RESOURCE DISCOVERY IN MOBILE CLOUD COMPUTING: A CLUSTERING BASED APPROACH

Dissertation submitted to Jawaharlal Nehru University
in partial fulfillment of the requirements
for the award of the degree of

Master of Technology
in
Computer Science & Technology

By
Priyanka Athwani

Under the Supervision of
Prof. D. P. Vidyarthi

School of Computer and Systems Sciences
Jawaharlal Nehru University
New Delhi -110067, India
June-2015

जवाहरलाल नेहरु विश्वविद्यालय

SCHOOL OF COMPUTER AND SYSTEMS SCIENCES

JAWHARLAL NEHRU UNIVERSITY

Certificate

This is to certify that dissertation entitled **"Resource Discovery in Mobile Cloud Computing: A Clustering Based Approach"** is being submitted by **Ms. Priyanka Athwani** to the **School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi-110067, India**, in the partial fulfillment of the requirements for the award of the degree of **"Master of Technology"** in **"Computer Science & Technology"**. This work is carried out by herself in the School of Computer and Systems Sciences under the supervision of **Prof. D. P. Vidyarthi**. The matter personified in the dissertation has not been submitted for the award of any other degree or diploma.

Dean

Prof. C. P. Katti
School of Computer and Systems Sciences
Jawaharlal Nehru University
New Delhi-110067

Dean
School of Computer & Systems Sciences
Jawaharlal Nehru University
New Delhi-110067

Supervisor 18.6.15

Prof. D. P. Vidyarthi
School of Computer and Systems Sciences
Jawaharlal Nehru University
New Delhi-110067

जवाहरलाल नेहरु विश्वविद्यालय

SCHOOL OF COMPUTER AND SYSTEMS SCIENCES

JAWAHARLAL NEHRU UNIVERSITY

Declaration

I hereby declare that the dissertation work entitled **"Resource Discovery in Mobile Cloud Computing: A Clustering Based Approach"** in partial fulfillment of the requirements for the award of degree of **"Master of Technology"** in **"Computer Science & Technology"** and submitted to the School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi-110067, India is the authentic record of my own work carried out during the time of Master of Technology under the supervision of **Prof. D. P. Vidyarthi**. This dissertation comprises only my own work. This dissertation is less than 14,000 words in length, exclusive tables, figures and references. The matter personified in the dissertation has not been submitted for the award of any other degree or diploma.

Priyanka Athwani

Priyanka Athwani
M.Tech (2013-2015)
School of Computer and Systems Sciences
Jawaharlal Nehru University
New Delhi-110067
India

………Dedicated to my family

Acknowledgement

Though only my name appears on the cover of this dissertation, a great many people have contributed to its production. I owe my gratitude to all those people who have made this dissertation possible and because of whom my master's experience has been one that I will cherish forever.

My deepest gratitude is to my advisor, Prof. D. P. Vidyarthi. I have been amazingly fortunate to have an advisor who gave me the freedom to explore on my own and at the same time the guidance to recover when my steps faltered. He taught me how to question thoughts and express ideas. His selfless time and care were sometimes all that kept me going and finish this dissertation. I hope that one day I would become as good an advisor to my students as Prof. D. P. Vidyarthi has been to me.

I thank Prof. C. P. Katti, Dean, School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi, for the academic support and the facilities provided to carry out the research work at the center.

It's very important to have a nice social environment outside the lab. Jawaharlal Nehru University itself has taught me several good lessons and also has given me several good friends. In JNU, very rarely I have felt that I am staying away from my home. In this regard, I would like to thank my seniors (Achal, Shiv Prakash, Pawan, Isha, Sunil, Neetesh, Jyoti, Gaurav, Atul and Shrish), my friends (DN Jha, Bindu, Sudhakar, Shraddha, Deepti and Shikha) and many others whose name is not mentioned here for creating a wonderful social environment.

Finally, my family has supported and helped me along the course of this dissertation by giving encouragement and providing the moral and emotional support I needed to complete my thesis. To them, I am eternally grateful.

(Priyanka Athwani)

Abstract

Mobile Cloud Computing is an emerging technology in today's digital era which enables mobile users to exploit the services provided by the Cloud. It avails the advantages of both Mobile Computing and Cloud Computing and at the same time it leaves out the pitfalls of these technologies. In traditional Cloud Computing, only the resources provided by the stationary Cloud were borrowed but the cooperation based architecture of Mobile Cloud Computing opened the door to utilize resources of mobile devices of local network itself. In cooperation based architecture of Mobile Cloud Computing, mobile devices share resources such as CPU, memory, bandwidth and application etc. to perform some task.

Resources are needed to perform any task. Since there may be multiple mobile devices in cooperation based architecture serving as providers, it is necessary to find the appropriate devices. There are some issues for fulfilling this vision such as connectivity, offloading, resource discovery and heterogeneity etc. Among these issues, resource discovery is the most important issue. Mobile devices rely on battery power; therefore it is necessary that the employed resource discovery method must be energy efficient. The method should be delay sensitive as well because services are performed immediately.

Adaptive resource discovery in Mobile Cloud Computing is one of the existing works for resource discovery method. A centralized server measures the energy consumption in flooding mode and centralized mode and selects the most suitable mode for the next time slot for resource discovery purpose. Although this approach is good enough, it purely relies on the centralized server for selecting the best mode. It does not mention the delay required for performing resource discovery task. Both these methods; flooding or centralized, have their own demerits. For example, flooding incurs a huge traffic whereas centralized method has single point of failure along with the overhead to maintain consistent information to the server. Centralized method cannot be applied in the case of ad hoc network where the Internet connectivity is not available.

The proposed method is a scalable approach and based on the clustering of the mobile devices where the mobile devices are large in number in a network. In this method, though a centralized server is maintained but is just used to store the identification of the cluster heads. Thus, the information stored on the server is very less and therefore a local mobile node can play the role of the server. For reliability purpose, this information can be replicated to another server also. There will almost be negligible overhead in doing this, because the information to be replicated is very less.

The performance of the proposed method depends heavily on the parameter on which the clustering method is applied. Keeping this in mind, clustering has been done based on the parameters such as battery power, distance, mobility and signal strength. Battery power is the most crucial parameter for the longevity of the mobile devices. Mobility, distance and signal strength parameters are responsible for maintaining the stable clusters.

Performance evaluation, of the proposed model, is done by the simulation. The experiments have been carried out on large number of requests and other varying mobility parameters. A comparative analysis is also done for the energy consumption and delay requirement of various other contemporary methods which depicts the effectiveness of the proposed method.

Table of Contents

List of Figures

Chapter1

Introduction

The idea of Cloud Computing is not new. It was already seeded fifty decades ago. In 1969, Leonard Kleinrock [1], one of the chief scientists of the original Advanced Research Projects Agency Network (ARPANET), said: "*As of now, computer networks are still in their infancy, but as they grow up and become sophisticated, we will probably see the spread of 'computer utilities' which, like present electric and telephone utilities, will service individual homes and offices across the country.*" This concept has become a reality now and the technology which is actualizing this thought is termed as Cloud Computing.

Cloud Computing enables the users (individual, group or public) to use the services or resources provided by Cloud on pay per use basis, anytime and from anywhere. The only requirement is that they must have some sort of medium, most probably Internet to access these services. Many agencies have defined Cloud Computing considering their different aspects but most appropriate definition, given by National Institute of Standards and Technology (NIST) [2], is as below.

"Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g. networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction".

Many technologies such as Parallel and Distributed Computing, Cluster Computing, Grid Computing, Utility Computing and Virtualization have contributed to foster the Cloud Computing. Another definition which defines '*Cloud*' in the most relevant way is given by Buyya et al. [3] as follows.

"A Cloud is a type of parallel and distributed system consisting of a collection of interconnected and virtualized computers that are dynamically provisioned and

presented as one or more unified computing resources based on service-level agreements established through negotiation between the service provider and consumers."

Most of us, who are using the Internet, are using the Cloud services knowingly or otherwise. Using any email service, social networking sites or any online application are example of the Cloud services. Google Documents, Flickr, Windows Azure, Hadoop and Amazon EC2 are some specific and popular examples of Cloud services. The following keynotes validate that the Cloud is beneficial to service providers as well as users.

From User's Perspectives-

- A startup company cannot afford to invest in huge infrastructure.
- Users want to access their data any time ubiquitously.
- Many companies want to focus on their application logic. They don't want to bother about the maintenance and scalability issues.
- Some companies want huge infrastructure but for a very short period of time.

Service providers are having lot of resources/services. They provide these services to users and ultimately make money from them directly or indirectly.

From Service Provider's Perspectives-

- Service providers may have surplus infrastructure, so they want to rent it.
- They may be having infrastructure and middleware also, so they are able to host applications.
- Some providers have infrastructure and application services to provide.

1.1 Essential Characteristics of Cloud Computing

Some essential characteristics of Cloud Computing are as follows [2].

1.1.1 On Demand Self Service

Whenever a user needs services, she can access those services automatically without any human intervention.

1.1.2 Broad Network Access

Cloud services are available over a large network. Therefore, these services can be accessed through heterogeneous platforms such as PC, laptops, mobile phones etc.

1.1.3 Resource Pooling

Computing resources like CPU processing power, storage, memory etc. are pooled to handle multiple consumers' requests using a multi-tenant model. These resources are allocated and released dynamically according to consumer's demand. User has no idea about the exact location of the assigned resources. Sometimes, it is possible that a user can mention location such as country, state or a particular data center if the provider agrees on it.

1.1.4 Rapid Elasticity

Cloud resources are assigned rapidly as the need of resources increases. Similarly, resources are released quickly as soon as the requirement diminishes.

1.1.5 Measured Service

Cloud systems use a metering capability for control and optimization of resources. Transparency is provided to both consumers and service providers by the monitoring, controlling and reporting mechanism.

1.2 Cloud Service Models

The services, provided by Cloud, have been categorized in three models as shown in figure 1.1. Briefly, these are as follows [2].

1.2.1 Software as a Service (SaaS)

This model enables the consumers that they can access Cloud applications without installing them on consumers' site. This way, consumers become free from maintenance issues. They need not to pay the cost associated with software licensing. Applications can be accessed by any device with a thin client interface (web browser etc.) or any program interface. Google Documents, Facebook, Salesforce are some examples of SaaS.

