CoVNC : 효과적인 협업을 위한 VNC의 확장

이태호°*, 박진호*, 이홍창*, 이명준**
울산대학교 컴퓨터·정보통신공학부
*{soulfree°, jinop, myhuynii}@mail.ulsan.ac.kr
** mjlee@ulsan.ac.kr

CoVNC : An Extension of VNC for Efficient Collaboration

Taeho Lee°, Jinho Park, Hongchang Lee, Myungjoon Lee
School of Computer Engineering & Information Technology, University of Ulsan

요 약

VNC 프로그램은 RFB(Remote Frame Buffer) 프로토콜을 사용하여 멀리 떨어진 시스템의 자원을 GUI(Graphic User Interface)를 통해 효과적으로 제어 할 수 있는 기능을 제공한다. 이를 이용하여 효과적인 동기식 협업 작업을 지원하는 시스템을 구성할 수 있다.

본 논문에서는 Ultra VNC 프로그램과 RFB 프로토콜을 확장하여 같은 시간에 일어나는 인터넷 기반 협업 작업을 효과적으로 지원할 수 있는 CoVNC의 개발에 대해 기술한다. CoVNC를 사용하는 협업 작업장 제공자나 협업 참여자가 협업 시스템의 구성과 실행에 대한 노력을 들이지 않으며 협업 작업에 몰두 할 수 있도록, 서버 설정을 저장하는 기능과 클라이언트가 한 번의 클릭으로 서버 접속을 할 수 있는 기능을 제공한다. 또한 작업장을 제어 할 수 있는 권한과 제어할 수 없는 권한, 그리고 클라이언트의 권한과 접속을 관리하는 관리자 권한을 제공하여, 협업 작업에서의 혼선과 악의적인 작업 방해를 막을 수 있다. 이러한 기능을 지원하기 위하여 Ultra VNC 서버를 확장하여 CoVNC 서버를 개발 하였으며, Java Viewer 클라이언트를 확장하여 CoVNC 클라이언트를 개발 하였고, RFB 프로토콜을 확장하여 서버와 클라이언트가 새로운 기능에 대한 메시지를 주고받을 수 있도록 하였다.

1. 서 론

서로 떨어진 위치에서 긴밀한 협업작업을 진행하기위해, 최근 발달한 인터넷 통신망을 이용하는 협업 지원 도구를 많이 사용하고 있다. CoSlide는 WebDAV 프로토콜을 사용하는 전문적인 협업 지원 프로그램 중의 하나로써 개인, 그룹, 공개 작업장을 지원하는 작업장 기반의 협업 프로그램이다. CoSilde는 협업 작업자가 공개 작업장에 작업한 내용을 업로드하고, 그 것을 자신이나 다른 협업 작업자가 인터넷에 접속할 수 있는 어느 곳에서나 다운로드를 받아서 작업을 진행할 수 있는 비동기식 협업 시스템이다.

VNC는 프로그램은 GUI(Graphic User Interface)를 통한 원격 제어 프로그램으로써, 다른 지역에 위치한 시스템의 자원을 효과적으로 제어할 수 있는 기능을 제공한다. VNC는 서버와 클라이언트로 구성되어있으며, 자원을 원격 제어할 시스템에 서버를 설치하고 클라이언트 프로그램으로 서버에 접속하여 원격제어를 수행할 수 있다. 또한 서버에 동시에 여러 클라이언트가 접속할 수 있어서, 같은 시간에 진행되는 협업 작업에 유용하게 사용할 수 있다. VNC와 유사한 프로그램으로 마이크로소프트의 원격 제어 프로그램인 Remote Desktop과 시만텍의 PCAnyware와 같은 프로그램이 있으나 상용 프로그램이거나 소스가 공개되어 있지 않다. 따라서 소스가 공개되어 있는 VNC 프로그램을 확장하면 다양한 플랫폼에서 효과적인 협업을 지원하도록 할 수 있다.

