

텍스트와 이미지로 구성된 클립보드 콘텐츠를 위한 VNC 서버와 클라이언트의 확장

이 태 호*, 이 흥 창**, 박양수***, 이명준****

Extending VNC Server and Client for Sharing Clipboard Contents Composed of Text and Images

Tae-Ho Lee *, Hong-Chang Lee **, Yang-Su Park ***, Myung-Joon Lee ****

요 약

VNC(Virtual Network Computing)는 RFB (Remote Framebuffer) 프로토콜을 사용하여 데스크탑을 공유하는 소프트웨어로, VNC 서버가 실행되고 있는 원격 컴퓨터를 로컬 컴퓨터에 설치된 VNC 클라이언트(또는 뷰어)를 통해 제어할 수 있다. 원격 컴퓨터와 로컬 컴퓨터의 정보를 서로 교환하기 위하여, VNC는 클립보드 콘텐츠를 공유할 수 있는 기능을 제공한다. 그러나 현재 사용중인 VNC 소프트웨어들은 단지 텍스트 형식의 클립보드 콘텐츠를 공유하는 기능을 제공하며, 이미지와 같은 멀티미디어 클립보드 콘텐츠를 공유하는 방법을 제공하고 있지 않다.

본 논문에서는 텍스트와 이미지로 구성된 클립보드 콘텐츠를 공유할 수 있도록 RFB 프로토콜을 확장한다. 또한 개발된 프로토콜을 지원하기 위하여 무료 오픈소스 소프트웨어인 UltraVNC 서버와 JavaViewer 클라이언트를 확장한다. 개발된 VNC 소프트웨어를 통하여, 사용자는 원격 컴퓨터와 로컬 컴퓨터 사이에서 이미지와 텍스트가 포함된 클립보드 콘텐츠를 공유할 수 있다.

Abstract

VNC(Virtual Network Computing) is a desktop sharing system based on the RFB(Remote Framebuffer) protocol which allows you to control a remote computer running a VNC server through a VNC client(or viewer) on a local computer. To exchange information between the two computers, VNC provides the functionality of sharing the clipboard contents. Unfortunately, the current VNC softwares support only the clipboard text contents, not providing methods for sharing the clipboard multimedia contents such as images.

In this paper, we extend the RFB protocol to share the clipboard contents composed of text and images. Also, to support the developed protocol, we extend both the UltraVNC server and the JavaViewer VNC client which are free open-source softwares. Through the developed VNC softwares, users can exchange the clipboard contents including texts and images between the remote computer and the local computer.

▶ Keyword : VNC, 원격 제어 소프트웨어(Remote Access Software), RFB 프로토콜(RFB protocol), UltraVNC, JavaViewer, 클립보드(Clipboard)

• 제1저자 : 이태호 교신저자 : 이명준

• 접수일 : 2008. 4. 1, 심사일 : 2008. 6. 15, 심사완료일 : 2008. 7. 25.

* 울산대학교 컴퓨터정보통신공학부 석사과정 ** 울산대학교 컴퓨터정보통신공학부 박사과정

*** 울산대학교 컴퓨터정보통신공학부 교수 **** 울산대학교 컴퓨터정보통신공학부 교수

※ 본 연구는 지식경제부 및 정보통신연구진흥원의 대학 IT 연구센터 육성지원사업의 연구결과로 수행되었습니다. (IITA-2008-(C1090-0801-0039))

I. 서론

VNC(Virtual Network Computing)[1,3]는 사용자가 위치한 지역의 로컬 시스템을 이용하여 다른 지역에 위치한 원격 시스템의 자원을 사용할 수 있도록 하는 원격 제어 도구이다. VNC는 CUI(Character User Interface) 방식의 텔넷 원격접속 환경과는 다른, GUI(Graphic User Interface) 방식으로 원격 시스템의 자원을 사용할 수 있는 환경을 제공한다. 1999년 VNC가 AT&T에서 개발되어 소스 내용이 공개된 이래 Real VNC[3], Tight VNC[4], Vine Server[5], UltraVNC[2] 등의 여러 형태의 오픈소스 형태로 개발되고 있다. 또한 이와 비슷한 원격 제어 프로그램으로 윈도우즈 플랫폼에서 사용가능한 Microsoft 윈도우즈 XP, 2003등에 내장된 Remote Desktop 프로그램과 시맨틱의 PcAnywhere 프로그램[6] 등이 있다. 이러한 원격 제어 프로그램들은 상용 프로그램으로 제작되어, 영리를 목적으로 배포되기 때문에 지원하는 플랫폼이 한정되어 있다. 그러나 오픈소스로 개발된 VNC 프로그램은 Mac OS 플랫폼에서 사용가능한 버전, PDA에서의 원격시스템 제어가 가능한 버전 등 다양한 플랫폼에서 사용할 수 있도록 지원되고 있다. 또한 윈도우즈 XP 운영체제의 Remote Desktop 원격 제어 프로그램은 멀티세션을 지원하지 않아, 하나의 시스템을 하나의 사용자만이 사용 가능하다. 이와 반면에, 리눅스 운영체제나 윈도우즈 2003 운영체제의 경우 멀티세션을 지원하므로 VNC를 사용하면 여러 클라이언트가 동시에 서로 다른 화면에서 원격 시스템을 제어할 수 있다.

VNC를 비롯한 여러 원격 제어 프로그램은 로컬 시스템과 원격 시스템 서로간의 클립보드 내용을 공유할 수 있는 기능을 제공하고 있다[7]. 클립보드는 사용자가 임의로 선택한 콘텐츠를 잠시 복사해두는 임시 기억공간을 말하며, 사용자는 클립보드에 복사한 콘텐츠를 필요에 따라 클립보드에서 읽어서 사용할 수 있다. 따라서 클립보드 공유기능을 사용하면 원격 시스템과 로컬 시스템간의 다양한 작업에서 데이터를 입력하고 수정하는 작업의 진행을 효율적으로 진행할 수 있다. Remote Desktop의 경우 원격 시스템과 로컬 시스템간의 클립보드를 공유하여 텍스트뿐만 아니라 이미지, 윈도우즈 오브젝트와 같은 다양한 형태의 콘텐츠를 공유하여 사용할 수 있다. 오픈소스로 공개된 VNC 프로그램 역시 클립보드 공유가 가능하지만, Remote Desktop과는 달리 유니코드 형식의 텍스트 공유만이 지원되고 있다. 따라서 원격 시스템과 로컬 시스템의 클립보드 공유를 사용하여 다양한 형태의 콘텐츠를 공

유할 수 있도록 VNC 프로그램을 확장할 필요가 있다.

