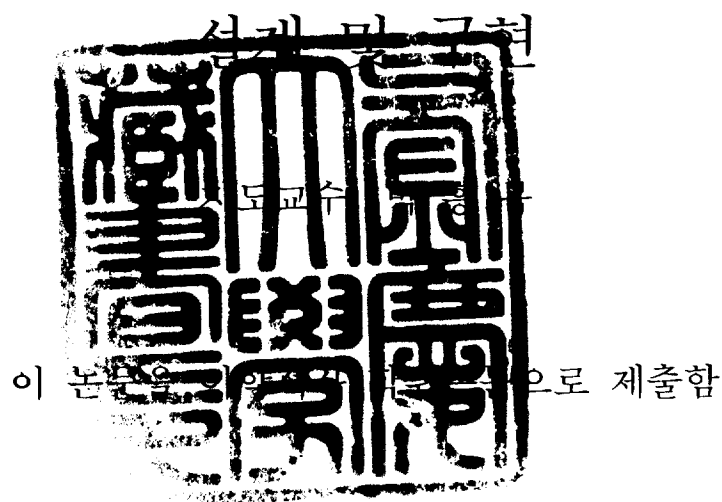


이학석사 학위논문

원격 컴퓨터 제어와 모니터링을
위한 모바일 디스플레이 시스템



2006년 2월

부경대학교 대학원

전자계산학과

천희자

천희자의 이학석사 학위논문을 인준함

2005년 12월 22일

주 심 공학박사 박승섭



위 원 공학박사 정순호



위 원 이학박사 박홍복



목 차

표 목 차	II
그 립 목 차	III
ABSTRACT	IV
1. 서 론	1
2. 관련 연구	3
2.1 기존 시스템 고찰	3
2.2 컴퓨팅 패러다임의 변화	4
2.3 모바일 플랫폼	6
2.4 JNI(Java Native Interface)	7
3. 시스템 설계	8
3.1 Mobile Viewer	9
3.2 VNC Monitor	12
3.3 VNC Server	13
3.4 Cellular RFB(CRFB) 프로토콜	17
4. 시스템 구현 및 결과 분석	19
4.1 시스템 구현	19
4.2 결과 분석	24
5. 결론	28
참고문헌	30

표 목 차

표1. 이동 통신사에 따른 무선 인터넷 플랫폼	7
표2. RFB 프로토콜	18
표3. CRFB 프로토콜	18
표4. 제안 시스템과 SVNC 비교	25
표5. 제안 시스템과 Rajicon 비교	26

그 립 목 차

그림1. 컴퓨팅 패러다임의 변화	5
그림2. J2ME의 구조	6
그림3. 시스템 구성도	8
그림4. 메시지 다이제스트 생성	9
그림5. VNC Server의 이미지 처리 과정	15
그림6. 현재 실행중인 프로그램 리스트를 구하는 C++ 코드	16
그림7. 프로그램 리스트를 구하는 dll을 호출하는 자바 클래스	17
그림8. 시스템 수행 과정을 나타낸 시퀀스 다이어그램	20
그림9. 모니터 실행 화면	21
그림10. 로그인 화면	21
그림11. VNC Server List	21
그림12. 전체 화면	22
그림13. 메인 메뉴	22
그림14. 서브 메뉴	22
그림15. Zoom in 화면	22
그림16. 마우스 클릭	23
그림17. 클릭 실행 화면	23
그림18. 메모리 사용량	23
그림19. 프로그램 리스트	23
그림20. 마우스 클릭	24
그림21. 오류 메시지	24

A Design and Implementation of Mobile Display System for Remote Computer Control and Monitoring

Hee-Ja Chon

Dept. of Computer Science Graduate School of
Pukyong National University

Abstract

With the beginning of the Internet and Web, IT technology moves toward Ubiquitous computing. Gradually, networking among computing devices is integrated and user's mobility is emphasized. Mobile devices' multimedia function and wireless networking ability guarantee the user's mobility so that we can use the mobile devices as the interface devices for interaction. In the networking enable environment, a user can see the data on the peripheral display devices and also remote objects on the user's mobile device. Therefore this paper designs and implements Mobile VNC system. That system can control and monitor a remote computer by using a cellular phone which is the most popular one among the mobile devices. This proposed System receives remote computer's screen images, controls GUI, monitors

the computer screen and system resources by a cellular phone. This system consists of the Mobile Viewer which runs on the cellular phone, VNC Server that runs on a number of remote computers and VNC Monitor that manages various VNC Servers. Screen images that are transmitted by Mobile Viewer are encoded as the PNG format. That format is supported by SK-VM that is the Java execution environment of the cellular phone. According to implementation's results, the size of image was about 2KB~8KB. As the result of the transmission test by 65 times, the average frame per second was 4.93fps in the emulator and 0.8fps in the cellular phone. Consequently the proposed system is more efficient than existing systems.

1. 서론

인터넷과 웹을 시작으로 IT 기술은 빠르게 발전하고 있다. 무선 네트워크와 소형 컴퓨터 기술의 발전은 모바일 컴퓨팅(Mobile Computing)을 이끌었고, 이제는 언제 어디서나 네트워킹이 가능한 유비쿼터스 컴퓨팅(Ubiquitous Computing)이 도래하고 있다. 현재 유비쿼터스 컴퓨팅은 기능의 내재성을 강조하는 방향과 컴퓨팅 디바이스의 이동성을 보장하는 방향으로 구현이 이루어지고 있는데 [1,2,15], 사용자와 디바이스의 이동성을 보장하기 위해 여러 컴퓨팅 환경이 서로 네트워킹되고 있고, 특히 모바일 컴퓨팅 디바이스의 무선 네트워킹도 강화되고 있다.

무선 네트워크 환경과 모바일 디바이스는 원격지에 위치한 컴퓨터 제어와 센서 모니터링, 데이터 전송 등의 원격 제어 시스템에 사용자의 이동성을 보장할 수 있다. 그리고 PDA와 휴대폰과 같은 모바일 디바이스들은 내장된 멀티미디어 기능과 휴대성, 무선 인터넷 기능으로 원격 제어 시스템의 컨트롤러로서 사용되고 있다 [3]. 따라서 사용자는 이동중에 자신의 모바일 디바이스 사용하여 네트워크에 존재하는 자원과 데이터에 접근할 수 있을 뿐만 아니라, 모바일 디바이스를 원거리에 위치한 대상을 디스플레이하는 장치로서 사용할 수 있다. 즉 모바일 디바이스에 제어 대상을 나타내는 원격 디스플레이 시스템(Remote Display System)이 제시되고 있다[3,4,5]. VNC(Virtual Network Computing)는 제어 대상인 원격 컴퓨터의 바탕화면을 다른 컴퓨터의 디스플레이 장치, 즉

모니터에 나타내는 공개 프레임워크이다[6]. VNC는 VNC 서버와 VNC 클라이언트로 구성되고, VNC 클라이언트는 VNC 서버가 전송하는 원격 컴퓨터의 화면 이미지를 전송받고 GUI 제어를 수행한다. 클라이언트와 서버는 통신을 위해 정의된 RFB(Remote Frame Buffer) 프로토콜을 사용한다[7].

본 논문은 모바일 디바이스 중에 가장 대중적으로 사용되고 있는 휴대폰을 VNC 클라이언트로 사용하여 원격 컴퓨터를 모니터링하고 GUI를 제어하는 모바일 디스플레이 시스템을 설계 및 구현하였고, 또한 JNI(Java Native Interface)를 사용하여 시스템 자원 모니터링 기능을 추가하였다. 휴대폰은 포인팅 장치의 부재, 키보드 및 화면 크기에 제약으로 표준 컴퓨터를 모델로 구상된 RFB 프로토콜을 사용하기에 적합하지 않으므로 RFB 프로토콜을 휴대폰 규격에 적합하게 수정한 CRFB(Cellular RFB) 프로토콜을 사용한다. 원격 컴퓨터의 화면 이미지는 휴대폰의 자바 실행 환경인 SK-VM이 지원하는 PNG(Portable Network Graphics) 포맷으로 인코딩한다.