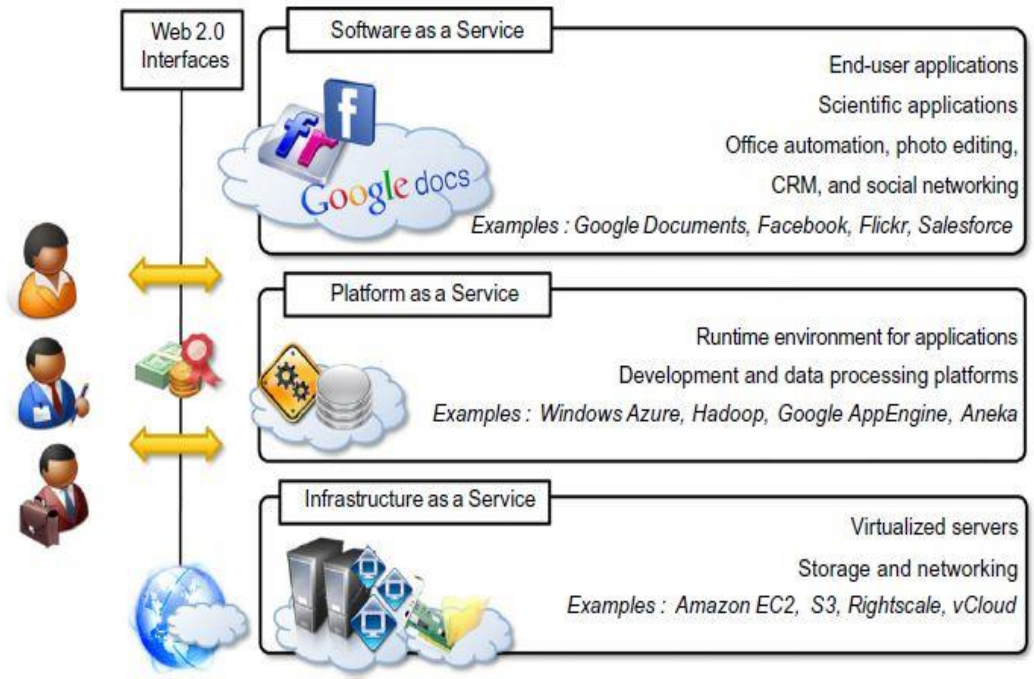

Figure 1.1: Cloud Service Models [4]

1.2.2 Platform as a Service (PaaS)

In this model, Cloud provides development tools, programming languages, libraries and services so that the users can deploy their own applications on the Cloud without worrying about the underlying infrastructure. Users have their own control on the deployed application and configuration settings but the control and management issues of infrastructure such as storage, network, operating system, memory requirements, all are handled by the provider. Hadoop, Windows Azure, Google app engine are leading PaaS providers.

1.2.3 Infrastructure as a Service (IaaS)

IaaS model provides users the Cloud infrastructure such as storage, processing and network. This infrastructure may be physical (hardware) or logical (Virtual machines). Consumers can use these resources according to their requirement, for example to deploy applications, to run a software, for storage purpose etc. Still user has control over operating system, storage, applications and some network components. Amazon EC2 and SSS are some major providers of IaaS.

1.3 Cloud Deployment Models

Depending on the design, four types of Cloud deployment models has been suggested.

1.3.1 Private Cloud

This type of Cloud is conceived, designed and used to serve explicitly for a particular organization. The responsibility of managing and operating the Cloud is solely the responsibility of the organization which itself or with the help of third party maintains it.

1.3.2 Community Cloud

When some consumers of same community (same objectives) from different organizations share the resources of Cloud, it is termed as Community Cloud. Here, same community implies to groups that have similar concerns.

1.3.3 Public Cloud

This type of Cloud is used to serve general public. Cloud infrastructure is deployed in the premises of the Cloud provider.

1.3.4 Hybrid Cloud

Hybrid Cloud is a combination of the above mentioned types of Cloud i.e. private, community or public.

1.4 Mobile Cloud Computing (MCC)

In recent years, use of mobile devices has increased rapidly. This has fulfilled the dream of "*Services at the fingertips anywhere, anytime*". But some constraints of mobile devices e.g. limited battery life, scarcity of resources (computation power, storage, bandwidth etc.) etc. have been a bottleneck in achieving this. Researchers thought that emergence of Cloud Computing can be the most prominent solution to address these problems. The amalgamation of Cloud computing with mobile industry became as icing the cake to meet the need of twenty first century digital mission. According to the research of ABI (Allied Business Intelligence), Cloud computing services will drive the revenue of $3.6 billion in mobile enterprise application by 2019 [5].

Mobile Cloud Computing (MCC) is made of two most potential technologies of IT industry: Mobile Computing and Cloud Computing. MCC can be defined as leveraging the cloud computing services into the mobile devices (Smartphones, laptops etc.) through wireless networks. This states that all the computation and storage will be outside the mobile devices [6]. This brings the power to not only the smartphones but also to a wide variety of mobile users. Some of the existing applications of MCC are Google Map, Gmail and Cisco's WebEx.

1.5 MCC Architecture

MCC can be viewed in two different ways which define its architecture [7]. These are as follows.

1.5.1 Agent Client Architecture

This is the traditional view of MCC which says that mobile devices only use the services provided by the Cloud to enhance their functionalities. Agent Client Architecture of MCC is shown in figure 1.2.

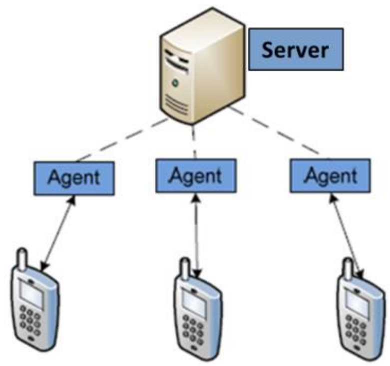

Figure 1.2: Agent Client Architecture of MCC [7]

Mobile devices are normally having the least number of resources, so its processing and storage can be offloaded and performed on a centralized server. Central server is responsible for overall resource scheduling and management.

Advantage

'Application independence on mobile devices and OS' is the main benefit gained by this architecture.

Disadvantage

Longer delays are possible when the cloud is at a distant location from the mobile devices.

1.5.2 Cooperation Based Architecture

Today's smartphones and more powerful devices have given us a direction of thinking that why these devices itself can't be a part of the Cloud? In this architecture, mobile devices themselves share resources to form a Cloud through wireless communication. This new paradigm is also termed as "*Fog Computing*" [8]. Architecture of MCC, based on cooperation, is shown in figure 1.3.

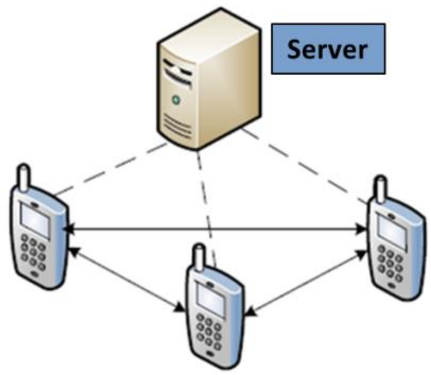

Figure 1.3: Cooperation Based Architecture of MCC [7]

Advantages

Longer delays can be reduced to a great extent using local proximity. Also, proper utilization of mobile device resources is ensured.

Disadvantage

If there is a violation of meeting the needs of users, one is bound to switch to Agent-Client Architecture.

1.6 Advantages of MCC

MCC has various advantages. Some of these are mentioned as below [9, 10].

1.6.1 Reduced Power Consumption/Improved Battery Lifetime

Battery is the most crucial component in any mobile device. By offloading heavy computational loads to the cloud, it is possible to reduce the load on the mobile devices. Hence, lesser energy is consumed to compute the results.

1.6.2 Huge Processing Power and Storage

Mobile devices have limited memory and computation power because of its size and cost. By utilizing the remote resources of Cloud, applications have no bar of processing and storage limitation.

1.6.3 Easy Sharing

Cloud provides us a platform where huge data sets can be stored. This allows us the ease of sharing our content e.g. images can be uploaded and processed in the Cloud instantly after capturing and users can share these images whenever they want it and from anywhere.

1.6.4 Achieving Reliability

Transferring the job of running applications or storage to the Cloud provides high reliability because data is backed-up on multiple computers in the Cloud. MCC also facilitates mobile users by giving the provision of virus scanning, authentication and malicious code detection services which may take a long time when performed on the device itself.

1.6.5 Location Free Access

MCC enables the users to access their data without having bound to be in a particular location.

Apart from these, all other advantages of Cloud computing are also inherited by MCC which are extended further.

1.6.6 Scalability

Cloud computing can handle the varied requirements of users in a flexible manner whenever the needs grow or shrink.

1.6.7 Multi-tenancy

Multi tenancy in Cloud saves the development and maintenance cost because many users are provided the access on the same software instance.

1.6.8 Dynamic Resource Provisioning

This ensures that users' demands are satisfied without any wastage of resources because resources are allocated and released on demand without any pre-reservation.

1.6.9 Ease of integration

Different kinds of services from various service providers can be integrated easily to serve the user demands in a better manner.

MCC is also useful to overcome the disadvantages of Cloud computing, as mentioned below.

- In Cloud computing, the cause for a longer delay is server location at a distant place. But in MCC, this delay can be reduced using cloudlet [11] and nearest Cloud of any mobile device.

Figure 1.4: Cloudlet Serving Different Devices [11]

Cloudlet is nothing but a cluster of computers or simply a resource rich computer which is always connected to Internet and used to serve mobile devices. It is like "*data center in a box*" bringing the Cloud closer to mobile users. Cloudlet can be set up like Wi-Fi access points on distinct locations such as business premises or a coffee shop. A general concept of Cloudlet is shown in the figure 1.4.

- In Cloud Computing, connectivity to Internet is must but in MCC, one can also use Bluetooth to connect with mobile devices in case of network failure.

1.7 Applications of MCC

Although there are numerous applications of MCC, some typical applications are listed below [12-14].

1.7.1 Image Processing

One can use MCC to recognize the text written in some other language by taking the image of that text. One can request a Cloud to process that image and to convert it in his/her own language.

1.7.2 Mathematical Tools

Complex mathematical computation can be offloaded to cloud to get the results quickly.

1.7.3 Antivirus Applications

Viruses, malwares, spywares are major threats which are harmful for anyone's data. To run antivirus software on a mobile device can be a very time consuming task. It can be done in Cloud by running antivirus software on the mobile clone which is pre synchronized with mobile.

1.7.4 Natural Language Processing

A foreign traveler can communicate to the local citizens by converting its native language to a desired one.

1.7.5 Searching a file

In today's world of smartphones, storage may be too large. Because of this, searching a file may take huge time. To improve the search time, one can perform searching process on mobile clone in the Cloud. Voice-based searching is also helpful for visually challenged people.

1.7.6 Gaming Applications

Some games e.g. chess require quick response time and they have small data sets. So its computation can be offloaded to Cloud.

1.7.7 Crowd Computing

Videos/photos from multiple mobile phones can be collected and merged to form a single video to view different angles of a sight. For example, in disaster management local citizens can be asked to take different pictures of destructed area and then to send it on a central local server. Rescue team can create a map from these pictures and accordingly use it for rescue operation. Another example of this can be "*to find a lost child*". In a crowded area, if a child is lost, an SMS can be sent to mobile phone users in a suspicious place to take pictures of their surrounding area and to send it on a local server. Police, along with the child's parents, can see those images and find the lost child's image among them.

1.7.8 M-Commerce (Mobile Commerce)

It is a business model used for commerce using mobile devices. M-commerce applications include mobile payments, mobile ticketing and mobile messaging. Various challenges in M-commerce (security and low bandwidth) could be addressed by the integration of Cloud computing.

1.7.9 M-Learning (Mobile Learning)

Learner can access remote resources of learning and teacher can also establish better communication with students. Teacher can also check the knowledge level of students and answer their doubts in a timely manner. All these are possible only by MCC because without MCC, issues like battery constraint and resource scarcity will surely create obstacle on the way to this application.

1.7.10 M-Healthcare (Mobile Healthcare)

MCC really can be a boon for monitoring health parameters such as blood pressure, level of alcohol and pulse rate. Health emergency system can be activated immediately and accordingly actions can be taken regarding patient's health.

1.8 Issues in Mobile Cloud

There are several issues related to MCC [14, 15] as listed below.

1.8.1 Limited Bandwidth

Since MCC operates in wireless network, it suffers from the bandwidth limitation as compared to the traditional wired network.

1.8.2 Resource Discovery

In MCC, there can be multiple mobile devices which will be serving as Clouds. Automatic resource discovery is necessary to find the required resources, preferably nearby to the current location.

1.8.3 Connectivity and Availability

Mobile devices are not always connected to the Cloud because of cost factor, traffic congestion and network related issues. This will result in the unavailability of Cloud services.

