© 2016 IJSRSET | Volume 2 | Issue 3 | Print ISSN : 2395-1990 | Online ISSN : 2394-4099
Themed Section: Engineering and Technology

Remote Desktop Controller Using Android

Muneer P, Mohammed Shareef C, Vipin K R, Prof. Joby George

Department of Computer Science, M. G. University, Kerala, India

ABSTRACT

Remote Desktop controller is the software used to control computers remotely using other computing devices in network. The software allows the users to communicate with the computer remotely using an Android phones. The user can see the computer display output right in his Android phone and can also give input to the computer using the Android phone just as he uses the keyboard and mouse of the computer. User can also get the audio output of the computer in the Android phone. In short, Android phone used as a Remote Desktop Controller can replace two input devices and two output devices of the computer; keyboard, mouse, monitor and speakers. Our remote desktop controller project enables complete control over a desktop (or laptop) computer using an android phone. The advantage of touch interface of android phone is utilized to produce a remote controller app with efficient UX. Both keyboard and mouse inputs are produced right from the android device. The device displays the computer screen with frame rate between 10 and 15 fps. Devices with RAM above 1GB can produce screen at minimum rate of 12 fps.

Keywords: Android Device, Desktop Computer, Remote Desktop Controller, Mouse, Keyboard

I. INTRODUCTION

Nowadays computing is going extremely mobile due to introduction of Android phones and tablets. Unlike mobile computing devices, desktop computers are known for their computing capacity. It will be useful if we can club these advantages of the two entirely different classes of computing devices. Since desktop computers are bulky and lacking mobility, remote desktop computing can make significant difference. Unlike desktop computers, mobile computing devices, especially Android devices, have the advantage of a very useful and handy input interface. Touch input makes it much easier to control the device. Using the remote desktop controller, the common input methods, mouse and keyboard, to desktop computers are implemented via touch interface. The project concentrates on simplifying control of desktop computer via android devices. The Android mobile platform is the most popular mobile platform and can be used as a remote for any desktop PC.

The desktop PC can be controlled with any android device connected to the same network as the former is connected. More than portability or mobility, remote control is also a significant advantage. Desktop computers seriously restrict the users' posture. He has to sit down in front of the computer. But our remote desktop controller app brings the desktop at the user's finger tips no matter where or how he sits. Our project was to develop a remote desktop controller application in android for computers. Any trusted friend of the user can assist him in doing something in computer. Remote assistance has made simple via mobile devices and the one who intends to assist does not need any knowledge what remote assistance is or how to set it up. The app also lets the user share his computing resource with anyone.

Also, it is made sure that it won't compromise the computer's security in anyway. The most common uses are accessing one's office computer at home, enabling to manage files, use available resources, and generally everything the user would be able to do if he were actually at the computer in question. Thousands of businesses over the world use and rely on RDS as a core

part of their IT framework. From multinational corporations to educational institutes offering long-distance-learning and so many more besides, there are so many uses to RDS in innumerable industry sectors.

The core purpose of RDS is to allow workers to perform their duties from literally anywhere at any time. The only thing needed is a computer and a network connection. In another perspective, the ease of use and flexibility of smart phone is combined with excellent computing capacity by using the android application. Thus, in short, remote desktop technology allows a person to use a handheld device and control a different computer somewhere else.

II. METHODS AND MATERIAL

1. Related Work

The Android device should be able to detect computer systems that are available in the network for remote access so that the android user gets a list of listening computers in the application program. The user should be prompted for login credentials if he attempts to remotely control a computer. The computer should check whether the login credentials are authentic and the user who is trying to login is a legitimate one. Furthermore, when user is allowed to take control over the remote PC, he should be given options to choose from, whether to use remote keyboard, remote mouse or access full control over the computer.