본 논문의 구성은 다음과 같다. 2장에서 관련 연구를 소개하고, 3장은 제안하는 시스템의 세부 사항을 설명한다. 4장에서는 구현 및 결과를 분석하고, 끝으로 5장에서 결론을 기술한다.

2. 관련 연구

2.1 기존 시스템 고찰

원격 제어와 관련된 다수의 연구 중에서 휴대폰을 디스플레이 장치로 사용하여 제어 대상을 나타내는 접근 방식을 제안하는 시스템으로 Rajicon[4]과 SVNC 시스템[5]이 대표적이다. Rajicon은 사용자 인터페이스가 뛰어난 시스템으로 사용자의 편의성이 크지만 윈도우즈(Windows) 운영 체제에 한정되어 실행되고, 웹 서버의 CGI 스크립트 사용으로 인해 보안 문제가 발생할 수 있다.

SVNC 시스템은 본 시스템과 유사하게 VNC를 기반으로 한다. 그러나 SVNC 시스템은 VNC 서버와 SVNC 뷰어(Viewer) 사이에 웹 서버인 SNVC 프록시(Proxy)를 위치시켜 이미지 및 명령을 전송하므로 프록시가 실행 불능 상태일 경우 SVNC 뷰어와 VNC 서버 사이에 통신이 이루어지지 않는다. 즉, SPOF(Single Point Of Failure) 문제가 발생하게 된다. 그리고 다수의 VNC 서버가 실행 중일 경우, 각 서버가 실행 중인지 종료되었는지에 대한 가용한 정보를 얻을 수 없다. 따라서 다수의 VNC 서버를 관리하는 기능과 다수의 컴퓨터의 종료 및 재부팅과 같은 동시 제어 명령, 그리고 원격 컴퓨터에 대한 시스템 정보와 자원 정보를 알기 위해 기본적인 시스템 모니터링 명령이 필요하다.

따라서 본 논문에서 제안하는 시스템은 VNC 모니터(Monitor)를 사용하여 다수의 VNC 서버를 관리하고, 동시 제어 명령을 수행한

다. VNC 서버와 휴대폰 상의 VNC 클라이언트인 모바일 뷰어(Mobile Viewer)와 직접 연결되어 이미지와 명령을 전송하므로 통신중에 모니터의 개입을 최소화하였고, VNC 서버는 JNI를 사용하여 컴퓨터 자원에 접근할 수 있다.

2.2 컴퓨팅 패러다임의 변화

컴퓨팅 패러다임은 제 1세대 메인 프레임 기반 컴퓨팅, 제 2세대 개인용 컴퓨터(PC) 기반 컴퓨팅을 거쳐 1991년 마크 와이저가 제시한[8] 제 3세대 유비쿼터스 컴퓨팅으로 이동하고 있다.

제 1세대와 2세대는 전통적인 데스크탑 컴퓨팅 환경으로 컴퓨팅 기기와 애플리케이션이 모두 고정적이었다. 이에 반해 현재 무선 인터넷의 발달과 소형화된 모바일 디바이스의 컴퓨팅 능력이 향상되어, 모바일 컴퓨팅 환경에서 사용자는 이동중에 네트워킹이 가능하고, 네트워크에 존재하는 자원과 데이터, 서비스에 접근할 수 있다. 그리고 컴퓨팅 기기들이 환경에 내재되어 자연스러운 컴퓨팅이 가능한 편재형 컴퓨팅(pervasive) 연구가 진행되고 있다. 따라서 유비쿼터스 컴퓨팅은 이동성을 강조한 모바일 컴퓨팅과 기기의 내재성을 강조하는 편재형 컴퓨팅을 통합하여[9] 보다 편리한 인간 중심의 인터페이스를 제공하고, 장소와 시간에 따라 변화하는 서비스를 가능하게 할 것이다.

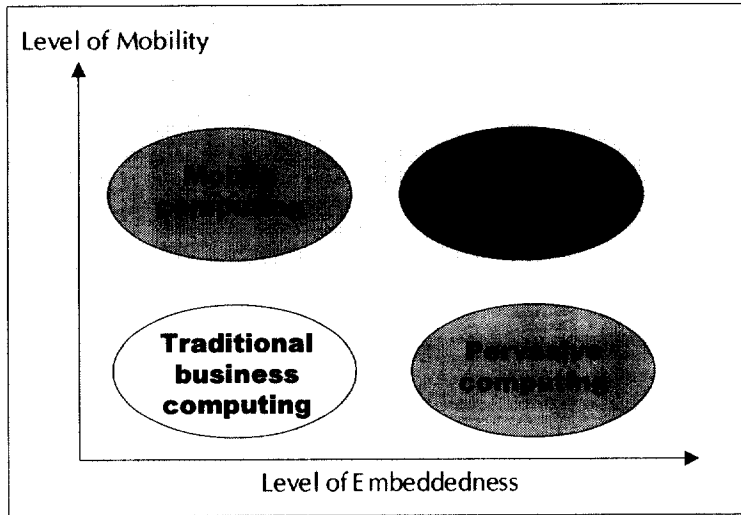


그림 1 컴퓨팅 패러다임의 변화[9]

10 여년전부터 미국과 유럽 등에서 국가적인 차원으로 유비쿼터스 컴퓨팅에 대한 연구가 진행되어 왔다. 미국은 가장 포괄적으로 유비쿼터스 컴퓨팅 연구가 실시되고 있으며, 스마트 디바이스에 의한 서비스, HP의 Cool Town 프로젝트 등의 상업적 연구가 진행되고 있다. 유럽은 컴퓨팅 장치를 사물과 환경에 내장하여 인간의 눈에 보이지 않는 상태에서 서비스를 제공하는 사라지는 (Disappearing) 컴퓨팅이 유비쿼터스 컴퓨팅과 같은 개념으로 추진되고 있다.

우리나라도 정부통신부의 u-Korea 정책에 따라 유비쿼터스 네트워크를 구축하여 국가 차원의 지능 기반 사회를 만들고, 새로운 경제 가치 창출을 목표로 하고 있다. 또한 유비쿼터스에 대한 일반 대중의 관심이 증가하고 있으며, 삼성의 홈비타(HomeVITA),

KT의 디지털 라이브 존(Digital Life Zone) 등 유비쿼터스 기술을 응용한 기술을 선보이고 있다[16]. 그리고 유비쿼터스 학회, 유비쿼터스 코리아 포럼 등이 설립되어 관련 기술 연구 개발과 산업 경쟁력을 확보할 수 있도록 학문적 연구를 계속하고 있다.

2.3 모바일 플랫폼

J2ME는 자바 기술을 모바일에 적용한 것으로 핸드헬드 디바이스, PDA, 휴대폰 등의 모바일과 임베디드 디바이스를 위한 플랫폼이다. J2ME는 크게 CLDC(Connected Limited Device Configuration)/MIDP(Mobile Information Device Profile)로 구성된다. J2ME의 실행 환경은 KVM(Kilobyte Virtual Machine)으로 소형의 자바 가상 기계를 목표로 한다. 그림 2는 J2ME의 구조를 나타낸다.