본 논문에서는 작업 공간을 기반으로 하는 비동기식 협업 시스템인 CoSlide와 같이 작업 공간을 기반으로 하되, 동기식 협업을 지원할 수 있도록 VNC 프로그램을 확장한 CoVNC에 대해 기술한다. CoVNC는 윈도우즈 플랫폼에서 동작하는 서버와 자바 기반의 클라이언트로 구성되어 있다. CoVNC는 서버가 작동하는 원격 시스템을 하나의 작업장으로 사용하도록 하며, 협업 참여자들이 클라이언트 프로그램을 통해 작업장의 자원을 제어하도록 한다. 그리고 CoSlide의 협업 참여자들로 하여금 CoVNC 작업장에 손쉽게 접속할 수 있도록, CoVNC 서버에 대한 정보가 저장된 파일을 이용하여 원클릭으로 접속할 수 있는 기능을 제공한다. 또한 기존의 VNC 프로그램은 인증된 모든 클라이언트에게 제어권한을 허용하고 있으며, 클라이언트가 자신의 제어권한을 자율적으로 관리하게 되어있다. 그렇지만 효율적인 동기적 협업 작업을 위해서는 작업 진행 중의 협업 참여자간의 혼선이 일어나지 않도록 제어권한에 대한 엄격한 관리가 필요하다. 따라서 CoVNC는 제어가 가능한 권한과 제어가 불가능한 권한을 나누었다. 그리고 협업 참여자의 권한

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

을 관리할 수 있는 작업 관리자 권한을 가지고 있으며, 작업 관리자가 협업 참여자들의 제어권한을 변경 할 수 있도록 하는 기능을 제공하고 있다. 그리고 이러한 권한을 인증 받을 수 있도록 암호를 이용한 인증과정을 제공하고 있다.

본 논문의 구성은 다음과 같다. 2장에서는 VNC서버와 클라이언트에 대한 소개와 클라이언트 인증 과정을 설명한다. 3장에서는 동기식 협업을 지원하기위한 CoVNC 서버와 클라이언트의 설계에 대해 기술하고, 4장에서는 그 구현을 기술한다. 끝으로 5장에서는 CoVNC에 대한 결론과 향후 보완과제를 제시하고 있다.

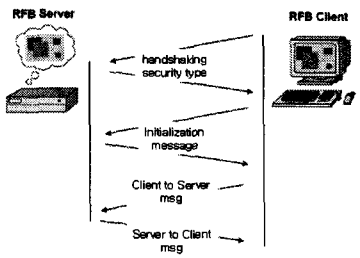
2.관련연구

2.1 VNC 서버와 클라이언트

VNC는 서버와 클라이언트 프로그램으로 나누어진다. 현재 배포되고 있는 VNC 서버 프로그램으로는 Real VNC 서버, Tight VNC 서버가 있으며 윈도우즈 플랫폼에서만 사용가능한 Ultra VNC 서버 프로그램이 있다. 많은 사람들이 작업도구로 사용하는 윈도우즈 플랫폼을 협업 작업장으로 하기 위해, CoVNC 서버는 Ultra VNC 서버를 확장하였다.

VNC 클라이언트는 보통 VNC 서버 프로그램과 같이 배포되고 있다. VNC 클라이언트 프로그램으로는 Real VNC 서버, Tight VNC 서버, Ultra VNC 서버와 같이 배포되고 있으며, 서로 다른 버전의 서버에 접속하는 것도 가능하다. CoVNC 클라이언트는 Java Viewer 클라이언트를 확장하였다. Java Viewer는 Java Runtime Edition을 사용할 수 있는 모든 환경에서 실행 가능하다.

VNC 서버와 클라이언트는 RFB 프로토콜을 사용하여 서로 통신한다. RFB 프로토콜은 Remote Frame Buffer의 줄임말로써 GUI를 사용해 멀리 떨어진 원격 시스템을 제어하는데 사용하는 프로토콜을 말한다. RFB 프로토콜을 사용한 원격 제어는 [그림1]과 같은 과정을 거친다.



[그림 3] RFB 프로토콜 통신에 사용되는 메시지 유형

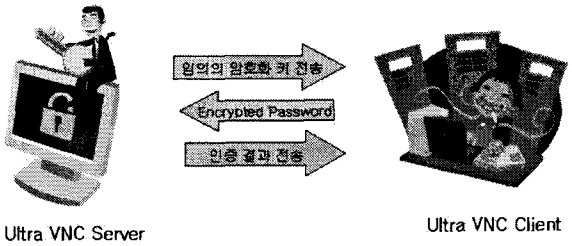
2.2 Ultra VNC 서버의 설정 분석

Ultra VNC 서버에서는 인증 암호, 접속 포트, 접속할 Display 세션의 수, 클라이언트의 입력을 무시하는 등의 기본적인 설정을 포함하여, 서버의 바탕화면에 설정된 배경그림을 배제하거나, 파일 전송 기능 가능하게 하거나 하는 등의 다양한 설정을 할 수 있다. 이러한 서버 설정은 윈도우즈의 레지스트리에 저장하고 불러온다.