1.8.4 Heterogeneity

Mobile devices operate in a highly heterogeneous network e.g. 3G, WLAN etc. OS platforms and other technologies also differ to a great extent. In such an environment, it is very difficult to provide all the desired features of MCC such as energy efficiency, continuous connectivity etc.

1.8.5 Offloading

It has been shown that offloading a small computation may result in more energy consumption as compared to local execution. Because of this, it is very difficult to determine whether to offload the computation on Cloud or not. If yes, then how much portion should be offloaded to save the energy?

1.8.6 Security and Privacy

Same Cloud can be used by multiple mobile users. In such an environment, how to ensure that users' data is secured.

1.8.7 Reliability

Clouds should be able to continue providing resources despite of any kind of failure.

1.8.8 Pricing

In MCC, multiple entities and users are responsible for executing a task. Therefore, the problem of setting a fair pricing scheme is definitely an issue.

Chapter 2

Resource Discovery in MCC: The Problem

As discussed in chapter 1, there are numbers of issues in mobile cloud computing. Resource discovery is one of the important issues, which has been addressed in this dissertation work. The details are as follows.

In the cooperation based MCC architecture, multiple mobile devices will be serving as Clouds. Therefore, to accomplish a task using the services of Cloud, it is necessary to find out the required devices with the desired resources. This process is called resource discovery. Once the resources are discovered, resource allocation is needed to perform the task. A situation describing the need of resource discovery method is shown in figure 2.1.

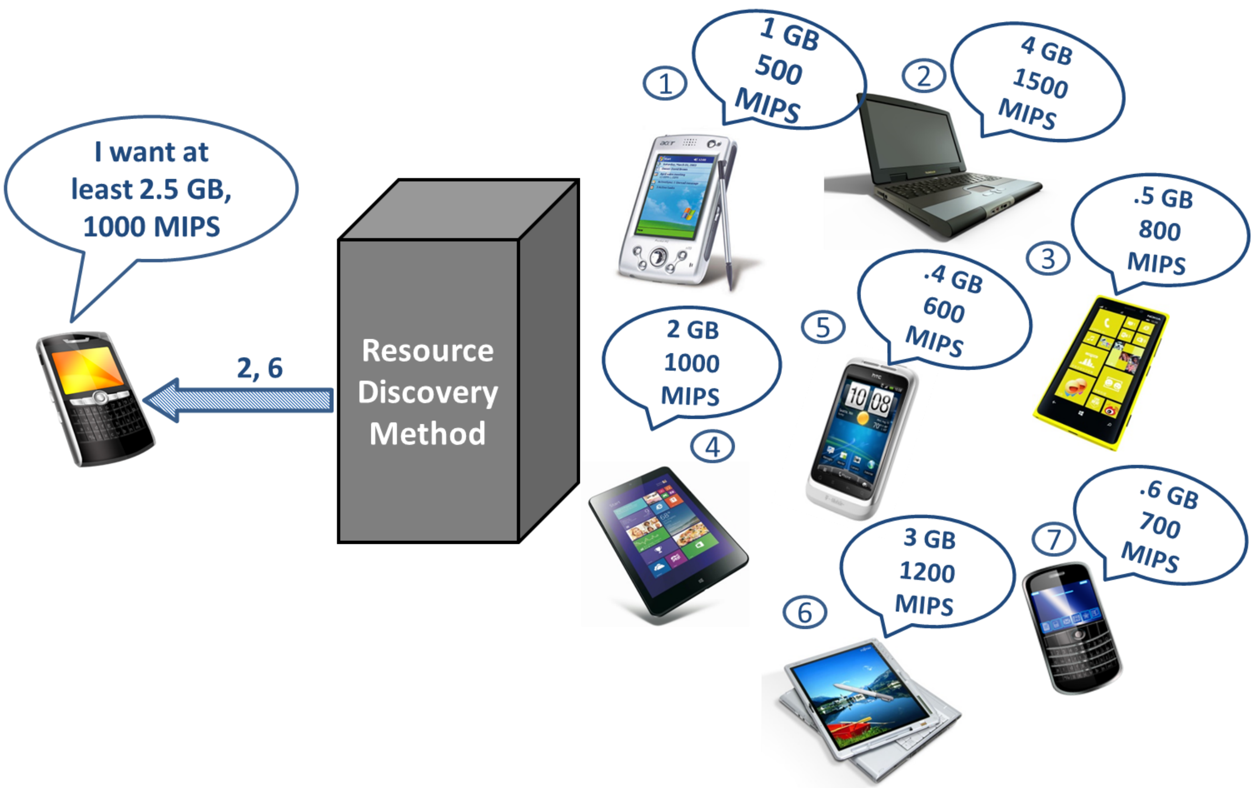

Figure 2.1: Resource Discovery Process

The environment of resource discovery in cooperation based architecture can be understood by the figure 2.2.

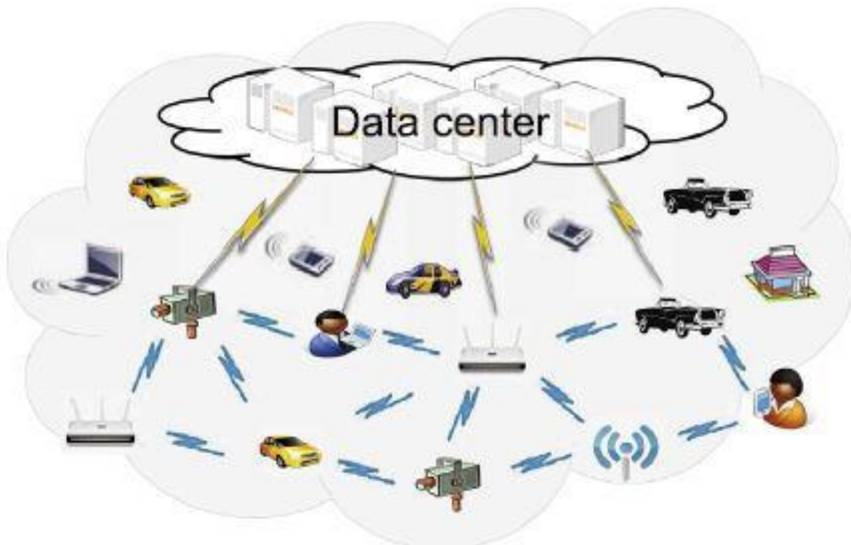

Figure 2.2: Resource Discovery in Cooperation based Architecture [16]

In this environment, resources are not fixed at a particular location. As the device moves, its resources also move from one place to another. Therefore, whenever a device requires some resources, it has no idea of which device is having the required resources. For the identification of the devices, which will be serving as service provider, a mechanism called resource discovery is needed.

2.1 Issues in Resource Discovery

In the context of MCC, following issues must be kept in mind while designing the solution for the resource discovery.

2.1.1 Mobility of Nodes

Due to mobility of the nodes, it is very much possible that the resources discovered previously are not present at the time of servicing the request. Therefore, resources must be discovered when it is needed and such information should not be stale.

2.1.2 Battery Limitation

The node, with very limited remaining battery lifetime, should not be considered as service provider as the battery will be drained quickly. Additionally, the method for resource discovery must be energy efficient because mobile devices are equipped with very limited power backup due to the small size of devices.

2.1.3 Delay Minimization

There is no point in discovering an appropriate resource with huge delay because resources are needed as soon as possible. So it is warranted that while discovering the nodes, delay should be minimized as much as possible.

2.1.4 Scalability

Resource discovery method should be applicable to a large network where the number of devices may be quite large. It should be based on the capability of devices to handle, service and forwarding the requests and also to maintain the desired information.

2.2 Parameters for Resource Discovery

Some parameters based on which a resource discovery is made have been listed as below.

2.2.1 Distance

While performing resource discovery, the closer neighborhood devices may be given preference in comparison to the nodes which are far from the requesting device. It is because serving a nearby device is economical in terms of battery power, collision, and delay.

2.2.2 Signal Strength

Wireless signal strength is also a major factor for deciding the provider nodes as there might be a case in which two nodes, located nearby, have a very large and thick obstacle between them. Due to obstacle, signal strength may be very weak and so the required service is interrupted frequently.

2.2.3 Coverage

Resource discovery can be performed to serve different scenario. It can be classified mainly in two groups:

I. For the whole network
II. For the part of a network

In the first case, resource discovery is performed globally in the network and all the devices which are having resources are identified. Identification information is given to the requester node eventually.

In the second case, a criterion can be set up so that only some devices with the resources are identified. In this mode, whole network is not covered for the resource discovery. It is beneficial in the case where many devices are having the resources. Using this mode, local discovery can be performed to save energy and other QoS parameters.

2.3 Resource Discovery Architecture

The architecture for the resource discovery can be classified as follows [17].

- Directory based Architecture
- Directory less Architecture
- Hybrid Architecture

2.3.1 Directory Based Architecture

In this architecture, a directory maintains the information of the resources of the network. Further, the directory can be centralized or may be distributed amongst the nodes of the network. Centralized directory scheme has the flaw of "single point of failure" and scalability problem. Therefore, in a large scale and resource constraint network distributed approach is preferred.

Distributed approach is further classified as – Backbone based, Distributed Hash Table (DHT) based and Cluster based. In Backbone based architecture, a backbone of directory nodes is formed. Service providers register their services to one or more directory nodes. Backbone nodes are in constant communication with each other. Whenever a client requires resources, it contacts to the backbone nodes.

In DHT based scheme, services are represented by keys and the keys are mapped to a particular region using a hash function. It is necessary to use the location information in this method. In Cluster Based scheme, several nodes make a cluster based on some factors e.g. distance, mobility, etc. Each cluster contains only one directory node which is responsible for storing the information of its local nodes.

In Directory based architecture, huge communication cost is needed to maintain the directory otherwise consistency will be lost.

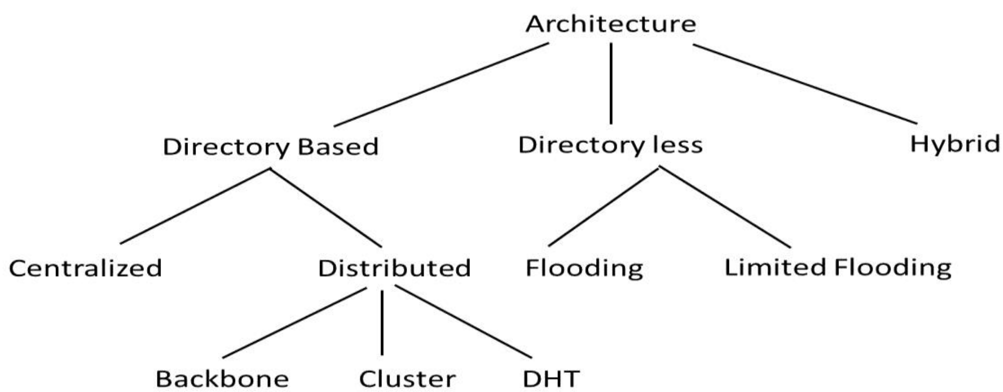

Figure 2.3: Classification of Resource Discovery Architecture

2.3.2 Directory less Architecture

In this architecture, no information is maintained in a directory. Service providers and clients directly communicate with each other using broadcasting. Since broadcasting causes lot of traffic, in the network, several methods can be used to limit the load such as advertising range bounding, peer to peer information caching and selective forwarding. However, in these approaches resource information availability is reduced.

2.3.3 Hybrid Architecture

In hybrid approach, service providers register their services/resources to a directory in their vicinity. If no directory is found then the information is broadcasted. Clients work in a similar manner.

