A new Remote Desktop Approach with Mobile Devices: Design and Implementation

Authors Teng-Yao Huang[1], Hai-Hui Wang[2.], Chun-Lung Peng[3.], Hsin-Mao Huang[4.]
[1 2 3 4.] Department of Computer Science and Information Engineering, Taipei Campus, Aletheia University, Tamsui, Taipei, Taiwan
am001871@au.edu.tw [1.]
am001909@ au.edu.tw [2.]
am001959@ au.edu.tw [3.]
xmhuang@ au.edu.tw [4.]

Abstract: As the mobile device industry grows rapidly, smart phones and tablet computers have become the indispensable devices in our daily lives. Even more, one could have owned many devices at the same time. Although, the traditional personal computers or laptops has more powerful computing performance than mobile devices, but, compared with convenience and portability, the mobile device would be a better choice. Eventually, this leads to the increase of the variety of devices in our living; however, each of these devices has its own functions, which is not always replaceable from other devices, and how will these devices interact has become one of the most important discussion in the field of computer science.

In this paper, we will focus on the integration of personal computer, which has better computing capacity, and mobile device, so it allows the user to control the mobile device through their personal computer via Wi-Fi or 3G. This allows the user to work with their personal computer and mobile device at the same time.

The integration of both devices will improve the usage of the resources provided by these devices, such as phone call communications, GPS and G-sensor, so that we will be able to use our PC sufficiently, and contribute to computer technology in the future.

Keywords: Mobile Device; Remote Control; Remote Desktop.

1. Introduction

1.1 Research Backgrounds

Since the increase in the usage of mobile devices, the functions of these devices are no longer confined in phone calls and SMS. The developers enhance the device by adding more tools that will satisfy customers' needs. These hardware such as GPS, Bluetooth, E-compass and Ambient light sensors, are exactly what lakes in our personal computers.

However, the computing capacity of mobile devices is limited, and we still focus our work on personal computer at most of the time. This paper will integrate the features of both devices, and hope to solve problems corresponds to convenience. In scenarios like, typing an SMS with a small keyboard on the touchscreen or taking a call while your phone is not around.

Our system allows you to remote control your mobile device from your personal computer through wireless network. Therefore, are able to make phone calls, SMS, and all other applications in the mobile device from personal computer.

1.2 Research Problem

Due to that the main operating system for most mobile devices are based on Android. So, we will focus on Android version 4.0.3, which is the general version, as a research platform.

The main functions in our research are divided into audio, visual interface and input event. Wherein, the audio streaming and screenshot are restricted in the Android system.

Therefore, our research encounters two major problems:

(1) The audio cannot be recorded without consent of the caller based on legal norms from certain countries or regions. So, in order to avoid the illegal behavior, Google does not allow applications to access audio streams.

(2) Additionally, obtaining the screenshot of the mobile devices is restricted by operating system, to avoid unworthy applications getting device information, such as account numbers, passwords, etc. Thus, the screen features of our system will be limited.

In order to complete the functions used in our system, some adjustments in the android operating system are needed.

1.3 Research Purpose

Our research aims to combine the advantages of mobile devices and personal computers. We could enjoy the larger screen and speakers of the desktop computers or type with an actual keyboard instead of the virtual keyboard, which is not so smooth, on the mobile device. Thus, let the mobile device to retrieve the functions which are removed due to convenience.

The research platform for our experiment consists of a device based on Android operating system version 4.0.3, and a personal desktop computer or notebook equipped with Windows 7 operating system. The screen display of the mobile device is transferred to the screen of the personal computer through network connection. The users will be able to view the screen display of the mobile device on the personal computer instantly, as well as using the mouse and keyboard to simulate the input touch event on a mobile device. Even if using the keyboard to directly type the text into the device's input box. In the case during receiving or making a phone call, the audio streaming will continue to send to the personal computer and broadcast by speaker, at the same time, the voice streaming captured through the microphone of the personal computer will be send to the mobile device.

