**CSE 5345-Fall 2016**

**Project : Wireless P2P File Sharing**

Submitted by: Parth Shah (1001246682)

Rohit Chawla (1001234979)

Shivesh Ranjan Singh (1001231391)

**Introduction**

Wi-Fi Peer-to-Peer network for file sharing allows Android devices to connect directly to each other without any intermediate access point. Using API we can discover and connect to other devices when each device is connected through Wi-Fi. This has longer range than Bluetooth and communicate efficiently over larger distance. This is quite useful if you want to share data among users, such as videos, ringtones, photos, multiplayer game etc.

In this we have two major components: -

**Peer**:Nodes containing the data. It can join and leave network at it’s own will.

**Tracker:** Maintains a list of Peers in network in real time and shareable files available at each peer. The list of pair is dynamic and is updated when a peer joins or leaves the network.

The tracker has complete list of peers in network and its shareable files and any peer in the list can fetch the list of files and can download it on its local as in when required. Each peer has file transfer running as background service. It can request any file directly from other peer.

**Motivation**

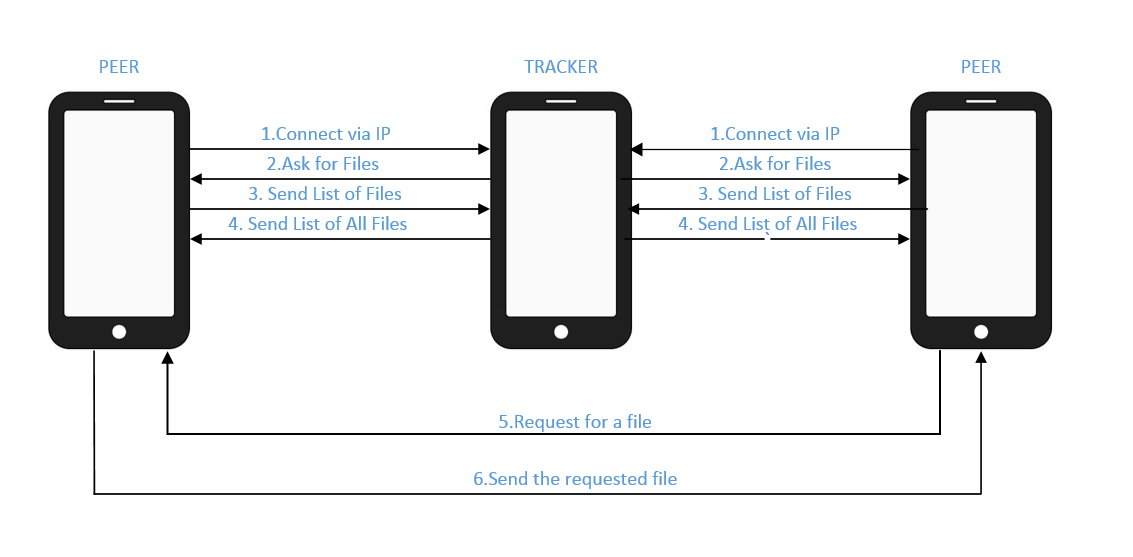
Technology is evolving day by day and is becoming more and more convenient and user friendly with each passing day. It's the nature of science to always be advancing, and so the continued application of constantly evolving science drives the change in technology. We had Bluetooth for file transfer between different android mobile. Bluetooth has limitation of distance so with Wi-Fi we can surely overcome this limitation and can transfer wide range of data in short span of time. With the help of this project we can transfer large video files, pictures, ringtones and other media files in an effective manner

**Problem Statement**

To implement a wireless P2P network with peers and tracker for file sharing using Wi-Fi methodology. Tracker is responsible to maintain the list of peers and shareable files on files. Once the connection is established, any peer can request for a file stored on any other peer and it should be downloaded on the requesting peer. Kindly find the requirement as below: -

* Tracker maintains the real time list of all peers
* Peers can leave and join the network at their will. Tracker has to update the list in such condition.
* If a peer leaves the network when other is downloading from it then application cancels the download in between and error message is generated.
* Peers notifies tracker of the shareable files which shares this info with all the peers connected in network.
* Peers should maintain a list of all the files in the network and any file from different peer can be successfully downloaded in it.
* Downloaded file is saved successfully in file structure of the requested peer.

**Architecture (Design)**



ARCHITECTURE EXPLANATION

It contains 2 important members – Peer and Tracker

* First step is to establish 2way communication between peer and tracker by exchanging IPs and making the connection
* Now Tracker requests connected peers for their local files
* Peers send their list to Tracker who combines the list from all peers.
* Tracker sends the combined list to all the connected peers
* The tracker has a button which can be used to show number of peers connected.
* All the peers are also connected to each other and can share or download as desired.
* When a peer leaves the network then a notification stating the IP address of the peer that left.

**Technology used and Implementation (short)**

* Java
* XML
* Android 4.0 or later devices
* appcompat
* Socket Programming

We have implemented this using socket programming in Android. We have 2 projects developed, one for the peer and the other for the tracker.