2.4 Related Work

Resource discovery has been an important issue in the field of Mobile Ad-hoc Network (MANET), Wireless Sensor Network (WSN) and Peer to Peer Computing etc. Many methods have been suggested for this purpose. Some practical implementations are – JINI, Salutation, Universal Plug And Play (UPnP), Service Location Protocol (SLP) and Bluetooth Service Discovery Protocol (SDP) [18]. SLP, JINI and Salutation have been classified as Centralized Directory Based Architecture.

SLP defines three types of agent for service discovery purpose; User Agents (UA), Service Agents (SA) and Directory Agents (DA). UAs are responsible for performing service discovery process. SAs are responsible for advertising the services with attributes and location. DAs are responsible for storing information about the services. These agents communicate through multicasting. SLP is not suitable for large network of devices. JINI uses lookup servers for storing resources information. Clients' requests are handled by these servers. The devices which are based on Java (Java Virtual Machine), only can take advantage of JINI because it only supports Java. Salutation is based on directory called Salutation Manager (SLM). Service information of a device is registered to its own SLM. Each SLM frequently checks the service availability of other SLM.

UPnP is based on two entities called control points and devices. Control points act as directories. Multicasting is used for service request as well as service advertisement. Therefore, it is also not suitable for the large network. Bluetooth service discovery protocol (SDP) is used specifically for Bluetooth enabled devices. It uses two modes for service discovery i.e. service searching and service browsing. Service searching is a method in which a requester generates a query containing all the required attributes of a service. Service provider only replies with the specific service which was queried. In service browsing mode, a requester sends a generic query to the provider and provider replies with all the services which he has. SDP is only capable of finding the services in one hop neighborhood.

A location based method has been proposed in [19]. The Content Server (CS) nodes advertise their content along a certain geographical direction. In that path, nodes will cache the information of the content and CS. These nodes are called Content Location Server (CLS). Client requests are also forwarded in such direction so that query should intersect at least one CLS. Although, this mechanism is good enough for dense network, it is not well suited for a sparse network.

Crossing layer strategies are those, in which service discovery mechanisms are combined with routing layer protocols [20]. Definitely, the number of update messages is significantly less but compatibility problem will surely arise. Different implementations are needed for different types of routing protocols.

A method, similar to the proposed model in this work, is proposed in [21]. The main theme of this method is also clustering but it is based on only directory coverage i.e. number of nodes in the neighborhood of directory. It does not consider any other parameters e.g. mobility and battery lifetime etc.

To the best of our knowledge, the only research work in the field of resource discovery in MCC is in [16]. In this, the energy consumption is calculated using both the methods; centralized (directory based) and flooding (directory less). In centralized architecture, all requests are sent and serviced by a central directory. In flooding based architecture, requests are flooded throughout the network. Among these two methods, the method which is more efficient for the next time slot (based on the previous time slots) is selected for the purpose of resource discovery. If a large scale network is considered, the simple centralized method cannot be preferred because a single directory cannot handle all requests. Similarly, flooding method incurs a very heavy traffic. Hence, a more suitable approach is required for a network where the numbers of nodes are dynamically exceeding. Also, this adaptive method does not eliminate the flaw of ‘single point failure’ because the computation of selecting the appropriate method for next time slot is carried out on a centralized server. Delay is also more because the server is located at a distant location.

Chapter 3

Resource Discovery Process: The Proposal

As discussed earlier, the work in this dissertation deals with the resource discovery in a Cloud with heterogeneous network. For this, two modes of network are considered as ad-hoc Wireless LAN (WLAN) and 3G network. When a device has to communicate in a local network in its vicinity, it uses WLAN mode to save energy. A device uses 3G mode when the communicating node is at a distant location. The proposed model performs the resource discovery process by forming the clusters of devices. The details are as follows.

3.1 Clustering Method

Clustering method deals with the clustering of the devices which is done based on some characteristics. Each node, in a cluster, is either treated as a cluster head (CH) or a cluster member (CM). CH is responsible for maintaining the resource information of its cluster members. Clustering is performed in a distributed manner [22].

The clustering process, followed in this work, is based on four parameters.

3.1.1 Mobility

The clustering of the devices is decided based on the mobility parameter of the devices. The node which is comparatively less mobile with respect to its neighbor devices is given the preference to become a CH. The reason for this being the longevity of the cluster head. A node with least mobility is more likely to continue and serve as a cluster head for a long period of time. This will serve the requirement of a stable cluster formation as least mobile nodes can remain the CH for a longer period and hence avoiding the overhead of frequent cluster formation.

3.1.2 Distance

Another parameter to decide the cluster head is the distance of the mobile nodes with each other. The CM is kept at most one hop distance from its CH. So clusters are formed with the nearby devices. This way, while performing the local discovery among a cluster, energy is saved and delay is minimized.

3.1.3 Signal Strength

Another important parameter may be the signal strength of the mobile devices. This parameter is not explicitly maintained. The calculation of mobility is also based on the received signal strength [23].

3.1.4 Battery Power

Mobile devices are equipped with the battery to sustain. Preference of becoming a CH is given to a mobile node which is having high remaining battery power with respect to its neighbors. This parameter is important because CH node has more responsibility as compared to CMs. This way, CH node can serve for a longer duration as it is having more battery power.

3.2 Cluster Formation

For the cluster formation, each mobile device is referred as a node. The steps of the clustering algorithm are as follows.

1. Each node sends a hello packet to its 1-hop neighbors. The signal power which is detected at the receiver node, in this step, is denoted by $R_{power1.}$ After moving some distance, again nodes send second hello packet to their neighbors. The signal power detected by the receiver node at this step is denoted by R_{power2}.
2. A relative mobility metric ($Mob_P Q$) at a node P with respect to node Q is calculated as given in equation 3.1 [23].

$$Mob_P Q = 10 \log_{10} \frac{Rpower2\ (P \rightarrow Q)}{Rpower1\ (P \rightarrow Q)} \qquad (3.1)$$

Where, P is the receiver node and Q is the sender node.

3. Each node now sends this relative mobility metric and its own remaining battery power bat_{power} to all its neighbors.
4. Every node calculates an aggregate mobility metric based on the variance of all received relative mobility metric. The aggregate mobility metric at node P is calculated as given in equation 3.2 [23].

$$Mob_{aggr}(P) = \frac{\Sigma_{Q=1}^{k}(Mob_{P}Q)^2}{k} \qquad (3.2)$$

   Where, k is the number of neighbors of node P.

5. The aggregate mobility value which is calculated in the previous step is sent to every neighbor.
6. Every node calculates a function value $Clus_{func}$ as given in equation 3.3.

$$Clus_{func} = w \times Mob_{aggr} + (1-w) \times bat_{power} \qquad (3.3)$$

   Where w is the weight factor and $0 \leq w \leq 1$.
7. The function value which is calculated in the previous step is also sent to each neighbor.
8. The node which has the minimum $Clus_{func}$ among its neighbors, declares itself to be a CH. Other nodes will declare themselves to be CM.
9. The node which has more than one CH in its neighbors will join the CH who has the minimum $Clus_{func}$ value among those CHs.

This algorithm ensures that no two CH are 1-hop neighbors. For simplicity, steps 8 and 9 are written in an abstract way. The detailed working can be understood by considering two types of packets as given below [22].

1. CH packet
2. Join packet

Whenever a node decides to become a CH, it sends a CH packet to all its neighbors. Upon receiving CH packet neighboring nodes come to know that there is a CH in their 1 hop distance. Similarly, whenever a node decides to become a cluster member of a particular CH, it sends a join packet to all its neighbors and hence informing other nodes that it is not in the competition of becoming a CH.

This results in two cases as follows.

On receiving a CH Packet

When a node x receives a CH packet from a neighbor y, it checks that all other neighbor nodes except node y which have $Clus_{func}$ value less than $Clus_{func}$ value of node x, has already sent a Join packet. In this case, node x understands that there is only one neighbor node y which is CH and having $Clus_{func}$ value less than its own. Therefore, node x will join node y's cluster and the algorithm execution will stop because it already knows what its role is going to be. But, if there is at least one node which is having $Clus_{func}$ value less than node x's $Clus_{func}$ value and it has not sent a CH packet or Join packet, then node x will simply wait. It will not join node y's cluster.

On receiving a Join Packet

When a node x receives a Join packet from neighbor y, it checks if node x is a CH or not. If it is CH, node x checks whether node y wants to join its cluster. It further checks if all the neighbors which have $Clus_{func}$ value greater than the $Clus_{func}$ value of x, have already sent a Join packet. If this is the case, node x stops the execution of algorithm. If x is not a CH, it will check that all the neighbors which have $Clus_{func}$ value less than its own $Clus_{func}$ value have already sent a Join packet. It implies that x is the node with highest $Clus_{func}$ value among its neighbors. So, it declares itself to be a CH. Further, if all the neighbor nodes of x which have $Clus_{func}$ value greater than $Clus_{func}$ value of x have joined some cluster, node x stops the execution of algorithm. In this case, there will be a cluster consisting a single node x.

3.3 Cluster Maintenance

After some time of the cluster formation, it is possible that the nodes move from one cluster to another cluster. In that case, the property on which clusters are formed may be violated. Therefore, it is necessary to maintain the cluster by reassigning the role of nodes as CH or CM wherever necessary. For cluster maintenance, we assume that each node is aware of its link failure from a node and new link formation to a node.

Link Failure

If a node x finds that a link is broken from node y then the following case is needed to be checked.

Case 1: The node x is a CH and node y is a CM of its cluster. In this case, node x removes node y form its cluster.

Case 2: The node x is a CM and node y was its CH. In this case, node x checks if there is another CH which has minimum $Clus_{func}$ value in its neighborhood than its own. If CM finds such node, it becomes the member of that CH. Otherwise; it declares itself to be a CH.

New Link

If a node x finds a new link with node y and node y is a CH, then the node x checks that node y's $Clus_{func}$ is less than its own. If it is true, node x becomes CM of node y.

3.4 Resource Discovery

The model considers a Centralized Information Server (CIS) which is responsible for storing the CH's information. Information may be in the form of IP address, location or some other context. In the proposed work, the IP address is taken as information. CIS may be an access point nearby the devices or it may be a device at a distant location. Since the CIS is responsible only for storing the information of CHs of the network, its information can be replicated to some other devices to achieve reliability in case of failure of CIS.

3.4.1 Steps in Resource Discovery Process

The following steps are taken for the resource discovery process.

1. Clustering is performed in the local network as shown in figure 3.1.

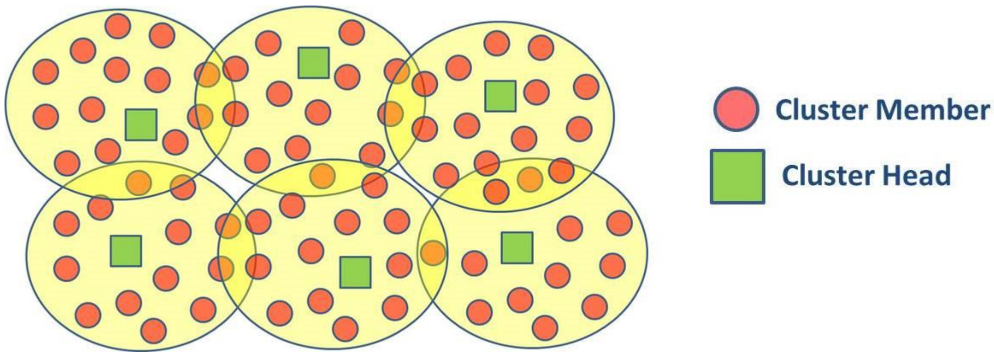

Figure 3.1: Cluster Formation

2. After clustering the CHs; information is stored in CIS as shown in figure 3.2. Each CM node registers its resource information to its CH.

