جـامـعـة الـشـرق الأوسـط
MIDDLE EAST UNIVERSITY
Amman – Jordan عمان - الأردن

Bandwidth Saving Method Based Sharing Data Downloaded From Cloud Between Smartphones In Wifi Network

طريقة لتوفير عرض النطاق الترددي على اساس مشاركة البيانات المحمله من السحابة بين الهواتف الذكية في الشبكة اللاسلكية

By

ALI ABBAS MOHAMMED ALSHAMMARY

401210102

Supervisor

Prof. MOHAMMED ALHAMID

Submitted in Partial Fulfillment of the requirements of the

Master Degree in Computer Science

The Middle East University for the Graduate Studies

Amman-2014

II

إقرار تفويض

إني علي عباس محمد الشمري أفوض جامعة الشرق الأوسط بتزويد نسخ من رسالتي للمكتبات أو المؤسسات أو الهيئات أو الأفراد عند طلبها.

التوقيع:

التاريخ: 23 / 7/ 2014

Authorization statement

I ALI ABBAS MOHAMMED ALSHAMMARY, Authorize the Middle East University to supply a copy of my Thesis to libraries, establishments or individuals upon their request.

Signature:

Date: 2014-07-23

قرار للجنة المناقشة

نوقشت هذه الرسالة وعنوانها (طريقة لتوفير عرض النطاق الترددي على اساس مشاركة البيانات المحمله من السحابة بين الهواتف الذكية في الشبكة اللاسلكية) وأجيزت بتاريخ 2014/7/23.

لجنة المناقشة **التوقيع**

1- د.هبة حسن ناصر الدين رئيس اللجنة والعضو الداخلي (جامعة الشرق الاوسط) ....

2- أ.د. محمد دباس الحميد المشرف (جامعة الشرق الاوسط) ....

3- أ.د. عبد السلام وليد عربيات عضو اللجنة الخارجي (جامعة البلقاء التطبيقية) ....

COMMITTEE DECISION

This is certifying that the thesis entitled " Bandwidth Saving Method Based Sharing Data Downloaded From Cloud Between Smartphones In Wifi Network" Was successfully defended and approved on July 23th 2014.

Examination Committee Members	**Signature**
Dr. Hebah, Hassan, Nasereddin Associate Professor, Department of Computer Information Systems (Middle East University)	....
Prof. MOHAMMED ALHAMID Professor, Department of Computer Information Systems (Middle East University)	....
Dr. Abdulsalam W. Alarabeyyat Associate Professor, Department Software Engineering (Al-Balqa' Applied University,)	....

Table of Contents

List of Figures

List of Tables

List of Abbreviations

Abbreviation	Meaning
BW	Bandwidth
DTMF	Dual-tone multi-frequency signaling
IP	Internet Protocol
IT	Information Technology
LAN	Local Area Network
MAUI	Making smartphones last longer with code offload
MCC	Mobile Cloud Computing
MB	Mega Byte
P2P	Peer Two Peer
PKI	public key infrastructure
MDP	Markov Decision Pross Algorithm
RFID	Radio-frequency identification
RACE	Resource A ware Collaborative Execution
SQLite	ACID-complint and implements most of SQL Standard
UDP	User Datagram Protocol
VM	virtual machine
WLAN	wireless local area network
WIFI	Wireless Fidelity

Abstract

Cloud computing has become an important task for mobility, and has become a feature in every aspect of our daily lives with the dynamic character of the side. However, like any new technology; mobile cloud computing has some problem . And one of the most influential challenges is the bandwidth consumption of the ability of mobile data in order to promote the application of communication and data transfer between the smartphones and the cloud.

In this thesis, the researcher suggested the idea to reduce the bandwidth consumption based on the cooperative model for mobile phone applications to enhance the overall communication between mobile devices and the cloud in WLAN networks and LAN. Also, the researcher tried to answer two important questions to demonstrate the importance of the idea and applied: Is the result of the operation of this model show a significant reduction of bandwidth consumed? and what are the alternatves perform data transfer between mobile devices?

Where the application was designed for the universe is able to post the data from the cloud between mobile phones in the same network using a small database application architecture common in mobile devices.

The researcher used which approach to analyze the result according to the comparison of results obtained using the proposed application in order to prove the importance and validity of this idea .

The researcher showed a several of the results : There is an important value for the consumption of bandwidth that can diminishes the researcher using the idea that floated over the LAN networks and reduce download time compared to download files directly from the cloud. Also, the value of the bandwidth consumed becomes less whenever desired file is larger. Bluetooth, LAN and ways of wireless communication can be used effectively, rather than the means of communication bandwidth consumer. Where they can use this idea in the company and the institute that you need to work on to download files from the cloud, as well as the use of this idea in private networks, which are located in a single building or a specific place.

الخلاصة

أصبحت الحوسبة السحابية المتنقلة مهمة لغرض التنقل .كذلك اصبحت ميزة في كل جانب من جوانب حياتنا اليومية ذات الطابع العملي . مثل اي تكنولوجيا جديدة .الحوسبة السحابية المتنقلة لديها بعض المشاكل .واحد من التحديات الاكثر تاثيرا هو استهلاك عرض النطاق الترددي من قدرة البيانات المتنقلة .من اجل تعزيز تطبيق الاتصالات ونقل البيانات بين اجهزة الهواتف الذكية والسحابة.

في هذه الرسالة اقترح الباحث فكرة لتقليل استهلاك عرض النطاق الترددي على اساس نموذج تعاوني للتطبيقات الهواتف الذكية لتحسين التواصل بين الاجهزة النقالة والسحابة في الشبكات المحلية اللاسلكية بشكل عام

حاول الباحث الاجابة على سؤالين هاميين لبيان اهمية هذه الفكره وتطبيقها: هل نتيجة التنفيذ هذا النموذج نحصل على انخفاض كبير في استهلاك عرض النطاق الترددي؟وما هي البدائل لنقل البيانات بين الهواتف الذكية؟

حيث تم تصميم نموذج تطبيق بشكل عام قادر على مشاركة البيانات المحملة من السحابة في ما بين الهواتف الذكية في نفس الشبكة باستخدام معمارية عامة تطبيقية ذات قاعدة بيانات صغيرة في اجهزة الهواتف الذكية.كذلك استخدم الباحث منهجاً لتحليل النتيجة من اجل اثبات اهمية وصلاحية هذه الفكرة.

حيث اظهر الباحث نتائج عديدة منها: هناك قيمة هامة للاستهلاك عرض النطاق الترددي الذي استطاع الباحث ان يحد منها باستخدام الفكرة المطروحة عبرشبكات المحلة اللاسلكية .وتقليل قت التحميل مقارنة بالتحميل الملفات مباشرة من السحابة.ايضا ان القيمة المستهلكة لعرض النطاق الترددي لتصبح اقل كلما كان الملف المراد تحميله اكبر.بلوتوث وشبكات المحلية اللاسلكية وسبل الاتصالات يمكن ان تكون فعالة عند استخدامها في هذا المجال بدلاً من وسائل الاتصالات المستهلكة لعرض النطاق الترددي. كذلك يمكن استخدام هذه الفكرة في الشركات والمؤسسات التي تحتاج في عملها على تحميل الملفات من السحابة فضلاً عن استخدام هذه الفكرة في الشبكات الخاصة التي تقع في مبنى واحد او مكان واحد محدد.

Chapter One

Introduction

1. Introduction:

Nowadays, Smart phones have an ability of supplying huge level of applications, where many of which required high computational power. This causes some issues because smart phones device are limited resource devices with specific computation power, memory, storage, and energy. While, the cloud computing technique supply unspecific dynamic resources for computation, storage, and service provision. Therefore, many workers expanding the cloud computing advantages to mobile devices to decrease as much as possible the smart phones limitations (Dinh and Wang 2013).

(Hoang T. Dinh et al 2013) Mobile Cloud Computing (MCC) is a combination of cloud computing and mobile networks to reach the advantage for mobile users network operators, where the processing of data happen outside of the mobile device. Since mobile devices are helped with feeling abilities, a cloud made up of mobile devices will be able to supply the users with situation and location perceptible services as well, driving to a more personalized testing. Mobile Cloud applications have the strength of a server-based computing infrastructure attainable via a mobile's application interface. It does not only allow non Smartphone owner to access the same mobile applications, but it allows the applications themselves to become stronger. Cloud Computing is a kind of computing indicate saving/sharing programs and data over the network such as internet rather on the PC or driver. The main characteristics of the cloud computing are: the availability, minimizing the cost, supplies a broad network access either by smart phones or by PC, scalability and so on. One of the most powerful characteristics of cloud computing is its flexibility of sharing the storage resources whereas the bandwidth is the most critical part in the cloud computing infrastructure.

It has become an important and efficient topic of the industrial environment since 2007 (Han Qi and Abdullah Gani,2012). Mainly, cloud computing is defined as a big level of services which are supplied by an Internet-based cluster system.

1.2 Advantages of Cloud Computing

If utilized completely and to the reach necessary, working with data in the cloud can immensely advantage all kinds of businesses. Explained below are some of the advantages of this technology (Cloudtweaks , 2014) :

1.2.1 **Cost Efficient**

Cloud computing is completely the most cost efficient manner to use, handle and improve. Traditional desktop software costs organizations a lot in term of finance.

1.2.2 **Almost Unlimited Storage**

Saving data in the cloud awards you almost unlimited storage capacity. Hence, you no more want to worry about running out of storage space or increasing your current storage space availability.

1.2.3 Easy Access to Information

Once you record yourself in the cloud, you can access any data from anywhere, where there is an Internet connection exist. This convenient properities lets you move beyond time zone and geographic location issues.

1.3 Disadvantages of Cloud Computing

Despite its many advantages, as defined above, cloud computing also has its disadvantages. Businesses, especially smaller ones, need to be aware of these disadvantages before going in for this technology. The Risks Involved in Cloud Computing (Almrot &Sebastian Andersson2013).

1..1 Technical Issues

Though it is true that information and data on the cloud can be accessed anytime and from anywhere at all, there are times when this system can have some serious of weakness. You should be aware of the fact that this technology is always prone to outages and other technical issues.

1.3.2 Security in the Cloud

The other major issue while in the cloud is that of security issues. Before adopting this technology, you should know that you will be surrendering all your company's sensitive information to a third-party cloud service provider.

Like everything else, cloud computing too has its pros and cons. While the technology can prove to be a great asset to your company, it could also cause harm if not understood and used properly. So, in this application that combine the advantage of cloud computing to handle some issues in mobile devices.

The main problem and issue of mobile cloud computing born from the mobile devices and wireless networks characteristics, as well as their own limitation. Some of these issues makes application modeling, programming and development on mobile devices more complex than on the fixed and specific cloud devices. It is practical to envision the outlook mobile clouds as crossbred, where the users themselves would participate as cloud resources, but with the ability to communicate the remote servers in situations of high connectivity and other cases like access fees, obtainable battery, and response time. This would need the mobile cloud structure to be proactive and self-adaptive. To abstract, the causes for sharing/offloading work from a mobile device would be: fixed calculation capability, fixed battery power, fixed connectivity, chance to collect more sensing data, access to various content/data groups, and to make use of idling processing power. In mobile cloud computing area, the issues of mobile devices, quality of wireless communication, kinds of application, and supply from cloud computing to mobile are all mainly critical factors that impact estimating from cloud computing (Dinh and Wang 2013).

Mobile cloud applications carry the mobile data storage from the mobile phones to the cloud, that which increase the user ability to utilize our applications and data (Dinh and Wang 2013).

However cloud suppliers cannot interest from any technology that minimize the user bandwidth bills, so is not possible to develop in one.

In it's organizations they defines the important relations among the performance of applications and bandwidth either by increasing bandwidth or implementing technology that prefer utilizing of bandwidth that is actually obtainable (Fernando 2013).

1.4 Problem Statement

Nowadays, the mobile cloud computing application become very popular and many companies are using it. some cloud-based applications. But the main issue faced many companies that it is hard to control its remote sites without spending much of money to hire new IT personnel for each office.

the main problem is the bandwidth limitation and the highest bandwidth consumption when the user downloads many files and decrease the download speed, this will decrease the performance at whole.

When we study this proposed idea we noted that the bandwidth consumption is the main problem in mobile cloud computing, here we explained the problem when many users send a many request to the cloud, this will cause a high consuming of the bandwidth which decrease the network performance at whole. So, we proposed a solution to prevent the redundancy request of a such file and this will preserve the network bandwidth.

This will cause a redundancy problem, to obviate this problem, this work modeled an infrastructure that enable file being participate among mobile devices on the same network. by using multi little shared databases to prevent data frequency, which outcomes in many effective data. This manner can fix immoderate bandwidth consumption or poor performance problems.

In addition, the dynamic mobile wireless area with specific existence bandwidth is very difficult for mobile users to supply consistent and services provided by cloud computing systems. Many of the existence bandwidth capacities that supplied by telecommunication organization is to prevent abuse by a little number of user, thus leaving efficient bandwidth for everybody else. And often, the bandwidth issues are large enough that many users look to reach them. But as more users depending on the Internet for many of their television showing, these capacities may become more of limitations.

1.5 Objectives

In this thesis, to handle the bandwidth consumption in mobile cloud computing area, it is critical to minimize the bandwidth cost in this cloud area and handle high-speed load of data. In order to reach this goal, the following objective must be achieved.

- Reducing the frequency of data requests from the cloud
- Make ability to participate the received data to prevent second call back request to the cloud.
- Define a mobile application model that can render the goal of file participating among users on the specific network.
- Define such algorithm that will control the request of the cloud computing and preside them to the suitable download resources.
- Download huge files from the cloud at a high speed, with small bandwidth and storage consuming.