Since convenience is more important, the main focus lies on assuring full control for the user over the remote control. As part of this, we have to conveniently implement mouse movements, mouse press, mouse release, mouse click, mouse double click, etc. for all the mouse buttons. Also, the computer screen needs to be visible to the android user. Since computer screen is often larger than most devices, the screen or its relevant part has to be conveniently relayed to the android user. The screen resolutions also differ. Similarly, to implement keyboard input, the mobile has to show a keyboard in screen and use touch interface for keyboard input. The computer keyboard is largely different from the Android keyboard except in the case of letters.

The user should get access to mouse while he can watch the screen. Keyboard should also be available at the same time.There should not be considerable delay between the initiation of an input from the touch interface of the Android device and its implementation on the PC side. Since mouse input, keyboard input and screen streaming have to work in parallel, neither of these should affect others in anyway. The user at the android device should always feel that the system is responsive enough and the proposed system should not consume much of CPU time and memory so that the user should be able to even play video games in it.

The application on the computer side allows the user to set username and password for the computer for remote access. There is a module in the PC side which broadcasts packets to enable remote PC discovery in network. The Android device on the other side should prompt the user for username and password for login. The same module in PC is responsible for credential validation and acceptance or rejection. Based on this, the client displays an options list to choose from. On choice by the user, the Android device informs the computer about his choice.

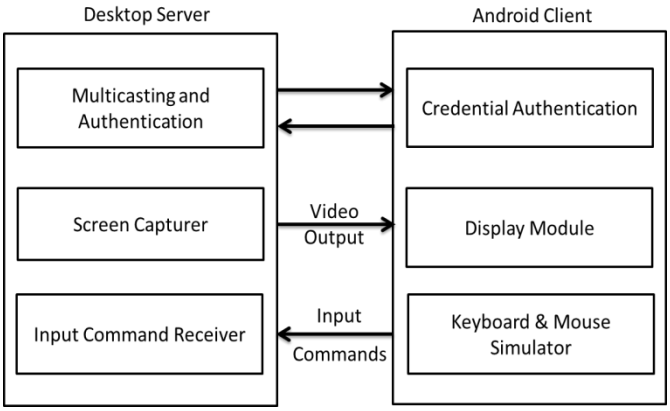

Figure 1. System Architecture

The desktop computer opens new ports for required modules. The other modules include screen streaming module or screen capturer module, input command receiver module. The screen capturer module communicates with the display module. Although initially it is two-way communication, later it needs only one way communication. The input command receiver module communicates with keyboard and mouse simulator module in the Android device which makes use of the touch interface to receive inputs from user and convert it to appropriate input command. Virtually, it is a single module. But based on the choice it may be two

different modules working in parallel, the keyboard input module and the mouse input module.

Although these are modules in general, there are sub modules wherever required. And also there are combined and separate versions of input reception module since there are different scenarios where either one of the inputs or both are required.

2. Detailed Design

As the proposed system should support, mouse input, keyboard screen streaming, and all the three functions together for full control. For singular use, there should be separate and independent modules for mouse input and keyboard input reception. But when it comes into full control, it is important that the input command reception threads shall not degrade the performance of screen streaming threads. Thus, for the full control part, both the mouse input and keyboard input modules should be combined to ensure minimal CPU usage. In case of screen streaming, the speed mainly depends on the amount of traffic, processing done on the PC side after capturing the screen image and the processing done on Android side before displaying it on the screen.

A. Input Design

A process of converting user originated inputs to a computer-based format. Input design is an important part of development process since inaccurate input data are the most common cause of errors in data processing. Erroneous entries can be controlled by input design. It consists of developing specifications and procedures for entering data into a system and must be in simple format. The goal of input data design is to make data entry as easy, logical and free from errors as possible. In input data design, we design the source document that capture the data and then select the media used to enter them into the computer.

Input design is a part of overall system design which requires careful attention. Often the collection of input data is the most expensive part of the system, in terms of the equipment used; it is the point of most contact for the users with the computer system; and it prone to error. If data going into the system is incorrect, then the processing and output will magnify these errors.In our proposed system, the focus is how the keyboard and mouse inputs for the remote computer are obtained from the Android device.

