IJREAT International Journal of Research in Engineering & Advanced Technology, Volume 3, Issue 3, June-July, 2015
ISSN: 2320 – 8791 (Impact Factor: 2.317)
www.ijreat.org

Remote Access To PC

Vivek Kumar[1], Vishal Kumar[2], Devaraj Verma C[3]

[1,2]6th Sem, Student, Department Of Information Science and Engineering, PES Institute of Technology,BSK-3rd Stage, Bangalore-560085, Karnataka, India
[3]Assistant Professor, Department Of Information Science and Engineering, PES Institute of Technology,BSK-3rd Stage, Bangalore-560085, Karnataka, India

Abstract: This paperinvestigates the remote access to PC. The problem is to design and implement an algorithm to create an android based interface through which we can easily access our PC/Laptop and adding some additional features like keyboard access, voice commands, live screen sharing etc. The constraints include tracing coordinates of mouse on the desktop,executable commands to start an application,accessing input from keyboard, setting up a TCP connection between client and the server.The ideas in the paper were implemented on Android as it provides beautiful and easy to handle interface.

I. INTRODUCTION

The problem is to design and implement an algorithm to create an android based interface through which we can easily access our PC/Laptop and adding some additional features like keyboard access, voice commands, live screen sharing etc. The constraints include tracing coordinates of mouse on the desktop,executable commands to start an application,accessing input from keyboard, setting up a TCP connection between client and the server.

We primarily focused on developing algorithm, which is easy to implement without compromising on its effectiveness and performance. To understand how these various methods work, first we need to look at the structure and working of android.

Android apps are written in the Java programming language. The Android SDK tools compile your code—along with any data and resource files—into an APK: an Android package, which is an archive file with an .apk suffix. One APK file contains all the contents of an Android app and is the file that Android-powered devices use to install the app.

Once installed on a device, each Android app lives in its own security sandbox:

- •The Android operating system is amulti-user Linux system in which eachapp is a different user.
- •By default, the system assigns each app a unique Linux user ID (the ID is usedonly by the system and is unknown tothe app). The system sets permissionsfor all the files in an app so that only theuser ID assigned to that app can accessthem.
- •Each process has its own virtual machine (VM), so an app's code runs inisolation from other apps.
- •By default, every app runs in its ownLinux process.Android starts the process when any of the app'scomponents need to be executed, then shuts down theprocess when it's no longer needed or when the systemmust recover memory for other apps.

An APK contains at a minimum, the directories and files shown in Figure 1. This AndroidManifest.xml file is most important in the research. This is stored in a binary XML format and must be converted to a plain text format before becoming human-readable. This file contains information such as the minimum Android version the app was designed for, the main activity (which is launched upon opening the app) and other details important to the basic functionality of an Android app. Most importantly for our purposes, it contains declarations of the Android permissions the app requires. Another file that will be used within this research is the classes. dex file, which contains the binary code of the app compiled to Dalvik byte code. Programmers are free to add as many directories and files as needed to fulfill their requirements. Due to the inclusion of the manifest file detailing every file contained within an app, the structure is quite flexible. Android apps are required to go through the application signing process before they can be installed onto a device. By default an Android system will not install an application if it is unsigned. This includes both physical and emulated Android systems. Generally for an organisation that releases Android apps, there is a single private key used to sign all their applications. By signing different applications with the same private key, they are able to share code and data as Android considers them to be within the same process.

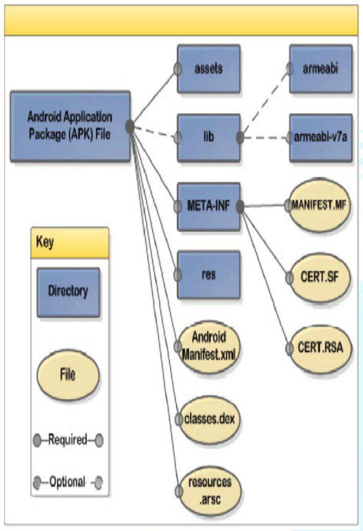

Figure 1:APK File Structure

II. MOTIVATION AND AIM

The broad objective that we set ourselves was to develop an android application, which would allow the user in the simplest possible way to access PC remotely. Such an application could have a number of possible uses:

- To open an application using voice commands.
- Tracing the mouse movement on the desktop.
- Keyboard access is given to the user.
- To obtain Live capturing of screen.

III. LITERATURE REVIEW

Literature reviews were done based on the previous implementations of some remote access android projects. The review covered the theory and concepts which will be used in generating this project. Literature review is showing the research related to the project topic.

All the weakness on the current or existing systems were identified in order for this project to overcome and the strength of the existing system are identified and studied so that it can be implemented in the system.

Previously we use remote access to control television, projector, etc., but this concept is used to remotely access laptop and its various features in our project.

Earlier the interface between the user and device was wired but this project makes us create an interface that is wireless and equally efficient and secure.

In our project we are using an android based platform to access our pc remotely. We are using client server architecture and setting up a TCP connection between client and server. The client and server must belong to the same network and use the same IP address for setting up the connection.

We are providing an android interface using which we can trace the mouse movement and all of mouse’s functionalities. We are adding some additional features like voice commands, keyboard interface and live screen sharing.