Ultra VNC 서버는 실행 시 윈도우즈의 작업표시줄에 Tray Icon 형태로 실행되는데, Ultra VNC 서버의 설정을 변경하거나 서버를 종료하는 등의 명령을 내리기 위해서 Tray Icon을 클릭하므로 써 작업을 수행할 수 있다. 이 때 서버의 설정을 입력하는 윈도우가 실행될 때 Ultra VNC 서버의 설정 값을 저장하고 있는 레지스트리의 값을 불러오며, 서버 설정을 완료하고 설정 윈도우 상의 'OK'나 'Apply' 버튼을 클릭하면, 레지스트리에 설정 값을 저장하게 된다.

2.3 Ultra VNC의 클라이언트 인증 과정

Ultra VNC 프로그램은 RFB 프로토콜을 사용한다. Ultra VNC의 인증과정은 RFB 프로토콜을 사용한 통신 과정에서 첫 번째에 해당하는 'handshaking, security type' 과정에서 수행된다. 서버에 새로운 클라이언트가 접속하면 서버는 클라이언트의 상태를 확인하고, 인증에 사용되는 암호를 암호화하는데 사용할 키를 무작위로 생성한다. 그리고 클라이언트에게도 그 키를 전송한다. 그리고 클라이언트는 전송받은 암호화 키를 사용해 입력한 암호를 암호화하고 서버에게 전송하면, 서버는 클라이언트의 인증 암호를 비교하여 인증결과를 클라이언트에게 전송한다. 그 과정을 그림으로 나타내면 다음 [그림2]와 같다.



[그림 4] Ultra VNC의 인증과정

3. CoVNC를 이용한 동기식 협업 시스템의 설계

3.1 원클릭 CoVNC 접속환경

CoVNC 서버는 윈도우즈 레지스트리에 저장된 서버 환경을 XML 문서로 저장하고, 저장한 XML 문서를 불러오는 기능을 제공한다. CoVNC 서버 설정 윈도우에 저장과 불러오기 명령을 내릴 수 있는 'SAVE' 와 'LOAD' 버튼이 있다. CoVNC 서버의 설정을 저장할 때는 서버의 인

터넷 상의 IP를 확인하고, 설정을 입력하는 윈도우의 컴포넌트의 설정 상태를 확인하며, 그것을 XML Parser를 사용해 XML 문서 형태로 저장하도록 한다. 이때 CoVNC 서버에 접속하기 위해 필요한 암호는 제외한다. 그리고 저장된 XML 문서를 이용해 서버 설정을 불러올 때에도 XML Parser를 사용하여 파싱하여 CoVNC 서버 설정 윈도우의 컴포넌트를 설정하도록 한다.

CoVNC 클라이언트는 서버 설정 내용이 저장된 XML 문서를 다운로드 받아서 그 문서를 통해 원클릭 실행을 하도록 한다. CoVNC 클라이언트는 프로그램을 실행 서버 설정 XML 파일을 파싱한다. 파싱 과정을 통해 얻은 CoVNC 서버의 IP와 포트를 이용하여, IP와 포트 번호를 입력하는 과정을 거치지 않아도 CoVNC 서버에 접속할 수 있다.

3.2 CoVNC 사용자 권한 확장

CoVNC는 [표 1]과 같이 세 가지 권한을 가진다.

세부사항 / 권한명	암호 및 서버 설정 변경	접속한 클라이언트 관리	원격 시스템 자원 제어	원격 시스템 모니터링
Admin	○	○	○	○
Full Access	X	X	○	○
View Only	X	X	X	○

[표 1] 제어 권한

Full Access와 View Only 권한은 원격 시스템의 자원을 제어할 수 있는 여부에 따라 구분한 것이며, Admin 권한은 Full Access와 View Only 권한을 변경하거나 접속 중인 특정 클라이언트를 접속 해제 할 수 있는 관리자 권한이다. 각 클라이언트는 인증위해 입력한 암호에 의해 권한을 부여 받는다. 인증할 수 있는 암호는 각 권한 당 하나씩 존재하며, 윈도우즈 레지스트리에 암호를 저장하게 되는데 VNC Encrypt 매크로를 이용해 암호화 된 값이 저장된다.