1.6 Motivation

Bandwidth is one of critical problems in mobile cloud computing since the radio resource for wireless networks is more rare when contrasted with the traditional wired networks. Many researchers proposed different solutions to participate the specific bandwidth between mobile users that placed in the specific area. There are many applications specified to minimize the bandwidth consuming in cloud computing. So, in this work focus on:

- Specifying the users requests of data or resend the requests in different directions to a local storage resides on local network instead of cloud request. However, specifying users to particular data requests can minimize the bandwidth, it is still based of rejected access to data; this will not stop high traffic requests to the cloud.
- Constructing native storage database that will avail network users rather than cloud may decrease traffic load and bandwidth consuming, but in big organization this will generate big load on the local database system, and will not serve the primary goal of a cloud computing structure which is decreasing resources, hardware, software, and
- This work migrated preceding mobile cloud computing applications and improved them to reach perfect stage of bandwidth consumption and equitable distribution of data. However single shared storage that establishes on the local network, this may improve mobile application building to contract little storage amplitude that will be shared between various network users. In addition, to decrease the requests on the cloud, this work modeled an algorithm that will polish and redirect users' requests to the most suitable download location, whether it is on the cloud, or on a close user mobile storage.

1.7 Contribution

As previously explained, mobile cloud computing has various problems especially in bandwidth consumption. Thus, in this proposed application, which was verified using Android application, a new contributed idea was proposed that helps in reducing the bandwidth consumption by decreasing the frequent request and by using the shared database from other users in the same network. However, this application differentiate the request path of the same file, by directing it to the cloud in the first time, then to the mobile that first downloaded it in the second time.

Moreover, to get more enhancement to the download speed, each time the file will be requested, it will be downloaded from the shared memory of all the previous phones that requested it previously, in an intelligent way, of dividing the file in small chunks and storing it in the shared memory. The use of the shared memory enhances the security of the files on the mobile, by preventing the other requester mobiles to use the same data memory of the hosting mobile.

1.8 Proposed System

In this thesis proposed solution depends on generating a tiny shared databases that establish on mobile devices structure, so in this proposed, each receiver mentions the importing traffic flow and attempts to correspond its chunks with a close data source before acquiring that data from the cloud immediately.

The shared database in each computer reserve 30% of memory in each device, this will effect negatively at the storage capacity of the device. But in this proposed application can override this problem by designing an option in application that helps us to delete the unnecessary files in each device's database. The deleted files may be the unnecessary or the not used files. This will happen by restricting 30% of the phones main memory and assign it to the newly proposed memory, which is the shared memory. This restriction will be done by the proposed system itself. It will automatically restrict this 30% to the shared memory.

Every user will be downloading little amount of bits from different mobile customers shared databases that establish on the specific WIFI network, and at the same time-sharing downloaded bits with other various users. Downloading mechanism is executing on the background, and just when the user of mobile is in-active process, this mechanism can be controlled by a such algorithm, which will defined which files each user would likely to request, and list the download sequence for these files before they were really requested by the customers, so the files will be preparation at request time.

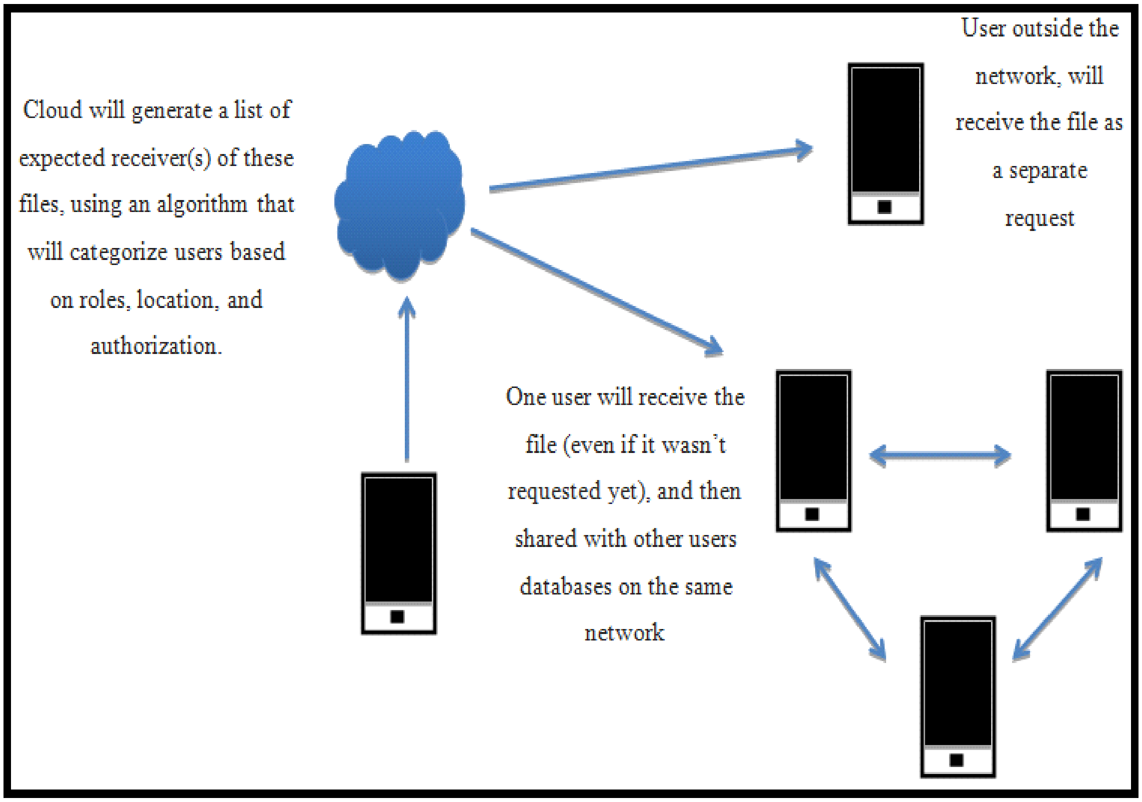


Figure 1-1: System Architecture sharing files between cloud users (Proposed System).

In the above figure, each mobile device in the same network may send a request to each other to obtain the specific response that need, so the received file will share with other user in the same network, but when the device locate outside the network, it will receive a file by a separate request to the cloud.

Figure 1-2 explains the enhanced proposed Mobile Application model that defines a local database file exist on each mobile device that locate in the specific area that has specific WIFI network, and this database will be shared between these nearby peers in the same WIFI.

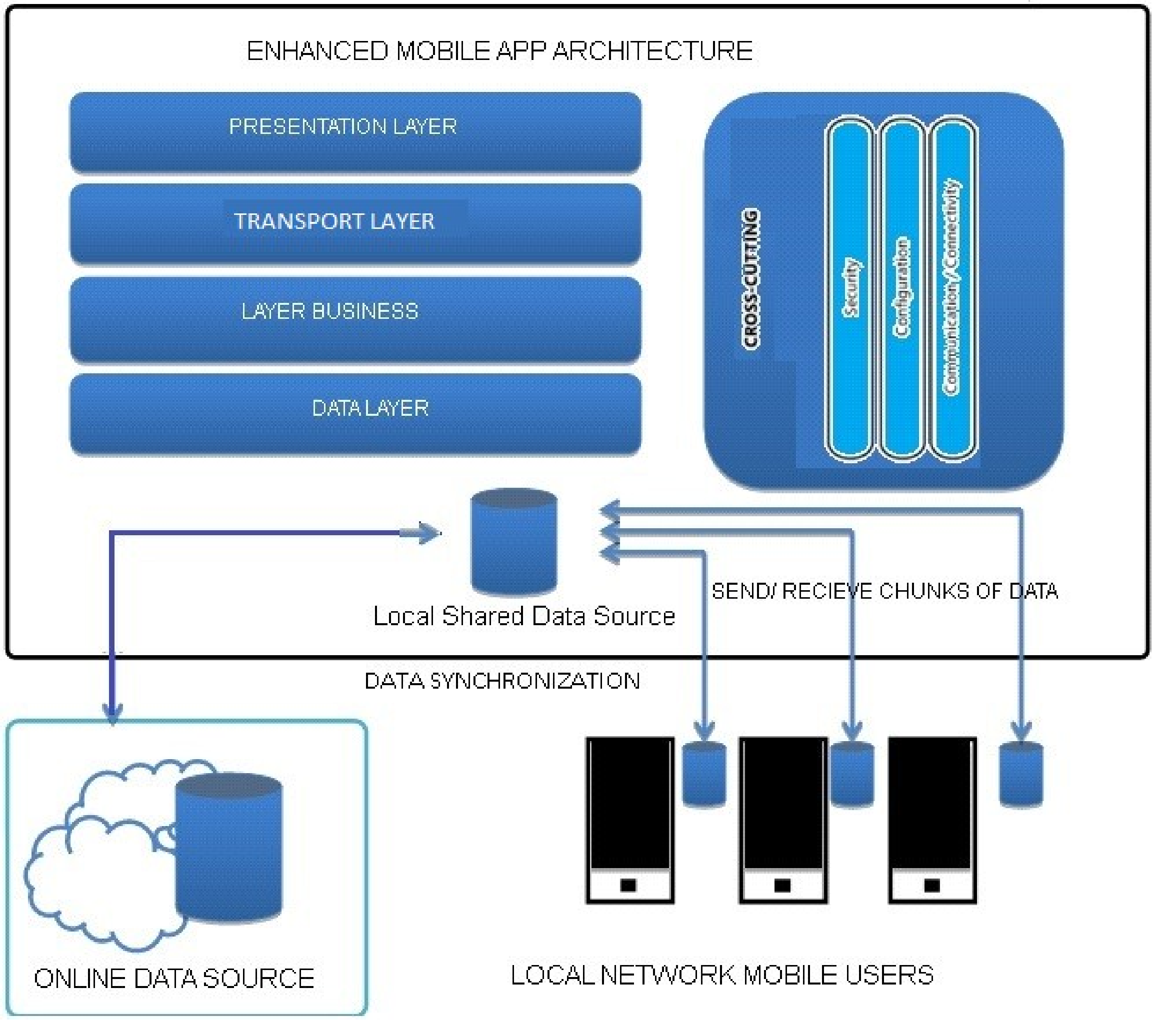


Figure 1-2: Enhanced Mobile App architecture (Proposed System)

1.9 Advantages of Proposed System

this proposed application provides a user permission to download huge files from the cloud, while decreasing the bandwidth consuming because each device has its own database file which may share with nearby devices.

This solution solves the issues of downloads a large files and conveniently solve the issue with seamless download process running in the background, so the user can use and share once it is ready.

1.10 Research Challenges

The research problems of this proposed model are as follow:

- This proposed mobile application architecture, is more efficient when there's groups of people located on the same network (eg. Same WIFI), it is not going to be successful if each cloud user using external dedicated network (different networks).
- In this proposed work, while every mobile device is going to be sending and receiving data on regular basis, this would decrease the battery life.
- This Suggested mobile application model needs enhanced mobile devices with enough storage, and high processing power to execute the communication channel smoothly.

1.11 The proposed Architecture

In this proposed work explains an application architecture for sharing data among users on a specific network, to minimize multiple redundant requests to the cloud. If the cloud file being requested by a user, the file will be shared between other users of the specific network once it chunks arrives, so that a second request of the same file will be downloaded from the first request user's mobile device, or from another device of the same network, that has already downloaded parts of the file.

this application architecture needed an algorithm to control these highly active requests, and to direct files accordingly to appropriate network users and will use a java script to achieve this and SQLite Database (for relational database) to create own database file for each user.

Example

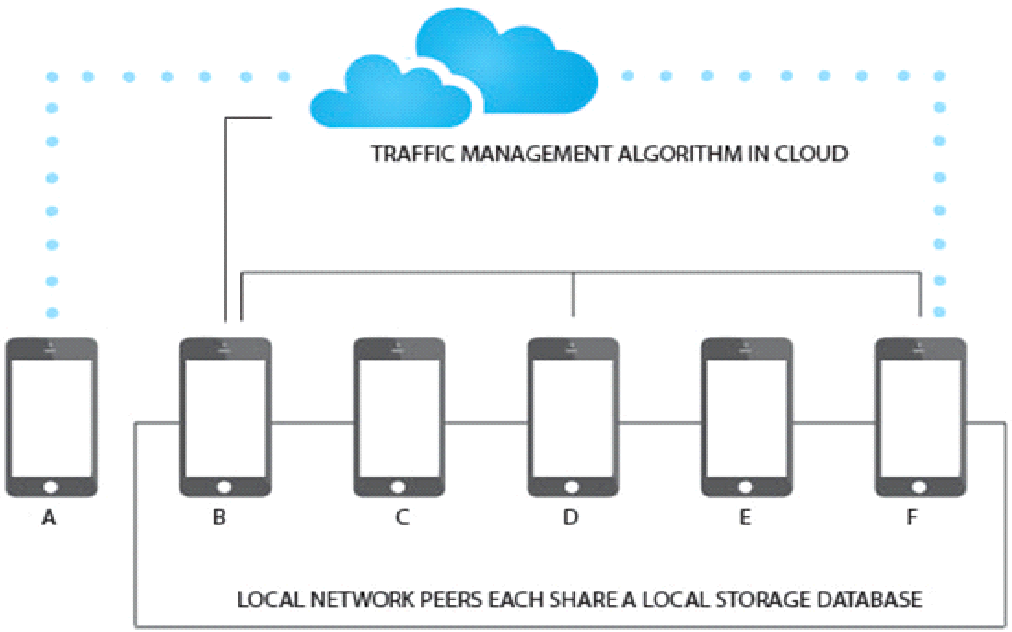


Figure 1-3:Sending/ Receiving Files from the cloud Process (Proposed System)

Data sharing Scenario, as shown in the above figure 1-3

- Mobile user (A) sends some data to the cloud.
- Cloud decides to send the data to mobile user (B), as an output of determination related on data authentication, mobile user location and other rules.
- Mobile user (B) decides to prioritize sharing the data with mobile user (D) and mobile user (F), based on information from the cloud.
- Mobile user (F) still able to communicate with the cloud, while receiving data from peers, since the application architecture split the shared database from the direct cloud communication database.