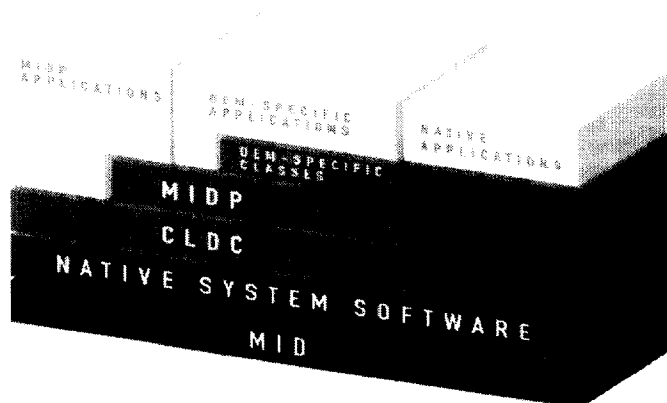


그림 2 J2ME의 구조[10]

국내 모바일 단말기의 플랫폼은 자바 기반의 VM 방식과 C 기반의 바이너리 코드(Binary Code)를 사용하는 방식, 그리고 국내 표준 플랫폼으로 WIPI가 있다. SK-VM은 J2ME(Java 2 Micro Edition)를 기반으로 작성된 애플리케이션을 다운 로드해서 실행시킬 수 있는 플랫폼이다. SK-VM은 SK 텔레콤의 사내 벤처인 XCE에서 clean room 방식으로 J2ME를 구현한 것으로 XVM(eXtended Virtual Machine)을 사용한다[10]. 그리고 그림 2의 OEM-SPECIFIC CLASSES에 해당하는 SK 텔레콤 자체의 자바 서비스 API를 제공한다. 본 논문은 시스템 개발과 실제 휴대폰 상의 실행을 위해 SK-VM을 사용한다. 표 1은 이동 통신사와 번호에 따라 사용하는 무선 인터넷 플랫폼을 정리한 것이다.

이동통신	번호	플랫폼
SKT	011, 017	SK-VM(J2ME + SKT API)
KTF	016, 018	Brew
LGT	019	Java Station(J2ME + LGT API)

<표 1> 이동 통신사에 따른 무선 인터넷 플랫폼

2.4 JNI(Java Native Interface)

JNI는 자바 기술중의 하나로 자바 이외의 언어로 작성된 프로그램

램을 자바 가상 머신위에서 실행할 수 있는 인터페이스를 제공한다[11]. 즉 C, C++과 어셈블리어로 작성된 응용 프로그램과 라이브러리를 실행할 수 있으므로 하드웨어 접근, 운영체제의 특수한 기능 제어 등을 위해 사용할 수 있다.

3. 시스템 설계

제안하는 시스템은 원격 컴퓨터에서 실행되는 VNC 서버(s), 웹 서버에서 실행되는 VNC 모니터, 휴대폰 상의 모바일 뷰어로 구성되며 통신을 위해 CRFB 프로토콜을 사용한다. 시스템 구성은 그림 3과 같다.

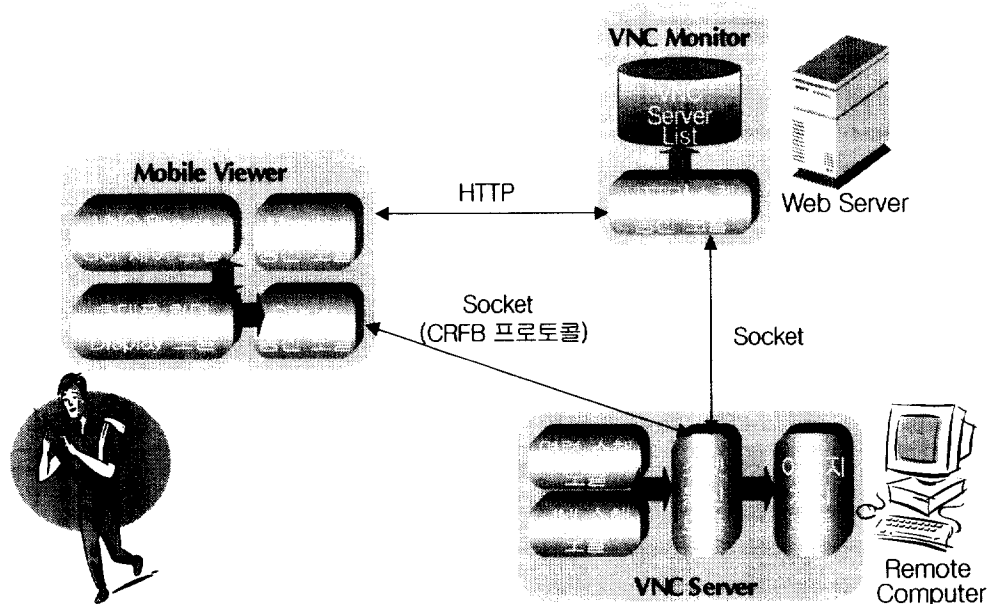


그림 3. 시스템 구성도

3.1 Mobile Viewer

모바일 뷰어는 휴대폰에서 실행되는 VNC 클라이언트로서 VNC 서버가 전송하는 화면 이미지를 기반으로 GUI 제어와 모니터링을 수행한다. SK-VM은 통신 시 1개의 소켓만을 사용할 수 있으므로 모니터와의 연결은 HTTP 커넥션(Connection)을 사용하고, VNC 서버와의 통신에는 소켓을 사용한다. 뷰어의 실행이 시작되면 사용자는 VNC 모니터가 실행되는 웹 서버의 IP와 인증을 위해 ID와 패스워드를 입력한다. 뷰어는 타임 스탬프(Timestamp)와 랜덤 수(Random Number), 입력한 ID와 패스워드를 SHA1 해쉬 알고리즘을 사용하여 160bit의 메시지 다이제스트(MD : Message Digest)를 구하고 이것을 모니터에 전송하여 사용자 인증을 요청한다. J2ME는 암호화를 위한 자바 암호 표준인 JCA(Java Cryptography Architecture)와 JCE(java Cryptography Extension)를 포함하지 않으므로 Bouncy Castle 라이브러리가[12] 제공하는 함수와 클래스를 사용한다. 인증 수행 과정은 그림 4와 같다.

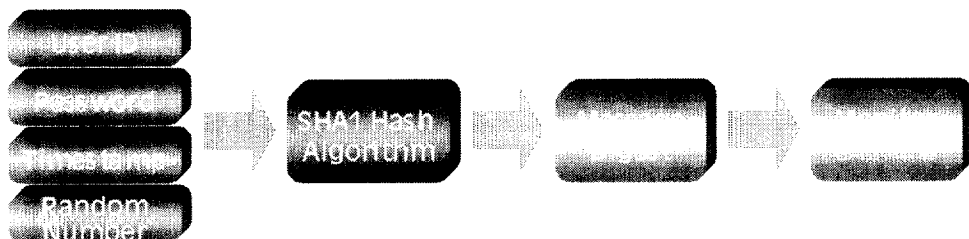


그림 4 메시지 다이제스트 생성

인증이 성공하면 모니터로부터 VNC 서버의 IP 리스트를 전송 받고, 제어를 시작할 VNC 서버 IP를 선택해서 서버와 뷰어 사이에 일 대 일 소켓 통신을 시작한다. 뷰어는 서버 리스트를 저장하고 있으므로 리스트에서 다른 서버를 선택하면 선택한 서버로 제어 이동이 이루어진다.

모바일 뷰어는 원격 컴퓨터의 화면을 보기 위해 Zoom in과 Zoom out 화면 모드와 컴퓨터의 GUI를 제어하기 위해 Mouse Click, Double Click 기능을 가진다. 그리고 원격 컴퓨터의 화면 네비게이션을 위한 스크롤링(Scrolling) 기능과 다수의 원격 컴퓨터 제어를 위한 동시 제어 명령, 그리고 시스템 자원 모니터링을 위한 명령들을 제공한다. 이러한 명령은 메인 메뉴와 서브 메뉴로 구분해서 제공되고 사용자는 상·하 버튼을 사용하여 선택한다. 메인 메뉴는 다음과 같다.