Figure *1* shows the system flow chart, our system is separated to PC control end and mobile device end. After running the program both ends connects via TCP. The application on the mobile device will constantly send screenshot as well as the output audio streaming of the device. Then, these data are passed to the PC control end to do the corresponding outputs. The user can control the mobile device's keyboard and touch event through the graphical user interface on the PC control end.

1.4 Framework

In chapter 2, we will briefly discuss on researches that are related to our work. In the following chapter, we will look through the Android kernel, permission of Superuser on the device, screen shot transfer, synchronization of audio streaming, as well as touch event and keyboard event simulation.

Finally, in chapter 4 and chapter 5, we will present the experimental results and our conclusions.

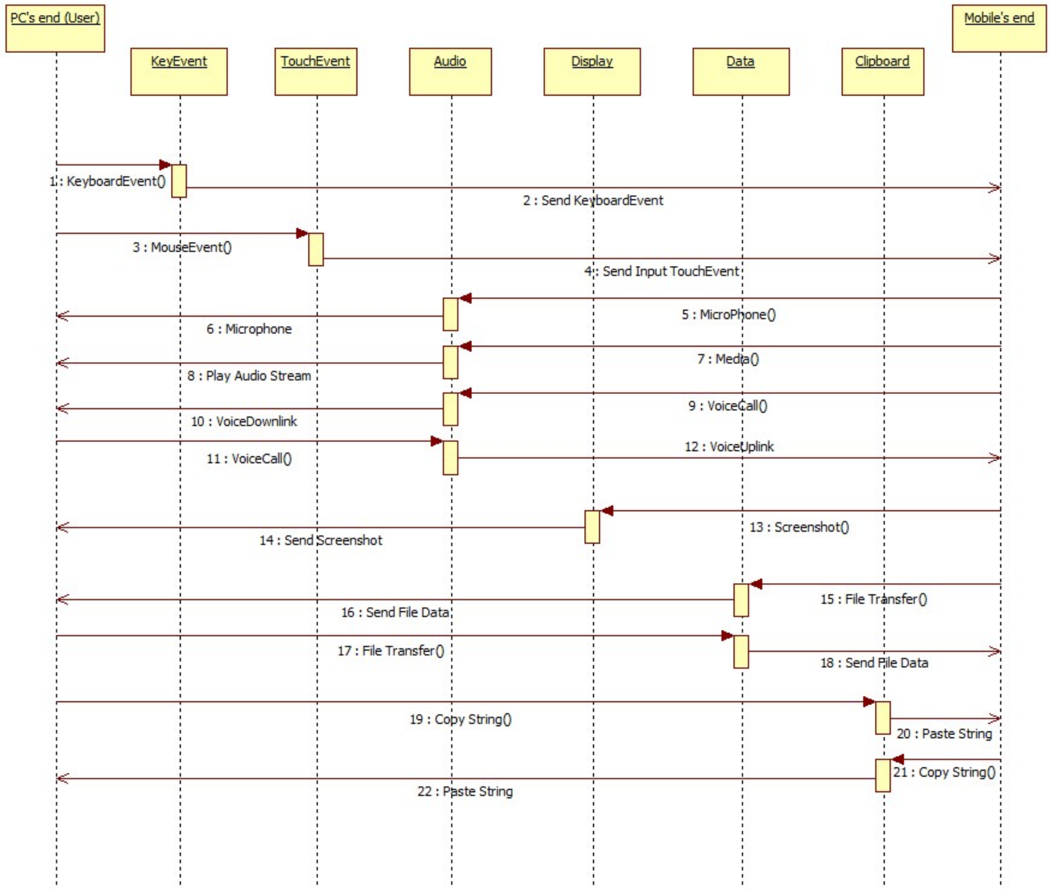


Figure 1. Sequence Diagram

2. Related Work

Connection architecture and data transmission of Remote Desktop is released by VNC (Virtual Network Computing) protocol during the earliest 2002 by American Telephone and Telegraph Company (AT & T). The architecture is mainly divided into the controlled end (client) and manipulation end (server) as shown in Figure 2.