The only concerned way of taking input s using an android device is using the touch interface. Since touch input is very flexible, it is enough to act both as keyboard and mouse. For availing full access to the remote system, we have to define different meanings for different Gestures. Scroll gesture on the screen display is used to determine scrolling the screen in the mobile. Similarly, long tap for right click, single tap for single click and double tap for double click. But when keyboard is considered, a layout is required. When the keyboard is displayed, the clicks on the buttons are conveyed to the remote PC via the network.

B. Output Design

Designing computer output should proceed in an organized, well throughout manner; the right output element is designed so that people will find the system whether or executed. When we design an output we must identify the specific output that is needed to meet the system.

The main output of the system is the generation of inputs at the computer sides based on the gestures received at the Android device. Also the display of the computer's response in screen gives satisfactory output. Screen streaming at a reasonable frame rate is required to convince the user that the system is responsive. Unlike generation of keystroke and mouse click generation, screen streaming is seriously challenging. We have to keep the system stable so that the continuously fetching and writing of the screen image should exhaust the working memory. Also, there should be a minimum frame rate of 12 fps.

C. Screen Streaming

For the purpose of capturing the image, we relied upon the Robot class which was developed mainly for automation of software testing. Robot class, which is part of robot package, did the work perfectly. It gives a BufferedImage object which is in PNG format supporting both Alpha and colour components. Although removing alpha component will save some space but it may compromise speed. Since most phones' screens cannot display the whole screen with screen

elements distinctly visible, we decided to send and display only the relevant rectangular sub image of the screenshot. Thus we saved a lot in traffic and got a better speed. This rectangular sub image was cut out from the BufferedImage object, centred at mouse pointer. To save more in traffic, we also decided to send image only if the new capture is different from the earlier one. Sending a diff patch consumed much more time in the diff extraction process. Also, if there is only a slight difference (e.g. a notification at taskbar) it would take much time to detect this change and then still it has to transport the diff image. So, we left that attempt. Instead, for every capture, we checked whether there is any difference and if yes, the new image is sent. Also, when there is some mouse pointer movement, no doubt, we have to send the image.

Then comes the image object compatibility problem. We first preferred to write image using the ImageWriter in Java. But since the image API in dalvik is different from that of java, there isn't any compatible image reader in android. Thus the only way is to write it as an array of bytes. Since object serialization is much slower in dalvik when compared to java, we decided to encode the byte array to string using Base64 (radix-64) encryption. This also solved the question of how to delimit an array from the array of next image in sequence. Time complexity analysis also proved this method to be fastest.

D. Keyboard Input

Since the computer keyboard is entirely different from Android keyboard except in the letters' layout, there was a necessity to design a new keyboard layout including modifier keys like ctrl, alt and shift, function keys (F1,F2, F3..),and additional keys like Caps lock, Tab, Num lock, Enter etc. After the design of a keyboard layout, the virtual key numbering in Core java was adopted into Android side also. Assigning static integer variables for each key, communication with the PC from android was made easier. Following the same numbering as in the KeyEvent class in java simplified the approach. A thread at the PC side needed to be running as long as keyboard inputs were needed.

E. Mouse Input

Unlike keyboard input, receiving mouse input is more complicated since the received screenshots are centred at mouse pointer. Therefore, scrolling the screen in the android device actually means moving the mouse pointer, which will automatically determine the relevant sub image of the screen. Furthermore, determining different mouse gestures or inputs like left click, left double click, left press, left release, right click, right press and right release makes translating gestures on the touch screen more difficult. Any single touch will be the most commonly expected touch alternative for single left clicks. Also, a long tap is commonly expected to be the right click. So, these two possibilities are reserved and should be preserved as such for keeping user expectations.