- Zoom in : 휴대폰의 가로와 세로 크기만큼 원격 컴퓨터의 화면 이미지를 크로핑(Cropping)해서 나타낸다.
- Zoom out : 원격 컴퓨터의 전체 화면 이미지를 휴대폰 가로와 세로 비율만큼 축소하여 나타낸다.
- Mouse : 가로와 세로 4 픽셀의 마우스 이미지를 휴대폰 화면에 나타내고, 이것은 휴대폰의 상·하·좌·우 버튼을 사용하여 이동시킨다.
- Click : 마우스 이미지가 위치한 좌표에서 클릭 명령을 선택하면 컴퓨터 모니터 화면의 대응하는 위치에서 마우스 클릭 명령이

실행된다.

- Double Click : 마우스 이미지가 위치한 좌표에서 클릭 명령을 선택하면 컴퓨터 모니터 화면의 대응하는 위치에서 더블 클릭 명령이 실행된다.
- Sub Menu : 서브 메뉴를 화면에 나타낸다.
- VNC Server List : 모니터에 연결된 다수의 VNC 서버의 IP 리스트를 재요청한다.
- Option : Zoom in 모드에서 마우스 이미지의 이동에 따른 화면의 해당 영역을 표시하는 스크롤링 명령과 1초 간격으로 원격 컴퓨터의 화면 이미지를 요청하는 Auto Refresh 명령을 포함한다. Auto Refresh 명령 실행 시 실제 이미지 표시까지 1초 이상이 소요된다.

서브 메뉴는 다음과 같다. 서브 메뉴는 원격 컴퓨터의 시스템 자원에 접근하는 명령을 포함한다.

- Computer Info : 원격 컴퓨터의 OS, CPU, 메모리량에 대한 정보를 나타낸다.
- Memory Usage : 원격 컴퓨터가 사용중인 메모리량을 가시적으로 나타낸다.
- Running Program : 원격 컴퓨터에서 실행중인 프로그램 리스트를 보여주고 리스트중에 한 프로그램을 선택하면 종료시킬 수 있다.

- Rebooting : 원격 컴퓨터를 재부팅한다.
- ShutDown : 원격 컴퓨터를 종료한다.
- All Rebooting : 모니터에 연결된 VNC 서버가 실행중인 모든 원격 컴퓨터를 재부팅하는 동시 제어 명령이다.
- All ShutDown : 모니터에 연결된 VNC 서버가 실행중인 모든 원격 컴퓨터를 종료시키는 동시 제어 명령이다.

SK-VM은 소켓 통신과 HTTP 통신을 동시에 지원하지 않는다. 따라서 모니터에 전달되는 동시 제어 명령과 VNC Server List 명령은 뷰어가 서버와의 소켓 통신을 종료한 후 모니터로 재접속하여 처리된다.

3.2 VNC Monitor

VNC 모니터는 각각의 원격 컴퓨터에서 실행하고 있는 다수의 VNC 서버와의 연결을 모니터링하고 뷰어에게 실행중인 VNC 서버의 IP를 전송한다. 모니터는 서버와의 통신을 위해 소켓을 사용하고, 뷰어와는 HTTP 커넥션을 통해 통신한다. 모니터는 실행중인 VNC 서버가 접속하면 VNC 서버의 IP를 저장하여 관리하고, 모바일 뷰어가 처음 접속해서 패스워드 기반의 사용자 인증을 성공하면 IP를 뷰어에게 전송한다. 사용자 인증 모듈은 Bouncy Castle 라이브러리를 사용하여 구현하였고, SHA1 해쉬 알고리즘을[12] 사용하여 뷰어가 전송한 메시지 다이제스트의 인증 여부를

처리하는 과정은 다음과 같다.

1. 뷰어로부터의 MD에서 ID, 패스워드, 타임스탬프, 랜덤 수를 추출한다.
2. 추출한 타임스탬프, 랜덤 수와 모니터가 보유한 사용자 ID와 패스워드를 이용해서 MD를 재생성한다.
3. 뷰어로부터의 MD와 생성한 MD를 비교하여 일치하면 뷰어에게 VNC 서버의 IP 리스트를 전송한다. 인증이 실패하면 뷰어에게 알리고, 사용자는 ID와 패스워드를 재입력한다.

뷰어가 VNC 서버 리스트를 재요청한 경우 VNC 모니터는 현재 연결된 VNC 서버의 IP 리스트를 재전송하여 뷰어가 제어할 수 있는 VNC 서버에 대한 가용한 정보를 유지하도록 한다. 또한 뷰어로부터 동시 제어 명령을 받아서 다수의 VNC 서버에 전송하여 제어가 수행되도록 한다. 그러나 뷰어와 VNC 서버 사이의 이미지 및 GUI 통신에는 개입하지 않으므로 모니터의 실행 불능으로 인한 전체 시스템 저하가 발생하지 않는다.

3.3 VNC Server

VNC 서버는 원격 컴퓨터 상에서 실행된다. VNC 서버는 실행 시작 시 VNC 모니터에 자신의 IP를 등록한다. VNC 서버는 모바일 뷰어와 연결이 성립되면 컴퓨터의 화면을 비트맵 이미지로 캡

처한다. 비트맵 이미지는 PNG 포맷으로 엔코딩되어 모바일 뷰어로 전송된다. 컴퓨터의 화면 이미지는 모바일 뷰어에서의 화면 모드인 Zoom out과 Zoom in 모드에 따라 다르게 처리된다. Zoom out 모드에서 사용자는 컴퓨터의 전체 화면을 휴대폰에서 모니터링하므로 이미지 크기를 휴대폰 화면 사이즈에 맞게 스케일링(Scaling)한다. 스케일링 시, 유사 변환(Affine Transformation)을 사용하여 화면 이미지를 휴대폰의 가로와 세로 비율에 맞춰서 축소한다. 식 (1)과 같이 크기 변환 시, 좌표(x, y)는 x 방향으로 α 만큼, y 방향으로 β 만큼 축소되어 새로운 좌표 (x' , y')를 생성한다.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \alpha & 0 \\ 0 & \beta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (1)$$

스케일링 인수 α 와 β 는 휴대폰과 컴퓨터 화면의 가로와 세로 비율이다. 유사 변환 시, 이미지 축소에 따른 질적 저하를 줄이고 변형될 픽셀값을 결정하기 위해 양선형 보간법(Bilinear Interpolation)을 사용한다. 양선형 보간법은 인접한 4개 화소의 가중치로 각 픽셀의 값을 계산하고 각 가중치는 화소로부터의 거리에 정비례한다[13].

Zoom in 모드는 휴대폰의 가로와 세로 크기만큼 화면의 특정 영역을 크로핑한다. 다운샘플링과 크로핑한 이미지는 Java Image I/O 클래스 함수를 사용해서 PNG 포맷으로 엔코딩되고, 엔코딩된 이미지 바이트 스트림은 모바일 뷰어 통신 모듈을 통해서 휴대폰

으로 전송된다. VNC 서버의 이미지 처리 과정은 그림 5와 같다.