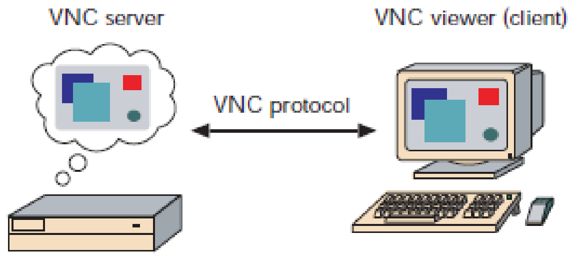


Figure 2. Structure of VNC[2]

Moreover, researches that send the data of the mobile device using remote control back for cloud computing are implemented in [8] [9] [10].

Researches which take into account that the user's PC might shutdown due to unexpected reasons, Priyanka developed a system that allows the user to boot, shutdown or restart the personal computer through network connections [11]. Other than that, there are two researches which uses RFB (remote frame buffer) transport protocol, to display the graphical user interface [12, 13].

Grant's research allows multiple users to input control simultaneously, by using cloud computing concepts, the multi-control input events is then integrated [14]. Research which focus on resolution, frame rate, compression ratio and color depth, particularly on the calculation of the transmission rate, while transferring the image[15]. In paper [16], it uses Wi-Fi Adapter as an example to show how to develop kernel files in an Android system.

In the past, there are many studies based on remote control between PC's desktop and mobile devices [1, 2, 3]. These systems can connect through the Internet to obtain real-time desktop of the remote computer, and controls made on the mobile device's end is then passed to the desktop computer to do the corresponding actions.

In 2012, Leo Sicard proposed a study that transmits the screen display of mobile device to a secondary screen instantly [4]. This study completes a similar real-time video transmission function of the remote desktop, but it does not handle the user's control and event transmission, the system is only used for viewing the screen of mobile devices on the secondary display.

Table 1. Functions of Related Development

Functions		**AirDroid**	**TeamViewer: QuickSupport**	**DroidVNC Server**	**Our System**
TouchEvent			V	V	V
Phone Data		V	V	V	V
Display			V	V	V
Media	Audio	V			V
	Video		V		V
Camera		V	V		V
Clipboard			V	V	V
Phone Call	Uplink				V
	Downlink				V

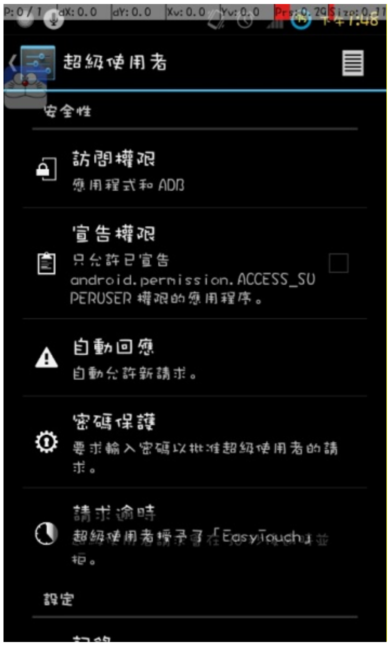
In addition, the opposite control that control mobile devices' desktop from PC also have proposed, such as TeamViewer:QuickSupport [5], AirDroid [6], Droid VNC Server [7]. In [6], AirDroid cannot display real-time screen of mobile devices on PC. In [7], Droid VNC Server is unable to provide the functions of real-time image of camera and video streaming. In [5], Team Viewer:QuickSupport cannot process the voice. Therefore, in this paper, we proposed a new approach to implement a remote desktop system that completely controls the mobile device from PC. In our system, it not only provides the above functions, but also allows to dial/receive phone call on the desktop of PC. Table 1 show the comparison of functions between our system and [5, 6, 7].

3. Methods

3.1 Permission of SuperUser on device

In experimental environment, we use HTC One V smartphone with Android 4.0.3 operating system. In order to achieve functions such as screen capture, input touch event and input keyboard etc., we must obtain the highest user privileges (root) of the device, root su executable file under the system/bin folder of your mobile device, and install the application superuser to obtain the highest authority and control all applications on the device.