본 논문에서는 VNC 프로그램을 확장하여 원격 시스템과 로컬 시스템의 클립보드에 저장된 다양한 형태의 콘텐츠 중에서, 텍스트와 이미지가 혼합된 콘텐츠 형태를 공유할 수 있는 방법에 대하여 기술한다. 이를 위해 VNC 프로그램이 사용하는 RFB 프로토콜[8]을 확장하며, 또한 확장된 RFB 프로토콜을 지원하도록 무료 오픈소스 소프트웨어인 UltraVNC 서버와 JavaViewer 클라이언트[2,3,4]를 확장한다. RFB 프로토콜에는 원격 시스템과 로컬 시스템간의 텍스트 형태의 클립보드 콘텐츠를 공유할 수 있는 메시지가 정의되어 있다[8]. 이러한 RFB 프로토콜 메시지를 확장하여 클립보드 이미지를 공유할 수 있도록 하고, 확장한 RFB 프로토콜을 지원하도록 VNC 서버와 클라이언트를 개발하여 기존의 텍스트와 함께 이미지가 포함된 클립보드 콘텐츠를 공유할 수 있도록 한다.

본 논문의 구성은 다음과 같다. 1장에 이어 2장에서는 VNC의 소개, VNC 서버와 클라이언트의 구조와 동작과정, VNC가 사용하는 RFB 프로토콜에 대한 소개를 하고, 3장에서는 RFB 프로토콜과 VNC가 지원하는 클립보드 텍스트 공유를 설명한다. 4장에서는 클립보드 이미지를 공유하기위해 확장한 RFB 프로토콜과 서버와 클라이언트를 설명한다. 5장에서는 확장한 VNC의 효율성에 대해 살펴며 6장에서는 결론과 향후 연구 과제를 제시하고 있다.

II. 관련연구

2.1 VNC 서버와 클라이언트

VNC는 먼 거리에 위치한 시스템을 GUI(Graphic User Interface)를 통해 원격 제어하기 위한 도구이다. VNC를 사용하면 X-window, 윈도우즈, Mac OS 등과 같은 시스템을 GUI를 통해 원격 제어할 수 있다. VNC는 서버와 클라이언트 프로그램으로 구성된다. VNC 서버는 원격 시스템의 화면 정보를 클라이언트에게 RFB 프로토콜을 통해 전달하는 역할을 한다. VNC 클라이언트는 VNC서버로부터 전달받은 RFB 프로토콜 메시지의 내용을 클라이언트 사용자가 볼 수 있도록 표시하며, 로컬 사용자의 입력을 RFB 프로토콜 메시지를 통해 VNC서버에 전달하는 역할을 한다. VNC는 다중 세션을 지원하며 VNC 서버에 접속하는 사용자는 서버에 설정된 특정포트와 VNC서버가 동작하는 IP 주소를 사용하여 접속할 수 있는데, 각 세션에 접속하기 위해서는 접속을 시도하는 포트번호를 사용해 해당 세션에 접속할 수 있다. 하나의

세션에는 동시에 여러 클라이언트가 접속할 수 있다.

대표적인 VNC 프로그램으로 Real VNC와 Tight VNC, UltraVNC가 있다. Real VNC와 Tight VNC는 리눅스와 윈도우즈에서 사용가능한 서버와 클라이언트, 그리고 자바로 작성된 클라이언트를 제공하고, UltraVNC는 윈도우즈 플랫폼에서만 사용가능한 서버와 클라이언트를 제공하며 자바로 작성된 JavaViewer 클라이언트를 제공한다.

2.2 UltraVNC 서버와 JavaViewer 클라이언트

2.2.1 UltraVNC 서버

UltraVNC 서버는 Real VNC와 Tight VNC를 확장하였으며 오직 윈도우즈 플랫폼에서 작동한다. 윈도우즈 플랫폼에 특화된 장점으로 윈도우즈의 사용자 계정을 사용해서 원격 시스템에 접근 할 수 있는 기능을 제공하고 있으며 원격 시스템의 화면을 전송하는 오버헤드를 줄일 수 있는 Mirror Driver 기능[2]을 제공한다. 그리고 전송 메시지 암호화 기능으로 DSM PlugIn이라는 기능[2]을 제공하여, 서버와 클라이언트가 주고받는 메시지를 암호화 할 수 있는 방법을 제공한다.

2008년 2월 현재, 전 세계의 운영체제의 시장 점유율 조사 결과(그림1)[9]를 보면 윈도우즈 플랫폼의 시장 점유율이 91.46%를 가지고 있다. 따라서 많은 사람들이 사용하는 윈도우즈 플랫폼을 원격 제어 하기에 가장 효과적인 VNC 프로그램은 UltraVNC이다.

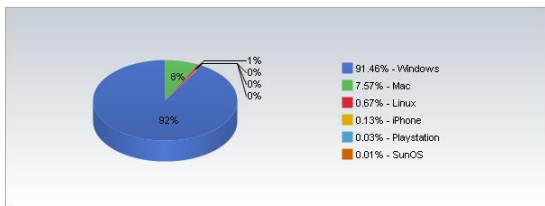


그림 1. 2008년 2월 운영체제 시장 점유율 조사
Fig. 1. Operating System Market Share for February, 2008

2.2.2 JavaViewer 클라이언트

JavaViewer 클라이언트는 자바 언어로 작성된 VNC 클라이언트 프로그램으로, JavaViewer 혹은 Java VNC 클라이언트라고 부르고 있다. JavaViewer의 경우 웹브라우저에서 애플릿으로도 작동이 가능하며, 특정한 플랫폼에서 사용가능하도록 작성된 플랫폼 전용 VNC 클라이언트 프로그램과 제공하는 기능은 동일하다. JavaViewer는 플랫폼 전용 VNC 프로그램에서와 같이 원격 시스템의 화면을 전송할 때 사용할 인코딩 방식과, 클라이언트의 입력을 무시하도록 하는

View Only 기능을 활성화 또는 비활성화 할 수 있다.

2.3 RFB 프로토콜

RFB(Remote FrameBuffer)[8]프로토콜은 인터넷을 통해 먼 거리에 있는 시스템을 GUI 기반으로 제어하기 위한 프로토콜이다. RFB 프로토콜을 이용한 GUI 원격 제어 도구로는 VNC가 있으며, VNC는 버전 별로 다양한 기능을 제공하기 위하여 RFB 프로토콜을 확장하여 사용하고 있다. RFB 프로토콜은 사용한 통신은 다음과 같이 세 가지 과정을 가진다.

- handshaking : 서버와 클라이언트의 RFB 프로토콜 버전을 확인하고, security type을 확인하는 과정이다.
- Initialization : 서버와 클라이언트가 서로 RFB 프로토콜을 주고받을 수 있는 준비 작업이 완료되었다는 메시지를 주고받는 과정이다.
- Server-to-Client, Client-to-Server Message : 클라이언트가 서버에게 원격 시스템에 대한 명령을 전송하고, 서버로부터 처리 결과를 전송받는 과정이다.