1.12 Definitions of some existed Terms:

- Mobile Cloud Computing (MCC) :it is a fresh application combining the mobile devices and cloud computing to generate a new infrastructure with different advantages (Dinh & Lee 2013)
- Bandwidth Consumption: is the maximum throughput of a logical or physical communication path in a digital communication system (Dinh & Lee 2013).
- Functional Requirement : this requirement specify what the product must do (Thomas Andrews 2006).
- Non- Functional Requirement: is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors (Thomas Andrews 2006).
- Broadcast: refers to a method of transferring a message to all recipients simultaneously. Broadcasting can be performed as a high level operation in a program (Sumon , Foez and Nayeema,2010)
- Unicast: is communication between a single sender and a single receiver over a network (Sumon , Foez and Nayeema,2010).
- Multicast: communication between a single sender and multiple receivers (Sumon , Foez and Nayeema,2010).
- Throughput: The ratio of the total data received by the end user and the connection time (Sumon , Foez and Nayeema,2010).

1.13. Thesis Outline:

This thesis consists of five chapters.

- Chapter 1 gaves introduction about the main idea of this application. We also explained the problem statement and a summary about the proposed application.
- Chapter 2 displays the literature review and related work of application, we talk about the different limitations that face the mobile cloud computing application and specifically we concern with bandwidth consumption issue.
- Chapter 3 talks about the requirements of proposed application. It explain the functional and non-functional requirements, the software requirement and the main button that needed in the main screen of application.
- Chapter 4 displays the design of proposed application, the application's screens and explain the main function of each one. It defined the result analysis of this project.
- Finally, Chapter 5 displays the conclusion of this thesis and the future work.

Chapter Two

Literature Survey

2.1 Mobile cloud computing issues:

Many researchers mention that Mobile Cloud Computing (MCC) faces many problems such as operational level issues, user level issues, service and performance level issues. This work concentrate here on the performance issue which faces bandwidth consumption problem. There were many methods in the past which described bandwidth consumption in mobile cloud computing, either by limiting the user request or redirect the request to local storage instead of cloud (Markus,2011).

2.1.1. Mobile commerce

Mobile commerce (m-commerce) is a business approach by using a mobile devices in cloud computing enviroment for commerce. The m-commerce applications generally fulfill some jobs that need mobility (e.g., mobile transactions and payments, mobile messaging, and mobile ticketing). The m-commerce applications have to face some challenges such as low network bandwidth. Therefore, m-commerce applications are integrated into cloud computing environment to express its issues.

(Yang *et al, 2010*) proposes a 3G E-commerce platform based on cloud computing. This approach integrates the advantages of both third generation (3G) network and cloud computing to increase data processing speed and security level by using the public key infrastructure (PKI).

2.1.2 Jin and Kwok approach

(Jin and Kwok,2011) explain a solution to increase the bandwidth by sharing the limited bandwidth via mobile customers who are located in the same area (e.g., a workplace, a station, and a stadium) and have the same content (e.g., a video file). The researchers model the interaction via the users as a coalitional game. For example, the users form a coalition where each member is responsible for a part of video files (e.g., sounds, images, and captions) and transmits/exchanges it to other coalition members. This results in the improvement of the video quality. However, the proposed solution is only applied in the case when the users in a certain area are interested in the same contents.

Also, it does not consider a distribution policy (e.g., who receives how much and which part of contents) which leads to a lack of fairness about each user's contribution to a coalition.

2.1.3 Cloudlets approach

Having low tower signal reception may lead to low bandwidth and high latency in the network. Wi-Fi is an example which improves latency but it decreases bandwidth when many mobile devices are present. Introduction of 4G will help to improve the bandwidth and latency .

Cloudlets explained by (Mahadev, et al. 2009) may also help to improve this issue. Cloudlet is a trusted small computer or cluster of computers which can reside near the user and connected to the internet. So when user does not want to offload to cloud due to delay or cost, can use a nearby cloudlet. It downloads user data from a centralized location to permit mobile users to local access and thus reduce latency.

When work of user finished, data can go back to the centralized location.So demand of mobile users for interactive real time response by low-latency, one- hop, and high-bandwidth wireless access to the cloudlet can meet. If no cloudlet is present at that time then the mobile device may use default mode which will send all the requirements of user to the distant cloud.

2.1.4 SAMI approach

(Zohreh Sanaei et al.2012) proposed an arbitrated multi-tier infrastructure model (SAMI) aim to increase the outsourcing flexibility, augmentation performance, and energy efficient, without considering delay time (idle energy consumption) for decision making the system to perform mobile computation augmentation or augmentation, bandwidth, and utilization costs for contents running on a remote servers and the impact of bandwidth cost on energy consumption and on the quality of augmentation solutions.

2.1.5 MAUI model

In (J. Flinn, et al. 2002) the authors of MAUI studied the network delay effected on their system when using offload method. And mentioned that of any application, the offloading methods to a surrogate make a performance improvement only if the surrogate is located nearby. But when the surrogate distance increased, the performance will decrease. For other application such as video games, the performance difference between using a nearby and a long distant surrogate can be as large as 50%. By taken consideration of energy usage, the author found that two applications (chess and video games) consumed more energy when the mobile computer offloaded methods over 3G network than if the method had not been offloaded, whereas WIFI transmission, which led to lower distance, enabled offloading to always decrease the energy usage of cell phones.

Also in (Eduardo Cuervoy et al. 2010), the author introduced another MAUI approach that portion the application at a run time in three steps:

First: uses code portability to create mobile application for the local execution on mobile device and the other for the remote execution in cloud. The current smart phones use an instruction set architecture different from desktop and servers, so MAUI is prepared to execute any program on different CPU architectures.

Second: MAUI uses programming reflection to identify which methods of the application are marked ‘remoteable’ attribute or not and type safety to extract only the program state needed by the ‘remoteable’ methods. Then, MAUI sends the necessary program state to the cloud.

Third: the MAUI imagines each method of an application and determines the communication costs. Then, the MAUI combines three factors, the communication cost, mobile energy consumption, and network status at a runtime to construct a linear programming formulation. It can make optimal decisions for partitioning based on this formulation. The authors find that MAUI can maximize the potential for energy accepted in Wireless Communications and Mobile Computing.

2.1.6 Exploits cloud resources model

(V. March, et al 2011) develop a mobile application framework exploits cloud resources. In this approach, three parts, namely mobile , cloud, and hybrid (cloud mobile) are presented to extend design of mobile applications and development it with least component level communication and dependency that deals high functionality and resource efficiency. The cloud component is running on the cloud and mobile component is running on the mobile, while hybrid components are free to run either locally or remotely depends on computing capability of underlying smart phone. In order to reduce resource consumption, resource intensive instructions are developed as cloud components. During runtime, the execution request is sent to the cloud and results will be sent back to mobile device.

2.1.7 Virtualized screen model

(Yan et al. 2011) Proposed a Virtualized Screen or screen performance in the cloud, this proposed doesn't always mean putting the entire screen-performance job in the cloud. In his method, just part of smart phone’s screen is virtualized in cloud, which includes a group of data in display images, text contents, video and audio. The other applications with power-heavy computing execute on cloud. Thus, applications and interactions are offloaded and run in cloud, and on the local smart phones device some light power consuming applications are circulated, which could effectively minimize power consuming and decrease the interaction delay. This technique added flexibility that supply balancing among client device and distant cloud. However, the multimedia application and huge graphical interface reasons big issue to thin –client developers in order to achieve a small bandwidth consuming.

2.1.8 Thin-client model

In (B. Gopikrishnan,et al 2012), the proposed thin-client separate the distant application and local user interface to use distant server which circulated as virtual machine on the cloud. The server and client communicate via interactive screen-remote technique. The client sends its input to the server and the server reply of screen update to the client. Then the screen-update model calculates whether the screen can be efficiently pressed and transmitted to the client or not. The multimedia pressed can stilt the devices and the cloud efficiently, and could assistance the development of cloud computing become a upheaval.

2.1.9 Data Storage Issues In Mobile Cloud Computing

Most devices have limited storage to hold applications, data, multimedia and operating system. So, cloud storage is the most obvious use of cloud computing in mobile applications. The main questions that need here are data transfer size optimization, and data persistence versus data availability. Data transfer size optimization means how much data to move in a single transfer. In ideal side, the data transfer strategy should also have a mount of parameterization to handle stepping up and down the chunk size relative to network bandwidth, since bandwidth is highly variable in mobile applications. Data availability is important factor for finishing tasks in a currently running process. Data persistence refers to keeping data in remote cloud until it is needed again in future. There is a trade-off between them which requires taking into consideration of network connectivity, bandwidth, device capacity.

2.1.10 Cooperative caching framework:

Caching can be used, but this used of cache on distributed databases requires. Data caching is one of the techniques widely used in wired and wireless networks to improve data access efficiency. Cooperative caching (Preetha and Jacob 2013) consists of multiple distributed caches to improve system response time. Having distributed caches permits a system to deal with concurrent client request as well as sharing contents. This can also reduce response time by concurrently retrieving objects from different cache sites. Concurrent retrieval of objects from different cache sites is beneficial as opposed to the remote cloud server which will result in latency and bandwidth issues.

Compared with traditional cloud computing mobile cloud computing poses a challenge in the way mobile device access data stored in the cloud. Having caches closer in proximity to certain group of users is an effective way to reduce average network latencies since there is a correlation between the location of the user and the object requested.

2.2 Bandwidth Issue In Mobile Cloud Computing

Now we concern about bandwidth. Bandwidth is one of the big issues in MCC since the radio resource for wireless networks is much scarce as compared with the traditional wired networks.

2.2.1 Bandwidth portioning model

(Jin and Kwok 2011) Propose a solution to portion the bounded bandwidth among mobile customers who are located in the same area and involved in the same media (e.g., a video file). The authors design the interaction among the users as a coalitional game. For example, the users form a coalition where each candidate is responsible for a part of video files (such as sounds, images, and captions) and sends it to other coalition candidates. This approach made an enhancement of the video quality. But this solution is only suitable when the users in appointed area are interested in the same contents. Also, it does not consider a distribution policy (such as who receives how much and which part of contents) that commands to a reduction of fairness about each user's participation to a coalition.

2.2.2 Data division policy model

In (Wang, Jung and Akella 2010), the authors explained the data division policy which assigns when and how much portion of obtainable bandwidth are shared among users from which networks (WiFi or WiMAX networks). It gathers user profiles (e.g., calling profile, signal strength profile and power profile) in periodic time and generates a tables by using Markov Decision Process (MDP) algorithm. Established these tables, the users assign whether or not to assist other users download some contents that they cannot receive by themselves because the restriction of bandwidth, and how much it should assist (e.g., 10% of contents). The authors build architecture, named RACE Resource-Aware Collaborative Execution, on the cloud to utilize the advantage of the computing resources for preserving the user profiles.

This technique is suitable for users who share the restricted bandwidth, to balance the trade-off between advantages of the assistance and energy costs.

2.2.3 Cognitive radio model

An effective network access management not only enhances link rendering for mobile customers but also improves bandwidth usage. Cognitive radio can be a best solving to attain the wireless access management in mobile communication area (Yucek and Arslan 2009). Cognitive radio maximizes the effective of the spectrum employment observably, by permitting forbidden customers to entry the spectrum specified to the authorized customers. When this approach is incorporated into MCC, best result may obtain that the spectrum can be applied more efficiently,

the spectrum deficiency can be fixed and thus millions of dollars for network suppliers can be saved (Chiang and Bostian 2011). But, Cognitive radio is determined as wireless communication technique in which every node connects through a special wireless system based on assessment of radio resource obtainable in heterogeneous wireless communication framework. Therefore, mobile customers in Mobile Cloud Computing must have the ability to discover this radio resource availability whereas confirming that the traditional services will not be intervened.

2.2. 4 Sonora model

(Fan Yang, et al 2012) Is a platform for mobile-cloud computing. Sonora is designed to backing the increase of mobile-cloud services. Sonora supplies providers with stream-based programming interfaces that consistently combine a wide range of presenting approaches from mobile, database, and distributed systems. Sonora's implementation engine is a fault-tolerant distributed runtime that uphold user-facing sensing and operation services in the cloud. Main characteristics of this engine are its dynamic load balancing manner, and an incoming error maintenance protocol that acts a checkpoint-based fractional rollback maintenance with re-execution that chosen. To explain the critical and power of the stream summary in explaining a complicated mobile-cloud services we estimate Sonora's model in the context of two services. We also validate Sonora's design, the result shows that Sonora is efficient, scalable, and supplies reactive fault tolerance.

2.2.5 Intelligent mobility management model:

One of the key issues encountered in a mobile cloud is the design of intelligent mobility management techniques that support user mobility while providing a seamless service (Klein, A., Mannweiler et al. 2010).

