Peer to Peer File Sharing on Android

Prepared By

Kiran Acharya

12MCEI42

Information & Network Security

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

MAY 2014

Peer to Peer File Sharing on Android

Major Project

Submitted in partial fulfillment of the requirements

For the degree of

Master of Technology in Computer Science and Engineering

Prepared By

Kiran Acharya

(12MCEI42)

Guided By

Prof. Zunnun Narmawala

Information & Network Security

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

MAY 2014

Certificate

This is to certify that the Major Project Report entitled ""**Peer to Peer File Sharing Android"** submitted by **Kiran Acharya (Roll No: 12MCEI42)**, towards the partial fulfillment of the requirements for the degree of Master of Technology in Computer Science and Engineering of Nirma University, Ahmedabad is the record of work carried out by her under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this major project part-I, to the best of my knowledge, haven't been submitted to any other university or institution for award of any degree or diploma.

Prof. Zunnun Narmawala
Assistant Professor
and internal Guide ,
CSE Department ,
Institute of Technology,
Nirma University, Ahmedabad.

Prof. Sharada Valiveti
Associate Professor
Coordinator M.Tech - CSE(INS)
CSE Department,
Institute of Technology,
Nirma University, Ahmedabad.

Dr. Sanjay Garg
Guide,Professor and Head,
CSE Department,
Institute of Technology,
Nirma University, Ahmedabad.

Dr K Kotecha
Director,
Institute of Technology,
Nirma University, Ahmedabad

Undertaking for Originality of the Work

I, **Kiran Acharya**, Roll. No. **12MCEI42**, give undertaking that the Major Project entitled "**Peer to Peer File Sharing on Android**" submitted by me, towards the partial fulfillment of the requirements for the degree of Master of Technology in **Computer Science & Engineering** of Nirma University, Ahmedabad, this work is originally done by me and no attempt of plagiarism has been made. Any event of similarity with any published work or any dissertation elsewhere will result in severe disciplinary action.

Signature of Student

Date:

Place:

Endorsed by

Prof. Zunnun Narmawala

(Signature of Guide)

Acknowledgements

Apart from the efforts made by me, the success of Masters dissertation depends on the help, support and guidelines of many people. I want to thank the people who played an important role in the successful completion of my project.

I would like to express the deepest gratitude to my guide, **Prof. Zunnun Narmawala**, Assistant Professor of Computer Science and Engineering Department, Institute of Technology, Nirma University, Ahmedabad, for his valuable guidance and continual encouragement throughout this work.His guidance has played a valuable role in understanding the project and the process to complete it.

It gives me immense pleasure in expressing thanks to **Prof. Sharada Valivetii**, PG INS - Coordinator, Computer Science Department, Institute of Technology, Nirma University, Ahmedabad for for an exceptional support and continual encouragement throughout the Major Project. The appreciation and continual support she has imparted has been a great motivation to me in reaching a higher goal.

A special thank you is expressed wholeheartedly to **Dr K Kotecha**, Hon'ble Director, Institute of Technology, Nirma University, Ahmedabad for the unmentionable motivation he has extended throughout course of this work.

I would thank Nirma Institute of Technology, all faculty members of Computer Engineering Department, Nirma University, and Ahmedabad to provide the environment to fulfill this project.

Most profound regards to my parents,brother and friends, for their consistent support and love.Finally, I must thank GOD for giving me the environment to study, people to help, opportunities to encash and potential to succeed.

-Kiran Acharya

12MCEI42

Abstract

Android provides us various functionality to perform various operations, Wifi-Direct is the part of the enhancement made in android 4.0 version. With this we can have peer to peer file sharing. In this Application named "Torrent" we used this and made an application through which you can transfer different kind of files between two devices.
Firstly the application was built and tested only for one to one communication after getting success in that we extended the application for multi-hop communication.As we worked on opportunistic network where we need to transfer the packet as soon as we get the opportunity in such an environment the application works very well. As we are using Wifi-Direct instead of Bluetooth the range and transfer rate has been an advantage.
Overall this application is an effort to provide a simple multihop peer to peer file sharing application which provides the user a simple interface to work with easily workable and approachable to all kind of audiences which can perform file sharing with out the need of any central server and internet.

Contents

List of Figures

Chapter 1

Introduction

File sharing provides the mechanism to share different kind of files within two devices. Peer to peer file sharing is a different approach for file sharing. This kind of file sharing became popular 1999 when napster came in existence with the cooperation of the users which have the file and those who want the file. Napster , Kaza are the first generation of peer to peer file sharing, after that bit-torrent came in existence that now works on the no central server requirement but it requires the internet to connect to the available peers.

Peer to peer file sharing in mobile applications are available now.In this application we are using android operating system. This is currently the most popular Os and widely used and provides mostly all the file sharing mechanisms. This application does not require any internet support for the file transfer we are using Wifi-Direct. Files will be transferred using network coding.

1.1 Problem Definition

To create an peer to peer file sharing application that uses Wifi-Direct to share the files in an opportunistic network. That will work perfectly with one to one as well as multihop communication.

1.2 Objective of Work

Provide a file sharing application with a simple user interface provides good speed to transfer the data between two devices by using the opportunity to transfer the packet as soon as the device come in contact.

1.3 Scope of Work

- Study of Pocket Switching Network,Peer to Peer file sharing architecture.
- User interface designing for the application.
- One to one peer communication and file transfer.
- Multihop communication and file transfer.
- Calculating bit rate, packet drop and success rate of the packets transmitted.

1.4 Methodology

- Developing the flow of the application.
- Integrate it with the file sharing android application.
- Measuring the performance of the application.
- Minimize the battery power, time and memory consumption.

Chapter 2

Literature Survey

Peer to peer file sharing using android Wifi-direct for this we need to study different kind of network,network coding , android and peer to peer file sharing applications as reference all these are included in literature survey.

Now for the efficient sharing of files instead of using the regular techniques network coding is preferred so that the efficiency of the data distribution increases, with that the research such as blue-torrent, M2M file-share are also studied to study the requirement for the file sharing application.

Bit torrent and Blue torrent are the different applications implemented and approached study of that also be a good help in building this application.

2.1 Pocket Switching Network

Delay tolerant network is a network that works in challenging conditions where the network links are disconnected very often and that cause long delays. This is effective where the internet protocols are difficult to use and also gives poor performance when used.

So the scenario where the network has no fixed pattern we can use Pocket Switching network, this is a class of delay tolerant network. When there is a central server services relay on the server and connectivity between the end to end devices not only became expensive but also dependable. There is also a part of network resources that are available in mobile phones but mostly remains unused such as local wireless bandwidth, storage

capacity and CPU cycles. To take the advantage of all the unused resources we need to build an application that could properly use all the resources as well as the time of connectivity between the resources, But this also has a drawback that power consumed is much higher then the power required in the central server system. So a mechanism is required to enhance the life time of the mobiles.This are the various terms that are to be considered when we are working with pocket switching network. [5]

2.1.1 Mobility

Mobility is a serious concern because the topology or the movement of the device does not have and fixed pattern or structure to be followed, as well as the availability of the device is also not fixed. In such case it is essential to have some percentage of predictability otherwise it becomes difficult to find the path to forward the packet.[1].

