Implementation of the Remote Control and Management System in the Windows O.S

Seung-Ju Jang

Dong-Eui University, Dept. of Computer Engineering

Summary
In this paper, I implemented the remote management system on the Windows Operating System. Remote management system in the Windows operating system was designed for the client and server function. A client function has the ability to allow the system to access from the server. This feature is responsible for monitoring the packet to request access to the client on the network. Server system capability has the ability to connect to the client system. The connection is made when we direct the client's IP address which we want to access in the server program. This system was programmed with Visual Basic. The experiment for the proposed functions was conducted.

Key words:
Remote Control and Management System, Implementation of the remote control system, Windows O.S, Client Module, Sever System, Experiment

1. Introduction

Recent IT technology has developed spectacularly. In 2010, Smart phones are the topic of next-generation IT technologies. According to the advance of these IT technologies, software technology pursues convenience using mobile phones. Computer system security and control is one of the basic needs of mankind from early days. But today it has to be updated with the rapidly changing technology to ensure vast coverage, remote control, reliability, and real time operation.

In this paper, among the software technologies pursuing convenience, I review the technology of remote management system and relevant technology. This technology has two aspects. One convenient side of this function is to overcome time and space constraints. Another bad side of this function is that opponents can eavesdrop this computer using this technique. This paper focuses on explaining the details of remote management systems using the technology.

This paper designed and implemented a program of remote management system on the Windows Operating System. The proposed remote management system program was designed using the network program. There are a number of ways to manage the remote system on the Windows operating system. This paper used the Windows Sockets Protocol among several ways. The functions of remote management system are composed of server and client. The system was programmed with Visual Basic Language.

This paper is composed of related study in chapter 2. In chapter 3, the proposed designing remote control system in the Windows operating system was explained. Chapter 4 focused on the experimental results for design of this paper. Finally, Chapter 5 refers to the conclusions.

2. Related Studies

The wide area network was used as a remote control system. There are many versions of the remote management system program. There are commercial versions and the freeware versions of this program. Among those programs, UltraVNC is a powerful, easy to use and free software that can display the screen of another computer (via internet or network) on your own screen. The program allows you to use your mouse and keyboard to control the other PC remotely. It means that you can work on a remote computer, as if you were sitting in front of it, right from your current location.

The controlling computer displays a copy of the image received from the controlled computer's display screen. The copy is updated on a timed interval, or when a change on screen is noticed by the remote control software. The software on the controlling computer transmits its own keyboard and mouse activity to the controlled computer, where the remote control software implements these actions.

The quality, speed and functions of any remote desktop protocol are based on the system layer where the graphical desktop is redirected. Other products such as Microsoft RDP use a kernel driver level to construct the remote desktop for transmission of data.

The following Table 1 shows that is used as a remote management systems application.

Table 1 :Remote Management System Protocol

Software	*Protocol*	*License*	*Client/server*	*Linux client*	*Microsoft Windows client*
Anywhere TS	RDP, ICA	Proprietary	Client Only	×	○

Manuscript received August 5, 2012
Manuscript revised August 20, 2012

Apple Remote Desktop	RFB(VNC)	Proprietary	Client & Server	×	×
Apple Screen Sharing (iChat)	Proprietary, RFB(VNC)	Proprietary	Client & Server	×	×
Chicken of the VNC	RFB(VNC)	GPL	Client Only	×	×
Citrix XenApp	RDP, ICA	Proprietary	Client & Server	○	○
Crossloop	RFB(VNC)	Proprietary	Client & Server	×	○
DameWare Mini Remote Control	RDP, Proprietary	Proprietary	Client & Server	×	○
EchoVNC	RFB(VNC)	GPL	Client & Server	○	○

As shown above Table 1, there are several protocols which are used in remote management systems application. Among them, RFB (Remote Frame Buffer) protocol and the RDP(Remote Desktop Protocol) protocol are mainly used. The next chapter will organize these two protocols.

RFB (Remote Frame Buffer) and RDP are the most used international standard network protocols for remote management systems application. This paper describes these two network protocols and related researches.

Remote Desktop Protocol (**RDP**) is a proprietary protocol developed by Microsoft, which provides a user with a graphical interface to another computer. The protocol is an extension of the ITU-T T.128 application sharing protocol.[1] Clients exist for most versions of Microsoft Windows (including Windows Mobile), Linux, Unix, Mac OS X, Android, and other modern operating systems. By default the server listens on TCP port 3389.[2] Microsoft currently refers to their official RDP server software as Remote Desktop Services, formerly "Terminal Services". Their official client software is currently referred to as Remote Desktop Connection, formerly "Terminal Services Client".

