

PocketDroid - A PC Remote Control

Chaitali Navasare, Deepa Nagdev⁺ and Jai Shree

Department of CSE, Bharati Vidyapeeth's College of Engineering for Women, Pune, Maharashtra, India

Abstract. In today's era, electronic devices and PC's are the inevitable parts ones' life. An important aspect of the technology is to remotely monitor these devices. We already have many Remote Control applications which provides an ease to control and monitor devices easily and quickly. This paper presents an application named PocketDroid, using which user can connect to any computer having Server Application running on it. It is basically an Android based Mobile Application for controlling a Target PC. User can have full access of the Target PC, provided its IP address is known. PocketDroid surrounds the Client and Server application. In which, the Server application has been implemented in JAVA and Client application in Android. As both JAVA and Android are open source platform thus they allows the development of new ideas and tests them with a set of open standards. One can use PocketDroid to share files between PC and android device, start and stop the applications installed on the Target PC, shutdown the Target PC and much more. This application also contributes for IT Administrators to remotely control any computer present in the network, allowing them to troubleshoot and solve problems faster.

Keywords: android, IP address, JAVA, Linux OS, pervasive computing, remote desktop, remote visualization, smartphone, wireless handheld devices.

1. Introduction

Initially mobile phones were developed only for voice communication but now-a-days the scenario has changed, voice communication is just one aspect of a mobile phone. There are other aspects which are major focus of interest. Two such major factors are web browser and GPS services. Both of these functionalities are already implemented but are only in the hands of manufacturers not in the hands of users because of proprietary issues, the system does not allow the user to access the mobile hardware directly. But now, after the release of android based open source mobile phone a user can access the hardware directly. He can design customized native applications to develop Web and GPS enabled services and can program the other hardware components like camera etc. This paper describes PocketDroid, an Android application designed to control the Remote Desktops.

Technological developments have enabled the creation of mobile devices with the technical features which were previously conceived only in PC architecture. With the advent of PocketDroid, here comes the need to integrate these devices so that interaction between the PC and mobile can be monitored and a better interaction can be accomplished. This paper proposes the implementation of mobile based PC control system using Android software stack [3, 4].

The advancements in 3G technology and wireless communication bring the convenient usage of mobile devices or internet. With the ease to access network, the remote systems are not limited to network protocols and its features.

The communication between the mobile devices and computer can be accomplished using internet.

⁺ Corresponding author. Tel.: + 020-27412614.
E-mail address: deepa.nagdev@gmail.com.

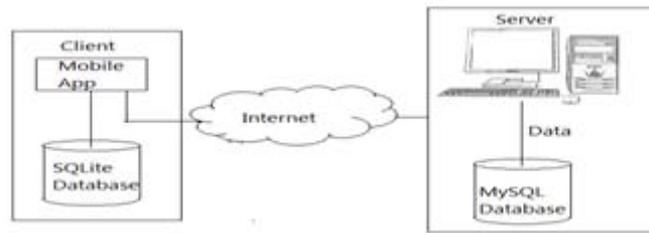


Fig. 1: Basic Architecture Diagram of PocketDroid.

Fig. 1 shows the basic architecture of the proposed system. The architecture consists of two terminals: (1) Android mobile phone and (2) Personal Computer. The details of the architecture are explained in further sections.

Section 2 gives an idea about Survey of existing work. Section 3 provides the details of the architecture. In Section 4 we will discuss about the design and Implementation. Section 5 goes through the security issues during communication. Finally, it is concluded in section 6.

2. Survey of Existing Work

The area of pervasive, or ubiquitous, computing was founded by Wieser (1991) who predicted that computers would one day be integrated into everyday objects and interacts with people seamlessly. There have been a number of research projects related to the use of the cell phone as a remote monitor and controller.

[1] Describes architecture for remote controlling based on Android. According to [1], there are different types of possibilities establishing connectivity between the Target PC and the Mobile Client such as USB Interface, Java Sockets, and Android Debug Bridge Client. Each of them has its own consequences. [2] Describes architecture for remote controlling based on Bluetooth. It is implemented in J2ME and the target PC is Windows based. It uses two technologies for creating connection – COM ports and JSR-82. Personal Computers do not support the JSR-82 API by default. [10] Uses the C#.Net technology for client and server. The connection is established using 802.11 link. The client is a PDA.

