

Android Desktop Control (ADC)

Santosh Dahifale, Ritukumari Pandey, Rahul Ballani, Sagar Ingle

Abstract: ADC is an application which enables the user to view and interact with Remote PC by using an Android phone via Internet. With this application the user can not only interact with the remote system of various operating systems(Linux, Unix, Windows ,Macintosh, etc.) but can also perform several operations like mouse click operations, sending a text message, opening documents ,manipulating files, FTP. It uses a simple Bilinear Interpolation Algorithm for scaling remote system's GUI with the android phone. It uses a SHA-1 algorithm for OTP.

Index Terms: OTP- One Time Password, SHA- Secure Hash Algorithm, FTP- File Transfer Protocol, Scaling, Android

I.INTRODUCTION

Android Desktop Control (ADC) is a client-server based system that is platform independent. ADC enables us to control the remote system and its peripheral devices from Android Phone. The client software runs on android device while the server part operates at the Remote PC. In this way ADC achieves remote display to the user's android device. It also allows us to fetch images from the server side and scale them on the client side. Android operating system is built on a Linux Kernel. The software stack contains java applications running on a virtual machine, and system components are written in java, C, C++, and XML. It achieves mobile computing without carrying any hardware. It allows us to be connected in the network and provides secure and easy access to information within the network. ADC requires two types of application. 1. Server application for a remote system under control.2. Client application for controlling the Server application. Client is responsible for viewing the screen and capturing user activities. On the Server side, all the application logic is executed on server. It also updates the graphic screen by sending a desktop view to the client.

II. LITERATURE SURVEY

1. Existing System:-

There are many existing systems which work as a connection mediator between Android phones and the desktop. But they have some limitations .Some of them are listed as below:

1.1 One of the applications is used for controlling desktop from cellular phone, but it uses Remote Frame Buffer (RFB) protocol, which is a slow protocol .Due to this, the working of that application is slow.

1.2. Another application is based on Central Server, in which the server and client (Android phone) are connected to the central server. Here the application logic is executed on the central server. So the speed of operation is low and the whole system solely depends on the central server.

2. Proposed System:-

The proposed system is a desktop sharing system which allows remote access of the system via Android phones. When the server and client starts, then Client (Android phone) request for OTP by entering IP address of the server machine. Then Server sends OTP via SMS to client. Entering that OTP, connection between client and server will be established. Through this connection, client can request for various services like Viewing Screen Shots, Keyboard Control, Mouse Control, Hardware Control, Messaging etc. Server receives all requests from client and processes the request by generating response. (Refer to fig 1.1)

3. Algorithms Used:-

3.1. Bilinear Interpolation:

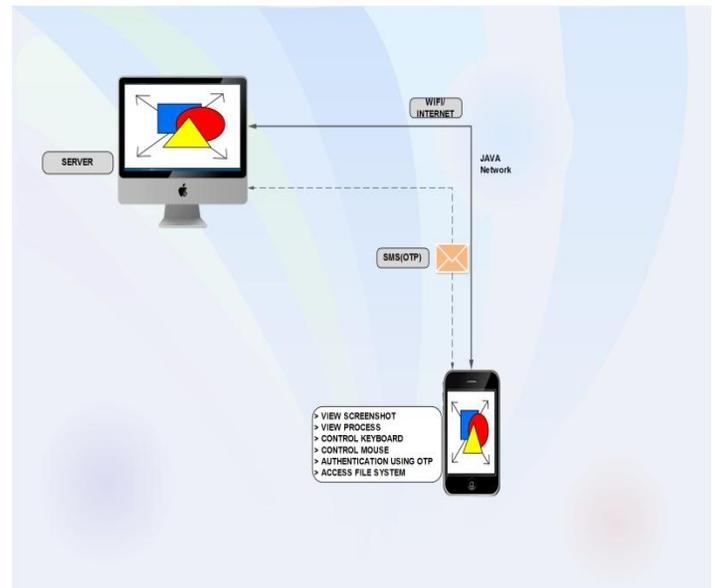
It is a scaling algorithm which uses only the four nearest pixel values which are located at diagonal directions from a given pixel in order to find the appropriate color intensity values of that pixel.

3.2. SHA1 (Secure Hash 1) Algorithm:

SHA1 is hashing algorithm which is used for converting the message to unique representation of message that is multiple of 512 bits in length, without loss of information about its exact original length in bits.1

(1)

Fig. 1.1 Client Server Interactivity.



4. Proposed Methodology

4.1. We have set the authentication for this application by entering the system's IP address on the android phone. If the IP address is valid, then the user request for One Time Password(OTP) by clicking on the Request OTP button. As soon as the OTP is generated the user has to enter the OTP in the Enter Password field. On clicking the OK button the android device is connected to the server PC. Refer the fig 1.2(Login Page).



Fig. 1.2 Login Page.