Microsoft provides the client required for connecting to newer RDP versions for downlevel operating systems. Since the server improvements are not available downlevel, the features introduced with each newer RDP version only work on downlevel operating systems when connecting *to* a higher version RDP server from these older operating systems, and not when using the RDP server in the older operating system.

RFB protocol has been used in other programs, i.e. VNC, RealVNC, UltraVNC. RFB is a simple structure using the GUI(Graphic User Interface) features that allow access to the remote system. It operates at the level of the frame buffer and uses at the X11 Windows, Windows systems, and Macintosh Operating System application. Specifically, the RFB is used at the VNC (Virtual Network Computing). Termination of the remote system is called the RFB client or viewer RFB. Termination of modifying frame buffer is called the termination RFB server [7].

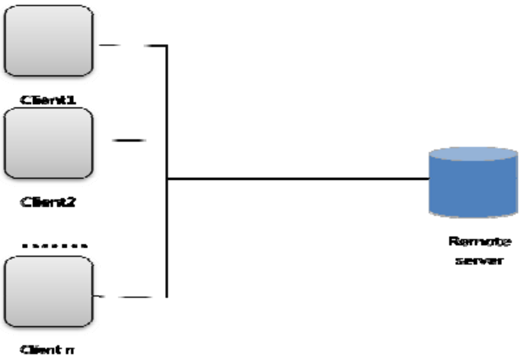


Fig. 1 the Traditional Remote Control System

Fig. 1 shows the traditional remote control system structure. A client is approaching the remote server by network. N client systems are connected to the remote server.

3. Design of the Remote System Management

This paper designs remote monitoring(management) system using a network protocol in Windows Operating Systems. The proposed system architecture is as follows Fig. 1.

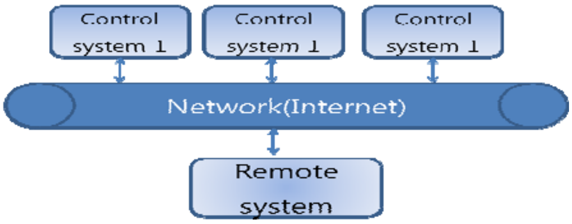


Fig. 2 Remote Control System Architecture

When network services are used to integrate business and accomplish remote control activities, the remote control program needs only to concentrate on the remote system. I design a processing framework for the Remote Control System as shown in Fig. 2. The framework is based on the network protocol and can include different side IP client.

The proposed remote management system is divided into a server functions and client functions. The server function enables you to manage the client system remotely by specifying a controlling system. Server function has remote management capabilities for managing the target(client) system. This operating feature is the following structure

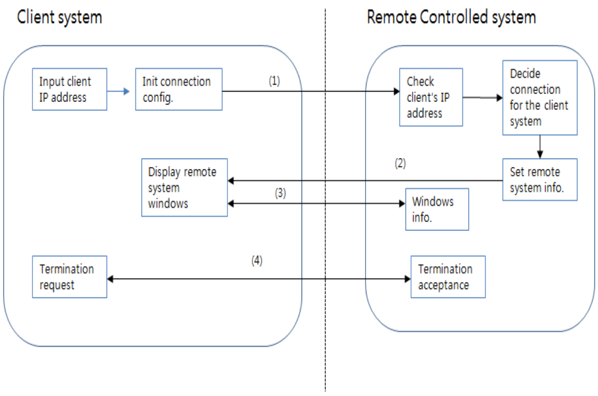


Fig. 3 init, connection, display, and termination Work Flow

Work flow of the remote control system is shown above Fig. 3. The server machine's(remote controlled system) IP address is entered on program. If the system is encountered, initializing configuration for setting up the connection. If the remote controlled system is not found, the system cannot connect two systems. When the system is initialized by the connection settings, the server system will connect using network protocol(as above (1)).

Roles and message flows related to the remote control system in Fig. 3. Remote controlled system gets client IP address and decide connection this system. Success execution of the remote controlled system will trigger the execution of its remote system information that is sent to the client system. The client coordinator should understand the remote windows system information received from the controlled system(as above (2)). Client module sends request codes and server module receives this information and resending windows system information to the client again(as above (3)). The termination code is achieved before termination. Client module decides the exit process. Client module sends terminates connection to the remote controlled system.

The program feature of clients function is running in the client system. The flow behavior of the client functions are shown Fig. 3, too.

To control remote client system, work flow of the server is shown above Fig. 3. When the system is initialized by the client connection settings, the server system will connect using Winsocket network protocol. The program feature of client function is running in the client system.