1)과 2)과정은 원격 제어를 하기 위한 준비 과정이며 원격 제어 메시지와 화면전송은 3)과정에서 일어난다. Server-to-Client 메시지는 원격시스템의 화면을 서버로부터 클라이언트로 전송하는데 사용된다. 원격시스템의 화면을 전송하기 위하여 프레임 버퍼를 사용하며, 화면의 일부 변경된 부분을 전송하도록 하고 있다. Client-to-Server 메시지는 클라이언트가 서버에 마우스나 키 이벤트를 전달하거나 해당 세션의 원격 제어 환경을 설정하는데 사용한다. 다음 [표 1]은 Server-to-Client 메시지의 종류와 역할이며, [표 2]는 Client-to-Server 메시지의 종류와 역할을 보이고 있다.

표 1. Server-to-Client message의 종류
Table 1. Classification of the Server-to-Client messages

Number	Name	비고
0	FramebufferUpdate	갱신된 화면의 프레임 버퍼 전송
1	SetColourMapEntries	색상 맵 정보
2	Bell	비프음을 내도록 함
3	ServerCutText	서버 클립보드의 텍스트 정보

표 2. Client-to-Server message의 종류
Table 2. Classification of the Client-to-Server messages

Number	Name	비고
0	SetPixelFormat	픽셀의 포맷을 설정
2	SetEncoding	픽셀의 압축 방식을 설정
3	FramebufferUpdateRequest	프레임 버퍼의 갱신 요청
4	KeyEvent	키보드 이벤트
5	PointerEvent	마우스 이벤트
6	ClientCutText	클라이언트 클립보드의 텍스트 정보

III. VNC의 클립보드 텍스트 공유

멀티프로그래밍을 지원하는 운영체제에서는 프로세스 간의 자료교환을 위한 수단으로 클립보드 기능을 지원하고 있다. 클립보드는 사용자가 임의로 선택한 콘텐츠를 잠시 복사해두는 임시 기억공간을 말하며, 사용자는 클립보드에 복사한 콘텐츠를 필요에 따라 클립보드에서 읽어서 사용할 수 있다. 클립보드에 저장될 수 있는 콘텐츠의 형식은 텍스트와 이미지를 비롯한 다양한 형식의 콘텐츠가 될 수 있으며 서로 혼합된 콘텐츠도 가능하다. RFB 프로토콜에서는 원격시스템과 로컬시스템의 클립보드에 저장된 텍스트 콘텐츠를 서버와 클라이언트가 서로 전송할 수 있도록 메시지를 정의하고 있으며, VNC는 이것을 통하여 원격시스템의 클립보드와 로컬시스템의 클립보드에 저장된 텍스트 콘텐츠를 서로 공유할 수 있는 기능을 제공하고 있다.

VNC를 사용하여 원격 저작 작업을 수행할 때 원격시스템과 로컬시스템간의 자료를 주고받기 위해서 VNC의 클립보드 공유기능을 사용하면 편리하다. 그러나 현재의 RFB 프로토콜을 통해서 VNC는 텍스트 콘텐츠만 공유할 수 있으며 이미지 콘텐츠를 공유하기 위해서는 FTP와 이메일, 웹 등과 같은 파일 전송을 이용하여야 한다. 따라서 클립보드 공유를 통하여 간편하게 이미지 콘텐츠도 공유할 수 있도록 확장하기 위하여, 현재 구현되어 있는 RFB 프로토콜과 VNC의 클립보드 공유방법을 살펴보도록 한다.

3.1 클립보드 텍스트 전송을 위한 RFB 프로토콜 메시지

RFB 프로토콜에서 원격시스템과 로컬시스템이 서로 클립보드 콘텐츠 전송하여 그 내용을 공유하는 것은 클라이언트의 요청을 서버가 처리하는 Server-to-Client, Client-to-Server 메시지 전달 과정에 해당된다. Server-to-Client 메시지와 Client-to-Server 메시지에는 원격시스템의 클립보드 텍스트를 로컬시스템으로 전송하는 ServerCutText와 로컬시스템의 클립보드 텍스트를 원격시스템으로 전송하는 ClientCutText 메시지를 정의하고 있다. 각 메시지는 메시지 형식 필드에 [표 1]과 [표 2]에서 보이고 있는 Number 값을 사용하여 전송하는 메시지 형식을 수신측에서 구별할 수 있도록 하며 메시지가 전송할 클립보드 텍스트의 길이와 텍스트 내용을 담을 수 있도록 정의되어 있다. 다음 [표 3]과 [표 4]는 각각 ServerCutText와 ClientCutText의 구조이다.

표 3. ServerCutText 메시지 구조
Table 3. Structure of the ServerCutText message

No. of bytes	Type	(Value)	Description
1	unsigned char	3	메시지 형식
3			padding
4	unsigned char(4)		텍스트 길이
length	unsigned char(length)		텍스트 내용

표 4. ClientCutText 메시지 구조
Table 4. Structure of the ClientCutText message

No. of bytes	Type	(Value)	Description
1	unsigned char	6	메시지 형식
3			padding
4	unsigned char(4)		텍스트 길이
length	unsigned char(length)		텍스트 내용

[표 3]과 [표 4]와 같은 구조로 정의된 ServerCutText와 ClientCutText 메시지를 지원하는 UltraVNC 서버와 JavaViewer 클라이언트는 원격시스템과 로컬시스템의 클립보드 텍스트를 공유할 수 있는 기능을 제공한다. 다음 [그림 2]에서는 원격시스템 클립보드에 저장된 'Server'라는 텍스트와 로컬시스템 클립보드에 저장된 'Client'라는 텍스트를 ServerCutText와 ClientCutText 메시지를 통하여 전송하는 모습을 보이고 있다.

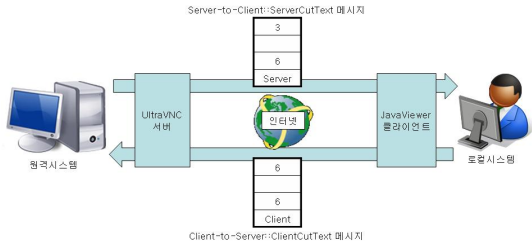


그림 2. ServerCutText 메시지와 ClientCut-Text 메시지를 통한 클립보드 텍스트 공유의 예
 Fig. 2. Example of sharing clipboard text through the ServerCutText message and the ClientCutText

3.2 UltraVNC 서버와 JavaViewer클라이언트의 클립보드 텍스트 공유

UltraVNC 서버와 JavaViewer 클라이언트의 클립보드 텍스트 공유 구현을 살펴보기 위해서 UltraVNC 서버와 JavaViewer 클라이언트가 클립보드 텍스트를 전송하는 RFB 프로토콜을 지원하는 방법을 살펴도록 한다. RFB 프로토콜에서는 ServerCutText와 ClientCutText 메시지를 통하여 클립보드 텍스트를 전송하고 있으므로 각 메시지를 지원하는 UltraVNC 서버와 JavaViewer 클라이언트의 기능을 살펴도록 한다.