Full Access 권한은 원격 시스템 자원을 제어하는 것이 가능한 권한이며, View Only 권한은 제어가 불가능하다. 이때 제어가 가능한 Full Access 권한 사용자가 CoVNC 서버 설정을 변경하거나, 종료시킬 수 없도록 하기위해, 각 과정을 수행하기 전에 Admin 권한을 인증하는데 사용한 암호를 묻는 과정이 존재한다.

Admin 권한의 클라이언트가 접속한 Full Access와 View Only 권한의 클라이언트를 관리하기 위해서, 접속한 클라이언트 목록을 보여주는 윈도우를 클라이언트 프로그램에 추가한다. 그리고 클라이언트 관리를 위해 RFB 프로토콜을 확장한다. 각 메시지를 통해 전송되는 데이터는 XML 형태이며, 확장한 내용은 다음과 같다.

1. CoVNC 서버에서 Admin 권한 클라이언트에게 현재 접속 중인 클라이언트 목록을 전송할 때 사용할 RFB 프로토콜 메시지를 추가한다. CoVNC 서버는 각 접속한 클라이언트가 인증 과정을 거쳐 접속하거나, 인증된 클라이언트가 접속을 해제 했을 때 이 메시지를 Admin 클라이언트에게 전송한다.
2. Admin 권한 클라이언트가 CoVNC 서버에 클라이언트 권한을 변경하거나 접속을 해제시키는 명령을 전송할 RFB 프로토콜 메시지를 추가한다. Admin 권한 클라이언트는 클라이언트 목록에서 선택한 클라이언트의 IP와 변경할 권한을 저장한 메시지를 CoVNC 서버에 전송한다. CoVNC 서버는 요청에 따른 처리를 하고 새로운 클라이언트 목록을 Admin 클라이언트에게 전송한다.

4.구현

4.1 원클릭 CoVNC 접속

CoVNC 서버의 설정을 입력받고 그것을 윈도우즈 레지스트리에 저장하며, 불러오는 것은 VNCPropertise클래스에서 이루어진다. VNCPropertise클래스에 서버 설정을 저장할 때 사용할 'SAVE' 버튼과 서버 설정을 불러올 때 사용할 'LOAD' 버튼을 추가하고 DialogProc() 함수에 각 버튼를 클릭했을 때의 이벤트를 정의한다. 'SAVE'버튼을 클릭했을 때, 파일 다이얼로그에서 XML문서를 이름을 지정하도록 하고, VNCPropertise클래스에서 각 컴포넌트의 설정된 값을 읽어 들인다. 그리고 각 컴포넌트의 ID를 Element 이름으로 정하고 읽어 들인 값을 요소의 Text 요소로 하여 XML 형태로 저장한다. 단, VNC서버에 접속에 필요한 암호는 저장하지 않는다. 또한 VSocket 클래스의 GetAddress() 함수를 사용해 서버의 인터넷 상의 IP를 얻어서 저장 파일에 저장한다.

'LOAD'버튼을 클릭했을 때, 파일 다이얼로그에서 XML 문서를 선택하도록 하고, 선택한 XML문서를 MSXML parser를 사용하여 파싱한다. MSXML parser의 DOM API를 사용해 XML문서를 분석하여 트리구조를 만들고, 각 Element 이름에 해당하는 VNCPropertise클래스 컴포넌트 ID를 찾는다. 그리고 Win32 API의 SendDlgItemMessage()함수를 사용하여 Element의 Text 요소에 저장된 값을 컴포넌트에 설정한다. 다음 [표 2]는 저장한 XML 문서를 보여주고 있다.

```
<?xml version="1.0" encoding="euc-kr" ?>
- <UltraVNC_preset>
  <server_ip>203.250.77.117</server_ip>
- <IDC_CONNECT_BORDER>
  <IDC_CONNECT_SOCK>1</IDC_CONNECT_SOCK>
  <IDC_SPECDISPLAY>0</IDC_SPECDISPLAY>
  <IDC_DISPLAYNO />
  <IDC_SPECPORT>1</IDC_SPECPORT>
  <IDC_PORTRFB>5901</IDC_PORTRFB>
  ...
```

