MASTER’S THESIS

DESIGN, IMPLEMENTATION AND TESTING
OF A MOBILE CLOUD

Author       Abdul Moiz
Supervisor   Prof. Marcos Katz
Second Examiner Prof. Jari Iinatti

November, 2015
ABSTRACT

Telecommunication industry has experienced a major breakthrough in the last few decades due to the immense development in information technology. Ubiquitous connectivity, expeditious increase in the number of low cost yet powerful smart devices and quantum leap in social networking are posing new challenges to cope up with the current as well as future requirements. While substantial amount of work has been done in this context, particularly on cooperative and cognitive networks, the very approach has certain limitations and shortcomings. The three characteristic challenges are enhance system throughput, dynamic environment adaptability and productive utilization of the available resources.

Here we present mobile cloud, a novel yet simplistic system model that employs cognitive and cooperative strategies to address all of these three challenges. The system exploits the short range link to establish a small social network among the nearby devices, adapts according to environment and uses various cooperation strategies to obtain efficient utilization of resources.

Lastly, we implemented an experimental mobile cloud and attentively assessed its performance with varying parameters and legacy approach used in the similar context. The analysis provided sound understanding of system model compliance with the primary objectives as well as with the future networks.

Keywords: D2D, Cognitive, Cooperative, Multi-radio communication, Content sharing, Social Networking, Android, Smartphones.
# TABLE OF CONTENTS

**ABSTRACT**

**TABLE OF CONTENTS**

**FOREWORD**

**LIST OF ABBREVIATIONS AND SYMBOLS**

1. **INTRODUCTION**
   1.1. Motivation ............................................. 8
   1.2. Related work and contribution ............................. 10
   1.3. Structure of the thesis .................................. 11

2. **COOPERATIVE AND COGNITIVE RADIO COMMUNICATION** 12
   2.1. Overview ............................................... 12
   2.2. Mobile Clouds .......................................... 13

3. **TEST-BED DESIGN AND IMPLEMENTATION FOR MOBILE CLOUD** 16
   3.1. Overview ............................................... 16
   3.2. Software platform ...................................... 17
   3.3. Hardware platform ...................................... 18
       3.3.1. Samsung Galaxy S4 .................................. 19
       3.3.2. Linksys wireless router EA6300 ..................... 19
       3.3.3. Mobile Cloud content server ......................... 20
   3.4. Multi-radio communication ................................ 20
   3.5. Mobile Cloud formation phases ......................... 22
       3.5.1. Assumptions ........................................ 23
       3.5.2. Cloud formation ..................................... 23
       3.5.3. Content distribution ................................. 24
       3.5.4. In-cloud content distribution ....................... 25
       3.5.5. Working algorithm .................................. 25
       3.5.6. Out-cloud content distribution ..................... 26
       3.5.7. Cloud management ................................... 31

4. **MEASUREMENTS AND RESULTS** 32
   4.1. Overview ............................................... 32
   4.2. General assumptions and environment variables .......... 33
   4.3. Test cases ............................................. 34
       4.3.1. Single node download time .......................... 34
       4.3.2. Mobile Cloud download time ......................... 36
       4.3.3. Effect of RSSI on Mobile Cloud .................... 42
   4.4. Discussion .............................................. 44
       4.4.1. Sophisticated cooperation strategies ............... 44
       4.4.2. Real-time content distribution ..................... 45
       4.4.3. Overall system performance ......................... 45
5. SUMMARY 48
6. REFERENCES 50
7. APPENDICES 54
FOREWORD

This thesis is written in completion to the Master’s in Wireless Communication Engineering, at University of Oulu, Finland. The completed work is a part of “Cognitive and Intelligent Solutions for Testing and Monitoring of Future Access Technologies” project. This project was funded by Tekes, VTT, EXFO, Bittium (formerly known as Elektrobit), NSN, RuggedTooling and University of Oulu. The main part of this thesis was completed at Centre for Wireless Communications, a research unit at University of Oulu. The purpose of the work was to develop cognitive testing methods for heterogeneous networks.

First of all, I would like to thank my supervisor, Professor Marcos Katz for his continuous support, incomparable supervision and mentoring. I am also grateful to Kanwar Saad bin Liaqat and Muhammad Ikram Ashraf for providing technical assistance, exceptional guidance and friendly working environment. I would also like to thank Professor Jari Iinatti for reviewing my thesis and expert opinion. Last but not least, my heartiest gratitude goes to all my family members for their love, affection, trust, patient and support throughout this study period. I dedicate this humble work to my honourable parents Abdul Hafeez and Taj-un-Nisa.

Oulu, Finland November 28, 2015

Abdul Moiz
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>3GPP</td>
<td>3rd Generation Partnership Project</td>
</tr>
<tr>
<td>AC</td>
<td>Alternating Current</td>
</tr>
<tr>
<td>ADB</td>
<td>Android Debug Bridge</td>
</tr>
<tr>
<td>AP</td>
<td>Access Point</td>
</tr>
<tr>
<td>API</td>
<td>Application Programmable Interface</td>
</tr>
<tr>
<td>BS</td>
<td>Base Station</td>
</tr>
<tr>
<td>CL</td>
<td>Cloud Leader</td>
</tr>
<tr>
<td>CN</td>
<td>Cloud Node</td>
</tr>
<tr>
<td>COIN</td>
<td>Cognitive and Intelligent Solutions</td>
</tr>
<tr>
<td>CPU</td>
<td>Central Processing Unit</td>
</tr>
<tr>
<td>CSMA/CA</td>
<td>Carrier Sense Multiple Access with Collision Avoidance</td>
</tr>
<tr>
<td>D2D</td>
<td>Device to Device</td>
</tr>
<tr>
<td>DIFS</td>
<td>Distributed Inter-Frame Space</td>
</tr>
<tr>
<td>DNS</td>
<td>Domain Name Server</td>
</tr>
<tr>
<td>GO</td>
<td>Group Owner</td>
</tr>
<tr>
<td>GPS</td>
<td>Global Positioning System</td>
</tr>
<tr>
<td>GSM</td>
<td>Global System for Mobile Communication</td>
</tr>
<tr>
<td>HD</td>
<td>High Definition</td>
</tr>
<tr>
<td>HSDPA</td>
<td>High Speed Downlink Packet Access</td>
</tr>
<tr>
<td>HSUPA</td>
<td>High Speed Uplink Packet Access</td>
</tr>
<tr>
<td>ICMP</td>
<td>Internet Control Message Protocol</td>
</tr>
<tr>
<td>IEEE</td>
<td>Institute of Electrical and Electronics Engineers</td>
</tr>
<tr>
<td>IoT</td>
<td>Internet of Things</td>
</tr>
<tr>
<td>IP</td>
<td>Internet Protocol</td>
</tr>
<tr>
<td>LTE</td>
<td>Long Term Evolution</td>
</tr>
<tr>
<td>LTS</td>
<td>Long Term Support</td>
</tr>
<tr>
<td>MAC</td>
<td>Medium Access Control</td>
</tr>
<tr>
<td>mAh</td>
<td>Mili Amphere Hour</td>
</tr>
<tr>
<td>MC</td>
<td>Mobile Cloud</td>
</tr>
<tr>
<td>MV</td>
<td>Metric Value</td>
</tr>
<tr>
<td>NFC</td>
<td>Near Field Communication</td>
</tr>
<tr>
<td>NRT</td>
<td>Non Real Time</td>
</tr>
<tr>
<td>OS</td>
<td>Operating System</td>
</tr>
<tr>
<td>OSI</td>
<td>Open System International</td>
</tr>
<tr>
<td>P2P</td>
<td>Peer to Peer</td>
</tr>
<tr>
<td>QoS</td>
<td>Quality of Service</td>
</tr>
<tr>
<td>RAM</td>
<td>Random Access Memory</td>
</tr>
<tr>
<td>ROM</td>
<td>Read Only Memory</td>
</tr>
<tr>
<td>RSSI</td>
<td>Received Signal Strength Indicator</td>
</tr>
<tr>
<td>SDK</td>
<td>Software Development Kit</td>
</tr>
<tr>
<td>SIM</td>
<td>Subscriber Identity Module</td>
</tr>
<tr>
<td>SINR</td>
<td>Signal to Interference and Noise Ration</td>
</tr>
<tr>
<td>SOHO</td>
<td>Small Office Home Office</td>
</tr>
<tr>
<td>SWNET</td>
<td>Social Wireless Network</td>
</tr>
<tr>
<td>TCP</td>
<td>Transmission Control Protocol</td>
</tr>
</tbody>
</table>
TDLS  Tunneled Direct Link Setup
TR1  Tree with 1 Relay node
TR2  Tree with 2 Relay node
TR3  Tree with 3 Relay node
UDP  User Datagram Protocol
UI   User Interface
URL  Uniform Resource Locator
USB  Universal Serial Bus
VM   Virtual Machine
WiFi Wireless Fidelity
WLAN Wireless Local Area Network

\(N\)                           set of cloud nodes in mobile cloud
\(t_s^x\)                      start time of download for node \(x\)
\(t_e^x\)                      end time of download for node \(x\)
\(t_p^x\)                      end time of download for node previous to \(x\)
\(\Delta t_x\)                 download time of a packet for node \(x\)
\(\Delta T_x\)                 total download time of the content for node \(x\)
\(\Delta T\)                   average download time of all the nodes
\(\Delta t_{trans}\)         average transmission delay
\(R_s\)                      system throughput
\(B\)                         bandwidth
\(C\)                         total number of packets for a content
\(L\)                         number of the last node
\(S\)                         content size
\(t_L\)                       total time for a packet to traverse to the last node
1. INTRODUCTION

In the recent years, technological advancement in all sort of wireless and mobile communication technology has increased tremendously. This achievement has provided people with ubiquitous connectivity regardless of time and location. High speed wireless connectivity and drastic improvement in the multimedia technology allow rich media content like high definition (HD) videos and pictures to be easily distributed and available at finger tips. Moreover, the immense growth of social media networking in the last couple of decades has changed the way of how people interact with each other. With the help of this social networking, not only personal friends are connected to each other but people with common interest are easy to find and they can readily share ideas and views with each other in an effective manner. This has further elevated the demand for multimedia services [1] due to the impressive penetration of social networks, it is more likely that the content is accessed more frequently within a small geological area of people.