3.2.1 원격시스템에서 로컬시스템으로의 클립보드 텍스트 전송

UltraVNC 서버는 원격시스템에서 발생하는 클립보드에 저장된 콘텐츠가 갱신되는 등의 윈도우 이벤트와 화면의 변경된 사항을 RFB 프로토콜의 Server-to-Client 메시지를 통해 로컬시스템의 클라이언트에게 전달한다. VNC서버는 이러한 전송을 여러 클래스로 나누어 처리한다. 다음 [표 5]는 VNC서버의 주요 클래스와 그 역할을 보이고 있다.

표 5. UltraVNC 서버의 주요 클래스와 역할
 Table 5. UltraVNC major classes and their actions

클래스	역할
vncServer	UltraVNC 서버의 각 클래스의 객체와 스레드를 생성하는 역할을 하며 클라이언트의 접속과 해제를 처리한다.
vncDesktop	원격시스템의 화면의 변경과 윈도우 이벤트를 감지한다. 클라이언트로부터 전송받은 명령을 원격시스템에서 실행한다.

vncClient	접속된 클라이언트 하나당 하나의 vncClient 객체가 생성된다. 클라이언트로부터 전달 받은 Client-to-Server 메시지를 처리하며 해당 클라이언트에게 Server-to-Client 메시지를 전송한다.
vncClientList	vncClient 객체들을 저장할 수 있는 클래스이다.
vsocket	소켓을 통한 메시지 전송과 수신을 담당한다.

UltraVNC 서버가 각 클래스에 구현된 함수를 사용하여 클라이언트에게 클립보드를 전송하는 과정은 다음과 같은 과정을 따른다.

- 1) vncDesktop 객체는 클립보드의 내용이 갱신되었다는 것을 감지하고 vncServer 객체의 updateClipText 함수를 실행한다.
- 2) vncServer의 updateClipText 함수는 vncClientList 객체를 이용해 모든 vncClient 객체의 updateClipText 함수를 실행한다.
- 3) vncClient 객체의 updateClipText 함수는 vsocket 객체를 통해 ServerCutText 메시지를 해당 클라이언트에 전송한다.

JavaViewer 클라이언트는 서버로부터 전송받은 Server-to-Client 메시지를 통하여 원격시스템의 화면을 사용자에게 표시한다. 다음 [표 6]은 JavaViewer 클라이언트의 주요 클래스와 그 역할이다.

표 6. JavaViewer 클라이언트의 주요 클래스와 역할
 Table 6. JavaViewer major classes and their actions

클래스	역할
UltraVncViewer	JavaViewer를 실행하기 위한 main 메서드와 init 메서드가 있으며 JavaViewer의 각 객체와 스레드를 생성한다. JavaViewer 윈도우 이벤트를 처리한다.
RfbProto	소켓을 통해 전송받은 Server-to-Client 메시지를 처리하며 소켓을 통해 서버에게 Client-to-Server 메시지를 전송한다.
VncCanvas	전송받은 Server-to-Client 메시지를 바탕으로 화면을 표시하며 사용자가 발생시킨 이벤트를 감지하여 RfbProto 객체를 통해 Client-to-Server 메시지를 전송한다.
ClipboardFrame	원격시스템의 클립보드에 전송할 텍스트를 입력하거나 원격시스템으로부터 전달받은 클립보드 텍스트를 표시한다.

ServerCutText 메시지를 통해 전달된 원격시스템의 클립보드 텍스트는 ClipboardFrame 객체의 setCutText 메서드를 통해 java.awt.TextArea 컴포넌트에 표시된다. 사용자는 ClipboardFrame 객체의 Text Area에 표시된 텍스트를 복사 또는 잘라내기 명령을 통해 원격시스템의 클립보드 텍스트를 로컬시스템에서 사용할 수 있다.

3.2.2 로컬시스템에서 원격시스템으로 클립보드 텍스트 전송

로컬시스템의 클립보드 텍스트를 원격시스템으로 전송하고자 할 때에는 ClipboardFrame객체의 java.awt.TextArea 컴포넌트에 클립보드 텍스트를 붙여 넣어야 한다. ClipboardFrame 객체의 대화상자가 포커스를 잃게 되면, java.awt.TextArea 컴포넌트에 쓰여진 텍스트가 ClientCutText 메시지를 통하여 UltraVNC 서버로 전송된다. ClientCutText 메시지는 메시지를 전송한 클라이언트의 vncClient 객체에 전달된다. vncClient 객체는 vncServer 객체의 UpdateLocalClipText 함수를 실행하며, vncServer 객체의 UpdateLocalClipText 함수는 원격시스템의 클립보드에 텍스트를 저장하는 vncDesktop 객체의 SetClipText 함수를 실행한다. SetClipText 함수는 원격시스템의 클립보드에 전송받은 자료를 저장한다.

IV. 클립보드 이미지 콘텐츠 공유를 위한 RFB 프로토콜과 VNC의 확장

위 3장에서 보인바와 같이 RFB 프로토콜과 VNC는 원격시스템과 로컬시스템의 클립보드 텍스트 공유기능을 제공하고 있다. 본 장에서는 전송한 클립보드 텍스트 공유기법을 기반으로 클립보드 이미지를 전송할 수 있도록 RFB 프로토콜을 확장한다. 또한 확장한 RFB 프로토콜을 지원하여 클립보드 이미지를 공유할 수 있도록 UltraVNC 서버와 JavaViewer 클라이언트를 확장하여 VNC서버와 클라이언트를 개발한다.

4.1. 원격시스템에서 로컬시스템으로의 클립보드 이미지 전송을 위한 RFB 프로토콜의 확장과 VNC 서버, 클라이언트의 개발

원격시스템의 클립보드에 저장된 이미지를 서버에서 클라이언트로 전송하여 클립보드 이미지를 공유할 수 있도록 확장한다. 본 연구에서는 클립보드에 저장된 단일 이미지 형식과 이미지와 텍스트가 혼합된 형식을 나누어 처리하도록 한다. 이것은 클립보드에 저장된 콘텐츠는 특정 단일 형식으로 처리

되기 때문이다(10). 예를 들어 이미지와 텍스트가 섞여 있는 콘텐츠는 두 가지 형식으로 나누어 처리되지 않고 이미지와 텍스트가 섞인 형식으로 처리된다. 따라서 이미지와 텍스트가 혼합된 형식을 공유하기 위하여 대표적인 형식인 RTF(Rich Text Format)(11)를 사용한다. 또한 단일 이미지를 RTF 형식으로 전송하는 것도 가능하지만, 단일 이미지 형식을 지원하나 RTF 형식을 지원하지 못하는 응용프로그램을 사용할 경우 문제가 될 수 있다. 따라서 단일 이미지를 DIB(장치 독립적인 비트맵)(10)형식으로 전송하도록 하며, 텍스트와 이미지가 혼합된 콘텐츠는 RTF 형식으로 전송하도록 한다. 이와 같은 전송을 위하여 RFB 프로토콜에 다음 [표 7]에서 보이고 있는 메시지를 추가한다. 추가한 ServerCutImage 메시지와 ServerCutData 메시지의 구조는 ServerCutText의 구조와 동일하며 메시지 형식 필드는 각각 [표 7]에 기술한 Number 값을 사용한다. ServerCutImage 메시지는 윈도우 플랫폼에서 DDB(장치 의존적인 비트맵)형식으로 저장되어있는 클립보드 이미지를 DIB(장치 독립적인 비트맵)(10)형식으로 전송하기 위하여 사용되며, ServerCutData 메시지는 이미지와 텍스트가 혼합된 클립보드 콘텐츠를 RTF 형식으로 전송하기 위하여 사용된다.