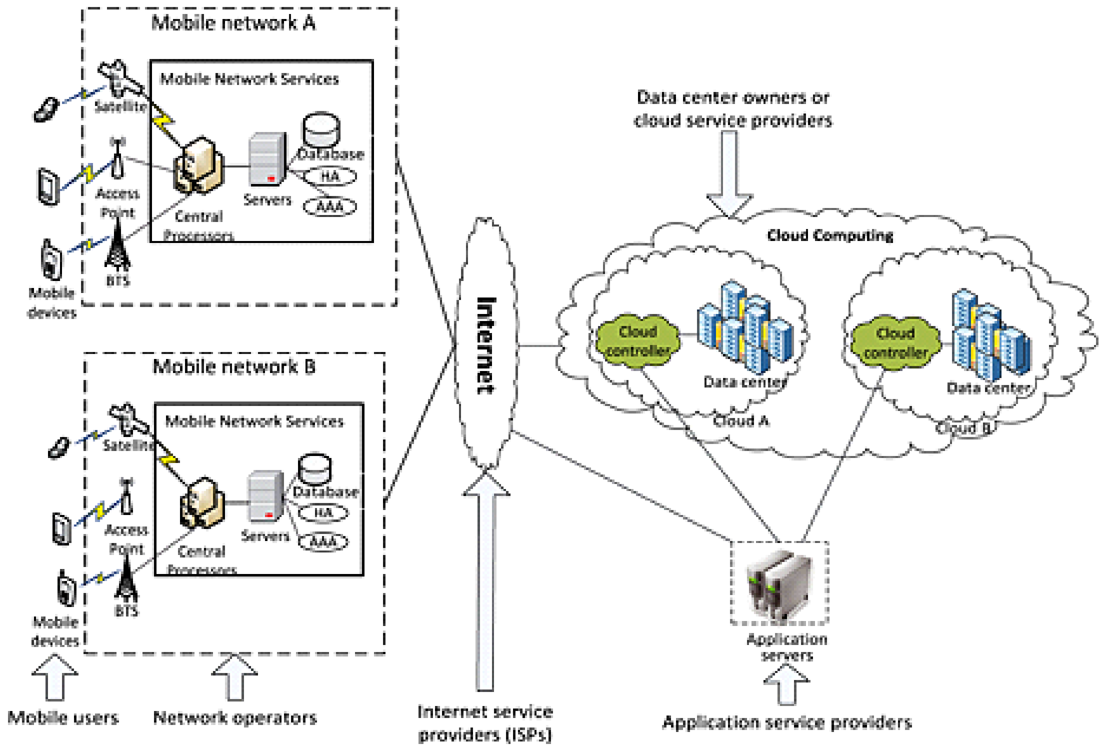


Figure 2-1: Mobile cloud computing architecture (Dinhthaihoang, et al. 2012)

There are six styles of Mobile Architecture (Klein, A., Mannweiler et al 2010):

- Thick Client: both application and application data resides on device
- Rich Client: application resides on device, data sourced from online sources.
- Thin Client: browser or small downloadable client, application nor data resident on device.

- Streaming Client: content and application are streamed from centralized server.
- Messaging Client: Short messages are delivered at anytime to the device, do not require user interaction to accept
- No Client: Voice, DTMF, or RFID often provide inputs, content delivered in same manner.

In this research, it is focusing on rich client mobile architecture, as it is the most suitable for a mobile cloud-computing infrastructure, and provide higher level of functionality.

Table 2-1: Challenges In Mobile Application Development (Klein, A., Mannweiler et al 2010)

Challenges	**Service Area**
Development approach, platform and version	Development
Multi device testing, memory profiling	Testing
Application Authentication, Malware protection	Security
Consistent UI, Intuitive	Usability

Multimedia streaming applications have disruptively occupied bandwidth in wire line Internet, yet today's fledging mobile media streaming still poses many challenges in efficient content distribution due to the form of mobile devices.

At the same time, cloud computing is gaining power as a promising technology to transform IT industry and many eminent enterprises are developing their own cloud infrastructures. However, the lack of applications hinders clouds' large-scale implementation (Jin, X., & Kwok, Y. K. 2010).

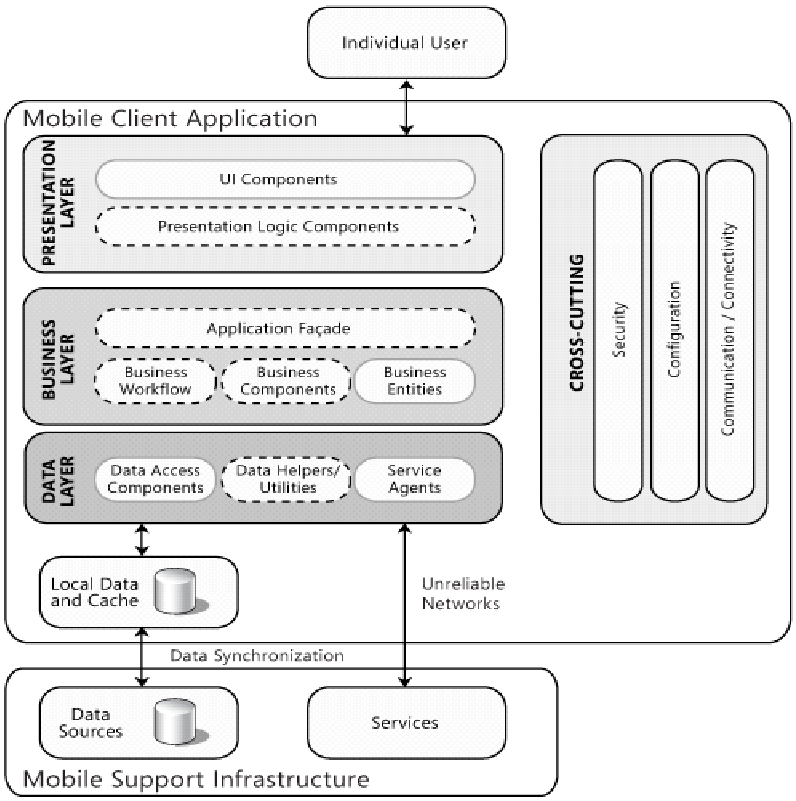


Figure 2-2: General Mobile Application architecture (Microsoft Corporation, 2009)

2.2.6 Distributed computing model

(Ketan B. Parmar,et al 2013) Is a technology that describes a type of distributed computing. This technique is mainly utilized in the context of resource exploitation and sharing via geographical borders. It assists in coordinating computing capacities, data storage and network resources through dynamically dispersed organizational architecture. These environments of application implicitly fetch with them two primary challenges depending grid computing administration of resources and the planning of applications. The heterogeneous quality of resources in terms of structure, power, configuration and obtainable complicates the issue of planning jobs. Reduction of make span is the main goal of grid planning. So, confirm specific features of the job must be considered and the metrics to be utilized should be established accordingly. The best choice for a grid scheduling environment is a heuristic algorithm called the Ant algorithm. The free time of the resources and the execution time of jobs are taken into account by the proposed Ant algorithm to achieve better scale utilization of resources and better scheduling of jobs. By using the standard benchmark problem, a number of intensive experiments are carried out in the evaluation study. The result illustrates that the proposed algorithm is capable of providing high quality job scheduling to grid resources. Thus the algorithm can be employed in case of real time grid environments to design efficient dynamic schedulers.

2.2.7 Think-Air framework

(Sokol Kosta at al 2011) Smartphones have exploded in popularity in recent years, becoming ever more sophisticated and capable. As a result, developers worldwide are building increasingly complex applications that require ever increasing amounts of computational power and energy. Here we talk about Think-Air, a framework that makes it simple for developers to migrate their smartphone applications to the cloud. Think-Air exploits the concept of smartphone virtualization in the cloud and provides method level computation offloading. Advancing on previous works, it focuses on the elasticity and scalability of the server side and enhances the power of mobile cloud computing by parallelizing method execution using multiple Virtual Machine (VM) images.

it evaluate the system using a range of benchmarks starting from simple micro-benchmarks to more complex applications. First, we show that the execution time and energy consumption decrease two orders of magnitude for the N-queens puzzle and one order of magnitude for a face detection and a virus scan application, using cloud offloading. We then show that if a task is parallelizable, the user can request more than one VM to execute it, and these VMs will be provided dynamically. In fact, by exploiting parallelization, we achieve a greater reduction on the execution time and energy consumption for the previous applications. Finally, we use a memory-hungry image combiner tool to demonstrate that applications can dynamically request VMs with more computational power in order to meet their computational requirements.

2.3 Related Work Compared with Proposed Work:

As mentioned previously, all the previous research concentrate with different level of issues that face the mobile cloud computing in general. Many of them focus on bandwidth consumption problem, different method was proposed to reduce the bandwidth consumption in order to enhance the mobile cloud computing at all either by portioning the available bandwidth among users or by data division policy which assigns when and how much portion of obtainable bandwidth are shared among users and so on. But in this proposed work, every user will be downloading little amount of bits from different mobile customers shared databases that establish on the specific WIFI network, and at the same time-sharing downloaded bits with other various users. The main advantage of our work that it solves the issues of downloads a large file and conveniently solves the issue with seamless download process running in the background, so the user can use and share once it is ready.

Chapter three

Implementation

Requirements

and

Analysis

3. Implementation Requirements and Analysis:

In this application , the main idea is to create a shared database between the users in the same network. Therefore, this shared database is the basic stone in our application as long as the user of the application will use it to organise data transportation between the users. However, In this section we explained this application in more details.

This application look to improve the bandwidth consumption when downloading a large files from the cloud to mobile devices, we are going to create an application that utilizes our proposed methodology and algorithm. However, the application will be built using java as an android application.

Here, the main architecture of application that sharing different files between a users that close to each others, that which minimize the downloading amount from the clouds in order to also minimize the consumption of bandwidth. To reach this decreasing the users should be connected together via peer to peer network or using a specific local area connection. The requirements analysis of software systems can be Functional or Non-functional

3.1 Functional and Non-Functional Requirement:

Functional Requirements describes what a software system should do, while non-functional requirements place constraints on how the system will do so

3.1.1 Functional Requirement:

In this thesis and depending in our idea we need some functional requirement and others of non-functional requirement. Functional requirement specify what the product must do. They relate to the actions that the product must carry out in order to satisfy the fundamental reasons for its existence. this application includes the primary buttons; each one has a specific function as follow:

- Main Activity:

  This screen (window) displays the activity that plays the important role in coordinating the navigation between activities that we need in our proposed application.

- Listing Files button:

  Because the necessary of exist a database file for each device that stores the name, extension, and the physical id of each file. This button will show all the files exist in database.

- Connecting Devices button:

  When pressed in this button, it show all the useful information about the network such as the IP address of each participate device in the same area network and other needed information.

- Getting a file button:

  When pressed in this button, it show all the requested that came from the other devices. It also navigate to a screen that has text boxes to enter the file name and file extension. Each device to sends either unicast or multicast broadcast using UDP broadcasting protocol, which is a standard for data transaction does not send acknowledgment for the sent packet, this is usually used for sending video streaming or voice data, because it is faster than TCP, this was used in the proposed system to request a specific file from other device's database. When the other devices receive the broadcast they will search their own database file to see if they do have the file or not.

- Requesting button:

  When press on this button, it show all the request that came from the other users (from any device, to which devices and so on).

- Clearing Database button:

  This button assists the user to delete the content of the database file.

3.1.2 Non-Functional Requirement:

Is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. This should be contrasted with functional requirements that define specific behavior or functions. The plan for implementing functional requirements is detailed in the system design.

1- Privacy: Storing information in the cloud could make your company vulnerable to external hack attacks and threats. As you are well aware, nothing on the Internet is completely secure and hence, there is always the lurking possibility of stealth of sensitive data.

2- Security: In this work, each device has its own database file so that the privacy and security can be maintained since no device has access to the other's database files. But in our application we necessarily need a sharing process between all users that connected in the same network, this will decrease the privacy of user database.

3- Availability: this special attributes permit us of free moving beyond time zone and geographic location issues. In this application using a cloud computing approach that supply a wide range of access any resources and anywhere you can access the Internet, and you can access them from a mobile phones.

- Interface requirements: in this proposed application this requirement appears clearly. The user interface will achieve in this system.

- Lifecycle requirements lifecycle requirements which can be classified under two sub-categories: Quality of the design, such as maintainability, enhance ability, portability; in our application these features are exist. The other kind is the limits on development, other software lifecycle phases, such as development time limitations, resource availability, methodological standards etc.

3.2 Software Requirements:

- Platform: Android
- Framework: Android AVD 4.4.2 under Java
- Editing tool: netbeans IDE 8.0.
- Back-End tool: SQLite Database (for relational database).

3.3 SQLite Database:

SQLite is a relational database management system. In contrast to other database management systems, SQLite is not a separate process that is accessed from the client application, but an integral part of it.

SQLite is ACID-compliant and implements most of the SQL standard, using a dynamically and weakly typed SQL syntax that does not guarantee the domain integrity.

SQLite is a popular choice as embedded database for local/client storage in application software such as web browsers. It is arguably the most widely deployed database engine, as it is used today by several widespread browsers, operating systems, android devices and embedded systems among others.

3.4 Specification:

1- Specific local area network where different of peers user connected together. These mobile devices sharing some files that may exist in one of them and other peers in the same local network need it.

2- The mobile phones that participate in this application shall be a good powerful device and located close together in the same local network, such as mobile A, mobile B and mobile C in the same network and peers together.

3- The application shall be possible to select an appropriate cloud provider(s) for internet working .

4- To achieve the main goal of our proposed idea, this application must Exist in all participant devices.

5- The application shall be possible to create a database file in each device to sharing with each other devices by redirect the frequent request to the close users in order to avoid the increasing in bandwidth consumption.

6- Resource discovery : This function is to search about a specific file that any device need it , this will do by send a request to other peers device, if the file find in any of them, it will return to the requested device with a unicast way.

7- Broadcast and unicast.

This framework scenario useed three devices (X, Y, and Z) which connected in the same network. Every device has its own database file which creates by using SQLite (a relational database management system) to stores all that exist in the device. To sending request or file the process use the User Datagram Protocol (UDP) broadcasting. The search operation make by multicast approach and the response will do by unicast approach when the file found in a such device.

Our thesis is aimed to supply an application to decrease the consumption of bandwidth in file sharing. However, in order to achieve the proposed goal, the project built based on sharing files using P2P network, LAN, and WLAN for the people who are located on the same building, company, or using the same network.

3.5 Implementation Requirement:

This application will search about the requested file in our own database file in the other devices to see if they do have the file or not and will use a hashing string search algorithm in case the file has an alias name. We use Rabin-Karp algorithm as shown in the following figure. This algorithm was created by Michael O. Rabin and Richard M.Karp in1987 (Permalink 2012)..