Thus, the application in our research will be able to access the complete data within the operating system through the highest authority obtained, to fulfill the Remote Desktop purpose of this paper. But, not all mobile device users has or willing to get the highest authority on their mobile devices.



P: 0 / 1 dX: 0.0 dY: 0.0 Xv: 0.0 Yv: 0.0 Prs: 0.29 Size: 0.11 下午1:48

超級使用者

安全性

訪問權限
應用程式和 ADB

宣告權限 □
只允許已宣告 android.permission.ACCESS_SUPERUSER 權限的應用程序。

自動回應
自動允許新請求。

密碼保護
要求輸入密碼以批准超級使用者的請求。

請求逾時
超級使用者授予了「EasyTouch」並拒。

設定

Figure 3. SuperUser obtained on the mobile device

Figure 4 shows the diagram of the system structure scheduled to completion. It allows the user to control the mobile device completely via personal computers. This includes access of data of the mobile device, input events and obtaining information of peripheral devices.

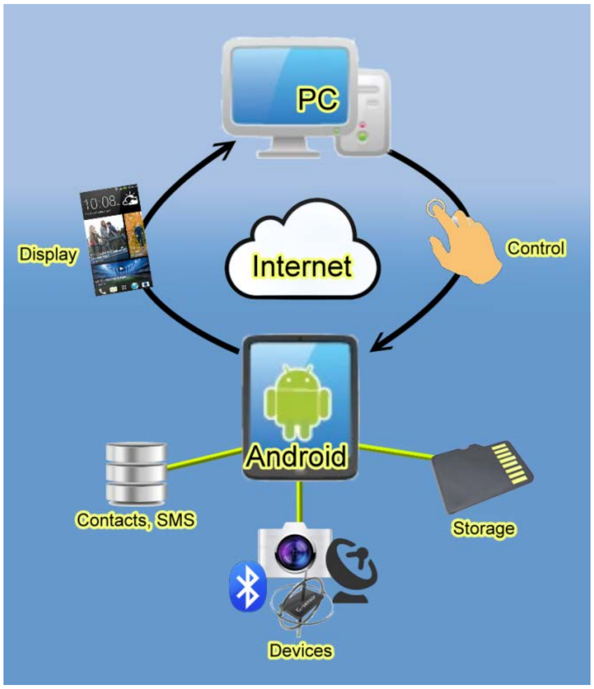


Figure 4. System Structure

3.2 Screenshot Transfer

After analyzing the structure of the Android operating system, our system displays from the libui’s EglWindows in the Libraries, then, transferred from display composition in the Surface flinger of the Framework, and then communicated between Framebuffer and hardware drivers.

Surface flinger communicates with hardware-drivers through the process of saving files, and at the same time, data transfer is made. Surface flinger will save the new screenshot in the system’s streaming file (/dev/graphics/fb0), each time the new screen image is calculated. However, the file format of the content is stored in order to drive the hardware, and it’s directly written on the equipment used. So, to be able to make use of this file in this system, this file will be converted into visible view for the users to see. The method first converts the pixel array (raw data) into RGB mode as shown in Table 2, and then uses Android’s Bitmap to convert to png.

Table 2. RAW to RGB

```
for (int m = 0; m < colors.length; m++) {
    int r = (piex[m * 4] & 0xFF);
    int g = (piex[m * 4 + 1] & 0xFF);
    int b = (piex[m * 4 + 2] & 0xFF);
    int a = (piex[m * 4 + 3] & 0xFF);
    colors[m] = (a << 24) + (r << 16) + (g << 8) + b;
}
```

3.3 Synchronization of Audio Streaming

Generally, audio structure can roughly be seen as three-parts, user, audio driver and hardware devices. The Android system divides user layer into Applications, Framework, Audio library and Audio HAL, shown in Figure 5. Due to the audio processing used in the system cannot be done with Android API (Application Programming Interface), some adjustments in the Audio library and Audio HAL layer is needed.