표 7. Server-to-Client 메시지에 추가된 메시지
Table 7. New messages added to the Server-to-Client Messages

Number	Name	비고
4	ServerCutImage	서버 클립보드의 이미지 정보
5	ServerCutData	서버 클립보드의 텍스트와 이미지가 혼합된 정보

위와 같이 확장한 RFB 프로토콜 메시지를 사용하여 클라이언트로 클립보드 이미지를 전송하는 VNC 서버를 개발한다. UltraVNC 서버의 클립보드 텍스트를 전송하기 위해 사용되던 세 가지 클래스를 다음 [표 8]과 같이 확장하였으며, 확장한 클래스를 통하여 [그림 3]과 같은 과정을 거쳐 개발된 클라이언트에 클립보드 이미지를 전송할 수 있도록 한다.

표 8. 개발된 VNC 서버의 확장 내용
Table 8. New/extended methods in the developed VNC server

클래스	추가 또는 확장된 함수	
	이름	내용
vncDesktop	DesktopWindowProc	클립보드 갱신을 감지하였을 때, 콘텐츠의 형식을 분석하는 기능을 수행할 수 있도록 확장함
vncServer	UpdateClipboardImage	vncClient 객체를 통하여 클립보드 이미지를 전송하기 위해 추가된 함수
	UpdateClipboardRtf	vncClient 객체를 통하여 이미지와 텍스트가 섞인 클립보드 콘텐츠를 전송하기 위해 추가된 함수
vncClient	UpdateClipboardImage	클라이언트에 클립보드 이미지를 전송하기 위하여 추가한 함수
	UpdateClipboardRtf	클라이언트에 이미지와 텍스트가 혼합된 클립보드 콘텐츠를 전송하기 위하여 추가한 함수

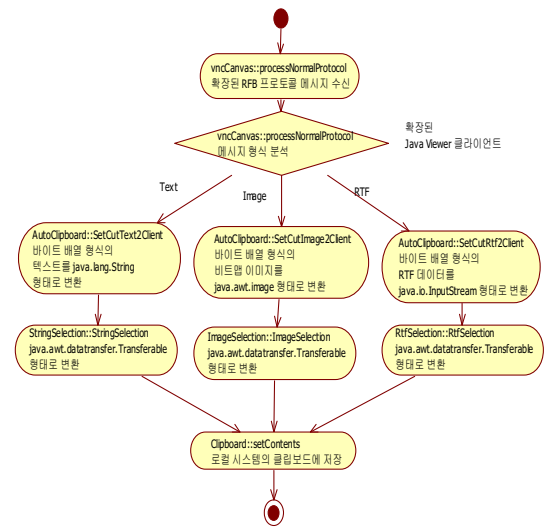


그림 4. 전송받은 클립보드 콘텐츠를 개발된 클라이언트가 로컬 시스템의 클립보드에 저장하는 과정

Fig. 4. Procedure for the developed client to store the received clipboard content to the local system clipboard

개발된 클라이언트는 서버로부터 전송받은 클립보드 콘텐츠를 java.awt.TextArea 컴포넌트를 거치지 않고 AutoClipboard 클래스를 통하여 로컬시스템에 즉시 저장하도록 한다. 그 결과 JavaViewer 사용자가 java.awt.TextArea 컴포넌트에 표시된 콘텐츠를 복사하는 번거로운 과정을 제거한다.

AutoClipboard 클래스에서 ServerCut 계열의 메시지가 담고 있는 클립보드 콘텐츠를 로컬시스템의 클립보드에 저장하기 위해 콘텐츠를 자바 클래스 객체를 통해 사용할 수 있도록 하여야 한다. 클립보드 콘텐츠를 자바에서 사용 가능한 클래스로 만드는 방법은 Transferable 인터페이스를 이용하여 클립보드 콘텐츠를 구현하는 방법[12,13]이다. 다음 [표 9]는 이미지 콘텐츠를 Transferable 인터페이스를 이용하여 클래스로 구현한 ImageSelection 클래스이다.

표 9. ImageSelection 클래스
Table 9. ImageSelection Class

```

import java.awt.datatransfer.*;
import java.io.IOException;
import java.awt.*;

public class ImageSelection implements Transferable {
    private Image image;
}
    
```

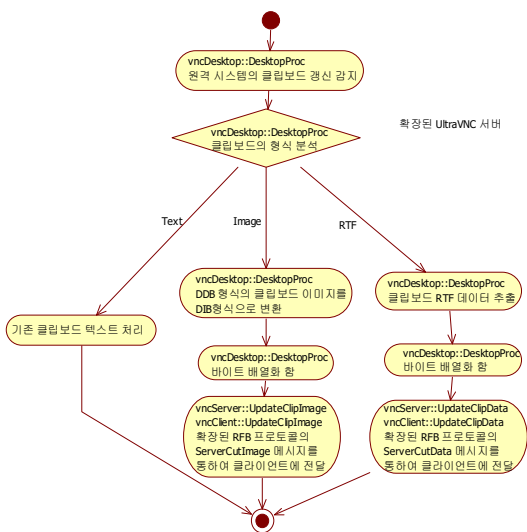


그림 3. 개발된 VNC 서버가 원격시스템의 클립보드 콘텐츠를 클라이언트로 전송하는 과정

Fig. 3. Procedure for the developed VNC to send the clipboard content of the remote system to clients

JavaViewer 클라이언트를 서버에서 전송된 원격시스템의 클립보드 이미지를 로컬시스템의 클립보드에 저장하도록 확장한다. 다음 [그림 4] 그 과정을 보이고 있다.

```

public ImageSelection(Image image) {
    this.image = image;
}

// Returns supported flavors
public DataFlavor[]
    getTransferDataFlavors() {
    return new DataFlavor[]{DataFlavor.imageFlavor};
}

// Returns true if flavor is supported
public boolean isDataFlavorSupported(DataFlavor flavor) {
    return DataFlavor.imageFlavor.equals(flavor);
}

// Returns image
public Object getTransferData(DataFlavor flavor) throws
    UnsupportedFlavorException, IOException {
    if (!DataFlavor.imageFlavor.equals(flavor)) {
        throw new UnsupportedFlavorException(flavor);
    }
    return image;
}
}
    
```

[그림 4]와 같은 과정을 통하여 이미지 콘텐츠를 로컬시스템의 클립보드에 저장하기 위하여 JavaViewer 클라이언트를 [표 10]과 같이 확장하였으며, [표 11]에서 보이는 클래스를 추가한다.