2.1.2 Availability

In pocket switching network availability plays a great role. The end device which is to receive must be on at the time the packet arrived there and it must have enough space for the upcoming packet. Otherwise there would be no chance the packet would reach the destination.If these two requirements are fulfilled then its easy to pass the packet from source to destination.

2.1.3 Security

Forwarding packets through different nodes creates an issue of security.Whether the intermediate nodes are trustworthy or not cannot defined. Irrespective of the central server where the is an authority for the the authentication here is no such security mechanism available. [1].

2.1.4 Forwarding

The key challenge in pocket switching network is forwarding the messages so that they could reach the destination. As we know in opportunistic network there is no scheme to know the status of node in the network. Hence the problem of forwarding is major to deal with that could be described as : In an opportunistic network we need to have some policy to forward the message in such a way that the time for the message to reach the

destination will remain minimum.

2.2 Peer-to-Peer File Sharing

Peer to Peer file sharing is a mechanism where nodes can act as a server and client at a same time.There is no centralized server for the content storage and content distribution.Peer to peer file sharing has different components that includes peers, seeds, torrent, and many more.
In three principles underlying P2P networks are identified:

1. Resource Sharing :P2P systems involve an aspect of resource sharing, such as disk space network bandwidth or services. By sharing resources, applications can be realized which could not be set up by a single node.[12]

2. Decentralization: As the content is available at different nodes but not at any central entity that eventually causes the decentralization in the network. There is no such responsible server to answer the queries for the request asked, and that is the main issue that so many illegal aspects are related with peer to peer file sharing.[12]

3. Self Organization: Due to decentralization there is no more database assailable to store the information or to retrieve the information. Therefore its a key requirement that self organization must be there in the nodes, based on the information they contains locally and with the interaction with the neighbor nodes they could find out the availability of the content. [12]

2.3 Network Coding

Network coding came in existence in 1999 but the work on it started 2003. There are so many projects that are currently working on network coding for example microsoft's avalanche and ncutils project etc. In this big files are divided into small files and then are forwarded to the destination from different paths. With the help of network coding we can assure the improvement in delivery of packet. [7]

Working

Network coding allows coding in intermediate nodes when communication is going on between source and destination.The content of large files is divided into smaller packets.Suppose the file is divided into n number of segments and each is of size k. Each intermediate block receives an encoded block and sends the encoded block by generating a new encoded packet by multiplying the packet with randomly generated encoded vector.Hence there would be a global encoding vector and an encoded packet received by the destination. The destination will decode the packet and regain the original packet.

2.3.1 Random Linear Network Coding

Mainly network coding is of two types; Linear Network Coding (LNC) and Random Linear Network Coding (RLNC). Normally the packets are simply forwarded by the router with out any processing in the packets. In linear network coding the number of packets are combined which it receives and then perform addition multiplication on those packets over Galois fields 2. This is performed at the router source or forwarding node. There is no concatenation performed in this network coding and the size of the packets also remain intact. If the packet generated by the source is of length n then next the destination will receive the file of length n.

In Linear network coding relevant coefficients should be used for the process of encoding and decoding of packets. This process has a requirement of a central authority for coefficient generation. Opposite of that in the wireless environment where the mobility the node is as well as the presence of different kind of networks The approach which is the most suitable is distributed approach.

Hence Random linear network coding provides a different method that is random generation for the encoding coefficients. Wireless networks are different then wired they have dynamic environment hence error rate is higher, similarly interference rate in between the channels is also higher. So according to this the protocols we need to use should be optimized under these conditions where the network topology is not fixed.

In RLNC, every node produces its own particular coding coefficient for each one encoded bundle. Likewise coefficients are sent to the goal in the bundle header. Along these lines, the goal can unravel the bundle without knowing system topology or encoding standards, regardless of the fact that the topology is not altered. Number of effective

transmission was measured for two cases in with and without arbitrary direct system coding. Recreation results demonstrates that there is an expansion in number of fruitful transmissions for separation more terrific than 500 separation units on account of arbitrary direct system coding.[7]

In LNC or RLNC there is higher parcel postpone as in that we need to defer the transmission of effectively arrived bundles until extra parcels have been gathered. In crafty system coding, as opposed to selecting a specific node to be the following jump forwarder, nodes in the system coordinate with one another to select a numerous nodes which can possibly be served as next-jump forwarder. From different nodes, the node which is closest to the end will send the parcel and other will drop the bundles. In this plan coordination amongst the nodes is needed. In RLNC, with expansion in number of nodes, blockage diminishes. In Rlnc,random era of the encoding cofficients guarantees with high likelihood a direct autonomy of the yield parcels from a node for a sufficiently substantial size q = 2m of the limited field Gf.for multicast situation, likelihood that RLNC is substantial is no less than (1 - d=q)n, where d is gathering size i.e. number of end of the line nodes, q=field size and n is number of connections.

2.4 Related Work

There are so many research that has been going on in the field of file sharing that is based on peer to peer file sharing. Peer to peer file sharing has not only limited to personal computers but also have the mobile applications that works with internet connection. With the advancement of technology here are some applications that not only works with the internet but also in the absence of internet with the help of the inbuilt Bluetooth and Wifi-Direct. Here we study the one famous bit torrent which works with internet, secondly Bluetorrent that works with the help of bluetooth.

2.4.1 Bittorrent

Bit Torrent is the best illustration of an end framework agreeable construction modeling. It empowers quick downloads with the base utilization of data transfer capacity. It expands the downloading speed by downloading the pieces at the same time from all the nodes that have that file.so accordingly the expansive documents might be downloaded

in least time.

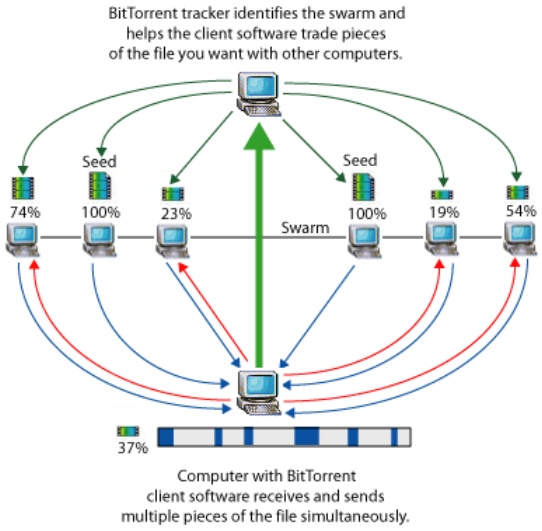

Figure 2.1: BitTorrent File Transfer[11]

Bittorrent parts expansive documents into little squares, which permits clients to download different pieces in parallel from diverse nodes. Once a client has downloaded a given piece, that persons workstation can promptly carry on as a server for that specific square and serve any other person looking for the record.