There are enormous projects and initiatives designed that allow remote control between devices. Even there are some initiatives that aim to control mobile devices. But most of them lack in use of open source platforms. So, we present an initiative in open source that covers this particular area of interest. The proposed platform is flexible and scalable.

This paper focuses on controlling through Android Platforms. This is an open platform that allows using other technologies (also open). In addition, Android platform allow the development of new ideas easily and test them with a set of open standards [11]. The prototype generated as implementation of the proposed architecture will be provided also as free software. According to data released by Nielsen [12], half of the consumers who recently purchased a Smartphone chose an Android Smartphone.

3. Architecture

Figure 2 shows the internal software structure of the client side and server side of PocketDroid System. Each side is divided into components with specific well defined functionality. Client is an android device which controls the remote PC. Commands issued by the user are saved using the SQLite Database, for the future references.

Requirement for client

The minimum android version on which the application is build is 2.1. The android device should have SDCard installed of minimum 1GB memory. Internet availability is must.

Server is the Linux/Windows based remote PC. Issued Command is executed here and the result is sent back to the client through sockets. Similar to client here MySQL is used for log creation.

Requirement for server

Minimum 256MB RAM is required. Internet availability is needed on the server side also. The components constituting the client side are: Client GUI, Saver/Loader, SQLite database, and Connector.

The Client *Graphical User Interface (GUI)* component is responsible for interacting with the server. It provides the necessary information such as user-name, password and IP address which can be used for further communication. This function is vital for having the command level control and displaying the files/folders of targeted PC.

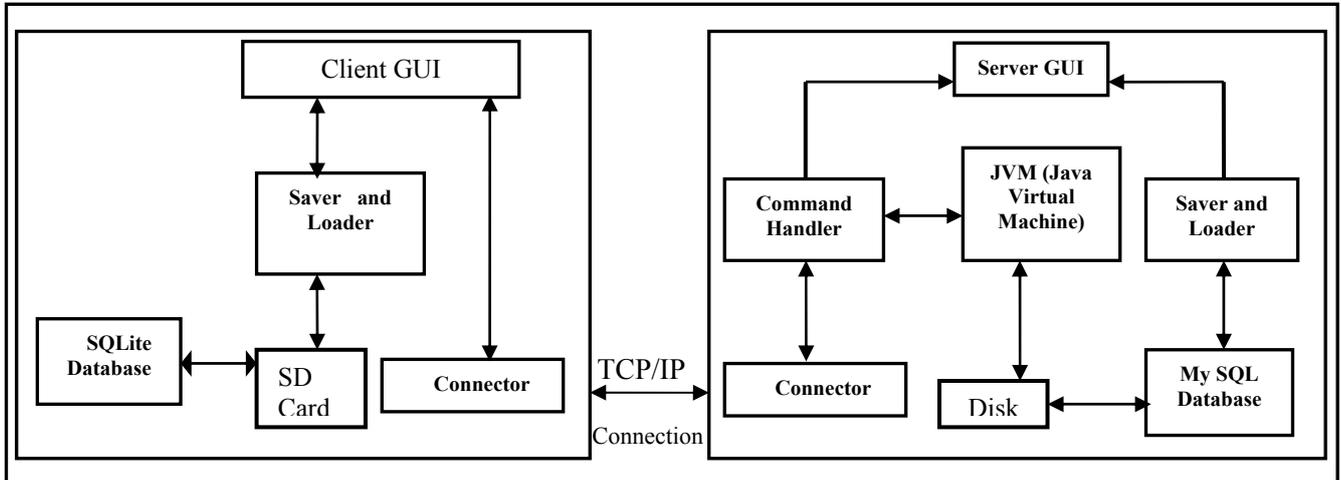


Fig. 2: Basic Internal Structure of PocketDroid [10].

The *Connector* is the layer that provides communication with the server side. It sits on top of the TCP/IP transport layer. It is responsible for initiating the connection and interacting with the server. The commands that are received from the Client are sent through the TCP layer to the Command Handler of server side.[10] *Saver and Loader* is used to store the data from GUI in the SD Card of android phone. *SQLite Database* is the database on client side which is used to store the users and their respective passwords. It is also used to store the log on client side. Log on client side contain the commands issued by the user, date and time of the command issued, for future reference.

The components constituting the server side are: Server GUI, Command Handler, JVM, Saver/Loader, Connector and MySQL database.