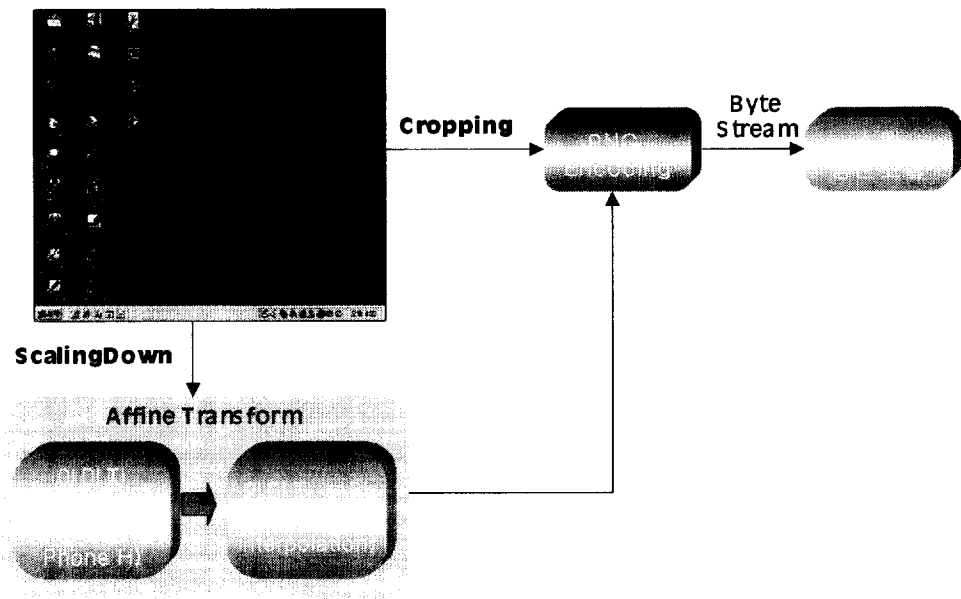


그림 5 VNC Server의 이미지 처리 과정

원격 컴퓨터의 GUI 제어 수행을 위해 VNC 서버는 모바일 뷰어로부터의 마우스 클릭과 더블 클릭 연산을 수행한다. 휴대폰과 컴퓨터의 화면 크기가 서로 다르므로 모바일 뷰어에서의 마우스 포인터의 좌표는 실제 컴퓨터 화면의 마우스 좌표와 차이가 난다. 따라서 모바일 뷰어로부터 전달된 x, y 좌표와 컴퓨터 화면 크기 등을 기반으로 휴대폰 화면에서 클릭(또는 더블 클릭)한 좌표를 산출한다.

VNC 서버는 GUI 명령뿐만 아니라 컴퓨터 정보, 메모리 사용량, 그리고 실행중인 프로그램 리스트를 구하여 모바일 뷰어에게 전달하여 사용자가 컴퓨터에 대한 시스템 정보를 확인하게 한다. 이러

한 정보는 시스템과 자원에 대한 접근을 해야 하므로 JNI를 사용하여 기능을 수행한다. 수행할 명령을 C++로 작성한 후 동적 링크 라이브러리(dll) 파일로 만든다. 뷰어로부터 해당 명령을 받으면 dll 파일을 로딩한 후 필요한 함수를 호출하고 결과를 텍스트로 뷰어에게 전송한다. 그림 6은 실행 프로그램을 구하는 C++ 코드이고, 그림 7은 자바에서 C++ 코드로 만든 dll을 호출하는 클래스이다.

```
JNIEXPORT jbyteArray JNICALL Java_NativeProcessTest_GetProcessList
(JNIEnv * env, jobject obj)
{
    int count = 0;
    BOOL bFirstModule = TRUE;
    CProcessIterator itp;
    CString nameList = _T("");

    if(pList.GetLength() >= 1)
        pList = "";

    for(DWORD pid = itp.First(); pid; pid = itp.Next())
    {
        TCHAR modname[_MAX_PATH];
        CProcessModuleIterator itm(pid);
        HMODULE hModule = itm.First();

        if(hModule)
        {
            GetModuleBaseName(itm.GetProcessHandle(), hModule,
                               modname, _MAX_PATH);

            BOOL bFirstWindow = TRUE;
            CMainWindowIterator itw(pid);

            for(HWND hwnd=itw.First();hwnd;hwnd=itw.Next())
            {
                ...
            }
        }
    }
}
```

그림 6. 현재 실행중인 프로그램 리스트를 구하는 C++ 코드

```

import java.util.*;

public class NativeProcessTest{

    static{
        System.loadLibrary("NativeProcessTest");
    }

    public native byte[] GetProcessList();
    public native byte[] GetModuleList();
    public native boolean killProcess(String data);
}

```

그림 7. 프로그램 리스트를 구하는 dll을 호출하는 자바 클래스

3.4 Cellular RFB(CRFB) 프로토콜

CRFB 프로토콜은 표준 컴퓨터를 모델로 설계된 VNC의 RFB 프로토콜을 휴대폰에서 사용할 수 있도록 필요한 일부분을 개선하였다. CRFB 프로토콜은 RFB 프로토콜의 메시지 타입을 그대로 사용하면서 필드를 추가 또는 삭제하였다. 아래의 표 2는 RFB 프로토콜의 PointerEvent의 구조를 나타낸다.

PointerEvent는 지정된 X, Y 좌표에 마우스 이벤트의 실행을 명령한다.

바이트	타입	설명
1	U8	메시지 타입
1	U8	button-mask
2	U16	X 좌표
2	U16	Y 좌표

<표 2> RFB 프로토콜

표 3은 CRFB 프로토콜에서 재정의한 PointerEvent의 구조를 나타낸다.

바이트	타입	설명
1	U8	메시지 타입
1	U8	화면 모드
1	U8	마우스 이벤트
2	U16	X 좌표
2	U16	Y 좌표

<표3> CRFB 프로토콜

CRFB 프로토콜에서 메시지 타입은 PointerEvent를 나타낸다. 추가된 화면 모드 필드는 뷰어의 두 가지 화면 모드인 Zoom in과 Zoom out을 구별하고, 마우스 이벤트 필드는 클릭과 더블 클릭을 구별한다. X 좌표와 Y 좌표는 휴대폰 화면에서 마우스 이미지의 X, Y 좌표를 뜻한다.

4. 시스템 구현 및 결과 분석

4.1 시스템 구현

VNC 서버는 RealVNC의 자바 공개 소스를 수정하여 J2SDK 1.4.2에서 구현 및 컴파일하였다[14]. 그리고 Microsoft Visual C++ 6.0을 사용해서 C++ 코드를 컴파일하고 dll 파일을 생성하였다.

VNC 서버는 Java Web Start 기술을 사용하여 원격지에 위치한 다수의 컴퓨터에서 다운로드해서 실행한다. 모바일 뷰어는 J2ME를 기반으로 구현하여 SK-VM 에뮬레이터 1.3.2로 테스트하고 SK-VM 1.2 버전이 탑재된 삼성 SCH-E380 휴대폰으로 실행하였다. VNC 모니터는 서블릿(Servlet)으로 구현하였고, Tomcat 4.1 웹 서버에서 실행한다. 그림 8은 시스템의 수행 과정을 UML의 시퀀스 다이어그램으로 나타낸 것이다.

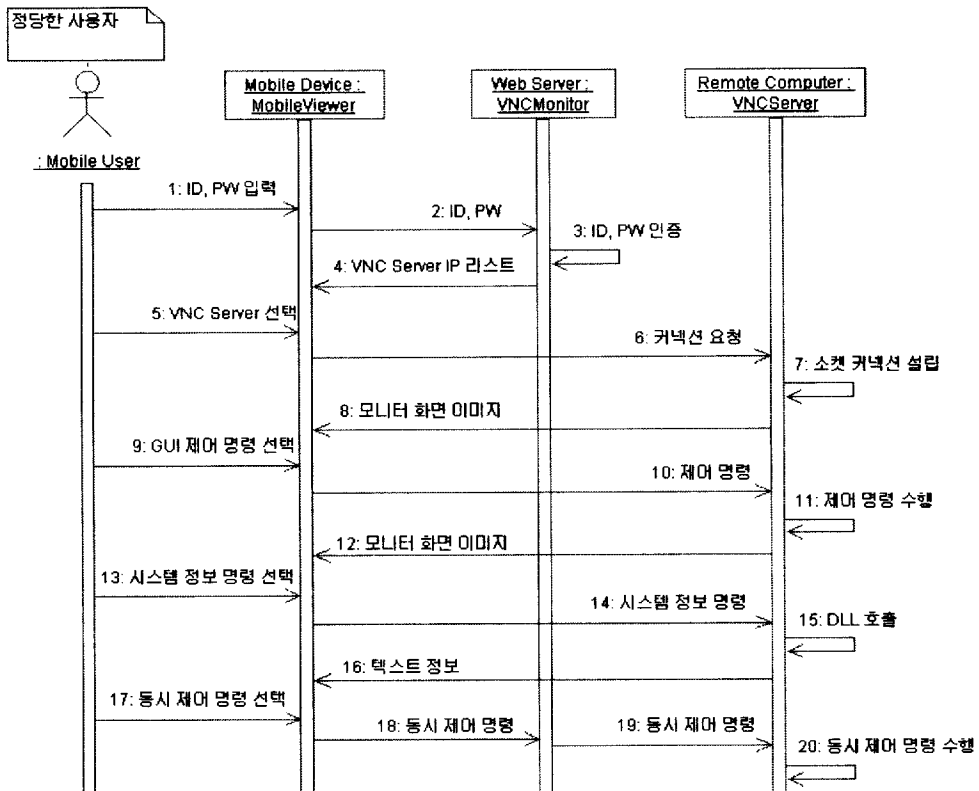


그림 8 시스템 수행 과정을 나타낸 시퀀스 다이어그램

그림 9는 VNC 모니터의 실행 화면이다. 실행중인 VNC 서버와의 연결 상태와 모바일 뷰어의 인증 여부를 텍스트로 나타낸다.