Bittorrent customer programming speaks with a tracker to discover different machines running Bittorrent that have the complete document (seed workstations) and those with a part of the record (peers that are normally at present downloading the document).

The tracker recognizes the swarm, which is the associated machines that have all of or an allotment of the document and are currently sending or accepting it.

The tracker helps the customer programming exchange bits of the document you need with different machines in the swarm. Your machine accepts various bits of the document at the same time.

On the off chance that you keep on running the Bittorrent customer programming after your download is finished, others can get .torrent records from your machine; your future download rates enhance on the grounds that you are positioned higher in the "one good turn deserves another" framework.

Downloading bits of the document in the meantime assists take care of a typical issue

with other distributed download routines: Peers transfer at a much slower rate than they download. By downloading numerous pieces in the meantime, the general velocity is incredibly made strides. The more machines included in the swarm, the quicker the record exchange happens in light of the fact that there are more wellsprings of each one bit of the document. Thus, Bittorrent is particularly valuable for substantial, prevalent records.

In Bittorrent, discovering the best possible booking of data over the overlay topology with the goal that nodes don't need to sit tight unnecessarily for new substance to arrive is extremely troublesome. The Bittorrent framework utilizes a consolidation of the Random and Local Rarest plans. First and foremost every nodes utilizes Random and after a couple of squares have been downloaded it switches to Local Rarest.it increments the download speed.

2.4.2 Bluetorrent

Bluetorrent is a shared record imparting convention that imparts the document between distinctive nodes utilizing Bluetooth. Blurtorrent like Bit-Torrent utilization document swarming, chiefly due to the little contact length of time and transmission capacity impediment. [8]

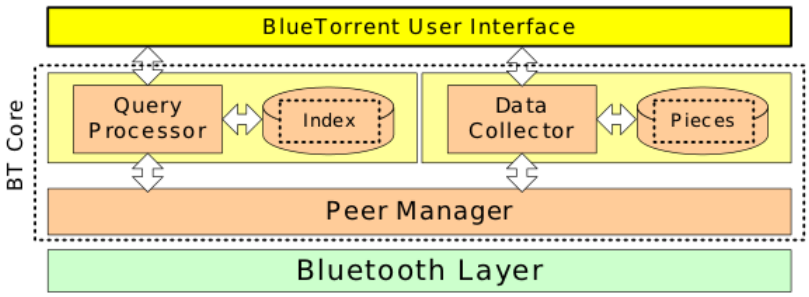

Figure 2.2: Bluetorrent Architecture

Peer Manager:This module is for peer management, means to choose the best peer among the rest. The nodes keeps the tracks of duration and frequency of connections they had been made so far to choose the best peer. From this the rank has been provided to the neighboring nodes. Distance and the duration of contact is taken into account to provide the rank to the node.

Query Processor: Query processor searches the content with the help of indices, this includes title, type , unique id and the other information related to the file. Bluetorrent uses a database for that so that the search could be executed locally in the device. The particular request is taken as a search string and passes to the database as a query whenever any node comes in contact looking for any content. As soon as the nodes receives the query it searches its database and then provides the response that the file is available or not. The response is automatically reported to the query organizer, that decides the next step, that is to download the file or not.

Data Collector: As Bluetorrent uses file swarming the file is divided into n pieces, and each of them are downloaded individually. The node keeps the bitmap of the available pieces for reconciliation. Whenever the connection is available the bit maps has been exchanged so that the available and missing pieces can be determined. The data transfer takes place in asymmetric mode.The size of the piece or block is decided on the basis of available bandwidth and the mobility patterns. The size of the bitmaps are very small so that does not create much overhead.

2.5 Community Detection

Community in the real world is the group of people those share the same interest and interacts often and has something in common. Anything such as work profile, interest, family could make them part of a same community. Communities when represented as graph they are the clusters that are connected with vertices of common properties and has the similar role in graph. hence community detection makes the information propagation easier in pocket switching network.

Identifying the vertices in the group that has large number of vertices connected with it could play an important role in the network. Similarly the vertices which connects two different modules also considered important. Stability in the group and connected edges defines a lotabout the role of particular node in a community.[2]

2.5.1 Methods to find Communities

There are different methods present communities in the network. Here some methods are explained those are highly adopted to find the communities.[2]

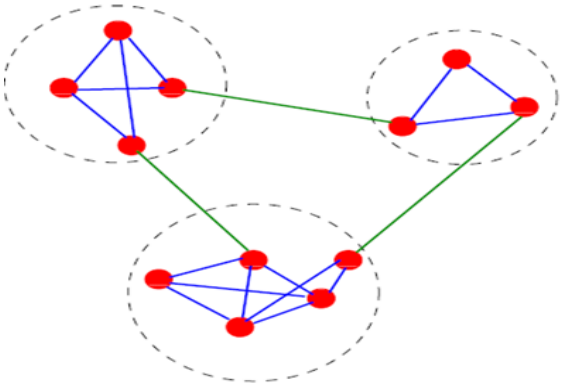

Figure 1: A graph representing three communities enclosed in dashed circles

Figure 2.3: Communities are presented in dashed lines

2.5.2 Graph Partitioning

The issue of diagram dividing comprises in partitioning the vertices in g gatherings of predefined size, such that the amount of edges lying between the gatherings is negligible. The amount of edges running between groups is called cut size. Fig. 2 exhibits the result of the issue for a chart with fourteen vertices, for g = 2 and bunches of equivalent size. Numerous calculations perform a division of the chart. Parcels into more than two groups are typically achieved by iterative bisectioning. In addition, as a rule one forces the demand that the groups have equivalent size. This issue is called least division and is NP-hard[2]. Method: the modules and the amount of edges lying between them. The beginning stage is a starting allotment of the diagram in two groups of the predefined size: such introductory part might be arbitrary or proposed by some data on the chart structure. At that point, subsets comprising of equivalent amounts of vertices are swapped between the two gatherings, so Q has the maximal expansion. The subsets can comprise of single vertices. To diminish the danger to be trapped in nearby maxima of Q, the method incorporates a few swaps that abatement the capacity Q. After an arrangement of swaps with positive and negative increases, the allotment with the biggest worth of Q is chosen and utilized as beginning stage of another arrangement of emphases.

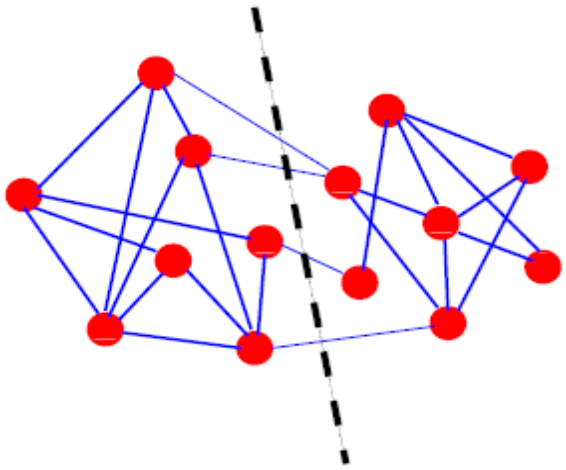

Figure 2.4: Graph partitioning: The dashed line shows the solution of the minimal bisection problem.

