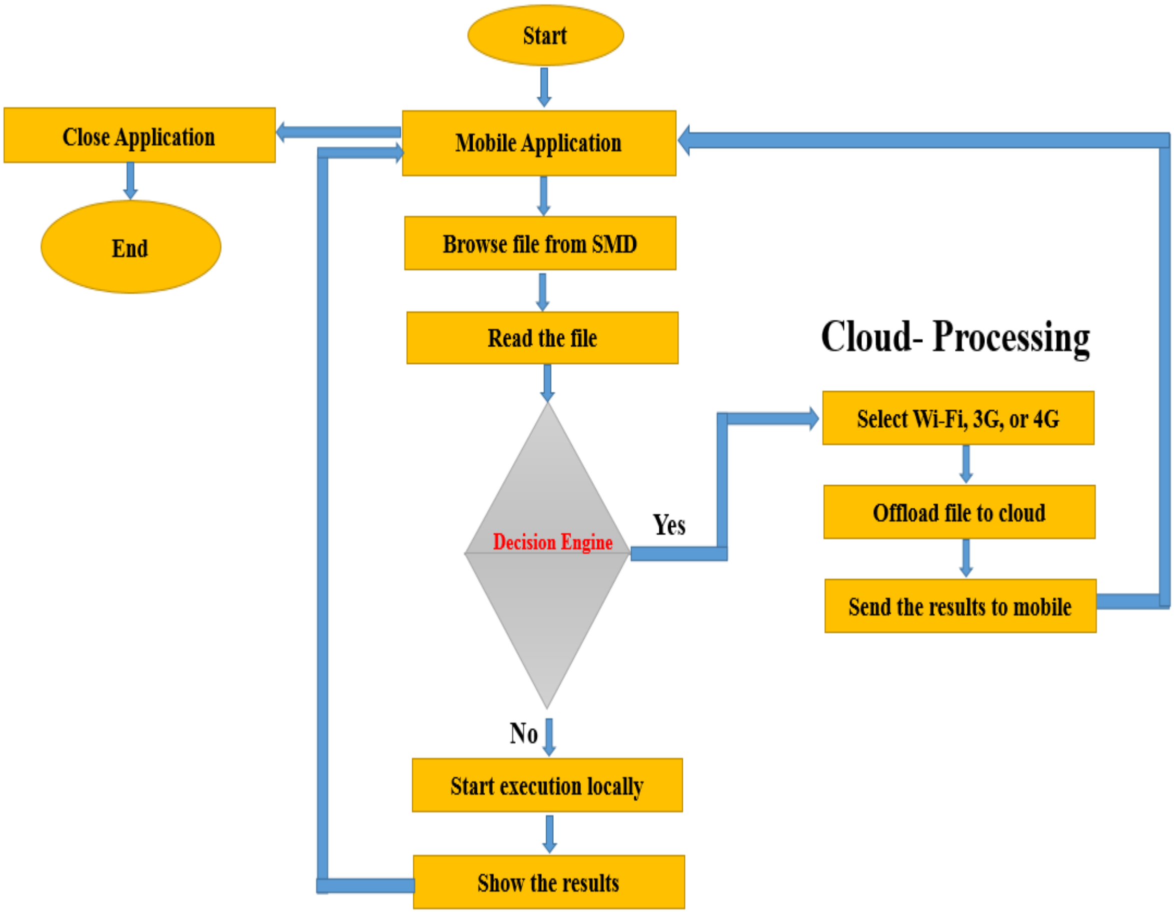


Figure 4. 3: Flowchart for the application processing

4.7.1 Proposed Algorithm

If (application-type=”text\doc\CSV”) {

If (file-size <=1mb) {

If (network-interface=4G)

Execute,

Else (if network-interface=3G)

Execute,

Else

Execute local

```
}
If (file-size >1mb and file-size <=3mb) {
        If (network-interface=4G)
        Execute
        Else
        Execute local
}
If (file-size>3mb and file–size<=40mb) {
        If (battery-level<10%) {
                If (network-interface=4G)
                Execute,
                Else if (network-interface=Wi-Fi
                Execute
                Else
                Execute local
                }
        Else
                {
                If (network-interface=Wi-Fi)
                Execute
                Else
                Execute local
                        }
```

```
        }

}
```

4.8 Experimental Results

Ninety experiments were conducted for both scenarios locally on the mobile device and remotely on the cloud server, for different files sizes, Table 4.3 shows the scenarios of processing the files:

Table 4. 3 scenarios and network activities

Scenarios	Location	Network interface	File size used
S1	Mobile	No network Connection	1 Mb
S2	Mobile	No network Connection	2 Mb
S4	Mobile	No network Connection	5 Mb
S5	Mobile	No network Connection	7 Mb
S6	Mobile	No network Connection	9 Mb
S7	Cloud	Wi-Fi	1 Mb
S8	Cloud	Wi-Fi	2 Mb
S9	Cloud	Wi-Fi	5 Mb
S10	Cloud	Wi-Fi	7 Mb
S11	Cloud	Wi-Fi	9 Mb

The above-mentioned experiments were conducted for a variety of file sizes which are: 1Mb, 2Mb, 5Mb, 7Mb, and 9Mb, on both locations of the mobile and the cloud.

As far as, the battery usage is concerned, this is the first stage of the experimentation to evaluate it on the smartphones and with the connection to a real cloud platform through the wireless network connection.

The experiments show the results as follows:

- **Power consumption:** is defined as electrical energy over the time supplied to run an electrical device, and this is discussed in detail in chapter 3 section 3.10. In these experiments power consumption is measured for the smartphone when different file sizes such as 1 Mb, 3 Mb and 7 Mb are executed, on the smartphone and on the cloud.

- **Processing Time:** is the period during processing the files locally or transferred to another location and obtain the results over the Internet, which is discussed in chapter 3 section 3.10. It is measured for different file sizes such as 1 Mb, 3 Mb and 7 Mb, which are executed on the node itself and by offloading to the core server which is CC.
- The investigation illustrated whether it was appropriate to execute locally or offloading by using CC. This investigation was completed by measuring the power consumption and processing time before executing the tasks. After that, they were compared to the results after execution.

4.8.1 Analysis of the Data Results

The results of the afore-mentioned investigation based on two experimental scenarios are illustrated in two figures 4.4 and 4.5 as seen below:

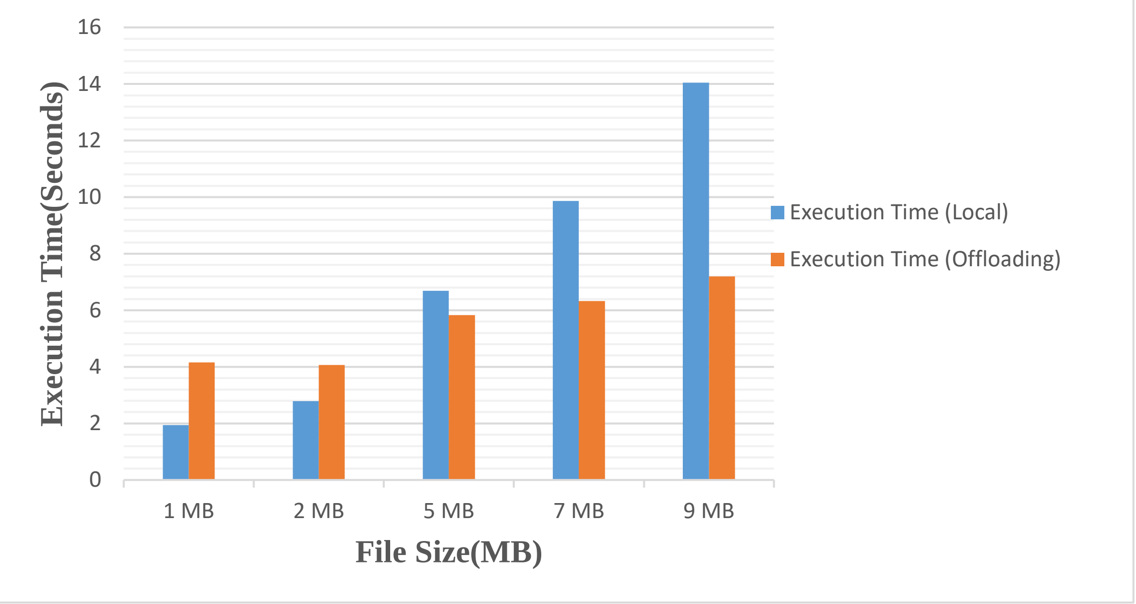


Figure 4. 4 Execution time for local and offloading:

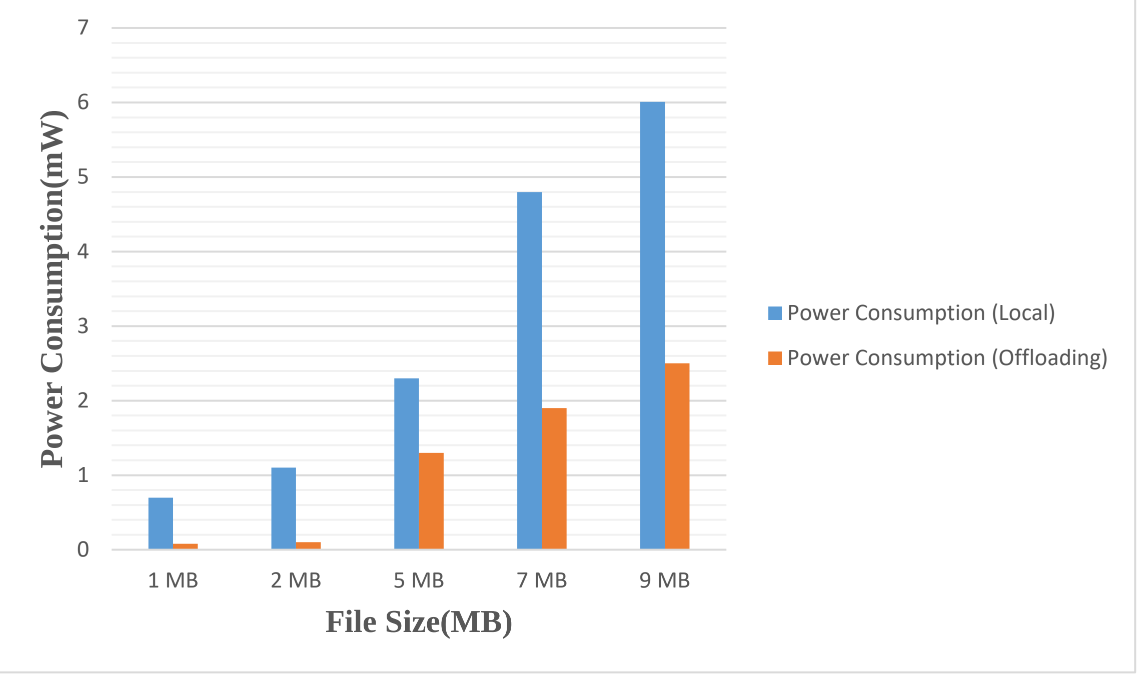


Figure 4. 5 Power consumption for local and offloading

- The first test that was completed for offloading the files as seen in figures 4.4. It has been studied to show the relation between the execution time and file size either locally or remotely. The results illustrated that the larger the file size, the higher the increase in processing time. These experiments evaluated the difference in file sizes ranging from 1 MB to 9 MB. It is noticeable from the results that an increase in the file size reflects a sharp increase in processing time for the mobile scenario. This differs from the cloud which reflects a slow and gradual increase in the processing time. For example, in the local scenario when the file size was 1 MB the processing time was 2 seconds and sharply increased to 14 seconds for 9 MB. Compared to the cloud which was 4 second for file size 1 MB and 7 seconds for 9 MB.
- In Figure 4.4 the execution time at file size 9 MB in the node is double compared to the cloud which is 7.2 seconds while on the node is almost 14.05 seconds.
- Figure 4.5 illustrates the results obtained for the power consumption for both mobile and cloud. The results correlate with those obtained for processing time as there is a sharp increase for local experiments and a gradual change for offloading on the cloud. For instance, at file size 7 MB the power consumption for the node is more than double compared to the cloud.

This supports the hypothesis of this research so it is paramount to employ this algorithm to help in making the most efficient decision to preserve the mobile battery and the time.

Table 4.4 below also illustrates the results for processing files locally and remotely using Wi-Fi connection. As described previously in some files sizes, the difference in both parameters is more than double and these results reflect that using cloud computing to process files is much more efficient than processing files on the mobile device.

Table 4. 4 Processing time and power consumption for local and offloading using Wi-Fi

File Size (Mb)	Local Node		Offloading	
	Execution Time(Minutes)	Power Consumption (Mw)	Execution Time(Minutes)	Power Consumption (Mw)
1	1.94	.7	4.16	.8
2	2.79	1.1%	4.07	1%
5	6.69	2.3%	5.83	1.3%
7	9.87	4.8%	6.33	1.9%
9	14.05	6.01%	7.2	2.5%

4.9 Discussion of the Results

In this chapter, the mobile application was deployed to offload files from smartphones to cloud server. The expectation was that offloading method could save the mobile battery and minimum delay. Therefore, the experiments were completed to analyse these two parameters.

The mobile application was used to check the processing time and the power consumption whilst offloading the files. The experiments have included various file sizes and a comparison was performed. After each test, it can be decided either to process the file locally on the mobile itself or to process it by the offloading to the cloud server. From the results, it is notable that this is a big jump in the execution time and the power consumption for big files but it is noted that in these circumstances it is vital to execute files on the cloud core by offloading, for example at file size 9MB the execution time is doubled in the local load. There is, in this case a big reduction in the power consumption.

From the experiments which were conducted in this chapter, the results indicate that cloud saves the battery and the delay for the smart-devices, approximately half.

As only Wi-Fi connection was used in chapter 4, for further research experiments, it would be recommended to consider a various network interfaces, such as 3G, and 4G to enable offloading mechanism which supports processing files on the real cloud. In addition, it is important to find out the power consumption and time delay for each network interface during uploading files, and to evaluate the benefits of each of them. It is noticeable that file size has an impact on both parameters, which are power consumption and processing time. As this was an important factor, a wider scope of file sizes should be used in further experiments to obtain a larger view of the impact of this factor on the results. These ideas will be explored in the following chapter, 5.

From the previous experiments, the results showed that there is an enhancement in reducing the power consumption and processing time comparing to other studies. The results from this research for power consumption were nearly 61% saving, on the other hand the results were in (Qian & Andresen, 2015), (Fekete, et al., 2013), and (Altamimi, et al., 2012) 39%, 38%, and 30% to 70%, respectably, as shown in the Table 4.5.

Table 4. 5 Results for previous studies and the proposed mobile application.

Study	Power Consumption (mW)	Connection
Qian & Andresen, 2015	39%	Wi-Fi
(Altamimi, et al., 2012)	30%-70%	Wi-Fi
(Fekete, et al., 2013)	36%-38%	3G, Wi-Fi
MECCA (author)	61%	Wi-Fi

5 Chapter 5 Experimental Modelling Network Interface for Power Consumption and Time

This chapter will reflect on the experimentations and the results in the previous chapter, in terms of the power consumption and the processing time of smartphones. In addition, as part of this stage of the research, there will be an examination of the limitations which were found in the experiments, in chapter four.

By addressing these factors, it is expected that a more precise analysis can be completed. The parameters will be estimated locally and then compared to the efficiency of how the smartphone executes by offloading on different kinds of network interfaces such as; WLAN, 3G, and 4G to cloud. Different scenarios will be tested for different file sizes and wireless network interfaces. The main parameter of the quality of service is the time needed to execute the file on the cloud versus the time needed to execute the file locally on the smartphone by a mobile application. The following parts explain in more detail about the experiments.

5.1 Introduction

To estimate power consumption for offloading strategy, a model will be developed for 3G, 4G, and WLAN. The developed model is able to accurately estimate the power consumption and processing time whilst offloading tasks to the cloud. A large number of experiments will be conducted in different file sizes for the three network interfaces.

5.2 Second Experiment Outlines

For more accuracy in offloading decisions, the power consumption should be estimated for each file, to compare it with the energy cost whilst processing locally on the mobile. In terms of the consumed power in mobile devices during offloading files, reductions are mainly caused by the communications activities.

In this Chapter the contributions to knowledge are as follows:

1. Estimate the processing time whilst processing files locally and offloading to cloud as seen in chapter four, but with additional file sizes.
2. Offloading files using WLAN and 3G/4G network interfaces from mobile to cloud significantly effects the power consumption due to communications activities.
3. Deploy model to compress files to help in reducing processing time.
4. Decision engine to decide whether to offload to the cloud or not, based on the file size.

This can be examined by Mobile Energy Cloud Computing Application (MECCA).

5.3 Offloading Files over WLAN, 3G, and 4G Networks

The experiments were conducted using WLAN, 3G, and 4G whilst offloading from mobile to cloud which are already discussed in detail in chapter three section 3.10. Power consumption and processing time were measured for different file sizes. Wireless communication is used to connect between two destinations through the radio waves. This technology is improved by introducing new generations each time. Every generation of the technology is enhanced in the topology such as high speed and frequency of the bandwidth. Before a couple of years as we know 4G became the successor of 3G. 4G technology provides good additional features, for example, provide Internet broadband in a high speed and high definition which enables users to use Video conferencing, Mobile TV, and video calling.

There are two types of wireless communication, they are distinguished in the following:

- **Cellular networks:** Is a communication network which is distributed over areas called cells, each of them served by transceiver called base station, and this provides the coverage for transmission data or voice. The standards in this type of network are CDMA, GSM, and TDMA. Different types of generations in this type of networks such as 2G, 3G, and 4G.

- **Wireless Local Area Networks:** Is a wireless distribution method for two or more devices to use high-frequency radio waves and generally it has an access point to the Internet. There is a coverage area for WLAN which permits users to move around with, in areas such as a small office or home. The standardization of WLAN is IEEE 802.11.

5.3.1 File Sizes and Format for the Second Experiment

File size is paramount to the experiments in this research. As seen in chapter 4, the file sizes used in the initial experiments were 1Mb, 3Mb, 5Mb, 7Mb, and 9Mb. However, after reflection, it was noticeable that the battery level and processing time were affected by the increase in file size. Therefore, it was deemed necessary to complete further experimentation which included larger file sizes, so that this impact could be analysed in more details. In addition to the initial file sizes, there are new files that will be used in this chapter. The file sizes have been selected are 1Mb, 3Mb, 5Mb, 10Mb, 15Mb, 20Mb, and 40Mb. As mentioned in methodology chapter section 3.9, the file format that will be used for this experiment will be, CSV. The reason that CSV files were also selected for this stage of the experimental process is to obtain results that are reliable and to maintain continuity of the research data results. In addition, this file type is structured in tables which supports the researcher to complete data calculations such as reading, updating, deleting, and analysis.

5.3.2 Parameters Measured For the Second Experiment

To maintain continuity in this research this set of experiments will use the same parameters as those seen in chapter 4. Therefore, in these experiments processing time and power consumption will be investigated. These parameters can be seen in more detail in chapter 3 section 3.10, including the method of measuring the power consumption.

5.4 Proposed Mobile Application Architecture

The new innovative system helps smartphone users to offload the files to the cloud taking into account the file size, for instance, the file size will be calculated in advance. The created new paradigm includes a decision engine to check the possibility of the offloading files, as well as to preserve smartphones energy. The new paradigm checks the possibility of the smartphones to save battery and time by sending the tasks to the cloud. The basis work for this technique is based on the file size, the location of processing, and the network interface type, essentially it is to check two of the most important parameters. This can be completed by measuring the processing time and battery usage during the performance of tasks locally on the mobile, and offloading files from the mobile device to the cloud. Figure 5.1 illustrates the processes during offloading files from mobile device to cloud server.

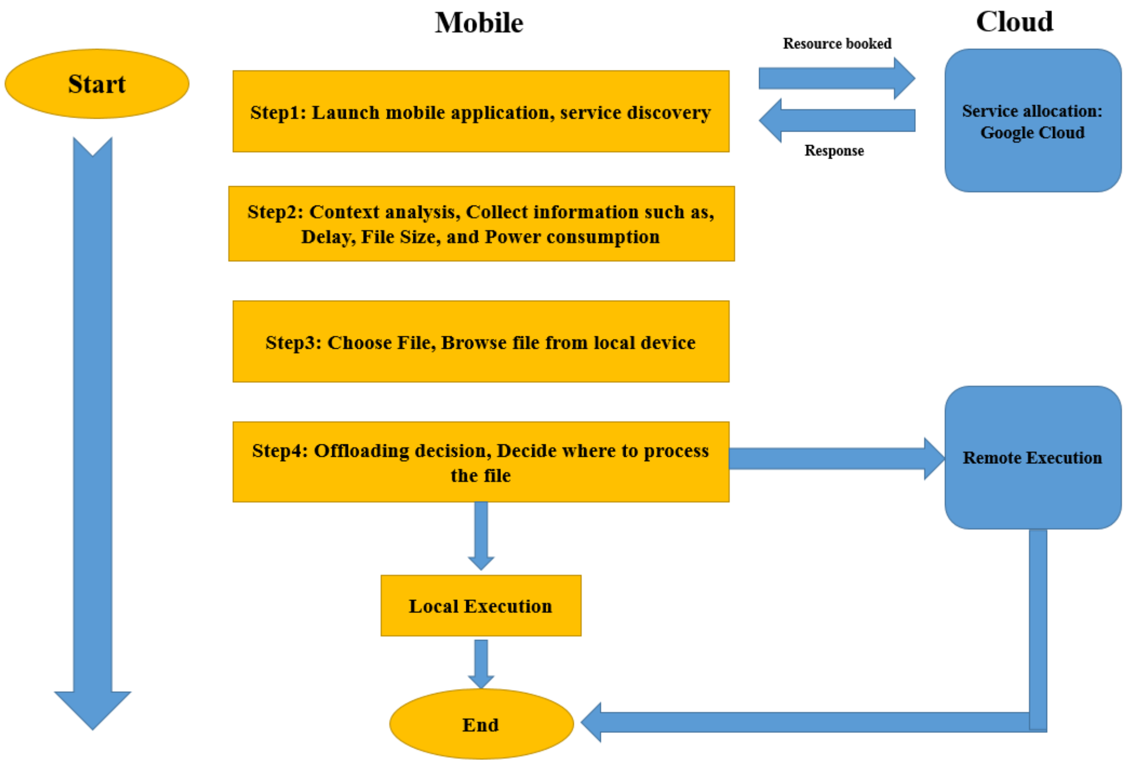


Figure 5. 1 Processes of application offloading to cloud

The smartphone offers a variety of mobile processing to the end users by using applications. On the other hand, Google cloud consists of the data centre, which is equipped with a pool of powerful resources and limitless hardware, such as software, infrastructure, and platform, this helps to provide end users for all the complex mobile computing in comparison to smartphones.

The proposed model consists of two parts, mobile device and cloud computing which is Google cloud, both are connected to the Internet as shown in Figure 5.2. In relation to the smartphone, there are two ways to gain access to the Internet, either through a cellular data such as 3G and 4G or a WLAN access point as discussed in section 5.5. However, the most important issue is in the route of the access and offloading files, because the power consumption varies among interface networks, and it is very important to know the energy cost of the upload and download of files through cloud computing when using different network interfaces.

The experiments were conducted in two experimental scenarios based on the location of processing files, as illustrated in figure 5.2. There are two scenarios to process the files, the first scenario is S1, where the task is processing on the mobile device itself, which means no need for the network connection. In this case, the processing of files is called local encoding. Mobile application browses the file then select the needed file to be processed. Meanwhile, the battery level and current time should be recorded before commencing the processing. The application starts to process the selected file and shows the results. After the processing is finished, the battery level and execution time are known and prepared to compare them with the initial reading. The second scenario is S2, processing files take place on the cloud server. This means, there are activities between the mobile device and cloud computing should be formed. Likewise, the mobile user chooses the file that they want to process, then the upload the selected file from mobile device to the cloud server, CC encoding starts to execute the file, once the processing is finished, the results are ready to be sent back to the mobile over the Internet. Battery level and time are recorded before offloading the file, and after processing the file, the new readings are known to be compared with the initial readings.

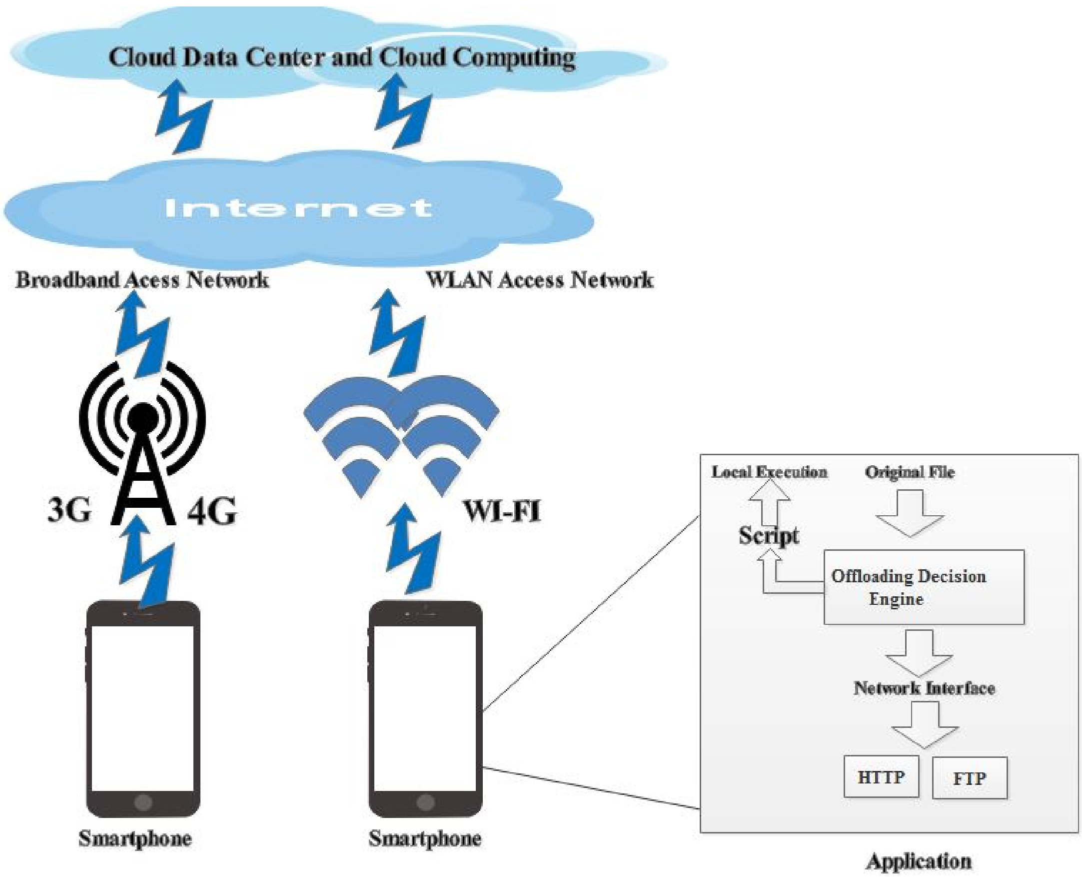


Figure 5. 2 Encoding scenarios offloading whilst using Wi-Fi, 3G, and 4G

To send and receive files through the cloud, the 3G, 4G, and WLAN communications are used to evaluate the battery usage and processing time on the cloud, and then the same file is performed locally on the smartphone. Figure 5.3 shows an overview of evaluating flow for different types of parameters that could effect on the delay and power consumption whilst processing files over the cloud.