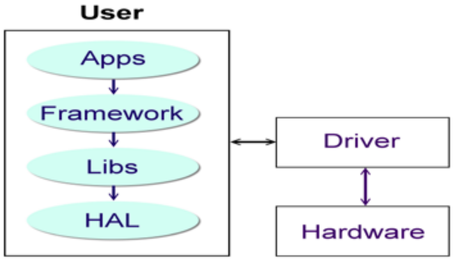


Figure 5. Android Audio Architecture

Thus, in dealing with the audio, the Android kernel processes the audio input and output most on

AudioTrack / AudioRecorder, AudioFinger and AudioPolicyService.

Wherein, AudioFinger played a role in continuously process the requests of AudioTrack / AudioRecorder from upper programs, or manage the audio equipment through Audio HAL, so, it is able to get the audio stream needed for our system if AudioFinger is modified, and then by writing JNI and Application framework, which will return the data to the Application layer.

3.4 Touch Event and Keyboard Event simulation

Processing Touch Event and Keyboard Event in Android:

1. Hardware driver reads input data from hardware
2. Write in streaming file (/dev/input/eventX)
3. Operating system detects the event files nonstop and processes it

Therefore, to handle the input events, the remote input of the user is written in the streaming file.

The only difference between Touch Event and Keyboard Event is that they have different data code, so both can be used in the same way. The code format can be found corresponds to the file (kernel/include/linux/input.h) in the Linux kernel. There are three parameters needed to be send, sequentially as: Event type, Event code and Event value. Table 2 shows the parameters our system will use.

Table 3. Event Argument

Event	Type	Code	Value
Key down	0x01	key code	1
Key up	0x01	key code	0
Touch set	0x03		x, y co-ordinates
Touch down	0x01	330	1
Touch up	0x01	330	0

4. Experiments and Results

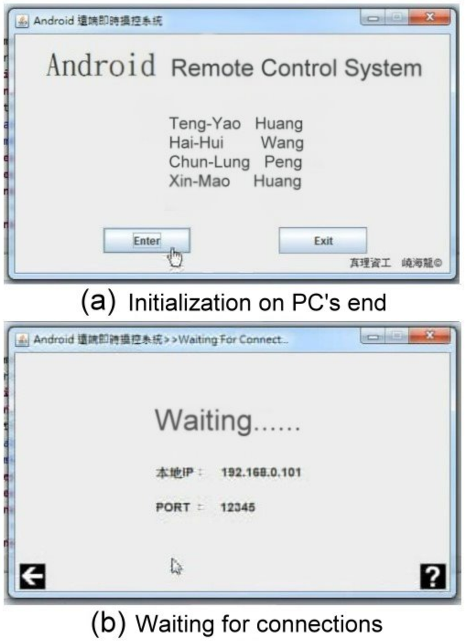
4.1 Devices

The hardware used for our research is as the follows:

1. ASUS Desktop PC，with INTEL®Core™ i5 CPU 2.80GHz, 4GB memory and 500GB hard disk drive.
2. HTC One V Smart phone，with Qualcomm MSM8255 CPU 1GHz, 3.7 inches (480*800) display, 512MB RAM、4GB ROM、Android 4.0.3 Operating System.

4.2 Remote Control Mobile Device

4.2.1 Initialization and Touch Simulation



(a) Initialization on PC's end

(b) Waiting for connections

Figure 6. Connection on PC’s end

Start up the program on the PC which then enters the on-line state, it will detect the PC’s IP address. Afterwards, start the application on the mobile device, enters the IP address to connect, as shown on Figure 6 and Figure 7.