The technological advancement has also revolutionized the industry of handheld devices. The concept of phone has changed from a just voice calling device to sophisticated smartphones which are portable computer by themselves. Smartphones are powerful yet cheap enough to be in the financial reach of an average human being. They are also equipped with multiple technologies such as processing capabilities with number of CPUs (Central Processing Units), multiple telecommunication technologies, large batteries with high power capacities, specialized cameras and sensors, larger display etc. These technologies can be considered as valuable resources that can be used collectively to benefit some or all the nearby devices. In addition to this, people nowadays carry several of these smart devices which have numerous resources to perform a task more intelligently and efficiently.

As a result of the above mentioned developments, cellular networks are expanding in terms of geographical area as well as the number of subscribers. Subsequently, cellular networks require sufficient telecommunication resources to sustain the current demands and also to cope up with the future requirements. One of the main resource is the usable frequency spectrum which is consistent but limited. Moreover, each frequency band has some advantages and limitations such as coverage area and data rates. Therefore, researchers are seeking new ways to utilize the limited resources in an efficient manner by exploiting the use of multiple frequency band simultaneously.

Conforming to the above discussion, this thesis work demonstrates the practical implementation of a system called Mobile Cloud (MC) that exploits the advantages of multiple frequency bands, adjusts according to the environment and uses cooperation between nearby devices to utilize the available resources in an efficient manner. Additionally, it also provides the opportunity to distribute the content among the devices with better system throughout and energy efficiency.

1.1. Motivation

Due to rapid increase in the number of smart devices, demand for rich media content, diffusion of social media and the upcoming network of Internet of Things (IoT) [2], telecommunication industry is facing new challenges and seeking new methods and
system models to cope up with the future requirements. Network operators are struggling to accommodate the high demand of cellular data without compromising on quality of service and number of subscribers. Researchers are finding new ways to utilize the multiple access technologies in the smartphones such as 3G, 4G, WiFi (Wireless Fidelity), Bluetooth, NFC (Near Field Communication) in a cooperative manner. Certainly, these technologies have some benefits, limitations and constraints. Some have limited coverage area or range but provide better data rates while other can communicate over several kilometers in trade of the data rates and power consumption. Besides this, smartphones are equipped with bigger touch screen displays and power hungry application which limits the battery performance to just a couple of days as investigated in [3], [4] and [5]. Evolution in battery technology is slower than the evolution of the integrated circuits as predicted by Moore’s law. The amount of energy a battery can store is doubling every 22 years whereas the computational power of integrated circuits is doubling every two years [6]. Battery contained in these is limited by its size and weight and therefore considered as the scarce resource. Moreover, with more common use of social media, one significant issue to consider is the content repetition and sharing. As we are much indulged in social media these days, contents are more likely to repeat within friends or in a group of people with same interest. Here, content can be an electronic newspaper, video or any sort of media. For instance, several people in an airport would like to read the newspaper at the same time while other might be interested to get live streaming of football world cup match. Therefore, several users are individually interested in the same content from base station by acquiring individual user bandwidth. Also the user’s device is consuming more power due to long distance communication.

In this regards, D2D (Device to Device) communication offers promising gains such as cellular data offloading, energy savings and bandwidth efficiency [7]. D2D also offers efficient distribution of popular content in short range. Moreover, to enable new paradigm for the communication, an initiative has been taken by 3GPP (3rd Generation Partnership Project) to introduce D2D communication in 5G technologies. For this, D2D communication is also included in 3GPP LTE (Long Term Evolution)(Release 12) [8]. D2D communication can be classified as in-band and out-band [9] D2D communication. In in-band D2D communication, licensed spectrum is used for both D2D and cellular links. In addition to this, there can be dedicated as well as shared radio resources for both cellular and D2D link. The primary disadvantage of using in-band communication is the interference cased by D2D links to cellular links and vice versa. In contrast, out-band communication has a leverage of using unlicensed band therefore eliminating the interference between both links. One of the key challenges in these D2D networks is the selection of the anchor node\textsuperscript{1} for the content distribution within a group of nodes. This anchor node can be selected based on various parameters such as energy, SINR (Signal to Interference and Noise Ratio), channel condition, etc.

\textsuperscript{1}In this thesis work, the terms anchor node, cluster head, cloud leader are interchangeable and used in the same context.
1.2. Related work and contribution

In recent years, much work has been done some of which related to cellular traffic offloading while other focused on D2D communication. Some of these works are discussed below.

D2D communication and its promising gains are discussed in [7]. In this article, several techniques and architectures have also been proposed for D2D communication in conjunction with cellular network assistance. This proposed architecture is simulated with slight changes in [10] and [11] whereas implementation is discussed in [12]. In this type of architecture, a D2D server residing in cellular network is mostly responsible for establishing the direct communication link between different devices rather than devices build link by themselves.

Similar kind of work is also done in Sangam architecture [13] where a D2D server is responsible for the group formation and it is assumed that devices making the group are interested in downloading the same content from the internet. Moreover, in this architecture, the content is divided into number of chunks and each device will download the specific number of chunks depending on a battery and/or CPU performance. After completion, the downloaded chunks are shared with other devices over WiFi ad-hoc connections using access point. Another yet similar technique is discussed and implemented in [14]. In this technique, content is also divided into several chunk and each device download specific number of chunks using cellular connection. The only difference in this architecture is the use of WiFi direct P2P (Peer to Peer) connectivity for further distributing the downloaded chunks to the other group members.

A different approach is used in [15] to offload the cellular traffic onto the local WiFi D2D communication. This architecture is called Subscribe-and-Send. In this architecture, devices subscribe on Content Service Provider for intended content but do not download it from cellular network until it meets the subscription deadline. Before this deadline, if other device in close proximity appears to have the same content available on its local storage, then this content will be downloaded by WiFi connection. Otherwise the device will download the content from the cellular connection and upon completion, device will also check if some other node has subscribed to the same content to distribute further.

Last but not the least, another related work is done [16] to download the content from internet using WiFi and cellular link interfaces simultaneously. In this work, content is divided into different chunk sizes depending on the interface. Higher the throughput of the interface, bigger will be the chunk size. This implementation is limited to only one device however it expandable to several devices as well.

This thesis work is part of the project called Cognitive and Intelligent Solutions for Testing and Monitoring of Future Access Technologies. The main focus of this project is on the heterogeneous and virtual networks. Therefore, to the best of our knowledge, the approach we have used in thesis work is different than the ones discussed in the above related works. Keeping in view the above motivation, the main factors being considered and optimized in this thesis work are the energy efficiency and system throughput. These two factors can be optimized in several different ways. Therefore, the main contribution and goals of this thesis work is in several aspects which are

1. An out-band D2D communication network using WiFi,
2. Self-organizing D2D network, and
3. Different content distribution techniques.

1.3. Structure of the thesis

In Chapter 2, we provide a comprehensive overview of cooperative and cognitive system of Mobile Cloud, its benefits and also shed some light on Mobile Cloud’s applications. Specifications of the entities in Mobile Cloud, their functionality and working of the overall test-bed system are briefly described in Chapter 3. Chapter 4 focuses on the performance statistics of the system, highlighting the strengths, limitations and major overheads. Moreover, benchmark of the Mobile Cloud performance against the legacy approach and recommendation of the new functionalities in the existing system are also discussed in Chapter 4.
2. COOPERATIVE AND COGNITIVE RADIO COMMUNICATION

2.1. Overview

A task can be performed much faster and in a much better way by efficiently using the resources in a cooperative manner. Cooperation is two or more objects working together, usually to obtain a common goal. Consider example of two car factories. Both factories contain different number of resources in terms of machinery for assembling the car. Each machine takes different amount of time to complete the process. Intuitively, some of the machinery will remain idle until the previous process is complete if the factories are working independently. The number of cars produced per day could be increased if both factories cooperate by sharing the machinery of each other and thus reducing the idle time. This idea is also used in processors which is called multi-threading. In multi-threading, a bigger task is performed by several number of cores by dividing the task into small parts and by using the cooperation. One a bigger scale, same is the idea used in cloud computing which is widely implemented in many different areas. Cloud computing involves computing by using the resources which are distributed over the network and these resources are shared within the cloud hence all interconnected to each other.

If we consider the old generation of cellular network architecture such as 1G and 2G cellular networks, we observe a centralized entity in each of the network architecture which is responsible for most of the critical tasks in a network. As we move ahead in the new generations of cellular network technologies like 3G, 4G and upcoming 5G, we notice a distributive trend in the network architecture [17]. The role of the entities are further divided or distributed to new entities. This is mainly done from core network till the base station. From the base station till the user equipment; each user equipment is treated individually and no cooperation is done between them. There are a number of resources available at the user equipment such as network links (multiple wireless interfaces), radio resources (space, time, frequency and power), sensors, storage, energy, etc. Therefore by utilizing these resources, we can further enhance the network performance and can make the network more efficient.