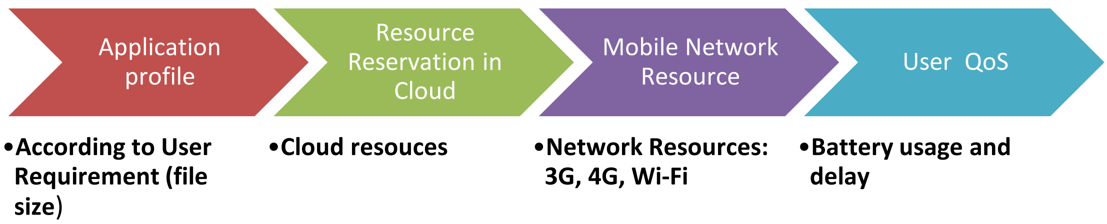


Figure 5. 3 Contextual Model and their corresponding parameters

After performing tasks on both locations, the results for the two parameters are known to compare. The decision engine chooses the efficient solution based on the file size in terms of power consumption and processing time. The previous processes are employed for all file sizes which were mentioned in section file attribute 5.3.1, and additionally for all network interfaces.

5.4.1 Decision Engine

The decision engine is implemented to find out the efficient decision for offloading. Two parameters help to take a decision which are the file size and network connection which will be measured in terms of power consumption and processing time. The two parameters as shown below:

- **File size:** after conducting the experiments for various file sizes, it is noticed there is an effect on the power consumption when increasing the size. This means it is important to take this in the account whilst offloading.
- **Connection Type:** based on the previous sections, and studies for each network interface there is an amount of power consumption. In this decision engine, it is valuable to consider the level of battery usage and link it to the file size.

After the appropriate decision has been taken, the tasks start execution either on the smartphone, or alternatively proceed on the cloud by offloading. This decision is based on the algorithm.

The aforementioned conditions which are file size and connection type can be seen in the following section within the proposed algorithm.

5.4.2 Proposed Algorithm

```
If (application-type=”text\doc\CSV”) {

    If (file-size <=1mb) {

        If (network-interface=4G)

        Execute,

        Else (if network-interface=3G)

        Execute,

        Else

        Execute local

    }

    If (file-size >1mb and file-size <=3mb) {

        If (network-interface=4G)

        Execute

        Else

        Execute local

    }

    If (file-size>3mb and file–size<=40mb) {

        If (battery-level<10%) {

            If (network-interface=4G)

            Execute,

            Else if (network-interface=Wi-Fi

            Execute

            Else
```

```
            Execute local
            }
        Else
            {
            If (network-interface=Wi-Fi)
            Execute
            Else
            Execute local
                }
        }
}
```

5.5 Compressed Files Model

Processing files in the mobile environment are very limited particularly for the recent rich applications because generally, mobile devices have limitations to their resources such as; processor, storage, and battery. This leads mobile devices to access through cellular to power resources to process heavy tasks. The increase in the high speed of Internet connections and at the same time, with reasonable rate through wireless and cellular technologies such as Digital Subscriber Line DSL, optical technologies, are enabling the shift from computing towards distributed computing. Mobile applications rely on data and physical resources which are available on the mobile itself. These innovative changes and progressions are creating historical developmental changes within telecommunications. A new era of computing is called distributing computing and recently use rented resources on demand or pay as you use. These powerful hardware and software's, enable mobile users to use them instead of doing the processing locally. CC is capable to run a complex computation and brings advantages through different types of services such as; PaaS platform as a service, IaaS infrastructure as a service, and SaaS software as a service.

Currently, mobile communication is growing rapidly. Users are able to interact with others in a short time. One of the most important issues during the transmission whilst sending data from sender to receiver is file size because more data demands more time (Patil, et al., 2014).

For the mobile users, the speed of communication is important, however, with the vast improvement in the telecommunications, file size is still one of the most crucial parameters that could increase the delay. To overcome this issue, compression of the file is one of the solutions. There are many benefits of compressing the files, the most widely recognised are to reduce storage requirements. Additionally, for data communications to help in increasing rate of transferring information. File compression can be performed on existing hardware such as smartphone by software for example Java. Figure 5.4 shows a simple data-compression diagram. Encoding scenarios where the original file is on the mobile.

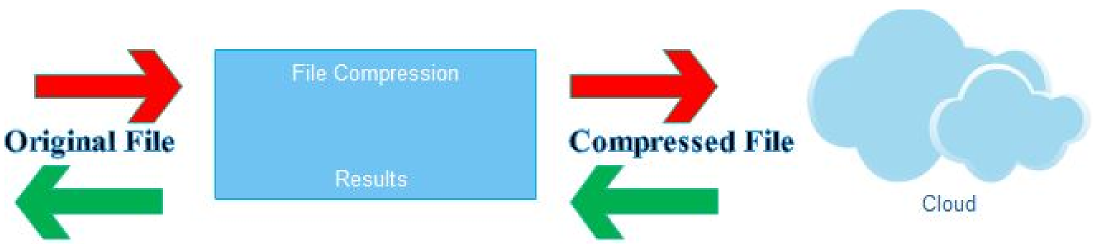


Figure 5. 4 File compression block diagram

In this research, the compressed model is proposed to reduce the processing time which means less delay, during offloading files from mobile device to cloud using 4G technology. The deployed mobile application compresses the file on the mobile before offloading it, then offloads it. In this research the benefit of the compressing is to reduce the processing time whilst processing files during the offloading process, this means minimum delay. The results will be shown in following sections.

5.6 Offloading Scenarios

As discussed in the previous chapters, the new media contains all different types of the data such as video, sound, text, and pictures. To execute the wide variety of media in a high performance, it requires a powerful resource for instance cloud computing. However, smartphones have limited functions. Mobile battery limitation is one of the biggest concern for the smartphones improvement. To overcome this issue, the new offloading technique is one of the best and proper ways for the power consumption for smartphones devices and battery life limitation. This novel method supports smartphones to execute the intensive tasks by forwarding them from the mobile to the CC.

The following experiments are extended to the earlier experiments to check the possibility of sending tasks to whether or not mobile devices could preserve energy and time by offloading to the cloud. In these experiments a large number of investigations conducted on the mobile device and real cloud with different files sizes to check two parameters which are, the processing time and battery usage in the whole scenarios, additionally with and without file compression technique.

There are different types of processing files scenario. In each scenario, the network interfaces such as Wi-Fi, 3G, and 4G were used in the scenarios to measure the accurate processing time and the battery level. Additionally, different file sizes were used in the entire experiments. The following scenarios explain each procedure:

1- **In the first scenario,** the mobile device reads the file which is located in the mobile gallery, and the task execution takes place locally on the mobile. This is the usual method where there is no offloading task.
2- **In the second scenario,** task execution takes place on the cloud which means there is an offloading task is going to happen. The file is available locally on the mobile. In this case, the mobile must upload the file to cloud by using different types of network interfaces. Then the cloud begins the execution task, once the execution is finished, the results send back to the mobile over the Internet.
3- **The third scenario,** compressed file. The smartphone compresses the file using Java, then upload it to the cloud using different types of network interfaces. Then the cloud starts doing the execution on the compressed file. After finishing the execution the mobile get the results back from cloud server over the Internet.

The previous scenarios were illustrated in figure 5.2 shows the proposed mobile application architecture for using network interfaces whilst offloading to cloud

5.7 Experiments Scenarios

A lot of experiments were conducted to evaluate the power consumption and required time on the real smartphone, while doing the offloading technique over the different kinds of network interfaces. The experiments were conducted from two perspectives regarding to the location of the tasks, which are on the smartphone itself and on the cloud. There are three scenarios regarding accessing cloud to offload tasks as discussed in the previous section. The figure below shows the topology for the mobile cloud computing MCC.

In the following experiments, large numbers of scenarios were performed for processing files locally and remotely in terms of power consumption and processing. From scenario S1 to scenario S7, the processing files were on the mobile itself. This means that there is no network connection used. From scenario S8 to scenario S14, the processing files was on the cloud. This means, mobile device connect to the cloud through different kinds of network interfaces. In these scenarios, there are three network interfaces, such as; 3G, 4G, and Wi-Fi. The experiments were conducted for each file three times. For example in scenario eight, the processing file for file size 1Mb, took place on the cloud, the mobile device connects to cloud through 3G, then 4G, and finally Wi-Fi. After the mobile is connected to the cloud, offloading of the file to cloud will start. Cloud starts processing the file then send the result back to mobile over the Internet.

The previous procedures are deployed to entire scenarios, Table 5.1 below shows the mentioned above:

Table 5. 1 Scenarios and network activities

Scenarios	Location	Network Interface	File Size
S1	Mobile	No network Connection	1 Mb
S2	Mobile	No network Connection	3 Mb
S3	Mobile	No network Connection	5 Mb
S4	Mobile	No network Connection	10 Mb
S5	Mobile	No network Connection	15 Mb
S6	Mobile	No network Connection	20 Mb
S7	Mobile	No network Connection	40 Mb
S8	Cloud	Wi-Fi, 3G, 4G	1 Mb
S9	Cloud	Wi-Fi, 3G, 4G	3 Mb
S10	Cloud	Wi-Fi, 3G, 4G	5 Mb
S11	Cloud	Wi-Fi, 3G, 4G	10 Mb
S12	Cloud	Wi-Fi, 3G, 4G	15 Mb
S13	Cloud	Wi-Fi, 3G, 4G	20 Mb
S14	Cloud	Wi-Fi, 3G, 4G	40 Mb

5.8 Experimental Results

During the conducting of the experiments, based on the aforementioned scenarios, the results were obtained and recorded in tables. The tables and graphs below show the results of the three scenarios, based on the network interfaces and the different file sizes to check the power consumption and processing time:

Table 5. 2 Power consumption in (mW) for while executing locally using Wi-Fi, 3G, 4G

File Size(Mb)	Local	WI-FI	3G	4G
1	634.4	1100.4	630	60
3	1366.4	1467.2	2937	234
5	2635.2	1834	6657	408
10	5368	2475.9	Test limitation	654
15	7856.8	6419	Test limitation	804
20	10443.2	9353.4	Test limitation	1245
40	21130.4	12929.7	Test limitation	1476

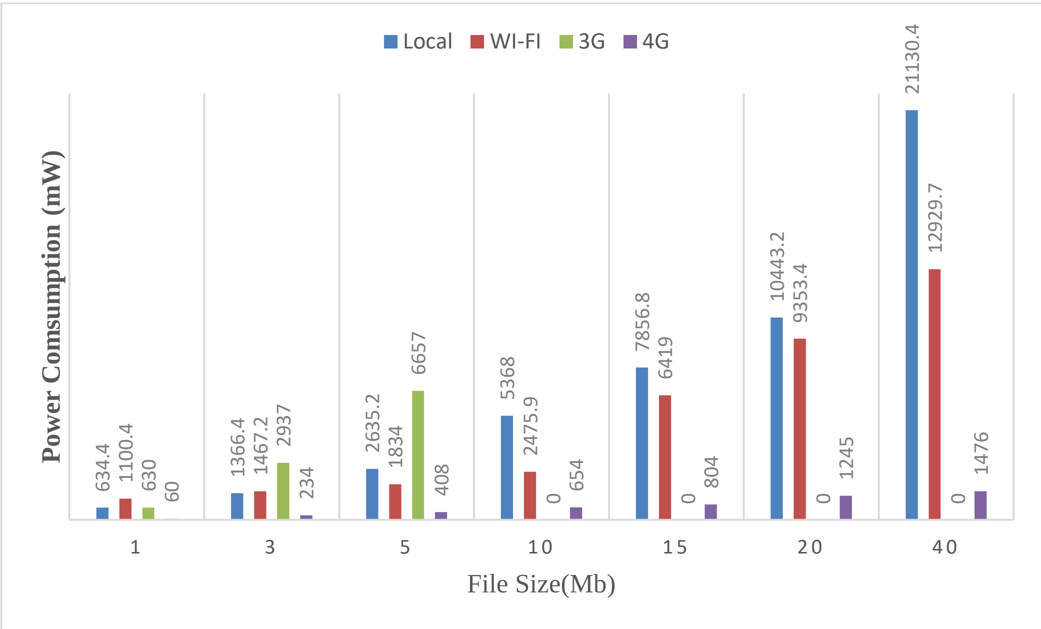


Figure 5. 5 Power consumption whilst executing locally using Wi-Fi, 3G, and 4G

Figure 5.5 describes the power consumption while using Wi-Fi, 3G, and 4G. For both scenarios which are executed locally and also offloaded to the cloud. As it can be seen in figure 5.5, it is noticed that the most efficient network interface to use while offloading, is 4G. For instance, when the file size is 40Mb it is clear to process the task over the cloud instead of processing it locally. It is noticeable during offloading whilst using Wi-Fi and executing locally, that it is better to process locally instead of offloading when using Wi-Fi. This is only for small file sizes as it clear in 1Mb and 3Mb, for instance in 1Mb the power consumption is 634.4 mW but in offloading using Wi-Fi is 1100.4 mW.

Table 5. 3 Processing time locally and offloading using Wi-Fi, 3G, and 4G

File Size	Local	3G	WIFI	4G
1	1.3	21	1.2	2
3	2.8	97.9	1.6	7.8
5	5.4	221.9	2	13.6
10	11	Test limitation	2.7	21.8
15	16.1	Test limitation	7	26.8
20	21.4	Test limitation	10.2	41.5
40	43.3	Test limitation	14.1	49.2

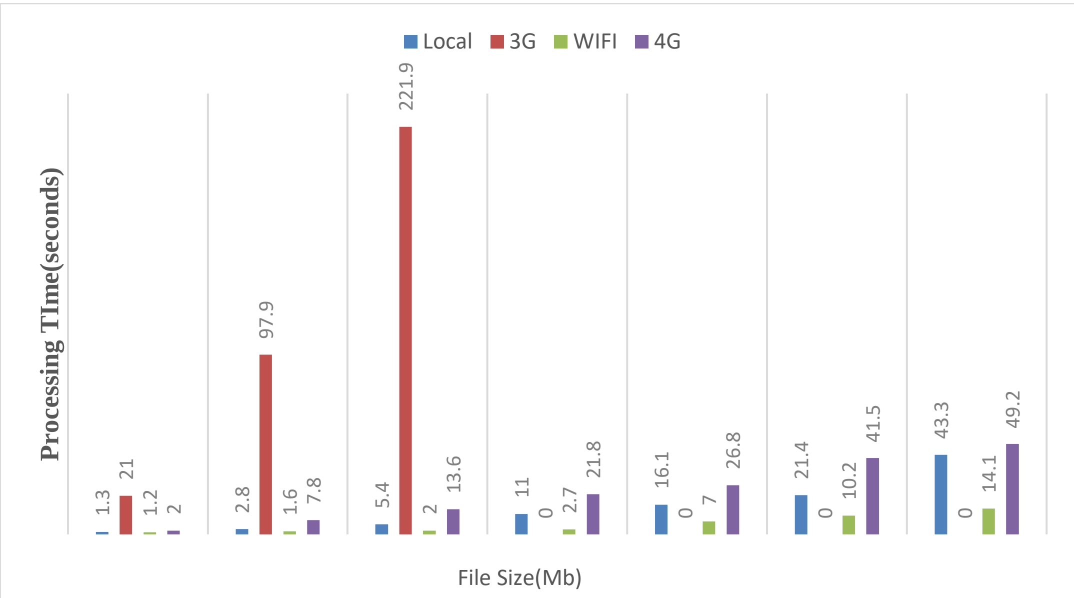


Figure 5. 6 Processing time locally and offloading using Wi-Fi, 3G, and 4G

Figure 5.6 shows the processing time when processing tasks locally and by offloading. The processing time for using Wi-Fi is the lowest time comparing to other network interfaces. For instance, the duration when the file 20 Mb is 10.2 seconds whereas in the mobile is 21.4 seconds and in 4G is 41.5 seconds. It can be seen from the figure the duration on the mobile is better than 4G as in 3G there was a limitation in the test. But at the same time there is more power consumption whilst processing on the mobile, as discussed in figure 5.5.

Table 5. 4 Power consumption for mobile and offloading using 4G

File Size (Mb)	Local	4G
1	634.4	60
3	1366.4	234
5	2635.2	408
10	5368	654
15	7856.8	804
20	10443.2	1245
40	21130.4	1476

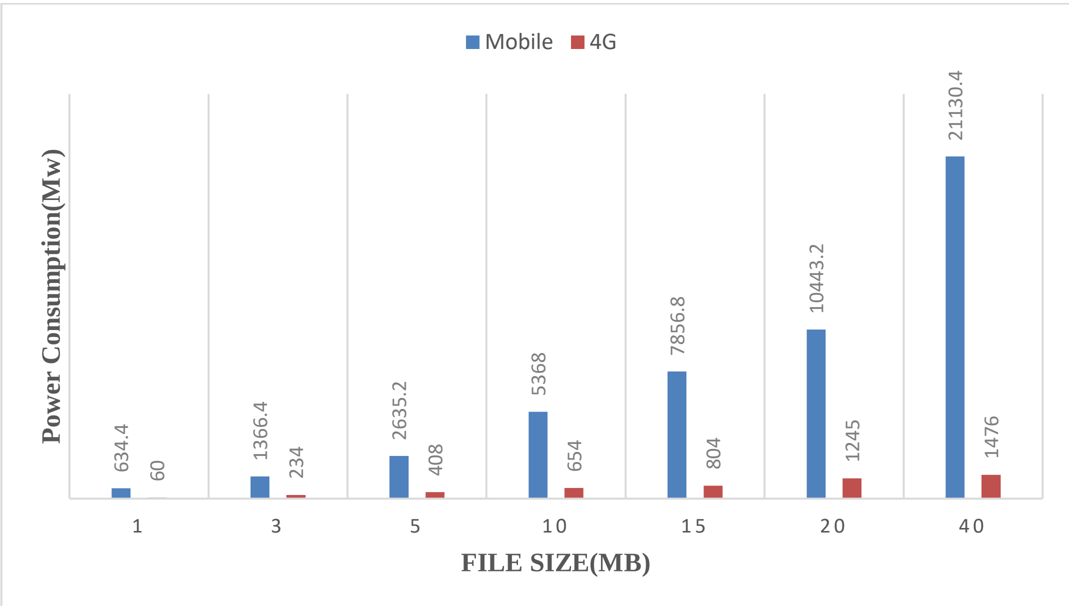


Figure 5. 7 Power consumption locally and offloading using 4G

Figure 5.7 shows the power consumption for locally and offloading by using 4G. It is clear the power consumption in offloading using 4G is better than execution locally. For instance, when the file size is 40 Mb the power consumption using 4G is 1476mW while locally it is 21130.4 mW. In this case, the mobile could save nearly 90% of the power while offloading and using the 4G technique.

Overall, the conclusion from the previous figures and tables illustrates that the most efficient option is to process the tasks by offloading to the cloud server using the 4G technique. This is because there is a reduction in the power consumption which supports mobile users to save their mobile batteries. However, from the perspective of the processing time there is a minimal delay compared to the processing on the mobile.

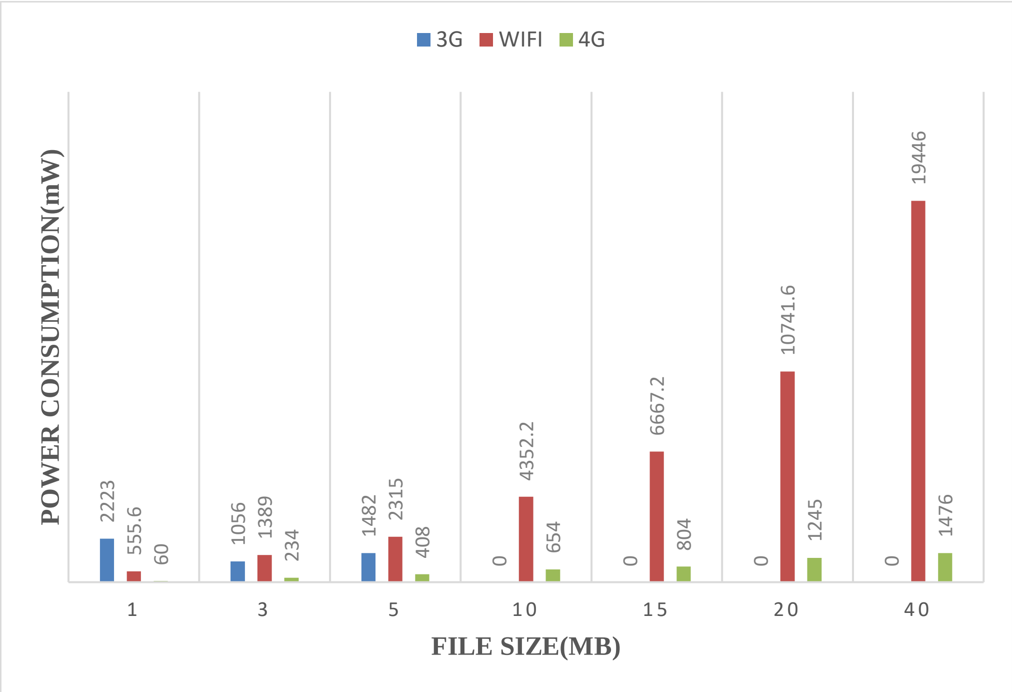


Figure 5. 8 Power Consumption 3G, 4G, and Wi-Fi

Figure 5.8 shows the comparison between 3G, 4G, and Wi-Fi. This figure displays the amount of power consumption during offloading files in different file sizes, they range from 1Mb to 40 Mb. As a result, the total power consumed in offloading is the sum of the sending tasks and executing the tasks. These results indicate that the new proposed model accurately recognises the power consumption which is required to complete offloading the tasks over the cloud. Additionally, the results highlight that the MECCA application accurately recognises the consumed power in the mobile device, and this leads to a suitable offloading decision. According to 3G, up until the file size is 5 Mb the mobile device could send tasks, however, all file sizes over 5 Mb resulted in a server timeout. Table 5.5 below compares the network interfaces for particular parameters. Table 5.5 below compares network interfaces for particular parameters:

Table 5. 5 Network interfaces comparison

Parameter	3G/4G	Wi-Fi
Power Consumption	Lower	Higher
Connection Speed	Higher	Lower
Processing time	Lower	Higher

A set of experiments were conducted to validate the power consumption model for 3G, 4G, and Wi-Fi when sending tasks to the cloud. Figures and tables below will illustrate the results.

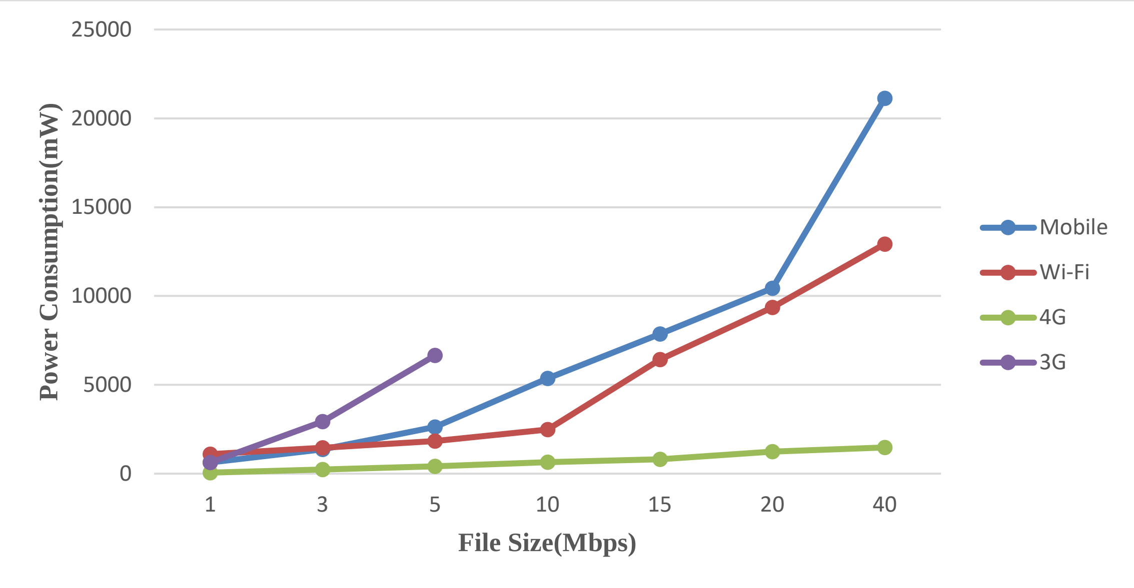


Figure 5. 9 Power consumption for different file sizes for mobile and offloading

Figure 5.9 compares the average of the power consumption for tasks performed locally on the mobile and offloading on the cloud by using different types of network interfaces, as the file size increases. The results show that in the offloading using 4G is the most efficient decision to minimise the power used, which helps mobile users to save the mobile battery.