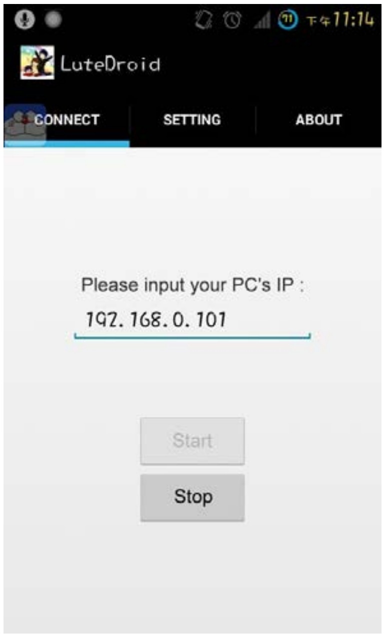


Figure 7. Connection on Device’s end

When initialize connections on both ends complete, the mobile device will synchronize two threads, one for continuously capture the screenshot from the Android operating system, it is then converted to the view back on the PC's end, the other thread will continue to capture the output of audio streaming on the device, and send to PC's end for uninterrupted playback.

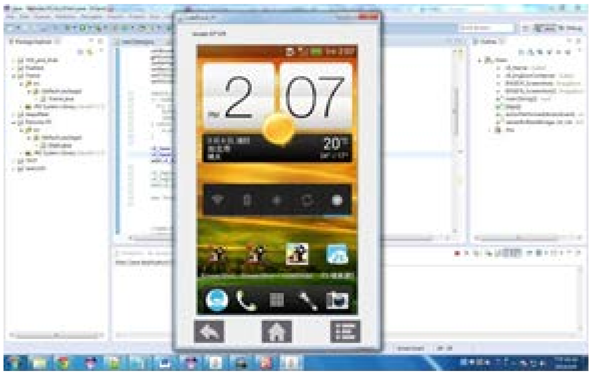


Figure 8. actual screen taken after connection

Figure 8 shows the real-time screen of the mobile device that showed on the PC's end after connection initializes. Figure 9 shows the actual device after connection. Users can directly control this screen by clicking or dragging with the mouse. The system will immediately pass input events to the mobile device's end, in order to complete the simulation of touch capabilities, and reach a remote control with a more humane graphic interface.

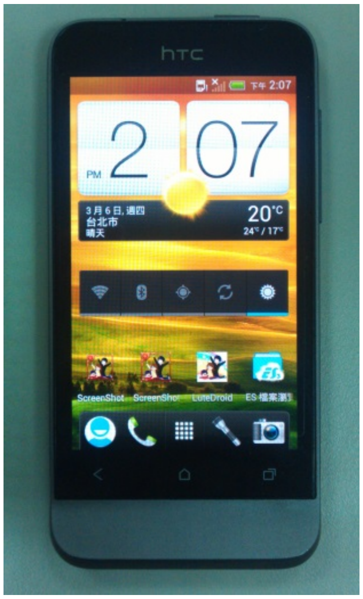


Figure 9. Actual phone image after connection

Figure 10 shows the real-time view stage when the action dragging is being done on the mouse from the PC's end. It is known that the smoothness of the screenshot on the PC's end will differ due to each operating system's update frequency.