표 10. 개발된 클라이언트에 확장 또는 추가된 메서드
Table 10. New/extended methods in the developed client

클래스	추가 또는 확장된 메서드	
	이름	내용
VncCanvas	processNormalProtocol	전송받은 ServerCutImage 메시지와 ServerCutData 메시지를 인식할 수 있도록 확장
RfbProto	readServerCutImage	전송받은 원격시스템의 클립보드 이미지를 로컬시스템의 클립보드에 저장하기 위해 추가한 메서드
	readServerCutData	전송받은 원격시스템의 이미지와 텍스트가 혼합된 클립보드 콘텐츠를 로컬시스템의 클립보드에 저장하기 위해 추가한 메서드

표 11. 개발된 클라이언트의 추가된 클래스
Table 11. New classes in the developed client

추가된 클래스	역할
AutoClipboard	전달받은 클립보드 콘텐츠를 로컬시스템의 클립보드에 저장하는 기능을 한다.
ImageSelection	이미지 콘텐츠를 로컬시스템의 클립보드에 저장하기 위해 Transferable 인터페이스를 이용하여 구현한 클래스이다.
RtfSelection	로컬시스템의 클립보드에 저장할 이미지와 텍스트를 혼합한 콘텐츠를 Transferable 인터페이스를 이용하여 구현한 클래스이다.

다음 [그림 5]는 개발된 VNC 서버와 클라이언트를 사용하여 클립보드 이미지를 X-window 원격시스템에서 윈도우즈 로컬시스템으로 전송한 결과이다.

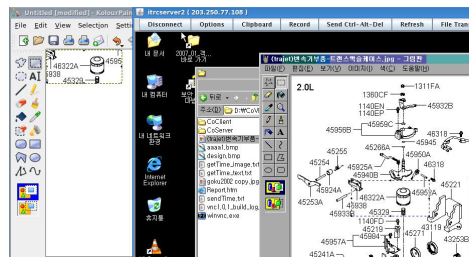


그림 5. 윈도우즈 원격시스템의 클립보드 이미지를 X-window 로컬시스템으로 전송한 예
Fig. 5. Example of sending the clipboard image in the remote system(windows) to the local system(Linux X-window)

4.2. 로컬시스템에서 원격시스템으로의 클립보드 이미지 전송을 위한 RFB 프로토콜의 확장과 VNC 서버, 클라이언트의 개발

개발된 클라이언트로부터 로컬시스템의 클립보드 이미지를 개발된 VNC 서버에 전송하도록 하여 개발된 VNC 서버가 원격시스템의 클립보드에 콘텐츠를 저장하도록 확장한다. 다음 [그림 6]은 개발된 클라이언트가 클립보드 이미지를 서버에 전송하는 과정이다.

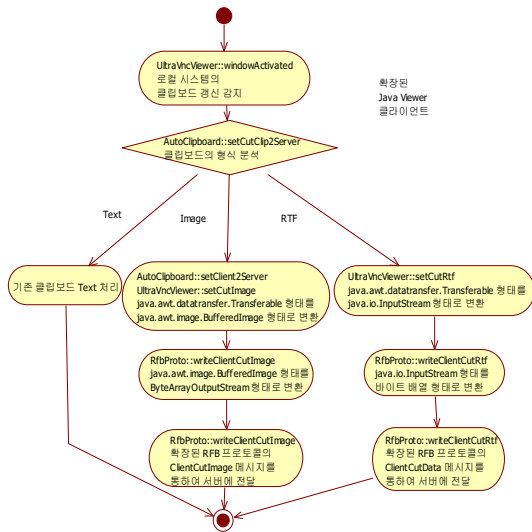


그림 6. 개발된 클라이언트가 클립보드 이미지를 전송하는 과정
Fig. 6. Procedure for the developed client to send the clipboard content of the local system

개발된 클라이언트가 위와 같은 과정을 거쳐 개발된 서버에 클립보드 이미지를 전송하기 위해서 [표 11]에서 보이고 있는 새로운 RFB 프로토콜 메시지를 추가한다. 각 메시지의 구조는 ClientCutText의 구조와 동일하며 메시지 형식 필드에서는 [표 12]에 기술한 값을 사용한다. 각 메시지는 이미지 포맷과 이미지와 텍스트가 혼합된 포맷을 전송하기 위해 정의되었다.

표 12. Client-to-Server 메시지에 추가된 메시지
Table 12. New messages added to the Client-to-Server Messages

Number	Name	비고
7	ClientCutImage	클라이언트 클립보드의 이미지 정보
8	ClientCutData	클라이언트 클립보드의 텍스트와 이미지가 혼합된 정보

기존 JaveViewer에서는 로컬시스템의 클립보드 텍스트를 서버에 전송하기 위해서 Text Area에 텍스트를 붙여 넣는 과정을 거쳐야 한다. 개발된 클라이언트는 클라이언트 윈도우가 활성화 될 때 로컬시스템의 클립보드의 콘텐츠가 갱신되었는지를 검사하도록 한다. 다른 응용프로그램의 이미지를 복사하기 위해서는 이미지를 담고 있는 응용프로그램의 윈도우를 활성화 시키게 되며 클라이언트 윈도우는 비활성화 된

다. 그 후 원격제어를 수행하기 위해 클라이언트 윈도우를 활성화 시켰을 때, 갱신된 클립보드의 콘텐츠를 얻어올 수 있도록 하여 별도의 붙이기 작업을 생략할 수 있도록 한다. 다음 [표 13]은 개발된 클라이언트의 내용이다.