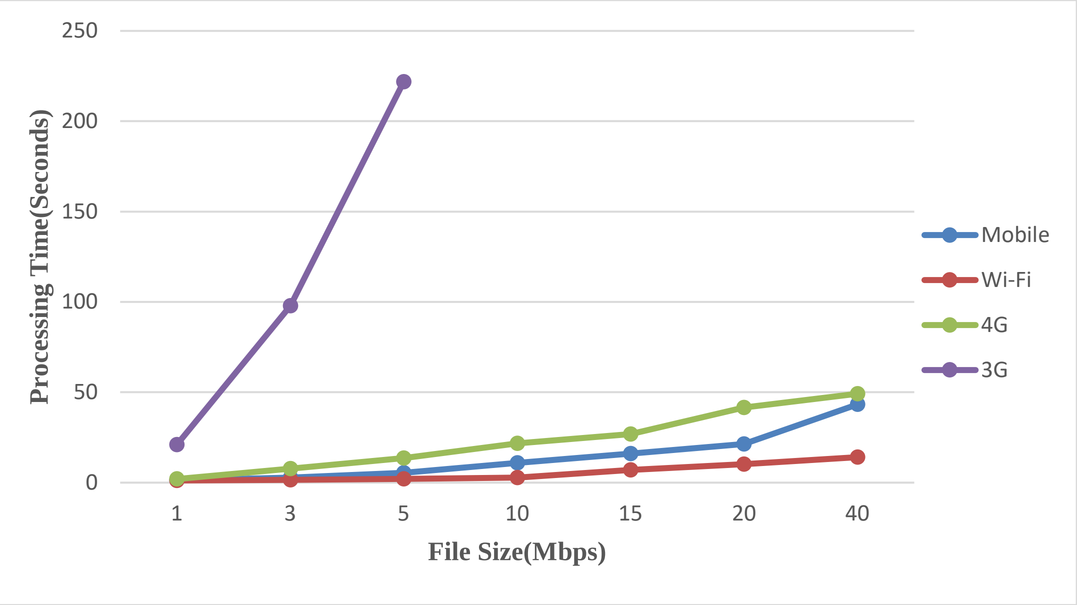


Figure 5. 10 Processing time for mobile and offloading

Figure 5.10 compares the processing time for tasks which were executed on the mobile and on the cloud by offloading, by using different types of network interfaces, as the file size increases for both. The results show that during offloading, using Wi-Fi is the appropriate choice to save the time. The duration of processing in the mobile device and Wi-Fi is minimal until the file size reaches 3 Mb, but after that, the time is increased by more than double on the mobile. For instance, for the file size 3 Mb, the processing time on the mobile is 2.8 seconds compared to 1.6 on the cloud. Whereas at 5 Mb, the processing time on the mobile is 5.4 seconds, compared to 2 seconds on the cloud. These indicators lead to support in the decision engine system. Furthermore, it is noticeable that when using 3G the processing time is severely high in comparison to other network interfaces.

5.9 Compressed Files Results Discussion

The table below illustrates the scenarios for processing files while the file is compressed. The transferring file from normal file to compress is implemented on the mobile. To process files over the cloud, the mobile device compresses files through the developed application, when the file is compressed, the mobile connects through 4G to the cloud and then offloads the compressed file. Table 5.6 below shows the scenarios for the different file sizes.

Table 5. 6 Scenarios and network activities

File Size (Mb)	Scenarios	Location	Network Interface	Mode
1	S1	Cloud	4G	Compressed File
3	S2	Cloud	4G	Compressed File
5	S3	Cloud	4G	Compressed File
10	S4	Cloud	4G	Compressed File
15	S5	Cloud	4G	Compressed File
20	S6	Cloud	4G	Compressed File
40	S7	Cloud	4G	Compressed File

Figure 5.11 below illustrates the results for the normal files without compression and the compressed files for different file sizes during offloading when using the 4G technology.

The time is inclusive of the compression, offloading, processing, and receives the results on the mobile from the cloud. The results show that the delay in the compressed files is less than normal files. For example, in the compressed file, when the file size is 20 Mb the processing time is 24.2 seconds, whilst in the normal file for the same file size is 41.5 seconds. As it is noticeable, when the file size is 3 Mb and it is compressed, the processing time is 3.3 seconds, when compared to the same file size without compression the processing is 7.8 seconds. This illustrates clearly the processing time is more than double time. From the previous graph, it is apparent the processing time for the compressed file whilst offloading using 4G technology is less than the non-compressed file for every file size that was tested.

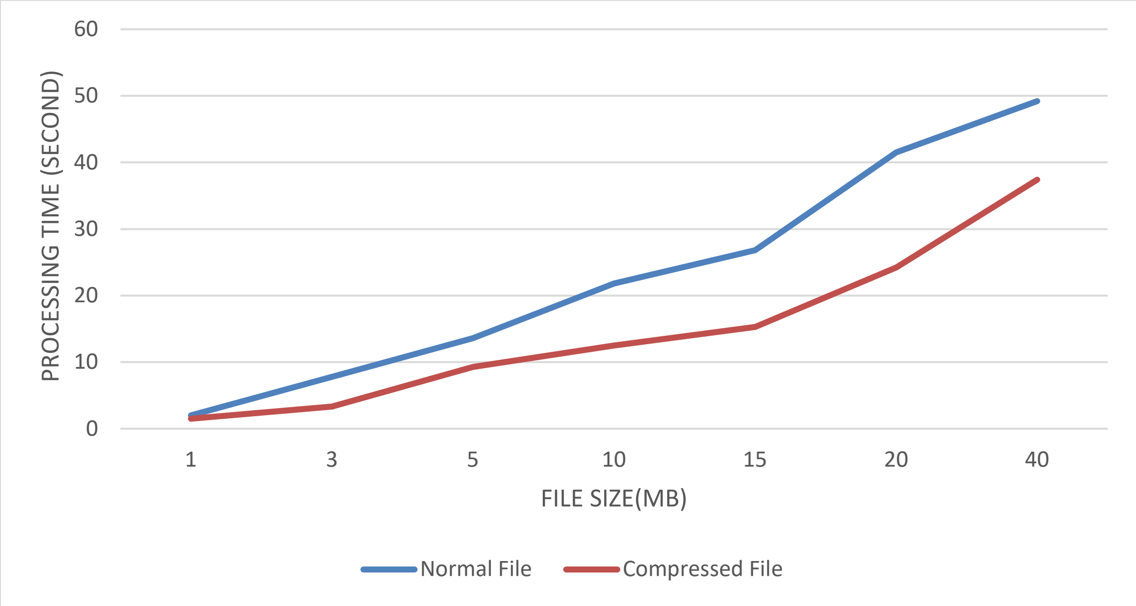


Figure 5. 11 Processing time while offloading using 4G for normal and compressed file

Table 5. 7 Processing time for offloading using 4G for compressed and non-compressed files

File Size (Mb)	Processing Time-Normal File(Sec)	Processing Time-Compressed File(Sec)	Average of reducing time
1	2	1.5	25%
3	7.8	3.3	58%
5	13.6	9.3	32%
10	21.8	12.5	43%
15	26.8	15.3	85%
20	41.5	24.2	42%
40	49.2	37.4	24%

Table 5.7 presents the results and the average percentage of the reduction in the processing time. The results show that there is an enhancement in reducing the time whilst offloading files with the use of the 4G connection. For example, in file size 15 Mb the processing time for the normal file was 26.8 seconds, but for the compressed file was 15.3 seconds. So the average of the reduction in the processing time is 85%. Overall, there is a significant value in the reduction of the processing time if the file is compressed prior to offloading.

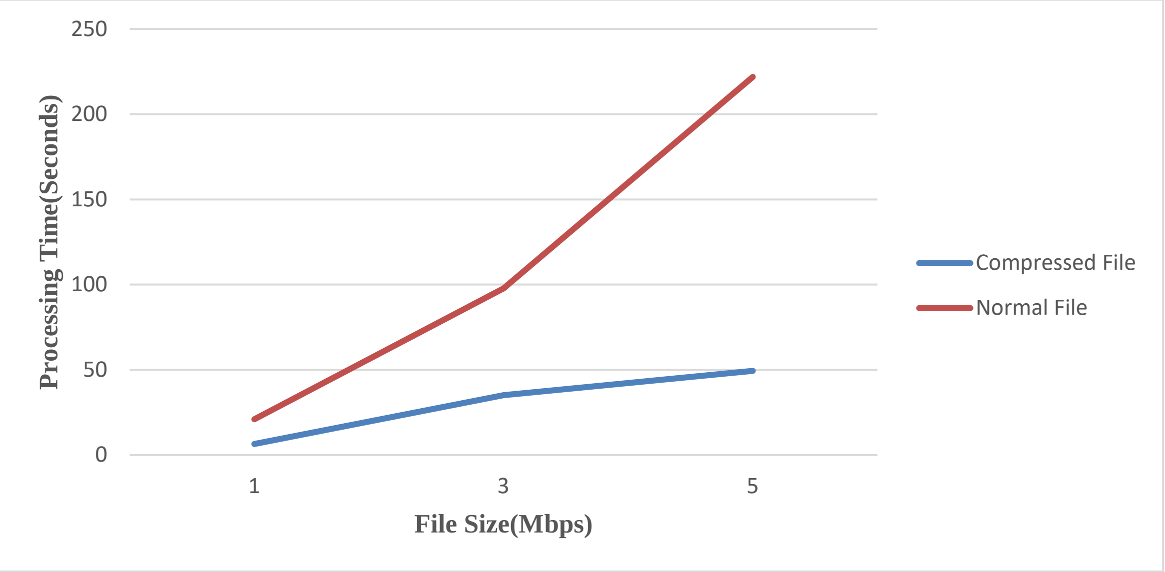


Figure 5. 12 Processing time while offloading using 3G for normal and compressed file

Figure 5.12 shows the processing time whilst offloading the files to cloud using 3G technology. For the three file sizes 1 Mb, 3 Mb, and 5 Mb, the results demonstrate the processing time for compressed files is less than normal files. It is clear in the file size 5 Mb, that when the file is compressed, the processing time was 50 seconds, comparing to normal file which was nearly 250 seconds, so it is one fifth.

From the figure 5.12 and 5.13 it is evident that when the mobile compressed the file then offload to cloud, the processing time has a minimal delay.

5.10 Discussion of the Results

The aim of this chapter was to find out the impact of the network interfaces, Wi-Fi, 3G, and 4G on the power consumption and processing time whilst offloading to the cloud. In addition, to examine the effect of increasing the file sizes. The outcomes from the experimental results illustrate that offloading the heavy tasks from the mobile device to the cloud is beneficial. According to the file size, the results indicate that there are greater benefits to process it on the cloud as the file size increases. In contrast, it is noticeable that smaller files are processed more efficiently on the mobile itself. Therefore, there is a trade-off and it is important to consider which path the file would be processed most effectively. This has been addressed by using an innovative algorithm which makes the decision of how the file should be processed. The most important reasons to use the proposed algorithm are to limit power consumption and minimise the processing time.

The results revealed that using 4G connection reduced the power consumption by up to 90% compared to Wi-Fi 61%. This means both of the network interfaces have a significant potential to save the mobile battery which is known to be a major limitation for mobile users.

In addition, compression of the files was significantly beneficial in relation to the minimal delay as discussed previously.

Although these results reflect that there is a benefit to upload to the cloud due to the potential of saving mobile battery and minimum delay, one issue that was highlighted, is that if the dedicated server was busy, the file was unable to be processed. The implication of this is that the mobile users would have to re-upload the file. This could be a disadvantage as there is a possibility that it would impact the mobile battery further, and consume the mobile data if the user was using the 4G connection. Similarly, if the mobile user chooses to upload the same file more than once, this affects both parameters.

For the previously stated reason, two ideas will be added in MECCA in the following chapter, these are: to offer a secondary server when the main server is busy, and to save a copy of the file on the cloud server.

6 Chapter 6 New Model for Offloading to Cloud: FUR and SUR

In this chapter, new functions processes will be developed in the mobile application to facilitate the processing of files over the cloud computing using various network interfaces. A variety of experiments will be conducted to test the proposed processes. Additionally, to examine their effect on the battery usage and processing time in terms of mobile devices.

6.1 Introduction to the Third Experiment

Task offloading technique from mobile devices to the cloud server is capable to enhance the computing tasks of mobile devices, which support to extend their battery life with minimum delay. In addition, to ensure offloading is valuable, it is noticed from the previous chapters, that there was a need to develop new functions in terms of reducing the mobile device battery usage and reduce the processing time. The following functions will be created to support offloading files from smartphones to the cloud server. The new processes are, offloading tasks to another server if the first server is busy and also to save a copy of the file after processing on the server.

6.2 Third Experiment Outlines

The purpose of this stage of experimentations is to address and enhance the QoS for the mobile users, such as mobile battery and reduce delay. To enable this to be investigated the following new objectives will be implemented to process the tests.

1. Offer a secondary server to serve the mobile users if they get acknowledgment from the main server that it is busy.

2. Save a copy from the file after processing for the first time to save battery and for less delay.

These new features will be integrated into the mobile application which is already used in the previous chapter.

6.3 Experiment Three Methodology

To maintain consistency in this research the experimentation that will be completed within this chapter will use the same parameters as the experiment detailed in chapter 5. The file sizes selected are: 1 Mb, 3 Mb, 5 Mb, 9 Mb, 10 Mb, 15 Mb, 20 Mb, and 40 Mb. In addition, as mentioned in the methodology chapter CSV file format will be used to maintain continuity of the results. The network interfaces that will be used to connect mobile and cloud will be Wi-Fi and 4G only. This differs to the previous experiments as there will no longer be use of 3G, as there was a limitation in the experimentation tests.

6.4 Experimental Setup Tools

To conduct the experiments for processing files, there are two perspectives, client and server. From the clients view point, the smartphone is required to install the deployed mobile application. Samsung Galaxy Tab 4 T335 will be used in each scenario, the specifications are: processor core Quad-core (4 Core), operating system Android 5.1.1, wireless LAN standard IEEE 802.11a, and flash memory capacity 16 GB. After the smartphone is ready for the use, files 1 Mb, 3 Mb, 5 Mb, 9 MB, 10 Mb, 15 Mb, 20 Mb, and 40 Mb, are required to be copied on the mobile SD card. Subscriber identification module (SIM) card is necessary to gain access to the Internet using cellular generation 4G which will be used by the mobile device. After these outlined elements are in situ, the mobile users are ready to commence the testing processes.

From the perspective of the server part, there is a dedicated server from Google cloud with high specifications as it is described in chapter 3 section 3.7. In addition, several types of tools such as NetBeans, and Java are required to be installed to support the running of the program. After both parts are equipped, mobile users have two options to gain access to the Internet through either Wi-Fi or cellular data. In these experiments for Wi-Fi network, Anglia Ruskin University (ARU) wireless network will be used, which is called 'eduroam' network. The closest access point to the test base is nearly 4-5 metres.

For the 4G based experiments, the network that was chosen was Vodafone Company, because it uses two different frequencies as shown below:

- The first is 2100 MHz and is on band one. The benefits from this band is that the 4G has Long Term Evolution (LTE) which means a high-speed network available.
- The second is frequency is 800 MHz and it is on band 20. The value of this frequency is that it penetrates a good signals into buildings

The two frequencies combined to provide an overall better coverage specifically within buildings where the experiments scenarios were conducted. Compared to other companies, for example EE, they only use one frequency which 2800 MHz which is overall better speed of 4G, but they do not have the benefits of the second frequency which aids 4G signal within buildings.

After reviewing the available telecommunication companies, it was deemed that Vodafone has the best coverage among them on the ARU campus.

6.5 Mobile Application Architecture

The developed application which is used in the previous chapters, will be used in this chapter but with the new functions to fulfil the aims. The new features could help in reducing the power consumption and with minimum delay. Due to the fact, there is a secondary server to support the processing whilst the primary server is occupied with another request from mobile users. In addition, there is a development of another service which duplicates the file on the server to prevent uploading the files for a second time, which is expected to reduce the delays in processing time and the limitations to the battery.

The mobile application design consists of two main parts:

1. **Mobile Application:** this application is installed on the smartphone. It consists of two parts in terms of the deploying (Layout and class activities) which are as follows:
    a. **Main Activity:** contains a method called on-create to establish the activities
    b. **Connectivity:** the purpose of this is to check the connection type, either 4G or Wi-Fi.
    c. **File array:** this help helps to view the files as a list in the smartphone.
    d. **File chooser:** enables the mobile users to select the file for processing.
    e. **Get file:** take the selected file to commence processing either on the mobile or on the cloud.
    f. **Receive:** supports to demonstrate the results on the mobile screen.
2. **Server Application:** which is located on the cloud server. Different types of tools such as NetBeans and Java were installed on the server to support and facilitate the

processing of files. After finishing the processing, the application server sends the results back to the mobile users.

This mobile application architecture can be seen below in Figure 6.3, and it will be described in section 6.7.

6.6 Power Measurement Process

Each smartphone has different power consumption properties, as well as requiring different kinds of power models to estimate the battery drainage. Smartphones do not contain these types of models to measure the accurate power consumption. As discussed in chapter three section 3.10 PowerTutor was used to estimate the power consumption for chapter 4 and 5. A new power measurement process was produced from Android Debug Bridge (ADB). ADB is a command-line tool which allows users to communicate with a device either a connected Android device such as Samsung and, HTC or an emulator. The ADB command helps in a variety of device actions, in this research, it will be used to estimate the precise power consumption for the deployed application, which is MECCA application. The ADB command works as a client-server program, this will be divided into three components:

- **Client:** In this part, the commands will be sent. The client runs on the development machine. The client can send a request by issuing an ADB command, using Command Prompt Command CMD.
- **A daemon (adbd):** which allows the commands to run on a device. The daemon operates as a background process for each mobile device.
- **Server:** This part helps to manage the communication between the client and the daemon.

To run mobile power measurement processes on CMD there are several steps required as shown below:

1. Connect Android device to the computer machine over Universal Serial Bus (USB).
2. Display the list of devices by typing: AppData\ Local\ Android\ sdk\ platform-tools\ adb devices.
3. Stop ADB server: AppData\ Local\ Android\ sdk\ platform-tools\ adb kill-server.
4. Start ADB server connection again: AppData\ Local\ Android\ sdk\ platform-tools\ adb devices.
5. Reset battery dumps: AppData\ Local\ Android\ sdk\ platform-tools\ adb shell dumpsys batterystats –reset.
6. Disconnect the Android device by unplugging it from the computer machine.
7. Run the application, select the file, choose the connection type, and then upload it. This step will be explained in detailed in section 6.8.
8. Connect Android device with the computer machine over USB.
9. Display the list of devices by typing: AppData\ Local\ Android\ sdk\ platform-tools\ adb devices.
10. Copy the results file from Android device to computer machine by using pull and push commands: AppData\ Local\ Android\ sdk\ platform-tools\ adb shell dumpsys batterystats > batterystats_BatteryResults1.txt.

After performing the previous steps the text file is ready and shows ‘BatteryRults.txt’, it contains the estimated power separately for each application that the mobile users used at the same time, in detail. The most important factor in the text file in relation to this research is the battery measurement for the MECCA application.

The two figures below show the previous steps in detail:

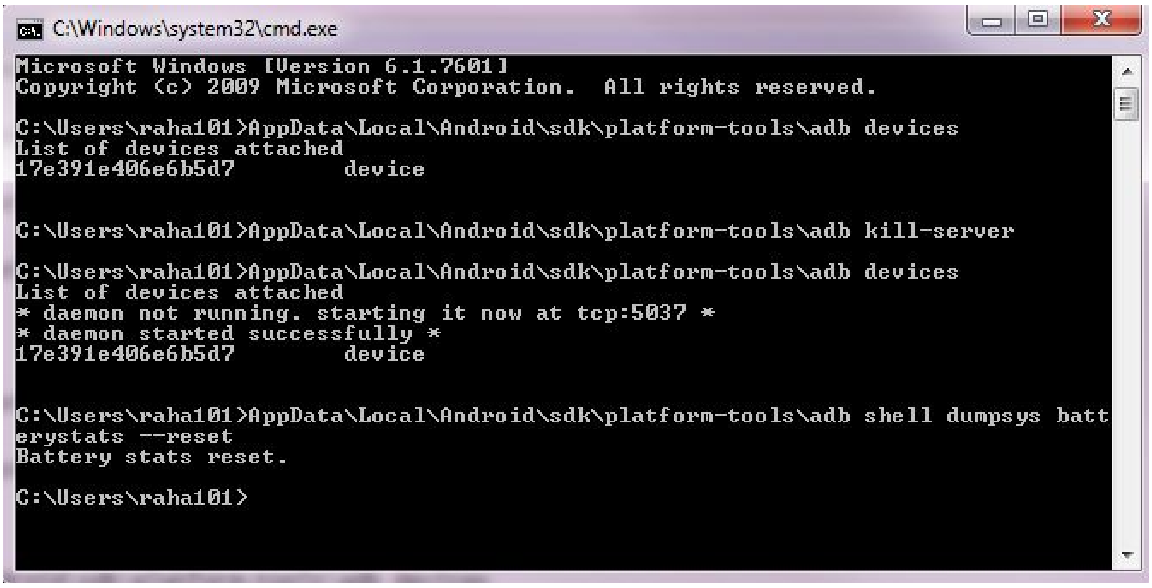


Figure 6. 1 Command lines from step 2 to 5

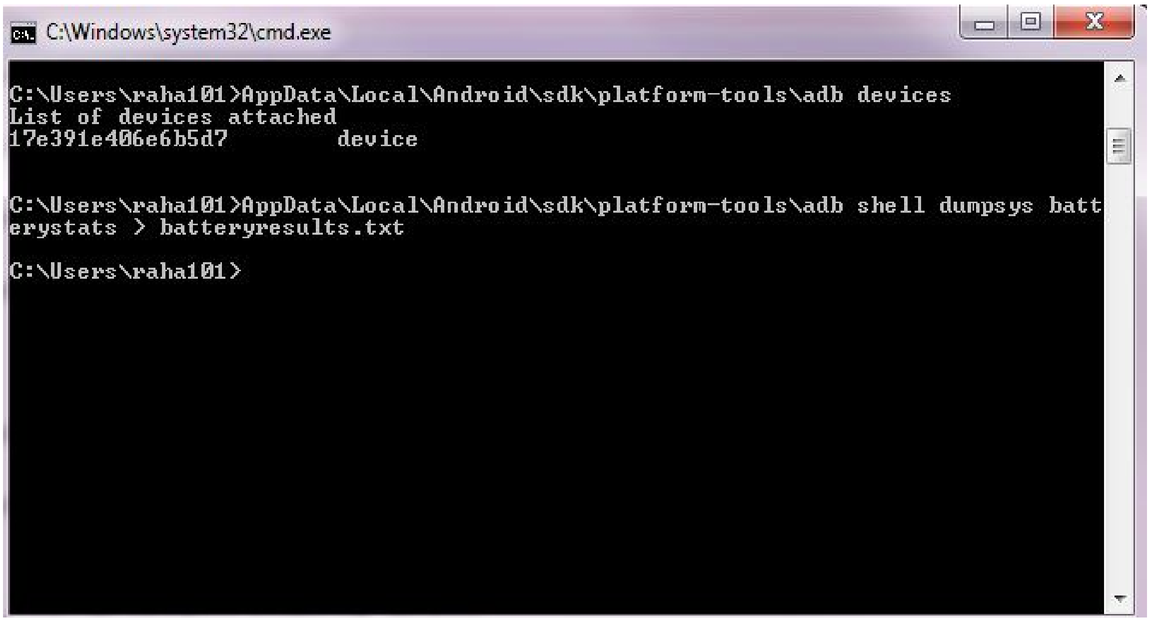


Figure 6. 2 command lines from step 5 to 7.

6.7 Experiments Scenarios

During the experimental process, it is important to obtain the results many times. In this chapter around 140 experiments will be conducted at a random time. The variables that will be considered during the experiments will be the network interface, time of experiments, file size, and mobile battery usage. Each file will be tested 5 times using the Wi-Fi and 5 times using the 4G, during offloading files from the mobile device to the cloud. This means, that each file will be tested 20 times, 10 times for the FUR and 10 times for the SUR. The reason for this, it is important to provide an average of the results which will reflect and validate the outcomes of the proposed mechanism.

In the first time of offloading, First Uploading Round (FUR), the server is completely empty, this means there is no saved file on it. Once the mobile user sends the file, a new copy of it will be saved on the Google cloud and it is ready to process. The previous steps which are discussed in section 6.6 will be used to measure the accurate power consumption during offloading, processing, and sending the results back to the mobile user, on the Google cloud using 4G and Wi-Fi. A new mechanism which was discussed earlier in the mobile architecture section 6.5 supports to save the offloaded files on the cloud server. This could effectively help to reduce the mobile battery usage with minimum delay. When the same file needs to be offloaded on the Google cloud this is called, Second Uploading Round (SUR). This is because the file is saved on the server during the FUR. After processing the file on the server, the battery usage and the time will be measured. In addition, the results of processing files will send back to the mobile user, and it will be shown on the mobile device screen as seen in figure 6.3. Table 6.1 below illustrates the scenarios for FUR and SUR for a variety of file sizes and network interfaces on the cloud.

Table 6. 1 Scenarios for FUR and SUR

File Size (Mb)	Scenarios	Location	Network Interface	FUR	SUR
1	S1	Cloud	Wi-Fi, 4G	10	10
3	S2	Cloud	Wi-Fi, 4G	10	10
5	S3	Cloud	Wi-Fi, 4G	10	10
10	S4	Cloud	Wi-Fi, 4G	10	10
15	S5	Cloud	Wi-Fi, 4G	10	10
20	S6	Cloud	Wi-Fi, 4G	10	10
40	S7	Cloud	Wi-Fi, 4G	10	10

The mobile user chooses one of the network interfaces either 4G or Wi-Fi. Once the connection is chosen, the mobile user browses the available files on SD card which is located on the mobile storage. After the file is chosen to be offloaded to the cloud, the processing commences. After a while, the results will be shown on the mobile screen with the measurement of the processing time. Figures 6.3, 6.4, 6.5, and 6.6 show the results of offloading for file size 5 Mb to cloud server whilst using Wi-Fi and 4G connections. The screens show the results for performing files and the time of the performance in millisecond for FUR and SUR.

