

# 클립보드 이미지 전송을 위한 VNC 서버와 클라이언트의 확장

이태호<sup>0\*</sup>, 김정현<sup>\*</sup>, 이홍창<sup>\*</sup>, 이명준<sup>\*\*</sup>  
울산대학교 컴퓨터·정보통신공학부  
\*{soulfree<sup>0</sup>, mycpu, myhuynii}@mail.ulsan.ac.kr  
\*\* mjlee@ulsan.ac.kr

## Extending VNC Server and Client for Transmitting Clipboard Images

Taeho Lee<sup>0</sup>, Junghyun Kim, Hongchang Lee, Myungjoon Lee  
School of Computer Engineering & Information Technology, University of Ulsan

### 요 약

VNC(Virtual Network Computing)는 원격 제어 소프트웨어의 하나로서 다른 지역에 위치한 컴퓨터의 자원을 제어할 수 있는 기능을 제공한다. VNC는 원격 시스템과 로컬 시스템과의 자료 교환을 위하여 원격 시스템과 로컬 시스템간의 클립보드 내용을 공유하는 기능을 제공하고 있으나, 현재까지 개발된 VNC 소프트웨어들은 단지 문자 데이터를 공유하는 방법만을 제공하고 있고, 이미지와 같은 멀티미디어 데이터를 공유하는 방법을 제공하고 있지 않다.

본 논문에서는 클립보드에 저장된 이미지 데이터를 공유할 수 있도록 RFB 프로토콜을 확장하였다. 그리고 확장된 프로토콜을 지원하기 위하여 오픈소스 VNC 소프트웨어인 Ultra VNC 서버와 플랫폼 독립적인 클라이언트인 JavaViewer를 확장하였다. 사용자는 확장된 VNC를 통하여 원격 시스템과 로컬 시스템간의 클립보드 이미지 공유를 원활하게 수행할 수 있다.

### 1. 서론

Virtual Network Computing(이하 VNC)는 원격 제어 프로그램으로서 다른 지역에 위치한 시스템의 자원을 제어할 수 있는 기능을 제공한다. VNC는 Remote FrameBuffer(이하 RFB) 프로토콜을 사용하여 GUI(Graphic User Interface)를 통하여 원격 시스템의 자원을 효과적으로 제어할 수 있는 환경을 제공하며, 원격 시스템에 설치하는 서버와, 로컬 시스템에 설치하여 서버에 접속할 수 있도록 해주는 클라이언트로 구성된다. VNC와 비슷한 목적으로 사용되는 프로그램으로 Windows 플랫폼의 원격 제어 프로그램인 Remote Desktop이 있다. Remote Desktop은 로컬 시스템과 원격 시스템의 클립보드 내용을 서로 전송할 수 있는 기능을 제공하고 있다. 클립보드는 사용자가 임의로 선택한 데이터를 복사해두는 임시 기억공간을 말하며, 사용자는 클립보드에 복사한 데이터를 필요에 따라 불러내어 사용할 수 있다. Remote Desktop은 원격 시스템과 로컬 시스템 간의 클립보드 내용을 서로 전송하여 문자뿐만 아니라 이미지, Windows 오브젝트와 같은 다양한 형태의 데이터를 공유하여 사용할 수 있다. 현재 배포된 VNC

또한 원격 시스템과 로컬 시스템 간의 클립보드 공유 기능을 제공하고 있다. 하지만 RFB 프로토콜에서는 원격 시스템과 로컬 시스템의 클립보드에 저장된 내용 가운데, 유니코드 형식의 문자 데이터만을 전송할 수 있도록 정의 되어있다. 따라서 VNC가 제공하는 클립보드 공유 기능은 단지 문자 데이터를 공유 할 수 있고, 다양한 형태의 클립보드 데이터 공유 방법을 제공하고 있지 않다. 따라서 본 논문에서는 클립보드에 저장된 이미지 데이터 공유를 지원하는 VNC 대하여 기술한다. 확장된 VNC 시스템은 서버와 클라이언트로 구성되며, 서버는 Ultra VNC를 확장하였고, 클라이언트는 JavaViewer를 확장하였다. 또한 서버와 클라이언트 간의 클립보드 이미지 공유를 위하여 RFB 프로토콜을 확장하도록 한다. 사용자는 확장된 VNC 시스템을 사용하여 원격 시스템과 로컬 시스템간의 클립보드 이미지 공유를 원활하게 할 수 있다.