Now comes the cognitive part. Cognition is a cyclic process of observing or sensing the environment, process the collected information, understand the results and react accordingly [18]. Cognitive cycle is shown in Figure 1. The literal meaning of cognition in engineering language is to get input from some sort of sensors, process the data from the sensors and act accordingly. The word smartphone is used with the mobile device if it has basic features of phone as well as advance computational and communication capabilities. Moreover smartphones these days are more equipped with many sensors which can sense the environment around them and can act accordingly. For instance, a group of smartphones can sense who has the best channel condition for cellular link among them and then make it as an anchor node for using the cellular network.
2.2. Mobile Clouds

Based on the discussion above and focusing to investigate the power and bandwidth efficiency, there was a need of a communication network architecture that can exploit the best qualities of individual access technologies in terms of battery and power efficiency. There are several architectures which have been proposed earlier with promising simulation results. One of them is cognitive cooperative Mobile Clouds [19]. A Mobile Cloud is a collaborative arrangement of mobile devices called Cloud Nodes (CN) that communicate with each other through short-range links while devices could also be connected to cellular base station using cellular links [19]. The node(s) which has the optimum resources serves as cloud leader (CL). These resources can be processing power, energy, sensors, etc. Moreover, the CL can also act like a relay node between the cellular network and Mobile Cloud. All the nodes within a cloud can be connected via any short range access technology such as WiFi or Bluetooth. Figure 2 shows the generic architecture of Mobile Clouds.
Mobile Clouds can be further classified into three major types of architecture [20]. These architectures are differentiated based on the decision-making entities which are

1. **Central cellular controlled:** In this architecture, base station is responsible for managing and controlling all the cooperative communication between the nodes.

2. **Distributed cloud leader:** In this architecture, CL is the decision making entity and it is responsible for managing the cooperation between the nodes.

3. **Hybrid:** This architecture shares the role of decision-making entity between the base station and the cloud leader. Base station is responsible for the formation and management of the Mobile Cloud while CL is responsible for the cooperation of nodes for content distribution.

In this thesis work, the distributed cloud leader architecture is implemented and its performance is analyzed. Furthermore, there are several cooperation strategies that can be developed for each of the Mobile Cloud architectures discussed above. These cooperation strategies are forced, altruistic, egoistic, social and embedded technical cooperation [18]. For instance, embedded cooperation strategy is developed and implemented in this thesis in which only one node (CL) downloads a video file from the base station while other nodes (CNs) cooperate to distribute the file within the cloud. In contrast to this, traditional cellular network architecture supports nodes only to download the content individually from the base station.

With the Mobile Cloud, the idea of social caching as termed Social Wireless Network (SWNET) [21] can also be implemented. In social caching, the contents which are high in demand in an area such as a university department campus, an airport or the content might be interesting for the people with common interest; these contents can be stored in smartphone and can be made available to download for the people nearby at no or very low cost. This social caching along with the Mobile Cloud can also save significant amount of battery power [22] as well as bandwidth [23].

There has been plenty of theoretical research done on cooperative networks and social caching as explained in the related work however practical aspects of these architectures have not been much explored yet. Several practical parameters need to be addressed during realization of theoretical models. In this thesis work, we will propose a design of a Mobile Cloud testbed which is also practically implemented. The testbed demonstrates a data transfer application for self-organizing Mobile Clouds and also incorporating concept of social caching. The main purpose of the application is to provide a practical testbed for researchers to plug-in their theoretical models of cloud formation and social caching. The practical results can then be compared with the theoretical models. The comparison can then lead to identification of practical parameters which caused deviation from their theoretical results.

### 2.2.1. Applications

Mobile Cloud can have numerous applications in different areas based on the principles of cognition and cooperation. Some of these applications are discussed below.
Traffic offloading

Mobile Cloud can be used for offloading the cellular data traffic for the popular video content. Video these days are of very high quality or high definition. Therefore these HD videos have bigger volume/size in terms of bytes and also cause the smartphone battery to drain more quickly. Also these videos can occupy reasonable bandwidth when downloaded from the internet via cellular links. A white paper on global mobile data traffic forecast by Cisco [1] states that major portion of the mobile data traffic is comprised of video streaming and this will continue to increase in the upcoming years. Therefore by using Mobile Clouds, we can save bandwidth as well as battery by cooperation and using the short range communication links. In addition to this, the nodes that were not able to afford the high definition video can also enjoy it.

Efficient content distribution and sharing

Social networking has had a tremendous impact in our lives in last decade and with this increases, sharing the content within the group of people having the same interest can have notable impact on saving resources. Mobile Cloud can also provide a small social network for the people sharing the common interest. A good example is of the airport. Instead of each person reads the news from the internet individually via WiFi or cellular network resulting in the content repetition, content can be downloaded only once and shared with the other people sitting nearby.

3D video and photography

3D (three dimensional) pictures and videos are very popular and demanding these days. It also requires a special hardware and software to achieve the 3D effects in pictures and videos. In this regards, Mobile Cloud can provide a feasible solution by using multiple smartphone’s camera in a synchronized manner in order to capture 3D pictures and videos.

Portable surround sound system

An easy and interesting application for Mobile Cloud is the portable surround sound system without any need to specialized hardware. The only requirement for this application is the synchronization between the audio speakers of multiple smartphones. For instance, three smartphones synchronized together, each producing different audible tone, can achieve a surround sound effect.

Interactive classrooms

Another useful application of Mobile Cloud is in interactive classrooms. It can be used in the school where teacher can share the course content with the students carrying a tablet computer and can evaluate the performance of each student in real-time without getting in paper hassle. This can also be used to evaluate the performance of the teacher by using learning analytics. In this way, Mobile Cloud can be useful for immediate feedback from teacher to the students as well as from students to the teacher and the school management.
3. TEST-BED DESIGN AND IMPLEMENTATION FOR MOBILE CLOUD

3.1. Overview

In this section, we provide several key design details that make the implementation of our Mobile Cloud system compliant to the existing cellular infrastructure as well the upcoming technologies. As of today and to the best of our knowledge, our system model is novel and unique in its nature, yet its simplicity serves as the proof of our concept. Let us have a brief look on the key objectives of this implementation work.

1. First objective is the formation of Mobile Cloud using distributed cloud leader architecture in which CL is responsible for the cooperation and management of all CNs.

2. To bridge uninterrupted communication between cellular network and Mobile Cloud via CL by using multi-radio interfaces.

3. The distribution of non-real time video content in the cloud by using four different cooperation strategies. This is the primary objective on which most of the system performance relies on.

4. Last but not the least, management of CNs and selection of new CL upon withdrawal of present CL.

The implementation of Mobile Cloud architecture requires both hardware and software. Hardware and software are closely related to each other and selecting the best combination of both is one of the critical part of any project. The overall structure of Mobile Cloud test-bed is shown in Figure 3.

Figure 3. Overall system architecture.
3.2. Software platform

Generic definition of software or computer program is the set of any instructions that direct a computer to perform any operation [24]. Software can be categorized into two main classes, system software and application software. System software is also called operating system (OS) that is designed to run on computer’s hardware, e.g., Windows OS, Linux, Mac OS, etc. Application software or simply app is designed to perform user desired functions and tasks e.g., internet browser, video player, spreadsheet application, etc. Application software are dependent on system software which means that the application software are designed for specific system software.

In this implementation work, Mobile Cloud application software is the program designed to achieve the above mentioned objectives. As application software is dependent on OS, therefore, it is crucial to select the appropriate OS that provides enough functionality to obtain the optimum results. There are couple of choices available for OS but the major market holders are Android by Google and iOS by Apple. A brief comparison is given in Table 1.

Based on the above discussion, Android appears to be the optimum solution for this implementation work. Therefore, Android as a software platform is used to implement the Mobile Cloud application. There are few important things need to be addressed about Android. These are as follows:

- Android consists of layered environment based on the foundation of a Linux based kernel long-term support (LTS) branch as shown in Figure 4.

![Android system architecture](https://source.android.com)

- All the Android applications are run within an instance of software called Dalvik Virtual Machine which is also an open source technology.

---

1. [https://source.android.com](https://source.android.com)
2. Jailbreaking is the process of removing software restrictions imposed by iOS. Jailbreaking permits root access to the iOS file system and manager allowing user to customize some of the system files [25].
Table 1. Comparison between Android and iOS

<table>
<thead>
<tr>
<th>Factors</th>
<th>Android</th>
<th>iOS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Kernel</strong></td>
<td>Linux</td>
<td>OS X and Unix</td>
</tr>
<tr>
<td><strong>Source model and customization</strong></td>
<td>Open source¹, Completely customizable</td>
<td>Closed source. No driver or OS level customization is permitted unless jailbroken².</td>
</tr>
<tr>
<td><strong>Programming language</strong></td>
<td>Java and part of code can be in C/C++</td>
<td>Objective C</td>
</tr>
<tr>
<td><strong>Software development kit</strong></td>
<td>Android studio and Eclipse. Each versions of Android has several SDK version or API level. There is compatibility issues with SDK versions</td>
<td>Xcode is used for app development but require Apple machines and iOS licenses. Fewer and compatible SDK versions</td>
</tr>
<tr>
<td><strong>Supported Hardware</strong></td>
<td>Can be ported on any customized development board</td>
<td>Runs only on Apple devices</td>
</tr>
<tr>
<td><strong>Technical support</strong></td>
<td>Extensive technical support available on the internet</td>
<td>Extensive technical support available on the internet</td>
</tr>
<tr>
<td><strong>Applications packages</strong></td>
<td>Due to open source OS, most of the open-source community targets Android for app development. Hence we can find various tools, apps and packages with source code available to reuse. This helps in rapid application development</td>
<td>Packages are usually available with some price tag. Most applications are also closed source</td>
</tr>
<tr>
<td><strong>Language learning curve</strong></td>
<td>Relaxed</td>
<td>Steep</td>
</tr>
</tbody>
</table>

- The security of Android OS is also superior in comparison to other mobile phone operating systems such as iOS and Windows. This is enabled by sandbox in which applications are run in a secluded part of the operating system. Applications require permissions in order to access phone resources such as network, memory, installation. Even before an Android application is installed, the operating system is able to determine all types of resources that are required by it. This prompts the user to install application only in circumstances that he or she may trust whatever access that it might require.