표 13. 개발된 클라이언트의 확장 내용
Table 13. New/extended methods in the developed client

클래스	추가 또는 확장된 메서드	
	이름	내용
UltraVnc Viewer	windowActivated	JavaViewer 윈도우가 활성화 되었을 때 로컬시스템의 클립보드 갱신 여부를 확인하도록 확장
	setCutImage	setCutImage2Server에서 DIB형식으로 변환된 클립보드 이미지를 자바 소켓으로 전송하기 위하여 java.io.OutputStream 형식으로 변환하는 메서드
	setCutRtf	추가된 메서드로 클립보드 콘텐츠가 이미지와 텍스트가 혼합된 형식일 때 이것을 자바 소켓으로 전송하기 위하여 java.io.OutputStream 형식으로 변환하는 메서드
AutoClipboard	setCutClip2Server	갱신된 클립보드 콘텐츠의 형식을 구분하기 위해 추가된 메서드
	setCutImage2Server	클립보드 콘텐츠가 이미지일 때 java.awt.Image 형식의 클립보드 이미지를 DIB형식으로 변환하여 서버로 전송하기 위하여 추가한 메서드
RfbProto	writeClientCutImage	서버에 클립보드 이미지를 전송하기 위하여 추가한 메서드
	writeClientCutRtf	서버에 이미지와 텍스트가 혼합된 클립보드 콘텐츠를 전송하기 위하여 추가한 메서드

개발된 VNC 서버는 ClientCutImage와 ClientCutData 메시지를 전송받고 [그림 7]과 같은 과정을 거쳐 원격시스템의 클립보드에 콘텐츠를 저장한다. 또한 [표 14]는 개발된 VNC의 내용을 보이고 있다.

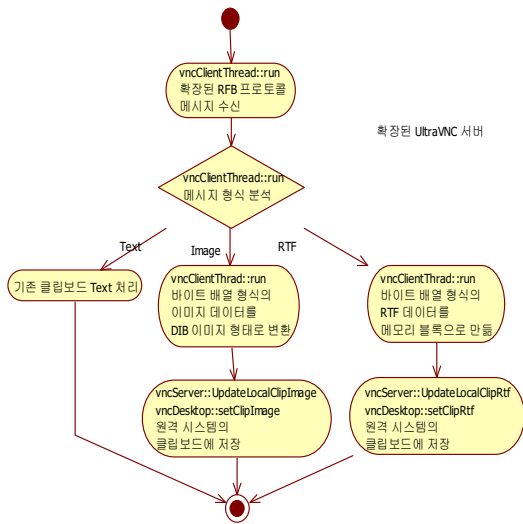


그림 7. 개발된 VNC 서버가 클립보드 콘텐츠를 원격시스템의 클립보드에 저장하는 과정

Fig. 7 Procedure for the developed VNC server to store the received clipboard content

표 14. 개발된 VNC 서버의 확장 내용

Table 14. New/extended methods in the developed VNC server

클래스	추가 또는 확장된 함수	
	이름	내용
vncClientThread	run	vncClient 객체와 연결된 클라이언트가 전송한 클립보드 콘텐츠의 형식을 구별할 수 있도록 확장하며 이미지의 경우 DIB 형식으로 만들며, 이미지와 텍스트가 혼합된 콘텐츠는 RTF형식으로 만들도록 확장
vncServer	UpdateLocalClipImage	vncDesktop 클래스의 setClipImage 함수를 호출하기 위하여 추가한 함수
	UpdateLocalClipRtf	vncDesktop 클래스의 setClipRtf 함수를 호출하기 위하여 추가한 함수
vncDesktop	setClipImage	전송받은 로컬시스템의 클립보드 이미지를 원격시스템의 클립보드에 저장하기 위하여 추가한 함수
	setClipRtf	전송받은 로컬시스템의 이미지와 텍스트가 혼합된 클립보드 콘텐츠를 원격시스템의 클립보드에 저장하기 위하여 추가한 함수

다음 [그림 8]은 개발된 VNC 서버와 클라이언트를 사용하여 클립보드 이미지를 X-window 로컬시스템에서 윈도우즈 원격시스템으로 전송한 결과이다.

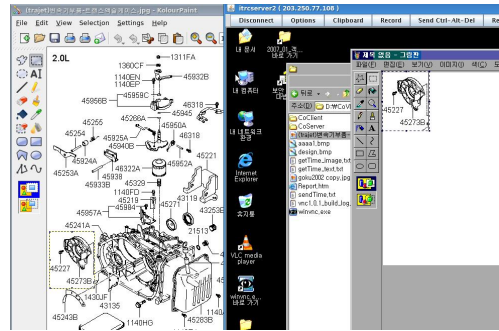


그림 8. X-window 로컬시스템에서 윈도우즈 원격시스템으로 클립보드 이미지를 전송한 예

Fig. 8. Example of sending the clipboard image in the local system(Linux X-window) to the remote system(windows)

V. 구현결과

클립보드 이미지를 공유할 수 있도록 개발된 VNC 서버와 클라이언트의 효율성을 검증한다. 다음 [표 15]는 오픈소스로 공개된 GUI 원격 제어 도구별로 가능한 클립보드 콘텐츠 공유를 플랫폼과 클립보드 콘텐츠 별로 보이고 있으며 본문에서 개발한 개발된 VNC만이 텍스트와 이미지가 혼합된 클립보드 공유를 지원하고 있다.

표 15. 원격 제어 도구 비교표

Table 15. Comparison table for remote control tools

분류	플랫폼		콘텐츠		
	원격시스템 (서버)	로컬시스템 (클라이언트)	텍스트	이미지	혼합
오픈소스 원격제어도구	윈도우즈	윈도우즈	○	×	×
	윈도우즈	리눅스	○	×	×
	리눅스	윈도우즈	○	×	×
	리눅스	리눅스	○	×	×

TightVNC	윈도우즈	윈도우즈	O	X	X
	윈도우즈	리눅스	O	X	X
	리눅스	윈도우즈	O	X	X
	리눅스	리눅스	O	X	X
UltraVNC	윈도우즈	윈도우즈	O	X	X
	윈도우즈	리눅스	O	X	X
	리눅스	윈도우즈	X	X	X
	리눅스	리눅스	X	X	X
개발된 VNC	윈도우즈	윈도우즈	O	O	O
	윈도우즈	리눅스	O	O	O
	리눅스	윈도우즈	X	X	X
	리눅스	리눅스	X	X	X

클립보드 공유를 통한 이미지 콘텐츠 공유의 효율을 측정한다. 원격시스템으로는 Dell사의 Power Edge 1800 서버 시스템을 사용하며 이 시스템의 제원은 3.4GHz의 성능의 두 개의 Intel Xeon 프로세서와 2GB의 주기억장치로 구성되어 있다. 원격시스템의 플랫폼은 마이크로소프트의 윈도우즈 2003 서버이다. 로컬시스템으로는 Intel 펜티엄4 2.0 프로세서와 512MB의 주기억장치로 구성된 시스템으로 CentOS 5.1 리눅스 플랫폼을 사용한다. 두 시스템은 초당 최고 4 메가바이트를 전송할 수 있는 NAT 환경으로 연결되어 있다. [표 16]는 개발된 VNC 서버와 클라이언트를 통한 클립보드 콘텐츠 별로 소요된 시간을 보인다.