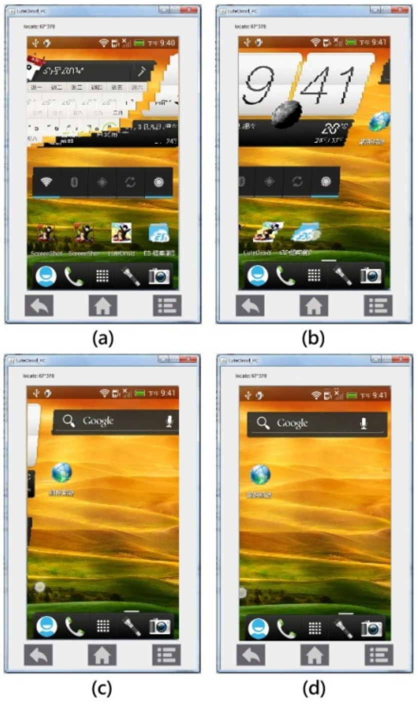


Figure 10. Drag View on PC's end

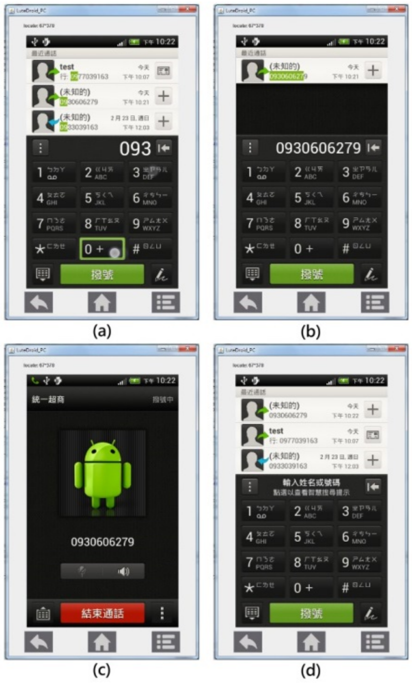


Figure 11. Phone call via remote control

4.2.2 Remote Phone call

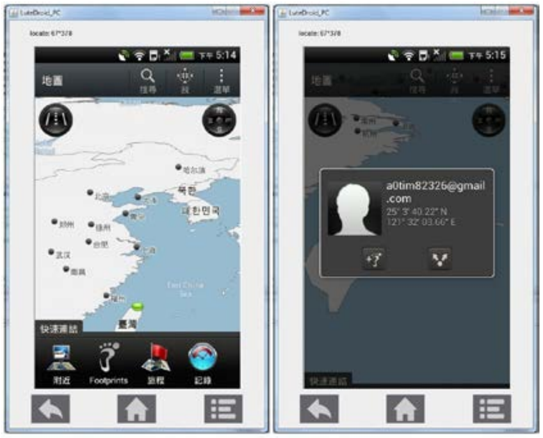
In the outgoing calls section, click the call button on the phone screen, after entering the general phone dialer screen, now, you could already use the mouse on the screen to the view history, or the numeric keypad to enter the number using the mouse. Since it has been controlled by a PC, of course, you can use the keyboard on the PC to type in phone numbers, or even copy a phone number from the web, it is allowed to copy a piece of numbers from the PC's end and paste it on the mobile device's end.

Moreover, the user can complete the phone call using the audio output of the computer, by connecting the PC to the mobile device via A2DP in the communication range of Bluetooth.

4.2.3 GPS via remote control

Through real-time simulation of touch screen operations and display, you can completely control the mobile device through computer, including running applications directly on the device.

In the experiments, we use the map applications for example, allows users to obtain accurate GPS location of mobile devices in order to determine the location of the users and mobile device. Figure 12 shows using the GPS program from the mobile device's applications on personal computers.



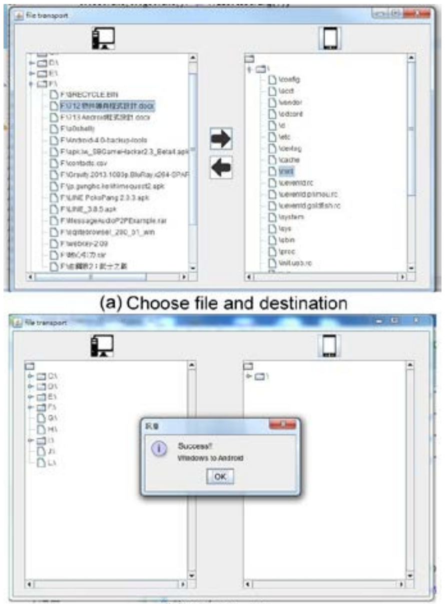
(a) Enter map app (b) Get GPS information

Figure 12. GPS application on PC's end

4.2.4 File Transfer

Figure 13 shows the bidirectional file transfer user interface, click the file transfer function keys, both PC and mobile device will access to system files structure tree, and displayed on the window's sides on the screen.

In our experiments, we transfer file from PC to mobile device for example, the file structure tree of the PC is shown on the left side of the window, the user can select the file wished to be transferred, and select the destination folder using the file structure tree of the device on the right side. Finally, click the button with the right arrow, then, the system begins to transfer the file; it will show a notification once it is completed.