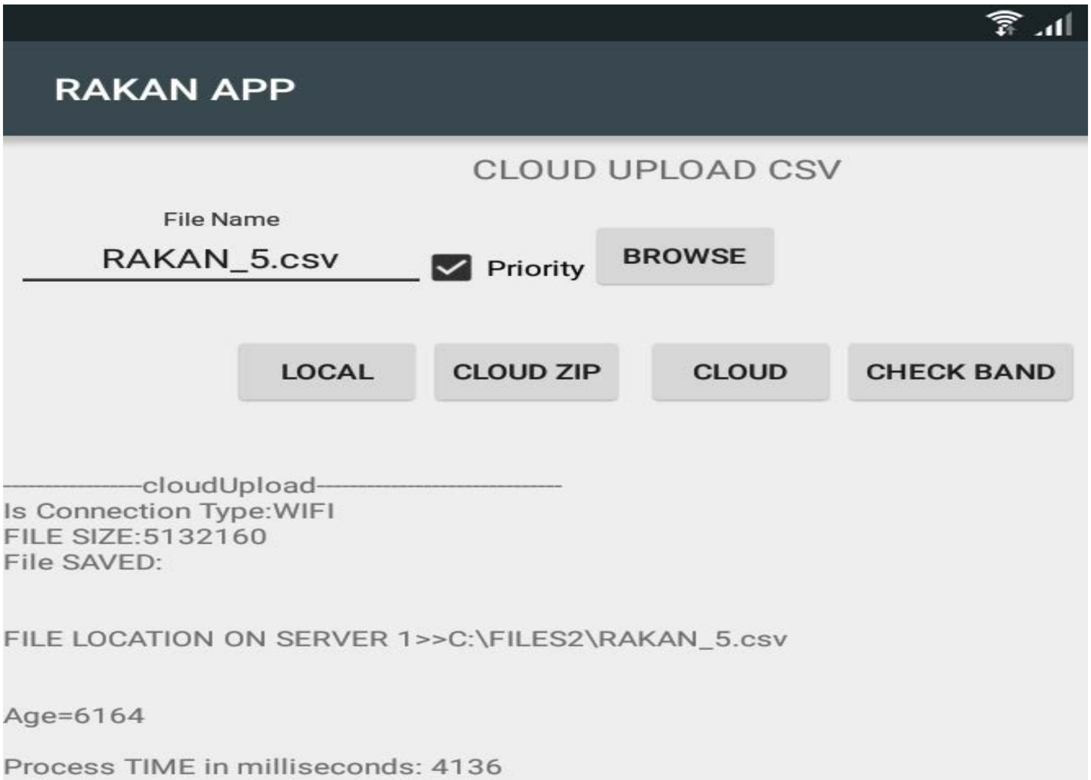


Figure 6. 3 FUR for offloading file 5Mb to cloud using Wi-Fi

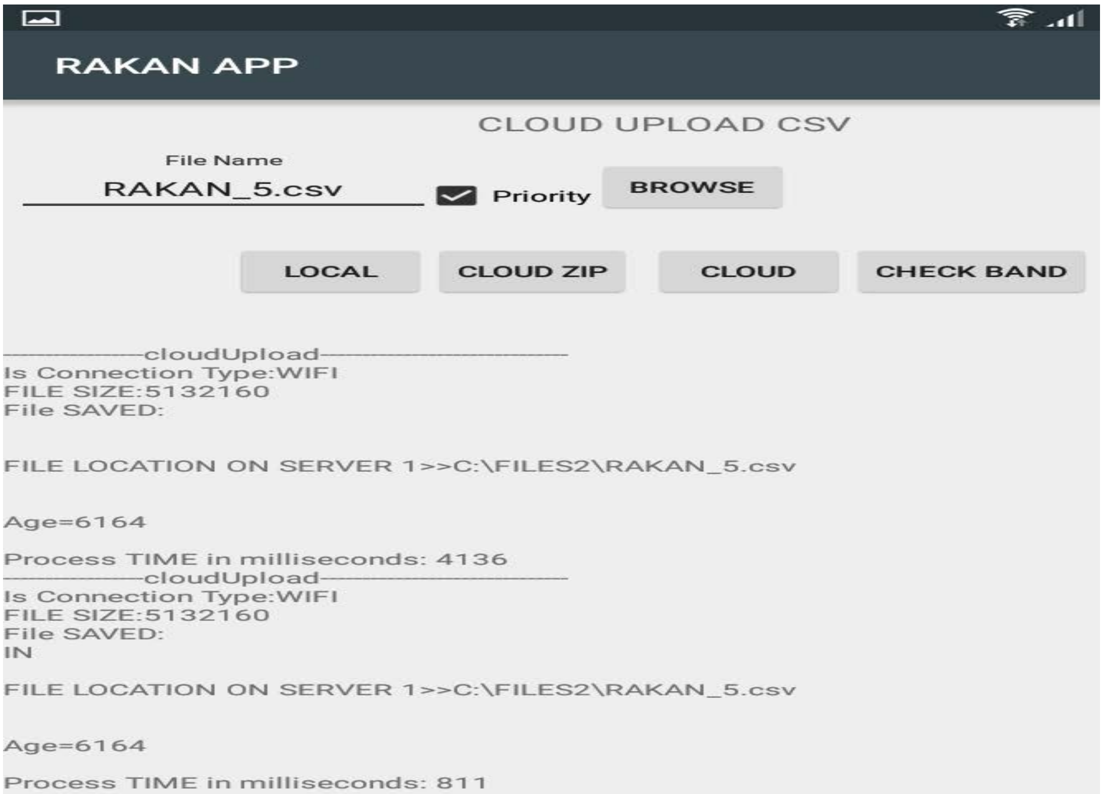


Figure 6. 4 FUR and SUR for offloading file 5Mb to cloud using Wi-Fi

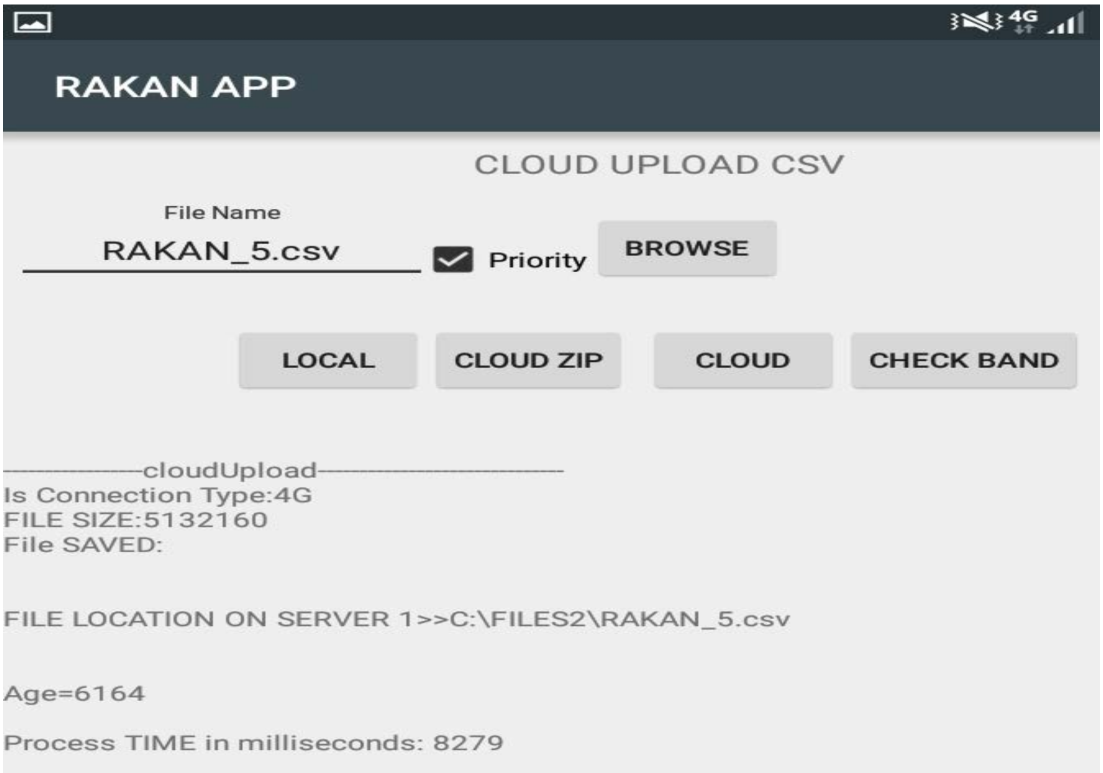


Figure 6. 5 FUR for offloading file 5Mb to cloud using 4G

Figure 6. 6 FUR and SUR for offloading file 5Mb to cloud using 4G

The mobile battery usage measurement is essential, this process was discussed in detail in section 6.6. Once this is completed, the experiments will be repeated for the SUR as it can be seen in figures 6.4 and 6.6. The reason for the second round is to find out if there is a significant impact by saving the file on the Google cloud server. The benefit of this method is that it can be effective on the power consumption or the processing time or even on both of them. In addition, the experiments can be repeated on different network interfaces such as 4G, but the server should be empty of the files. This means that the used file needs to be removed from the server before the 4G experiments are conducted. This can be completed by copying the following link on the browser: http:// 104.154.110.104/ RAKAN_WEB/ DELETE_FILES? SURE=yes. This step needs to be repeated when changing the network interfaces from Wi-Fi to 4G or the opposite. The 104.154.110.104, is the IP address for the dedicated cloud server.

6.8 The Beneficial Aspects of MECCA Application

The goal of MECCA is to increase the benefits for the mobile users when the files are required to be offloaded from smartphone devices to Google cloud. In this part, Mecca components will be presented to demonstrate how they combine into one architecture structure for creating a mobile application.

The developed MECCA model support end users to decide where to offload the files for remote execution, which is a cloud server, additionally, selecting the suitable connection. In each time the processing is initiated by the available server. MECCA uses its optimisation processing to make the decision whether to offload or not. The following section will explain in more detail about the decision engine. Once offloading of the file is completed, the processing on the Google cloud will commence.

In MECCA model, each function is responsible to estimate and reduce the cost of offloading, such as the file which is required to be transferred, network interfaces, availability of the servers, and the existence of the file on the server. All these variables will be used to formulate the benefits of offloading or to execute locally on the smartphone.

Figure 6.7 illustrates an overview of the proposed model (MECCA) architecture. As it appears in the figure below, in the smartphone part, MECCA consists of four elements: 1) **Interface** which enables the mobile users to choose the file and the method of the processing, 2) **Proxy** which manages and controls the file transfer for offloading method, 3) **Profiler** helps to collect information such as connection type and file size. In addition, main activities such as array to show the files, file chooser to support in selection the file, finally, T**ranscript** to execute files if the process takes place on the smartphone. On the server part, MECCA has three components, 1) **Proxy** which offers similar roles to the client part. 2) **Profiler** to save and compare the saved file before offloading it additional time.3) in **Transcript** element is to coordinate and process the incoming requests from the smartphones to the servers. Figure 6.7 shows the proposed application architecture.

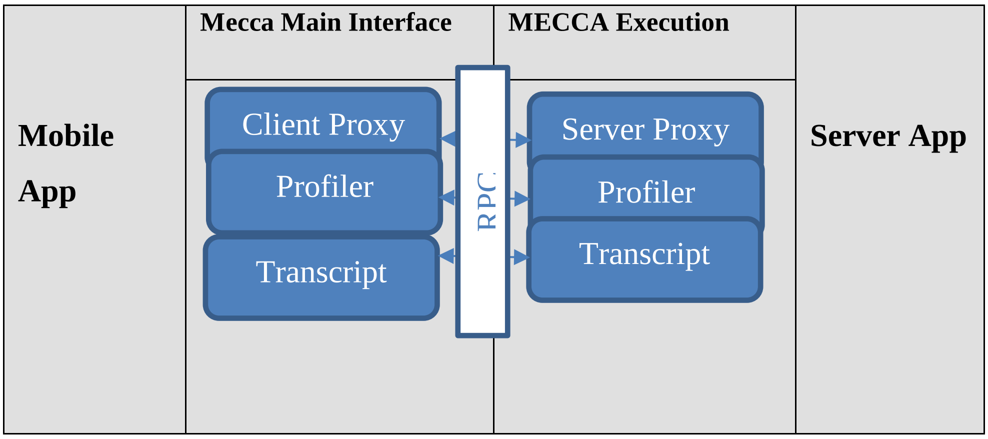


Figure 6. 7 Application Architecture

To commence processing the files, MECCA application on both parts smartphone and server components are ready for execution of the tasks. As mentioned earlier, the MECCA model supports two options: 1) process the files locally on the smartphone which sometimes could incur more energy consumption and delay. 2) Send the files over the Internet to the cloud server which is already appointed for MECCA application.

Subsequently, when the MECCA is ready to process the files on both sides, end users should select the file, processing location, and network connection type. Inside the application as mentioned in section 6.2, the functions are expected to support the reduction of the battery usage and the time delay. For example, when the client offloads the file to Google cloud to process it, the results of the processing will be sent back to the client over the Internet. If the mobile user needs to process the same file later, in addition, it will affect the mobile battery usage and increase the time delay. In this case, one of the proposed functions of the server part called 'checking saved files' supports the model to save a copy of the file after processing. When there are additional uploads of the same files by the mobile users, the application will read the file and check with the server. Within the server part there is a process called 'checking save file' to compare the selected file from the mobile with the saved files on the backup of server one and server two as illustrated in figure 6.8. This will benefit the mobile users as the file will not be required to be uploaded again as it already exists on the server, and therefore if the file does not have to be re-uploaded it could save the mobile battery and the time.

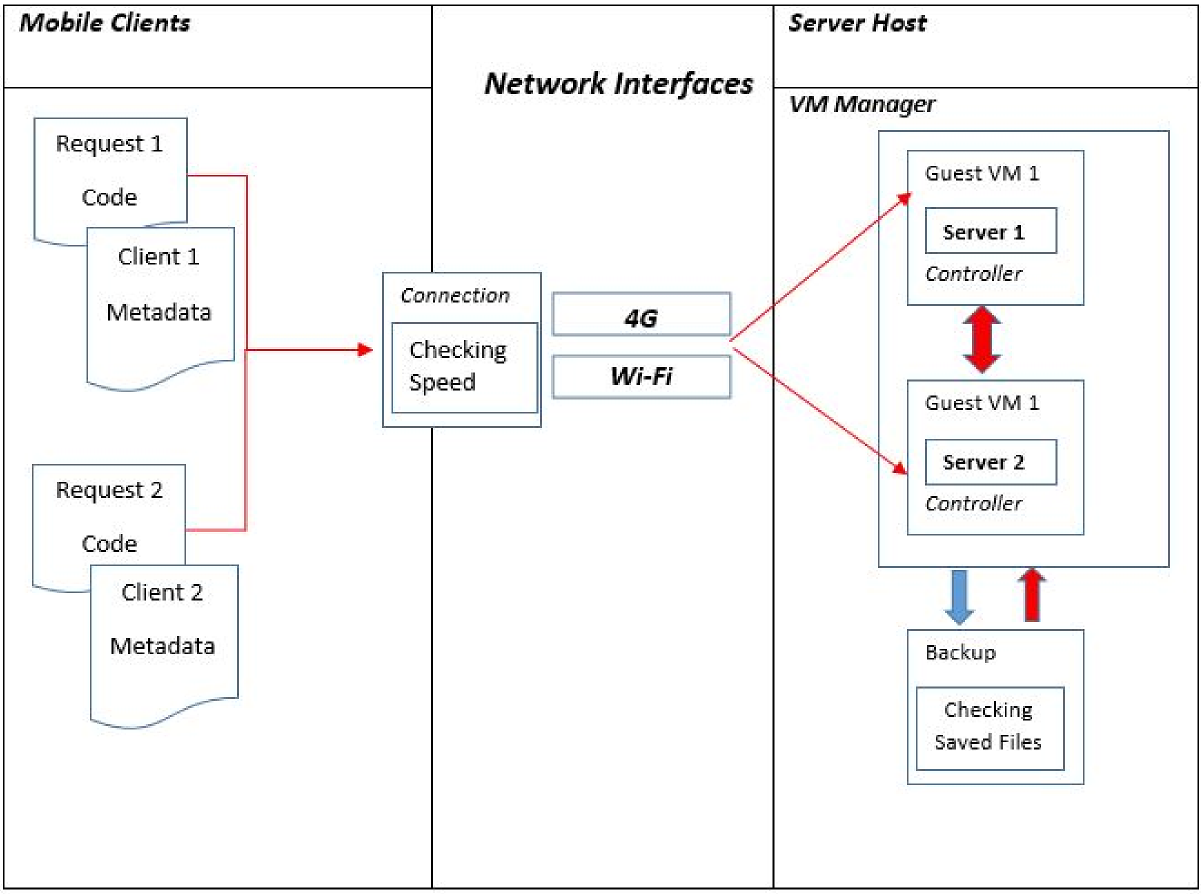


Figure 6. 8 Mobile Application Architecture Components

According to the quality of service in terms of the delay, another server will be used for the experiments in this chapter. The benefit from this, is to serve a big amount of incoming requests from the mobile users. When the request is sent to the server and if the server is equipped with another request, the new request automatically will be transferred to the second server. In this case, there is no delay for the mobile user to wait until the main server finishes processing the file. Additionally, it should also not miss the request from the clients, so the smartphone could save time and battery as it transfers the requests directly to the second server. The previous work is organised by the controller which is located in the server part as shown in figure 6.8.

6.9 Decision Engine System

The decision engine is a tool that requires input from users which is processed using a specific pre-established criteria or formula to direct the outcome to acquire the optimum solution. In MECCA model, there is a Decision Engine (DE) to find out whether to offload tasks remotely or to process it locally. Inside the DE there are conditions to take the suitable choice as shown below:

- **File size:** is one of the most important factors that could affect the smartphones resources whilst processing, offloading, and editing. So it is valuable to take it into account. Here in this research, it is paramount to consider file size in conjunction with the connection type.
- **File name:** it is one of the new features which will be added to MECCA, to compare the selected file with the stored files on the server. In order to make the decision of whether it needs to be uploaded or taken from the previously uploaded files. This is illustrated in figure 6.8.
- **Last Modified:** For each file, there are properties, one of these is the last modification, which shows date and time for the most recent updates on the file, for example, 05 June 2017, 15:48:49.
- **Connection type:** as discussed in previous sections, for each network connection such as 3G, 4G, and Wi-Fi, there is an amount of power consumption and delay, which would have an impact because of the file size. So, it is significant to consider the connection type juxtaposed with the file size.

6.10 Experiments Results Discussion

In the following experiments scenarios, 420 tests were executed for offloading files to cloud server to check and reduce the power consumption for smartphone devices. The experiments tests were conducted at a random time for 120 times, the remaining experiments tests were for scheduled time and a secondary smartphone device was used, these experiments will be presented in the appendix. The tables and figures below will illustrate and discuss the results of the random tests in terms of mobile battery usage and processing time.

6.10.1 Power Consumption Results

The tables and figures below illustrate the results for the power consumption whilst sending and receiving files between the cloud and the mobile device using Wi-Fi and 4G connections in Milliampere hour (mAh). The results will be for FUR and the proposed mechanism SUR as discussed in section 6.7, and to investigate the benefits of it. Tables and figures show the results for files sizes, 1 Mb, 3 Mb, 5 Mb, 10 Mb, 15 Mb, 20 Mb, and 40 Mb for both rounds.

Table 6.2 below illustrates the average of the usage of the mobile battery, while offloading files to Google cloud for different file sizes and for receiving the results of processing back. These results can also be seen visually in figure 6.9. The average for each file was obtained in relation to the offloading for both rounds, using 4G and Wi-Fi network interfaces. From the table below, there is a big difference in the average of the mobile battery usage between FUR and SUR. For example, the average of the mobile battery usage for file size 10 Mb in the FUR when using Wi-Fi was .3922 mAh, compared to the same file size and network connection but in the SUR the average was .05946 mAh. For 4G connection, for example file size 20 Mb, the average of the battery usage for FUR was 1.3408 mAh but in the SUR was .0659 mAh. The previous two examples illustrated the difference between FUR and SUR.

Table 6. 2 Average of the power consumption whilst using Wi-Fi and 4G connections

	Wi-Fi		4G	
File Size (Mb)	**FUR (mAh)**	**SUR (mAh)**	**FUR (mAh)**	**SUR (mAh)**
1	0.18182	0.03604	0.0574	0.03396
3	0.2674	0.0462	0.21402	0.037176
5	0.3636	0.051	0.3076	0.03986
10	0.3922	0.05946	0.6202	0.0425
15	0.4668	0.06816	0.6558	0.05898
20	0.547	0.08176	1.3408	0.0659
40	0.6328	0.124	2.638	0.1468

Table 6.3 illustrates the percentage of the reduction of the power consumption whilst offloading files to the cloud server with the new mechanism SUR. There is a significance in the reduction for the power consumption in both connections Wi-Fi and 4G when using SUR mechanism. For example, when offloading the file size 5 Mb, the power consumption is reduced in both network connection Wi-Fi and 4G to 86% and 87% respectively.

Table 6. 3 Average of the reduction for the power consumption for FUR and SUR

File Size (Mb)	FUR and SUR using Wi-Fi	FUR and SUR using 4G
1	80%	41%
3	83%	83%
5	86%	87%
10	85%	93%
15	85%	91%
20	85%	95%
40	53%	95%

Overall, from tables 6.2 and 6.3, the results illustrated that the new mechanism SUR supports the reduction in the power consumption by more than the triple in some file size. In addition, the average of the reduction of power for SUR in 4G connection is more efficient compared to SUR in Wi-Fi connection. This can be seen in files 3 Mb, 5 Mb, 10 Mb, 15 Mb, and 20 Mb. The outcome of these results reflect that it is crucial to use the 4G connection in the SUR, because it supports to save the mobile battery.

Figure 6.9 below demonstrates the average for the power consumption in mAh whilst processing files from different sizes on the cloud. The network interfaces were Wi-Fi and 4G. For Wi-Fi connection, it is noticeable that the increase in the power consumption was gradual while increasing the file size. On the other hand, the increase in the power consumption for 4G connection was steady for files 1 Mb, 3 Mb, 5 Mb, and 10 Mb. At 15 Mb, there was a sharp increase for all files which is clearly illustrated in figure 6.9. For example, in file size 15 Mb the power consumption was .655 mAh compared to file size 40 Mb which was 2.4024 mAh, this means, there is an increase of more than triple. The benefit from the SUR can be seen in figure 6.9 for both connections, Wi-Fi and 4G. The reduction in the power consumption for the SUR was better in 4G connection than Wi-Fi. For example, in Wi-Fi for file size 5 Mb the power consumption average was 0.051 mAh compared to the same file size in 4G which was .03986. A further example, for file size 10Mb was .05946 mAh, .0425 mAh, respectively.

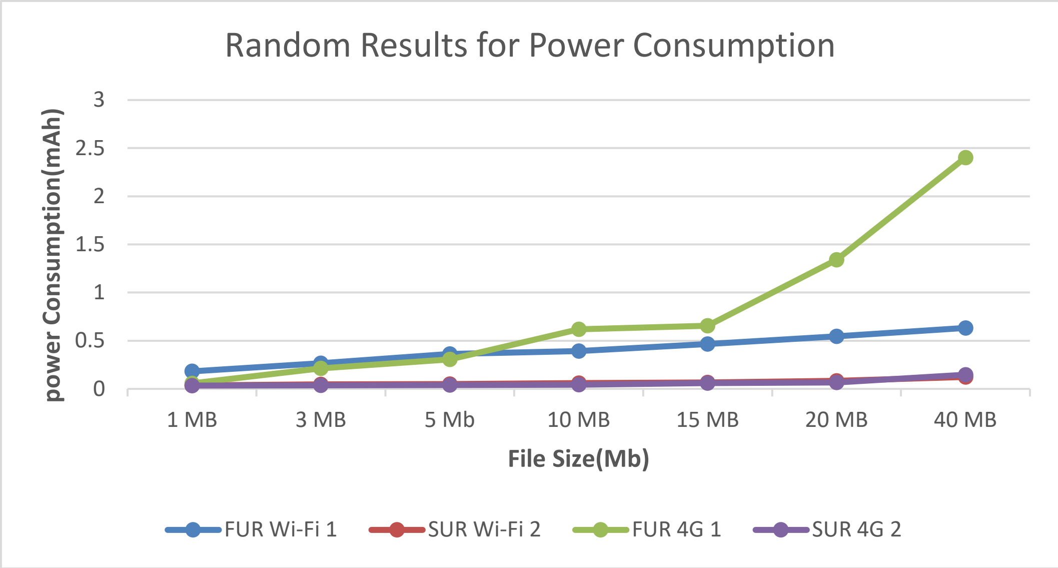


Figure 6. 9 Power consumption for processing files whilst using Wi-Fi, 4G

Figure 6.10 below illustrates the average for the power consumption for FUR and SUR whilst offloading to cloud server using Wi-Fi connection. The results show there is a significant enhancement in the reduction of the power consumption. For example, the enhancement for file size 10 Mb was more than four times in the SUR. In FUR the average for power consumption was .3922 mAh but in the FUR was .05946 mAh. As discussed in figure 6.9, the average of the increase in the power consumption in the FUR was gradually increased with the file size, but in the SUR was slightly increased for the entire file sizes, except from the file size 40 Mb which was noticeably increased compared to file size 20 Mb.