On inadequate charts, a somewhat distinctive heuristic permits to bring down the intricacy to O(n2). The segments found by the strategy are emphatically subject to the introductory setup and different calculations can improve. It is desirable over begin with a great figure about the looked for allotment; overall the results are very poor. In this manner the system is commonly used to enhance the segments found through different systems, by utilizing them as beginning arrangements for the calculation.

Calculations for chart dividing are bad for group identification, in light of the fact that it is important to give as enter the amount of gatherings and in a few cases even their sizes, about which on a fundamental level one knows nothing.

2.5.3 Hierarchical Clustering

Informal organizations, for example, regularly have a progressive structure. In such cases, one may utilize progressive grouping calculations, i. e. bunching systems that uncover the multilevel structure of the diagram. Various leveled grouping is extremely basic in informal community dissection, science, building, promoting, and so on. The beginning stage of any progressive grouping technique is the meaning of a likeness measure between vertices. After a measure is picked, one registers the comparability for each one sets of vertices, regardless of in the event that they are associated or not. At the end of this

process, one is left with another n x n grid X, the comparability network

Since groups are combined focused around their common closeness, it is fundamental to characterize a measure that gauges how comparable bunches are, out of the grid X. This includes a few intervention and a few solutions exist. In single linkage bunching, the likeness between two gatherings is the base component xij, with i in one gathering and j in the other. Actually, the most extreme component X ij for vertices of distinctive gatherings is utilized within the methodology of complete linkage grouping. In normal linkage bunching one need to figure the normal of the X ij.

Progressive grouping has the playing point that it doesn't oblige a preparatory information on the number and size of the bunches. Then again, it doesn't give an approach to segregate between the numerous segments acquired by the methodology, and to pick that or those that better speak to the group structure of the diagram.[2]

2.5.4 Partitional Clustering

Partitional clustering shows an alternate famous class of strategies to discover bunches in a set of information focuses. Here, the amount of groups is reassigned, say k. The focuses are installed in a metric space, so that every vertex is a point and a separation measure is characterized between sets of focuses in the space. The separation is a measure of uniqueness between vertices. The objective is to discrete the focuses in k bunches such to amplify/minimize a given expense capacity focused around separations between focuses and/or from focuses to centroids, i. e. suitably characterized positions in space. Probably the most utilized capacities are recorded beneath:

1. Minimum k-clustering. The expense work here is the distance across of a group, which is the biggest separation between two purposes of a bunch. The focuses are characterized such that the biggest of the k bunch measurements is the most modest conceivable. The thought is to keep the bunches extremely minimal".

2. k-clustering total: Same as least k-clustering, yet the distance across is supplanted by the normal separation between all sets of purposes of a group.

3. k-center: For each one bunch i one characterizes a reference point xi, the centroid, and processes the most extreme width of the separations of each one group point from the centroid. The groups and centroids are self reliably picked such to minimize the biggest worth of distance across.

4. k-median: Same as k-focus, yet the most extreme separation from the centroid is supplanted by the normal separation.

2.5.5 Divisive Algorithm

A basic approach to recognize communities in a chart is to distinguish the edges that join vertices of distinctive groups and evacuate them, so the bunches get separated from one another.

Here edges are picked according to the characteristics of measures of edge centrality, assessing the noteworthiness of edges according to some property or method running on the outline. The steps of the computation are:

1. Computation of the centrality for all edges;

2. Removal of edge with biggest centrality: if there should be an occurrence of ties with different edges, one of them is picked at arbitrary;

3. Recalculation of centralities on the running diagram;

4. Iteration of the cycle from step 2.

2.6 Distributed Community Detection

There are numerous strategies accessible for brought together group location yet these are just helpful when the information must be dissected in a logged off mode. In this technique firstly the hints of the versatility examples are gathered and after that diverse structure of the portability are planned. These planned are further utilized within choosing a few things, for example, diverse sending procedures, distinctive efforts to establish safety and provisions. Then again in appropriated communities the neighborhood communities are caught by the mobiles when they interacted with one another. As self arranging maps they locate the communities themselves as opposed to depending on the

focal server. There are two separate routines accessible to discover dispersed communities.

1. K-clique
2. Modularity and Simple

Particularity and k-CLIQUE have marginally preferable execution over their SIMPLE partner. That is normal in most circumstances since they require more data and computation; particularly the processing many-sided quality of MODULARITY is O(n4) in the most pessimistic scenario, where n is the span of the system investigated in this way. Nonetheless, since a variable of n2 is helped by the assessment of every R, which truly is prone to be limited by O(k2) where k is the normal level of a vertex in the diagram, the most detrimental possibility execution is hence O(n2k2).[1]

2.7 Result of Literature Survey

Literature survey confines that there are so much reseach is going in the direction of efficient and fast file sharing, and for that a lots of efforts and new techniques are arising day by day. these techniques are rather based upon the routing or the bluetooth and different features that could be useful.
while talking of android phones it provieds different features such as Wi-Fi direct and bluetooth and most of the people those who are having smartphones are using android. hence the applications it supports are also a way higher than any other phone operatinf system. So with the help of such and adaptive operating system it is quite efficient to build a netwrok. A network that has no need for network connection as well as no need for the infrastructure to perform file sharing action.
Hence using the previous work the base of the new technique could be judged at the ground level and new technique could be build.

Chapter 3

Issues in Peer to Peer File Sharing in Pocket Switching Network

Pocket switching network is made up by smart phones or devices that comes in contact with each other but using this network and performing file sharing is a big issue because there is so much to consider and take into account while developing such an application. There are few things that has to be kept in mind before deciding the approach for the file sharing application that will use the android mobile network. Here are some shortcomings that are present in the real world applications:

3.1 Opportunistic Networking

In this kind of networking we need to take into account the time limit for which the devices came in contact as well as the forwarding that needs to be taken place at that time. The use of the contact time should be efficient enough so that it performs the action in an appropriate manner.We need to take advantage of the connectivity time so that more data could be transferred in limited amount of time.

3.2 Mobility

Mobility patterns of nodes affect the speed, throughput, and reliability of data dissemination in opportunistic network. Thus, understanding the real mobility is vital for designing and evaluating protocols over mobile peer-to-peer networks. In particular, designing mathematical synthetic mobility model is desired for opportunistic network research.