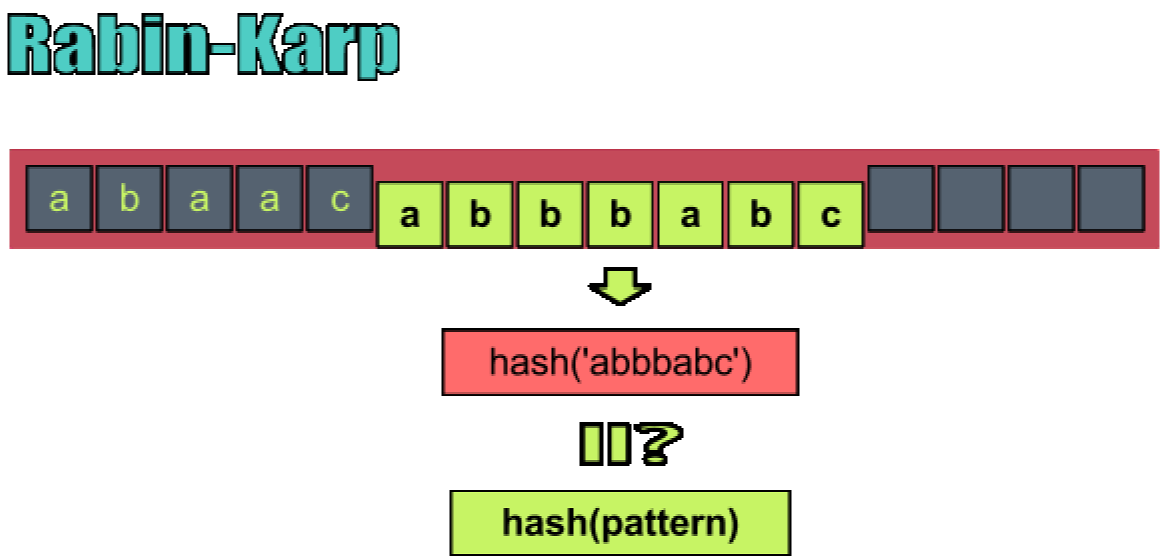


Figure 3-1: Rabin-Karp hashes the pattern and the sub-string

The hash function may vary depending on many things, so it may consist of ASCII char to number converting, but it can be also anything else. The only thing that need is to convert a string (pattern) into some hash that is faster to compare (Permalink 2012).

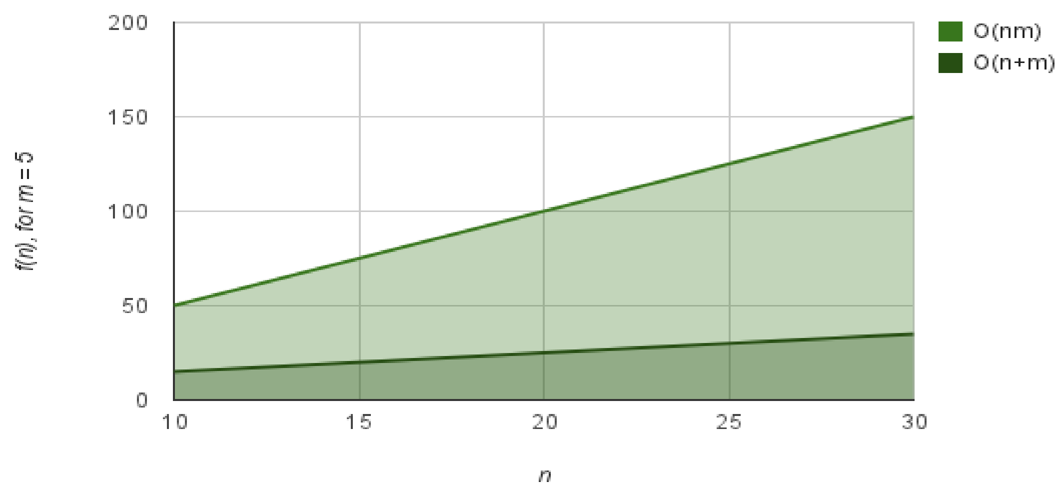


Figure 3-2: Complexity of Rabin-Karp algorithm(Permalink 2012).

The above figure, show the complexity of the Rabin-Karp algorithm, it has the complexity of O(nm) where **n**, of course, is the length of the text, while **m** is the length of the pattern. When compared to brute-force matching find that brute force matching complexity is O(nm), so as it seems there's no much gain in performance. However it's considered that Rabin-Karp's complexity is O(n+m) in practice, and that makes it a bit faster (Permalink 2012).

When this algorithm compared with the suitable searching algorithm, for example in brute force sub-string matching checked every single character from the text with the first character of the pattern. Once it have a match between them it shift the comparison between the second characters of the pattern with the next character of the text.

Chapter four

System

Implementation

and

Result Analysis

4.1 System Architecture:

In this proposed project will designed a model that enhance the mobile cloud computing applications by sharing database files among different users that located in the same local network that reduce the redundant request to the cloud in order to minimize the bandwidth consumption.

This proposed system solved several problems found on other systems, mainly its independence on other devices or the location, also its mobility, that allows the user to roam in different locations while it is still operable. On the other hand some algorithms, discussed in chapter 2, require time delay during the operation, that each file even that was requested previously, requires more time to be downloaded, but on this system, this file does not require time delay, because it is based on the chunks that distributed all over the mobile phones.

This project is aimed at providing a application to decrease the consumption of bandwidth in file sharing. However, in order to achieve the proposed goal the project built based on sharing files using P2P network, LAN, and WLAN for the people who are located on the same building, company, or using the same network.

In order to prove the practical side of thesis, we are going to build an application that uses the proposed application and algorithm. However, the application will be built using java as an android application.

In this work, the idea apply by are to decrees the consumption of bandwidth by sharing files among the users of the network instead of downloading files from the clouds so that users can send and receive files from each others.

However, to achieve the goal of this idea the users should be connected together through peer to peer network or using a local area connection so that the bandwidth won’t be consumed as displayed in the following figures:

Assume that have 3 devices; device A, B, and C. Device A wants to download file x from the cloud.

In the first request, device A send a request to the cloud to download a specific file, so it consume a such amount of bandwidth between the device and the cloud. After that this file will save in the A's database.

Then, when any device that connected with device A in the same local network (for example device B or C) want the same file (A's file that downloaded from the cloud), it send a request to the other device that exist in this network. Because each device will has its own database file that stores the name, extension, and the physical id of each file. Such device sends either broadcast or multicast broadcast using UDP broadcasting protocol and each device that received this request check if the request file exist in its own database or not. If it exist, it will send by a unicast to the device that need it.

When the other devices receive the broadcast they will search their own database file to see if they do have the file or not; this process will make by using a hashing string search algorithm in case the file has an alias name (i.e. file aaa may be exist with multiple possibilities like aAa, AAA, Aaa).

If the requested folder does not exist then the user can download it from clouds.

4.2 The Architecture Of Our Proposed Application

In more details, our proposed system is consists of three processes; each process has its significant rule in the functionality and attribute of the system as the following:

1- Requesting and Response:

When a user need a file he request it from the other users, and the other users determine whether they want to accept or decline the request. This process helps in improving the application's security by giving the user the ability to decide what and with whom to share and guarantees the user's privacy.

For sending request or file we are going to use the User Datagram Protocol (UDP) broadcasting according to the two different following scenarios:

A- When user request a file he use UDP multicast which means the user will send a message of the file name and extension to every other device in the network.

B- When user wants to response to a request he will send the file or any other message to the user who requested the file using unicasting. The following algorithm is about how to obtain the IP address:

Getting the IP addresses;

We can add IP address manually and store than in the database so each time . we need to some request we use than

IntelAddress is a data type in android which store ip address;

```
Function getips()
IntelAdress getBroadcastAddress()
{
WifiManager wifi = Context.getSystemService(Context.WIFI_SERVICE);
DhcpInfo dhcp = wifi.getDhcpInfo();
Int broadcast = (dhcp.ipAddress & dhcp.netmask) I ~ dhcp.netmask;
Byte[] quads = new byte [4];
For (int k=0 ; k<=4; k++) quads [k] = (byte) ((broadcast >>K * 8) & 0XFF;
Return IntelAdress.getByAdress(quads);
}
```

As we notice from this code, we use a function named (getips) to broadcast a request to obtain the IP address for each device because it needs when request a specific file from device's database .

Incase of electricity down or any other reason of interrupting connection during sending of file an error will be shown to the both sender and recever based on the following peice of code:

```
Connection Manager Packet = Connection Manager ( ) ;
If (Packet.is Connected= = false)
 {
Context Context  = get Application Context  ( ) ;
CharSeguence text = " Connection Error " !!
Toast toast =Toast . make Text (Context,text
Toast. LENGTH_SHORT),
}
Toast.Show ( ) ,
```

2- Searching Files:

The problem in file names searching is that the file can has different names. In order to solve this problem searching the file depending on its stream of binaries is the best and maybe the only way. However, for this project we can’t solve this problem because we are building this project for the smart devices and using a such application (searching the binary streams) need big capabilities of computing speed which is not really available for smart devices and mobile phones. Therefore, we tried to solve this problem by using the hash string search algorithm or (Rabin-Karp algorithm) that mentioned previously in implementation requirement section.

Rabin-Karp algorithm uses hashing to find anyone of a set of pattern strings. However, the complexity of this algorithm for a text of length n and p patterns of combined length m is O(N+m) for the best case and O(nm) for the worst case.

```
1-  Function RabinKarp (string s[1..n], string sub[1..m]
2-  hsub: = hash(sub[1..m];  hs:= hash (s[1..m])
3-  for I = 1 to n-m-1
4-  if hs = hsub
5-          If s[1..i + m-1] = sub
6-                  return i
7-      Hs := hash (s[i+1 .. i+m]
8-   return not found
```

Lines 2, 5, and 7 each require O(m) time. However, line 2 is only executed once, and line 5 is only executed if the hash values match, which is unlikely to happen more than a few times. Line 4 is executed *n* times, but only requires constant time. So the only problem is line 7.

So what we are going to do to solve the problem of recalculating the hash value in line 7 is to reuse the value of hash which we calculated in line 2. Therefore the algorithm will be like this:

```
1-  Function RabinKarp (string s[1..n], string sub[1..m]
2-  hsub: = hash(sub[1..m];  hs:= hash (s[1..m])
3-  for I = 1 to n-m-1
4-      if hs = hsub
```

5- If s[1..i + m-1] = sub

6- return i

7- s[i+1 .. i+m] = s[i.. i+m-1] – s[i] + s[i+m]

8- return not found

Note that the file extension will be searched separately so that it won't be included in the hash therefore it won't affect the time of search. For more algorithms show "Appendix A".

As we mentioned previously, The supported database of android is SQLite Database which provide a great capabilities of data storing and archiving. However, in this project we built up three classes in order to deal with the SQLite.

The MySQLiteHelper class. This class is responsible for creating the database. The onUpgrade() method will simply delete all existing data and re-create the table. It also defines several constants for the table name and the table columns.

```
package s.m.SharingPro;
import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
public class TodoDatabaseHelper extends SQLiteOpenHelper {
  private static final String DATABASE_NAME = "FilesTable.db";
  private static final int DATABASE_VERSION = 1;
```

```
  public TodoDatabaseHelper(Context context) {
    super(context, DATABASE_NAME, null, DATABASE_VERSION);
  }
  // Method is called during creation of the database
  @Override
  public void onCreate(SQLiteDatabase database) {
    TodoTable.onCreate(database);
  }
  // Method is called during an upgrade of the database,
  // e.g. if you increase the database version
  @Override
  public void onUpgrade(SQLiteDatabase database, int oldVersion,
      int newVersion) {
    TodoTable.onUpgrade(database, oldVersion, newVersion);
  }
}
```

4.3 User Interface Design

Now, in this section we display the main interface's screens that used in our proposed application. This proposed application must download in each mobile device, and the database for each user must been in the device itself, that which decrease the load in the network. The main interface of proposed application as follow:

1- The main screen in our design, this screen includes a five buttons, each of them do a specific function as we mentioned in the software requirement section.

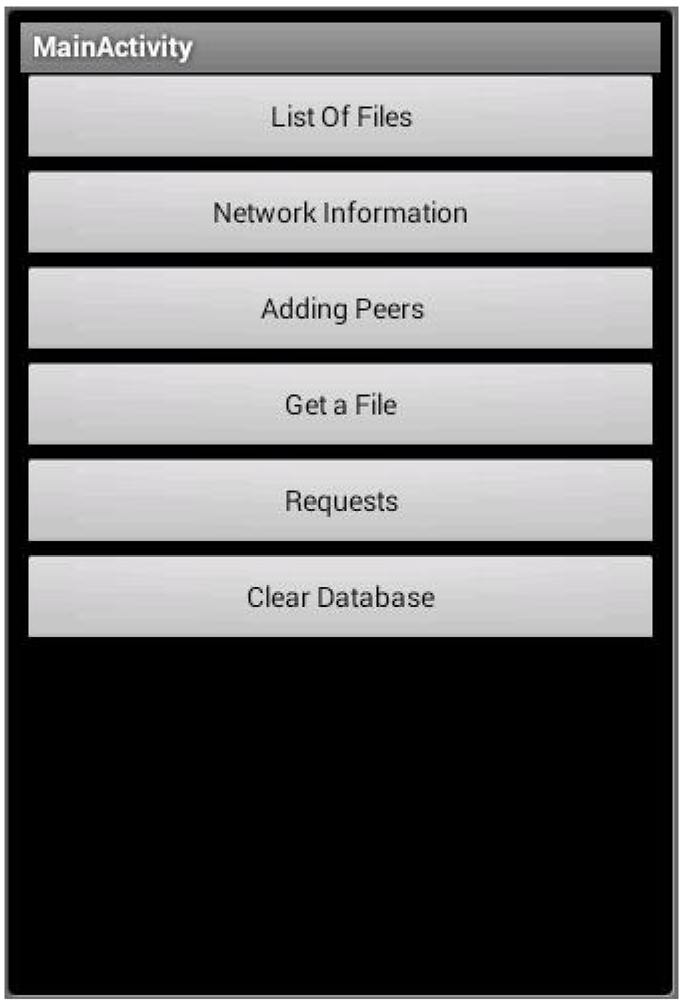


Figure 4-3: Main Screen of Our Application

2- The search screen. In this screen as we notice, we enter the file name that we needed and the extension, also we put a two options that specified if we want to send a request to a specific device or user, or if we want to broadcast the request to all other users. Then we pressed on the "Send Request" button in order to send a search request, the search process will use a search function with search algorithm to check if other devices fond the request file or not.