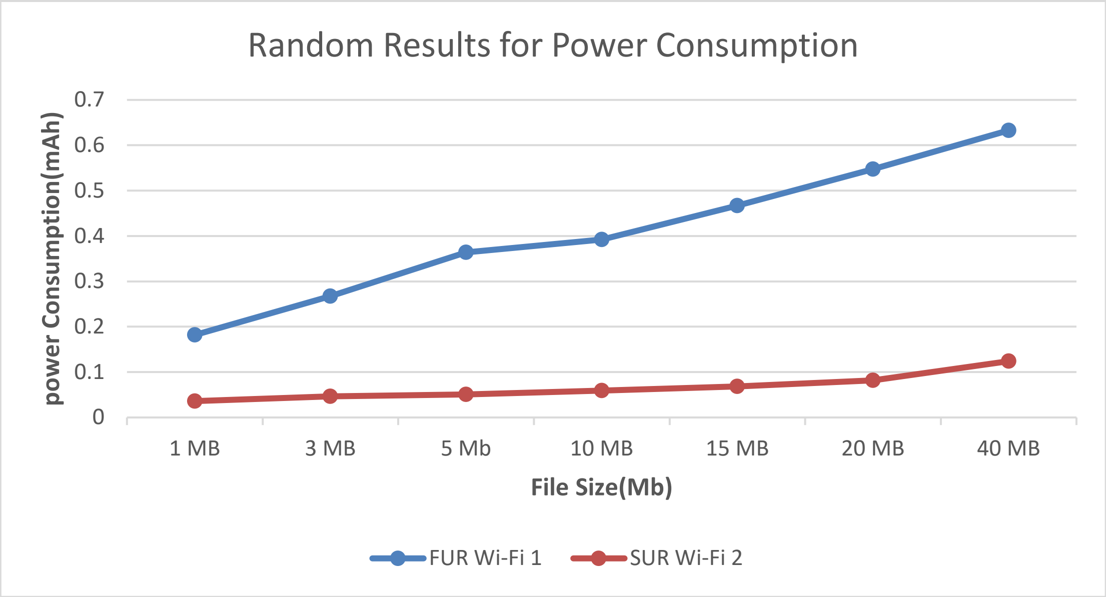


Figure 6. 10 Power Consumption when using Wi-Fi in FUR and SUR

Figure 6.11 illustrates the average of mobile battery consumption for FUR and SUR whilst the various sized files were sent to the Google cloud using the 4G connection. In FUR, the increase of the file size had a significant impact on the power consumption but it was not stable. For the SUR, it is clear there is a substantial reduction in the power consumption, for example, file size 20 Mb in the FUR was 1.3408 mAh and for SUR was .0659 mAh. The increase in the power consumption was slightly apparent for all file sizes.

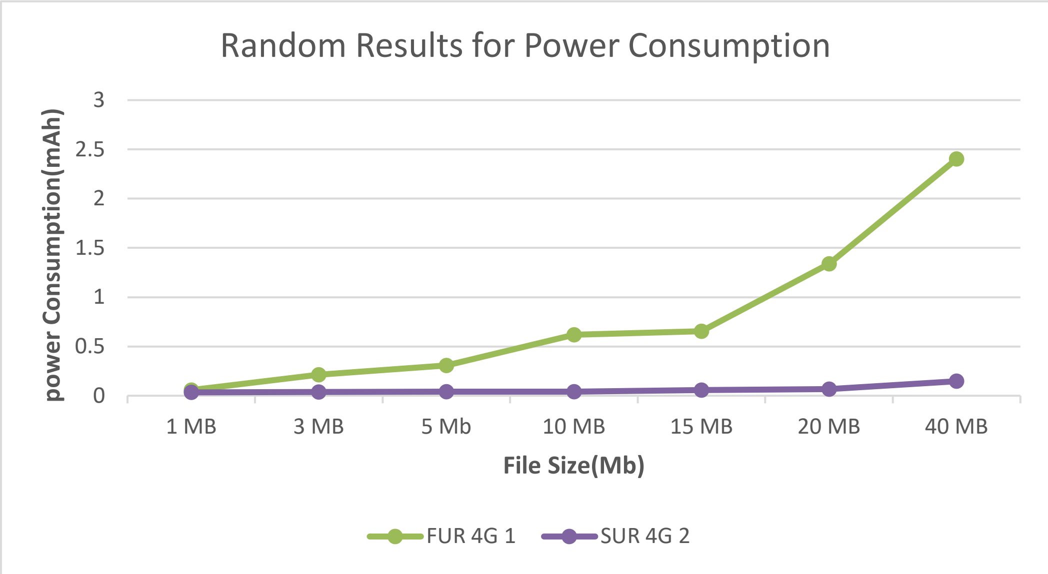


Figure 6. 11 Power Consumption when using 4G in FUR and SUR

Figure 6.12 shows the difference between Wi-Fi and 4G connection in terms of the reduction of the power consumption in SUR mechanism. For both connections, it is evident the increase in the power consumption was stable apart from file size 40 Mb. 4G connection is slightly lower than Wi-Fi, this means it has saved the mobile battery. It is noticeable there is a rapid increase in the consumption of the power between file size 20 Mb and 40 Mb in both connections.

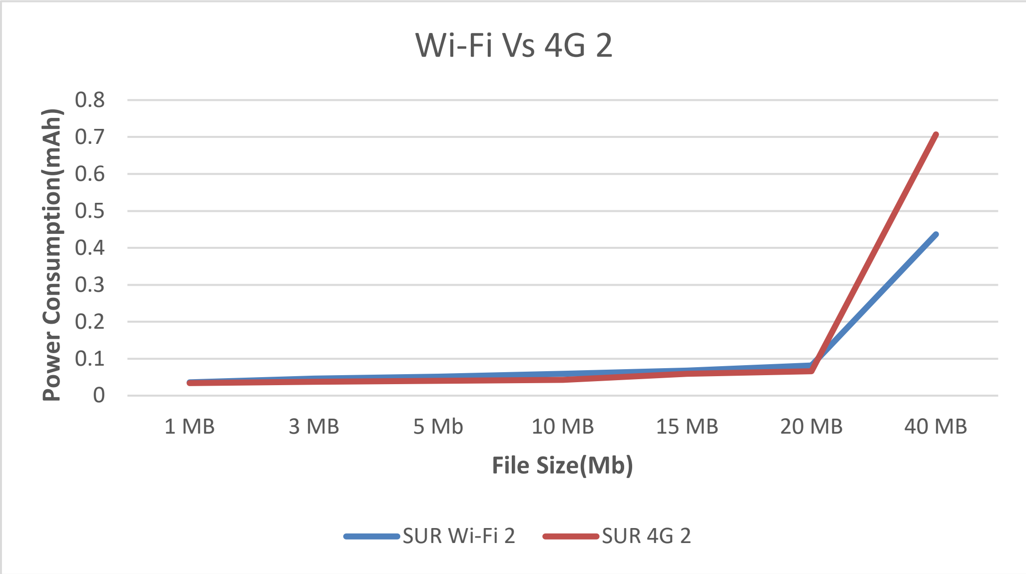


Figure 6. 12 Power Consumption when using SUR for 4G and Wi-Fi

From the previous figures 6.9, 6.10, 6.11, and 6.12, it is noticeable that there is a substantial saving in the power consumption whilst using the new technique SUR in both network interfaces, Wi-Fi and 4G technology. So in this instance, the smartphones users will save more of the mobile battery while processing the files on the cloud using SUR. In addition, it is better to use the 4G connection in the SUR to save more battery as the results have shown.

The list of tables below from 6.4 to 6.10 for the files sizes, 1 Mb, 3 Mb, 5 Mb, 10 Mb, 15 Mb, 20 Mb, and 40Mb illustrate that the five readings, and the average, for the power consumption in both connection Wi-Fi and 4G. The readings were for the two rounds which are FUR and SUR. The Wi-Fi readings were significantly lower for both rounds compared to the 4G connections. In addition, when reviewing the five individual results for power consumption, for each file size, for both network interfaces, it is apparent that the results were fluctuating. In each table there is also an average for the five readings, which gives an overall view of the power consumption by the network interface.

Table 6. 4 Power consumption for file size 1Mb using Wi-Fi and 4G

	Wi-Fi		**4G**	
Reading	**FUR (mAh)**	**SUR (mAh)**	**FUR (mAh)**	**SUR (mAh)**
1	0.228	0.0387	0.0541	0.0261
2	0.118	0.0382	0.0439	0.0341
3	0.1301	0.0468	0.0616	0.0297
4	0.215	0.0314	0.0535	0.0396
5	0.218	0.0251	0.0739	0.0403
Average	**0.18182**	**0.03604**	**0.0574**	**0.03396**

Table 6. 5 Power consumption for the file size 3Mb using Wi-Fi and 4G

	Wi-Fi		**4G**	
Reading	**FUR (mAh)**	**SUR (mAh)**	**FUR (mAh)**	**SUR (mAh)**
1	0.266	0.048	0.153	0.0668
2	0.297	0.0507	0.327	0.051
3	0.287	0.0473	0.253	0.0495
4	0.24	0.0407	0.277	0.0143
5	0.247	0.0443	0.0601	0.00428
Average	**0.2674**	**0.0462**	**0.21402**	**0.037176**

Table 6. 6 Power consumption for the file size 5Mb using Wi-Fi and 4G

	Wi-Fi		4G	
Reading	**FUR (mAh)**	**SUR (mAh)**	**FUR (mAh)**	**SUR (mAh)**
1	0.317	0.0502	0.259	0.0372
2	0.328	0.056	0.366	0.0491
3	0.306	0.0421	0.494	0.0346
4	0.346	0.0541	0.219	0.0487
5	0.521	0.0526	0.2	0.0297
Average	**0.3636**	**0.051**	**0.3076**	**0.03986**

Table 6. 7 Power consumption for the file size 10Mb using Wi-Fi and 4G

	Wi-Fi		4G	
Reading	**FUR (mAh)**	**SUR (mAh)**	**FUR (mAh)**	**SUR (mAh)**
1	0.293	0.0597	0.356	0.0392
2	0.401	0.0617	0.605	0.0205
3	0.363	0.058	0.779	0.0473
4	0.336	0.0597	0.666	0.0526
5	0.568	0.0582	0.539	0.0529
Average	**0.3922**	**0.05946**	**0.589**	**0.0425**

Table 6. 8 Power consumption for the file size 15Mb using Wi-Fi and 4G

	Wi-Fi		4G	
Reading	**FUR (mAh)**	**SUR (mAh)**	**FUR (mAh)**	**SUR (mAh)**
1	0.485	0.199	0.769	0.0666
2	0.528	0.0283	0.54	0.0582
3	0.398	0.0565	0.648	0.0747
4	0.512	0.0364	0.629	0.0513
5	0.411	0.0206	0.693	0.0441
Average	**0.4668**	**0.06816**	**0.6558**	**0.05898**

Table 6. 9 Power consumption for the file size 20Mb using Wi-Fi and 4G

	Wi-Fi		4G	
Reading	**FUR (mAh)**	**SUR (mAh)**	**FUR (mAh)**	**SUR (mAh)**
1	0.876	0.0901	1.69	0.0742
2	0.479	0.0881	1.06	0.068
3	0.415	0.0825	0.924	0.0495
4	0.557	0.0645	1.93	0.0597
5	0.408	0.0836	1.1	0.0781
Average	**0.547**	**0.08176**	**1.3408**	**0.0659**

Table 6. 10 Power consumption for the file size 40Mb using Wi-Fi and 4G

	Wi-Fi		4G	
Reading	**FUR (mAh)**	**SUR (mAh)**	**FUR (mAh)**	**SUR (mAh)**
1	0.567	0.127	3.2	0.174
2	0.548	0.122	2.4	0.167
3	0.634	0.104	2.512	0.151
4	0.731	0.148	2	0.128
5	0.684	0.119	1.9	0.114
Average	**0.6328**	**0.124**	**2.4024**	**0.1468**

6.10.2 Processing Time Results

Another vital parameter has been measured using the deployed mobile application. The parameter is processing time for executing several of files on the cloud server.

The tables and figures below present the results for sending the files and processing them, then receiving the results back from the cloud server. The results will be presented for both FUR and SUR in a millisecond for both network connections, Wi-Fi and 4G. Moreover, there will be a further comparative analysis between the Wi-Fi and 4G network interfaces.

Table 6.11 below shows the average processing time for offloading files over the cloud. The processing time included the time taken to, upload, process, and send the results back to the mobile device. As discussed in section 6.7 there are two rounds, in this table, the processing time will be provided for both of them to evaluate the significance of the proposed SUR. From the average, it is noticeable for all file sizes that there is a large significance in the reduction of the proccing time.

Table 6. 11 Processing time for FUR and SUR using Wi-Fi

File Size (Mb)	FUR- Wi-Fi (Millisecond)	SUR- Wi-Fi (Millisecond)
1	2674.4	345.8
3	4224.8	597.8
5	5601.4	702.8
10	6124.6	1302.8
15	9700.8	1818.2
20	10960.2	1979.8
40	12814.6	3407.6

Figure 6.13 illustrates the average of the processing time for various file sizes. There is an increase in the processing time in both rounds, but in SUR it is noticeable the increase is slight. From the figure below, there is a significant value in the reduction of the processing time in the SUR. For example in file size 20 Mb, the FUR was 10960.2 Millisecond compared to SUR was 1979.8 Millisecond, this means the time is reduced by more than five times.

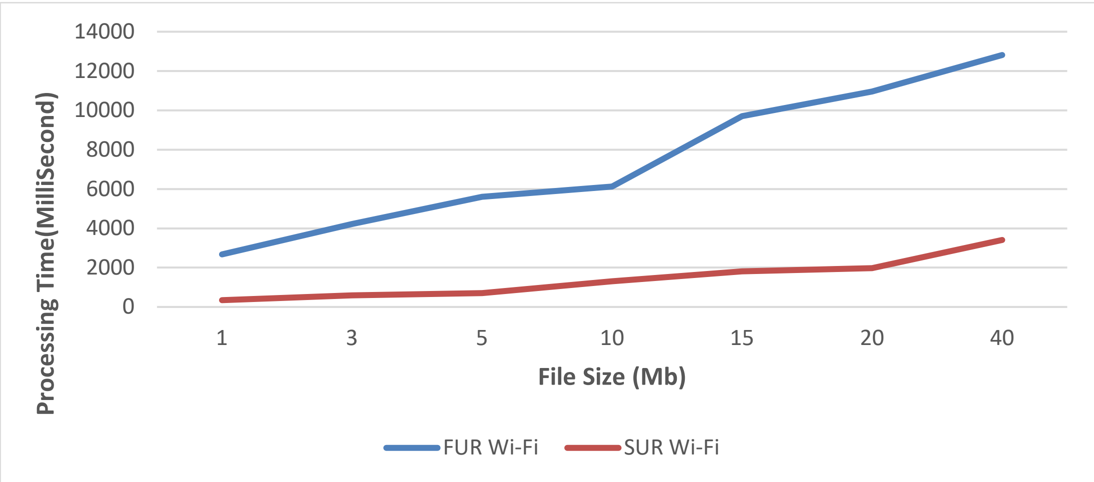


Figure 6. 13 Processing time for FUR and SUR using Wi-Fi

Table 6.12 below presents the processing time for sending files and returning the results back to the mobile users in milliseconds. As discussed in table 6.11, there is a significant improvement in using SUR, this can be seen in the table below for all of the file size scenarios. For example, in the File size 5 Mb, the processing time in FUR was 10013 milliseconds compared to SUR was 875 milliseconds.

Table 6. 12 Average of processing time for FUR and SUR using 4G

File Size (Mb)	FUR- 4G (Millisecond)	SUR- 4G (Millisecond)
1	3745	640.4
3	6696.4	793.2
5	10013	875
10	20573.2	1335.4
15	19529.2	1698
20	34861	2315
40	74307.6	3398.2

Figure 6.14 below shows the results of offloading files to the cloud server. The results illustrate the large difference between FUR and SUR, for example for file size 40 Mb the FUR was 74307.6 milliseconds and for the SUR was 3398.2 milliseconds. It is clear in the SUR the processing time was less than the time in FUR. This proves the value of developing the new mechanism which is SUR.

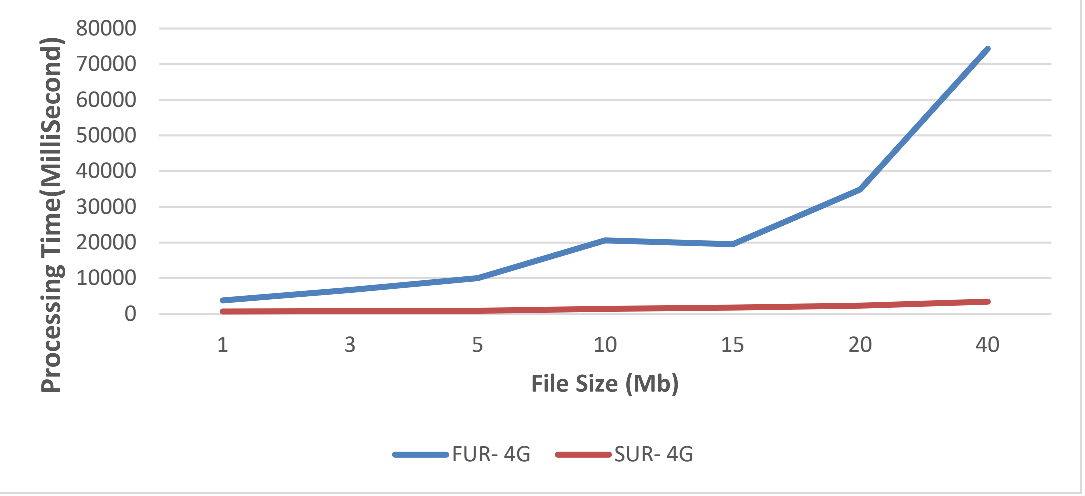


Figure 6. 14 Processing time for FUR and SUR using 4G

Table 6.13 below illustrates the average of the processing time for the various files sizes in milliseconds. The processing time is for two network interfaces 4G and Wi-Fi, this included sending, processing, receiving results from mobile to cloud for both FUR and SUR. Overall, there is a significant reduction in processing time whilst using SUR in offloading for 4G and Wi-Fi, but it is clear in Wi-Fi connection the processing time is less compared to 4G. For example for file size 1 Mb, the processing time was 345.8 milliseconds in Wi-Fi, whilst in 4G was 640.4 milliseconds.

Table 6. 13 Average of processing files for FUR and SUR using Wi-Fi and 4G

	Wi-Fi		4G	
File Size (Mb)	**FUR (Millisecond)**	**SUR (Millisecond)**	**FUR (Millisecond)**	**SUR (Millisecond)**
1	2674.4	345.8	3745	640.4
3	4224.8	597.8	6696.4	793.2
5	5601.4	702.8	10013	875
10	6124.6	1302.8	20573.2	1335.4
15	9700.8	1818.2	19529.2	1698
20	10960.2	1979.8	34861	2315
40	12814.6	3407.6	74307.6	3398.2

From the previous tables and figures, it is noticeable that there is a substantial reduction in the processing time whilst using SUR technique. The large reduction was in both Wi-Fi and 4G connection. This means, the developed technique SUR supported mobile users to send the heavy files and process them on the cloud, then obtain the results which were sent back to the mobile device with minimal delay.

The list of tables below from 6.14 to 6.20 are for the files sizes, 1 Mb, 3 Mb, 5 Mb, 10 Mb, 15 Mb, 20 Mb, and 40 Mb, they present the five readings and the average for the processing time for Wi-Fi and 4G connection for FUR and SUR when offloading to the cloud. The readings were for uploading, processing on the cloud, Sending the results back to the mobile users, and this has been measured in milliseconds.

Table 6. 14 Processing time for File size 1Mb using Wi-Fi and 4G

	Wi-Fi		**4G**	
Reading	**FUR (Millisecond)**	**SUR (Millisecond)**	**FUR (Millisecond)**	**SUR (Millisecond)**
1	2732	281	3346	557
2	1902	276	5261	670
3	2520	385	2887	705
4	2365	429	3589	666
5	3853	358	3642	604
Average	**2674.4**	**345.8**	**3745**	**640.4**

Table 6. 15 Processing time for File size 3Mb Wi-Fi and 4G

	Wi-Fi		**4G**	
Reading	**FUR (Millisecond)**	**SUR (Millisecond)**	**FUR (Millisecond)**	**SUR (Millisecond)**
1	3526	556	7136	747
2	5762	562	6835	775
3	4010	431	6931	705
4	2940	623	6926	875
5	4886	817	5654	864
Average	**4224.8**	**597.8**	**6696.4**	**793.2**

Table 6. 16 Processing time for File size 5Mb using Wi-Fi and 4G

	Wi-Fi		**4G**	
Reading	**FUR (Millisecond)**	**SUR (Millisecond)**	**FUR (Millisecond)**	**SUR (Millisecond)**
1	7127	665	13188	965
2	6289	693	10952	909
3	4891	649	10704	896
4	5376	834	7999	834
5	4324	673	7222	771
Average	**5601.4**	**702.8**	**10013**	**875**

Table 6. 17 Processing time for File size 10Mb using Wi-Fi and 4G

	4G		**Wi-Fi**	
Reading	**FUR (Millisecond)**	**SUR (Millisecond)**	**FUR (Millisecond)**	**SUR (Millisecond)**
1	6558	1114	11405	1336
2	5415	1271	17777	1450
3	6427	1317	21295	1304
4	7112	1289	20945	1327
5	5111	1523	31444	1260
Average	**6124.6**	**1302.8**	**20573.2**	**1335.4**

Table 6. 18 Processing time for File size 15Mb using Wi-Fi and 4G

	Wi-Fi		4G	
Reading	FUR (Millisecond)	SUR (Millisecond)	FUR (Millisecond)	SUR (Millisecond)
1	10130	3057	21104	1586
2	8964	1497	21770	1727
3	9020	1561	18843	1818
4	8627	1361	19038	1707
5	11763	1615	16891	1652
Average	**9700.8**	**1818.2**	**19529.2**	**1698**

Table 6. 19 Processing time for File size 20Mb using Wi-Fi and 4G

	Wi-Fi		4G	
Reading	FUR (Millisecond)	SUR (Millisecond)	FUR (Millisecond)	SUR (Millisecond)
1	12263	2219	37198	2348
2	11161	2045	30639	2608
3	10004	2001	26864	1921
4	13251	1709	41165	2124
5	8122	1925	38439	2574
Average	**10960.2**	**1979.8**	**34861**	**2315**

Table 6. 20 Processing time for File size 40Mb using Wi-Fi and 4G

	Wi-Fi		4G	
Reading	FUR (Millisecond)	SUR (Millisecond)	FUR (Millisecond)	SUR (Millisecond)
1	12361	3426	85466	3643
2	11735	3535	72016	3820
3	11921	3217	74315	3512
4	14359	3751	70158	3019
5	13697	3109	69583	2997
Average	**12814.6**	**3407.6**	**74307.6**	**3398.2**

Table 6. 21: Average of reducing the processing time using Wi-Fi and 4G

File Size (Mb)	FUR and SUR using Wi-Fi	FUR and SUR using 4G
1	87%	83%
3	86%	88%
5	87%	91%
10	79%	94%
15	81%	92%
20	82%	93%
40	74%	95%

Table 6.21 shows the average of reduction of the processing time whilst offloading files to Google cloud. The average of the processing the files was for FUR and SUR when using Wi-Fi and 4G connections. From the table it is clear, that there is a large reduction in the processing time when using SUR. For example, the average of the saving time for the file size 10 Mb was 94% whilst using 4G connection compared to Wi-Fi which was 79%. Overall this means, using the 4G connection in the SUR is more efficient to save the time compared Wi-Fi.

6.11 Discussion of the Results

The benefits of doing these experiments were to resolve the weaknesses outlined in the previous chapter, through the implementation of new features in the mobile applications. There was a need to have a second server and to save files on the cloud. Then to link these features with network interfaces to investigate if one of them was more valuable while the file is already on the server and to process over the cloud, known as SUR. In addition, the fact that there is the availability of a second server means, that the need of re-uploading due to 'server busy' is eliminated and therefore automatically limits the negative impact on the battery and processing time.

The results were reviewed and analysed, they indicated that through the implementation of the new two features battery usage and processing time were improved whilst using both network interfaces Wi-Fi and 4G. This was recorded to be more than 80% for the SUR compared to FUR in most cases. For example, when the file size was 3Mb, the reduced power consumption for the SUR is 83%. For processing time, there is a significant decreasing for example, in the file 10 Mb, the time is reduced to 94% when using the proposed technique.

The results illustrate that overall during FUR, Wi-Fi was more efficient in reducing power consumption for big files apart from files 1 Mb, 3 Mb, and 5 Mb. In contrast, in relation to power consumption, 4G is the most improved whilst using the new feature, SUR. For example, for file size 10 Mb the reduction of the power consumption was 93% on 4G, compared to Wi-Fi which was 85%. However, the performance for Wi-Fi in FUR was significantly lower than 4G, which shows that although the improvement between two rounds was 93% for 4G, it still used more battery than Wi-Fi. In addition, this pattern of efficiency and improvement for the network interfaces are parallel for processing time.

7 Chapter 7: The Mathematical Model

7.1 Introduction

In this chapter, the mathematical model will be developed and validated for the power which is consumed during network activities when offloading tasks from mobile devices to cloud server. In this model, the 4G network interface will be considered. Samsung Galaxy Tab 4 T335 will be used in each scenario. Developing a mathematical model to measure the mobile devices battery usage is important to validate the model results with the experimental model.

Developing a mathematical model for the energy consumption which is consumed by network activities is essential to make task offloading useful. The offloading decision depends on the estimation of the power consumption cost, and the file size. This will be modelled mathematically, for offloading the files from the mobile device to the cloud and process them.

Channel Quality Indicator (CQI), is vital in depicting the channel information to allocate Proper Modulation and Coding Scheme (MCS). On the other hand, obtaining CQI values to Transmission Time Interval (TTI) can increase the error rate. This means, to provide a reliable and accurate CQI with minimum overhead is a complex task (Abdulhasan, et al., 2014).

CQI is the information for the user equipment to be sent to the network. It is an indicator to measures the communication quality either good or bad. It suggests:

1. The current quality of the communication channel.
2. Data to the user equipment which can be obtained and transfer to throughput

If the channel quality is poor the user equipment reports that there is a high CQI, the network would send a large transport block size according to the value of the CQI. This occurs because there is a high probability that the user equipment failed to read files. Due to this error in the UE side there is a negative acknowledgment which is sent to the network, which means the network has to retransmit the files, which causes consumption to the resources.

The cellular networks such as 3G and 4G, have Radio Resource Controller (RRC) to manage the communication between the mobile user’s devices, for example the UE, and the network. The purpose of the RRC is to offer a high performance for the mobile connectivity by improving the network throughput, and decreasing the UE power consumption, and signalling latency. So, it is noticeable, the RRC has a crucial impact on the power consumption for the smartphones devices. Two cases will affect the latency and the power consumption in different scenarios, one is connected when exchanging the data, which increase the network latency and the link setup signalling. The other has allowed latency but affect the UE battery quickly which means more power consumption. To overcome these two cases, the RRC initiate a trade-off between them and chooses one of them, and this based on different conditions (Altamimi, et al., 2015).