For mouse movement we are tracing the X and Y coordinates of the screen using the function mousemove( ) of robot class.

For voice command we are using Google speech recognition which does voice to text conversion which is getting passed to our server. Now we match the text content with our voice command directory in our server and if true the particular application will run else will ask for proper voice command.

In Live screen sharing we are reflecting the snapshots of our pc to our remote device which projects that image of the snapshots on our remote device.

IV. FUNCTIONAL REQUIREMENTS

- The user traces the mouse movement on the desktop.
- Keyboard access is given to the user.
- Voice command can be given as input.
- Live capturing of screen can be provided to the user.

V. DESIGN

- Executable commands to start an application.
- Tracing coordinates of mouse on the desktop.

- Accessing input from keyboard.
- In the earlier stages of the development of the app, the application used to crash if there was no network connection or the client and server are not in the same network.
- The client and server must belong to the same network and use the same IP address for setting up a TCP connection.

VI. CONNECTION SETUP

The **Transmission Control Protocol** (**TCP**) is a core protocolof the Internet Protocol Suite. TCP provides reliable, ordered, and error-checked delivery of a stream of octets between applications running on hosts communicating over an IP network. TCP is the protocol that major Internet applications such as the World Wide Web, email, remote administration and file transfer rely on.

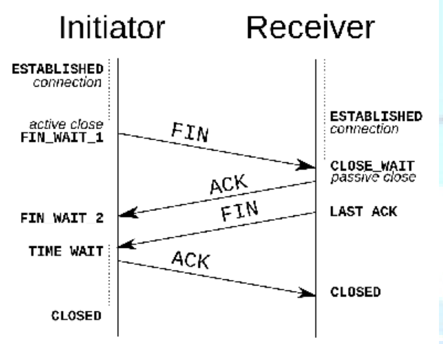

Figure 2: TCP Connection

VII. RESULTS

We can access PC remotely using this app. However certain assumptions are taken into account.

• Interface more advanced than mouse.

• User can use screen as touchpad.

•Providing left click and right click functionality similar to mouse.

• Voice, live screen and keyboard functions are implemented.

Figure 2 shows a TCP connection between the client and server.

Figure 3 shows the homepage of the app.

Figure 4 shows the advanced features of the app.

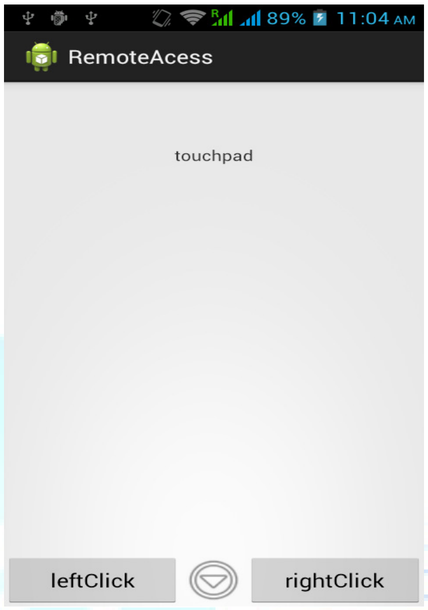

Figure 3:Homepage

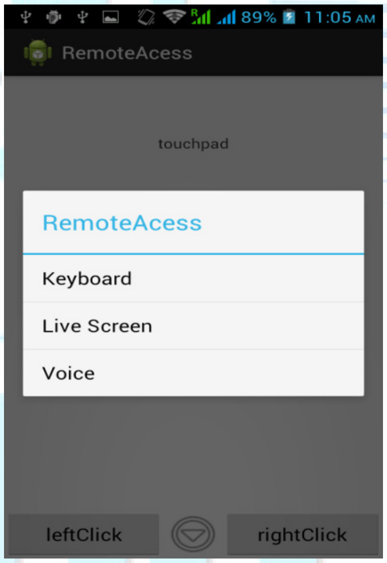

Figure 4:Advanced Features

VIII. IMPLEMENTATION

1. Mouse Touchpad: User can use screen as touchpad.Provides left click and right click functionality similar to mouse. We are tracing the X and Y coordinates of the screen using the function mousemove( ) of robot class.

2.Keyboard Interface:It is similar to wireless keyboard.We are defining each and every key in the server side

3. Voice Commands: For voice command we are using Google speech recognition which does voice to text conversion which is getting passed to our server. Now we match the text content with our voice command directory in our server and if true the particular application will run else will ask for proper voice command.(Commands to open notepad,chrome).

4. Live Screen Sharing: We are reflecting the snapshots of our PC to our remote device which projects that image of the snapshots on our remote device. We are using ScreenCapture() function.

CONCLUSION

We can access PC remotely using this app. However certain assumptions are taken into account.

1. Interface more advanced than mouse.
2. User can use screen as touchpad.
3. Providing left click and right click functionality similar to mouse.
4. Voice, live screen and keyboard functions are implemented.

FUTURE ENHANCEMENTS

1. Restart and Shut down functionality.

2. Starting more applications using voice commands such as VLC Media Player,iTunes,etc.

REFERENCES

[1] The New Boston Series Tutorials by Travis for android development(2012).

[2] https://www.7tutorials.com/connecting-remotely-windows-desktop-windows-7.html