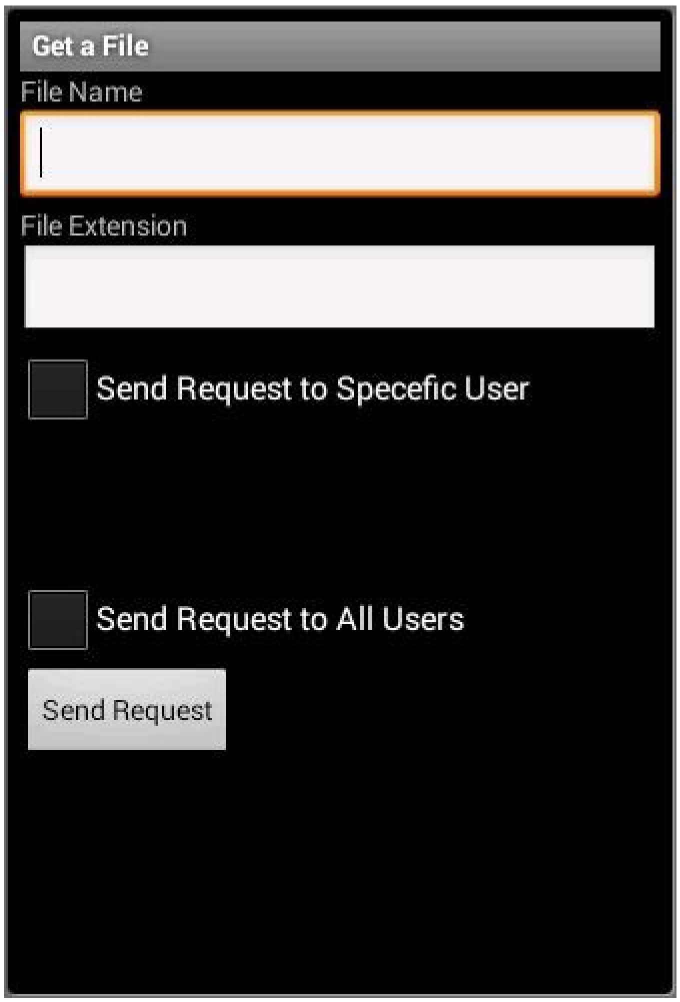


Figure 4-4: The Search Screen

3- Clear Database screen. In this screen we have a question if we want to clear our database file or not, if yes we press on the button which named " Clear Database" as we display below:

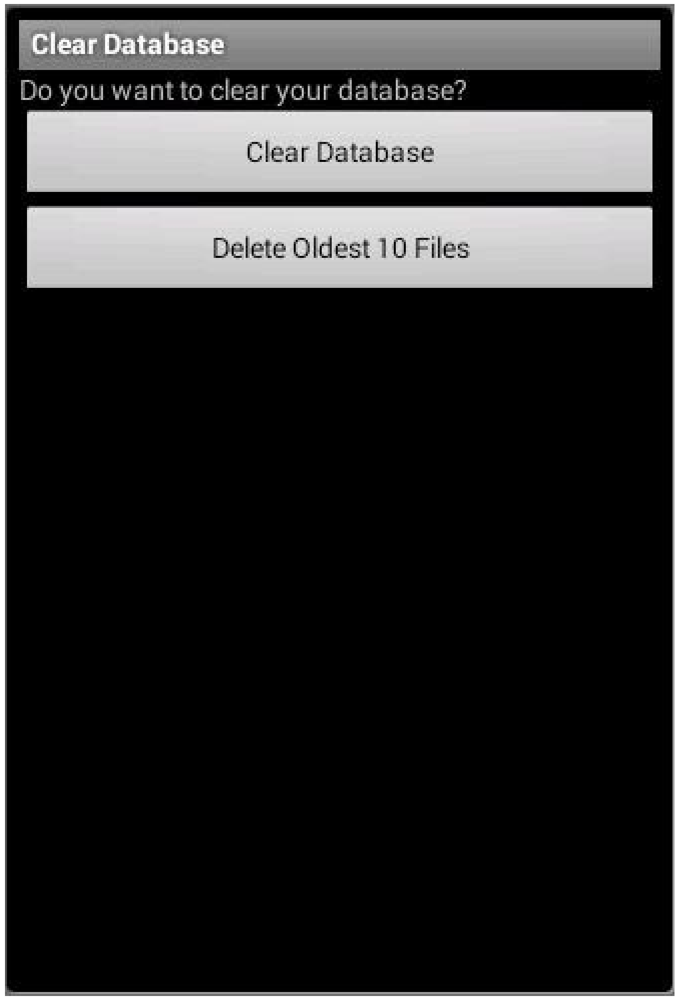


Figure 4-5: Clear Database Screen

4- Connected device screen: This screen give us an information about the traffic in a specific network where the devices connected. It explained the amount of transmitted byte and the received byte in this network.

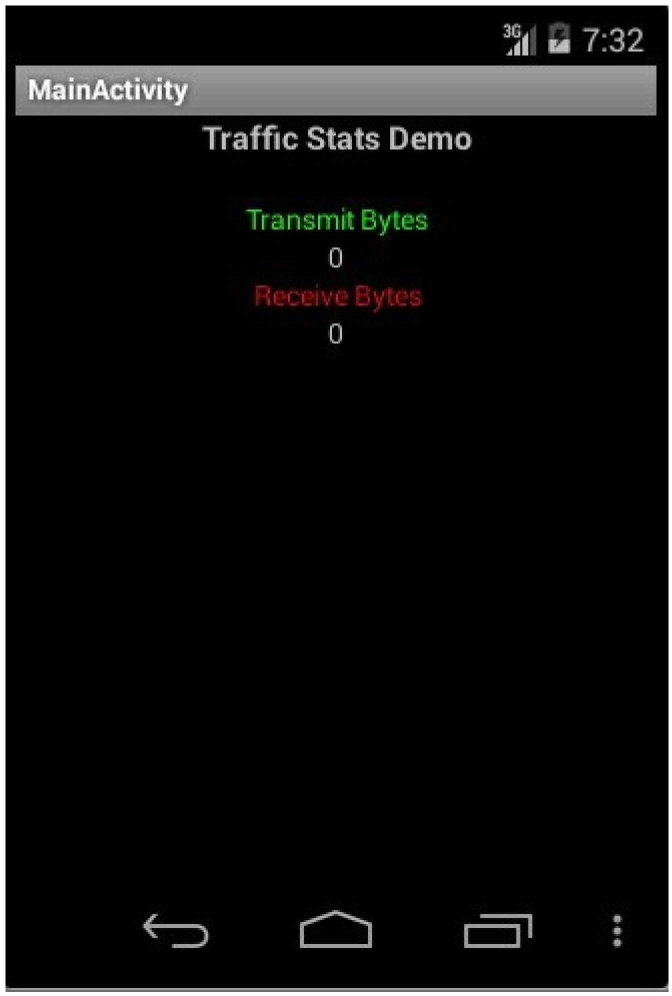


Figure 4-6: The network status screen

4.4 Results Analysis:

The performance of our proposed application refers to determine many parameters especially the bandwidth consumption that consumed when request a different files from the cloud. However, in order to achieve the proposed goal, the project built based on sharing files using many different kinds of networks such as P2P network, LAN, and WLAN for the people who are located on the same building, company, or using the same network.

Our project uses the UDP (User Datagram Protocol), which it is packet-oriented; its main purpose is only to transmit one packet from an application to another. The protocol does not try to compensate for possible packet loss on the way, nor does it ensure that packets are received in the same order as were sent. We use this protocol because the main advantage of it, which is the greatly improvement delay. Since the loss of a single packet does not delay the receiving of all the following packets until the lost one is retransmitted, beside this advantage, the UDP is connectionless oriented protocol.

The results show that the amount of bandwidth consumption is mainly related to the network type that we are used. In other term, if the network is P2P or the connection between devices is Bluetooth or infrared, the bandwidth consumption is reduced a large amount, so the performance may reach to a perfect level.

And when we use a LAN network may also reach a perfect level of results that improve and enhance the bandwidth consumption in mobile cloud computing application because we also using the infrastructure of the LAN network.

When talking about networking and data transferring between clients then bandwidth consuming and allocation will be the major concern. Bandwidth has nothing to do with P2P networks such as Bluetooth and infrared and even WiFi, which can be use them freely. The same applied when talking about a local area network (LAN).

A practical test to verify the proposed algorithm was done using the software application that simulates the algorithm, this software were developed on Android language, based on Android operating system. The developed application were installed on three types of smart phones to ensure that the algorithm is applicable regardless of the smart phone vendor, and also to proof the contribution of the algorithm in different environments, which are in this case different smart phones with different specifications and different internet bandwidths.

The test uses Galaxy S4, Nexus 7 and HTC One smart phones. It also uses one Mega Byte (1 MB) Internet speed and 2MB internet speed. The test were done six times on each scenario using a 90MB file. On each test this file was first requested from the cloud, the consumed Bandwidth (BW) from the internet and the download time were measured. Then the same file was requested again from another smart phone in the same network, the consumed BW from the internet and the download time were measured again. The test scenarios are as follows summarized in the below table.

Mobile 1: Galaxy S4.

In the first mobile phones; Galaxy S4, initially a 90MB file were requested from the cloud, using internet with 1mbs/s speed. It took 17 minutes to be downloaded. However, the same file were requested again from another smart phone in the LAN, with zero consumed bandwidth. In this case the proposed algorithm checked that this file was downloaded before, then it determined the user that have downloaded the file, then it automatically forwarded the destination route of the request to that user instead from the cloud, and finally downloaded the file from the shared memory for that phone. It was allowed to download the file using three types of mediums, for testing and comparison issues only, however, the final decision was to use only Wireless (WiFi) medium because of its speed. The results were as follow.

Downloading the file using Bluetooth it took about 12 minutes. It needs 29 minutes if using infrared and 45 second if using the LAN (WiFi) network. The proposed system defines that in the first request the file downloaded from the cloud, but next the same file will be downloaded from the shared memory of the first mobile. This is done automatically using the contributed algorithm. However, if the same file was requested from a third smart phone, the download speed and time will be reduced to the half, because the contributed algorithm will download the file from both of the mobiles that requested it first.

Mobile 2: Nexus 7

Here, the same file of 90 MB were requested form the cloud again by using internet with 1mbs/s speed then it took about 15 minutes. However, when we the same file was requested again no bandwidth was consumed and if this file was sent using Bluetooth it needs about 10 minutes. And needs 25 minutes if it used infrared, and 40 second if using the LAN (WiFi) network. The proposed system defines that in the first request the file downloaded from the cloud, but next the same file will be downloaded from the shared memory of the first mobile.

Mobile 3: HTC One

In the third phone, also the file of 90 MB was requested initially form the cloud by using internet with 1mbs/s speed then it took about 19 minutes. However when it was requested again the same file from sharing database and with no bandwidth consumed and if it was sent using Bluetooth, it needs about 13 minutes. And needs 32 minutes if using infrared and 45 second if using the LAN (WiFi) network. The proposed system defines that in the first request the file downloaded from the cloud, but next the same file will be downloaded from the shared database of the first mobile phone.

In the second part of the test, the same mobile phones and the same files were used but with internet speed of 2mbs/s as follow:

Mobile 1: Galaxy S4

The file of 90 MB was requested form the cloud by using internet with 2mbs/s speed then it took about 7 minutes to be downloaded. In the second file request, the same previous file from sharing database was downloaded with no bandwidth consumed, and it was sent using Bluetooth with about 12 minutes. And with 29 minutes if using infrared and 45 second if using the LAN (WiFi) network. The proposed system defines that in the first time the file downloaded from the cloud, but next it will be downloaded from the shared database of the first mobile phone.

Mobile 2: Nexus 7

In this mobile phone, the file of 90 MB was requested form the cloud by using internet with 2mbs/s speed then it needed about 6 minutes. In the second file request, from the shared database, if using Bluetooth it needs 10 minutes but with no bandwidth consumed. And needs 25 minutes if using infrared and 40 second if using the LAN network. The proposed system defines that in the first time the file downloaded from the cloud, but next it will be downloaded from the shared database of the first mobile phone.

Mobile 3: HTC One

In the third phone, it was also request of the 90MB file form the cloud by using internet with 2mbs/s speed then it needs about 8 minutes. In the second file request of downloading the same previous file from sharing database with no bandwidth consumed, using Bluetooth about 13 minutes. And need 32 minutes if it uses infrared and 45 second if using the LAN network. The proposed system defines that in the first time the file downloaded from the cloud, but next it will be downloaded from the shared database of the first mobile phone.

The following tables summarizes the above results:

Table 4.2: Download Time and Bandwidth Consumption in Direct Download from the Cloud, Using Two Internet Download Speeds, and a 90 mb File

File Destination Mobile	Download Time	Bandwidth Consumption	Internet Download Speed
S4	17 minutes	90 MB	1 mb/s
Nexus 7	15 minutes	90 MB	1 mb/s
HTC	19 minutes	90 MB	1 mb/s
S4	7 minutes	90 MB	2 mb/s
Nexus 7	6 minutes	90 MB	2 mb/s
HTC	8 minutes	90 MB	2 mb/s

Table 4.3: Download Time of mobile-to-mobile on a WIFI network, With Zero Bandwidth Consumption, Using a 90 mb File

File Destination Mobile	File Source Mobile	Download Time
S4	Nexus 7	45 s
	HTC	45 s
Nexus 7	S4	40 s
	HTC	40 s
HTC	S4	45 s
	Nexus 7	45 s

Table 4-4: Download Time of mobile-to-mobile using Bluetooth, with Zero Bandwidth Consumption, Using a 90 mb file

File Destination Mobile	File Source Mobile	Download Time
S4	Nexus 7	12 m
	HTC	12 m
Nexus 7	S4	10 m
	HTC	10 m
HTC	S4	13 m
	Nexus 7	13 m

As noted from the above tables, we conclude a result of downloading a file of size 90 MB by using different type of smart phones, each one has a different features. In our result we make a two request, the first request sent to the cloud, but the second one sent to other devices that located in the same network. In the second request, we test a bandwidth by different network connection; Bluetooth, infrared, and LAN. Firstly, we test the bandwidth with speed 1 mbs/s and then with 2mbs/s.