[표 2] CoVNC서버 설정 XML 문서

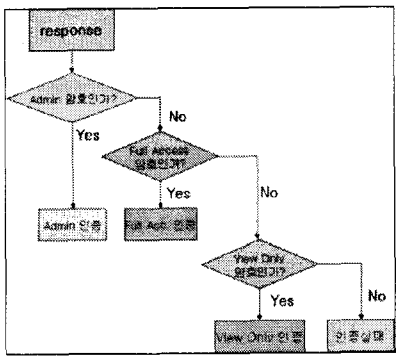
CoVNC 클라이언트를 실행 시 서버 IP와 포트 번호를 매개변수로 입력하여 실행할 수 있으며, 입력하지 않을 때 예외를 발생한다. 이때 예외 처리부분에 서버 설정 파일을 불러올 수 있는 javax.swing.JFileChooser를 실행하도록 한다. 불러온 XML 문서에서 서버의 IP, 포트 정보는 가장 상단에 위치해 있으며, XML 문서 구조를 특별히 분석해야 할 필요는 없으므로, JAXP의 SAX를 사용해 파싱한다. 파싱하여 얻은 IP와 포트를 main()메서드의 매개변수에 적용하여 클라이언트가 해당 CoVNC서버에 접속하도록 한다.

윈도우즈 플랫폼에서 XML 문서를 더블클릭하여 CoVNC 클라이언트가 자동으로 실행되도록 하기위해, 자바로 작성된 CoVNC 클라이언트 프로그램을 JBuilder의 'Native Executable Builder' 기능을 이용해 윈도우즈 Native 응용프로그램으로 만든다. 또한 서버 설정 파일 확장자의 연결프로그램을 Native 응용프로그램으로 설정한다.

4.2 CoVNC 사용자 권한

CoVNC 서버 설정을 담당하는 vncproperties 클래스에 Full Access의 암호와 View Only의 암호를 입력 받을 수 있는 Edit Box를 추가하고 두 암호를 저장할 맴버변수를 추가한다. 기존 VNC 접속을 위해 사용하는 암호는 Admin 권한으로 인증하기 위한 암호로 사용한다. 'OK' 혹은 'Apply' 버튼을 클릭하여 서버 설정을 완료하면 세 가지 암호가 서로 중복되지 않았는지 검사한다. vncproperties 클래스의 Edit Box에 입력된, 적용하기 전의 암호와 현재 서버에서 사용하고 있는 암호가 다를 수 있다. 따라서 서버 설정 윈도우의 암호입력 컴포넌트에 입력된 암호를 서로 검사하고, vncserver 클래스의 맴버 변수에 저장되어 적용되고 있는 암호를 서로 검사한다. 암호가 서로 중복되지 않으면 윈도우즈 레지스트리에 저장하며, 각 암호를 저장하여 실제 인증에 사용하는 vncserver 클래스의 맴버변수 m_passwd, m_full_acc_passwd, m_view_only_passwd에 적용한다.

CoVNC 클라이언트가 인증을 하기 위해 서버로 부터 전송받은 암호화 키를 사용해 암호화한 인증 암호를 서버로 전송 한다. CoVNC 서버에서 접속한 클라이언트로부터 메시지를 전송 받고, 처리결과를 클라이언트로 전송하는 것은 해당 vncclient 클래스 객체를 통해 이루어진다. 따라서 vncclient 객체는 vncserver 클래스의 맴버변수에 저장된 인증에 필요한 세 가지 암호를 클라이언트가 전송한 암호와 비교한다. 비교하는 순서는 다음 [그림 4]와 같다.



[그림 3] 클라이언트의 인증암호를 비교하는 순서

인증 실패가 발생하지 않았을 경우, vncserver 클래스는 vncclient 객체에서 권한 값을 저장하는 m_authmode 맴버 변수에 해당 권한을 적용하고, 해당 클라이언트에게 해당 권한 인증에 성공했다는 정보를 해당 클라이언트에게 전송하게 된다. 인증 결과 메시지를 RFB 프로토콜을 이용해 전송하기 위해 "handshaking, security type" 과정의 Authentication 부분의 AuthResponse 메시지를 확장하였다. 확장한 내용은 다음 [표 3]과 같다.

define	value	비고
VncAuthOk	0	Admin 인증 (기존 인증 성공)
VncAuthFailed	1	인증 실패
VncAuthTooMany	2	인증 실패가 너무 많이 일어남
VncAuthFullAcc	3	Full Access 인증(추가)
VncAuthViewOnly	4	View Only 인증(추가)