3.3 Forwarding and Cache Strategies

To design an efficient forwarding and caching strategy for information dissemination in a time-variant intermittent-connected wireless networks. Due to node mobility, the network connectivity is highly dynamic and unpredictable. In mobile peer-to-peer network, often there is no end-to-end path between the source and destination. Classic end-to-end proactive and reactive unicast/multicast routing protocols may not work efficiently, because routing tables need to be updated and exchanged between neighboring nodes frequently (due to the time-variant network connectivity).[10]

These produce a large amount of control traffic which may dramatically slow down the network throughput. Instead, traffic is relayed by encountered nodes in a hop-to-hop basis from source to the destination. At each contact between two nodes, each of the two nodes locally decides which data to relay for both their own interests and other nodes interests. Apparently, the opportunistic network is a resource-constrained network in the sense that: each node meets other nodes only from time to time and the inter-contact time between the same pair of nodes can be very long; during each node meeting, a pair of nodes only have limited contact time before they move away from radio range thus can only exchange limited amounts of data; nodes may be only willing to share limited power and cache for helping to disseminate information for the public good. Thus, content forwarding and cache management is essentially a distributed resource allocation problem that should optimize network resource usage for best possible Quality of Service (QoS) of end users.[10]

3.4 File Discovery

The difficulties of an agreeable record imparting framework are in

> both record disclosure and document download. Document revelation (seeking for metadata) is dissimilar to looking on the Internet, which might be actualized midway. To the best of our learning, document revelation (metadata looking) in the DTN (as opposed to sending inquiries to the Internet through DTN nodes) has not been

examined in the recent past, and it is trying to empower proficient seeking on DTN nodes, which might be segregated from the netwrok.[9]

3.5 Capacity

Capacity of the network is also a major concern in this senario because the network. how much replication of the packets need to be done is a big issue. because dependent on that the bandwidth of the network is been consumed. As well as the packets that are been forwarded the path should be optimized enough so that the packets that are transmitted should not create congestion.

Chapter 4

Implementation

File sharing provides the facility to the users to share the different file they have with other users. Peer to peer file sharing provides the advantage the client can act as server as well as client. It could be the source of the file at the same time it can download the file from other classes as a client.
In this application we have used Wifi-Direct. Wifi-Direct does not require any wireless access point to connect with each other. This is relevant for applications that share data among the users such as file sharing , photo sharing or multi layer gaming. it has the advantage of the wifi ad-hoc as well as the Bluetooth. it provides higher range for the data transfer as well as the speed of the transfer compared to the Bluetooth is quite high.

4.1 Android

Android provides Wifi-Direct in the versions above 4.0. This version provides multitasking environment ,deep interactivity and powerful new ways for communication and sharing. It also supports Wifi-Direct for more reliable connectivity with out any internet or tethering. It lets us connect directly to the nearby peers. Android has different components to provide different functionality. Android apps are written in JAVA programming language. and android SDK tools compile the code with the data and the resource files into an APK an android package which is an archive file with an .apk suffix. Apk contains all the content of the android application and use to install the application on the device.

Why Android

- Each application has its own virtual machine which is isolated from other application.

- It has the principle of least privilege it means by default each application can use only those components that it require. Application cannot access the part of the system for which it does not have permission. that creates very secure environment.
- It supports wifi-Direct and its possible to two apps to share the same Linux user ID, in which case they are able to access each others files.
- An application request permission to access the device data such as mountable storage(SD Card), Camera, Bluetooth etc.
- Android's market is 81.3 % that's the highest in comparison with windows, ios and other mobile OS. That makes it convenient to approach more audiences for the application.

Components of Android Application

There are different components of android application, each of them serve different purpose. The life cycle of these components defines how they created and how get destroyed. So here some details about the components :

1. Activities: Activities interact with the user so activity presents you the screen to interact with.Activities in the system are manages on the stack whenever the activity started its placed on the top and come in the running stack the previous activity cannot be resumed until the top running activity got finished. Activity is implemented as the subclass of activity. Activity has different methods.the full life time of the activity is from oncreate() to ondestroy(), visible lifetime is from onstart() to onstop() and forground lifetime is from onpause() to onresume().
2. Services: Services runs in background to perform the operations. if any other application is running on the system the service continues to run in the background. services perform interprocess communication. to create the service we need to implement the subclass of the service class. Some methods are present in the service class we need to override to create the service. StartService(), Stopservice() are the methods to invoke services.
3. Content Providers: Content provider manage the access to the data. Content providers are the standard interface that connects data in one process with code

running in another process. There is no need to create the content provider for the application if there is a need to share data, it could be established with the implementation of the subclass of content provider.

4. Broadcast receivers: To receive the system wide broadcast announcements as well as for providing the response towards it we use broadcast receiver. to use this we need to implement the subclass of the broadcastreciver.

5. Intents and Intent filter: To request any action from the application we need to pass intent as an object. there are two types of intents implicit and explicit.Intent object carries the information which components to start with in the application.

6. Manifest: This should be present at the root of the directory. it defines the permissions that are required for the application. This defines the API level require for the application. This application works on the API level Equals to or greater then 14. With all that it defines the software as well as the hardware requirements of the application.

7. Resources: Application interface requires images as well as layouts these are defined in resources. In this project we are using two layouts "file.xml" and "main.xml". the layout files are defined in xml. layout files are present as the user interface for the user. It consists of different tabs, buttons and textfields.

8. Generated Java Files: There are two autogenerated java files present in the system. First is R.java and second is Buildconfig.java. R.java file contains the unique identifiers for the resources for the easy access of the elements. while buildconfig.java has a Debug constant thats automatically set according to built type.

4.2 Android API level 14

The difficulties of a helpful document imparting framework are in

Wi-Fi shared (P2p) permits Android 4.0 (API level 14) or later gadgets with the suitable fittings to unite specifically to one another through Wi-Fi without a halfway get to point (Android's Wi-Fi P2p skeleton conforms to the Wi-Fi Alliance's Wi-Fi $direct^tm$ affirmation program).using these API's it is not difficult to uncover and interface with the gadget with the same wifi help. The correspondence is more rapid and more separations

