File Sharing Android App

Contributors

Tejas Ladhani
19IT058

Shrey Makwana
19IT059

ABSTRACT

This paper presents an android application to share different types of files with other android devices using WIFI (WIFI direct).

INTRODUCTION

The File Transfer Application is a simple android application that can be used to transfer your files from one android device to another android device in close proximity. The idea behind this application is to make transfers easy and fast for people having internet issues.
This app solves the problem of transferring files from one android device to another android device without the internet. It eliminates the old Bluetooth transfers and transfers files using wifi which provides much faster data transfers. You can share music, images, documents, etc without any mobile data usage.

TECHNOLOGY:

This application works on P2P(Peer To Peer) networking technology.

P2P networking is a type of networking where there is no need for a centralized server , each node works both **as server and client.**
The individual users in this network are referred to as **peers**. The peers request for the files from other peers by establishing TCP or UDP connections.Each node has a particular ID assigned, by which they can identify internally.

When one peer makes a request, it is possible that multiple peers have the copy of that requested object. Now the problem is how to get the IP addresses of all those peers. This is decided by the underlying architecture supported by the P2P systems. By means of one of these methods, the client peer can get to know about all the peers which have the requested object/file and the file transfer takes place directly between these two peers.

Working/Steps :

Enabling - Disabling - Discovering WIFI :

We have used WifiP2pManager class provided by Google Android.This class provides the API for managing Wi-Fi peer-to-peer connectivity. This lets an application discover available peers, setup connection to peers and query for the list of peers. When a p2p connection is formed over wifi, the device continues to maintain the uplink connection over mobile or any other available network for internet connectivity on the device.

So after , discovering other devices (Client/Server), we need to establish the connection between the two.

Server- Client connection:

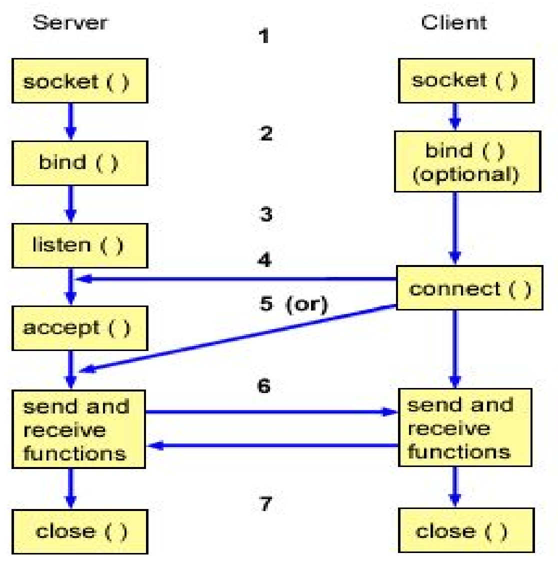
(Sending requests and getting response)

Client will send a request to the server , to which the server will respond .We have achieved this feature by socket programming in our app.

Socket :

A network socket is a software structure within a network node of a computer network that serves as an endpoint for sending and receiving data across the network. The structure and properties of a socket are defined by an application programming interface for the networking architecture.

Sockets allow communication between two different processes on the same or different machines.



Following is the flow of events for a socket:

1. The socket() API creates an endpoint for communications and returns a socket descriptor that represents the endpoint.

2. When an application has a socket descriptor, it can bind a unique name to the socket. Servers must bind a name to be accessible from the network.
3. The listen() API indicates a willingness to accept client connection requests. When a listen() API is issued for a socket, that socket cannot actively initiate connection requests. The listen() API is issued after a socket is allocated with a socket() API and the bind() API binds a name to the socket. A listen() API must be issued before an accept() API is issued.
4. The client application uses a connect() API on a stream socket to establish a connection to the server.
5. The server application uses the accept() API to accept a client connection request. The server must issue the bind() and listen() APIs successfully before it can issue an accept() API.
6. When a connection is established between stream sockets (between client and server), you can use any of the socket API data transfer APIs. Clients and servers have many data transfer APIs from which to choose, such as send(), recv(), read(), write(), and others.
7. When a server or client wants to stop operations, it issues a close() API to release any system resources acquired by the socket.