3.3. Hardware platform

Hardware is one of the key factors defining the performance of Mobile Cloud application. Following we describe the hardware used in this thesis.
3.3.1. Samsung Galaxy S4

In Mobile Cloud, CN plays a vital role in the overall performance of the system. If CNs have less resources, e.g., processing power, RAM (Random Access Memory), etc., the overall system performance can get effected. Therefore, in order to show the absolute potential of our proposed system model and capability to comply the future requirements, CN must have impressive hardware specification and features. In this implementation work, we are treating smartphone as CN. At the time of the thesis, there were several options available to choose from the smartphones however the best among them was the Galaxy S4 Android smartphone by Samsung. Some of the key technical specifications [26] of this smartphone are given below:

1. Qualcomm Snapdragon 600 Quad-core processor with clock speed of 1.9 GHz.
2. System RAM of 2GB.
3. Internal memory of 16GB and can support up to 64GB as an external micro SD card.
4. It supports 3G HSDPA 850 / 900 / 1900 / 2100 as well as 4G LTE 800 / 850 / 900 / 1800 / 2100 / 2600 cellular networks. Also it is compatible with legacy GSM (Global System for Mobile Communication) network.
5. It can support up to 42.2 Mbps on HSDPA (High Speed Downlink Packet Access) and 5.76 Mbps on HSUPA (High Speed Uplink Packet Access). On LTE network, it can support up to 100 Mbps downlink and 50 Mbps on Uplink.
6. For WiFi, it supports 802.11 a/b/g/n/ac standards.
7. Android Jelly Bean (v4.3) operating system.
8. Lithium-Ion battery with 2600 mAh (Mili Amphere Hour) capacity.

3.3.2. Linksys wireless router EA6300

In computer networking, router is a device that connects multiple networks and transfers data between them. Routers these days come in large variety. Some are connecting large enterprises while others are for home and office use. The ones used for home and office are usually termed as SOHO (Small Office Home Office) devices. These routers are capable of performing multiple tasks and have several functions depending on the usage e.g., access point, switch, bridge etc. In this implementation work, Linksys wireless router EA6300 is used as a wireless access point. Wireless access point is a device that connects multiple wireless devices together. In context of our implementation work, wireless access point is connecting CNs all together using WiFi protocol. The main reasons for using this particular router are, 1) it supports the latest 802.11ac WiFi standard which can achieve data rate up to 1.3Gbps [27], and 2) it supports new 5GHz band having the advantage of less interference than the predecessor 2.4GHz band. Most of the WiFi devices support 2.4GHz band whereas 5GHz band is comparatively new and less number of devices support to operate in this band. Therefore we
can expect to have less interference due to less number of 5GHz supporting devices in the surroundings.

### 3.3.3. Mobile Cloud content server

In information technology, a server is a computer program or a machine that awaits to fulfill the requests from other machines or software (clients) [28]. The purpose of a server is to store and share the content and data among the clients. For instance, when we type `www.google.com` in web browsers, a request is sent to a server with specific IP address. In response to this request, the server returns the code for webpage that displays the website. In this implementation work, the video contents with different sizes are being stored in the server provided by University of Oulu. This server provides the specific content upon request and can be accessed by a website link via internet. However, this server is used only for demonstration purpose and might not be available in the future. Therefore, any content can be downloaded with valid website link if required in future.

### 3.4. Multi-radio communication

Multi-radio communication is the main foundation of this thesis on which all the performance enhancements are based on. By the term multi-radio communication, it is meant that multiple radios can work simultaneously side by side without interrupting with each other. In this thesis we are concerned only with the radio of two interfaces and that are WiFi radio and cellular network radio (e.g., 3G or 4G). Considering the main objective of this thesis as described in the above sections, multi-radio communication is essential so that we can download the content from the internet via data connection of the cellular network and at the same time, we can distribute the content to the other nodes in the Mobile Cloud as well via WiFi links. By using multi-radio communication, we can communicate with the mobile network and meanwhile also exploit the benefits of short range communication i.e., WiFi.

In this thesis, we use a modified version of Android Jelly Bean v4.3 in our smartphones. The reason behind this modification of the Android OS is its own security. There are some functions which are not allowed to be controlled programmatically and they can only be performed by the OS itself. Note that multi-radio communication is not allowed in Android. Android has set priorities for different network interfaces and WiFi has high priority than the cellular network. In stock version of Android, one of the following three case is possible:

1. **Case 1:** Given that WiFi is connected, then turn on cellular data connection. Cellular data connection will be automatically torn down.

2. **Case 2:** Given that cellular data connection is active, then turn on the WiFi connection. Existing cellular data connection will be torn down automatically.

3. **Case 3**: Given that WiFi is connected, then WiFi is disabled by user. System will check if cellular data connection is enable. If it is true, the system will automatically reconnect to the cellular data connection.

Therefore both WiFi and data connection on cellular data connection cannot work simultaneously. One important thing to mention here is that the application for the Mobile Cloud is working on the application layer in the OSI (Open System International) layered model and Android does not allow users to access the physical or data link layer even if the device is rooted\(^4\). This is mainly due to security, stability and layered structure of the Android OS itself. Therefore the only way to obtain the multi-radio communication in Android OS is to modify the kernel of the operating system itself. However this is not very straight forward and can lead to some serious damage to the device as well.

Android OS is designed to optimize the battery power therefore Android OS recommends to keep only one interface active. In this thesis work, multi-radio communication is achieved in two steps. In first step, it is necessary to change the source code of Android OS by modifying the module which manages the network connections. This module is named as *Service* in the source code. The service module further contains a Java class called *ConnectivityManager.java* which is responsible for handling the WiFi and cellular network connections. Furthermore, in this ConnectivityService class, a handler [30] called *NetworkStateTrackerHandler* tracks the state of the two interfaces by using **handleMessage** method. We implemented a new statement that if the 3G interface is active and WiFi is available, then 3G should remain active so that we can use both the interface simultaneously.

Before proceeding to the second step, let us have a brief review of the following concepts and terminologies.

- **Routing tables**
  In computer network, routing tables defines the route to a particular destination or IP. Routing table contains data or list of routes to particular destination with the metric associated with this route. The metric can be composed of many things e.g., distance, hops, loss etc. Routing tables are stored in computer memory and are managed by the OS. Whenever a node wants to communicate with another node, it searches the destination IP address in its routing table. If node finds the address in the routing table, it transmits the packet on the respective route and if node does not find the destination address in the routing table, it forwards the packet to the default gateway.

- **Default gateway**
  In a general sense, a gateway is a node or a device on a computer network that is used to connect one network to another network or we can say that gateway serves as an access point to another network. Local area network or sometimes called local network is the group of the nodes which share a common communication line and the common example is some number of devices connected to a wireless router used in our homes. And all the nodes communicate with each other via their MAC (Medium Access Control) and IP (Internet Protocol)

---
\(^4\)Rooting is the process of allowing users of Android operating system to acquire the privileged control over various Android subsystems [29].
addresses. So if a node wants to communicate with another node which is not in its local network or not listed in its routing table, then a request is sent to default gateway. Usually, in home networks, the default gateway is the IP address of the router itself.

Since both connections are active now, as a second step, the problem is to direct the specific data or packet to the desired network interface. As we know, all the nodes in the cloud communicate with each other at regular intervals and the content from the internet needs to be downloaded without interrupting the connection between all the cloud nodes. In order to download the content from the internet without interrupting the communication within the nodes, we modify the kernel routing table in the smartphone. The detailed procedure can be found in Appendix 1. We delete the existing default gateway in the routing table and insert new default gateway with the IP address assigned to the smartphone by the cellular network for data connection. Now whenever there will be a need to access the internet to download the content, the request will be sent to cellular network without interrupting the communication between the other nodes in the cloud.

### 3.5. Mobile Cloud formation phases

The Mobile Cloud operation has three main phases which are

- Cloud Formation
- Content Distribution
- Cloud management.

Different architecture for Mobile Cloud can use different techniques for each of the phase and the entities involved in it, but the basic concept of each of the phases is the same for all the architectures. In this thesis, we only focus on the distributed cloud leader (self-driven Mobile Cloud) architecture. The general architecture of our Mobile Cloud application is shown in the Figure 5.

![Figure 5. Mobile Cloud application model.](image-url)
Before going into details of each phase, it is important to look at the assumptions that have been made.

### 3.5.1. Assumptions

Following are the assumptions that have been made for this self-driven Mobile Cloud architecture:

- All the nodes/devices are connected to each other and communicating through WiFi.
- There is one CL in the self-driven architecture. CL is selected based on the given criterion e.g., processing power, available battery, etc.
- All the nodes are interested in the same content e.g., typical in social networks.
- The content is non-real time (NRT) e.g., picture, Word document file, etc.

### 3.5.2. Cloud formation

The cloud formation phase can be further divided into two stages:

(a) **Service and neighbor discovery**

In this very initial stage, all the nodes in the cloud broadcast their IDs and the metric value. The Metric Value (MV) is a number generated by the combination of different parameters for example RSSI (Received Signal Strength Indicator) from base station to cloud node, available battery level, computational power, available free space for content, SIM (Subscriber Identity Module) data package for cellular network and power supply charging status. These parameters can have different weight for the metric value depending upon its importance and also according to the need. However, for simplicity, we are taking only two parameters into account i.e., the available battery level and AC (Alternating Current) charging status. In this way, all the nodes get information about their neighbouring nodes. Mathematically, MV can be represented as

\[
MV = \begin{cases} 
(Available \text{ battery level} + 10), & \text{if connected to AC charger} \\
(Available \text{ battery level}), & \text{otherwise}
\end{cases}
\]

(b) **Cloud leader selection**

It this stage, every node selects its cloud leader based on metric value. One important issue to be noted is that each node is independent of selecting its own cloud leader. Therefore, there can be multiple cloud leaders within the cloud if we incorporate other parameters for calculating the metric value. For instance, if we refer to Figure 6, CN 2 can make CN 1 as its CL while CN 4 can make CN 3 as its CL. Further more, two cloud leader can be used to download the content from two different networks hence balancing the network load. For simplicity in our
practical implementation of test-bed, we consider only one Mobile Cloud and one cloud leader. This is elaborated more in Figure 6.