The client program is initialized first. After initializing the client program, the client system allows to access from the server system. If there is a request to connect from the server system, the client program should be allowed to connect from the server system. After connection from the server, it initializes the display information and display information of the client system is transmitted to the server system. If a server or a client system requests a connection termination, the connection is terminated from both systems. Otherwise, the display information of client system is sent to the server system continuously.

```
For
For
Capture screen data of specific axis value;
Capture windows screen;
If sending data Then
    Sending display data;
Else
    Increment X axis value to 1;
End If
End for
Initialize X axis to 0;
Increment Y axis value to 1;
Capture windows screen;
If sending data Then
    Sending display data;
Else
    Increment X axis value to 1;
End If
End for
Initialize Y axis to 0;
```

Fig. 4 server side program

Fig. 4 is a server side program. First, the program capture screen data of specific axis value. And it capture windows screen. After the program is captured screen, it will first need to send display data. If no more sending data, it will just increment X axis value to 1. When X axis data sending is finished, it capture windows screen and will first need to send display data until no more Y axis.

```
Initialize display system;
Set the window size (X, Y) for display information;
Set the display indication block size;
Set the navigator display indication block size;
If current window's size is bigger than remote PC window's size,
decrement display screen size;
Set display screen window to central;
Display the remote windows;
```

Fig. 5 client side program

Fig. 5 is a client side program. First, the program should be able to initialize display system. Next, the program set the window size (X, Y) for display information and set the display indication block size. And it set the navigator display indication block size. If current window's size is bigger than remote PC window's size, the program decrements display screen size. It set display screen window to central and display the remote windows.

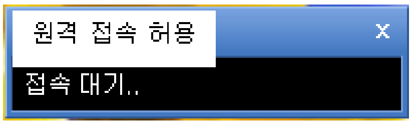
4. Experiment and Evaluation

The design of the proposed system was actually implemented on the system. The designing implementation was experimented how the system is operating normally. The experiment environment is as follows. In order to implement the proposed idea, the hardware of the system is is comprised of Windows XP Operating System, Core 2 duo 2.13GHz CPU and 2 GBytes of memory.

The remote program development environment is supported by insight desktop a comprehensive network and MS VBasic compiler tool-chain.

The experiments for the implementation system was composed of the client drive capability, the server drive capability, and an integrated environment driving capability. I can use the interface of the implemented program to remotely control the instrument and for an immediate remote system. The implemented program running in a server and client system is used to link the network protocol.

To test the client drive capability, the server program specifies the client computer IP address, and the client program is installed at the client system. The actual program was run on the client system like Fig. 6. Fig. 6 shows the implementation result in the remote client system and the control by the remote operator interface.

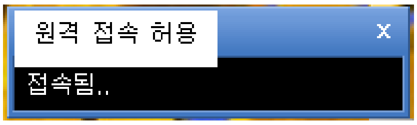


원격 접속 허용

접속 대기..

Fig. 6 The Client Application Program waits for Response Server

Fig. 6 shows the client program user interface and its one interface item, the connection to the remote system. When testing, a remote server system can access the system that running the client program. The program runs as a sort of daemon process. If the other party is made connection, the program responds to a connection.



원격 접속 허용

접속됨..

Fig. 7 Server Permits Connection from Client

Fig. 7 shows the connection client system from server user interface and its one interface item, the connection to the client system. When testing, a remote server connects to the client system, the following text message which is "waiting for connection" is changed to "Connected".

The test has been carried out under conditions where all the contents of the client system can be accessed to enable remote management permission.

Contrary the behavior of the client system, Fig. 8 shows the screen of a server program is operating on the server machine. I can see that our design has that if entering the IP address of the remote access system, the remote system will be connected.

As for normal operation in this screen control in Fig. 8, after successful login from server to the remote(client) systems, the environment will operate like that the server computer will be directly connected to a remote system. After successful conduction to the remote system, Fig. 8 shows the behavior screen of the program.

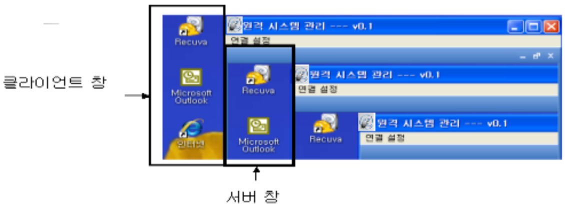
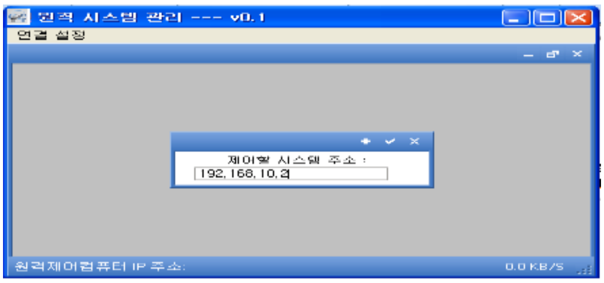