Programmatically:
To achieve it programmatically ,
we will use

java.net.Socket class: represents the socket between the client and the server

java.net.ServerSocket class : provides a mechanism for the server application to listen to clients and establish connections with them.

(will add as we proceed)

FEATURES:

1.Enabling /Disabling WIFI and Hotspot .

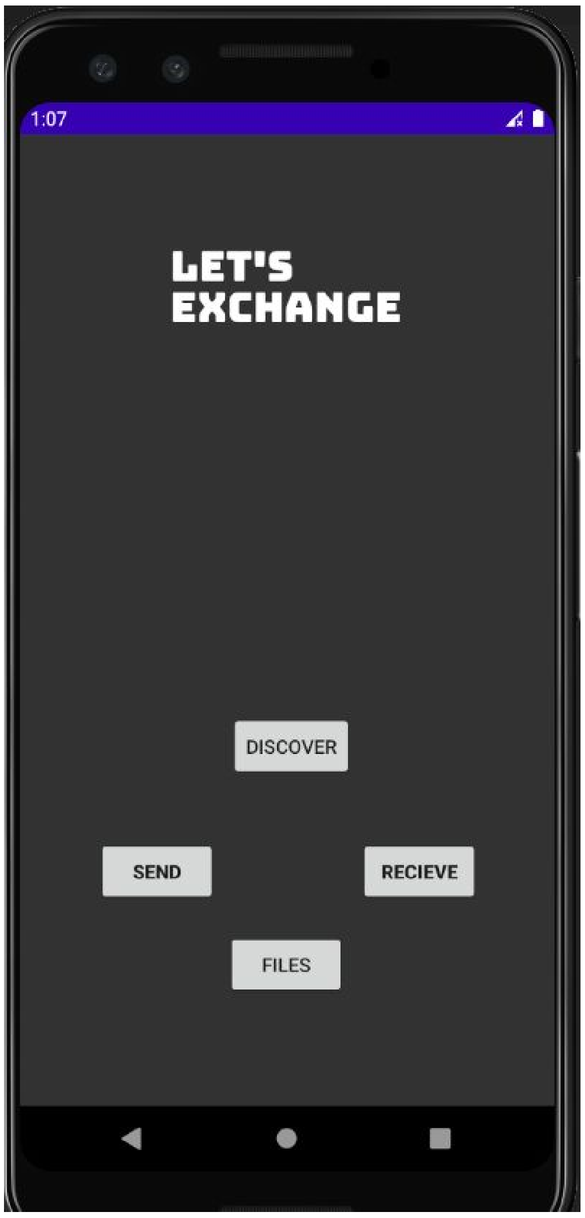
2. Discovering and displaying the nearer devices.

3.Establishing connection between the devices.

4. Accessing and displaying the files stored in external(shared storage space) storage .

5. Selecting single or multiple files (with different valid extensions)

6.Sending selected files to other devices.



CONCLUSION:

In the paper we have mentioned how our application works and what it is capable of doing. It provides detailed insight into what technology and what kind of API’s we are using to accomplish with our application.
Till now we have learned a lot about how android works and this project has been quite a learning experience for both of us, and we’ll try to learn as much as possible from this project.

REFERENCES

https://developer.android.com/reference/android/net/wifi/p2p/WifiP2pManager

https://www.geeksforgeeks.org/socket-programming-in-java/

https://developer.android.com/guide/components/intents-common

https://www.ibm.com/support/knowledgecenter/ssw_ibm_i_71/rzab6/howdosockets.htm

https://developer.android.com/guide/topics/permissions/overview

https://developer.android.com/guide/components/fundamentals