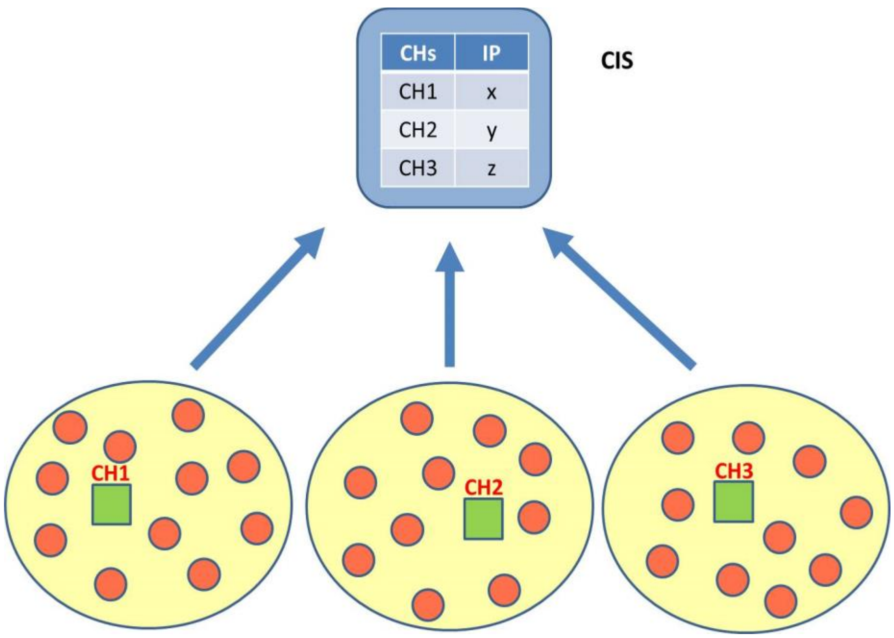

Figure 3.2: CHs Register their Information to CIS

3. Any requester node, which needs resources, will first contact to its CH as shown in figure 3.3.

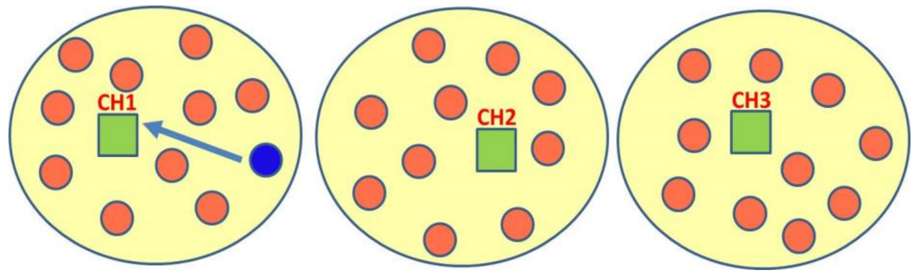

Figure 3.3: Requesting Node (in blue) asking to CH for Provider's Information

4. If the cluster members of its CH have the required resources, CH will respond to the requester node by giving the identity of the nodes which have the required resources.
5. If cluster members do not have the required resources, CH will send a request to CIS for the identity of other CH members as shown in figure 3.4.

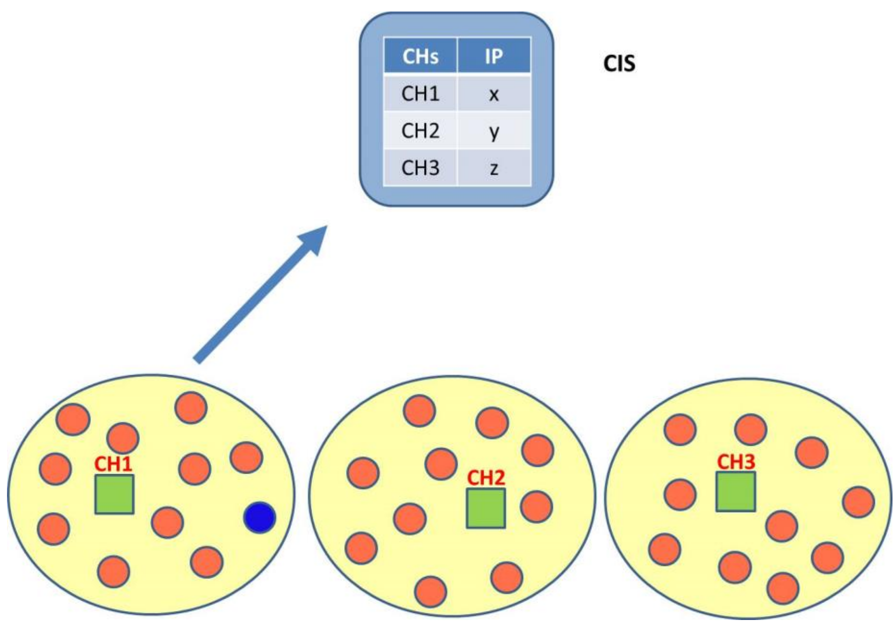

Figure 3.4 CH1 asking to CIS for other CH Information

6. CIS will respond about other CHs as shown in figure 3.5.

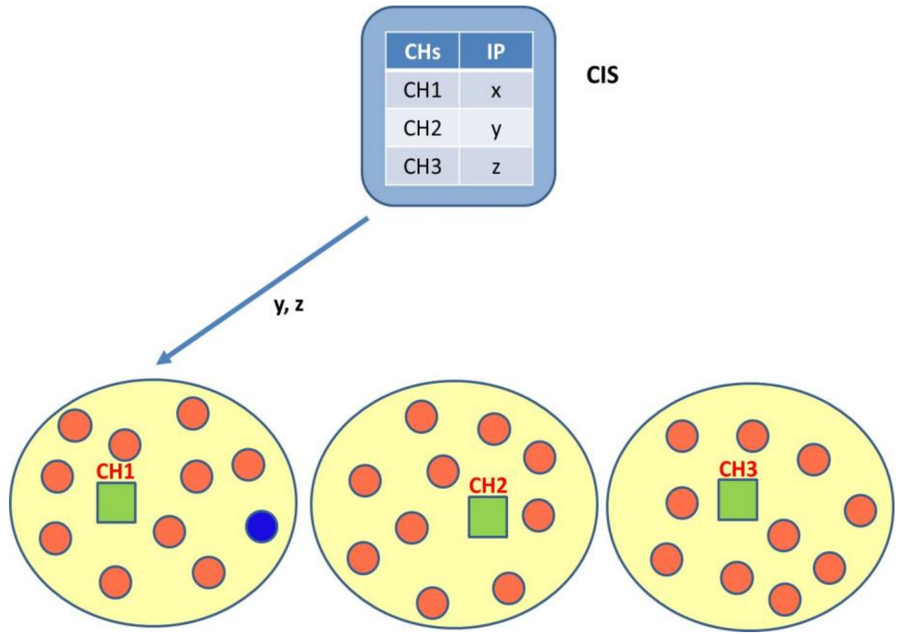

Figure 3.5 CIS is giving Information of other CHs to CH1

7. Requesting CH will contact other CHs asking if their CMs have required resources as shown in figure 3.6.

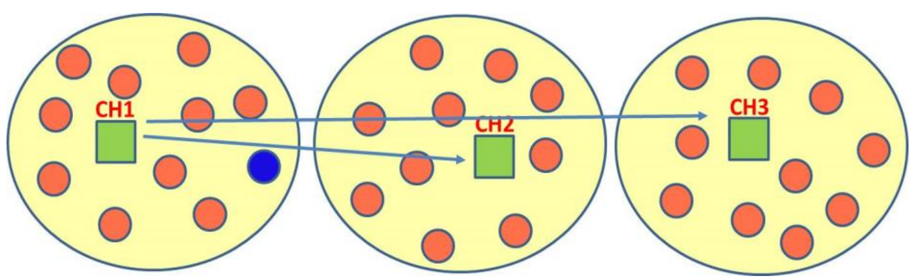

Figure 3.6 CH1 is Contacting to other CHs for Provider Information

8. Other CHs will respond with the identity of provider nodes to the requester CH as shown in figure 3.7.

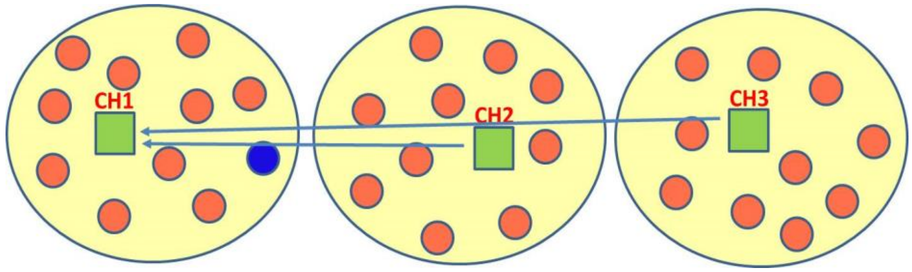

Figure 3.7 Other CHs are giving Provider Information to CH1

9. CH will give the information of provider nodes to the requester node as shown in figure 3.8.

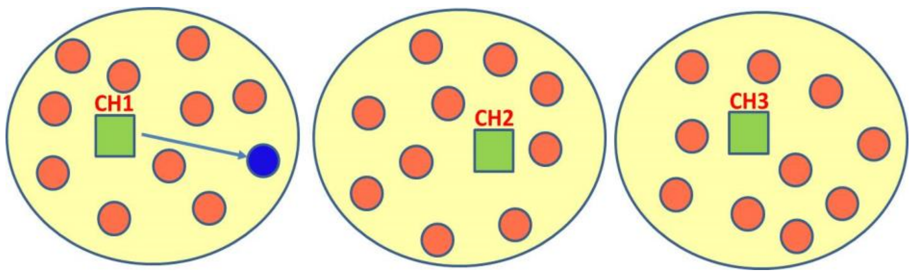

Figure 3.8 CH1 is giving Provider Information to Requesting Node

In this chapter, a model for the resource discovery has been proposed. It also discusses many of the related terms and definitions. The proposed model (algorithm) has been implemented (by simulation) and the performance results are given in the next chapter.

Chapter 4

Performance Evaluation

The proposed model for resource discovery has been evaluated considering two QoS service attributes; energy and delay. The model has been compared with three basic methods – centralized, flooding and limited flooding approach as discussed below [16].

- **Centralized Approach**

  This is the directory based approach for resource discovery. In this method, a centralized server is responsible for saving all the resource information. The mobile devices will register their resource information with the server as soon as they join the network. Whenever a device requires any resource, it contacts the server and the server will respond to the requester node about the identification of the provider nodes.

  Although this method is easy to implement, it incurs huge delay because of the centralized server which is assumed to be located at a remote distance. Updating resource information on the centralized server is necessary otherwise information will be stale. This method is also susceptible to single point of failure because all the requests are handled by a single server.

- **Flooding Approach**

  This is directory less approach for resource discovery. The requester node floods the information to the whole network. Whenever a node receives the request for the first time it broadcasts the request to its neighbor nodes and neighbor nodes do the same until all nodes receive request. The nodes which are having the required resources will send reply to the requester node.

  This method is very fast because in flooding shortest path is guaranteed and method is used to serve in a local network only. No updation is needed because there is no registration of resource information but it incurs very heavy traffic because request is forwarded by all the nodes.