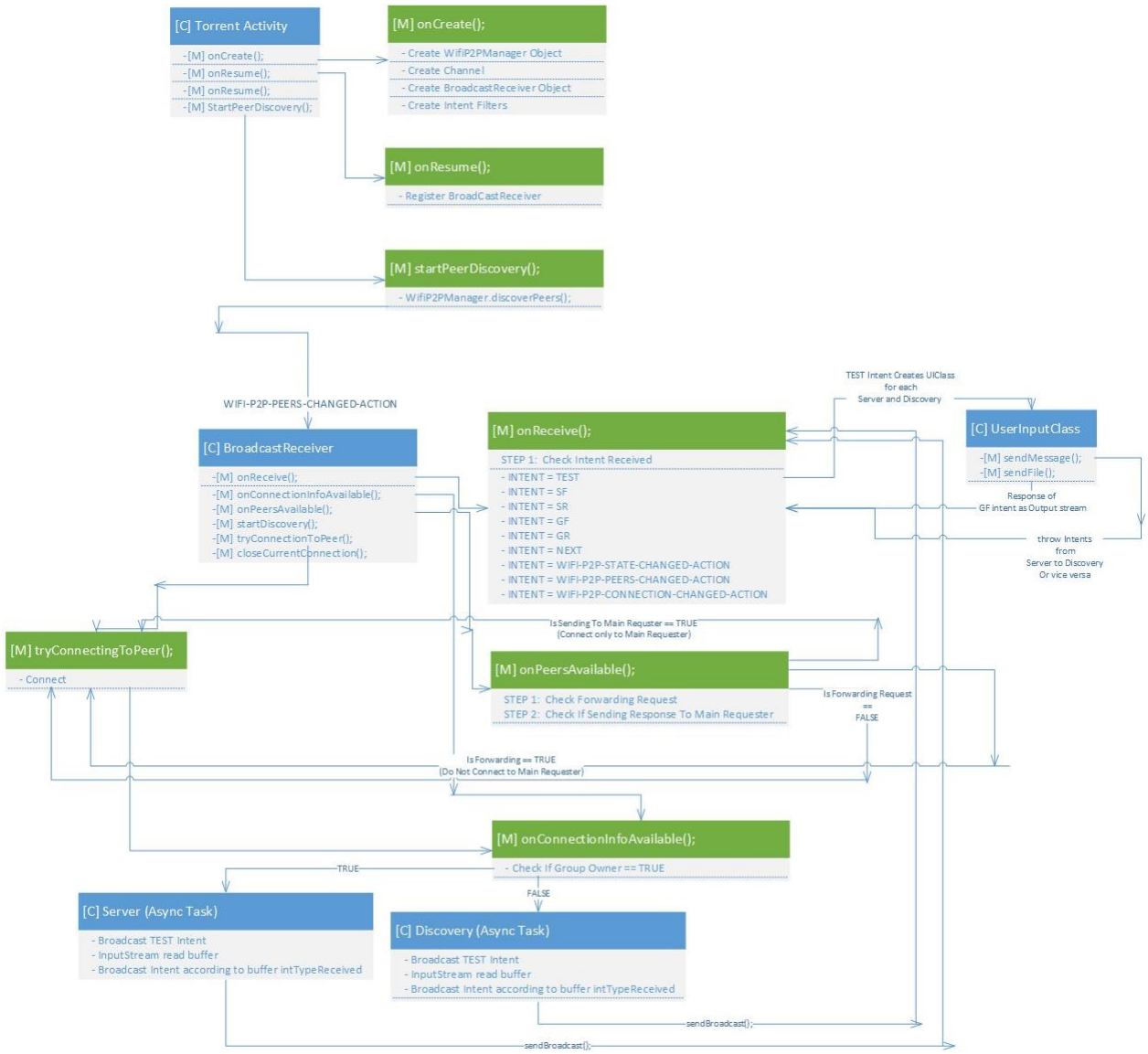

Figure 4.1: Phases of Application

then Bluetooth. This requisition works with API level 14. In spite of the fact that API level 14 gives so much offices yet in the meantime it has some drawbacks.at a solitary time you can associate with stand out companion and in the event that its multihop document imparting then the record need to be exchanged firstly to the gathering manager then to the gathering holder will advance the document to the customer.

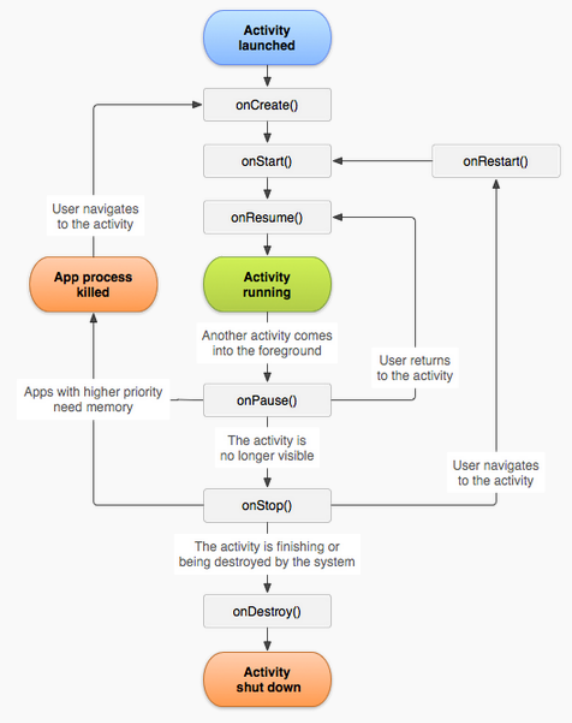

Figure 4.2: Life Cycle of an Activity

4.3 Application Flow

4.3.1 Activity Class

This activity includes so many phases. Normally an android activity life cycle looks like that. In this application we have TorrentActivity.java class that extends the activity class. This application have only this activity. It initialize various components of the application, like WifiP2pManager, Channel, IntentFilter and WifiDirectBroadcastReceiver.Now the working of different class.

As we know the activity includes such life cycle our activity also has different methods define.

- onCreate(): Two different objects are created of different classes. First is WifiP2pManger class, Second is BroadcastReceiver class.With that Channel and intent filter is created.

- Onresume(): We register Broadcastreceiver. Discription of broadcast receiver is defined next.

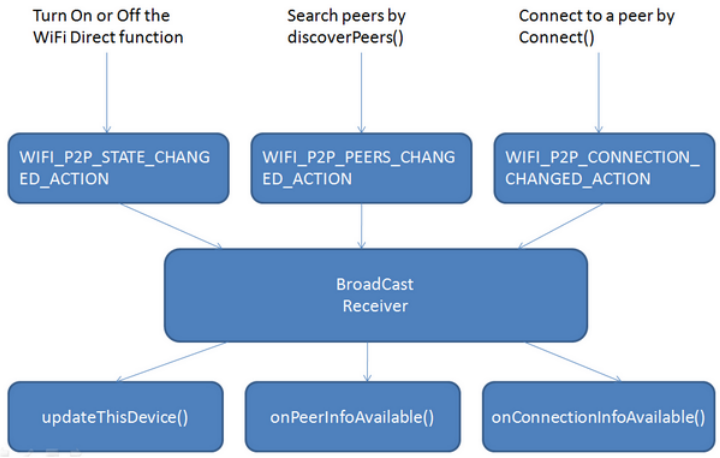

Figure 4.3: Broadcast Receiver Class Intents and Methods

- StartPeerDiscovery(): When this method is invoked WifiP2pManger.discoverpeer() method is called. This passes an intent to the BroadcastReceiver class.

4.3.2 BroadcastReceiver Class

We have to have a show recipient class in light of the fact that a BroadcastReceiver permits to accept goals show by the Android framework, so requisition can react to intents that we are intrigued by. The essential steps for making a telecast recipient to handle Wi-Fi P2p expectations are as takes after:

- Make a class that enlarges the Broadcastreceiver class. For the class constructor, we must have parameters for the Wifip2pmanager, Wifip2pmanager.channel, and the action that this show collector will be enlisted in. This permits the telecast beneficiary to send redesigns to the action and in addition have entry to the Wi-Fi fittings and a correspondence station if necessary. Wifip2pmanager is really essential and here are the strategies, interfaces and the goals characterized under that.