![Figure 6. Cloud formation and CL selection.](image)

### 3.5.3. Content distribution

Content distribution phase is the core part Mobile Cloud implementation. If we consider the legacy setup in which each node in connected to the cellular network and given that all the nodes are interested in the same content, then each node will download its own copy from the internet by using its cellular connection. Therefore by using Mobile Cloud, content can be distributed more efficient by just downloading the content once and distribute it among the other nodes by using WiFi.

As we are focusing on the group of people who have some common interest, therefore it is likely that popular content is already available with some node in the cloud. Therefore rather than using the cellular link, content can be directly fetched from node using WiFi. If the content is not available within the cloud, then the content can be downloaded from the internet using cellular link.

In Mobile Cloud, the basic idea is to distribute the same content to all the nodes in the cloud and intuitively the best approach is to broadcast the content to all the nodes. However in WiFi networks, broadcasting is not a reliable solution. It requires specialized mechanisms in order to achieve successful content transfer as well as notable performance gain. To achieve reliable broadcasting in WiFi networks, the optimal solution is to use application specific Network Coding. Considering the scope of this thesis work, content is distributed using unicast communication among the nodes.

Content distribution can be categorized into two main types and we name these as

1. In-cloud content distribution.
2. Out-Cloud content distribution.

### 3.5.4. In-cloud content distribution

If we consider a group of users which have common interest and one wants to share some content with others, this can be done by in-cloud distribution technique. In this technique, the content is already available in the cloud and it can be present at any node and therefore any other node in the cloud can see and download the content directly from the parent node without using the mobile network resources. For instance, consider Figure 7, in this scenario, CL requests content from CN 4. Similarly, CN 2 requests content from CN 3. In this way, we can cache the popular content within the cloud and save bandwidth from the cellular link. Moreover, nodes can also save battery power by communicating over a short range links.

### 3.5.5. Working algorithm

Working algorithm for in-cloud content distribution technique is fairly simple. Each node in the cloud gets the list of the available content on every other node. Once a node decides to download the content from other node, it sends a request to the parent nodes for the desire content and then it can directly download the content from the parent node. This is also shown in the Figure 7.
3.5.6. Out-cloud content distribution

Content can be downloaded from the internet via CL if the desired content is not available locally. As shown in Figure 8, whenever a node wants to download content from the internet, a request will be sent to CL, content will then be downloaded by CL via cellular link and distributed to all the nodes in the cloud.

![Figure 8. Out-Cloud content distribution.](image)

We have demonstrated four methods for the content distribution which are

1. Star
2. Tree with one relay node (TR1)
3. Tree with two relay nodes (TR2)
4. Tree with three relay nodes (TR3).

In order to elaborate further on the implemented content distribution techniques, let’s define following terminologies:

- \( N = \{1, \ldots, n\} \) as set of CN in MC
- \( t_s^x = \) start time of download for node \( x \in N \) \( t_s^1 < \ldots < t_s^n \)
- \( t_e^x = \) end time of download for node \( x \in N \) \( t_e^1 < \ldots < t_e^n \)
- \( t_e^p = \) end time of download for node previous to \( x \)
- \( \Delta t_x = \) download time of a packet for node \( x \)
- \( \Delta T_x = \) total download time of the content for node \( x \)
- \( \Delta t_{\text{trans}} = \) average transmission delay
Since the whole content is distributed in the form of packets, therefore, the above terminologies represent time for a single packet transfer unless mentioned in the description. Moreover, considering node $x$, $t^p_e$ will be different for different content distribution techniques.

**Star topology**

In this method, CL is only responsible for the content distribution. The content is distributed in descending order of MV. In this case, if we consider CN 4 and CN 7, their respective $t^p_e$ will be $t^3_e$ and $t^6_e$. It is depicted in Figure 9.

![Figure 9. Content distribution in star topology.](image)

If a node needs a content from the internet, it sends a request to the cloud leader. The cloud leader will receive the request, download the content from the internet and distribute it to all the other nodes in the cloud as shown in the figure. Mathematically, we can represent total download time for a content as

$$ t^x_s = t^p_e + \Delta t_{trans} $$

(1)

$$ \Delta t_x = t^p_e - t^x_s $$

(2)

$$ \Delta T_x = \sum_{i=1}^{C} \Delta t_x(i) $$

(3)

where $C$ denotes total number of packets for a content.

**Tree topology**

In this distribution method, CL as well as relay nodes are responsible for the content distribution in the cloud. Three variations of this method have been demonstrated in this implementation depending on the width of the tree or relay nodes. They are named as TR1, TR2 and TR3. The tree formation algorithm is the same for all the methods which is shown as flow chart in Figure 10.
In this distribution technique, all the nodes in the Mobile Cloud are arranged in a tree structure in such a way that the parent node has the higher MV than the child node. In this way, a logical tree structure is formed with the descending order of their MV. However, the content will still be downloaded via cloud leader.

The mathematical representation for this method is the same as expressed in (2-4). However, total time for a packet to traverse to the last node is different and hence affecting the total download time for a content. To elaborate more on this, consider the total time for a packet to traverse to the last node $t_L$ which can be computed as

$$t_L = t_1^* - \Delta t_{\text{trans}} + f(L) \cdot (\Delta t + \Delta t_{\text{trans}}), \quad (4)$$

where $f(L)$ represents the function for calculating the number of previous nodes. This $f(L)$ varies with different topologies as we will see later. Let us now discuss each of these three methods and examine their advantages and drawbacks.

1. **TR1:** In this method, there is only one relay node between each node and therefore each node distributes content to only one node as shown in the Figure 11.
In this scenario, the total time for a packet to traverse to the last node $t_L$ can be calculated as

$$t_L = t_s + n\Delta t + (n - 1)\Delta t_{\text{trans}},$$

where $n$ shows the total number of nodes. Theoretically, this method takes the longest time for a packet to reach to the last node. However, in terms of energy efficiency, every node in this method contributes equally to distribute the content except the last node which has the lowest MV.

2. **TR2:** This method has tree width of two and therefore uses two relay nodes. In this way, every node distributes the content to further two other nodes as shown in Figure 12.

---

**Figure 11.** Content distribution in TR1.

**Figure 12.** Content distribution in TR2.
If we consider CN 4 and CN 7 in Figure 12, their respective $t_e^p$ will be $t_e^3$ and $t_e^6$. Therefore, for this case, $t_L$ can be calculated by using (4), where

$$ f(1) = 1 $$

$$ f(x) = \begin{cases} 
  f(\lfloor \frac{x}{3} \rfloor) + 1 & \text{mod}(\frac{x}{3}) = 0 \\
  f(\lfloor \frac{x}{3} \rfloor) + 2 & \text{mod}(\frac{x}{3}) = 1 \\
  f(\lfloor \frac{x}{3} \rfloor) + 3 & \text{mod}(\frac{x}{3}) = 2 
\end{cases} $$

For example, if we consider $n = 7$, then

$$ t_L = t_s^1 + 5\Delta t + 4\Delta t_{\text{trans}}. $$

In this method, some of the nodes have leverage of not distributing content further e.g., in Figure 12, four nodes are not contributing in the content distribution thus saving energy from the battery. However, the $t_L$ in this method is less than as compared to TR1.

3. TR3: In this method, the width of the tree is three which means that there are three relay nodes. Therefore each node is distributing the content to three other nodes in the cloud. Figure 13 illustrates this case.

![Figure 13. Content distribution in TR3.](image)

In this figure, if we consider CN 4 and CN 7, their respective $t_e^p$ will be $t_e^3$ and $t_e^6$. Therefore, for this case, $t_L$ can be calculated by using (4), where

$$ f(1) = 1 $$

$$ f(x) = \begin{cases} 
  f(\lfloor \frac{x}{3} \rfloor) + 2 & \text{mod}(\frac{x}{3}) = 0 \\
  f(\lfloor \frac{x}{3} \rfloor) + 3 & \text{mod}(\frac{x}{3}) = 1 \\
  f(\lfloor \frac{x}{3} \rfloor) + 1 & \text{mod}(\frac{x}{3}) = 2 
\end{cases} $$

Similarly, if we consider $n = 7$ as an example shown in Figure 13, then

$$ t_L = t_s^1 + 5\Delta t + 4\Delta t_{\text{trans}}. $$
One important thing to mention is that, if we consider (8) and (11), the performance for both cases is the same, however, the difference will get prominent if we increase the number of nodes. Also, in this method, some of nodes are not involved in the content distribution hence saving battery power.

3.5.7. Cloud management

In our architecture, cloud management is done by the nodes themselves. Discovery message serves as a keep-alive message and broadcasted with regular interval. Its purpose is to maintain the cloud if any of the node leaves. It also helps to find if there is any other eligible candidate for CL. When a node leaves the cloud, every other node will wait for a certain time period after which the node will be removed from the neighbor’s list. Similarly, a new CL will be selected if old CL leaves the cloud.
4. MEASUREMENTS AND RESULTS

4.1. Overview

In this section, we analyze the measurements and performance results of proposed test-bed with variant of different distribution techniques. First, let us consider the physical setup of the Mobile Cloud test-bed.