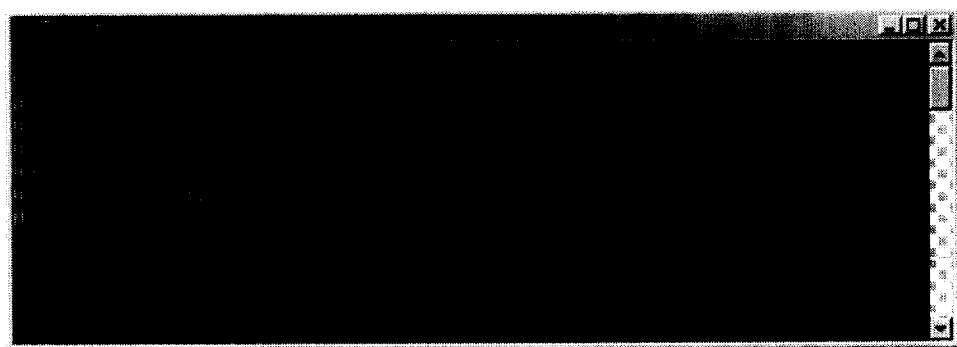


그림 9. 모니터 실행 화면

그림 10은 모바일 뷰어를 시작하면 처음 나타나는 화면으로 VNC 모니터에 접속하기 위해 ID와 패스워드를 입력한다. 사용자 인증이 성공하면 뷰어는 모니터로부터 현재 실행중인 VNC 서버의 IP 리스트를 전송받는다. IP 리스트는 그림 11과 같이 휴대폰 화면에 나타난다. 사용자가 제어할 VNC 서버의 IP를 선택한 후, 오른쪽 아래의 Connect 버튼을 누르면 VNC 서버와 통신이 시작된다.



그림 10. 로그인 화면

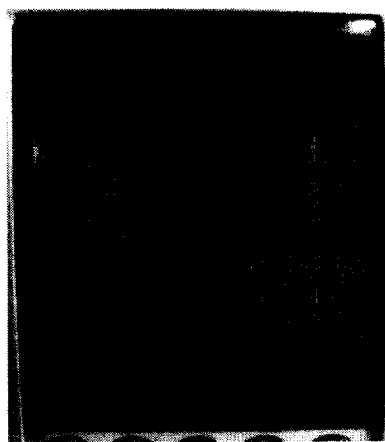


그림 11. VNC Server List

그림 12는 VNC 서버가 실행중인 원격 컴퓨터의 전체 화면 이미지이고, 그림 13은 모바일 뷰어가 제공하는 메인 메뉴, 그림 14는 서브 메뉴를 나타낸다. 메인 메뉴 중에 Zoom in을 선택하면 그림 15와 같이 화면의 특정 영역이 휴대폰 화면 크기에 맞춰서 나타난다.

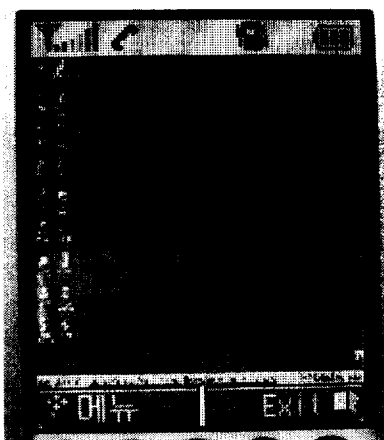


그림 12. 전체 화면

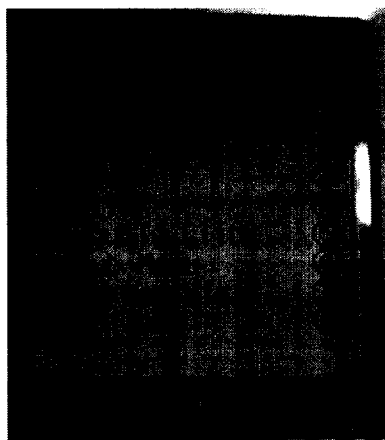


그림 13. 메인 메뉴

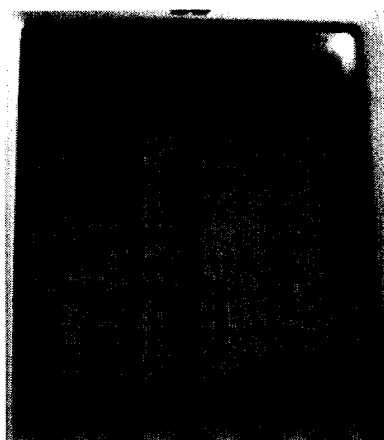


그림 14. 서브 메뉴

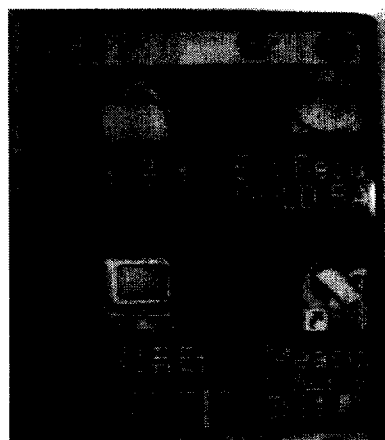


그림 15. Zoom in 화면

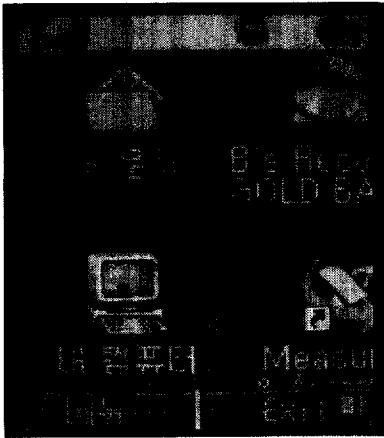


그림 16. 마우스 클릭

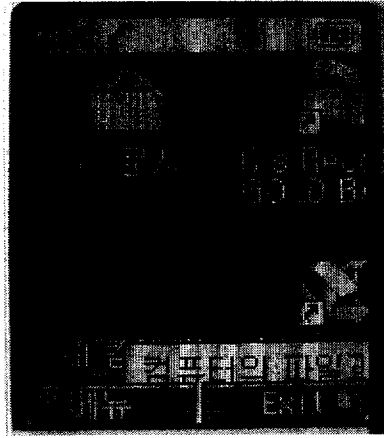


그림 17. 클릭 실행 화면

그림 16과 같이 휴대폰 화면에서 마우스 이미지를 내 컴퓨터 아이콘에 위치시킨 후 메뉴에서 Click을 실행하면, 원격 컴퓨터의 바탕 화면에서 내 컴퓨터 아이콘이 클릭되고, 그림 17과 같이 클릭된 화면 이미지가 휴대폰에 나타난다.