III. RESULTS AND DISCUSSION

IMPLEMENTATION

The implementation stage involves careful planning, investigation of the existing system and its constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

The application at computer side is implemented using java programming language. The main advantages of using java were that it has the robot class and it has data types and classes compatible with the android dalvik, which is actually a clone of java. Initially, the computer side application should allow the computer administrator to setup the login credentials for remote login. Along with that, the user can allow or disallow auto acceptance of login request. If disallowed, the computer admin has to grant permission manually for remote control. Otherwise, access will be granted as soon as the user enters valid credentials.

After receiving corrected login account details, the user is allowed time to choose one of the control modes; remote mouse mode, remote keyboard mode and the full control mode. Based on the selected mode of access, the application should launch corresponding threads. It may be keyboard input receiver thread, mouse input receiver thread, and both combined as single thread or screen capturer thread.

For the purpose of capturing the image, we relied upon the Robot class which was developed mainly for automation of software testing. Robot class, which is part of robot package, did the work perfectly. It gives a

BufferedImage object which is in PNG format supporting both Alpha and colour components. Although removing alpha component will save some space but it may compromise speed. Since most phones' screens cannot display the whole screen with screen elements distinctly visible, we decided to send and display only the relevant rectangular sub image of the screenshot. Thus we saved a lot in traffic and got a better speed. This rectangular sub image was cut out from the BufferedImage object, centred at mouse pointer. To save more in traffic, we also decided to send image only if the new capture is different from the earlier one. Sending a diff patch consumed much more time in the diff extraction process. Also, if there is only a slight difference (e.g. a notification at taskbar) it would take much time to detect this change and then still it has to transport the diff image. So, we left that attempt.

For every capture, we checked whether there is any difference and if yes, the new image is sent. Also, when there is some mouse pointer movement, no doubt, we have to send the image. Then comes the image object compatibility problem. We first preferred to write image using the ImageWriter in Java. But since the image API in dalvik is different from that of java, there isn't any compatible image reader in android. Thus the only way is to write it as an array of bytes. Since object serialization is much slower in dalvik when compared to java, we decided to encode the byte array to string using Base64 (radix-64) encryption. This also solved the question of how to delimit an array from the array of next image in sequence. Time complexity analysis also proved this method to be fastest.

The remote desktop controller faces two types of users, the system administrator, and the android user. The system admin can set the username and password for remote login. If he allows auto-accept, when the PC is listening for remote access requests and the android user enters the correct password, he will be instantly granted permission for remote access. If he disallows it, the system admin has to review the remote access request and grant access manually. Then the user will be shown with options to choose (different control modes) from if he has entered correct password. Then based on the chosen option, the corresponding type of control will be granted to the user so that he could use the system remotely.

The android user can easily scroll the computer screen in his android screen. Also, wherever he gives a single touch, it is transformed to a touch at the corresponding position in the computer screen. Similarly long tap is transformed to right click. If the user swipes up from the bottom of the android phone, a specially designed keyboard will appear and can be used to type as in the computers keyboard.

IV. CONCLUSION

The growing importance of mobile devices enhances the need for mobilising desktop computers. The processing capability of desktop computers which is its main advantage over mobile computing devices is utilised well with remote desktop control. A person can use his office or home computer through his android device equipped with this remote desktop controller application. Similarly one can grant access to his computer for his friends without sharing his user account password. A desktop computer restricts its user to a certain posture, but the remote desktop controller dismisses this restriction. Moreover, the application provides means to watch or monitor any malign usage of one's personal computer by someone else.

V. REFERENCES

[1] Remote Control of Mobile Devices in Android Platform Angel, Gonzalez Villan, Student Member, IEEE and Joseph Jorba Esteve, MemberIEEE.

[2] 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.

[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] H. Shen, A high-performance remote computing platform, Proc. Of IEEE International Conference on Pervasive Computing and Communication (PerCom 2009), pp. 1-6, Mar. 2009.