![Setup of the proposed Mobile Cloud test-bed.](image)

Figure 14. Setup of the proposed Mobile Cloud test-bed.

Figure 14 shows the basic setup for entities in the Mobile Cloud test-bed. All the CNs are capable of communicating with each other via WLAN (Wireless Local Area Network) while CL is able to communicate on WLAN as well as the cellular network. In our test scenarios we measured different performance parameters such as download time, system throughput, RSSI effect and bandwidth efficiency. In order to analyze the effect of these performance parameters, we consider two content distribution topologies.

- **Star topology:** In this configuration, CL is only responsible for distributing the content among all CNs in a sequential manner, where distribution sequence depends on the parametric value.
- **Tree topology:** Tree based content distribution follows the classical tree structure for the delivery of the content. CL acts as root node of the tree and triggers the content delivery process which follows by relay nodes. The width of the tree defines the number of relay nodes attached to a parent node.
4.2. General assumptions and environment variables

The following assumptions and environmental variables\(^1\) are considered for execution of test cases in the test-bed.

- We assumed only one Mobile Cloud and a single cloud leader. The node selected as cloud leader operates on the customized Android OS.

- All the nodes are assumed to be interested in the same content.

- It is considered that cloud formation and CL selection is already done for collecting the test results. Therefore, formation of the cloud is always done prior to the execution of each test scenarios.

- No mobility is considered, such that every node in the network is assumed to be stationary.

- Every node in the network operates on the WiFi IEEE 802.11ac standard.

- Contents are stored on a fixed content server which is accessible via cellular link\(^2\).

- We only consider non-real time data transmission.

- The test location of the test cases chosen based on the cellular RSSI values for the CL. In our proposed work, test results are collected over low, normal and high values of RSSI.

- In order to get the complete download time, two time stamps are saved in the log file\(^3\), one at the start and one at the end of the download. The total download time for node \(x\) for the content can be calculated as follow:

\[
\Delta T_x = \text{end time} - \text{start time},
\]

where approximate start time is the time just before downloading the first packet of the desired content and end time is the approximate time just after the reception of the last packet of the desired content.

- Once the total download time is measured, system throughput \(R_s\) can be calculated as follows:

\[
R_s = \frac{S}{\Delta T} \times N,
\]

where \(\Delta T\) represents the mean of total download time for the content of size \(S\) and \(N\) shows the total number of nodes.

---

\(^1\)These are the variable that tend to change the simulation results e.g., the distance from BS to CL, separation between the nodes, D2D communication protocol etc.

\(^2\)http://www.ee.oulu.fi/~amoiz/downloads/

\(^3\)/sdcard/Download/COIN_testResult/COIN_RESULT.txt
4.3. Test cases

4.3.1. Single node download time

Description

The main objective of this test case is to provide baseline results for download time of single node with respect to proposed Mobile Cloud based approach. The contents are accessible through 3G network. For this test case, we consider a single node which is connected to 3G cellular network and therefore total download time is only measured on this node.

Input and number of iterations

The URL\(^4\) (Uniform Resource Locator) of the requested content is accessed through UI (User Interface) of our developed application. For the given test case, the file size is 5MB. In order to have better inside of the results, we took 10-12 iterations of each test case for different locations.

Results

Figure 15 demonstrates the impact of RSSI (dBm) over the download time. It is shown that, the download time increases as RSSI value gets lower. Download time varies randomly with respect to RSSI for short span of RSSI values, but the overall trend of download time versus RSSI is decreasing. For instance, at -109 dBm, the download time is approximately 7s whereas at -108 dBm, the download time has increased up to 19s (approximately). Similarly, at -106 dBm, the download time again decreases to just 5s. Since RSSI is being controlled only by placing the CL at different locations and there is no absolute control on the environment, this therefore introduce the abrupt and random change in download time due to unpredictable behavior of the radio channel, traffic load on base station, performance of the content server, etc. Moreover, under these circumstances, the signal modulation scheme cannot be controlled as well which produces undesirable behavior as it can be observed in the figure. Here, it is important to mention that if the trails (iterations) are performed for large number, the trend of download time can be easily identified which is decreasing with lower RSSI values. Download time has notable change at high RSSI values whereas low RSSI has comparatively low effect on download time.

\(^4\)http://www.ee.oulu.fi/~amoiz/downloads/5mb\_video.mp4
In order to normalize this random effect, we can take the average of the download time over specific of RSSI values. The elaborated results showing RSSI effect on the download time are presented in Figure 16.
Figure 16 shows the average download time taken by a single node to download the content. Figure 17 shows effect of the RSSI value over the throughput. It is worthy to note, that in all these test results presented in Figure 16 and 17, the average download time and average RSSI (dBm) values are considered.

4.3.2. Mobile Cloud download time

Description

For this test case we consider a Mobile Cloud composed of 7 CNs. A cloud leader is selected for the given set of nodes based on the given criterion. The cloud leader is connected to the cellular network. Cellular and WLAN networks are considered for the test scenario. WLAN network is enabled within the Mobile Cloud whereas, the cloud leader is used to bridge between WLAN and cellular network. Here, we assume that the desired content is not available in the local cloud. In the first phase, a node in Mobile Cloud requests a content from internet via cloud leader from the content server. Subsequently, the cloud leader fetches the content and distributes it to all the cloud nodes. The content is distributed by the cloud leader in both star and tree topological manner. In order to minimize the radio channel effects on the performance, each node is kept in a fixed position. Moreover, cellular RSSI value is more prone to the environment changes, thus the position of the CL is selected such that the value of cellular RSSI lies between 55 dBm to 60 dBm. Each node measures its own start and end time of the content downloading process.

Input and number of iterations

Through UI of the application, the URL of the requested content is input to the application. There are five different content sizes with different downloading links. Content sizes are 1MB, 5MB, 10MB, 15MB and 19MB. The number of iterations is 6-8 at a
fixed location for this specific test scenario. The main objective of this test case is to measure the total download time and total system throughput.

Results

Let’s discuss the method and factors for the calculation of average download time of the topology. The average download time for all the nodes is calculated by using the following equation:

$$\Delta T = \frac{\sum_{n=1}^{N} \sum_{i=1}^{I} \Delta T_n(i)}{I \times N},$$  \hspace{1cm} (14)

where $I$ is the number of iterations, $N$ is the number of nodes and $\Delta T_n(i)$ is download time at iteration $'i'$ for node $'n'$.

One important factor is also the impact of the garbage collector in Android OS for the calculation of the download time. In some iterations of these test cases, it is noticed that the download time of the cloud leader is actually greater than one of the other cloud node. This is mainly because of the Android garbage collector. In Dalvik version of Android runtime, garbage collector stops the “world event”, which means that any managed code that is running will stop until the garbage collection operation is complete. For further explanation, consider that content download started at cloud leader and before it ends, garbage collection occurs therefore holding all the other events including content downloading. Eventually, it changes the actual download time it was supposed to take due to this garbage collection process. To mitigate this effect, the same content is downloaded for several iterations and then the mean value of download time is obtained for all iterations. To get an idea of how the download time varies between nodes for considered topologies, consider Figures 18 to 21. These figures shows that the download time for cloud leader is always smaller than all the other nodes. The time delay due to garbage collector has been diminished due to averaging the results.

![Figure 18. Download time of nodes in TR1 with cellular RSSI -60 dBm](image)

"Figure 18. Download time of nodes in TR1 with cellular RSSI -60 dBm."
Figure 19. Download time of nodes in TR2 with cellular RSSI -85 dBm.

Figure 20. Download time of nodes in TR3 with cellular RSSI -100 dBm.

\textsuperscript{5}TRx: Tree with x relay node
Let us consider the same test scenario to analyze performance of different topologies in terms of download time for different content sizes. In general, download time is directly proportional to the content size which means that by increasing the content size, download time increases accordingly. However, different distribution topologies differ in download time for downloading the same content from the same content server while keeping cellular RSSI constant. In order to understand this phenomenon, consider the total system end-to-end delay presented in the Figure 22.

The total end-to-end delay is divided into two parts, cellular and Mobile Cloud network delay shown in Figure 22. The cloud leader is the demarcation point between these two networks. Usually, cellular network delay is caused by the cellular network entities. On the other hand, Mobile Cloud delay is caused by the transmissions in the process of cloud formation and content distribution. These delays have different
dependencies for example, in case of cellular network, the delay comprised of many factors such as, radio channel propagation, local network traffic, load on the remote host, and format of the content requested [31]. Therefore, due to cellular network dependencies and having no access to it, it is hard to predict and calculate the delay on the cellular side. Furthermore, our focus is more on the Mobile Cloud side instead of cellular network. In case of Mobile Cloud delay, several factor can effect the delay such as load on router, number of nodes, distance between the nodes. Thus, in order to control the Mobile Cloud delay, we fixed the number of nodes, their positions and use of WiFi 5GHz instead of 2.4GHz to reduce interference.

It is worthy to mentioned again that this test case is carried out at fairly good cellular RSSI values between -55 dBm to -60 dBm. Since 3G uses adaptive modulation and coding in its physical layer [32] [33] [34], the total transmission time is lower due to higher data rates and better modulation scheme [35]. One the other hand, the Mobile Cloud has different distribution topologies and each of them differs in total download time for successful content delivery. This difference is due to the nodes arrangement in each topology and number of transmissions.

Nodes arrangement in each topology has already been discussed in previous sections. In order to understand the number of transmissions, let us have a very brief look on the basic operation of 802.11 MAC protocol [36] [37] which is called Distributed Coordination Function (DCF). DCF is based on Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). It works in a manner that whenever a node is ready for the transmission of a data frame, it first senses the availability of the channel or medium. If channel is idle and in order to avoid the collision, node waits for a time period called distributed inter-frame space (DIFS) followed by random backoff time interval. This random backoff interval is slotted and it is decremented when the slot is sensed idle. Node will continue to transmit the data frame after backoff reaches to zero. If the channel is found to be busy, backoff timer halts or freezes until the channel state become idle again [38].