[표 3] AuthResponse 메시지의 확장 정의

기존 VNC 클라이언트에서는 입력을 서버에게 전달하지 않도록 하는 기능이 있다. 그러나 CoVNC는 View Only 권한을 구현하기 위해, 클라이언트 프로그램이 입력을 전송하지 않도록 설정하는 것이 아니라, 서버가 클라이언트가 전송한 입력을 무시하도록 하였다. 클라이언트의 포인트 입력(마우스 입력)과 키보드 입력을 무시하거나 허용하는 것을 설정하기 위해 해당 vncclient 클래스 객체의 맴버 함수인 EnablePointer(), EnableKeyboard()의 매개변수에 false를 전달함으로써 해당 클라이언트의 입력을 무시하도록 할 수 있다.

Full Access 권한의 사용자가 임의로 서버의 설정을 변경하거나 서버를 종료시킬 수 없도록 vncproperties에서 설정 윈도우를 활성화 하는 ShowAdmin() 함수의 첫 부분과 WndProc() 함수에 서버를 종료하는 이벤트를 처리하는 부분에 Admin 암호를 묻는 다이얼로그를 실행한다. 또한 Admin 권한의 클라이언트가 서버 설정을 변경할 때 Full Access 권한의 클라이언트가 작업장을 제어하지 못하도록 Full Access 권한의 클라이언트의 EnablePointer(), EnableKeyboard() 의 매개변수에 false를 전달한다. Admin 권한의 클라이언트가 설정 작업을 완료하거나 취소했을 때 다시 Full Access 권한의 클라이언트의 EnablePointer(), EnableKeyboard() 의 매개변수에 true를 전달한다.

CoVNC 서버가 Admin 권한의 클라이언트에게 클라이언트 목록을 전송하기위해서 RFB 프로토콜에 ClientListMsg 메시지를 Server To Client Message에 추가 구현한다. ClientListMsg 메시지의 구조는 다음 [표 4]와 같다.

No. of Bytes	Type [value]	Description
1	U8 [101]	message type
1		padding
4	U32	length
length	U8 array	XML

[표 4] ClientListMsg 메시지 구조

클라이언트 목록을 전송하는 메시지임을 알리기 위해 101이라는 새로운 값을 정의하였다. 그리고 1바이트의 여유 공간을 둔 후, 클라이언트 목록의 전체 길이를 4바이트 크기로 전송하고, 그 길이만큼의 XML 내용을 바이트 배열에 담아 전송하도록 하였다. 클라이언트의 목록은 XML 형식은 Type과 IP라는 Attribute에, 접속한 클라이언트의 정보를 저장하여 바이트 배열의 형식으로 전송한다. Type에는 클라이언트의 권한을 설정하며 IP에는 클라이언트의 인터넷 상의 IP 주소를 설정한다. 클라이언트에 대한 정보는 vncclient 클래스에 추가한 m_authmode와 m_client_name 맴버 변수를 참조함으로써 얻을 수 있다. 다음 [그림 4]는 클라이언트 목록을 담은 XML 문서의 예이다.

```
<?xml version="1.0" encoding="euc-kr"?>
<ClientList>
  <Client Type="Admin" IP="203.250.77.129"/>
  <Client Type="ViewOnly" IP="203.250.77.112"/>
  <Client Type="ViewOnly" IP="203.250.77.102"/>
</ClientList>
```

[그림 4] XML으로 만들어진 클라이언트 목록의 예

클라이언트 목록을 전송받은 Admin 클라이언트는 XML 문서를 JAXP의 SAX API를 사용하여 파싱하며, 이러한 작업을 위해 ClientListXmlParser 클래스를 새로 정의하였다. 파싱하여 얻은 클라이언트의 정보를 javax.swing.JTable에 표시하도록 함으로써 Admin 클라이언트가 클라이언트 목록을 확인할 수 있다.