본 논문의 구성은 다음과 같다. 2절에서는 VNC의 소개, VNC 서버와 클라이언트의 구조와 동작과정, VNC가 사용하는 RFB 프로토콜에 대한 소개를 하고, 3절에서는 클립보드 이미지 전송을 위한 RFB 프로토콜 확장에 대한 설명을 한다. 4절에서는 클립보드 이미지 전송을 위한 VNC 프로그램의 확장에 대한 설명을 하고, 5절에서

\* 본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT연구센터 육성지원사업의 연구 결과로 수행되었음.

는 결론과 향후 연구 과제를 제시하고 있다.

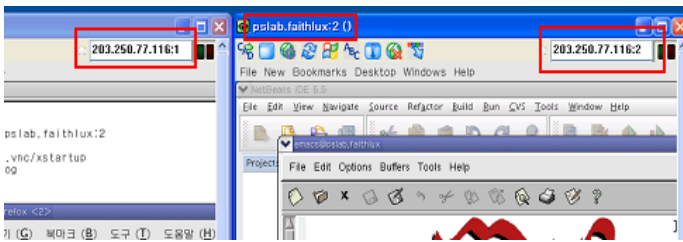
## 2. 관련연구

### 2.1 VNC 서버와 클라이언트 소개

VNC는 서론에서 언급한 바와 같이 서버와 클라이언트 프로그램으로 나누어져 있다.

서버는 원격 시스템의 화면 정보를 클라이언트에게 RFB 프로토콜 형태로 전달하는 역할을 하고 클라이언트로부터 전달받은 RFB 프로토콜 메시지의 내용을 원격 시스템에 전달한다.

클라이언트 프로그램은 서버로부터 전달받은 RFB 프로토콜 메시지의 내용을 로컬 사용자가 볼 수 있도록 표시하며 로컬 사용자의 입력을 RFB 프로토콜 메시지 형태로 서버에 전달하는 역할을 한다. 클라이언트 프로그램으로 원격 시스템에 접속하는 사용자는 서버에 정해진 포트(5900~5906번으로 사용자 임의로 바꿀 수 있음)를 사용하여 접속할 수 있는데, 멀티세션을 지원하는 원격 시스템에 설치된 서버에 접속할 경우 그림 1과 같이, 각 포트번호마다 서로 다른 화면으로 원격 시스템을 제어할 수 있다.



[그림 1] 포트번호로 구분되는 VNC클라이언트 세션

VNC 서버 클라이언트 프로그램으로 Real VNC와 Tight VNC, Ultra VNC 등이 있다. 본 논문에서는 원격 시스템의 클립보드 내용이 하나만 존재하는 Windows 플랫폼을 원격 시스템으로 사용하고자 하므로, Windows 사용자 계정 인증, 파일전송 등의 기능을 제공하는 Ultra VNC 서버와, Java Virtual Machine이 설치 가능한 어떠한 로컬 시스템에서도 사용가능한 JavaViewer 클라이언트 프로그램을 확장할 대상으로 하였다. 이 서버와 클라이언트는 RFB 프로토콜 3.6버전을 사용하고 있다.

## 2.2 Ultra VNC 서버와 JavaViewer클라이언트의 구조와 동작과정

### 2.2.1 Ultra VNC 서버

Ultra VNC 서버는 winvnc클래스에서부터 실행된다. winvnc 클래스는 vncdesktop 클래스의 객체를 생성한다. vncdesktop은 원격 시스템의 화면에서 일어나는 일

을 감지하는데 쓰이며 VNC 클라이언트가 전송한 명령을 원격시스템에 전달하도록 구현되어 있다. 또한 VNC 서버는 접속한 클라이언트에 해당하는 vncclient 클래스 객체를 만들고, 이러한 클라이언트 객체를 모을 수 있는 vncClientList 객체에 vncclient 객체를 추가하여, 접속되어 있는 모든 클라이언트를 관리하도록 구현되어 있다. 또한 각 vncclient 객체들은 하나씩 스레드를 실행하여 그 객체에 해당하는 클라이언트가 전송한 RFB 프로토콜 메시지를 전달 받는다. 그리고 vncclient 객체가 RFB 프로토콜 메시지를 전달 받았다면, vncclient 객체는 vncdesktop클래스 객체에 RFB 프로토콜 메시지를 전달하고, vncdesktop클래스 객체는 RFB 프로토콜의 헤더와 메시지의 내용을 분리하여 원격 시스템이 클라이언트의 명령을 수행하는 동작과정을 가진다.