In the first request of such file of 90MB consumed 90MB bandwidth from the internet when requested in the first time. However, if it was requested again, second request of the same file (maybe by the same device or other one but the same file that downloaded in the first request), which done from the cloud, no matter how many times, the file will be downloaded from the shared database memory of the smart phones that initially requested the file, which indicate that no bandwidth will be consumed in these requests.

Taking into consideration the download times comparison of cloud vs. WIFI, the proposed approach has two advantages over the cloud download, these are zero-bandwidth consumption and a much shorter download time. Even if we use higher internet speed connection, which can give lower download time, but the zero-bandwidth remains a major advantage.

This result proof the enhanced performance of the contributed approach. Which determines if the file was requested in this LAN, the file will be downloaded from the shared memory of the mobiles that first used to request it. Then in the third and forth and whatever request more, the file will be downloaded from several shared memories of mobiles, which enhances the speed of the download.

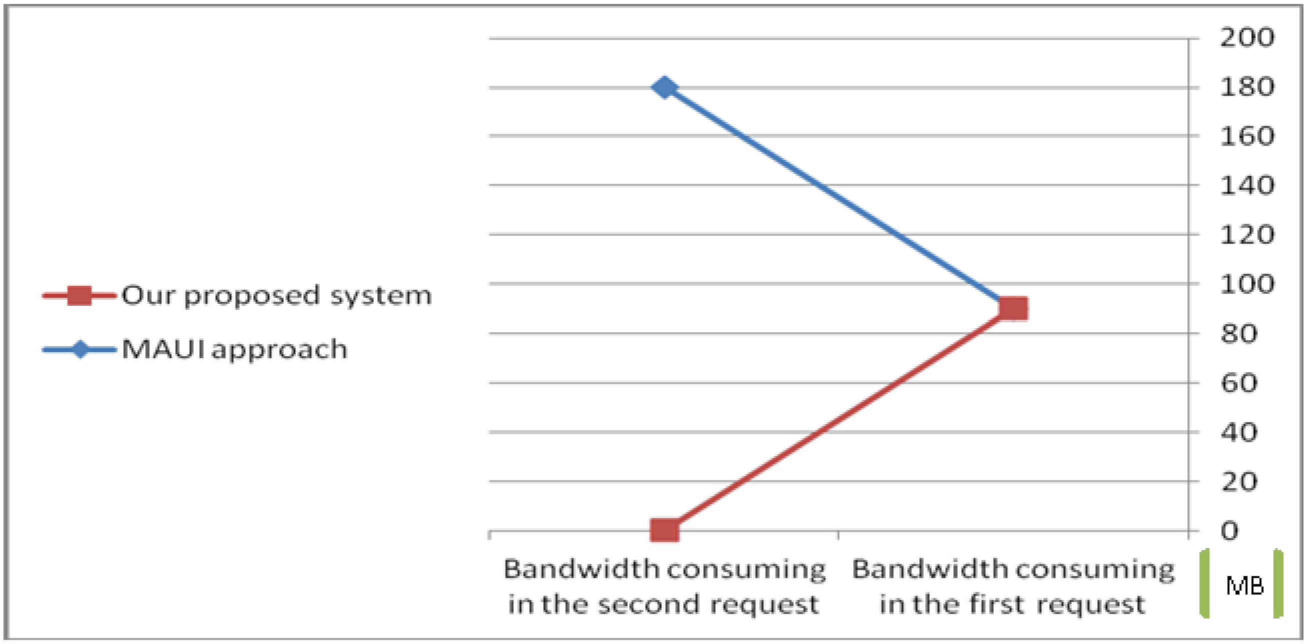
This contributed algorithm proofed its performance in reducing the consumed bandwidths from the internet, which is limited and very small especially for mobile phones. Moreover, it enhances the download speed of the files in the same LAN.

When the proposed algorithm was compared with previous similar work such as MAUI approach, which was explained previously in related work chapter, it was obtained that the obtained results enhanced the speed and consumed BW, several times that the previous similar works. The below table shows the comparison when applying the same file, mobile phone and Internet speed on both approaches.

	File size	**Bandwidth consuming in the first request**	**Bandwidth consuming in the second request**
MAUI approach	90MB	90MB	180MB
Our proposed system	90MB	90MB	0

Table 4-5: MAUI & the proposed system comparison

The table above showed a comparison between our proposed work and the MAUI approach (Eduardo & Aruna, 2010), this comparison concern with the bandwidth consuming when make two requests of a such file (90 MB), as we explained previously, the first request in our approach sent to the cloud but the second request to the same file sent to the peers devices, so no bandwidth was consumed. However, the MAUI approach consumed a bandwidth in each file request from the cloud and this bandwidth very small compared with our system (note the file size), but in our system the second request to the same file will be with no bandwidth consumption.



Our proposed system
MAUI approach
200
180
160
140
120
100
80
60
40
20
0
Bandwidth consuming in the second request
Bandwidth consuming in the first request
MB

Chapter Five

Conclusion

and

Future work

5.1 Results of Comparison between Cloud-based and WIFI-based Downloads

The aim of this experiment is to evaluate the effectiveness of using the proposed scheme of downloading a file from a source smart phone to a destination smart phone over the WIFI network without going through the internet, and to compare the results with cloud download for the same file.

The metrics of comparison are bandwidth consumption in MB and download time is units of time (seconds or minutes). A data file of 90 MB in size is used in the experiment, for download in both the cloud and the wifi environments. Three different modern smart phones are used: STC, S4 and Nexus 7, they share the Andriod operating system.

The experimental results are shown in tables 4-2, 4-3 and 4-4.

Table 4-2 shows the results of downloading directly from the cloud, using the three smart phones. The bandwidth is 90 MB for the three smart phones, but there is a minor difference in download times due to differences in smart phones specifications. The noticeable result is that the download time, which has a minimum of 6 minutes in a 2 mb/s download speed connections, and 12 mb/s in a 1 mb/s connection.

File Destination Mobile	Download Time	Bandwidth Consumption	Internet Download Speed
S4	17 minutes	90 MB	1 mb/s
Nexus 7	15 minutes	90 MB	1 mb/s
HTC	19 minutes	90 MB	1 mb/s
S4	7 minutes	90 MB	2 mb/s
Nexus 7	6 minutes	90 MB	2 mb/s
HTC	8 minutes	90 MB	2 mb/s

Table 4-3 shows the download time among the three smart phones. The band width is not included as this is a case of zero-bandwidth consumption. Each smart phone is used as a source for download in one situation, and as a destination for the download in another. The average download time is xx seconds, about one tenth of the minimum download time of the 2 mb/s cloud download speed.

File Destination Mobile	File Source Mobile	Download Time
S4	Nexus 7	45 s
	HTC	45 s
Nexus 7	S4	40 s
	HTC	40 s
HTC	S4	45 s
	Nexus 7	45 s

Table 4-4 shows the download time among the three smart phones using the Bluetooth network. The average download time is much higher than the equivalent average download time in the WIFI network, and it is close to the cloud download time, but again it is zero-bandwidth consumption.

File Destination Mobile	File Source Mobile	Download Time
S4	Nexus 7	12 m
	HTC	12 m
Nexus 7	S4	10 m
	HTC	10 m
HTC	S4	13 m
	Nexus 7	13 m

5.2 Conclusion:

All the previous sections explained this work; in this section, here conclude that application as follow: this proposed a such application consisted many devices, each device that connected in the same local network has its own database file so that the privacy and security can be maintained since no device has access to the other's database files. When the file can be found in many different name and we can't make the device search the stream of binaries for each file due to the limitations of computation power of smart devices. But if the file is not exists on any device the device who requested the file has to download it from the cloud, in this case the idea will be useless because the device will consume the bandwidth. In our application many available ways to be used for sending the files between users without consuming the bandwidth such as; Bluetooth (P2P), infrared (P2P), Wi-Fi direct and LAN network.

5.3 Future Work:

As we explained in the previous chapter , we defined the motivation to proposed our application, also we defined the methodology of our application, may be the weakness of our work is the security decreasing because the third party (the cloud) and the low privacy because the data sharing between devices . So in my own opinion and as a future work we may be enhance an application that maintain the bandwidth consumption beside the security enhancement may be by data encryption or any other ways. And also may suggest to apply our application using different devices that don't support android operating system like iphone and Nokia

References:

Almrot.E, Andersson.S, (2013). Astudy of the advantages &disadvantages of mobile cloud computing versus native enviroment, *Bachelor Thesis In Software Engineering*.

Andrews.T, (2006)."Functional versus Non-Functional Requirements and testing".

Constandache.I, Bao.X, Azizyan.M,Choudhury.R.R.(2010) human localization using mobile phones, *in: Proceedings of the Sixteenth Annual International Conference on Mobile Computing and Networking*, MobiCom'10, ACM, New York, NY, USA, 2010, pp. 149–160.

Cuervoy.E, Balasubramanianz.A, Wolmanx.D.C, Saroiux.S, Chandrax.R, Bahlx.P. (2010), MAUI: Making Smartphones Last Longer with Code Offload, *Copyright 2010* ACM 978-1-60558-985-5/10/06.

Debnath.S.K, Ahmed.F, and Islam.N,(2010). "Performance Evaluation of Unicast and Broadcast Mobile Ad-hoc Networks Routing Protocols" *,International Journal of Computer Science and Information Security*, Vol. 7, No. 1, 2010.

Dinh, H. T., Lee, C., Niyato, D. and Wang, P. (2013). "A survey of mobile cloud computing: architecture, applications, and approaches". *Wirel. Commun. Mob. Comput*., 13: 1587–1611. doi: 10.1002/wcm.

Dinhthaihoang, (2012)" Describe general architecture of Mobile Cloud Computing." , *Network Authentication, Authorization, and Accounting*, 27 February(2012).

Fernando.N, Loke.S.W,Rahayu.W. (2013, January). Mobile cloud computing*: A survey, Future Generation Computer Systems,* Volume 29, Issue 1, Pages 84-106, ISSN 0167-739X.

Fesehaye, D., Gao, Y., Nahrstedt, K., & Wang, G. (2012, September). Impact of Cloudlets on Interactive Mobile Cloud Applications. In Enterprise Distributed Object Computing Conference (EDOC), *2012 IEEE 16th International* (pp. 123-132). IEEE.

Flinn.J, Park.S, Satyanarayanan.M,(2010) Balancing performance, energy, and quality in pervasive computing, *in: Proceedings of the 22nd International Conference on Distributed Computing Systems*, 2002, IEEE, 2002, pp. 217–226.

Ge.F, Lin.H, Khajeh.A, Chiang.C.J, Eltawil.A.M, Bostian.C.W, Feng.W, and Chadha.R, “Cognitive Radio Rides on the Cloud,” *in Military Communications Conference (MILCOM), pp*. 1448, January 2011.

Gopikrishnan .B, Mani. Mr. A. (2012). Desktop Solution for Mobile Environment Using Mobile Cloud Computing, *International Journal of Advanced Research in Computer and Communication Engineering* ,Vol. 1, Issue 1, March 2012, ISSN 2278 – 1021.

Jeong.S, Zhang.X, Kunjithapatham.A, and Gibbs.S, (2010), "Towards an elastic application model for augmenting computing capabilities of mobile platforms," *Mobile Wireless Middleware, Operating Systems, and Applications*, pp. 161–174, 2010.

Jin, X., & Kwok, Y. K. (2010, December). Cloud assisted P2P media streaming for bandwidth constrained mobile subscribers. *In Parallel and Distributed Systems* (ICPADS), 2010 IEEE 16th International Conference on (pp. 800-805). IEEE.

Jin.X and Kwok.Y.K, (2011)."Cloud Assisted P2P Media Streaming for Bandwidth Constrained Mobile Subscribers," *in Proceedings of the 16th IEEE International Conference on Parallel and Distributed Systems* (ICPADS), pp. 800, January 2011.

Joy.P.T & Jacob.K.P,(2013)" Cooperative Caching Framework for Mobile Cloud Computing" *Global Journal of Computer Science and Technology Network, Web & Security* , Volume 13 Issue 8 Version 1.0 Year 2013.

Jung.E, Wang.Y, Prilepov.I, Maker.F,Liu.X, and Akella.V. (2010) User-profile-driven collaborative bandwidth sharing on mobile phones,"in Proceedings of the 1st ACM

Workshop on Mobile Cloud Computing & Services: Social Networks and Beyond (MCS), no. 2, 2010.

Kemp.R, Palmer.N, Kielmann.T, Bal.H, (2011), Cuckoo: A Computation Offloading Framework for Smartphones*, Second International ICST Conference, MobiCASE 2010, Santa Clara,* CA, USA, October 25-28, 2010, Revised Selected Papers.

Ketan B. Parmar, Nalinbhai N. Jani, Pranav S. Shrivastav and Mitesh H. Patel,(2013). "Mobile Grid Computing: Facts or Fantasy?", *International Journal of Multidisciplinary Sciences and Engineering*, vol. 4, No.1, January 2013.

Klein, A., Mannweiler, C., Schneider, J., & Schotten, H. D. (2010, May). Access schemes for mobile cloud computing. *In Mobile Data Management (MDM), 2010 Eleventh International Conference on* (pp. 387-392). IEEE.