Success!
Windows to Android
OK

(a) Choose file and destination

(b) Notification

Figure 13.File Transfer from PC to device

5. Conclusion & Future work

In this paper, we proposed a new approach to implement a remote desktop system that controls the mobile device from PC. This system mainly focuses on developing for Android operating system, which successfully connects mobile device to personal computer through a stable internet communication that completes the integration between both devices. In addition, we also proposed the solution to solve three significant problems which control mobile devices completely: real-time screenshots from mobile device, input event control and audio synchronization. Therefore, we successfully and effectively integrate many devices that surround in personal, and adequately use various kinds of functions that they provide.

Currently, the real-time remote control system for mobile devices is complete, but it is needed to improve the transmission of data through the Internet. Such as to match with the increasing resolution of the mobile devices, if no additional processing are made, and the screenshot is directly transferred to the personal computer, then this may led to a large data transfer which takes longer for the image to show that will cause the program to delay. In the future, will focus on the data processing before transmission; find the most suitable compression method for the current Internet speed, let the users to have the best operating experience.

References

1. Heena Aggarwal, Mayuri Kadhane, Sayali Kadam, Anjumara Inamdar, "COMPDROID -

Remote Desktop Access through Android Mobile Phone", International Journal of Science and Modern Engineering (IJISME) ISSN: 2319-6386, Volume-2, Issue-1, December 2013.

2. B.W.Balkhande, Sonam Gavhane, Rasika Phanse, Monica Sadafule, "Remote Desktop on Mobile", International Journal of Innovations in Engineering and Technology (IJIET).
3. Neha Baravkar, Supriya Kumari, Miranuddin Shaikh, Harsha Thadani, Prof. Shubhangi Kale, "Monitoring PCs using Android", International Journal of Scientific & Engineering Research, Volume 4, Issue 4, April-2013.
4. Leo Sicard, Professor Jakob E. Bardram, "TIDE - Using Device Composition on Tabletop Computers to Extend the Smartphone Experience", IT University of Copenhagen, Denmark, March 1st, 2012.
5. TeamViewer-QuickSupport (http://www.teamviewer.com/zhtw/products/mobile-device-support.aspx#android)
6. AirDroid (http://web.airdroid.com/)
7. Droid VNC Server (http://opensourceexcedio.wordpress.com/2010/10/28/droid-vnc-server/)
8. V.L.Divya, "Mobile Application Platform on Cloud Server", 2011 International Conference on Advancements in Information Technology, vol.20, 2011 IACSIT Press, Singapore.
9. Md. Sanaullah Baig, Rajasekar M., Balaji P., "Virtual Network Computing Based Remote Desktop Access", International Journal of Computer Science and Telecommunications, Volume 3, Issue 5, May 2012.
10. DHARSAN. A. MAHESH., A. MANJU., "Desktop Viewer Solution For Mobile Computing Using Virtual Network Computing", International Journal of Communications Networking System, Vol 01, Issue 02, December 2012.
11. Priyanka V. Kampasi, Y. C. Kulkarni, "Creating an Intelligent Environment in Mobile Technology", International Journal of Engineering Science and Innovative Technology, Volume 2, Issue 6, November 2013.
12. B. Gopikrishnan, Mr. A.Mani, "Desktop Solution for Mobile Environment Using Mobile Cloud Computing", International Journal of Advanced Research in Computer and Communication Engineering, Vol. 1, Issue 1, March 2012
13. M.Deepak, Mrs. P.Visalakshi, "Remote Desktop Access Using Remote Frame Buffer in Mobile Cloud Environment", International Journal of Modern Engineering Research, pp-30-34.
14. Kai Li, Grant Wallace, "Virtually Shared Displays and User Input Devices", Princeton University, Princeton.
15. L. Ciminiera, F. Lamberti, G. Paravati, A. Sanna, "A Novel Approach to Support Quality of Experience in Remote Visualization on Mobile Devices", Politecnico di Torino, Dipartimento di Automatica e Informatica, Italy
16. C. Li, "The Development of Android Software and Kernel Files by Using Example of Wi-Fi Adapter", IJCSI International Journal of Computer Science, Issues, Vol. 10, Issue 2, No 3, March 2013.