The *Server GUI* is the User Interface on the target computer which is used for interaction. The responsibility of the GUI is to provide functionality such as viewing the log, disconnecting the server, changing user settings, etc. *Command Handler* is used to issue command on Client side and is sent to the server for execution. This command is executed on the server side and the result of the commands is sent to the client for displaying the same. In this application at a time only one command is executed. After execution of one command then only next command can be executed. *JVM (Java Virtual Machine)* is a virtual machine capable of executing Java byte code. It is the code execution component of the Java software platform. *Saver and Loader* is used to save and retrieve the data stored in the disk or database. *MySQL database* is used to store the information of user while installing the server side application. It is also used to store the log on server side. *Connector* is used to connect to the client and transfer data between client and server during communication.

4. Design and Implementation

The system basically involves communication between an android phone and a remote desktop. The remote desktop and android phone can communicate with each other using the IP address and ports. Data can be transferred using ports i.e. socket of android phone and socket of computer. This transferred data can take the form of commands or files. The Server is implemented in Java and Client in android.

4.1. The designing of PocketDroid is divided into three phases:

- 1) Running of basic Linux or Windows Commands via android application: The basic operations such as “dir”, “ipconfig”, run and stop any application; etc can be issued from the android terminal.
- 2) Provide a mechanism for graphically interacting with files and folders of target PC: Contents of the provided path will be displayed graphically. Basic file operations can be performed such as “cut”, “copy”, ”paste”, etc.
- 3) Remote Visualization: This mechanism allows the client to actually view the screen of target PC on Client`s screen and actions performed in an efficient way. This will be our future work.

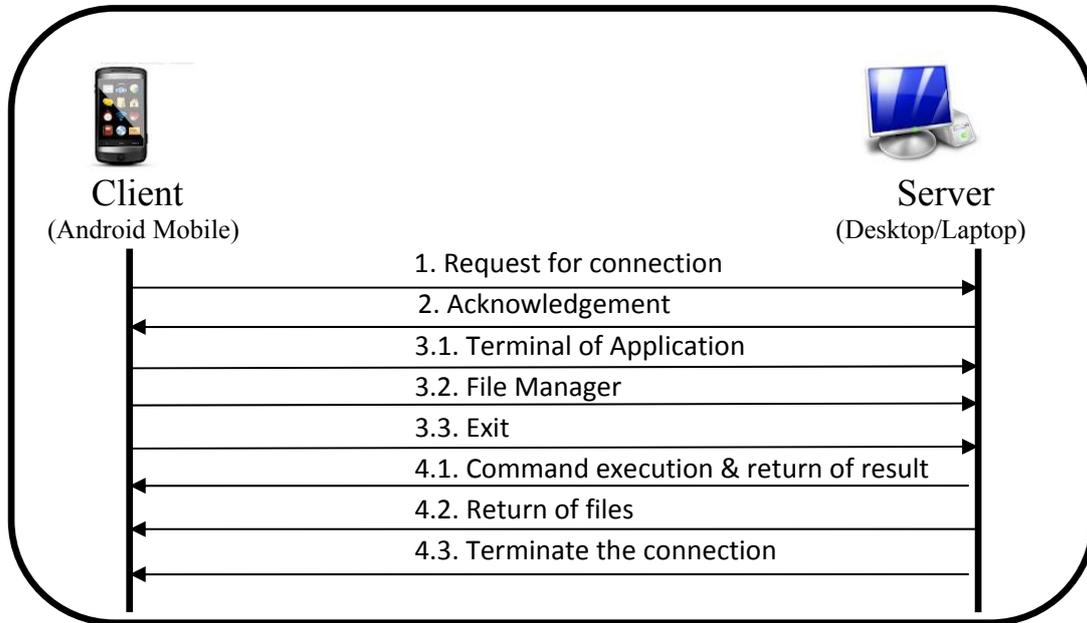


Fig. 3: Overview of PocketDroid

4.2. Options

1.) Request for connection: The client (android application) requests for connection to the server by providing the IP address.

2.) Acknowledgement: The server accepts this request and connection is established. Then the Menu is displayed on the client side having following options:

- Terminal
- File Manager
- Exit

4.3. The user on the client side selects one of the options

4.3.1.) Terminal: If the user opts for terminal, then the user is switched to the “Terminal” screen. Through this the command level execution takes place. The user can issue any appropriate command such as DIR, START NOTEPAD.EXE etc and this command is sent to the server via TCP/IP stream. Go to step (4.4.1).