Kosta.S,Aucinas.A, Hui.P, Mortier.R, Zhang.X,(2011). ThinkAir Dynamic resource allocation and parallel execution in cloud for mobile code offloading.

Kumar.K and Lu.Y,(2010) " CLOUD COMPUTING FOR MOBILE USERS: CAN OFFLOADING COMPUTATION SAVE ENERGY?", 0018-9162/10/$26.00 © 2010 *IEEE.*

Lu.Y, Li.S, and Shen.H, (2011), Virtualized Screen: A Third Element for Cloud_Mobile Convergence, 1070-986X/11/$26.00 _c *2011 IEEE*.

March.V, Gu.Y, Leonardi.E, Goh.G, Kirchberg.M, Lee.B.S.(2011). Cloud: Towards a New Paradigm of Rich Mobile Applications, *Proc. MobiWis, 2011*.

Microsoft Corporation, " Microsoft Application Architecture Guide, 2nd Edition", © 2009 Microsoft Corporation. All rights reserved.

Permalink to computer algorithm , "Computer Algorithms: Rabin-Karp String Searching", April 2, 2012.pp. 14-23.

Prasad.M.R, Gyani.J, Murti. P.R.K. (2012). Mobile Cloud Computing: Implications and Challenges. *Journal of Information Engineering and Applications www.iiste.org ISSN* 2224-5782 (print) ISSN 2225-0506 (online) Vol 2, No.7.

Qi.H,Gani.A,(2012)" Research on Mobile Cloud Computing: Review Trend and Perspectives", 2012.

Sanaei.Z, Abolfazli.S, Gani.A, Shiraz.M.(2012). "SAMI: Service-Based Arbitrated Multi-Tier Infrastructure for Mobile Cloud Computing," *1st IEEE International Conference on Communications in China Workshops* (ICCC), PP. 14 - 19, 27 June 2012 .

Satyanarayanan.M, Bahl.P.V, Caceres.R, Davies.N.(2009). "The Case for VM -based Cloudlets in Mobile Computing" Carnegie Mellon University, *Microsoft Research, AT&T Research, Lancaster University. October-December* 2009 (vol. 8 no. 4)

Schüring.M,(2011) " Mobile cloud computing – open issues and solutions", 2011, *University of Twente* ,P.O. Box 217, 7500AE Enschede .

Yang. X, Pan .T, Shen. J.(2010). On 3G mobile e-commerce platform based on cloud computing, Computer Science and Information Technology College, Zhejiang Wanli University, Ningbo, China 2010; 198–201.

Yang.F, Qian.Z, .Chen.X , Beschastnikh.I, Zhuang.L, Zhou.L and Shen.J.(2012) "Sonora: A Platform for Continuous Mobile-Cloud Computing" *Microsoft Research Asia, University of Washington.*

Yucek.T and Arslan.H, "A survey of spectrum sensing algorithms for cognitive radio applications," *in IEEE Communications Surveys & Tutorials*, pp. 116 - 130, March 2009.

Zohar.E, Cidon.I, Mokryn.O. (2011). The Power of Prediction: Cloud Bandwidth and Cost Reduction. *ACM SIGCOMM Computer Communication Review - SIGCOMM '11*. Volume 41 Issue 4, August 2011. Pages 86-97. ACM New York, NY, USA.

APPENDIX 1

Evaluation Code

In order to prove the practical side of thesis, we are going to build an application that uses the proposed methodology and algorithm. However, the application will be built using C# for windows phone application.

1- Methodology:

In this project, the idea we are using to decrees the consumption of bandwidth when downloading files is to share the files among the users of the network so that users can send and receive files from each other's instead of downloading files from clouds. However, to achieve the goad of this idea the users should be connected together through peer to peer network or using a local area connection so that the bandwidth won't be consumed as the following:

Assume we have 3 devices; device A, B, and C. Device A wants to download file x.
Device A broadcast a message to device B and C at the same time.
Device B and C receive the message.
Device B and C searching their own installed applications list using hashing string search algorithm. However let's assume that Device C found the application so the user of the device C decides whether if he wants to send the requested file for device A or not.
If devices B and C couldn't find the requested file they send a message through the proposed application to device a telling they didn't find the requested file so device A can download it from the cloud.
Searching the file name →
Using hashing strings search algorithm to make sure if the file is exists in the device or not with many possibilities.(i.e. file aaa may be exist with multiple possibilities like aAa, AAA, Aaa).

```
Function Search (string s [1...n], string sub [1...m])
   For I from 1 to n-m+1
         For j from 1 to m
               If s [i+j-1] ≠ sub[j]
                 Jump to next iteration of outer loop
                 Return I
                 Return not found
         Else
              Return Sub[j]    →    (the file found so user decides whether he wants
to send the file or not).
```

2- Interface:

The interface of our application will consists of the following:

- Network Information:
  This part will provide information about the network that the device is connected to; also it will contain the name and ip address of the peers on the network.
- My files:
  Provides a list of the files stored in the device and gives ability to remove the unwanted files.
- Request a file: usage this part the application broadcasts a request to the devices connected to the same network to send the wanted file.
- Received requests:
  This part gives the user to explore and answer the received request.

- **Network information:**

1- Design:

```
<phone:PhoneApplicationPage
    x:Class="Sharing_Mnager_pro.netinformation"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:phone="clr-namespace:Microsoft.Phone.Controls;assembly=Microsoft.Phone"
    xmlns:shell="clr-namespace:Microsoft.Phone.Shell;assembly=Microsoft.Phone"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    FontFamily="{StaticResource PhoneFontFamilyNormal}"
    FontSize="{StaticResource PhoneFontSizeNormal}"
    Foreground="{StaticResource PhoneForegroundBrush}"
    SupportedOrientations="Portrait" Orientation="Portrait"
    mc:Ignorable="d" d:DesignHeight="768" d:DesignWidth="480"
    shell:SystemTray.IsVisible="True">

    <!--LayoutRoot is the root grid where all page content is placed-->
    <Grid x:Name="LayoutRoot" Background="Transparent">
        <Grid.RowDefinitions>
            <RowDefinition Height="Auto"/>
            <RowDefinition Height="*"/>
        </Grid.RowDefinitions>

        <!--TitlePanel contains the name of the application and page title-->
        <StackPanel x:Name="TitlePanel" Grid.Row="0" Margin="12,17,0,28">
            <TextBlock x:Name="ApplicationTitle" Text="Network Change Detector" Style="{StaticResource PhoneTextNormalStyle}"/>
```

```
        </StackPanel>

        <!--ContentPanel - place additional content here-->
        <Grid x:Name="ContentPanel" Grid.Row="1" Margin="12,0,12,0">
            <Grid.RowDefinitions>
                <RowDefinition Height="Auto"/>
                <RowDefinition Height="Auto"/>
                <RowDefinition Height="Auto"/>
                <RowDefinition Height="*"/>
                <RowDefinition Height="Auto"/>
                <RowDefinition Height="Auto"/>
                <RowDefinition Height="Auto"/>
            </Grid.RowDefinitions>
            <TextBlock Grid.Row="0" Text="Available Network Interfaces" FontSize="{StaticResource PhoneFontSizeLarge}"/>
            <ListBox Grid.Row="1" x:Name="lbNetworkInterfaces" ItemsSource="{Binding}" Background="LightGray" >
                <ListBox.ItemTemplate>
                    <DataTemplate>
                        <TextBlock Text="{Binding}" Margin="5,5,0,5" Foreground="Black"/>
                    </DataTemplate>
                </ListBox.ItemTemplate>
            </ListBox>
            <TextBlock Grid.Row="2" Text="Events" FontSize="{StaticResource PhoneFontSizeLarge}"/>
            <ScrollViewer Grid.Row="3" Background="LightGray" BorderThickness="1">
                <ListBox x:Name="lbNetworkChanges" ItemsSource="{Binding}">
                    <ListBox.ItemTemplate>
                        <DataTemplate>
                            <TextBlock Text="{Binding}" Foreground="Black"
                                    FontSize="{StaticResource PhoneFontSizeNormal}" TextWrapping="Wrap" Margin="5,10,0,10"  />
                        </DataTemplate>
                    </ListBox.ItemTemplate>
                </ListBox>
            </ScrollViewer>
            <TextBlock Grid.Row="4" Text="Networking Status" FontSize="{StaticResource PhoneFontSizeLarge}"/>
            <Grid Grid.Row="5" Background="LightGray" >
                <Grid.RowDefinitions>
                    <RowDefinition Height="Auto"/>
                    <RowDefinition Height="Auto"/>
                    <RowDefinition Height="Auto"/>
                </Grid.RowDefinitions>
                <Grid.ColumnDefinitions>
                    <ColumnDefinition Width="Auto"/>
                    <ColumnDefinition Width="*"/>
                </Grid.ColumnDefinitions>
                <TextBlock Grid.Row="0" Grid.Column="0" Text="IsNetworkAvailable" Foreground="Black" Margin="5,5,0,5" />
                <TextBlock Grid.Row="0" Grid.Column="1" x:Name="tbIsNetworkAvailable" HorizontalAlignment="Center" Foreground="Black" />

```

```
                <TextBlock Grid.Row="1" Grid.Column="0" Text="IsWiFiEnabled" Foreground="Black" Margin="5,5,0,5" />
                <TextBlock Grid.Row="1" Grid.Column="1" x:Name="tbIsWiFiEnabled" HorizontalAlignment="Center" Foreground="Black"/>

                <TextBlock Grid.Row="2" Grid.Column="0" Text="IsCellularDataEnabled" Foreground="Black" Margin="5,5,0,5" />
                <TextBlock Grid.Row="2" Grid.Column="1" x:Name="tbIsCellularDataEnabled" HorizontalAlignment="Center" Foreground="Black"/>
            </Grid>
            <Button Grid.Row="6" x:Name="btnChangeNetworkSettings" Content="Change Network Settings" Click="btnChangeNetworkSettings_Click"/>
        </Grid>
    </Grid>

    <!--Sample code showing usage of ApplicationBar-->
    <!--<phone:PhoneApplicationPage.ApplicationBar>
        <shell:ApplicationBar IsVisible="True" IsMenuEnabled="True">
            <shell:ApplicationBarIconButton IconUri="/Images/appbar_button1.png" Text="Button 1"/>
            <shell:ApplicationBarIconButton IconUri="/Images/appbar_button2.png" Text="Button 2"/>
            <shell:ApplicationBar.MenuItems>
                <shell:ApplicationBarMenuItem Text="MenuItem 1"/>
                <shell:ApplicationBarMenuItem Text="MenuItem 2"/>
            </shell:ApplicationBar.MenuItems>
        </shell:ApplicationBar>
    </phone:PhoneApplicationPage.ApplicationBar>-->

</phone:PhoneApplicationPage>

```

3- Code:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Animation;
using System.Windows.Shapes;
using Microsoft.Phone.Controls;
using System.Collections.ObjectModel;
using Microsoft.Phone.Net.NetworkInformation;
using Microsoft.Phone.Tasks;

namespace Sharing_Mnager_pro
```

```
{
    public partial class netinformation : PhoneApplicationPage //declaration of class that inherets the attributes of the phone applicatation page
    {
        public ObservableCollection<string> Changes { get; set; }// List of all changes detected while the app is running.


        public ObservableCollection<string> NetworkInterfaces { get; set; } // List of all currently available network interfaces


        public netinformation() // Constructor
        {
            InitializeComponent();

            Changes = new ObservableCollection<string>();  // Initialise the Changes list.


            lbNetworkChanges.DataContext = Changes; // Bind the ListBox to the Changes list

            NetworkInterfaces = new ObservableCollection<string>(); // initializing the object of ObservableCollection type
            lbNetworkInterfaces.DataContext = NetworkInterfaces;

            // Subscribe to the NetworkAvailabilityChanged event
            DeviceNetworkInformation.NetworkAvailabilityChanged += new EventHandler<NetworkNotificationEventArgs>(ChangeDetected); // declaring the event handler
            UpdateNetworkInterfaces();  // calling the method UpdateNetworkInterfaces
            UpdateNetworkStatus();  // calling the method UpdateNetworkStatus

        }


        void ChangeDetected(object sender, NetworkNotificationEventArgs e)// declaring method
        {
            string change = string.Empty;
            switch (e.NotificationType)
            {
                case NetworkNotificationType.InterfaceConnected:
                    change = "Connected to ";
                    break;
                case NetworkNotificationType.InterfaceDisconnected:
                    change = "Disconnected from ";
                    break;
                case NetworkNotificationType.CharacteristicUpdate:
                    change = "Characteristics changed for ";
                    break;
                default:
                    change = "Unknown change with ";
                    break;
```

```
            }

            string changeInformation = String.Format(" {0} {1} {2} ({3})",
                        DateTime.Now.ToString(), change, e.NetworkInterface.InterfaceName,
                        e.NetworkInterface.InterfaceType.ToString());

            // We are making UI updates, so make sure these happen on the UI thread.
            Dispatcher.BeginInvoke(() =>
            {
                Changes.Add(changeInformation);
                UpdateNetworkInterfaces();
                UpdateNetworkStatus();


            });
        }
        private void UpdateNetworkInterfaces()
        {

            NetworkInterfaces.Clear();
            NetworkInterfaceList networkInterfaceList = new NetworkInterfaceList();
            foreach (NetworkInterfaceInfo networkInterfaceInfo in networkInterfaceList)
            {
                NetworkInterfaces.Add(networkInterfaceInfo.InterfaceName);
            }
        }

        private void UpdateNetworkStatus()
        {
            tbIsCellularDataEnabled.Text = (DeviceNetworkInformation.IsCellularDataEnabled) ? "Yes" : "No";
            tbIsNetworkAvailable.Text = (DeviceNetworkInformation.IsNetworkAvailable) ? "Yes" : "No";
            tbIsWiFiEnabled.Text = (DeviceNetworkInformation.IsWiFiEnabled) ? "Yes" : "No";
        }
```