### 2.2.2 JavaViewer 클라이언트

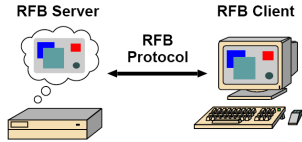
JavaViewer 프로그램의 실행 클래스인 VncViewer클래스를 실행하면 VNC 서버에 접속하는데 필요로 하는 암호를 입력하는 AuthPanel 클래스 객체가 실행된다. 클라이언트가 서버에 접속되었다면, 클라이언트 프로그램은 서버로부터 전달받은 RFB 프로토콜 메시지를 바탕으로 원격 시스템의 화면을 사용자에게 보여주는 VncCanvas 클래스 객체를 생성하며, VncCanvas 클래스 객체는 java.awt.Canvas 객체를 생성한다. 그리고 VncCanvas 클래스 객체는 서버에서 전달받은 RFB 프로토콜 메시지를 RfbProto 클래스 객체에게 전달하여 RFB 프로토콜 메시지에서 화면에 대한 정보를 얻어내고, 생성한 java.awt.Canvas 객체에 원격 시스템의 화면을 java.awt.Image 형식으로 그려서 표시하도록 구현되어 있다. 또한 java.awt.Canvas 객체에 마우스 이벤트 리스너와 키 이벤트 리스너가 추가되어있어 클라이언트 사용자가 VncCanvas에서 입력한 이벤트를 처리하도록 되어있으며, 이때에도 RfbProto 클래스 객체를 이용해 해당 이벤트를 RFB 프로토콜 메시지로 만든다. 따라서 Rfbproto 클래스는 서버로부터 전송받은 RFB 프로토콜 메시지를 해석하고 클라이언트에서 서버로 보낼 정보를 RFB 프로토콜 메시지로 만들어 전송하는 역할을 담당한다.

## 2.3 RFB 프로토콜에 대한 소개와 구조

### 2.3.1 RFB 프로토콜의 소개

RFB는 Remote FrameBuffer의 줄임말로 GUI 원격 시스템을 원격 제어를 하는데 사용되는 프로토콜을 지칭한다. RFB 프로토콜은 두 가지 endpoint를 가지는데, 하나

는 원격 시스템에 접속하는 사용자가 위치한 remote endpoint로 이를 RFB 클라이언트 또는 viewer라고 부르고, 변경된 framebuffer의 내용이 전송되는 endpoint는 RFB 서버라고 한다.



[그림 2] RFB 서버와 클라이언트

RFB 프로토콜은 TCP/IP 프로토콜을 바탕으로 설계되었기 때문에, RFB 서버와 클라이언트는 IP 주소를 이용해 서로의 위치를 파악하며 서로간의 신뢰성 있는 통신을 한다. RFB 서버와 클라이언트의 통신은 3가지 단계를 가진다. 첫 번째 handshaking 과정으로 서버와 클라이언트의 RFB 프로토콜 버전을 확인하고 security type을 확인하는 과정이다. 두 번째 Initialisation 과정으로 서버와 클라이언트가 서로 초기화 과정이 완료되었다는 메시지를 주고받는 과정이다. 마지막 세 번째 과정은 일반 프로토콜 메시지를 주고받는 과정이다.

### 2.3.2 RFB 프로토콜의 구조

모든 RFB 프로토콜 메시지는 메시지 유형(message-type)을 나타내는 헤더로 시작되며 메시지의 내용이 담긴 데이터(message-specific data)가 뒤 따른다. 2.3.1절에서 기술한 RFB 프로토콜 통신의 3가지 통신 단계에 따라, RFB 프로토콜의 유형을 크게 5가지로 나눌 수 있다. 첫 번째로 Handshaking Message와 두 번째 Security Types는 RFB 프로토콜 통신의 3가지 통신 단계의 첫 번째 과정에서 쓰이는 유형이다. 세 번째 Initialisation Message는 3가지 통신 단계의 두 번째 과정에서 쓰이는 유형이다. 네 번째 Server to client message로 서버가 원격 시스템의 화면을 클라이언트로 전송 하는데 사용되는 Display 프로토콜에 해당되며 다섯 번째인 Client to server Message는 클라이언트가 사용자의 입력을 서버로 전송 하는데 사용되는 Input 프로토콜에 해당한다. 이 네 번째와 다섯 번째 유형은 3가지 통신 단계 중 세 번째 단계에 쓰이는 유형이다.