4.3.2.) File Manager: If the user opts for File Manager, then the user is switched to the “File Manager” screen which displays the root drives. The operations required for file such as CUT, COPY, PASTE etc. can be performed. The user can see the contents of the selected drive by just tapping on it. On tapping on one of the drive, path of that drive is sent to the server. Go to step (4.4.2).

4.3.3.) Exit: The request for terminating the connection is sent to the server .Go to step (4.4.3).

4.4. The execution on the server side:

4.4.1.) The server executes the command by using following JAVA code:

```
Runtime rt = Runtime.getRuntime();Process pr = rt.exec("cmd /c "+str);
```

The result is sent to the client via TCP/IP stream and the result is displayed on the Terminal of the Client application.

4.4.2.) The server sends the files of the provided path in JSON format. The format is as follows:

```

{"Files":{"no_of_files":"3","Page":"1"}
 {"Filedetail": [
  {"name":"Important.txt","Path":"C:\\","size":"100kb"}
  {"name":"Eclipse.exe","Path":"D:\\","size":"100kb"}
  {"name":"Song1.mp3","Path":"E:\\","size":"100kb"}
 ] }}

```

Fig. 4: JSON format for File Transfer

On the client side the Json File is parsed and the file names are displayed in the list form. The further processing of files is done as above.

4.4.3.) The server responds to the client by disconnecting the connection.

5. Security Management

Due to the increase in use of Smartphone and the number of attacks on such devices, it has become the need of implementing the security mechanisms. Android Platform was designed to enable applications to run in specialized sand-boxes, preventing the spread of the malware. The security between the applications and the system is applied at the process level by standard Linux facilities [7, 8].

In Android Platform, to use INTERNET in an application special permission are required to be taken. Due to these permissions execution of malicious code can be avoided.

To prevent from unauthorized users, authentication of user using username and password is done.

In order to make PocketDroid secure and prevent it from malicious attacks, user authentication is used in this application.

6. Future Work:

As a continuation of work in this application, we would include the encryption algorithm to prevent data leakage. We will also put efforts for displaying the screen of the target PC on the android phone itself for the purpose of better visualisation.

7. Conclusion

This paper proposes an architecture which will control the targeted PC using android mobile phones. During implementation some general requirements for the application were identified. Furthermore, this application does not require any additional equipments or software. Thus, the application will work fine irrespective of whether the server is used in Linux or Windows platform. Security will be maintained on the client side as android needs permission for internet access. We hope that this application will be beneficial for all those who wish to enhance this application by adding their views.

8. References

- [1] Angel Gonzalez Villan, and Josep Jorba Estev, "Remote Control of Mobile Devices in Android Platform" IEEE transactions on Mobile Computing, 2011.
- [2] Qadeer, M.A.; Agrawal, R.; Singhal, A.; and Umar, S, "Bluetooth Enabled Mobile Phone Remote Control for PC", International conferences on Advanced Computer Control, 2008.IEEE DOI 10.1109/ ICACC .2009.91
- [3] Android. <http://www.android.com> Retrieved 3rd Sep, 2011.
- [4] What is Android? <http://developer.android.com/guide/basics/what-is-android.html> Retrieved 3rd Sep, 2011.
- [5] Gmote. <http://www.gmote.org/> Retrieved 21st Aug, 2011.
- [6] RemoteDroid. <http://remotedroid.net/> Retrieved 23rd Aug, 2011.
- [7] Portokalidis, G., Homburg, P., Anagnostakis, K., Bos , H.,Paranoid Android: Versatile Protection For Smartphones ", In Proceedings of the 26th Annual Computer Security Applications Conference (December 2010), pp. 347-356.
- [8] William Enck, "Understanding Android Security", IEEE Security & Privacy, 2009,

- [9] Application Fundamentals- Android Developers. [http:// developer.android.com/guide/topics/fundamentals.html](http://developer.android.com/guide/topics/fundamentals.html)
- [10] Yenel Yildirim and Ibrahim Korpeoglu," PocketDrive: A System for Mobile Control of Desktop PC and its Applications Using PDAs"
- [11] Dean Jezard, Johnny Makkar, David Holding-Parsons, Google Android Whitepaper, TigerSpike, 2008,
- [12] Nielsen, "U.S. Smartphone Market: Who's the Most Wanted?",<http://blog.nielsen.com/nielsenwire/?p=27418>, April 26th,2011