WifiP2pManager

The primary class we need to work with is WifiP2pManager, which you can acquire by calling getSystemService (WIFI_P2P_SERVICE). The WifiP2pManager includes APIs that allows to:

- Initialize your provision for P2p associations by calling initialize().
- Discover close-by gadgets by calling discoverpeers().
- Start a P2p association by calling connect().
- The Wifip2pmanager.actionlistener interface permits get callbacks when an operation, for example, uncovering companions or interfacing with them succeeds or falls flat.
- Wifip2pmanager.peerlistlistener interface permits get data about uncovered companions. The callback gives a Wifip2pdevicelist, from which you can recover a Wifip2pdevice object for every gadget inside reach and get data, for example, the gadget name, address, gadget sort, the WPS arrangements the gadget helps, and that's just the beginning.
- The Wifip2pmanager.groupinfolistener interface permits accept data around a P2p bunch. The callback gives a Wifip2pgroup object, which gives bunch data, for example, the manager, the system name, and passphrase.
- Wifip2pmanager.connectioninfolistener interface permits accept data about the current association. The callback gives a Wifip2pinfo object, which has data, for example, whether a gathering has been structured and who is the gathering holder.
- We have to have admittance to the distinctive assets of the framework henceforth we have to get consents, for example, ACCESS_WIFI_STATE,CHANGE_WIFI_STATE,INTERNET.
- A channel that interfaces the requisition to the Wifi p2p structure. An occasion of Channel is acquired by doing an approach initialize(context, Looper, Wifip2pmanager.channellister
- In the telecast recipient, check for the plans onreceive().there are distinctive goals that could be accepted. we have to perform essential activities relying upon the goal that is accepted. Firstly the purpose WIFI_P2P_STATE_CHANGED_ACTION will be accepted and afterward Onpeersavailable() we have to check if the associate is sending the solicitation or the this is the first ask for in the event that its an unique ask for then the reaction might be distinctive. After that tryconnectingpeers() system conjures. This will prompt interface the gadgets.

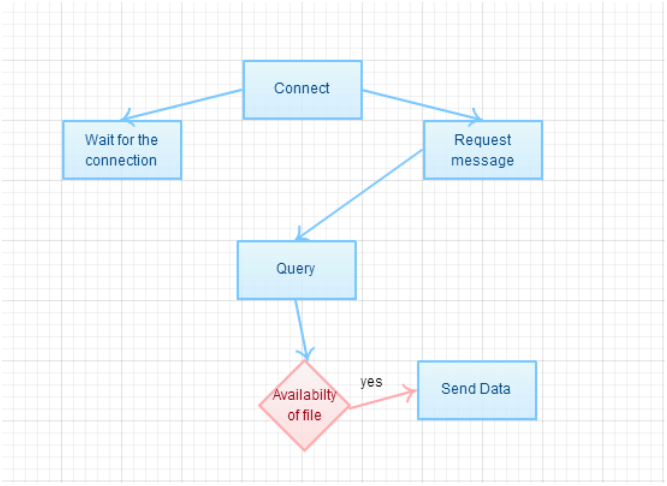

Figure 4.4: Client Server socket creation

- If the association will secured then We have to check whether the gadget is the gathering holder or not. This might be resolved with the utilization of Onconnection data Available. In the event that the gadget is the gathering manager then the that will be exchanged to the server class and if not then the reaction will go the revelation class.

4.3.3 Server & Discovery Class

This class will create a sever and client socket at the specified port and blocks it until it receives the connection response. This class extends the AsyncTask class This will make it run in background with the other methods running.When the client invokes the accept method the connection will establish. Working of the server and client class.

- We have created some custom intents in the activity. TEST intent is one of them. This intent makes the connection process validate and then the connect button will appear on the screen. The TEST intent of the server class will be received by the client class and vice-verse.

- TEST intent will create the UserInput class for client and server.

4.3.4 UserInputClass

User input class is created as the response of the test intent.

- AS soon as the TSET intents work will finish the UserInput class will be created and the searchFile() option will appear on the screen. As soon as the user will input the text to be searched then the next custom intent will invoke that is "SF". SF will look for the particular file in the sdcard's application folder .

- SR intent will be the search response of the particular search. If the response will be true then the next intent will be thrown that is 'GF' that get file.

- GF will be followed by get response. This intent will action by sending the file.

Sending File

Sending and receiving file would be done on the basis of network coding. Network coding works as an alternative for routing. We are using random linear network coding. RLNC is performed by following these steps.

- Firstly we included the ncutils library that an online source available for the random linear network coding. This could be done by adding NcUtilsCodec library in the user defined libraries. this library gives a usage of irregular system coding over $F2^8$. This library is quicker than the above however is less adaptable. It incorporates local code to quicken both encoding and deciphering.

- This Java library might be utilized to execute backing of arbitrary network coding in a Java provision. The API is exceptionally straightforward. On the off chance that the stage is underpinned the library utilizes JNI and a local library to quicken encoding and interpreting else it falls again to an unadulterated Java execution.

- This library has so many classes. Like FiniteField. This class returns the finite field for the 2^8. This performs addition, multiplication, inverse and division operations to calculate the finite field.

- VectorHelper class will Multiplies vector stored in src by coeff and adds to vector stored in destination. The result is saved in destination.

- java encoder this will encode the packet will be divided into segments and the encoding will be performed. The encoder will be created at the source. this will be transferred to the output stream.

- java decoder this will decode the packet at the destination.It will decode the packets received from the input stream and then reform the original file.

- Advantage of using RLNC: block propagation scheduling issue could be generally improved when RLNC is utilized, in light of the fact that a associate can recoup the first substance in the wake of getting n directly free blocks. RLNC successfully improves the substance dissemination convention by dispensing with the requirement for information reconciliation. RLNC makes all coded squares similarly paramount. From the perspective of data entropy, each coded square sent into the system is novel however holds an equivalent measure of data entropy, since all servers and associates perform randomized system coding on all gained squares. Subsequently, an associate does not have to hunt down a specific chunk.as a result, the normal download time could be diminished altogether. An alternate attractive property is that RLNC can upgrade the framework vigor to server and associate takeoffs. Without RLNC, just a little rate of companions can complete the process of downloading in the event that the server leaves the framework early.

4.4 Multihop Communication

When the SF intent is passed and the response that we receive is SNR that search negative response at this time we get to know that the device we are requesting the file from does not have that file. hence we had received a negative response. Negative response leads us to the new scenario that is multihop communication in file sharing.

Suppose we have three devices A,B and C. Device A is in the range of device B and device B is in a range of device C. Now device A requests the file and B does not have it then device A will receive a negative response from the device B. At a single time the device can connect to only one device hence to forward the request device B will terminate the connection with device A by invoking the closeconnection() method.

As the result to the SNR intent the source device's MAC address and search request will be stored by device B and then it will forward the connection request with the same process of one to one connection. But as it is in the forward mode hence after the TEST intent directly search response will be passed and the query will have the same file that

was requested by A.

AS soon as SR intent means search response intent will be true then GF get file intent will be called and device B will buffer the file and after that terminate the connection.