The structure and specifications for RRC are described and developed in the 3GPP standards. There are two major states for RRC, the first is RRC-Connected, the connection data is created where is the dedicated resources are assigned to the UE, in this state the power is high. When the UE gets a notification from the network broadcast the transition to RRC-Connected has occurred and the data will be received. To create the connection, the UE sends a connection request to the network by the signalling procedures. The second major state is RRC-Idle, where the radio resources are allocated to the UE, and the power state is low. In this state, the UE adjusts to the shared control channel. (Grigorik, 2013). Furthermore, the operation for the aforementioned of the RRC states, the entire energy was consumed whilst transferring the data, which consists of three parts, tail energy, data transfer, and promotion signalling.   (Qian, et al., 2011) And (Huang, et al., 2012).

Universal Mobile Telecommunications System (UMTS) Architecture: is a third generation mobile cellular system for the network, and it is based on GSM standard. In UMTS the key factor that affects the network energy efficiency and application performance is the RRC. This system consists of three components: one is Universal Terrestrial Radio Access Network (UTRAN), second, Core network (CN), and finally, the handset which is the mobile device. TRAN and CN consist of two components which are the node for the base station, and Radio Network Controller (RNC). The purpose of RNC is to govern the element in UTRAN and to control multiple nodes. The central CN is the backbone of the cellular network (Qian, et al., 2011).

The following factors impact how the UE measures the CQI:

1- Signal to noise ratio (SNR)

2- Signal to Interference and Noise Ratio (SINR)

3- Signal to Noise and Distortion Ratio (SNDR)

The objectives of a good network are:

1- Signalling delay: low
2- Power consumption: low
3- Throughput: High

There are three states in RRC

1- **IDLE state (low power):** User Equipment (UE) tunes to control channel counted (control traffic) Connected state: higher power radio+ radio resources tune to UE.
2- **Dedicated Chanel (DCH)**
3- **Forward Access Chanel (FACH)**

The aforementioned states will be discussed in the following section:

The connection occurs when there is a chance to send while the buffer is low. Hence the request is sent for the connection through the signalling procedures. When the mobile device is switched on, this means IDLE state by default. Till this stage, the mobile device does not establish an RRC connection with the RNC. Thus, no radio resource is assigned, and the mobile device could not send any of user data. So, the power consumption of the mobile radio interface is nearly zero at ADLE state. For the DCH state, the connection is established at the RRC state and the mobile device is generally allocated and dedicated. The radio power consumption for DCH is the highest which is (600 to 800 mW). The last state is FACH, and in this state the RRC link connection is established but there is a low speed for both uplink and downlink. This state is created for an application with low data throughput rate (Qian, et al., 2011). State Transitions: there are two kinds of RRC state transitions which are discussed by Qian, et al., (2009) as follows:

Promotion State: (IDLE→DCH, FACH→DCH, and IDLE→FACH).

Demotion State: going in reverse directions.

Promotions (demotions) switch from a state with lower (higher) radio resource and radio power consumption to another state requiring more (less) radio resources and power. Figure 7.1 below shows the RRC state mechanism for Promotions and Demotions.

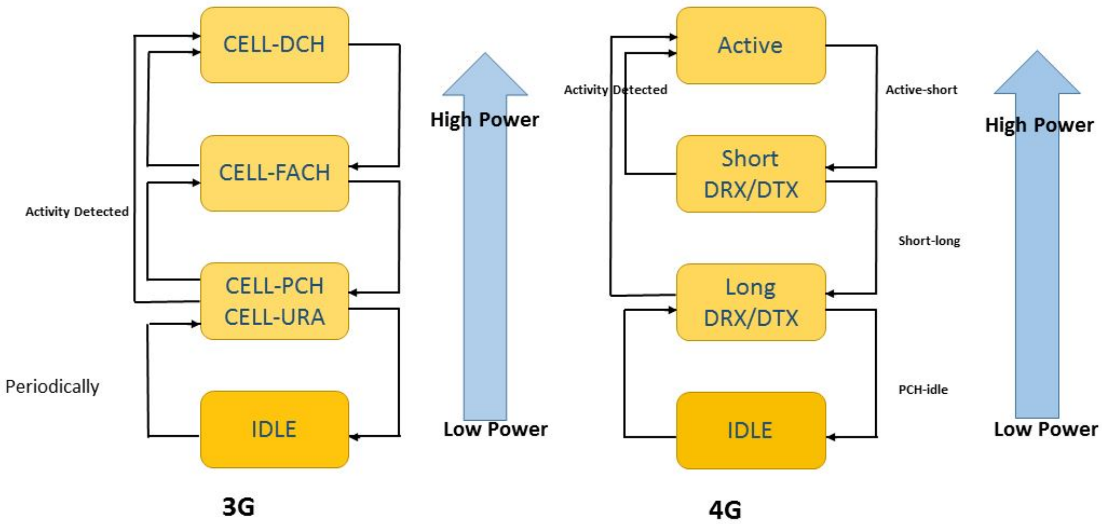


Figure 7. 1 3G and 4G RRC state mechanism

Form Figure 7.1, the total energy which is consumed in RRC state consists of:

1. Promotion signalling energy
2. Data transfer energy
3. Tail energy

General energy is

$$E_{3G\,/\,4G} = E_{PS} + E_{\mathrm{trx}} + E_{mail}. \ (\mathrm{Altamimi, et\ al., 2015})$$

Where

Ps: Signalling Power

E_{trx}: Data Energy

E_{mail}: Tail time Energy

Energy= Power x time

Where

Power: smartphone power in W

Time: Smartphone time in second

$$E_{3G\,/\,4G} = \left(P_{ps}\ x\ T_{ps}\right) + (P_{trx}\ x\ T_{trx}) + (P_{tail}\ x\ T_{tail})$$

E_{ps}, E_{tail} are constants for each smartphone hence E_{ps} and E_{tail} are computed separately.

For P_{trx} x T_{trx}

P_{trx} is constant as power, but T_{trx} is variable depending on size of data (F) and the data rate as:

T_{trx} = F (amount of data in bits) ÷ R_{trx}

*Example: T_{trx} = (1000 bytes * 8) ÷ 2Mb*

In wireless networks, we have limited resources (low spectrum), high error rate, higher delay as compared to wired, hence we work hard to increase spectrum and reduce delay, due to the BER limitation. New protocol ARQ and hybrid AQR are useful to protect the lower layers from received errors. Transfer Control Protocol (TCP) is an end-to-end protocol and hence is limited by the delay.

The achieved data rate R_{trx}= min (R_{tcp}, $R_{3G/4G}$), as R_{tcp} and $R_{3G/4G}$ are limits of the rate due to TCP and wireless network.

TCP rate= CWD (Congestion window)/ RTT, but $R_{3G/4G}$= rate achieved at TCP layer which is limited by the rate of the lower layers (physical, MAC, PDCP), which is adaptive by maximising spectrum utilisation. Adaptation is implemented for each (TTI) Transmission Time Interval. In adaptation different modulation and coding systems are used using the (RSS) received signal strength and S/N. the receiver is a source of information by sending to TX, the current challenge condition using Channel Quality Index (CQI), which is calculated from RSS x S/N. Then, Tx selects the MCS according to the mapping from CQI and the user equipment (UE) energy to Modulation and Coding Scheme.

At slow state there is high power consumption at limited window size (IWD) until it reaches CWD. The mobile uses too much energy at this stage of TCP slow-start hence,

$$T_{trx} = T_{ss} + \frac{F - F_{SS}}{R_{trx}}$$

F_{ss}: data transferred during the slow start.

T_{SS} is the time of slow start before it reaches CWD from (IWD) window size. Where

T_{SS} = .5 ms x 3.5 = 1.75 ms.

TCP: data size = 1.5KB x 1000 x 8 = 12000 bits.

F_{ss} = Tcp size x 3.5 = 12000 x 3.5 = 42000 bits

R_{trx}= 10 ÷ (. 5 x 10) = 20000 b/s

$$T_{SS} = 1.5Kbyte\ x\ log_2 \frac{CWD}{IWD}$$

Where log_2 (CWD/IWD) is the expo growth of the window size which is usually 2

For examples

$$T_{trx} = T_{ss} + \frac{F - F_{SS}}{R_{trx}}$$

$$F_{SS} = 1.5Kbyte\ x\ log_2 \frac{CWD}{IWD}$$

1.5 x 1000 x 8= 12000 bits

$log_2\ (2/1) = 2^1$

F_{ss}*=12000 x 1= 12000 bits*

$$T_{ss} = R_{TT}\ x\ log_2 \frac{CWD}{IWD}$$

$$T_{ss} = (1.25ms\ x\ 4)\ x\ 1\ =\ 5\ ms$$

$$T_{trx} = 5ms\ + \frac{(\text{F(variable)} - 1.2 \text{ x } 10^4)}{5.4\ x\ 10^7}$$

Energy $_{3G,\,4G}$ *= Ptrx * Time*

Energy $_{3G,\,4G}$ *= 12 (watt) x 5.94 = 71.28 Joules*

Example of the T_{trx} for different file sizes files sizes:

File Size: 1 Mb:

$$T_{trx} = 15 + \frac{(\text{File siz} - 3.6 \text{ x } 10^4)}{5.4\ x\ 10^7}$$

$$T_{trx} = 15 + \frac{(8 \text{ x } 10^6\ - 3.6 \text{ x } 10^4\ )}{5.4\ x\ 10^7}$$

$$T_{trx}\ = 0.1474$$

File Size: 3 Mb:

$$T_{trx} = 15 + \frac{(\text{File siz} - 3.6 \text{ x } 10^4)}{5.4\ x\ 10^7}$$

As File Size: 3x8 M = 24 M = $2.4 \text{x} 10^7$

$$T_{trx} = 15 + \frac{(2.4 \text{x } 10^7\ - 3.6 \text{ x } 10^4\ )}{5.4\ x\ 10^7}$$

$$T_{trx}\ = 0.4437$$

File Size 5Mb:

$$T_{trx} = 15 + \frac{(\text{File siz} - 3.6 \text{ x } 10^4)}{5.4\ x\ 10^7}$$

$$T_{trx} = 15 + \frac{(4\text{x } 10^7\ - 3.6 \text{ x } 10^4\ )}{5.4\ x\ 10^7}$$

$$T_{trx} = 0.74$$

File Size 10Mb:

$$T_{trx} = 15 + \frac{(\text{File siz} - 3.6 \text{ x } 10^4)}{5.4\ x\ 10^7}$$

$$T_{trx} = 15 + \frac{(8\text{x } 10^7\ - 3.6 \text{ x } 10^4\ )}{5.4\ x\ 10^7}$$

$$T_{trx} = 1.48$$

File Size 15Mb:

$$T_{trx} = 15 + \frac{(\text{File siz} - 3.6 \text{ x } 10^4)}{5.4 \; x \; 10^7}$$

$$T_{trx} = 15 + \frac{(1.2 \text{ x } 10^8 \; - 3.6 \text{ x } 10^4 \;)}{5.4 \; x \; 10^7}$$

$$T_{trx} = 2.22$$

File Size 20Mb:

$$T_{trx} = 15 + \frac{(\text{File siz} - 3.6 \text{ x } 10^4)}{5.4 \; x \; 10^7}$$

$$T_{trx} = 15 + \frac{(8\text{x } 10^7 \; - 3.6 \text{ x } 10^4 \;)}{5.4 \; x \; 10^7}$$

$$T_{trx} = 2.96$$

File Size *40Mb*:

$$T_{trx} = 15 + \frac{(\text{File siz} - 3.6 \text{ x } 10^4)}{5.4 \ x \ 10^7}$$

$$T_{trx} = 15 + \frac{(3.2\text{x } 10^8 \ - 3.6 \text{ x } 10^4 \ )}{5.4 \ x \ 10^7}$$

$$T_{trx} = 5.92$$

7.2 Validation Results

7.2.1 One- Sample t-Test

The *t* test it is any statistical hypothesis test. A *t*-test is usually applied when the test statistic would follow a normal distribution. One of the benefits of *t* test it is to determine if there is a significant difference between two groups, which means there is a correlation between the results or they are similar.

In this research, two sets of data were figured out for the power consumption whilst offloading files from the mobile device to the cloud server as follows:

1. Results when using the developed mobile application using the 4G connection.
2. Results from the designing of the mathematical model.

To get the output from t test is by entering the readings which are already accomplished by the developed mobile application, and then enter the results from the designed mathematical model. Figure 7.2 below shows the processes of analyse one sample test. The readings: .15, .12, .17, .16, .21 are form the experimentation and the number .1244 is from the mathematical model.

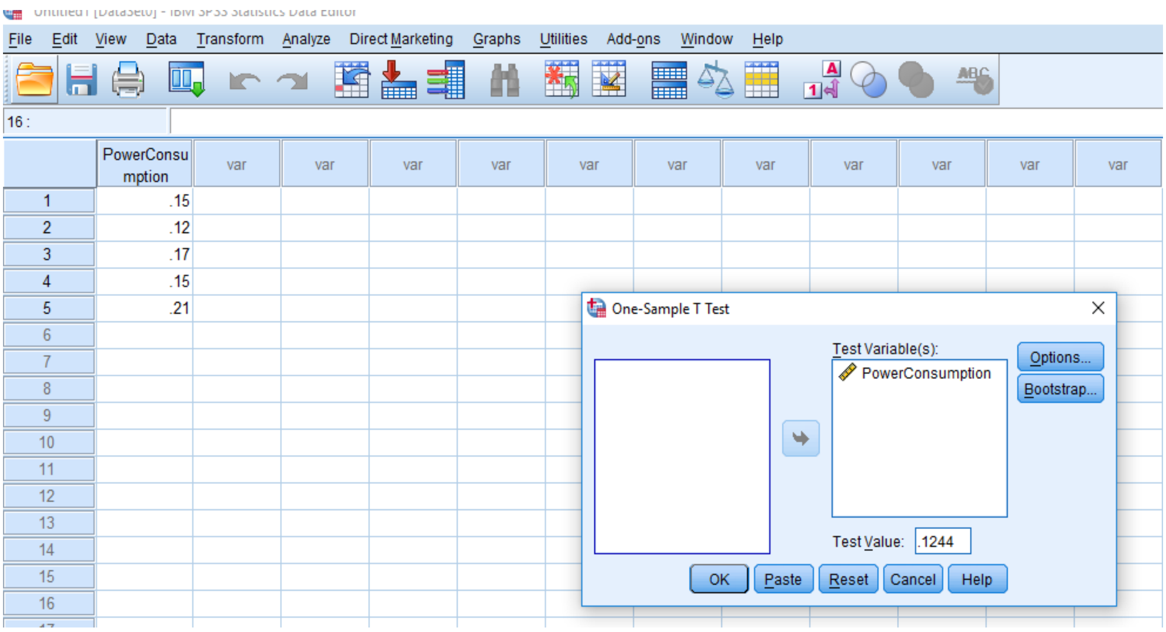


Figure 7. 2 *t* test to analyse one sample test

Table 7.1 below shows the results from *t* test model after adding the two sets of the data to *t* test statistical model.

Table 7. 1 Compare the experimental results with t test

File Size (Mb)	Sig	t	Means
1	.069	2.47	1.590
3	.173	1.657	.5942
5	.205	1.510	.8536
10	.077	2.374	1.7224
15	.668	-462	1.8217
20	.088	2.247	3.7233
40	.058	2.629	6.6727

7.2.2 Mathematical Model

The table below shows the results for power consumption for the mathematical model and the experimental model. The results are for processing various files between 1Mb to 40 Mb over Google cloud platform using the 4G connection. From the table below there is a correlation between the two models, for example, the power consumption in the mathematical model for file size 15Mb is 1.87 mAh, compared to experimental model is 1.8216 mAh.

Table 7. 2 Compare the power consumption for the mathematical model, experimental model

File Size (Mb)	Power Consumption Mathematical model (mAh)	Power Consumption experimental model (mAh)
1	0.1244	0.159
3	0.3744	0.59418
5	0.6244	0.8536
10	1.249	1.7223
15	1.87	1.8216
20	2.49	3.7233
40	4.99	6.6727

Figure 7.3 below illustrates the power consumption for processing file for both models, mathematical and experimental model. The power consumption for the experimental model is a slightly higher than the power consumption in the mathematical model. But it is clear from the figure the power consumption is almost the same in both models. Furthermore, there is a gradual increase in the power consumption for both models. In real experiments, there is more energy consumption due to signalling which are not included in the mathematical model.

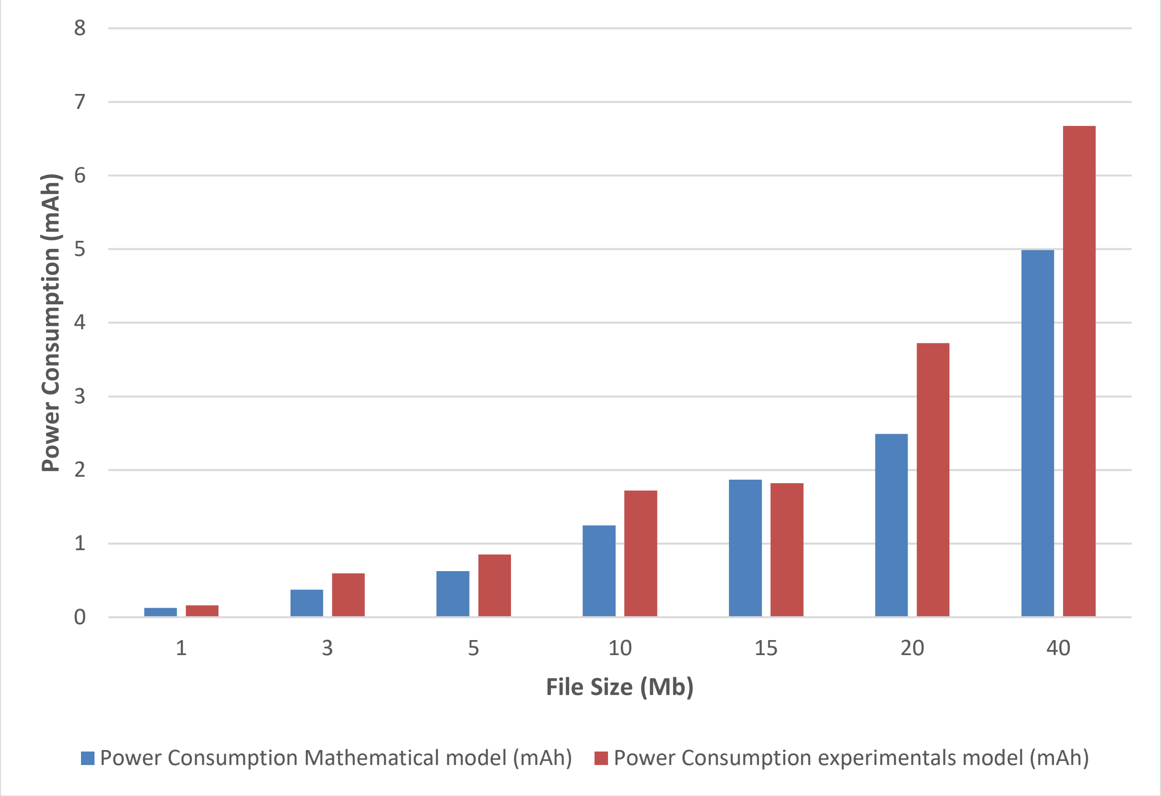


Figure 7. 3 Power consumption for mathematical and experimental

7.3 Previous Studies Results

According to the experiments (MECCA) system the mobile could save the power around 61% compared to (Altamimi, et al., 2012) work, so, the proposed new model saves more power. This means the new offloading technique succeeds to save mobile battery.

In the end, although of that JADE system was reduced the power consumption to 39% compared to MECCA system 61% which focused on offloading technique while taking into account network interfaces cost.

It is found that by using this algorithm, power consumption is reduced to 61% which is more enhanced than previous work of 30% to 38%. Table 7.3 below shows the results of the power consumption from the previous studies.

Table 7. 3 Previous Studies Results

Model Name	**Power Consumption (mW)**	**Connection**
(Qian & Andresen, 2015)	39%	Wi-Fi
(Altamimi, et al., 2012)	30%-70%	Wi-Fi
(Fekete, et al., 2013)	36%-38%	3G, Wi-Fi
MECCA	61%	Wi-Fi
MECCA	90%	4G

8 Chapter 8: Conclusion

8.1 Overview

Recently, the smartphones have an increasingly high demand from end users, and become one of the most vital devices in their hands, due to the use of innovative functions and applications which they offer, such as YouTube, online gaming, GPS, image processing, and different kinds of application for example processing files. The revolution in the internet, and network interfaces technology support the popularity of the smartphones. However, the smartphones have some limitations to process different kinds of applications and files. Nowadays, calculation and complex tasks require a powerful level of resources to process files efficiently. This means there is a requirement to save the mobile battery as much as possible, with minimum delay. Until now, smartphones have obstacles in the aforementioned issues. To outweigh that, this research investigates the limitations of smartphones in terms of battery and delay, and determines how to reduce the power consumption and processing of smartphones with preserving the performance.

This research has identified the most important issue for smartphones, and mobile application will be developed including decision engine to check the power consumption and the processing time based on the file size, while differentiating the kind of network interface used.

Mobile computing and cloud computing are two of the most and significant influential technologies that are expected to change the style of computing in the future. The two combined offer an exceptional chance to provide intensive computation services to the end user and content-rich and highly portable. The offloading technique was able to preserve mobile energy and enhanced the computing functionality of mobile devices.

Offloading tasks propose to support and develop mobile Android models to reduce the power consumption and processing time with the assistance of the cloud computing. Offloading method is one of the efficient ways which promises to support smartphone limitations. Different scenarios among three experiments chapters were conducted to indicate that offloading heavy tasks such as; calculation using CSV file, from the mobile device to cloud computing is beneficial. In addition, the research describes a robust method to decide if resource-intensive tasks should be executed on a mobile device locally or, on the other hand, they should be offloaded by sending the implied files to the cloud to be processed by a more powerful server. In this work, an extensive number and series of experiments were conducted to check the possibility of using offloading technique between mobile and cloud, in addition, to reduce the power consumption and processing time of workflows installed on smartphones.

In chapter four, offloading mobile Android model was created and installed on the smartphone, to process CSV files locally, or send them from mobile device to Google cloud platform. Additionally, it calculated the time which is used in the processing files and estimated the power consumption. The results proved that offloading technique reduced the power consumption and minimised the processing time compared to processing the files on the mobile, especially in the big files sizes. This means that the Google cloud platform saved the battery and time by nearly half using Wi-Fi connection. Furthermore, the most efficient method is to execute big files in terms of better performance and least time is by offloading. However, since the offloading technique is not the appropriate place to execute, in most cases it is much better to execute files on the node. In this issue, there is a trade-off, with power limitation in the local node. Therefore, in this chapter, an innovative algorithm is proposed based on the file size. The file size is measured then the decision is taken for the execution of the file, whether locally on the node, or offloading by sending the file to the core cloud. The most important issue is to preserve time while serving the file. Also important, especially for small nodes, is to preserve the energy limitation for big files, since power consumption will be very high. The cost of the power consumption, execution time, and file size for the core cloud, and local node are calculated to denote an input to the execution decision.

In chapter five, network interfaces were involved in the experiments to investigate if there is a crucial impact on the mobile battery and processing time when the model decide to offload to Google cloud server. In addition, increasing in the files sizes to find out if this facto has an impact on the parameters being measured. Again, the cloud server is beneficial for mobile users in terms of delay and save the battery. The results illustrated that it is beneficial for the smaller files sizes to be processed locally.

According to network interfaces, different scenarios and file sizes are conducted to estimate the power consumption and processing time of smartphones, whilst processing locally and when tasks are offloaded using WLAN, 3G, and 4G connections. The results demonstrated that offloading files using 4G technology reduced the power consumption by up to 90% compared to Wi-Fi 61%. From these results, it revealed that the network interfaces had a significant impact on the two parameters. Moreover, an algorithm was created to decide when to process locally or offload based on the two factors which are, file size and network interfaces. So, in this case, there is a trade-off considering the two factors. Compression model was developed to reduce the processing time. In this case, the file should be compressed then offload it to the cloud using 4G. The results show that the newly developed model decreased the processing time whilst performing the files. The average was between 25% - 85%.

Overall, for chapter five, the model is developed to reduce the power consumption and processing time. The algorithm is evaluated with a set of scenarios using Wi-Fi, 3G, and 4G. The result is summarized below:

- Offloading technique using 4G over performed than local, Wi-Fi, and 3G.
- Offered a novel model for connectivity, supports local and remote processing in terms of particular parameters such as delay and power consumption.
- When the file size is small, there is no much difference in power consumption between local and Wi-Fi.
- When the file size is increased, the power consumption using local is much higher than the Wi-Fi.

The aforementioned were tested by MECCA (Mobile Energy Cloud Computing application is the model used for the whole scenarios.

In chapter six, another server was added to serve more requests which were sent from mobile users, as well save files on the server instead of re-uploading files. The results illustrated that the new features saved the mobile battery and reduced the processing time. Moreover, in SUR, the improvements in reducing the power consumption were noticeable whilst using the 4G connection. In contrast, the enhancements in processing time were found in Wi-Fi compared to 4G. Overall, the results offer an insight into the offload possibility for different characteristics. Additionally, the results indicate how the network interfaces can impact on the objective functions such as power consumption and processing time. Finally, these results illustrate that the new cloud offloading algorithm can significantly improve the power consumption and the processing time.

It is worth noting that these experimentation work completed in one geographical location and this may be a limitation that should be addressed in further studies.

8.2 Reflection on the Research

At the outset of the research process there were various questions which required further investigation to form the research objectives which are seen in chapter 1 section 1.4. The following sections will reflect on the outcomes of this research through an explanation of the conclusion of each question in sequence.

1. To process tasks in high performance there is a requirement for mobile application to use the cloud server. In this research a mobile application was created to enable the mobile users to use powerful resources. The findings of this it can be initially seen in chapter 4, which was the first experiment in this research. Overall, from the experiments the processing files on the cloud server is more efficient than local on the mobile device.