## 3. 클립보드 이미지 전송을 위한 RFB 프로토콜 설계

### 3.1 클립보드 텍스트 전송을 위한 RFB 프로토콜 분석

RFB 프로토콜에는 서버가 원격 시스템의 클립보드에 저장된 텍스트를 클라이언트에 전송하는데 쓰이는 프로토콜인 ServerCutText와, 클라이언트가 로컬 시스템의 클립보드에 저장된 텍스트를 서버로 전송하는 데에 사용되

는 ClientCutText 프로토콜을 정의하고 있다. 표 1은 ServerCutText 프로토콜의 구조를 나타내고 있다. ClientCutText의 구조도 이와 같으며 message-type에 해당하는 value를 6으로 한다.

No. of bytes	Type	[Value]	Description
1	U8	3	message-type
3			padding
4	U32		length
length	U8 array		text

[표 1] ServerCutText 프로토콜의 구조

### 3.2 클립보드 이미지 전송을 위한 프로토콜 설계

RFB 프로토콜 메시지는 메시지 유형(message-type)을 나타내는 헤더로 시작되며 메시지의 내용이 담긴 데이터(message-specific data)가 뒤 따른다. 따라서 클립보드 이미지 전송을 위해서는 먼저 message-type에 사용할 value를 정해야 한다. 또한 이미지 전송을 위한 프로토콜은 2.3.1절에서 기술한 RFB 프로토콜 통신의 3가지 통신 단계의 세 번째 단계에서 사용되고, 서버에서 클라이언트로 전송하는 프로토콜과 클라이언트에서 서버로 전송하는 프로토콜을 각각 정의해야 한다. RFB Protocol Version 3.6버전에서는 클라이언트에서 서버로 전송할 때 사용하는 프로토콜 message-type의 경우, unsigned byte 형식으로 0~6, 255를 제외한 값을 사용할 수 있고 서버에서 클라이언트로 전송할 때 사용하는 프로토콜 message-type의 경우, unsigned byte 형식으로 0~3, 255를 제외한 값을 사용할 수 있다. 다음 표 2는 클립보드 이미지를 전송할 때 사용하는 프로토콜을 정의한 것이다. message-type의 경우, value를 100으로 정했으며, 서버에서 클라이언트 전송 할 때와 클라이언트에서 서버로 전송할 때 사용되는 구조는 동일하도록 정의하였다.

No. of bytes	Type	[Value]	Description
1	U8	6	message-type
3			padding
4	U32		length
length	U8 array		image

[표 2] 클립보드 이미지 전송을 위해 사용할 프로토콜의 구조

## 4. 클립보드 이미지 전송을 위한 VNC 확장

### 4.1 서버에서 클라이언트로 클립보드 이미지 전송

원격 시스템의 클립보드에 저장된 클립보드 이미지를

서버가 추출하여 그 것을 로컬 시스템의 클라이언트에게 전달하는 과정이다.

서버가 실행되는 플랫폼인 Windows의 클립보드 이미지는 DDB(Device Dependent Bitmap) 형식으로 존재하므로 다른 플랫폼의 클라이언트에게 전달하기 위해서 DIB(Device Independent Bitmap)으로 변환하는 과정이 필요하다. 또한 기존 RFB 프로토콜에 추가한 ServerCutImage 프로토콜은 이미지 데이터를 바이트 배열로 형식으로 전송하므로, DIB으로 변환한 이미지를 바이트 배열의 형식으로 만든다.

ServerCutImage 프로토콜을 전송받은 Java 클라이언트 프로그램은 바이트 배열 형식의 이미지 데이터를 java.image형태로 바꾸고, Java Virtual Machine 클립보드에서 사용가능한 java.awt.datatransfer.Transferable 형태로 만들어 클라이언트의 시스템이 가진 클립보드에 적용하도록 한다.