4.2 With this application the user is being able to perform various mouse clicking operations (enabling and disabling mouse), keyboard operations (typing on the text editor), Image fetching, creating various keyboard shortcuts. Refer Fig 1.3 (Options Pane).

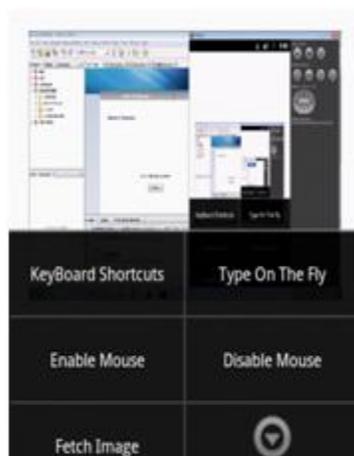


Fig.1.3 OptionsPage

4.3 The figure below gives us a detailed view of the keyboard shortcuts page. With this option we can create various keyboard shortcuts like Ctrl+C (for copy), Ctrl+V (for paste), Ctrl+A (for Select all) and many more.

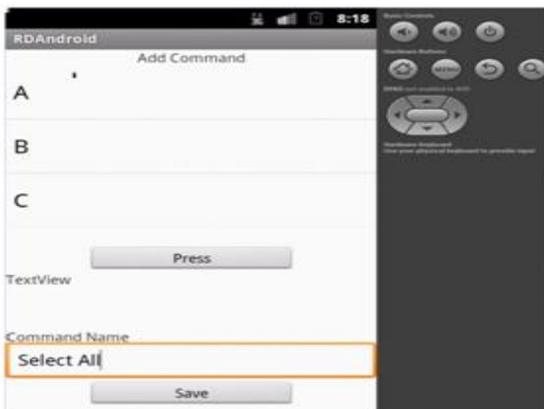


Fig.1.4 Keyboard Shortcut

III.FUTURE SCOPE

Currently we are working on Power Control of other devices (e.g. Fan, Tube Lights, Washing Machine, AC etc) through remotely controlled PC. In Future, we are planning to do power control of Remote PC in which we can ON/OFF our server PC through android phone remotely as well as memory & file sharing management at client side. And also we are planning to make client side platform independent (i.e. iOS, Windows, etc.)

IV.CONCLUSION

ADC is an application which allows us to view a computing desktop environment not only on the machine where it is running, but from anywhere on the internet. It allows us to control the peripheral devices of remote PC. It allows interacting and viewing with remote PC. There is no need of carrying hardware each time for accessing our home or friends PC. It allows us to be connected in the network which will provide secure, easy access to information within the network.

V.REFERENCES

- [1]. Md. Sanaullah Baig, Rajasekar M. and Balaji P Virtual Network Computing Based Remote Desktop Access, International Journal of Computer Science and Telecommunications, Vol. 3, No. 5, 2012.
- [2]. Margaret Butler, "Android: Changing the Mobile Landscape, IEEE Pervasive Computing", pp. 47, 2011.
- [3]. Ha-Young Ko, Jae-Hyeok Lee, Jong-Ok Kim, "Implementation and Evaluation of Fast Mobile ANDRODESK Systems", IEEE Transactions on Consumer Electronics, Vol. 58, No.4, 2012.
- [4]. Daniel Thommes, "RemoteUI: A High-Performance Remote User Interface System for Mobile Consumer Electronic Devices", IEEE Transactions on Consumer Electronics, Vol. 58, No. 3, 2012.
- [5]. Tristan Richardson, Quentin Stafford-Fraser, Kenneth R. Wood and Andy Hopper, "IEEE Virtual Network Computing Internet Computing", Vol.2, 1998
- [6]. W. Yu, J. Li, C. Hu, and L. Zhong, "Muse: a multimedia streaming enabled remote interactivity system for mobile devices," Proc. 10th International Conference on Mobile and Ubiquitous Multimedia, pp.216–225, 2011.
- [7]. ITU-T Rec. T.128, "Multipoint application sharing," Jun. 2008.
- [8]. R. A. Baratto, L. N. Kim, and J. Nieh, "THINC: a virtual display architecture for thin-client computing," Proc. 20th ACM Symposium on Operating Systems Principles, pp. 277–290, 2005.
- [9]. O. Boyaci and H. Schulzrinne, "BASS Application Sharing System," Proc. 10th IEEE International Symposium on Multimedia, pp. 277–290, 15-17 Dec. 2008.

- [10]. R. W. Scheifler and J. Gettys, "The X window system," ACM Transactions on Graphics, vol. 5, no. 2, pp. 79–109, Apr. 1986.
- [11]. H. Shen, "A high-performance remote computing platform," Proc. Of IEEE International Conference on Pervasive Computing andCommunication (PerCom 2009), pp. 1-6, Mar. 2009
- [12]. Jae-Hyeok Lee, Ha-Young Ko, Jong-Ok Kim, "Fast modified region detection for mobile VNC systems," Proc. of International Conference on Awareness Science and Technology (iCAST 2012), Aug. 2012