Fig. 8. Client Window Connection Status



원격 시스템 관리 --- v0.1

연결 설정

제어할 시스템 주소 :

192.168.10.2

원격제어컴퓨터 IP 주소:

0.0 KB/S

Fig. 9. The Client IP Address Input Screen

Fig. 8, Fig. 9 shows the screen of the connecting procedure from the server system to the client system. In Fig. 7, The system first asks the user to set the IP address of the emote client system.

Then the system begins to allow for the access of the client. If the client systems are allowed to connect, the program displays a normal message on the screen, asks the user to confirm whether the connection correct and then connects the program. The server windows display the client's sending information. The window of the server system will output unchanged information from the client system. Fig. 9 shows the output results.

Fig. 9 shows the result of experiment that the design of remote management information system in this paper. Its two test parts, client test, and server test. As this test, the server computer execution environment can be sent to the client computer fully.

5. Conclusions

I use Visual Basic language to design a test program, the remote control system for the Windows operating system. I also develop the client and the server features in the Windows system. I coordinate these two test programs to measure the data connectivity. The client function has the ability to allow access from the server system.

This feature is responsible for monitoring the packet to request access to itself on the network. The server has the ability to connect to the system which it wants to manage remotely. When the connection is made, the server program sets the client machine's IP address.

I test the correctness of the proposed functions in this paper. The test was divided into the client function and server function accurately. The client function is able to allow to connect in case of requesting from a remote server. As a result of tests, I confirmed that this feature operates normally. Similarly, on server function the test was worked out after input the client machine's IP address. As a result of testts, I confirmed that the client system was connected normally.

In this paper, I designed the remote management system in the Windows operating system that can be used for the purpose of convenience.

References

[1] VNC(Virtual Network Computing), www.wikipedia.org, 2010. 5.

[2] Ying-Wen Bai and Chia-Yi Chang, “Design and Implementation ofan Integrated Remote Test System for Mobile Phones”, IEEE International Symposium on Industrial Electronics, 2009.7.

[3] RDP protocol, www.wikipedia.org, 2010. 5.

[4] Microsoft, “Remote Desktop Protocol(RDP) Features and Performance, Microsoft, 2009.

[5] Tristan Richardson, “The RFB Protocol”, RealVNC Ltd, 2009. 11.

[6] Yiweih Zhang, Xiaohui Duan, Jingjie Wang, and Laixian Zhang, " Design and Implementation of Wireless Monitoring System Based on Windows Mobile ", 4th International Conference on Wireless Communications, Networking and Mobile Computing, WiCOM '08, pp.I-4, 2-14 Oct. 2008.

[7] C.H.Lee et. al., Remote Control System Using Web", Journal of Electrical Engineering, Vol. 39., No. 3, 2002. 9.

[8] Ying-Wen Bai and Chin-Chung Lee, "Design and Implementation of an Automatic Testing System for MP3 Players," IEEE Instrumentation and Measurement Technology Conference Proceedings, IMTC 2008 , pp.2205-22I 0, May 2008.

[9] B.S, Yoon, "Log Indexing for Remote System Diagonis in Mobile Environments", Gong-Ju Univ. M.S Thesis, 2003. 11.

[10] Baris Yuksekkaya, A. Alper Kayalar, M. Bilgehan Tosun, M. Kaan Ozcan, and Ali Ziya Alkar. "A GSM, Internet and Speech Controlled Wireless Interactive Home Automation System" IEEE Transactions on Consumer Electronics, Vol.52 No. 3, pp: 837-843,2006.

[11] Park, Jung so, “Design of Remote Management System Using Wireless Communication”, Kyoung-Hee Univ. M.S Thesis, 2009.11

Seung-Ju, Jang received a B.Sc. degree in Computer Science and Statistics, and M.Sc. degree, and his Ph.D. in Computer Engineering, all from Busan National University, in 1985, 1991, and 1996, respectively. He is a member of IEEE and ACM. He has been an associate Professor in the Department of Computer Engineering at Dongeui University since 1996. He was a member of ETRI(Electronic and Telecommunication Research Institute) in Daejon, Korea, from 1987 to 1996, and developed the National Administration Multiprocessor Minicomputer during those years. His current research interests include fault-tolerant computing systems, distributed systems in the UNIX Operating Systems, multimedia operating systems, security system, and parallel algorithms.