#### 4.2 클라이언트에서 서버로 클립보드 이미지 전송

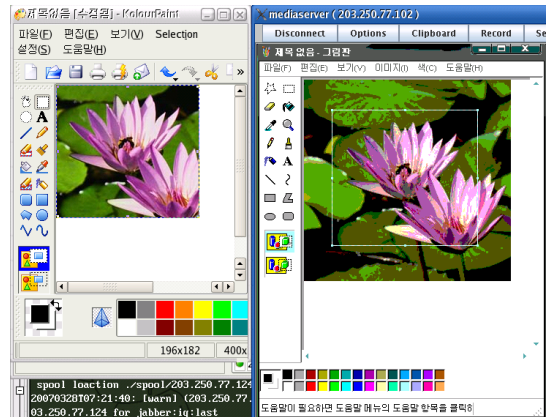
로컬 시스템의 클립보드에 저장된 이미지를 원격 시스템의 클립보드에 전달하는 과정이다.

Java 클라이언트는 클라이언트 시스템의 클립보드의 갱신을 감지하고, 클립보드의 내용이 이미지일 경우 java.awt.datatransfer.Transferable 형태로 이미지 내용을 얻는다. java.awt.datatransfer.Transferable형태의 이미지를 다시 java.awt.image.BufferedImage 형태로 변환하고 이것을 ClientCutImage 프로토콜을 사용해 서버로 전송하기 위해 ByteArrayOutputStream형태로 변환하여 서버로 전송한다. 이때 전송되는 이미지 형태는 DIB 형태이다.

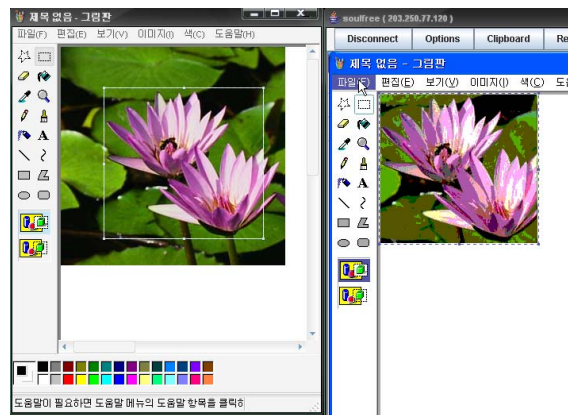
서버는 클라이언트로부터 전달받은 ClientCutImage 프로토콜에서 바이트 배열 형식으로 전달된 DIB 이미지를 원격 시스템의 클립보드에 적용한다.

#### 4.3 결과

Ultra VNC 서버와 클라이언트간의 클립보드 이미지 전송을 구현하여 Windows 플랫폼의 원격 시스템과 다양한 플랫폼의 로컬 시스템간의 클립보드 이미지 전송을 가능하도록 한다. 다음 그림 3은 Windows XP 플랫폼의 원격 시스템과 Windows XP 플랫폼의 로컬 시스템, 그림 4은 Windows XP 플랫폼의 원격 시스템과 Linux 플랫폼의 로컬 시스템간의 클립보드 이미지 공유를 보여주고 있다.



[그림 3] Linux <-> Windows XP 클립보드 이미지 전송



[그림 4] Windows XP <-> Windows XP 클립보드 이미지 전송

### 5. 결론 및 향후 연구

본 논문에서는 원격 시스템과 로컬 시스템의 클립보드 이미지 공유가 가능하도록 RFB 프로토콜을 확장하고, Ultra VNC서버와 JavaViewer 클라이언트를 확장하는 방법에 대해 기술하였다. 비록 원격 시스템의 클립보드 이미지는 Windows 플랫폼에서 사용하는 클립보드 이미지로 한정되어 있지만, 클립보드 이미지를 DIB로 만들고, 그것을 새롭게 정의한 RFB 프로토콜을 사용해 전달하여, 다양한 클립보드 데이터를 공유하는 방법을 마련할 수 있다. 다만 이러한 방식은 문자와 이미지가 섞인 클립보드 데이터를 공유할 수 없다. 그리고 Ultra VNC의 큰 장점인 전송 데이터의 암호화를 제공하고 있지 않다. 앞으로 이러한 부분을 지원하는 클립보드 공유를 연구할 것이다.

### 6. 참고 문헌

- [1] <http://www.wikipedia.org/>
- [2] Tristan Richardson, "The RFB Protocol", 2006
- [3] Charles Petzold, "Programming Windows, 5th Edition", 2006