Admin 권한으로 동작하는 클라이언트 프로그램은 특정 클라이언트의 권한을 바꾸거나 접속을 해제시키기 위해 명령을 입력받는 버튼이 있다. 각 버튼의 이벤트가 발행하면, JTable에서 사용자가 선택한 클라이언트 정보를 읽어 들인다. UserControlFrame 클래스는 그런 역할을 담당하며 변경할 권한과 선택한 IP 정보를 이용해 XML 문서를 만든다. 다음 [그림 5]는 Full Access 권한의 클라이언트를 View Only 권한으로 변경하는 정보를 담은 XML 메시지의 예이다.

```
<?xml version="1.0" encoding="UTF-8"?>
<ClientList>
   <Client Type="ViewOnly" IP="203.250.77.118"/>
</ClientList>
```

[그림 5] 클라이언트 권한 변경 정보를 담은 XML 메시지의 예

클라이언트 권한 변경 메시지를 CoVNC 서버로 전달하기 위해 RFB 프로토콜의 Client To Server Message를 확장한다. 추가한 메시지는 ClientAdminRequestMsg이며 그 구조는 다음 [표 5]와 같다.

No. of Bytes	Type [value]	Description
1	U8 [150]	message type
3		padding
4	U32	length
length	U8 array	XML

[표 5] ClientAdminRequestMsg 메시지 구조

Admin 권한의 클라이언트로부터 rfbAdminRequest 메시지를 전송받은 서버는 전송된 XML을 MSXML Parser를 이용하여 파싱한다. 파싱하여 얻어낸 권한을 변경할 클라이언트의 IP를 이용하여 해당 클라이언트의 vncclient 객체를 찾아내고 m_authmode 맴버 변수의 값을 변경하고, EnablePointer() , EnableKeyboard() 맴버 함수를 이용해 입력을 설정한다. 클라이언트의 접속을 끊는 요청에서는 vncclient 클래스의 Kill() 맴버 함수를 사용한다.

CoVNC 서버는 모든 사용자 권한 변경 작업이 끝나면 새로운 클라이언트 목록을 만든다. 그리고 ClientListMsg 메시지를 이용해 Admin 클라이언트에게 전달한다.

5.결론

CoVNC는 기존의 비동기식 협업 시스템의 CoSlide 사용자, 또는 CoSlide를 사용하지 않는 사용자가 동기적인 협업에 편리하게 참여 할 수 있도록 한다. 따라서 협업 작업장 제공자가 CoVNC 서버를 이용해 편리하게 작업장을 제공할 수 있도록 하며, CoVNC 클라이언트를 이용하는 협업 참여자가 손쉽게 작업장에 접근할 수 있도록 원클릭 접속 기능을 제공하고 있다. 또한 효율적인 협업 작업을 위해 작업관리자를 두어, 작업을 진행할 협업 참여자와 그렇지 않은 협업 작업자를 관리하도록 하였다. 따라서 CoVNC를 이용하면, 작업자 간의 명확한 책임 구분과 혼선 없는 효율적인 동기식 협업 작업이 가능하다.

CoVNC 서버는 세계적으로도 많은 사람들이 직무에 사용하고 있는 윈도우즈 플랫폼에서만 작동하며, CoVNC를 사용한 협업 작업도 윈도우즈 플랫폼 작업장에서만 이루어진다. 또한 기존 CoSlide 클라이언트 프로그램을 통해서 CoVNC 클라이언트를 실행하도록 하는 기능은 제공하고 있지 않다. 앞으로 다양한 작업장 플랫폼을 동기식 협업에 사용할 수 있도록 다양한 플랫폼에서 작동하는 CoVNC 서버의 개발이 필요할 것이며, CoSlide와 유기적인 연동 할 수 있도록 보완하여야 할 것이다.

6.참고 문헌

[1] "http://www.wikipedia.org" Wikipedia.
[2] "http://www.uvnc.com", UltraVNC.
[3] Tristan Richardson, "The RFB Protocol", 2006.
[4] Charles Petzold, "Programming Windows, 5th Edition", 2006.
[5] "http://msdn.microsoft.com", MSDN.
[6] "http://java.sun.com/j2se/1.4.2/docs/api/" JavaTM 2 Platform, Standard Edition, v 1.4.2 API Specification
[7] 이태호, 김정현, 이홍창, 이명준, "클립보드 이미지 전송을 위한 VNC 서버와 클라이언트의 확장", 한국정보과학회, 춘계학술대회, Vol. 34 No. 1, 2007년.
[8] 김동호, 박진호, 신원준, 이명준, "웹데브 기반의 효과적인 협업 작업 지원", 한국정보과학회, 추계학술대회, Vol.33 No. 2, 2006년