표 16. 클립보드 콘텐츠 전송 소요 시간 (단위: ms)
Table 16. Time for transferring clipboard contents

콘텐츠 크기	원격시스템-> 로컬시스템			로컬시스템-> 원격시스템		
	텍스트	이미지	혼합	텍스트	이미지	혼합
300KB	210	508	705	156	375	422
500KB	378	711	1009	297	660	730
700KB	540	1012	1344	521	827	914
1MB	603	1292	1517	723	1022	1108

위와 같은 측정은 원천 시스템의 클립보드로부터 콘텐츠를 추출하고 전송하며, 목적지 시스템의 클립보드에 저장하는데 까지 소요되는 시간이다. 가장 크기가 큰 1MB의 데이터는 DMB(Digital Multimedia Broadcasting)[14] 장치의 스

털사진 한 장 크기를 저장할 수 있으며, 한글 40만자 가량을 저장할 수 있는 크기이다. 클립보드 이미지 공유 기능을 지원하도록 개발된 VNC서버와 클라이언트를 사용하여 1MB 가량의 데이터 전송을 2초를 초과하지 않는 시간 내에 수행할 수 있다. 동일한 환경에서 ftp를 이용하여 1MB 가량의 데이터를 전송했을 때, 보통 1초미만의 시간이 소요되므로 클립보드 콘텐츠 전송 소요시간의 결과와 비교하여 크게 차이가 나지 않는다. 따라서 클립보드 공유를 통하여 데이터 전송작업을 수행하는 것이 빠르고 효율적이다.

VI. 결론

VNC 프로그램은 상용 GUI 원격 제어 도구가 지원하지 않는 다양한 플랫폼간의 원격제어를 지원하므로, 인터넷 환경을 통하여 다양한 플랫폼의 원격시스템을 하나의 로컬 시스템에서 제어하는데 매우 유용하다. 즉 원격 접속한 원격 시스템을 로컬 시스템의 어플리케이션과 같이 다룰 수 있으며, 두 가지 이상의 원격 제어를 동시에 할 수 있는 원격 제어 멀티태스킹을 수행할 수도 있다. 효과적인 어플리케이션의 멀티태스킹을 위해서 어플리케이션간의 콘텐츠 교환이 필요한데, 콘텐츠 교환을 수행 하기위해 클립보드를 공유하는 방법을 사용하는 것이 간편하다.

현재 존재하는 문서들은 이미지와 텍스트 내용이 섞여 있는 형태가 대부분이다. 그러나 기존 RFB 프로토콜과 UltraVNC 서버, 클라이언트에서는 텍스트 형태의 클립보드 콘텐츠만 공유하는 것이 가능하였다. 본 논문에서는 텍스트 형태의 클립보드 콘텐츠와 더불어, 이미지를 포함하는 클립보드 콘텐츠 공유를 위하여 RFB 프로토콜 메시지를 확장하였으며, 확장한 RFB 프로토콜을 처리할 수 있도록 UltraVNC 와 JavaViewer를 확장한다. 그 결과 개발된 VNC 프로그램을 통하여 다양한 플랫폼 간에 텍스트와 이미지가 혼합된 내용도 용이하게 공유할 수 있게 되었다. 또한 본 논문에서 다루지 않은 wav 콘텐츠나 html 콘텐츠와 같은 멀티미디어 콘텐츠 역시 유사한 과정을 통하여 클립보드를 통해 공유하도록 확장될 수 있을 것으로 생각한다. 이와 더불어 전송량이 큰 비트맵과 RTF를 클립보드를 통해 공유하기 위해서 클립보드 콘텐츠를 압축하여 전송량을 줄이는 기술이 필요할 것으로 판단되며, 앞으로 이러한 부분을 지원하는 VNC 클립보드 공유 기법을 개발할 예정이다.

참고문헌

- [1] Tristan Richardson, Quentin Stafford-Fraser, Kenneth R. Wood and Andy Hopper "Virtual Network Computing", IEEE Internet Computing, Volume 2, Number 1, January/February 1998
- [2] "http://www.uvnc.com", UltraVNC
- [3] "http://www.realvnc.com", RealVNC
- [4] "http://www.tightvnc.org", TightVNC
- [5] "http://www.redstonesoftware.com/products/vine/server/vineosx/index.html", Vine Server
- [6] "http://www.symantec.com/norton/products/overview.jsp?pcid=pf&pvid=pca121", PcAnywhere
- [7] Omata M., Go K., Imamiya A.. "Augmented Reality Clipboard with the twist-information presentation method", Advanced Information Networking and Applications, 2003. AINA 2003. 17th International Conference, 27-29 March 2003 Page(s):392 - 399
- [8] Tristan Richardson, "The RFB Protocol", 2006.
- [9] "http://marketshare.hitslink.com/report.aspx?qprid=2", Market Share
- [10] Charles Petzold, "Programming Windows, 5th Edition", 2006.
- [11] "http://msdn.microsoft.com", MSDN
- [12] "http://java.sun.com/j2se/1.4.2/docs/api/", Java™ 2 Platform, Standard Edition, v 1.4.2 API Specification
- [13] 정명조, 성영락, 이철훈, "리눅스 시스템에서의 자바 클립보드 클래스 라이브러리 설계 및 구현", 한국정보과학회, 춘계학술대회, Vol. 30 No. 1, 2003년 4월.
- [14] 이철수, "DMB 서비스의 비디오 규격, H.264", 한국방송공학회, 방송공학회지 제9권 제4호, 2004. 12, pp. 26 ~ 35

저자 소개



이 태 호

2008년 2월 : 울산대학교 컴퓨터정보통신공학부 졸업
 2008년 3월 ~ 현재 : 울산대학교 컴퓨터정보통신공학부 석사과정
 관심분야 : 협업시스템, 원격 컴퓨팅, 홈네트워크, 분산프로그래밍



이 홍 창

2008년 2월 : 울산대학교 컴퓨터정보통신공학부 석사
 2008년 3월 ~ 현재 : 울산대학교 컴퓨터정보통신공학부 박사과정
 관심분야 : 웹기반 정보시스템, 협업시스템



박 양 수

1978년 : 울산대학교 전산학과 졸업 (공학사)
 1981년 : 서울대학교 계산통계학과 졸업 (이학석사)
 1986년 : 서울대학교 계산통계학과 박사과정 수료
 1980년 ~ 현재 : 울산대학교 정보통신공학부 교수
 관심분야 : 분산처리, 컴퓨터알고리즘 등



이 명 준

1980년 : 서울대학교 수학과 졸업(학사)
 1982년 : 한국과학기술원 전산학과 졸업(석사)
 1991년 : 한국과학기술원 전산학과 졸업(박사)
 1982년 ~ 현재 : 울산대학교 컴퓨터정보통신공학부 (교수)
 1993년 ~ 1994년 : 미국 버지니아 대학 교환교수
 2005년 ~ 2006년 : 미국 캘리포니아 주립대학 교환교수
 관심분야 : 웹기반 정보시스템, 프로그래밍언어, 분산프로그래밍, 생물정보학, 센서네트워크 프로그래밍 환경