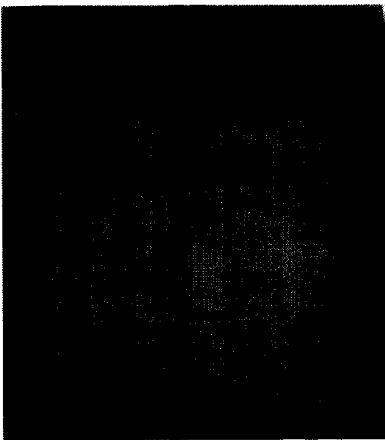


그림 18. 메모리 사용량

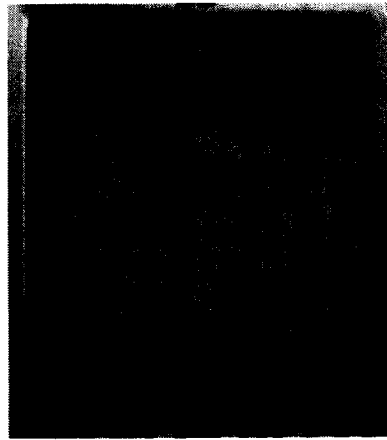


그림 19. 프로그램 리스트

그림 18은 VNC 서버가 실행중인 원격 컴퓨터의 메모리 사용률, 그림 19는 컴퓨터에 실행중인 프로그램을 리스트로 나타낸 것이다.

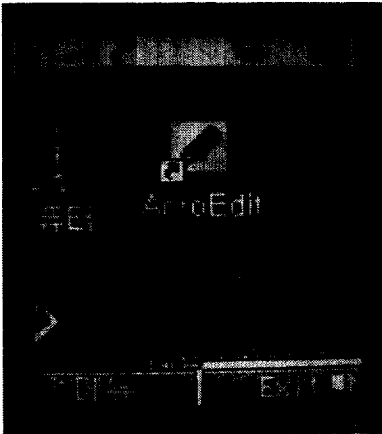


그림 20. 마우스 클릭

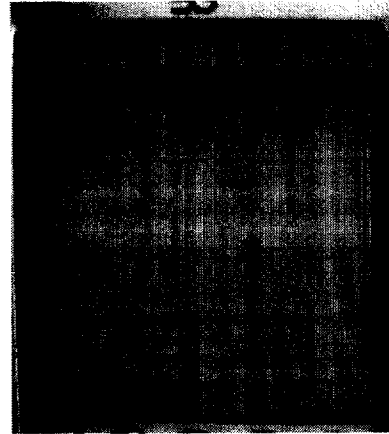


그림 21. 오류 메시지

뷰어는 명령이 실행되지 않은 경우 해당하는 오류 메시지를 사용자에게 알려준다. 예를 들어, 그림 20과 같이 바탕 화면의 아이콘을 클릭하는 명령이 실행되지 않은 경우에 그림 21과 같은 오류 메시지가 휴대폰 화면에 나타난다.

4.2 결과 분석

서버와 뷰어의 초기 접속 시간은 5초 정도 소요되고, 뷰어가 소켓 통신을 중단하고 HTTP 통신을 시도할 경우와 그 반대의 경우, 통신을 중단하는데 소요되는 시간을 고려해서 1초의 지연 시간을

할당하였다.

테스트는 에뮬레이터와 실제 삼성 SCH-E380 휴대폰에서 수행하였다. 제안 시스템의 VNC 서버가 전송하는 이미지 크기는 휴대폰의 화면 크기인 128×130이고, 32bit 색상의 PNG로 인코딩된다. 이미지 용량은 2~8KB이고, 이미지 프레임의 전송률은 0.4~1.1fps로, 즉 1개의 이미지 프레임을 전송하는데 0.9~2.6초가 소요되었다.

비교	시스템	제안 시스템	SVNC
이미지 포맷		PNG	BMP
크기		120 × 130 × 32bit	120 × 130 × 8bit
용량		2~8KB	12.2KB
초당평균전송률		4.93fps	3.9fps
에뮬레이터		SK-VM 1.3.2	J2ME Wireless SDK by NTT DoCoMo

<표 4> 제안 시스템과 SVNC 비교

표 4는 SVNC[5] 시스템과 본 시스템을 에뮬레이터로 테스트한 결과를 나타낸다. SVNC 시스템은 3비트의 Gray 이미지와 256 색상의 화면 이미지를 제공한다. 표 4에서는 256 색상의 이미지를 비교 대상으로 하였다. SK-VM에서 지원되는 이미지 포맷은 MIDP 규격에 정의된 포맷인 PNG와 XCE에서 자체 정의한 LBM이다. 본 논문의 제안 시스템은 컴퓨터의 화면 이미지를 PNG로

엔코딩하였다. SVNC 시스템에 비해 이미지 용량은 4KB 이상 감소했고, 초당 프레임 전송률(fps)은 4.93fps로 증가하였다.

비교	시스템	제안 시스템	Rajicon
이미지 포맷		PNG	GIF
크기		128 × 130 × 32bit	120 × 130 × 5bit
용량		2~8KB	2KB
초당평균전송률		0.8fps	0.5fps
휴대폰		삼성 SCH-E380	Panasonic P2101V SonyEricsson SO504i

<표 5> 제안 시스템과 Rajicon 비교

표 5는 실제 휴대폰으로 테스트한 제안 시스템과 Rajicon[4]을 비교한 것이다. Rajicon은 16 색상과 32색상의 화면 이미지를 제공한다. 표 5는 32 색상을 기준으로 한다. 제안 시스템과 Rajicon은 각각의 휴대폰 실행 환경에 맞게 이미지를 엔코딩하였다. 즉 제안 시스템은 PNG, Rajicon은 NTT DoCoMo의 DOJA라는 프로파일이 지원하는 GIF 포맷으로 화면 이미지를 엔코딩한다. 제안 시스템의 휴대폰은 데이터 통신으로 EVDO 기술을 사용하고 Rajicon은 FOMA, 즉 NTT DoCoMo의 IMT 2000 기술을 사용한다. 초당 전송률 테스트를 위해 Rajicon은 총 20회, 제안 시스템의 뷰어는 총 65회의 화면 이미지를 서버에게 요청하여, 각 이미지를 전송받아 화면에 표시하는 소요 시간을 계산하였다. 본 시스템의 평균

초당 전송률은 0.8fps로 1개의 이미지 프레임을 전송받아 표시하는데 평균적으로 1.22초가 소요되었고, Rajicon은 2.0초가 소요되었다. 따라서 제안 시스템의 이미지 용량은 상대적으로 크지만, 휴대폰과 원격 컴퓨터가 소켓 통신을 통해 직접 화면 이미지를 전송하므로 웹 서버를 통해 명령과 이미지를 전달받는 Rajicon보다 초당 전송률이 높게 측정된 것으로 추측된다.

실제 휴대폰 실행시 발생한 문제점으로 뷰어의 조작없이 원격 컴퓨터, 또는 VNC Server 프로그램이 종료된 경우 뷰어가 인식하지 못했다. 이 경우 사용자는 다른 실행중인 서버로 연결을 시도하거나 뷰어를 종료한다.

본 논문에서 제안한 뷰어는 SK-VM 1.1 이상 버전이 탑재된 휴대폰에서 실행할 수 있고, VNC 서버 프로그램은 Java Web Start 기술을 사용하여 각각의 컴퓨터로 다운로드해서 실행할 수 있다. 서버 프로그램은 Java의 보안 모델인 샌드박스(SandBox) 내에서 실행되므로 안전하게 실행할 수 있다. 제안한 시스템은 원격 컴퓨터의 화면을 휴대폰으로 볼 수 있으므로 텍스트 정보 뿐만 아니라 실시간으로 변화하는 그래프와 같은 개체도 확인할 수 있다. 또한 화면 상의 버튼 또는 아이콘과 같은 GUI 개체를 뷰어에서 클릭하거나 더블 클릭할 수 있고, 그 결과를 다시 휴대폰 화면으로 볼 수 있으므로 직접적인 제어가 가능하다. 그리고 JNI를 사용하여 컴퓨터 자원, 메모리 사용률, 실행 프로그램 등의 정보를 확인할 수 있고 사용자 임의대로 실행 프로그램을 종료시킬 수 있도록 하였다.