- **Limited Flooding Approach**

  It is also similar to the flooding approach as discussed above but in this, flooding is limited. Desired TTL (Time to Live) value can be found so that only some part of the network is covered. Using this approach, traffic can be reduced to a great extent but all resource information is not covered. This method is useful when the network is dense and there are plenty of resources in the network.

4.1 Energy Calculation

In the simulation two types of energies are counted; energy to transmit the message and energy to receive the message. Energy required for transmission and reception is also different for wireless LAN mode and 3G mode. Various parameters used for the energy calculation are as follows.

Energy to transmit a message in wireless LAN mode = E_trans_lan
Energy to receive a message in wireless LAN mode = E_recv_lan
Energy to transmit a message in 3G mode = E_trans_3G
Energy to receive a message in 3G mode = E_recv_3G

The ratio between the energy for 3G mode and wireless LAN mode is considered as 20:1 [24].

4.2 Delay Calculation

In the proposed model, four types of delay have been counted.

- Transmission delay (T_trans)
- Propagation delay (T_prop)
- Processing delay (T_proc)
- Queuing delay (T_queue)

Total delay *d* can be computed using the equation 4.1 [25].

$$d = T\_trans + T\_prop + T\_proc + T\_queue \qquad (4.1)$$

4.2.1 Transmission Delay

This is the delay required to transmit the packet onto the outgoing link. It can be defined as given in equation 4.2 [26].

$$Transmission\ delay\ (T\_trans) = L/B \qquad (4.2)$$

Where, L=length of the packet

B=bandwidth of the network

4.2.2 Propagation Delay

This is the delay required for a signal to travel from sender to receiver. It can be defined as in equation 4.3 [26].

$$Propagation\ delay\ (T\_prop) = d/v \qquad (4.3)$$

Where, d=distance between sender and receiver

v=velocity of signal

In wireless communication, v is taken as speed of light i.e. 3×10^8 meter/second.

4.2.3 Processing Delay

This is the delay taken by the routers (intermediate nodes in an ad-hoc network) for handling the packet i.e. processing the packet header, determining the next hop to transmit the packet.

For a particular application 'a', there are some parameters α_a, β_a, γ_a and δ_a which are used for calculation of processing delay [27].

Where, α_a is the processing cost per packet.

β_a is the processing cost per byte.

γ_a is the memory access per packet.

δ_a is the memory access per byte.

Number of instructions processed (*instr_count)* can be given by equation 4.4.

$$Instr\_count = \alpha_a + \beta_a * L \qquad (4.4)$$

As defined previously, L is the length of the packet in bytes.

Number of memory access (*mem_access)* can be given by equation 4.5.

$$\text{Mem\_access} = \gamma_a + \delta_a * L \qquad (4.5)$$

We have taken the application 'a' as header processing application and for this application the above parameters can be taken as listed in table 4.1[27].

Processing time (T_{proc_time}) can be given by equation 4.6.

$$T_{proc\_time} = instr\_count / f \qquad (4.6)$$

Where f = clock frequency of the processor.

This frequency f is taken as 233 MHz as listed in table 4.1 [28].

Additional memory access delay T_m can be given by equation 4.7.

$$T_m = mem\_access \times t_{mem} \qquad (4.7)$$

Where t_{mem} = average memory access time

We have taken t_{mem} as 4 ns as in [27].

Total packet processing delay (*T_proc)* can be given by equation 4.8.

$$T\_proc = T_{proc\_time} + T_m \qquad (4.8)$$

4.2.4 Queuing Delay

This is the delay spent in waiting queue of a router by a packet before getting processed. The average queuing delay (*T_queue)* spent by a packet can be given by equation 4.9 [29].

$$T\_queue = \frac{1}{\mu - \lambda} \qquad (4.9)$$

Where, μ is service rate (number of packets a facility can sustain)

λ is arrival rate (average rate at which packets are arriving)

Calculation of service rate μ can be given by the equation 4.10 [29].

$$\mu = B/L \qquad (4.10)$$

Where, B = bandwidth

L = packet length

4.3 Experimental Evaluation

Based on the above parameters, the proposed model is experimentally evaluated in this section.

4.3.1 Parametric Values

The values of various parameters for the experiments are given in table 4.1.

Table 4.1 Parameters for Simulation

Number of nodes	100
Rectangular area	1000m × 1000m
Wireless LAN range	250m
Message length L	1 KB
E_trans_lan	1 Joule
E_recv_lan	0.5 Joule

E_trans_3G	20 Joule
E_recv_3G	10 Joule
Wireless LAN bandwidth	54 Mbps
3G bandwidth	10 Mbps
α_a	4493
β_a	0
γ_a	868
δ_a	0
f	233 MHz
t_{mem}	4ns

Most of the input values conform to [16], [27].

4.3.2 Cluster Formation

Initially, the clusters of the nodes are formed as shown in figure 4.1. Red colored nodes which are marked as '+' denote CH and the numbers printed are the id of CH. The nodes which are same colored belong to the same cluster.

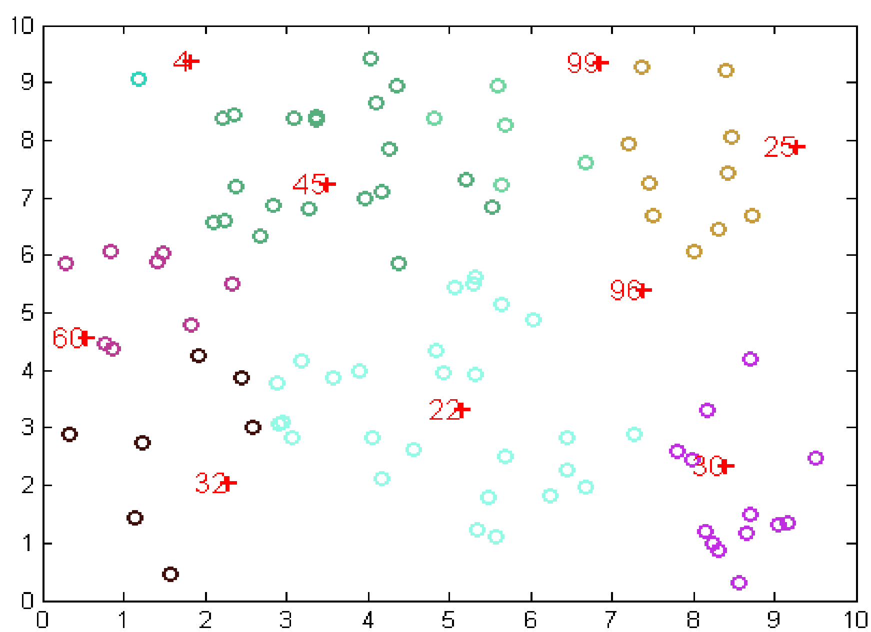

Figure 4.1: Cluster Formation

4.3.3 Experiments

All results have been taken as average value after 20 simulation trials. In the experiments, three types of resources have been considered; CPU, memory and software application. For CPU, the requirements lie between 300 to 1500 MIPS. Also, a node can request up to 4GB of memory.

Experiment 1

This experiment shows the energy and delay consumed by the whole network in the proposed model for 10 requests, 20 requests and up to 100 requests. It is shown in figure 4.2 and figure 4.3 for the energy and the delay respectively. As the number of requests exceeds, delay and energy consumption also increases.

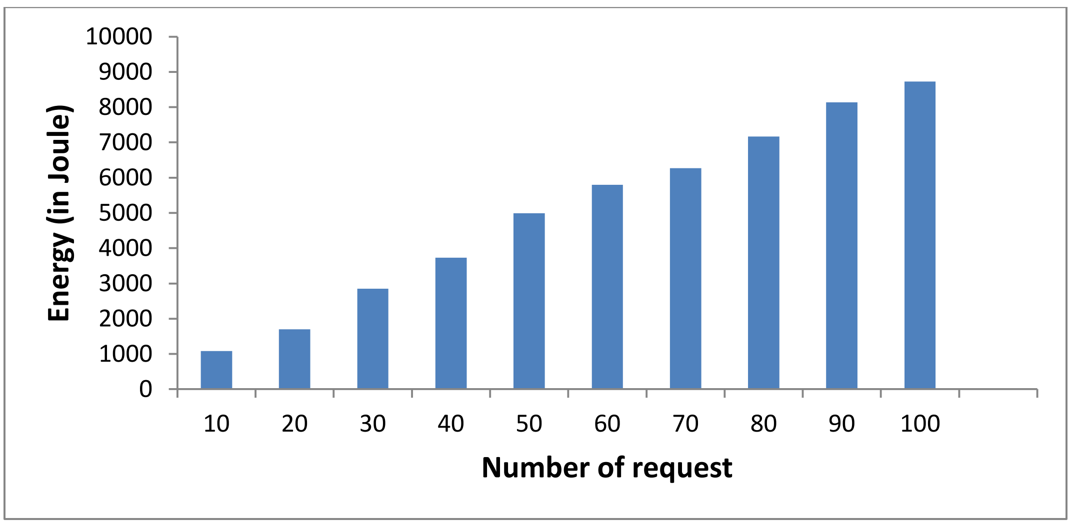

Figure 4.2: Energy Consumption in Proposed Resource Discovery Method

Experiment 2

This experiment deals with comparison of energy consumption and delay required in the following two cases –

Case 1: If 10% nodes i.e. 10 nodes are moving after each request.

Case 2: If 1% nodes i.e. 1 node is moving after 5 request.

The results are shown in figures 4.4 and 4.5.