From the above discussion, it can be concluded that the probability of collision can increase by increasing number of nodes as also investigated in [39]. In other words, the more nodes try to access the channel at the same time, the higher are the chances of collisions. Now, if we consider distribution topologies, it can be observed that some nodes in TR2 and TR3 are transmitting at the same time which is causing packet flooding on the network. This flooding can introduce extra transmission delay in the total download time of the content. Nodes in star and TR1 follows the sequential pattern and therefore the network is not flooded.

Figures 23 and 24 show the download time taken by the topologies to download different sizes of content from the internet keeping the 3G RSSI value fixed at cloud leader. It can be observed from the figures that that for the given scenario, Mobile Cloud delay is dominating the cellular network delay and therefore we can see that in topology TR1 has the shortest download time as compared to all the other topologies while TR2 and TR3 have approximately the same download time. The star topology has the comparatively the same download time as TR2 and TR3, however, CL is only involved in distributing the content to the other nodes which can reduce energy efficiency and can cause performance bottleneck.
Figure 23. Download time vs file size.

Figure 25. Throughput vs file size.
For system throughput, having knowledge of the content size and the total download time consumed, the results are calculated and presented in Figures 25 and 26.

4.3.3. Effect of RSSI on Mobile Cloud

Description

Let us consider the test case to analyze the effect of RSSI on the download time and throughput. Intuitively, the RSSI is inversely proportional to the distance from the cellular BS, hence the effect of distance from cellular base station (BS) to cloud leader is also taken into account for the calculation of download time. In this test scenario,
the environment is same as in the previous test case. In addition, we consider different values of RSSI. The measurements are taken at three different locations. There are many variables that affect RSSI value. To normalize this effect, the mean value of RSSI at the beginning and the end of the download is taken. Furthermore, it is assumed that the content is not available locally and the content size is constant for all the iterations.

**Input and number of iterations**

Through UI of the application, the URL of the requested content is input to the application. The file size considered in this test case is 5MB. The number of iterations is 8-10 at each locations for this specific test case.

**Results**

Let us analyse the effect of varying RSSI on the download time for each topology in Mobile Cloud. In Figure 27, it can be observed that download time increases as cellular RSSI value decreases. It is also apparent from the figure that, at lower RSSI values, cellular network delay becomes dominant instead of Mobile Cloud network delay due to which we can observe and differentiate the actual performance of different distribution topologies. This can also be related with the proposed theoretical model in section 3.5.6.

![Figure 27. RSSI vs average download time.](image)

Figure 27 shows system throughput which is the reciprocal of the download time. We can observe the decreasing trend in system throughput as the cellular RSSI is decreasing. One important thing to mention, that these results show the approximate behaviour of the performance parameter considered. This is due to many factors such as uncontrollable environmental variables of the cellular network.
4.4. Discussion

In this section, we discuss the several key aspects of the Mobile Cloud that are critical regarding the implementation work and need to be addressed.

4.4.1. Sophisticated cooperation strategies

One of the main argument is the selection of cooperation strategies. As we have discussed earlier that four cooperation strategies have been demonstrated in this implementation work, namely, star, TR1, TR2 and TR3. There are several other cooperation strategies that are more sophisticated and can be incorporated in Mobile Cloud architecture as some of them are already presented in [22]. However, there are certain limitation posed by the Android OS architecture due to which realization of more sophisticated schemes is difficult to achieve within given scope of work. For instance, User Datagram Protocol (UDP) for packet broadcasting seems to be the optimal solution for a scenario given that one or more nodes are interested in the same content. In this regard, the reason for not selecting UDP is its unreliable transmission and high packet loss rate. UDP uses a connectionless transmission model without handshaking dialogues and delivery acknowledgments, thus introducing unreliability and high packet loss in the transmission model [40]. Contrary to this, Transmission Control Protocol (TCP) for unicast communication ensures the reliable transmission by using three way handshaking and acknowledgement mechanism. Moreover, at the physical layer of Android, the OS tends to discard the whole broadcast packet if there is a bit error in the data frame, thus causing higher packet loss rate. Similarly, network coding also offers promising performance gain, however requires strict synchronization which is difficult to achieve in Android operating system.
4.4.2. **Real-time content distribution**

Real-time content is referred to the data that is presented as it is acquired. Usually, real-time content is not meant to be stored, instead, it is passed along to the end user immediately as it is gathered [41]. The most common examples of real-time data is video streaming and GPS (Global Positioning System) tracker. In context of Mobile Cloud, in which several nodes are interested in the same content, the most convenient approach is to broadcast the real-time content to all the nodes as cable TV operates. As stated in the previous discussion, broadcasting is not the reliable approach considering Android as the operating system of the device and thus can cause frame loss, jitter, synchronization error, etc. To mitigate these effects, a sophisticated yet reliable broadcast scheme and network coding are the viable solutions. However, this again poses new challenges to achieve reasonable performance in Android. Moreover, the current implemented Mobile Cloud architecture achieves the real-time content distribution however the performance is not up-to the mark and therefore not included in this thesis. Nevertheless, this task has been enlisted in the future works.

4.4.3. **Overall system performance**

We can now analyze the overall performance of Mobile Cloud, its shortcomings and compare it with the results obtained by the legacy approach presented in section 4.3.1. It can be observed from the Figure 29 that the download time increases as channel condition for cellular link gets worsen. At good RSSI values, performance of Mobile Cloud is as good as that of the single node. It is important to note that Mobile Cloud is serving seven nodes in approximately the same time as one node being served using legacy approach. As per our test results, time taken by a single node to download the content is 3.8s. One the other hand, Mobile Cloud takes 4.03s to download the content using TR1 method which gives us the difference of 203ms. However this can be improved further with more sophisticated data distribution techniques and can be made almost equal if we use broadcasting for distributing the content. Now if we consider the low RSSI values, the performance of Mobile Cloud degrades drastically. The main reason behind this is that all the nodes in the Mobile Cloud are connected to each other and therefore, if the performance of CL deteriorate due to cellular network, the performance of the other nodes in the Mobile Cloud will also depreciate.

The performance of Mobile Cloud is more notable if we consider system throughput as shown in Figure 30. We can observe that, Mobile Cloud outperforms at good cellular RSSI value. While the performance of Mobile Cloud versus the single node is approximately the same at low cellular RSSI values.

One of the main advantage of the Mobile Cloud is to save the bandwidth. For instance, considering legacy approach, if a single node acquires $B$ bandwidth to download the content from the cellular network, then $n$ nodes require $n \times B$ bandwidth. Contrary to this, Mobile Cloud requires only one node to connect to cellular link, therefore saving $((n - 1) \times B)$ bandwidth. Certainly, there is a trade off between bandwidth efficiency and the performance in terms of download time. Moreover, Mobile Cloud uses unicast communication between all the nodes for content distribution, one can assume it as individual data stream from cloud leader to each of the cloud node.
However currently our Mobile Cloud application does not fully support to handle all the data streams in parallel, instead it performs it in a serial manner.

4.4.4. Practical demonstration of Mobile Cloud

The practical demonstration of the implemented is available on the internet and can be seen in the video using the following link:
https://www.youtube.com/watch?v=P1vfL6AcEok

4.4.5. Future work

The core concept behind Mobile Cloud architecture is a cooperative and cognitive system which can analyze its environment parameters and act accordingly in an efficient way. In this thesis work, we implemented this concept using smartphones and considering the scenario where a group of smartphones in a close proximity are interested in the same content which we have already discussed in our previous sections. However the current implementation can be extended further to the next level to produce more concrete outcomes. Also there can be numerous other possibilities where our current concept of Mobile Cloud can be implemented and proven beneficial. Some of these proposed ideas are discussed below.

• WiFi Direct

WiFi Direct is an emerging P2P communication protocol belonging to the family of IEEE 802.11 standards. It enables multiple devices to communicate with each other without requiring any standalone or dedicated access point. WiFi Direct and Mobile Cloud and their compatibility issues have been discussed briefly in Appendix 3. WiFi
Direct can be incorporated in Mobile Cloud architecture if the issues addressed in appendix are resolved. One way of resolving these issues is by modifying the WiFi driver and the source code of the Android OS. Another way we can accomplish this is by implementing Mobile Cloud on an open source board in which we can customize application and drivers as per our need. By doing so, we can remove access point from our current architecture and thus can achieve better system performance.

- **Controlled environment**

As discussed in the previous sections, we have tested Mobile Cloud application on a live cellular network with uncontrolled environment variables like radio channel propagation delays, network latency, local network traffic, load on remote server etc. which produced some undesirable effects on system performance. Better performance results can be expected by optimizing and controlling these variables. Also this can give us better understanding of the system where it is lacking and where it should be optimized.

- **Multiple cloud leader**

In this thesis work, we have considered only one Mobile Cloud with one cloud leader. This can be further extended to multiple cloud in a cloud with multiple cloud leaders. With multiple cloud, we can increase the coverage areas whereas multiple cloud leader can enhance the system throughput as well as better system performance.
5. SUMMARY

Information technology is continuously changing almost every aspect of life and posing new challenges for telecommunication industry. While substantial amount of work has been done in this context, particularly on cooperative and cognitive networks, the very approach has certain shortcomings, which we broadly classify as three characteristic challenges.

1. Enhance data throughput
2. Adequate environment adaptability
3. Efficient resource utilization.

The first and foremost challenge is to enhance the overall throughput of the system. Sophisticated modulation techniques have increased the data throughput, nevertheless, it can further be improved by utilizing multiple frequency bands simultaneously as well as using an appropriate frequency band for its relevant type of communication. The second major challenge is to improve the adaptability of the system to the environment. Lastly, the efficient distribution of the content by utilizing the available resources in a cooperative manner. The content intended in this thesis is considered to share the common interest of a group of users or in a small social network, therefore assumed to be high in demand or accessed frequently.