5. 결론

현재 무선 네트워킹, 모바일 디바이스의 소형화와 컴퓨팅 기능의 향상으로 원격 제어 시스템은 휴대폰, PDA와 같은 디바이스를 도입하고 있다. 본 논문은 휴대폰을 원격 컴퓨터를 디스플레이하는 장치로 사용하는 모바일 디스플레이 시스템을 제안하였다. 본 시스템의 사용자는 언제 어디서나 원거리에 위치한 컴퓨터 화면을 직접 볼 수 있고, GUI 제어와 시스템 자원 모니터링을 수행할 수 있다. 그리고 실제 휴대폰으로 실행하여 시스템의 유용성을 검증하였다.

제안하는 시스템은 휴대폰과 원격 컴퓨터가 직접 소켓 통신으로 연결되므로 기존의 시스템[4,5]이 웹 서버를 경유하여 이미지와 명령을 전송하는 방식보다 효율적이다. 또한 휴대폰에서 모니터에 연결된 모든 컴퓨터에 적용되는 동시 제어 명령으로서 종료와 재부팅을 실행하였고 JNI 명령을 추가하여 시스템 자원 정보에 접근하였다. CRFB 프로토콜은 VNC 시스템의 RFB 프로토콜을 휴대폰과 컴퓨터의 화면 차이를 고려하여 개선했기 때문에 뷰어와 서버 사이의 통신에서 적합하게 사용할 수 있다.

앞으로 다가올 유비쿼터스 환경에서 제어 대상은 앞으로 컴퓨터뿐만 아니라 컴퓨팅이 가능한 모든 장치로 확대할 수 있고, 또한 제어 대상을 나타내는 디스플레이 장치도 PDA, 인터넷 TV 등등 다양한 기기의 활용을 기대할 수 있을 것이다. 본 시스템은 이동성에 중점을 두고 설계된 반면에 능동적인 사용자 인터페이스를

제공하는 측면은 부족하므로 앞으로 이 부분을 보완해야 할 것이고, 휴대폰 프로그램을 이동통신과 휴대폰 기종에 상관없이 실행시키기 위해서 한국 무선 인터넷 표준화 포럼에서 정해진 WIPI(Wireless Internet Platform for Interoperability)로의 변환을 고려할 수 있다.

[참 고 문 헌]

- [1] Zhexuan Song, Pyusuke Masuoka, Jonathan Agre, Yannis Labrou, "Task Computing for Ubiquitous Multimedia Services", Proceedings of the 3rd international conference on Mobile and ubiquitous multimedia, October 2004
- [2] Angela Carrillo-Ramos, Jerome Gensel, Marlene Villanova-Oliver, Herve Martin, "PUMAS:A Framework based on Ubiquitous Agents for Accessing Web Information Systems through Mobile Devices", Proceedings of the 2005 ACM symposium on Applied computing, March 2005
- [3] Toshiaki Uemukai, Takahiro Hara, Masahiko Tsukamoto, Shojiro Nishio, "A Remote Display Environment : An Integration of Mobile and Ubiquitous Computing Environments", Wireless Communications and Networking Conference(WCNC), 2002. vol.2 pp.618 - 624, 2002
- [4] Norman Makoto Su, Yutaka Sakane, Masahiko Tsukamoto, Shojiro Nishio, "Rajicon:Remote PC GUI Operations via Constricted Interfaces", Proceedings of the 8th annual international conference on Mobile computing and networking, Session: Systems Issues, pp.251-262, 2002
- [5] Buntarou Shizuki, Masato Nakasu, Jiro Tanaka, "VNC-Based Access To Remote Computers From Cellular Phones", Proceedings of the IASTED International Conference On

- Communication Systems and Networks(CSN 2002), pp. 74-79, September 2002
- [6] T. Richardson, Q. Stafford-Fraser, K. R. Wood, and A. Hopper, "Virtual Network Computing", IEEE Internet Computing, 2(1), 33-38, 1998
- [7] T. Richardson and K. R. Wood, "The RFB Protocol Version 3.3", AT&T Laboratories Cambridge, January, 1998
- [8] M.Weiser, "The Computer of the 21st Century", Scientific American, vol.265 no.3, Sept. 1991, pp.66-75
- [9] Kalle Lyytinen, Youngin Yoo, "Issues and Challenges in ubiquitous computing", Communications of the ACM, vol.45 no.12, December 2002
- [10] www.xce.co.kr
- [11] <http://java.sun.com/j2se/1.5.0/docs/guide/jni/>
- [12] <http://www.bouncycastle.org>
- [13] Gonzalez, Digital Image Processing
- [14] <http://www.uk.research.att.com/archive/vnc/>
- [15] 삼성경제연구소, "유비쿼터스 컴퓨팅 : 비즈니스 모델과 전망" 2003, 12
- [16] 김진성, "유비쿼터스 컴퓨팅의 연구 동향과 프레임워크", 한국 퍼지및지능시스템학회춘계학술대회논문집, pp551-554, 2004
- [17] 천희자, 서정희, 임영진, 김영완, 허지훈, 박홍복, "휴대폰을 사용한 이미지 기반의 원격 PC 데스크탑 제어 시스템의 설계

- 및 구현”, 한국정보처리학회추계학술발표대회논문집, 2004, 11
- [18] 천희자, 김영완, 서정희, 박홍복, “휴대폰을 사용한 원격 모니터링 및 정보 시스템 설계”, 한국해양정보통신학회추계종합학술대회논문집, vol8. no2, pp 605~608, 2004, 10
- [19] 천희자, 서정희, 박홍복, “원격 컴퓨터의 GUI 제어와 모니터링을 위한 Mobile VNC 시스템의 설계 및 구현”, 한국해양정보통신학회논문지, 제 9권 제 5호, pp.912~919, 2005, 9

감사의 글

석사 과정을 마치며 저를 이끌어주신 많은 분들께 감사한 마음을 전합니다.

먼저 작은 일에도 세세한 관심과 애정을 보여주신 박홍복 지도교수님께 깊은 감사를 드립니다. 그리고 부족한 논문을 살펴보며 지도해주시고 또한 격려해주신 박승섭 교수님과 정순호 교수님께 깊은 감사의 마음을 전하고 싶습니다. 수업을 통해 가르침을 주셨던 여정모 교수님, 윤성대 교수님, 김창수 교수님, 이경현 교수님, 박만곤 교수님, 신상욱 교수님, 故 박지환 교수님께 다시 한 번 감사를 드립니다.

저에게 언제나 필요한 조언을 아끼지 않으셨던 서정희 선배님과 인자하신 김영완 선생님, 마음을 나누었던 임영진 선배님과 한은영님, 졸업하신 허지훈 선배님과 김유경 선배님, 그리고 이영철 선배님께 감사드립니다. 연구실 식구인 김민희 후배님, 김경미 후배님, 권소연 후배님, 이왕근 후배님과 우리 연구실에서 생활했던 모든 선후배님들께 감사의 말을 전합니다. 그리고 따뜻하게 대해주신 윤가림 선배님과 윤요섭 선배님, 서철 선배님, 이해란 선배님과 대학원 선후배님들 모두에게 감사드립니다.

이 세상에서 가장 소중한 저의 부모님, 오라버니, 올케, 조카에게 감사드립니다. 언제나 제게 가족이 있다는 것이 큰 힘이 되었습니다.

끝으로 저와 인연이 닿았던 모든 분들께 감사한 마음을 전합니다.

2005년 12월

천희자 드림