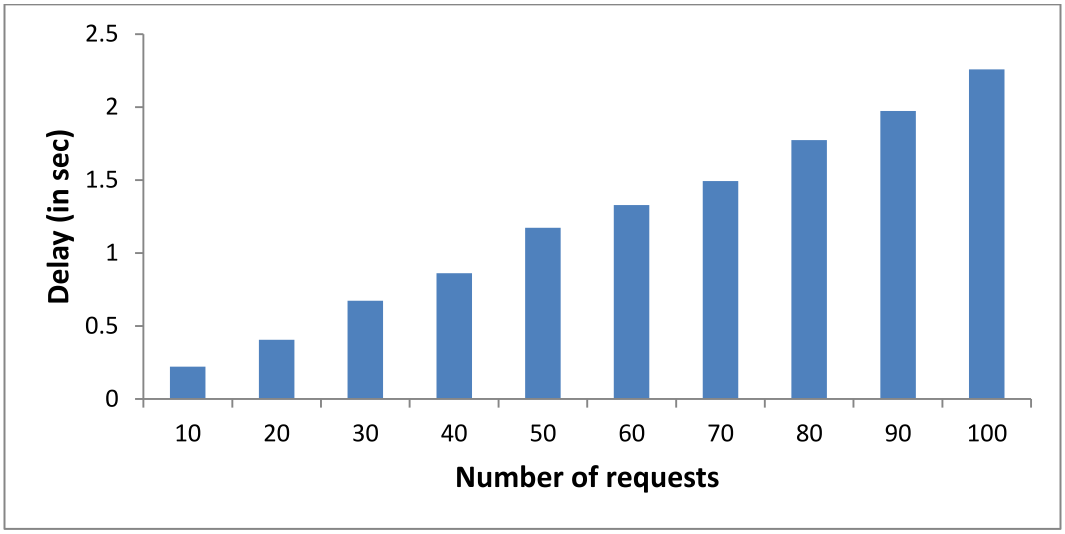

Figure 4.3: Delay in Proposed Resource Discovery Method

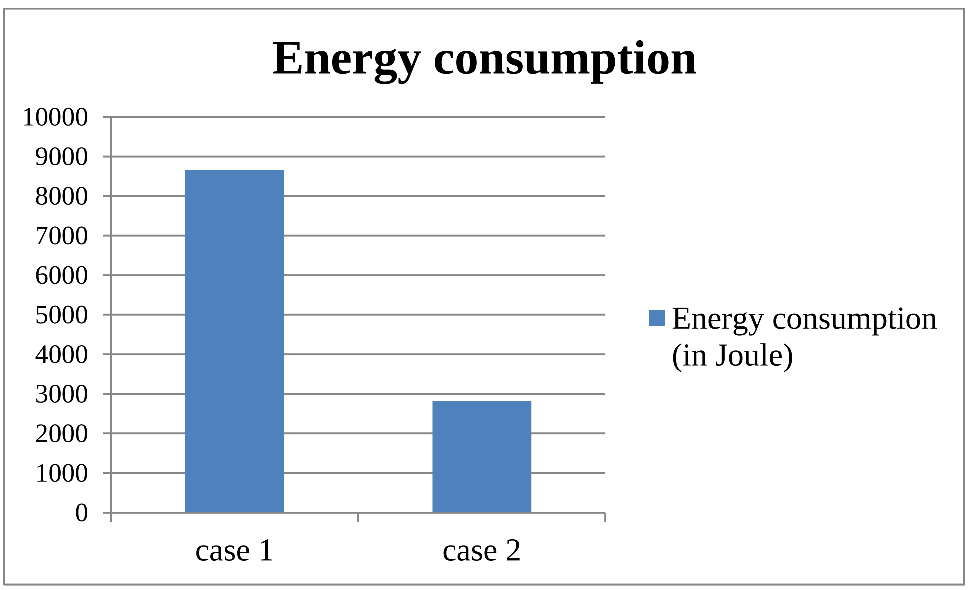

Figure 4.4: Comparison of Energy Consumption in Two Cases

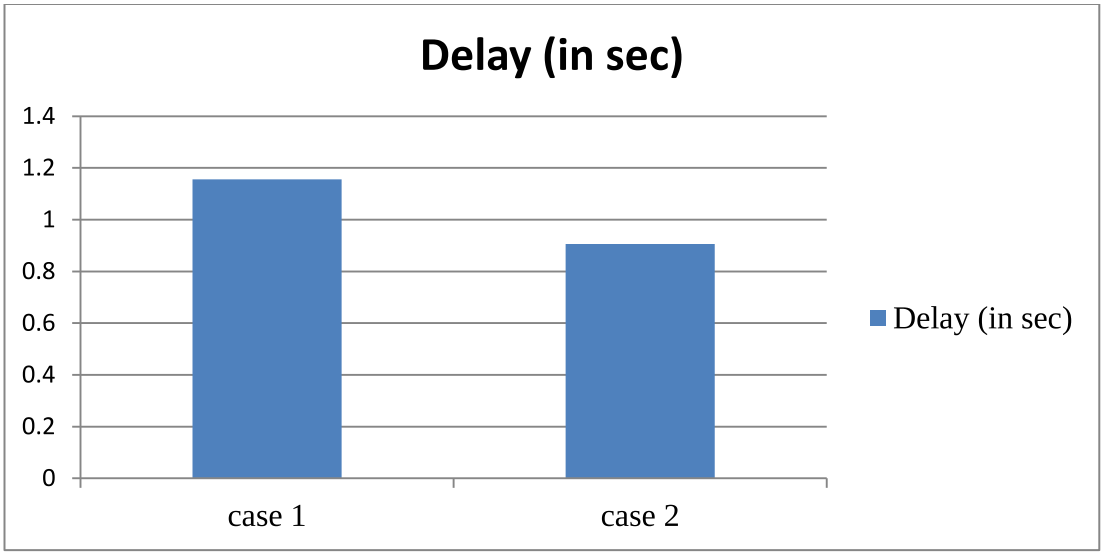

Figure 4.5: Delay Required in above Mentioned Two Cases

It is clear from figures 4.4 and 4.5 that if ten nodes are moving after each request, the energy consumption and delay required is much higher than the case in which only one node is moving after five number of requests. It is so because as the nodes move, the cluster property may be violated. Therefore, to maintain clusters properly, messages will be transferred form one node to another and resource information will be updated accordingly. To perform all these activities, energy will be wasted and some delay will incur.

Experiment 3

This experiment deals with the comparison of energy and delay required in the following two cases –

Case 1: In this case, 10% i.e. 10 nodes move after 5 requests.

Case 2: In this case, 10% i.e. 10 nodes move after each requests.

Results are depicted in figures 4.6 and 4.7.

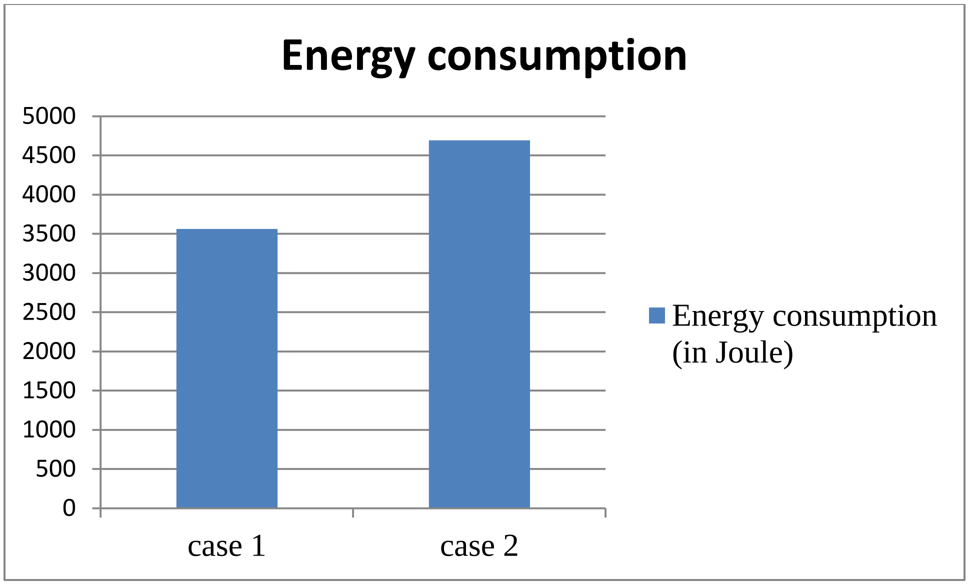

Figure 4.6: Comparison of Energy Consumption in Two Cases

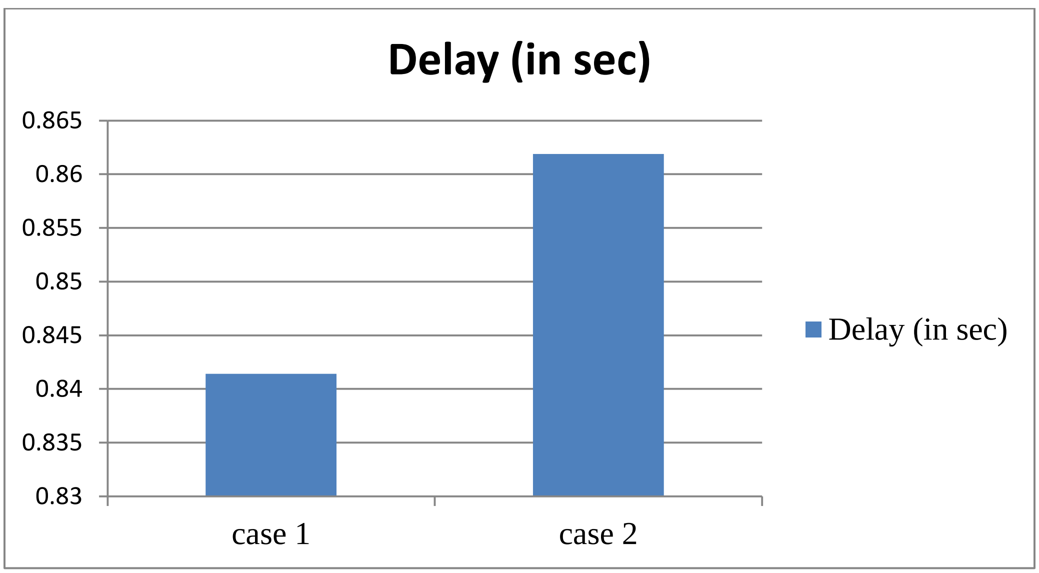

Figure 4.7: Comparison of Delay Required in Two Cases

It is clear from figures 4.6 and 4.7 that as more number of nodes are moving, more energy will be wasted and more delay is incurred to perform the task of resource discovery.

Experiment 4

This experiment deals with the comparison of energy consumption and delay requirement in the following four methods.

1. Basic flooding method
2. Limited Flooding method
3. Centralized method
4. Proposed method

Basic flooding method, Centralized method and the proposed method have already been discussed. Limited flooding method is the modification of basic flooding method. In this method, all nodes are not covered for resource discovery. Only some percentage of nodes is covered and sent requests for finding appropriate resources. In this experiment, 50% nodes are covered for resource discovery process. Since all nodes are not covered, energy and delay required will be less in comparison to basic flooding method but the drawback is requester node cannot find all resource information.

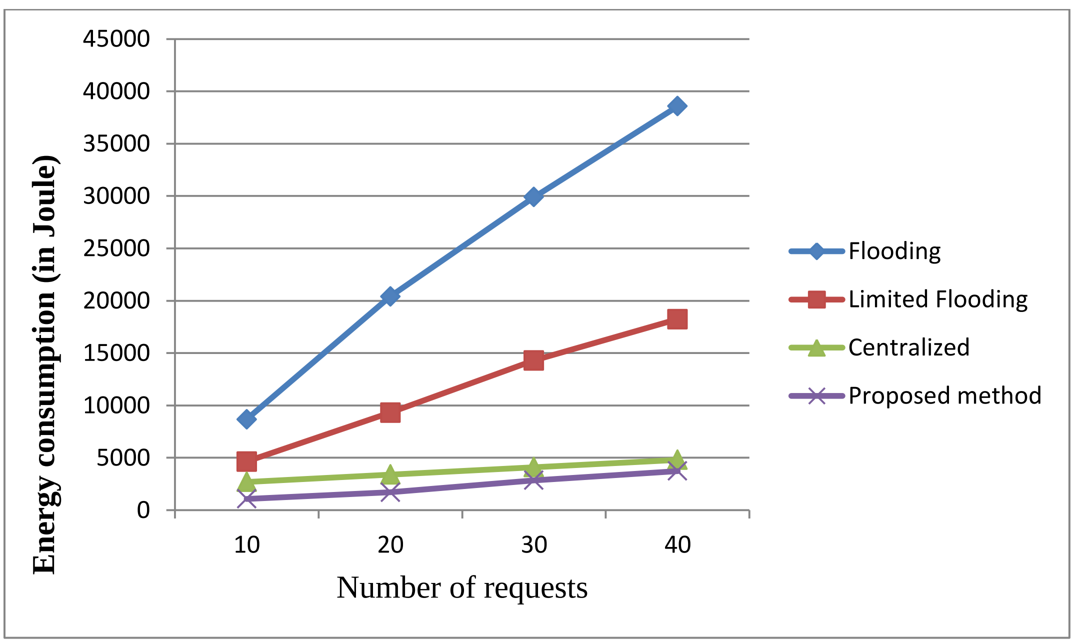

Figure 4.8: Comparative Energy Analysis

It can be observed from figure 4.8 that as the number of requests is increasing, the energy consumption between the centralized method and the proposed method is approaching to same. It is so because in this experiment it has been assumed that the resources for accomplishing the services have been taken from only one provider so only one updation is needed to the server. But in real case, resources can be taken from multiple providers. In that case, more information is to be updated to the server. Also, as the resources can be consumed by the devices itself, this information should also be updated to the server otherwise the information availability will not be consistent. This will take much higher energy than it has been shown in the figure 4.8.