Peer: The peer project contains all the data need at the peer side. It has 4 classes in it. Initially the peer will ask to enter the IP address of tracker. After we enter it, the peer connects to the tracker. A thread is started for sending its own files and for receiving files from the tracker. This happens as soon as the peer connects to the tracker. The peer will now have the full file list including shareable files from all peers. Peer 1 can connect to peer 2 to download a file.

Tracker: The tracker file has only one java class. Its main activity is to track the peers and download file list from each one of them. The complete file list is sent it all other peers in the network as well.

We have used socket programming to connect to various devices. Simple classes of socket programming are used. Network Interface is used to detect the various network interfaces.

A client thread is also used to send files to tracker.

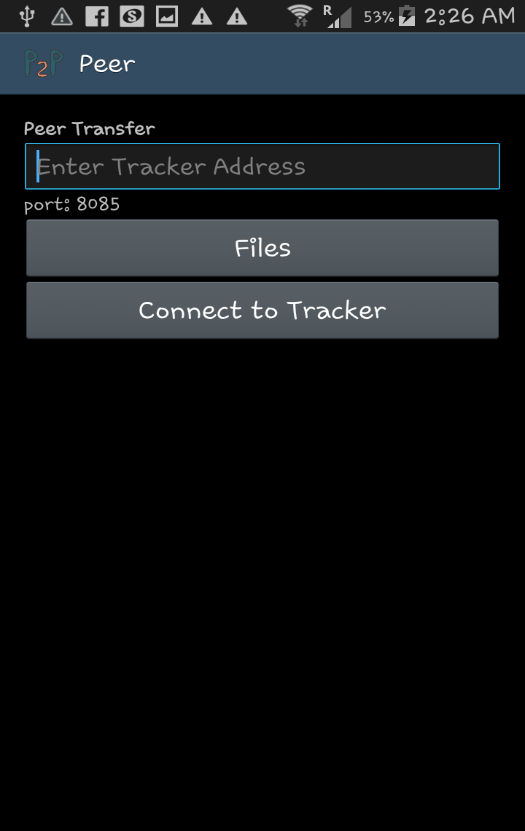
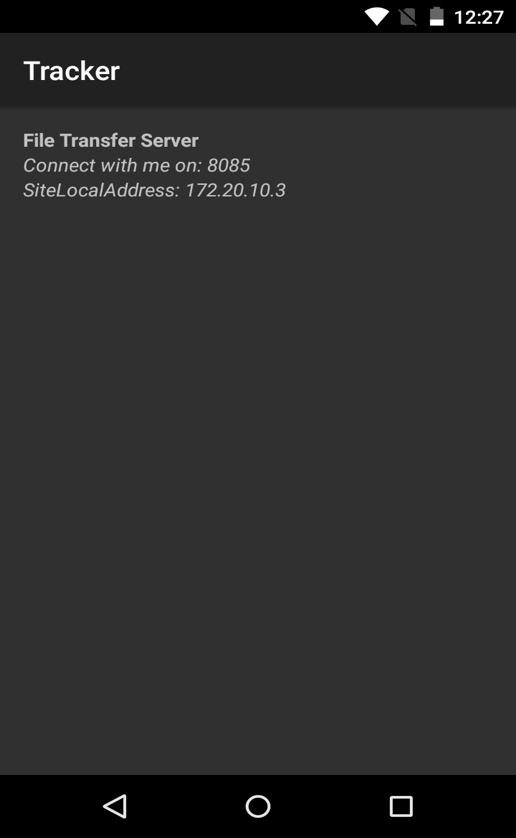
Our program requires android-support-v7-appcompat for compatibility purposes. The rest all can be run by downloading android-sdk and installing it on eclipse. You can go to the market place and download ADT. Use sdk manager to run programs.

**We have also successfully implemented transfer of video from peer to peer.**

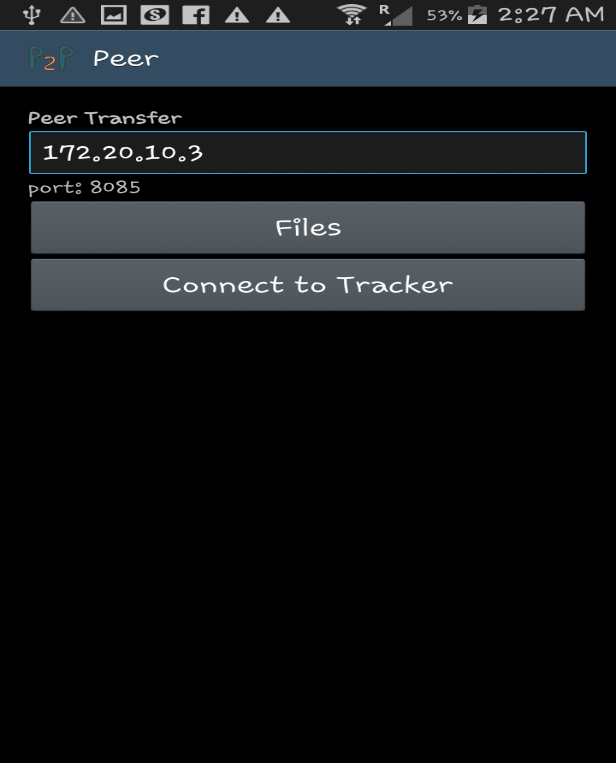
**Additionally, we have also kept track of number of connected peers to the tracker in real time. The Tracker shows how many number of peers are connected to them by the click of a button. When a peer leaves the system a notification with the peers IP address stating it has left the network.**