As already mentioned, the systems presented in the past were not necessarily developed with an intent to address all of the three challenges. Some of them were primarily focused on cellular traffic offloading while other focused on D2D communication, disregarding one or more of these challenges. Therefore the goal of this thesis was to develop a system model that addresses all the three challenges. Targeting this goal, we presented Mobile Cloud, a novel yet simplistic approach towards cooperative and cognitive network that has the following properties:

1. Self-organizing D2D network
2. A system model that exploits multi-radio communication
3. Sophisticated content distribution techniques.

Primarily, Mobile Cloud establishes a local self-organizing D2D network among the nearby devices and assign one of the device as cloud leader, based on a given criterion. Secondly, Mobile Cloud utilizes cellular radio interface together with WiFi to achieve the bandwidth efficiency. Unlike the legacy approach in which all the devices use individual cellular links to download the content, Mobile Cloud allows to utilize the cellular link of only the cloud leader and distribute the content locally using WiFi. Last but not the least, Mobile Cloud offers various cooperation strategies for the distribution of the content locally. These cooperation strategies aim for the efficient utilization of the resources available with the devices. In this way, Mobile Cloud caters all the three challenges defined above.
Mobile Cloud offers benefits for both the end consumer and the network operator. For consumer with limited resources, this system model provides better energy efficiency and improved system throughput. Moreover, it provides promising gain considering the popular media content. It also provides a small social network for a group of users with the common interest.

From the operator prospective, Mobile Cloud proposes several advantages. One of the major contribution of the Mobile Cloud is to save the bandwidth. Mobile Cloud also offers network operators to offload the data traffic for the media contents that is high in demand or the content which is more likely to be accessed repeatedly within a group of people in a confined proximity e.g., airport or a class room. Moreover, Mobile Cloud also offer to expand the network coverage to the areas where cellular link quality is not up-to the mark or suffers from excessive channel degradation.
6. REFERENCES


[34] Motorola, Nokia, Philips, TI & Altera, Joint 1xtreme proposal for 1xev-dv (2000).


[36] IEEE, Wireless lan medium access control(mac) and physical layer (phy) specifications (2012).


[42] IEEE, Ieee 802.11z-2010 - wireless lan medium access control (mac) and physical layer (phy) specifications amendment 7: Extensions to direct-link setup (dls) (2010).


7. APPENDICES

Appendix 1. How to run Mobile Cloud test-bed

Below is the procedure how to properly run the test-bed for Mobile Cloud. Before launching the test-bed application on the smartphones, following things are needed to be checked:

1. All the smartphones should be installed with the Mobile Cloud application.

2. Smartphones should be capable of communication both on WiFi and 3G/4G cellular network at the same time. Specifically speaking of the test-bed implemented in this project work, we select one cloud node and intend to make this node as the cloud leader therefore it must have the custom ROM (Read Only Memory) specifically built for this project. Guide for this is given in the section below.

3. WiFi and cellular data network should work simultaneously. Before running the application, make sure that all the devices are connected to the same network.

4. Now run the Mobile Cloud application on all the devices and wait for the neighbor discovery and cloud formation.

5. Once the cloud formation is done, any file can be downloaded from the internet. There should be a direct download link to the file and Mobile Cloud application only supports link with “http”, https will not work. Following example shows the link supported by the application.

   http://avideos.5min.com/749/5182749/518274873_2.mp4

Appendix 2. Enable multi-radio communication in Android

In order to enable radios for WiFi and cellular network to work simultaneously, the Android phone must have the root access for the Android operating system (or the phone is called rooted), contains the custom ROM specifically built for this project, phone must contain BusyBox installed in it and the phone must have USB debugger enabled. Busybox is an Android application that allows you or programs to perform actions on your phone using Linux commands. For rooting the Android phone and enabling the USB (Universal Serial Bus) debugger, extensive help is available on the internet, therefore, it is not covered in this document. However the custom ROM built for this project is not available on the internet.

We enable this feature by using ADB (Android Debug Bridge) terminal from personal computer (running Windows 7) which is described as below.

1. Connect the smartphone with the computer via USB cable. Make sure the USB drivers are already installed.

2. Connect to the WiFi access point (AP) and make sure the default gateway in the Wireless Network profile is set to “0.0.0.0”. It is recommended to use IPs of
primary and secondary DNS (Domain Name Server) as “8.8.8.8” and “8.8.4.4” respectively.

3. Also make sure that cellular data is enabled in the phone settings.

4. Now on your computer, run Windows Command Prompt with administrative rights.

5. Type and run “adb shell” to use terminal of the Android OS.

6. Login as a root user or administrator by running “su” command.

7. Check if there is any “default route” in the routing table by running “busybox route” command. This command will display the kernel routing table and if there is any default route, it should be deleted by “busybox route del default” command. Delete all the default route entries if there are multiple.

8. Now add a new entry for default route by using “busybox route add default dev rmnet_usb0”. In this command, rmnet_usb0 is the cellular network data interface used in Samsung smartphones and it can be different on different smartphones.

9. By now, the phone should be able to test the reachability of any remote host. This can be done by using for example “ping www.google.com”. The phone should now receive the ICMP (Internet Control Message Protocol) response with the round trip time and any packet loss.

Appendix 3. WiFi-Direct and Mobile Cloud

There are several P2P or D2D physical layer protocol in use these days and they are appropriate for different environment and serves different areas of expertise. As per scope of this project, we are only considering IEEE 802.11 (WiFi) as a standard communication for P2P or short range communication. Since P2P communication is possible in the original IEEE 802.11 standard by using AD-HOC operation mode. Due to its limitations like lack of power efficiency, quality of service (QoS) and security, this has never been widely deployed in the market. Another relevant protocol in the WiFi P2P communications is 802.11z, also known as Tunneled Direct Link Setup (TDLS) [42], which enables direct device to device communication but requires stations to be associated with the same AP [43]. WiFi direct is also an emerging 802.11 standard protocol enabling P2P communication without requiring any dedicated access point. The common thing among all these protocols is that the role of the device is dynamic in the sense that it can act both as a client as well as an access point.

Now let us compare WiFi direct with Mobile Cloud. The reason for comparing only WiFi direct is that it is available on most of the smartphones operating systems i.e., Android, Windows and iOS whereas other available P2P communication protocol are either not supported by smartphones or have some other limitations like range, connection speed, security etc. Despite of support of WiFi direct in the smartphone we are using, there are certain limitation due to which we cannot incorporate WiFi direct in Mobile Cloud application. These limitations are discussed phase wise like in Mobile
Cloud application. Please note that we are using the same terminologies as mentioned in the WiFi direct standard [44].

Let us consider a scenario where there are total four nodes naming node A, B, C and D. Let us also assume that node A has the highest metric value (which is also known as Group Owner Intent in WiFi Direct standard) followed by node B, C and D i.e., node A has highest tendency to become cloud leader and node B has metric value greater than node C and D. Node D has the lowest metric value in all the nodes. For simplification, we also assume that all the nodes support WiFi Direct and the network is open network without any security key or encryption.

1. In group (cloud) formation phase, WiFi Direct can be initialized on all the nodes by using its API (Application Programmable Interface) [45]. By initializing this, each node will discover the neighboring nodes to establish a P2P Group. Here we assume that application is started on all the nodes at the same time. But before a group formation is completed, group linkage has to be accepted by means of human interaction in the form of acceptance dialogue box. Therefore almost all the devices need to be linked to the group by using human action. The appearance of this acceptance dialogue box is determined by the fact who started the connection process. This group profile will be saved as “Known Groups” or “Stored Groups” in the device itself for future use.

2. One concern is that which node should start the group formation. Since WiFi Direct shares very limited information in the discovery message therefore it is not possible to advertise the metric value based on which we can decide which node to initiate the group formation. This group can be formed without any ambiguity or reorganizing the group owner in case if we randomly choose node A to initiate the formation. Things can get more complicated and very difficult to handle if we randomly choose group owner other than node A because multiple group owner is currently not supported by the Android API for WiFi Direct. Moreover a single device cannot join two groups at the same time [46].

3. After the P2P Group formation, all the nodes in the group negotiate with each other to decide the P2P Group Owner (P2P GO) which has similar role as of cloud leader in Mobile Cloud architecture. This P2P GO is selected based on GO Intent parameter which is defined as an integer in the API and can range from 0 to 15. This gives us less freedom to choose group owner as compared to Mobile Cloud in which metric value ranges from 0 to 100.

4. Let us assume that group formation is done and the P2P GO is also selected, the content sharing is same as in Mobile Cloud. Both in-cloud and out-cloud content sharing is possible and can be done without any ambiguity.

5. The foremost factor that limits the use of WiFi Direct is the group management. A new group has to be created again if the group owner leaves or changes. In other words, all the nodes have to disconnect from the previous group and join the new group. It can be thought of as “hard linkage” where connection has to be broken before making new one while in Mobile Cloud, the cloud leader transition does not require link breakage and therefore can be considered as “soft linkage”. Moreover, if the group profile does not match from any profile in the
known groups, the user’s interaction will be required again to accept the group linkage.

6. Another limiting factor for the group management is adding a new node in the existing group due to limited information in the discovery message. In API, there are some methods to find the group owner but in general, a connection needs to be established in order to know who the group owner is. There is some workaround to accomplish this. If a new node say node E wants to join the group whose group owner is node A. Node A needs to add local service and advertise it. On the other side, service discovery can be accomplished by methods available in API and hence node E can be connected to the existing group. This can become more complicated and difficult to manage if node E has metric value or GO intent higher than node A.