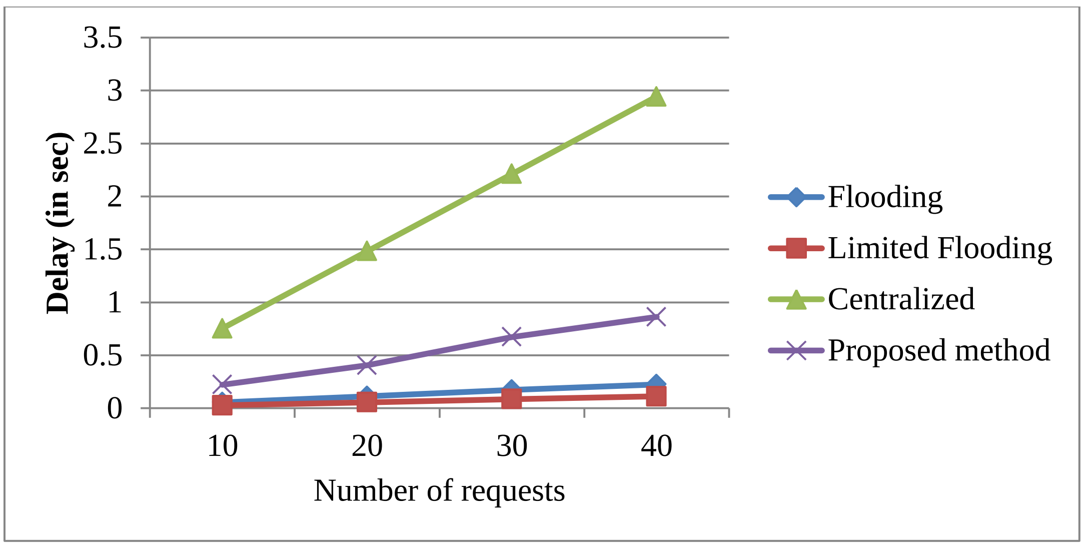

Figure 4.9: Comparative Delay Analysis

Figure 4.9 shows that the delay requirement for the flooding method and limited flooding method is very less as compared to other methods. Centralized method is taking a huge delay for performing resource discovery. Proposed method is moderate in terms of delay. It takes less delay in comparison to centralized method but more delay in comparison to flooding method. The behavior of the different resource discovery techniques in terms of *delay* may be attributed to the following reasoning. Least delay observed in flooding methods is due to its ability of ensuring shortest path of communication between a requester node and the corresponding provider node. On the other hand, huge delay in centralized method may be ascribed to the distant location of the central server which results in excessive delays (transmission, propagation, processing and queuing delays). The reason for moderate delay in the proposed method is that if the resources are available in the neighborhood, CH itself can provide the resources. If the resources are not available in the neighborhood then only global discovery is performed.

Chapter 5

Concluding Remarks

Cloud Computing is the fastest emerging paradigm in IT industry. With the invention of capable mobile devices it has become possible to create a big pool of these mobile devices resulting in Mobile Cloud Computing. Though, today's world of smartphones need the services that should be performed instantly and ubiquitously but it possess various challenges to be solved for leveraging the full advantages of Mobile Cloud Computing. Resource discovery is the very first and an important challenge to be addressed because it is the first requirement to service a request. Though, resources can be discovered by various methods, the method should be energy efficient because the sustainability of the mobile devices rely on battery power which is the most crucial component of any mobile device. This dissertation work addresses the problem of resource discovery by creating and maintaining the clusters of the mobile devices.

This chapter draws the conclusion of the work proposed in this dissertation and also guides about the future work regarding the resource discovery issue.

5.1 Conclusion

This work has been organized in five chapters. The summary for each is as follows.

Chapter 1 is focused on the basics of Cloud Computing and Mobile Cloud Computing. It first discusses the definition of Cloud Computing, essential characteristics, its service models, deployment models. Further it discusses with the definition of Mobile Cloud Computing, its architectures, advantages, applications and finally the issues involved in Mobile Cloud Computing.

In chapter 2, the problem of resource discovery has been discussed thoroughly. Various issues and parameters for resource discovery have been elaborated in this chapter. Three types of architecture for resource discovery problem have been discussed. Finally, the different methods for resource discovery which have been proposed in the past in different fields have been mentioned.

Chapter 3 describes the proposed method for the resource discovery. It deals with various parameters used for clustering method required for performing resource discovery. It tells the full process of cluster formation and cluster maintenance. The steps needed for resource discovery have been formulated at the end of this chapter.

In Chapter 4, various results have been shown by performing simulation. Various values for resources have been taken in a certain range. Experiments have been performed to show the energy consumption and delay required in the proposed method under the consideration of different number of request and different mobility parameter. Comparison has also been done among different methods of resource discovery with the proposed method.

Overall, the proposed model performs very effectively for the resource discovery process. In comparison to some other contemporary models, the performance of the model is much better. Thus, it has the potentiality to act as a resource discovery method in a mobile cloud.

5.2 Future Work

This work performs the resource discovery by forming clusters of devices but as the mobility of the devices is increased, energy consumed by this method also increases. It is because of the increased overhead to maintain the clusters. Thus, it is warranted to design a clustering method keeping the high mobility parameter in mind. Also, it is possible to have more than one parameters to be optimized while discovering a resource in mobile cloud. The future work will try to address these issues.

References

[1] L. Kleinrock, “A Vision for the Internet,” *ST J. Res.*, vol. 2, no. 1, pp. 4–5, 2005.

[2] P. Mell and T. Grance, “The NIST Definition of Cloud Computing Recommendations of the National Institute of Standards and Technology,” *Natl. Inst. Stand. Technol. Inf. Technol. Lab.*, vol. 145, p. 7, 2011.

[3] R. Buyya, C. S. Yeo, and S. Venugopal, “Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering IT Services as Computing Utilities,” in *Proceedings - 10th IEEE International Conference on High Performance Computing and Communications, HPCC 2008*, pp. 5–13, 2008.

[4] R. Buyya, C. Vecchiola, and S. T. Selvi, *Mastering Cloud Computing Foundations And Applications Programming*, 1st ed. Waltham: Morgan Kaufman, 2013.

[5] ABIresearch, “Cloud Application Development and Management Platforms to Drive US$3.6 BN in Mobile Enterprise Application Revenues by 2019,” 2014. [Online]. Available: https://www.abiresearch.com/press/cloud-application-development-and-management-platf/, June 2015.

[6] S. S. Qureshi, T. Ahmad, K. Rafique, and Shuja-ul-islam, “Mobile Cloud Computing as Future for Mobile Applications - Implementation Methods and Challenging Issues,” in *IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS), 2011*, pp. 467–471, 2011.

[7] L. Guan, X. Ke, M. Song, and J. Song, “A Survey of Research on Mobile Cloud Computing,” in *2011 10th IEEE/ACIS International Conference on Computer and Information Science*, pp. 387–392, 2011.

[8] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, “Fog Computing and Its Role in the Internet of Things,” in *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, pp. 13–16, 2012.

[9] D. Dev and K. L. Baishnab, “A Review and Research towards Mobile Cloud Computing,” in *2014 2nd IEEE International Conference on Mobile Cloud Computing, Services, and Engineering*, pp. 252–256, 2014.

[10] S. Weiguang and S. Xiaolong, “Review of Mobile Cloud Computing,” in *IEEE 3rd International Conference on Communication Software and Networks (ICCSN), 2011*, pp. 1–4, 2011.

[11] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, “The Case for VM-Based Cloudlets in Mobile Computing,” *Pervasive Comput. IEEE*, vol. 8, no. 4, pp. 14–23, 2009.

[12] N. Fernando, S. W. Loke, and W. Rahayu, “Mobile Cloud Computing: A Survey,” *Futur. Gener. Comput. Syst.*, vol. 29, no. 1, pp. 84–106, 2013.

[13] A. R. Khan, M. Othman, S. A. Madani, and S. U. Khan, “A Survey of Mobile Cloud Computing Application Models,” *Commun. Surv. Tutorials, IEEE*, vol. 16, no. 1, pp. 393–413, 2014.

[14] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, “A Survey of Mobile Cloud Computing: Architecture, Applications, and Approaches,” *Wirel. Commun. Mob. Comput.*, vol. 13, no. 18, pp. 1587–1611, 2013.

[15] M. Schüring, “Mobile Cloud Computing – Open Issues and Solutions,” in *15th Twente Student Conference on IT*, 2011.

[16] W. Liu, T. Nishio, R. Shinkuma, and T. Takahashi, “Adaptive Resource Discovery in Mobile Cloud Computing,” *Comput. Commun.*, vol. 50, pp. 119–129, 2014.

[17] C. N. Ververidis and G. C. Polyzos, “Service Discovery for Mobile Ad Hoc Networks: A Survey of Issues and Techniques,” *IEEE Commun. Surv. Tutorials*, vol. 10, no. 3, pp. 30–45, 2008.

[18] C. Cho and D. Lee, “Survey of Service Discovery Architectures for Mobile Ad hoc Networks,” *term paper, Mobile Computing, CEN 5531*, 2005.

[19] J. B. Tchakarov, “Efficient Content Location in Wireless Ad Hoc Networks,” in *IEEE International Conference on Mobile Data Management (MDM’04)*, pp. 74–85, 2004.

[20] J. A. Garcia-macias and B. Ju, “Service Discovery in Mobile Ad-Hoc Networks : Better at the Network Layer ?,” in *IEEE International Conference on Parallel Processing Workshps (ICPPW’05)*, pp. 452–457, 2005.

[21] F. Sailhan and V. Issarny, “Scalable Service Discovery for MANET,” in *Proceedings - Third IEEE International Conference on Pervasive Computing and Communications, PerCom 2005*, pp. 235–246, 2005.

[22] S. Basagni, “Distributed Clustering for Ad Hoc Networks,” in *Proceedings of the 1999 International Symposium on Parallel Architectures, Algorithms and Networks. IEEE Computer Society*, pp. 310–315, 1999.

[23] P. Basu, N. Khan, and T. D. C. Little, “A Mobility Based Metric for Clustering in Mobile Ad Hoc Networks,” in *International Workshop on Wireless Networks and Mobile Computing (WNMC2001)*, pp. 413–418, 2001.

[24] N. Ristanovic, J. Y. Le Boudec, A. Chaintreau, and V. Erramilli, “Energy Efficient Offloading of 3G Networks,” in *Proceedings - 8th IEEE International Conference on Mobile Ad-hoc and Sensor Systems, MASS 2011*, pp. 202–211, 2011.

[25] Com2networks, “Delays in Transmission.” [Online]. Available: http://com2networks.blogspot.in/2013/09/delays-in-transmission.html.

[26] Behrouz A. Forouzan, *Data Communications and Networking*, 4th ed. Mc Graw Hill, 2007.

[27] R. Ramaswamy, N. W. N. Weng, and T. Wolf, “Characterizing Network Processing Delay,” *IEEE Global Telecommunications Conference, 2004. GLOBECOM '04*, vol. 3, pp. 1629–1634, 2004.

[28] Wikipedia, “IXP 1200.” [Online]. Available: http://en.wikipedia.org/wiki /IXP1200, June 2015.

[29] Wikipedia, “Queuing Delay.” [Online]. Available: http://en.wikipedia.org/wiki /Queuing_delay, June 2015.