**Work Flow (screen shot of sequence of screens in the work flow as was shown in your demo.)**

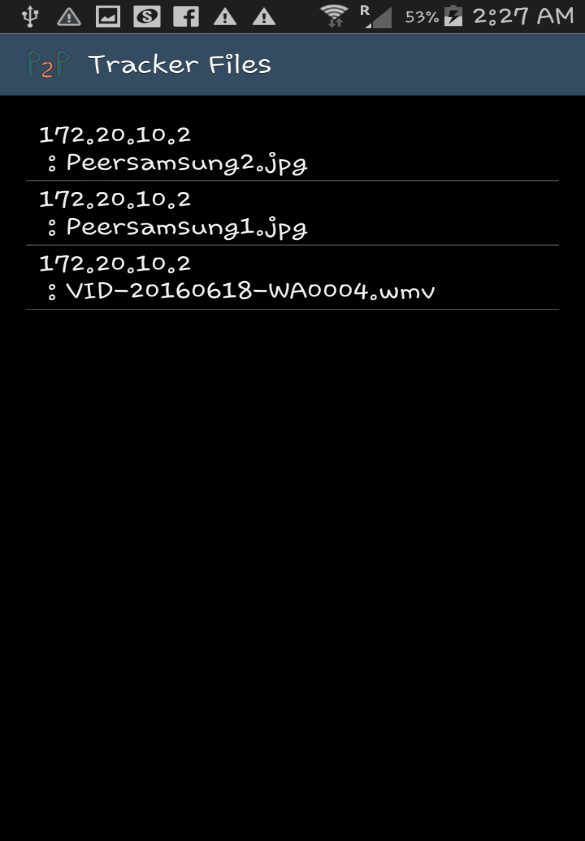
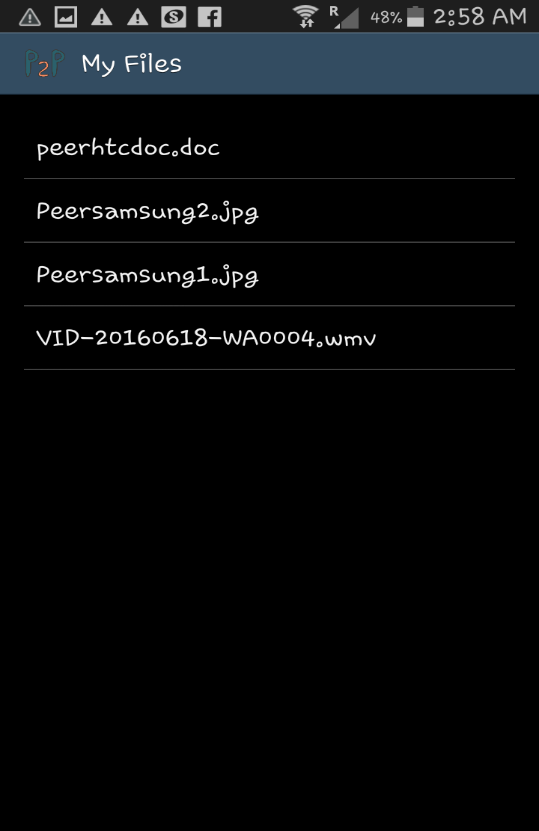
Step 1 -> Initial stage of Peer and Tracker

Step 2-> Establishing Peer-Tracker connection by IP



Step 3-> After successful connection, transfer file and check downloaded file in Peer.

Peer (before transfer) Peer (after transfer)

****

Tracker (After transfer)

**Challenges**

We were newbies for Android and we faced lot of difficulty in getting an understanding how Android architecture works. What are different commands ­­­­­for performing a task and how to utilize them in building up our code.

Wi Fi connectivity behavior was not consistent as we need to reconnect peers and tracker again and again to maintain connections as it was lost very inconsistently.

**Conclusion**

Wi-Fi peer to peer file sharing methodology seems to be much more effective and faster mode of transfer than previous methods available. We can share large video files too and Bluetooth restriction of short ranges is also overcome. This is very effective for application that share data among users, such as multiplayer game or a photo/video/ringtone sharing applications. Also P2P is more reliable as central dependency is removed. Failure of one peer doesn’t affect the functioning of others. It’s good for to connect small number of devices where high level of security is not required.

**References**

·        https://github.com/koush/android-support-v7-appcompat

·        https://developer.android.com/guide/topics/connectivity/wifip2p.html

·        https://developer.android.com/reference/java/net/Socket.html

·        https://developer.android.com/reference/java/net/Socket.html

·        https://github.com/cpsrnidh

·        http://stackoverflow.com/questions/31146788/testing-sockets-example-on- android-studio

·        https://stackoverflow.com/

·        https://github.com/koush/android-support-v7-appcompat