On the basis of the Mac address it will request A again and will pass intent TEST and then will pass the GF get file intent and will pass the whole file to the source. In this manner the whole file will be transmitted to the source and original requester of the file will get the file.

Chapter 5

Results

The main objective for making this application was to built an application that provides the peer to peer file sharing without any central server and internet connection. To use the enhanced feature of android that is Wifi-Direct in such a way that the application will be simple, fast and useful for the users. This application not only works for one to one connection but also for multihop connectivity.
Wifi direct technology has been reserched and well grasped through studies. An application is developed to study purpose. P2P connection maintained for the discovery, connection and data trasfer. With the help of this document the user could understand the background process that is running behind the main user activity or the user interface.
Though the API level 14 has so many drawbacks and so many are not properly implemented. But with the upcoming higher API levels its accepted with it Wifi-Direct will come as new and relevent technology for the data sharing with out any use of internet as well as centralization.

Here are some results that are considered with this application.

- Distance : With respect to bluetooth definately wifi direct provies better results,We calclulated the distance with two parameters one is with line of sight and other is in the presence of obstacles. We found out that the application works better and provides good range within line of sight, It works upto 95 to 100 meters properly when we are transferring the data and in the case of obstacles it works uptp 70-75 meters.And with the help of multihop this is feasible to extend this range.

- Data Transfer rate: Data transfer rate is also high in comparison with Bluetooth. That is upto 33mbps, that is quite high in comparison with Bluetooth.

- Security: Wifi Direct relies on WPA2 security which uses AES 256 bit conversion. Hence it uses key based encryption and provides enough security for an average user.

- Power Saving:Wifi-Direct has power saving option to use the system resources in an efficient way.

- Metadata: Files with the .doc,.pdf, .txt, .jpg, .png, .mp3, and .mp4 can be transferred through this application. Through this you can transfer the images, audio , vedio and text any kind for files.

5.1 Issues

As there is nothing that comes without flows this application also has some flaws. Some that are within the inbuilt system functionality and some are in the implementation.There are some issues that we have faced and still need to be resolved.

- Connectivity: As we all know that wifi-directs stops after 5 minutes andkeeps it self off automatically to save the power of the device it is an inconvenience in running this application for multihop. Again and again we need to go for the settings and need to make it on for the further transfer.

- Disconnecting group: Only the group owner can remove the group any other device in the network cannot initiate the disconnect method.

- No Reference : As till now applications that are available in market works only on one to one file sharing hence it leads to no reference to detect the flaws that appears during the run time even the developers site does not provides information about the multihop transmission.

- In progress: Many methods that are available in one API level has deprecated in the higher version hence it is difficult to built the application for all the API levels at tha same time. We are bounded to use a single API level for the implementation and that leads to approach lesser audiences.

Chapter 6

Conclusion

This application was built with a motive to research about the peer to peer file sharing application that can work on android and with out any internet and centralization could perform file sharing with better results. With the use of network coding to work in the area where the transfer need to be done on the basis of opportunity. The application is built successfully and all the aspects of android, Wifi-Direct and network coding are studied and implemented.Hence we can say that the goal is achieved.

We finally have an application that performs data transfer on an efficient transfer rate and for longer distance. The results are discussed primarily that shows that there are still a lot that could be done in this area and in this application only. We studied the limitations of Wifi-Direct but at the same time explored the benefits that we could take with this technology. Peer to peer file sharing with that has opened new doors for various data transfer application.

Network coding makes it efficient to work in the opportunistic scenario where the packets are highly probable to lost. where the status of the peers keeps on changing from minute to minute. In this this provides us higher bandwidth utilization makes the transfer faster.

Hence with the help of all the technologies and research in data transfer mechanism this application makes an efficient use of all the technologies.

Chapter 7

Future Work

Developing the application in which the file sharing could be done with the least delay and with no centralized sever and storage facility. Community detection would be applied that will help in finding the destination as soon as possible; because of this the security could also be maintained. The intermediate nodes in the forwarding path could be trusted and the optimization could be performed.
The main focus will be on the destination path detection. Secondly, the time analysis in which the destination remains on so that the packet could be forwarded to that, with the help of this data loss could be managed.
And at last finding solution to the energy consumption because keeping the Wi-Fi and Bluetooth on consumes a lot of energy. A better solution for that must be provided so that the user can easily use the application most of time without being concern about the battery.

Making advancement in the time to keep the wifi-direct live so the application does not remain bound to the time that t provides for the fie sharing.

Bibliography

[1] Ha Dang and Hongyi Wu., "Clustering and Cluster-Based Routing Protocol for Delay-Tolerant Mobile Networks". In 'TRANSACTIONS ON WIRELESS COMMUNICATIONS, VOL. 9', University of Louisiana at Lafayette, Lafayette, LA, June 2010. IEEE.

[2] Santo Fortunate. "Community detection in graphs". Complex Networks and Systems Lagrange Laboratory, ISI Foundation, Viale S. Severo 65, 10133, Torino, I-ITALY, 2010.

[3] Illes Farkas Gergely Palla, Imre Derenyi and Tamas Vicsek."Uncovering the overlapping community structure of complex networks in nature and society". Bupdapest,Hungary, 2005.

[4] Sindhura Tokala My T. Thai Nam P. Nguyen, Thang N. Dinh. "Overlapping Communities in Dynamic Networks: Their Detection and Mobile Applications. USA, 2011". University of Florida.

[5] James Scott Richard Gass Jon Crowcroft Pan Hui, Augustin Chaintreau and Christophe Diot. "Pocket Switched Networks and Human Mobility in Conference Environments". Philadelphia, PA, USA, 2005. University of Cambridge.

[6] Shu-Yan Chan Jon Crowcroft Pan Hui, Eiko Yoneki. Distributed ."Community Detection in Delay Tolerant Network". Kyoto, Japan, 2007. University of Cambridge

[7] Christos Gkantsidis and Pablo Rodriguez Rodriguezi."Network Coding for Large Scale Content Distribution".Microsoft Research Cambridge, CB3 0FD, UK.

[8] Sewook Jung,Alexander Chang, Dae-Ki Cho and Mario Gerla. "BlueTorrent: Cooperative Content Sharing for Bluetooth Users".Department of Computer Science, 2009.University of California, Los Angeles, Los Angeles, CA, USA.

[9] Armir Bujari, Claudio E. Palazzi, Daniele Bonaldo Universit degli Studi di Padov. "Performance Evaluation of a File Sharing DTN Protocol with Realistic Mobility"., USAPadova, Italy

[10] Amir Krifai.Towards better content dissemination applications for Disruption Tolerant Networks".of the University of Nice - Sophia Antipolis,2012.Computer Science

[11] Salvatore Spoto, Rossano Gaeta, Marco Grangetto, Matteo Sereno,"BitTorrent and fountain codes: friends or foes?" .Department of Computer Science University of Turin. Turin, Italy

[12] Jussi Kangasharju,Keith W. Ross,David A. Turner,"Optimizing File Availability in Peer-to-Peer Content Distribution".Dept. of Computer and Information Science,Polytechnic University.Brooklyn, NY