2. The end users save the mobile battery and time by offloading the heavy tasks to the cloud server. This was illustrated in chapters 4, 5, and 6. One example from chapter 4 illustrates that the processing time for file size 9MB is 14.05 seconds which is double in comparison to the cloud which is 7.2 seconds. A further example, from chapter 5, illustrates that offloading to the cloud using 4G at file size 40MB is 1476 mW, which is extremely more efficient than offloading locally on the device which used, 21130.4 mW.

3. The developed MECCA mobile application which was introduced in chapter 5, had integrated decision engine which decides the most efficient method in where to process the files in relation to battery usage impact. This can be based on different factors which were file size and network interface.
4. In chapter 5 there was an introduction of compressing files with the aim of reducing processing time. The evidence from the file illustrated that the compression of files was time efficient as 20MB was 24.2 seconds in comparison to the non-compressed file which was 41.5 seconds. Another example is that at file size 3MB the processing was reduced by half when it was compressed (7.8 seconds to 3.3 seconds). Additionally, this was addressed further in Chapter 6, with the option of the file being saved on the server to prevent time delays when offloading the file more than once. When processing for the SUR there were significant reductions in processing time. For example for file size 10MB there was a reduction of 94 % on 4G and 79% on Wi-Fi, which clearly demonstrates the efficiency of the extra feature. In addition, in Chapter 6 the results illustrated that the new technique supported the reduction of power consumption. For example, in file size 5MB the reduction in the SUR was Wi-Fi and 4G to 86% and 87% in respectively, in comparison to the initial upload.
5. Through each experiment there was consideration of network interface and file size to enable a clear analysis of the impact on the two parameters measured, battery usage and processing time. In addition, the new mechanism was invented to prevent re-uploading files which adversely affects the aforementioned parameters.

For the above-mentioned reflections, the methods which were used to answer the questions were conducted experimentally in different scenarios. The objectives of the research detailed in Chapter 1 Section 1.5 required various scenarios to be tested and analysed. The research objectives were successfully answered through a clear analysis of the three main stages in of experimentation which developed through from chapters 4 to chapter 6. Additional changes were made to the experimentation process to enable the testing to delve deeper into the investigation and to obtain a robust set of statistical data based on different limitations that the mobile user experiences. These factors included the consideration of network interfaces, files sizes and the creation of a bespoke application which was altered to do specific commands such as compression of files and the saving of files to save re-uploading. Through the consideration of these research objectives the research has investigated and analysed statistical data. This data supports the hypothesis of this research, which is that it is more efficient to use the bespoke application to reduce the battery usage with minimum delay.

8.3 Future Work

This research raises a number of areas which merit a further investigation. The proposed mobile application was only focused on two factors which are battery usage and processing time based on the file size. The previous factors were chosen to help the mobile user when they use the smartphones, to save more battery and minimum delay.

The proposed mobile application opens the door for other solutions which could help in energy consumption and other parameters, the solutions such as categorise the service or the tasks to different classes, for example, simple or complex and then link them with the proper network interface. In addition, buffering the data when the 4G cut off and connect it with Wi-Fi, once the 4G connection switches back to it. It is worthwhile for the mobile user to have the opportunity to update the file when it is saved on the server instead of uploading the entire file. Furthermore, in this research there was no emphasis on security aspects of cloud computing because the offloading was on the dedicated server. Another factor to consider for future study is other operating systems such as IOS which are suitable for Apple devices.

8.4 List of Publications

- **Aldmour, R.**, Yousef, S., Yaghi, M., Tapaswi, S., Pattanaik, K.K. and Cole, M., 2016. New cloud offloading algorithm for better energy consumption and process time. International Journal of System Assurance Engineering and Management, pp.1-4.

- **Aldmour, R.**, Yousef, S., Albaadani, F. and Yaghi, M., 2017, January. Efficient Energy and Processes Time Algorithm for Offloading Using Cloud Computing. In International Conference on Global Security, Safety, and Sustainability (pp. 364-370). Springer, Cham.

- **Aldmour, R.**, Yousef, S., Yaghi, M., and Kapogiannis, G., 2017. MECCA Offloading Cloud Model over Wireless Interfaces for Optimal Power Reduction and Processing Time. Cloud and Big Data Computing CBDCom 2017 San Francisco.

-

References

Alasuutari, P., Bickman, L. & Brannen, J., 2009. *The SAGE Handbook of Social Research Methods.* London, UK: Sage Publications Ltd.

Bahwaireth, K. et al., 2015. *Efficient techniques for energy optimization in Mobile Cloud Computing.* s.l., International Conference of Computer Systems and Applications (AICCSA), 2015 IEEE/ACS 12th .

Burns, R., 2000. *Introduction to Research Methods.* First ed. London, UK: Sage.

Duan, Q., Yan, Y. & V. Vasilakos, A., 2012. A Survey on Service-Oriented Network Virtualization Toward Convergence of Networking and Cloud Computing. *Transactions on Network and Service Management-IEEE,* 9(4), pp. 373 - 392.

Hashem, I. et al., 2015. The rise of “big data” on cloud computing: Review and open research issues. *ELSEVIER-Science Direct,* Volume 47, p. 98–115.

Hung, S.-H.et al., 2011. *An Online Migration Environment for Executing Mobile Applications on the Cloud.* Seoul, International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, IEEE.

McBurney, D. & White, T., 2009. *Research Methods.* Eighth ed. London, UK: Cengage Learning..

Wang, C.-M., Wu, Y.-S. & Chung, H.-H., 2017. *FUSION: A unified application model for virtual mobile infrastructure.* Taipei, Taiwan, Conference on Dependable and Secure Computing, IEEE.

Zhan, Z.-H.et al., 2015. Cloud Computing Resource Scheduling and a Survey of Its Evolutionary Approaches. *ACM,* 47(4).

References

Huang, J. et al., 2012. *A close examination of performance and power characteristics of 4G LTE.* Low Wood Bay, Lake District, UK, international conference on Mobile systems, applications, and services, ACM.

Papadimitriou, D. et al., 2012. Measurement-based research: methodology, experiments, and tools. *Computer Communication Review, ACM*, 42(5), pp. 62-68.

Qian, F. et al., 2009. *TCP revisited: a fresh look at TCP in the wild.* Chicago, Illinois, USA, SIGCOMM conference on Internet measurement, ACM.

Shiraz, M. & Gani, A., 2014. A lightweight active service migration framework for computational offloading in mobile cloud computing. *The Journal of Supercomputing- ACM*, 68(2), pp. 978-995 .

Abdulhasan, M. Q. et al., 2014. *A channel quality indicator (CQI) prediction scheme using feed forward neural network (FF-NN) technique for MU-MIMO LTE system.* Langkawi, Malaysia, International Symposium on Telecommunication Technologies (ISTT), IEEE.

Akan, A. & Edemen, Ç., 2010. *Path to 4G wireless networks.* Instanbul, International Symposium on Personal, Indoor and Mobile Radio Communications Workshops (PIMRC Workshops), IEEE.

Albasir, A., Naik, K. & Abdunabi, T., 2013. *Smart mobile web browsing.* Aizu-Wakamatsu, Japan, International Joint Conference on Awareness Science and Technology and Ubi-Media Computing (iCAST-UMEDIA), IEEE.

Aldmour, R. et al., 2016. New cloud offloading algorithm for better energy consumption and process time. *Springer, International Journal of System Assurance Engineering and Management*, pp. 1-4.

References

Aliaga, M. & Gunderson , B., 2000. *Interactive Statistics.* Second ed. Saddle, NJ, USA: PrenticeHall.

Altamimi, M., Abdrabou, A., Naik , K. & Nayak, A., 2015. Energy Cost Models of Smartphones for Task Offloading to the Cloud. *Transactions on Emerging Topics in Computing, IEEE,* 4(3), pp. 384 - 398.

Altamimi, M. & Naik, K., 2014. *A Practical Task Offloading Decision Engine for Mobile Devices to Use Energy-as-a-Service (EaaS).* Anchorage, AK, World Congress on Services, IEEE.

Altamimi, M., Palit, R., Naik, K. & Nayak, A., 2012. *Energy-as-a-Service (EaaS): On the Efficacy of Multimedia Cloud Computing to Save Smartphone Energy.* Honolulu, HI, International Conference on Cloud Computing, IEEE.

Andrews, J. et al., 2014. What Will 5G Be?. *Journal on Selected Areas in Communications, IEEE,* 32(6), pp. 1065 - 1082.

Anon., n.d. s.l., s.n.

Antonopoulos, A., Kartsakli, E. & Verikoukis, C., 2014. Game theoretic D2D content dissemination in 4G cellular networks. *IEEE, Communications Society,* 52(6), pp. 125 - 132.

Atre, H., Razdan, K. & r Sagar, R. K., 2016. *A review of mobile cloud computing.* Noida, India, International Conference Cloud System and Big Data Engineering (Confluence)-IEEE.

Balan, R. et al., 2002. *The case for cyber foraging.* Saint-Emilion, France, ACM.

References

Balan, R., Gergle, D., Satyanarayanan, M. & Herbsleb, J., 2007. *Simplifying cyber foraging for mobile devices.* San Juan, Puerto Rico, Proceedings of the 5th international conference on Mobile systems, applications and services, ACM.

Balasubramanian, A., Mahajan, R. & Venkataramani, A., 2010. *Augmenting mobile 3G using WiFi.* New York, ACM international conference on Mobile systems, applications, and services.

Birley, G. & Moreland, N., 1998. *A Practical Guide to Academic Research Paperback.* First ed. London, UK: Routledge.

Bonomi, F., 2010. The future mobile infrastructure: challenges and opportunities [Industry Perspectives]. *IEEE Wireless Communications (Volume:17 , Issue: 5),* October, pp. 4 - 5.

Bruce, T., 2013. *Smartphone History: Evolution & Revolution.* Kindle Edition ed. s.l.:Bruce Taplin.

Bryman , A. & Bell, E., 2011. *Business Research Methods.* Third ed. Oxford, UK: Oxford University Press.

Burns, R., 2000. *Introduction to Research Methods.* First ed. London, UK: Sage.

Chun, B.-G.et al., 2011. *CloneCloud: Elastic Execution between Mobile Device and Cloud.* Salzburg, Austria, ACM, pp. 301-314.

DePoy, E. & Gitlin, L., 2016. *Introduction to Research: Understanding and Applying Multiple Strategies.* Fifth ed. Missouri, USA: ElSevier.

References

Dev , D. & Baishnab, K., 2014. *A Review and Research towards Mobile Cloud Computing*. Oxford, International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud), IEEE.

Dillistone, A., Moreno, R. & Voisey, C., 2010. *Market Perspective: Commercial Mobile Video*. NY, USA, MoViD '10 Proceedings of the 3rd workshop on Mobile video delivery, ACM.

Dillon, T., Wu , C. & Chang, E., 2010. *Cloud Computing: Issues and Challenges*. Perth, WA, International Conference on Advanced Information Networking and Applications, IEEE.

Dodig-Crnkovic, G., 2002. *Scientific methods in computer science*. Sweden, Skövde, Suecia, Conference for the Promotion of Research in IT.

Easterbrook, S., Singer, J., Storey, M. & Damian, D., 2008. *Selecting Empirical Methods for Software Engineering Research*. First ed. London, UK: Springer.

Eduardo , C. et al., 2010. *MAUI: Making Smartphones Last Longer with Code Offload*. San Francisco, California, USA, ACM, pp. 49-62.

Fakhfakh, E. & Hamouda, S., 2017. Optimised Q-learning for WiFi offloading in dense cellular networks. *Institution of Engineering and Technology, IEEE*, 11(15), pp. 2380 - 2385.

Fekete, K. et al., 2013. *Towards an Energy Efficient Code Generator for Mobile Phones*. Budapest, Hungary, International Conference on Cognitive Infocommunications (CogInfoCom), 2013 IEEE 4th.

Fesehaye, D., Gao, Y., Nahrstedt, K. & Wang, G., 2012. *Impact of Cloudlets on Interactive Mobile Cloud Applications.* Beijing, Enterprise Distributed Object Computing Conference (EDOC), IEEE.

Fitzek, F. & Widmer, J., 2011. *Survey on energy consumption entities on the smartphone platform.* Budapest, Hungary, Vehicular Technology Conference IEEE.

Flores , H. & Srirama, S., 2014. Mobile Cloud Middleware. *Journal of Systems and Software,* Volume 92, pp. 82-94.

Gao, B., He, L. & A. Jarvis, S., 2016. Offload Decision Models and the Price of Anarchy inin Mobile Cloud Application Ecosystems. *Emerging Cloud-Based Wireless Communications and Networks-IEEE Access,* 3(IEEE), pp. 3125 - 3137.

Gao, B. et al., 2012. *From Mobiles to Clouds: Developing Energy-Aware Offloading Strategies for Workflows.* Washington, DC, USA, International Conference on Grid Computing, ACM.

Geng, Y. et al., 2015. *Energy-Efficient Computation Offloading in Cellular Networks.* San Francisco, CA, USA, International Conference on Network Protocols (ICNP), IEEE.

Gong , C. et al., 2010. *The Characteristics of Cloud Computing.* San Diego, CA, International Conference on Parallel Processing Workshops, IEEE.

Grigorik, I., 2013. *High Performance Browser Networking: What every web developer should know about networking and web performance.* First Edition ed. US: O'Reilly Media.

References

Grønli, T.-M., Hansen, J. & Ghinea, G., 2010. *Android vs Windows Mobile vs Java ME A comparative Study of Mobile Development Environments.* Samos, Greece, ACM-Proceedings of the 3rd International Conference on PErvasive Technologies Related to Assistive Environments.

Hashem, I. et al., 2014. The riseof “big data” on cloudcomputing:Reviewand open research issues. *Elsevier,* Volume 47, pp. 98-115.

Huber , F., Srirama, S. N. & Paniagua, C., 2012. Towards mobile cloud applications Offloading resource-intensive tasks to hybrid clouds. *International Journal of Pervasive- Emerald Group Publishing Limited,* 8(4), pp. 344-367.

Hung, S.-H.et al., 2012. Executing mobile applications on the cloud: Framework and issues. *Computers and Mathematics with Applications-Elsevier,* 63(2), p. 573–587.

Itani, W., Chehab, A. & Kayssi, A., 2011. *Energy-efficient platform-as-a-service security provisioning in the cloud.* s.l., Energy Aware Computing (ICEAC) IEEE.

Jararweh, Y. et al., 2014. *Scalable cloudlet-based mobile computing model. Procedia Computer Science.* s.l., Elseiver, pp. Volume 34, 2014, Pages 434-441.

Justino , T. & Buyya, R., 2014. *Outsourcing Resource-Intensive Tasks from Mobile Apps to Clouds: Android and Aneka Integration.* Bangalore, International Conference on Cloud Computing in Emerging Markets (CCEM), IEEE.

Kelenyi , I. & Nurminen , J., 2010. *CloudTorrent - Energy-Efficient BitTorrent Content Sharing for Mobile Devices via Cloud Services.* s.l., Consumer Communications and Networking Conference (CCNC), 2010 7th IEEE.

References

Kemp, R., Palmer, N., Kielmann , T. & Bal, H., 2012. Cuckoo: A Computation Offloading Framework for Smartphones. In: M. Gris & G. Yang, eds. *Mobile Computing, Applications, and Services.* Berlin Heidelberg: Springer, pp. 59-79.

Khan , A., Qadeer , M. A., Ansari , J. A. & Waheed, S., 2009. *4G as a Next Generation Wireless Network.* Kuala Lumpar, International Conference on Future Computer and Communication, IEEE.

Kosta, S., Aucinas, A. & Hui, P., 2012. *ThinkAir: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading.* Germany , IEEE.

Kothari, M., Khare, Y. & Dagar, A., 2017. Comprehensive Overview of Emerging Cellular Technology. *ProQuest, International Journal of Computer Applications.,* 162(11).

Kovachev , D. & Klamma, R., 2012. Framework for computation offloading in mobile cloud computing. *IJIMAI,* 7(10.9781), p. Pagination.

Kumar , K. & Lu, Y.-H., 2010. Cloud Computing for Mobile Users: Can Offloading Computation Save Energy?. *Computer Society, IEEE,* April, pp. 51 - 56.

Kumar, K., Liu, J., Lu, Y.-H. & Bhargava, B., 2013. A Survey of Computation Offloading for Mobile Systems. In: *Mobile Networks and Applications.* US: Springer US, p. 129–140.

Kumar, R., 2008. *Research Methodology.* First ed. New Delhi, India: APH Publishing Corporation.

Lane , N. D. et al., 2010. A survey of mobile phone sensing. *IEEE Communications Magazine (Volume:48 , Issue: 9),* 02 September, pp. 140-150.

References

Lauridsen, M., Mogensen, P. & Noel, L., 2013. *Empirical LTE Smartphone Power Model with DRX Operation for System Level Simulations.* Las Vegas, NV, USA, 78th Vehicular Technology Conference (VTC Fall), IEEE.

Lee, J., Yi, Y., Chong, S. & Jin, Y., 2014. Economics of WiFi Offloading: Trading Delay for Cellular Capacity. *Transactions on Wireless Communications, IEEE,* 13(3), pp. 1540 - 1554.

Li, W., Zhao, Y., Lu, S. & Chen, D., 2015. Mechanisms and challenges on mobility-augmented service provisioning for mobile cloud computing. *IEEE Communications Magazine,* 53(3), pp. 89 - 97.

Magurawalage, C., Zhang, J., Hu, L. & Yang, K., 2014. Energy-efficient and network-aware offloading algorithm for mobile cloud computing. *ELSEVIER,* Volume 74, p. 22–33.

Marlow, C., 2011. *Research Methods for Generalist Social Work.* Fifth ed. California, USA: Cengage Learning.

Masiperiyannan, S. et al., 2014. Security in Offloading Computations in Mobile Systems Using Cloud Computing. *IJAREEIE,* 4(3), pp. 7729-7734.

Ma, X. et al., 2013. When mobile terminals meet the cloud: computation offloading as the bridge. *IEEE,* 27(5), pp. 28 - 33.

Mell, P. & Grance, T., 2011. *The NIST Definition of Cloud Computing,* Gaithersburg,: National Institute of Standards and Technology.

References

Moghadasi, N., Arani, M. & Shamsi, M., 2015. A Novel Approach for Reduce Energy Consumption in Mobile Cloud Computing. *International Journal of Computer Network and Information Security- ProQuest,* Volume 10, pp. 58-69.

Monsen, E. & Horn, L., 2008. *Research Successful Approaches: Edition. USA:.* Third ed. USA: American Dietetic Association..

Naik, K., 2010. A survey of software based energy saving methodologies for handheld wireless communication devices. pp. 1-46.

Nossenson, R., 2009. *Long-term evolution network architecture.* s.l., IEEE.

Pathak , R., 2008. *Methodology of Educational Research.* New Delhi, India: Atlantic.

Patil, M., Sahu, V. & Jain, A., 2014. *SMS text Compression and Encryption on Android O.S.* Coimbatore, INDIA , IEEE-2014 International Conference on Computer Communication and Informatics.

Punch, K. . F., 2005. *Introduction to Social Research Quantitative and Qualitative Approaches.* Second ed. London, UK: Sage.

Qian , H. & Andresen, D., 2015. *Reducing Mobile Device Energy Consumption with computation offloading.* Takamatsu, International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), IEEE/ACIS.

Qian, F. et al., 2011. *Profiling resource usage for mobile applications: A cross-layer approach.* Bethesda, Maryland, USA, international conference on Mobile systems, applications, and services, ACM.

References

Qian, F. et al., 2011. *Profiling resource usage for mobile applications: a cross-layer approach.* Bethesda, Maryland, USA, 9th international conference on Mobile systems, applications, and services, ACM.

Qi, Q. et al., 2016. *Dynamic resource orchestration for multi-task application in heterogeneous mobile cloud computing.* San Francisco, CA, USA, Conference on Computer Communications Workshops (INFOCOM WKSHPS- IEEE.

Qi, Q. et al., 2016. *Dynamic resource orchestration for multi-task application in heterogeneous mobile cloud computing.* San Francisco, CA, USA, IEEE-Computer Communications Workshops (INFOCOM WKSHPS).

Rawashdeh, M., Alnusair, A., Mustafa, N. & Migdadi, M., 2016. *Multimedia Mobile Cloud Computing: Application models for performance enhancement.* Seattle, WA, USA, IEEE, International Conference onMultimedia & Expo Workshops (ICMEW).

Ristanovic, N., Le Boudec, J.-Y., Chaintreau, A. & Erramilli, V., 2011 . *Energy Efficient Offloading of 3G Networks.* Valencia, Spain, 8th International Conference on Published in: Mobile Adhoc and Sensor Systems (MASS), IEEE .

Rodriguez, N., Hui, P., Crowcroft, J. & Rice, A., 2010. Exhausting Battery Statistics Understanding the energy demands on mobile handsets. *ACM,* Issue 10.1145/1851322.1851327, pp. 9-14.

Sathiaseelan, A. & Crowcroft, J., 2013. Internet on the Move: Challenges and Solutions. *ACM*, 43(1), pp. 51-55.

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

References

Saunders, M., Thornhill, A. & Lewis, P., 2009. *Research Methods for Business Students.* Fifth ed. Harlow: Pearson Education.

Shafi, M. et al., 2017. 5G: A Tutorial Overview of Standards, Trials, Challenges, Deployment, and Practice. *Journal on Selected Areas in Communications, IEEE,* 35(6), pp. 1201 - 1221.

Sharifi, M., Kafaie, S. & Kashefi, O., 2012. A Survey and Taxonomy of Cyber Foraging of Mobile Devices. *IEEE Communications Surveys & Tutorials,* 14(4), pp. 1232 - 1243.

Shiraz, M. et al., 2015. Energy Efficient Computational Offloading Framework for Mobile Cloud Computing. *Journal of Grid Computing, Springer,* 13(1), p. 1–18.

Shiraz, M., Sookhak , M., Gani , A. & Shah, S., 2015. A Study on the Critical Analysis of Computational Offloading Frameworks for Mobile Cloud Computing. *Journal of Network and Computer Applications, ELSEIVER,* Volume 47, p. 47–60.

Shiraz, M., Gani, A., Khokhar, R. H. & Buyya, R., 2013. A Review on Distributed Application Processing Frameworks in Smart Mobile Devices for Mobile Cloud Computing. *IEEE, Communications Surveys & Tutorials* , 15(3), pp. 1294 - 1313.

Shu , P. et al., 2013. *eTime: Energy-efficient transmission between cloud and mobile devices.* China, INFOCOM. IEEE.

Shu, P. et al., 2013. *eTime: Energy-efficient transmission between cloud and mobile devices.* China, NFOCOM, 2013 Proceedings IEEE.

Simsek, M. et al., 2016. 5G-Enabled Tactile Internet. *IEEE Journal on Selected Areas in Communications, IEEE,* 34(3), pp. 460 - 473.

References

Vriendt, J. D., Lainé, P., Lerou, C. & Xiaofeng , A., 2002. Mobile Network Evolution: A Revolution on the Move. *IEEE Communications Magazine*, April, pp. 104-111.

Wang , J. et al., 2016. *Adaptive Application Offloading Decision and Transmission Scheduling for Mobile Cloud Computing.* Kuala Lumpur, Malaysia, International Conference on Communications (ICC), 2016 IEEE.

Wang, J. et al., 2017. Adaptive application offloading decision and transmission scheduling for mobile cloud computing. *China Communications-IEEE*, 14(3), pp. 169 - 181.

Wolski, R., Gurun, S., Krintz, C. & Nurmi, D., 2008. *Using bandwidth data to make computation offloading decisions.* Miami, FL, International Symposium on Parallel and Distributed Processing, IEEE.

Xiao, Y., Kalyanaraman, R. & Yla-Jaaski, A., 2008. *Energy Consumption of Mobile YouTube: Quantitative Measurement and Analysis.* Cardiff, UK, The Second International Conference on Next Generation Mobile Applications, Services and Technologies, IEEE.

Yang, L., Cao, J., Cheng, H. & Ji, Y., 2015. Multi-User Computation Partitioning for Latency Sensitive Mobile Cloud Applications. *Transactions on Computers-IEEE,* 64(8), pp. 2253 - 2266.

Yang, L. et al., 2016. Run Time Application Repartitioning in Dynamic Mobile Cloud Environments. *Transactions on Cloud Computing-IEEE,* 4(3), pp. 336 - 348.

Yanikomeroglu, H., 2012. *Towards 5G wireless cellular networks: Views on emerging concepts and technologies.* Carleton University, Ottawa, Canada, IEEE.

Zhang, S., Zhang, S., Chen , X. & Wu, S., 2010. *Analysis and Research of Cloud Computing System Instance.* Sanya, Hainan, Second International Conference on Future Networks, IEEE.

Zhang, L. et al., 2010. *Accurate online power estimation and automatic battery behavior based power model generation for smartphones.* Scottsdale, AZ, USA, (CODES+ISSS), 2010 IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis.

Zhou, B. et al., 2015. *A Context Sensitive Offloading Scheme for Mobile.* New York City, NY, International Conference on Cloud Computing, IEEE.

I.D.Corporation. Smartphone os market share, q2 2016. http://www.idc.com/prodserv/ smartphone-os-market-share.jsp, 2016-08

http://www.nielsen.com/content/dam/corporate/us/en/newswire/uploads/2009/04/itradio-apr09.pdf

Appendix

Appendix I. Results for Schedule Time

1. Power Consumption

The tables and figures below illustrate the results for power consumption when scheduled time during the tests:

Table 6.2: Power consumption for file size 1Mb in both rounds using Wi-Fi and 4G

Reading	First Round (Wi-Fi)	Second Round (Wi-Fi)	First Round (4G)	Second Round (4G)
1	0.534	0.0106	0.186	0.0494
2	0.568	0.0211	0.0489	0.0395
3	0.131	0.0341	0.0949	0.0516
4	0.0841	0.0361	0.106	0.0389
5	0.145	0.0762	0.0885	0.0479
Average	**0.29242**	**0.03562**	**0.10486**	**0.04546**

Table 6.2: Power consumption for file size 3Mb in both rounds using Wi-Fi and 4G

Reading	First Round (Wi-Fi)	Second Round (Wi-Fi)	First Round (4G)	Second Round (4G)
1	0.831	0.042	0.696	0.0471
2	0.234	0.0407	0.0497	0.0417
3	0.0968	0.0406	0.164	0.0523
4	0.123	0.0367	0.109	0.0398
5	0.318	0.0164	0.144	0.0223
Average	**0.32056**	**0.03528**	**0.23254**	**0.04064**

Table 6.2: Power consumption for file size 5Mb in both rounds using Wi-Fi and 4G

Reading	First Round (Wi-Fi)	Second Round (Wi-Fi)	First Round (4G)	Second Round (4G)
1	0.114	0.058	0.155	0.071
2	0.166	0.0505	0.203	0.0628
3	0.0853	0.0423	0.172	0.0875
4	0.173	0.0366	0.202	0.0998
5	0.281	0.0409	0.57	0.0573
Average	**0.16386**	**0.04566**	**0.2604**	**0.07568**

Table 6.2: Power consumption for file size 10Mb in both rounds using Wi-Fi and 4G

Reading	First Round (Wi-Fi)	Second Round (Wi-Fi)	First Round (4G)	Second Round (4G)
1	0.335	0.0553	0.175	0.0292
2	0.335	0.0581	0.245	0.0597
3	0.236	0.0668	0.356	0.0526
4	0.271	0.0599	0.424	0.0662
5	0.129	0.0601	0.256	0.046
Average	**0.2612**	**0.06004**	**0.2912**	**0.05074**

Table 6.2: Power consumption for file size 15Mb in both rounds using Wi-Fi and 4G

Reading	First Round (Wi-Fi)	Second Round (Wi-Fi)	First Round (4G)	Second Round (4G)
1	0.454	0.0231	0.36	0.0317
2	0.346	0.0572	0.43	0.0558
3	0.32	0.0793	0.405	0.0104
4	0.577	0.0675	0.421	0.0669
5	0.311	0.063	0.728	0.0655
Average	**0.4016**	**0.05802**	**0.4688**	**0.04606**

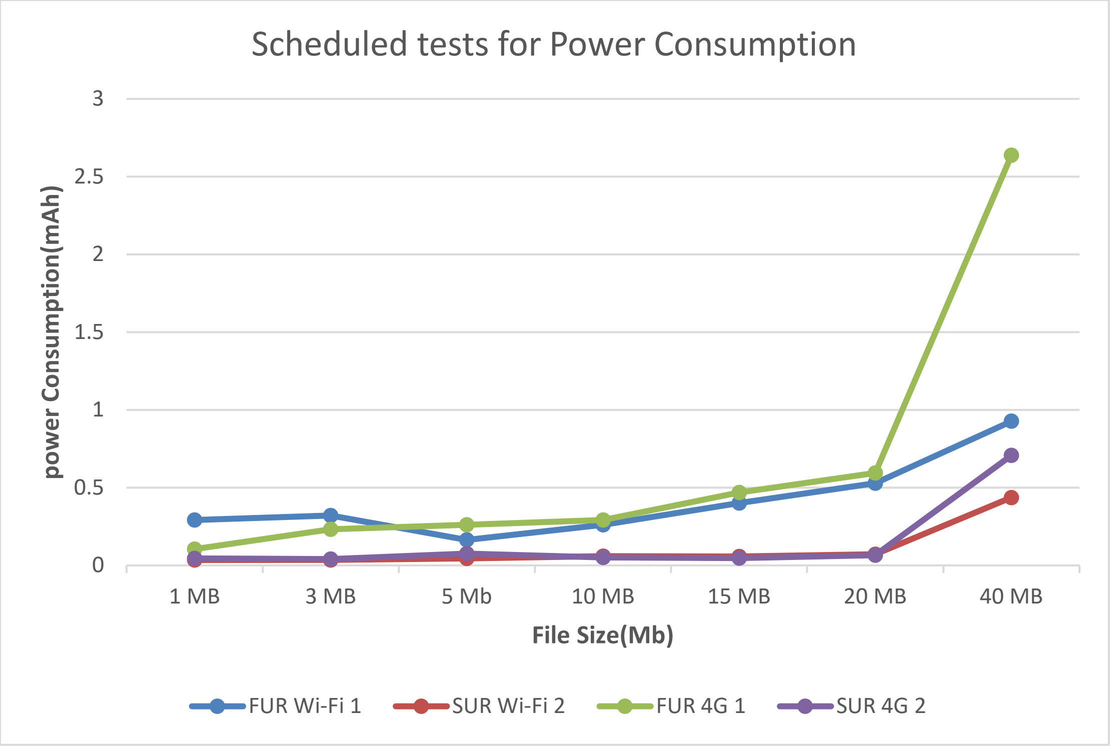
Table 6.2: Power consumption for file size 20Mb in both rounds using Wi-Fi and 4G

Reading	First Round (Wi-Fi)	Second Round (Wi-Fi)	First Round (4G)	Second Round (4G)
1	0.367	0.073	0.524	0.068
2	0.362	0.0619	0.614	0.0875
3	0.553	0.0784	0.446	0.0652
4	0.593	0.0661	0.669	0.0328
5	0.773	0.081	0.721	0.073
Average	**0.5296**	**0.07208**	**0.5948**	**0.0653**

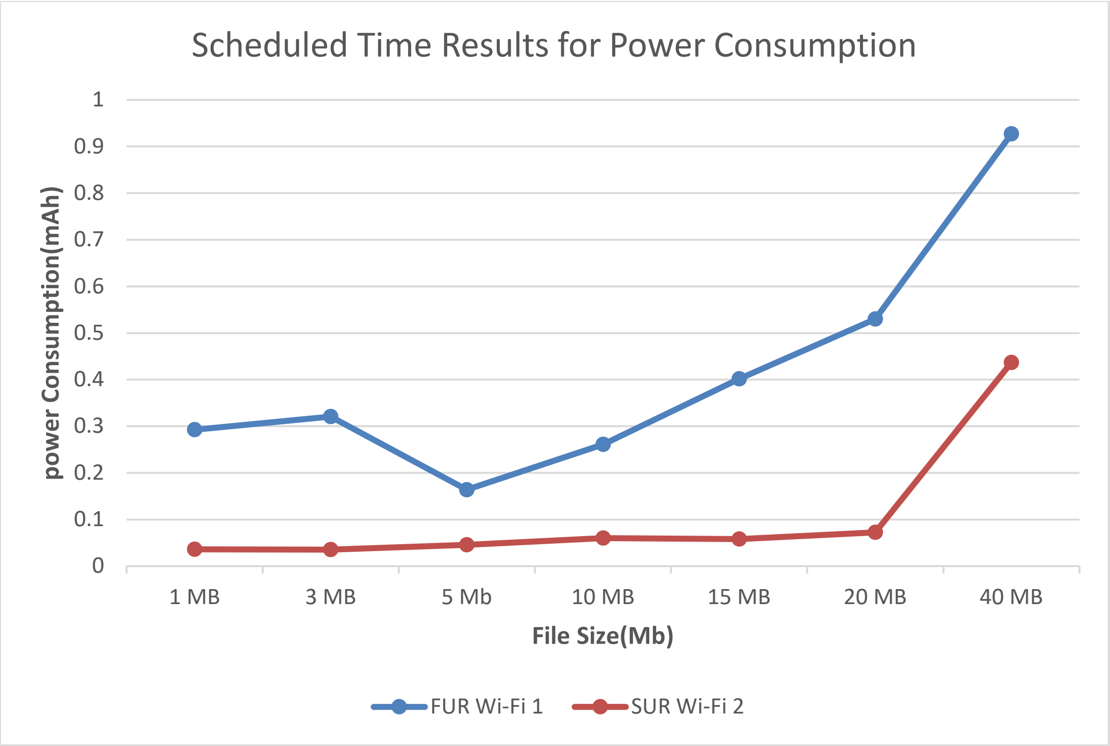
Table 6.2: Power consumption for file size 40Mb in both rounds using Wi-Fi and 4G

Reading	First Round (Wi-Fi)	Second Round (Wi-Fi)	First Round (4G)	Second Round (4G)
1	0.875	0.0905	2.09	0.0826
2	0.821	0.0841	3.5	0.943
3	0.964	0.763	1.8	0.724
4	1.04	0.649	2.7	0.859
5	0.934	0.597	3.1	0.927
Average	**0.9268**	**0.43672**	**2.638**	**0.70712**

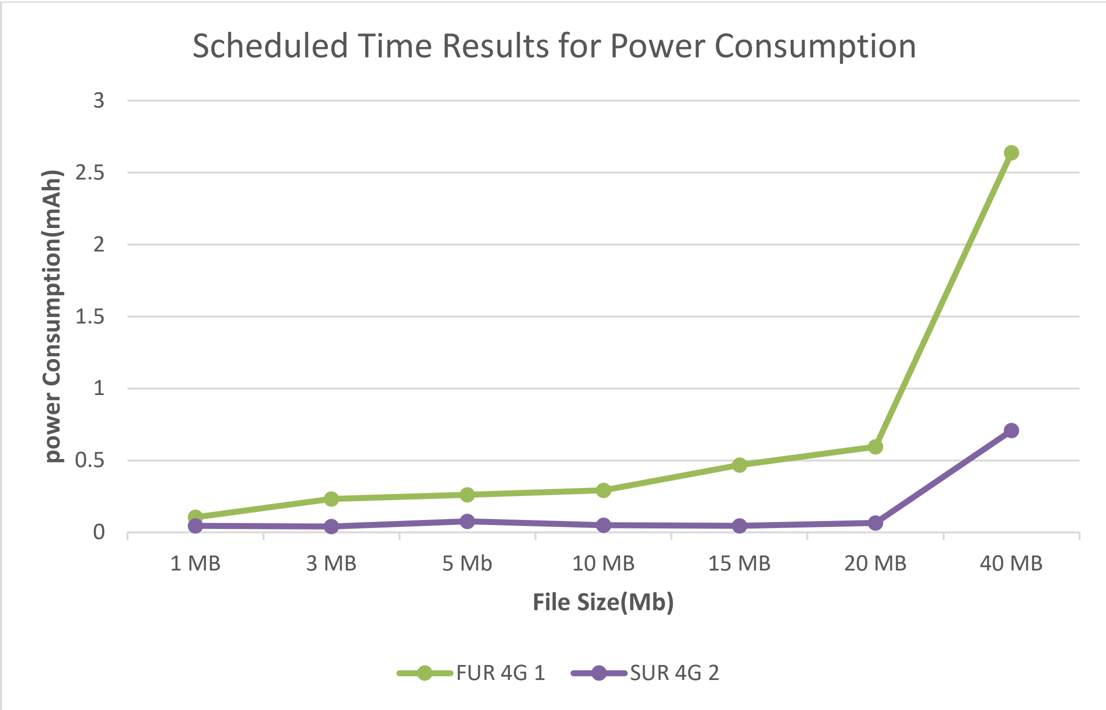
Appendix



Power Consumption for Scheduled time tests whilst using Wi-Fi and 4G in FUR and SUR



Power Consumption for Scheduled time whilst using Wi-Fi in FUR and SUR



Power Consumption for Scheduled time whilst using 4G in FUR and SUR

2. Processing Time:

Processing time for file size 1Mb in both rounds using Wi-Fi and 4G

Reading	First Round (Wi-Fi)	Second Round (Wi-Fi)	First Round (4G)	Second Round (4G)
1	1419	256	4219	631
2	4034	280	1748	624
3	1815	268	1664	687
4	2176	273	1509	580
5	1983	263	2297	667
Average	**2285.4**	**268**	**2287.4**	**637.8**

Processing time for file size 3Mb in both rounds using Wi-Fi and 4G

Reading	First Round (Wi-Fi)	Second Round (Wi-Fi)	First Round (4G)	Second Round (4G)
1	**3479**	**661**	**16181**	**834**
2	**7964**	**437**	**4081**	**902**
3	**3122**	**418**	**3147**	**754**
4	**4850**	**430**	**4150**	**600**
5	**4342**	**520**	**4769**	**702**
Average	**4751.4**	**493.2**	**6465.6**	**758.4**

Appendix

Processing time for file size 5Mb in both rounds using Wi-Fi and 4G

Reading	First Round (Wi-Fi)	Second Round (Wi-Fi)	First Round (4G)	Second Round (4G)
1	6491	702	4781	941
2	4038	658	5449	808
3	4222	690	4988	956
4	10458	641	8064	934
5	7473	682	6490	823
Average	**6536.4**	**674.6**	**5954.4**	**892.4**

Processing time for file size 10Mb in both rounds using Wi-Fi and 4G

Reading	First Round (Wi-Fi)	Second Round (Wi-Fi)	First Round (4G)	Second Round (4G)
1	13608	1040	7977	1288
2	10822	1236	11689	1409
3	8009	1691	11405	1327
4	6488	1031	9961	1419
5	5278	1168	10840	1421
Average	**8841**	**1233.2**	**10374.4**	**1372.8**

Processing time for file size 15Mb in both rounds using Wi-Fi and 4G

Reading	First Round (Wi-Fi)	Second Round (Wi-Fi)	First Round (4G)	Second Round (4G)
1	16019	1777	15406	1813
2	10822	1598	14622	1754
3	10493	1522	14062	1608
4	20902	2200	16345	1679
5	9390	1497	18142	1745
Average	**14239**	**1718.8**	**15715.4**	**1719.8**

Processing time for file size 20Mb in both rounds using Wi-Fi and 4G

Reading	First Round (Wi-Fi)	Second Round (Wi-Fi)	First Round (4G)	Second Round (4G)
1	10405	1856	23081	2114
2	9801	2017	21115	2055
3	13255	1852	15495	1824
4	15025	1809	19982	2199
5	19288	1991	21569	2153
Average	**13554.8**	**1905**	**20248.4**	**2069**

Processing time for file size 40Mb in both rounds using Wi-Fi and 4G

Reading	First Round (Wi-Fi)	Second Round (Wi-Fi)	First Round (4G)	Second Round (4G)
1	15290	3573	39272	2557
2	13705	3248	50986	2841
3	16547	3107	33589	2357
4	19640	2994	403954	2483
5	14735	2762	426710	2951
Average	**12814.6**	**3136.8**	**190902.2**	**2637.8**

Appendix II. Different Mobile Device Results

1. Power Consumption

The tables below shows the results for the power consumption when using different mobile device during processing files on the cloud.

Power consumption for file size 1Mb in both rounds using Wi-Fi and 4G

Reading	First Round (Wi-Fi)	Second Round (Wi-Fi)	First Round (4G)	Second Round (4G)
1	0.195	0.039	0.174	0.0553
2	0.0778	0.0493	0.14	0.0627
3	0.227	0.0776	0.165	0.0296
4	0.133	0.0258	0.147	0.0317
5	0.0992	0.0467	0.135	0.0284
Average	**0.1464**	**0.04768**	**0.1522**	**0.04154**

Power consumption for file size 3Mb in both rounds using Wi-Fi and 4G

Reading	First Round (Wi-Fi)	Second Round (Wi-Fi)	First Round (4G)	Second Round (4G)
1	0.154	0.0493	0.276	0.0737
2	0.233	0.0692	0.246	0.0754
3	0.189	0.077	0.231	0.0477
4	0.155	0.0406	0.302	0.0503
5	0.109	0.0381	0.295	0.0371
Average	**0.168**	**0.05484**	**0.27**	**0.05684**

2. Processing Time

Processing time for file size 1Mb in both rounds using Wi-Fi and 4G

Reading	First Round (Wi-Fi)	Second Round (Wi-Fi)	First Round (4G)	Second Round (4G)
1	2732	281	3346	557
2	1902	276	5261	670
3	2520	385	2887	705
4	2365	429	3589	666
5	3853	358	3642	604
Average	**2674.4**	**345.8**	**3745**	**640.4**

Processing time for file size 3Mb in both rounds using Wi-Fi and 4G

Reading	First Round (Wi-Fi)	Second Round (Wi-Fi)	First Round (4G)	Second Round (4G)
1	3526	556	7136	747
2	5762	562	6835	775
3	4010	431	6931	705
4	2940	623	6926	875
5	4886	817	5654	864
Average	**4224.8**	**597.8**	**6696.4**	**793.2**

Processing time for file size 5Mb in both rounds using Wi-Fi and 4G

Reading	First Round (Wi-Fi)	Second Round (Wi-Fi)	First Round (4G)	Second Round (4G)
1	7127	665	13188	965
2	6289	693	10952	909
3	4891	649	10704	896
4	5376	834	7999	834
5	4324	673	7222	771
Average	**5601.4**	**702.8**	**10013**	**875**

Processing time for file size 10Mb in both rounds using Wi-Fi and 4G

Reading	First Round (Wi-Fi)	Second Round (Wi-Fi)	First Round (4G)	Second Round (4G)
1	6558	1114	11405	1336
2	5415	1271	17777	1450
3	6427	1317	21295	1304
4	7112	1289	20945	1327
5	5111	1523	31444	1260
Average	**6124.6**	**1302.8**	**20573.2**	**1335.4**

Processing time for file size 15Mb in both rounds using Wi-Fi and 4G

Appendix

Reading	First Round (Wi-Fi)	Second Round (Wi-Fi)	First Round (4G)	Second Round (4G)
1	10130	3057	21104	1586
2	8964	1497	21770	1727
3	9020	1561	18843	1818
4	8627	1361	19038	1707
5	11763	1615	16891	1652
Average	**9700.8**	**1818.2**	**19529.2**	**1698**

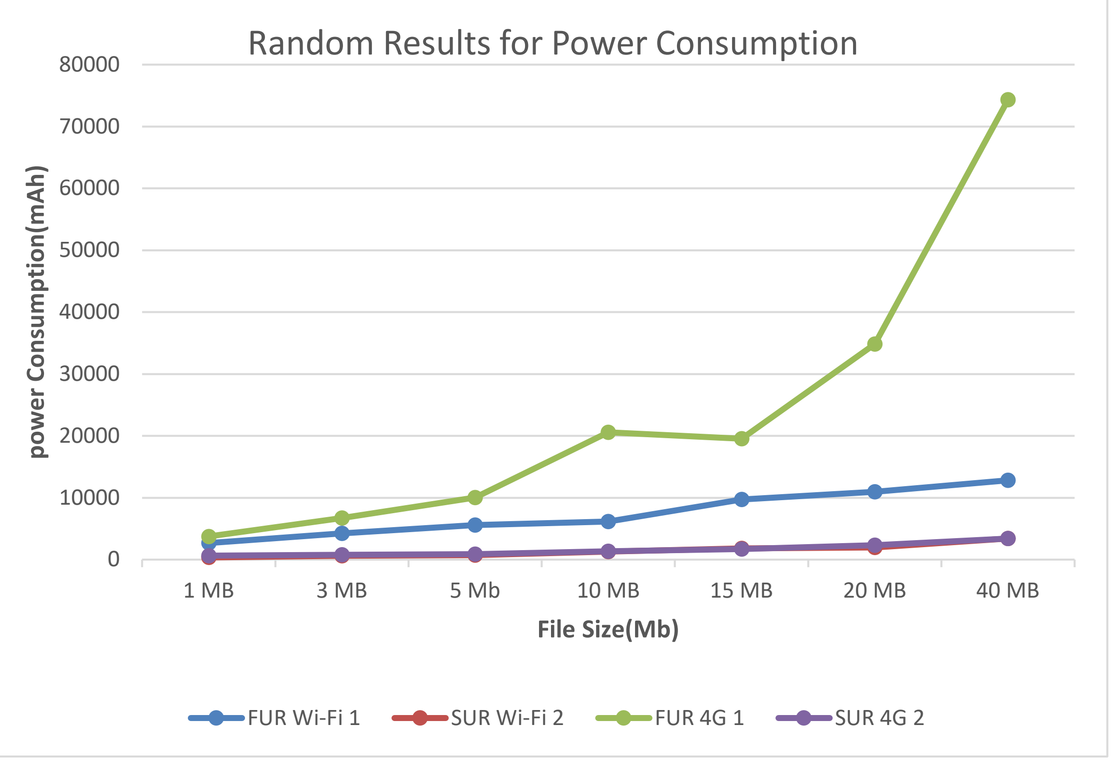
Processing time for file size 20Mb in both rounds using Wi-Fi and 4G

Reading	First Round (Wi-Fi)	Second Round (Wi-Fi)	First Round (4G)	Second Round (4G)
1	12263	2219	37198	2348
2	11161	2045	30639	2608
3	10004	2001	26864	1921
4	13251	1709	41165	2124
5	8122	1925	38439	2574
Average	**10960.2**	**1979.8**	**34861**	**2315**

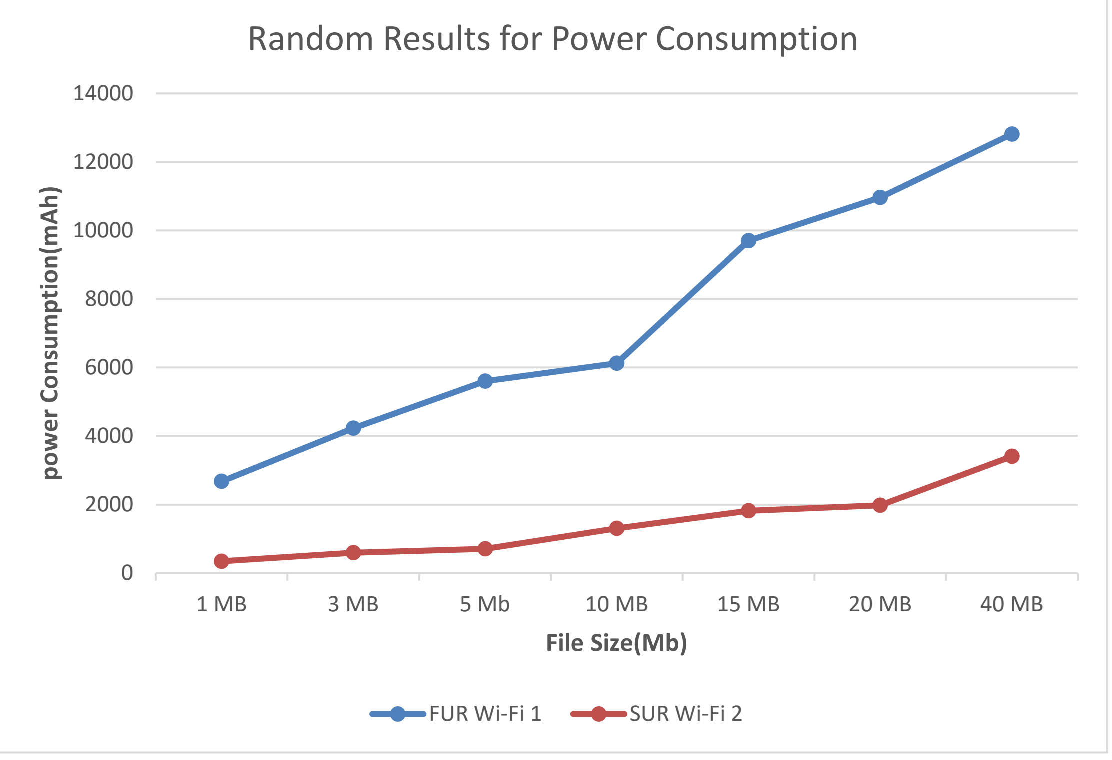
Processing time for file size 40Mb in both rounds using Wi-Fi and 4G

Reading	First Round (Wi-Fi)	Second Round (Wi-Fi)	First Round (4G)	Second Round (4G)
1	12361	3426	85466	3643
2	11735	3535	72016	3820
3	11921	3217	74315	3512
4	14359	3751	70158	3019
5	13697	3109	69583	2997
Average	**12814.6**	**3407.6**	**74307.6**	**3398.2**

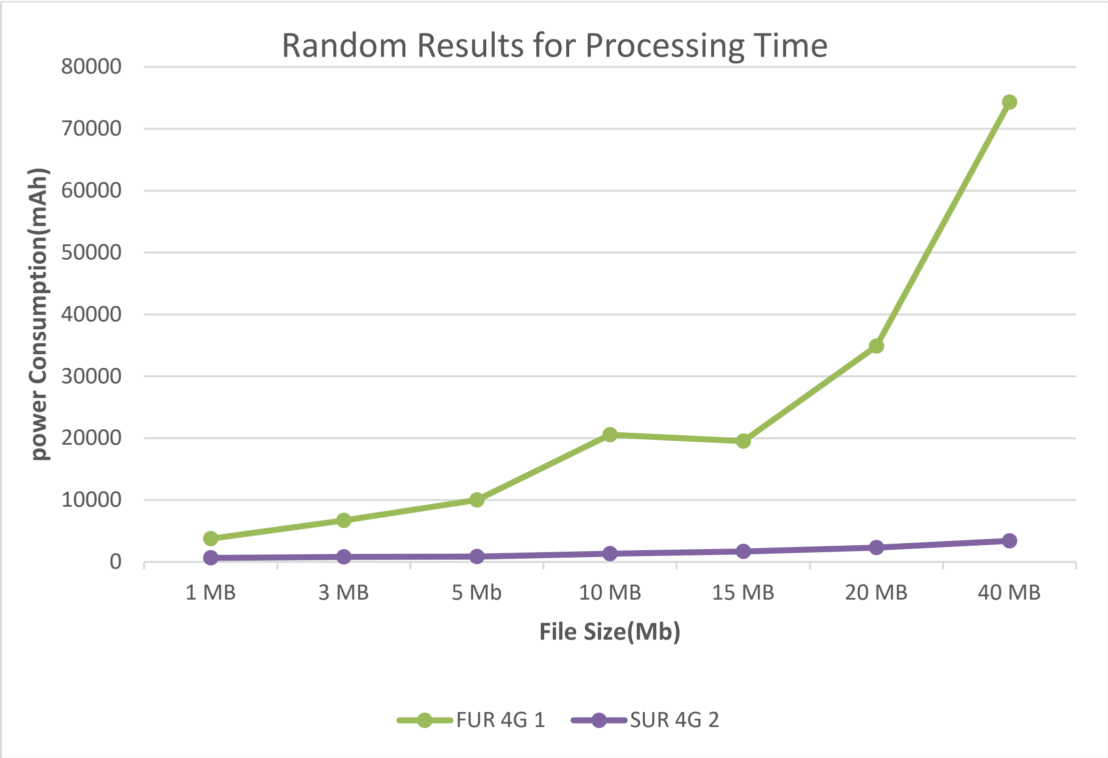
Appendix



Processing time for file sizes in both rounds using Wi-Fi and 4G



Processing time for file sizes in both rounds using Wi-Fi



Processing time for file sizes in both rounds using 4G