

모바일 기반 시연 시스템 '스마트 데모'의 설계 및 구현

Design and Implementation of A Mobile-based Demonstration System 'Smart Demo'

임충규

성공회대학교 멀티미디어시스템공학과

Choong-Gyoo Lim(cglim@skhu.ac.kr)

요약

어플리케이션의 시연은 일련의 상호작용을 포함하기 때문에 많은 시간이 소요된다. 어플리케이션을 시연해야 한다면 통상적으로 발표 세션과 별도로 시연 세션을 진행한다. 하지만, 모바일 단말기를 조작하여 무선으로 어플리케이션을 시연한다면 발표자가 직접 시연을 주도적으로 진행함으로써 전달하고자 하는 메시지를 효과적으로 전달할 수 있을 것이다. 이러한 시연 방법은 기존 모바일 단말기의 성능과 화면 크기의 제약 때문에 구현이 불가능하였다. 우수한 성능과 다양한 기능을 갖춘 스마트폰의 등장으로 스트림 동영상을 실시간으로 그리고 상대적으로 큰 모바일 단말기 화면에서 재생할 수 있고, 또한 시연자가 모바일 단말기 화면의 어플리케이션 영상을 보면서 어플리케이션을 직접 제어할 수 있기 때문에 새로운 시연 방법의 구현이 가능해졌다. 본 논문은 스마트폰을 활용한 모바일 기반 시연 시스템인 '스마트 데모(Smart Demo)'의 설계와 구현에 대해 기술하고자 한다.

■ **중심어** : | 어플리케이션 시연 | 모바일 시연 | 스마트폰 시연 | 실시간 스트리밍 | 생중계 비디오 |

Abstract

An application's demonstration takes much time because there are a number of interactions involved. When a demonstration is indeed necessary, a separate session of demonstration is commonly carried out besides a presentation. If one can control the application wirelessly using mobile devices, he can present his messages effectively by having a live self-paced demonstration. This method of demonstration has not been possible to implement due to the limitations on performance and/or screen size of feature phones. Lately commercialized smartphones, however, are equipped with better performance and various features so that they can be utilized to implement the new demonstration method. With these smartphones, one can view the application images right on mobile devices to control applications directly. This paper describes the issues of design and implementation of a mobile-based demonstration system, 'Smart Demo'.

■ **keyword** : | Application Demonstration | Mobile Demonstration | Smartphone Demonstration | Realtime Streaming | Live Video |

* 본 연구는 성공회대학교 2011학년도 교내 연구비 지원 사업의 일부로서 수행됨.

접수번호 : #110818-005

심사완료일 : 2011년 10월 21일

접수일자 : 2011년 08월 18일

교신저자 : 임충규, e-mail : cglim@skhu.ac.kr

1. 서론

우수한 성능과 다양한 기능을 갖춘 스마트폰의 출현은 산업 활동뿐만 아니라 일상생활에 대해서도 많은 변화를 일으키고 있다. 핸드폰, PDA와 같은 기존 모바일 단말기와 달리 스마트폰은 충분한 컴퓨팅 자원을 탑재하고 있어서[1][2] 기존의 업무를 거의 똑 같이 처리할 수 있기 때문이다[3-5]. 대표적인 서비스 중의 하나는 실시간 스트리밍 기반 생중계 서비스로서[6] [그림 1]과 같은 서비스를 스마트폰에서도 제공 받을 수 있다.

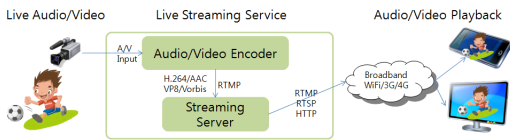


그림 1. 실시간 스트리밍 기반 생중계 서비스

이러한 기술을 활용한다면, 모바일 단말기를 활용하여 어플리케이션을 제어하는 모바일 기반 시연 서비스의 구현도 가능하다. 모바일 기반 시연 서비스는 [그림 2]처럼 사용자는 어플리케이션 호스트 시스템에서 전송되어오는 어플리케이션 영상을 직접 보고 어플리케이션 및 호스트 시스템을 제어한다. 모바일 단말기는 사용자 제어에서 발생하는 입력 이벤트를 호스트 시스템에 전달하고 호스트 시스템은 이를 어플리케이션에 반영하여 시연을 진행하는 서비스이다.

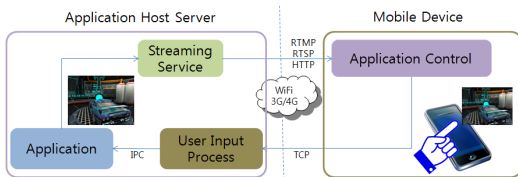


그림 2. 모바일 기반 시연 서비스

시연은 일반적으로 많은 사용자 인터랙션이 필요하며 많은 시간이 소요된다. 특히, 많은 키보드 입력이 요구된다면 짧은 발표 시간에 같이 진행할 수 없다. 짧은 시연이더라도 발표자가 어플리케이션과 호스트 시스템

을 직접 조작하는 경우, 적지 않은 시간이 소요되거나 발표 시스템과 시연 시스템의 조작이 원활하지 않아 발표 흐름을 방해하는 경우가 있다. 이러한 이유로 별도의 발표 세션을 갖거나 시연 장면을 축약하여 동영상으로 제작하여 발표 내용에 포함시킨다.

모바일 기반 시연 서비스는 기존의 피쳐폰이 보유하고 있는 VOD 중심의 스트리밍 기술과 제한적인 성능을 사용하여 구현이 거의 불가능하다. VOD를 위한 스트리밍 기술은 수초 이상의 스트리밍 지연이 불가피하고[7], 고품질 영상을 실시간으로 재생하기에 피쳐폰의 성능은 미흡하다. 최근에는 스트리밍 지연이 0에 가까운 실시간 스트리밍 기술이 생중계 서비스의 필요에 따라 출현하고 있으며[8] 대부분의 스마트폰은 고품질의 영상을 별 문제없이 재생할 수 있기 때문에 모바일 기반 어플리케이션 시연 서비스를 어렵지 않게 구현할 수 있다.

본 논문은 모바일 기반 어플리케이션 시연 서비스 시스템인 ‘스마트 데모(Smart Demo)’의 개념을 수립하고, 시스템의 설계와 구현에 대해 서술한다. 2장에서 주요 배경기술에 대해서 조사한다. 3장에서는 시연 시스템의 설계에 대하여, 4장에서는 시연 시스템의 구현에 대해서 기술한다. 5장에서 결론을 맺는다.

II. 배경 기술

1. 실시간 스트리밍 기반 생중계 기술

1.1 실시간 인코딩 기술

실시간 스트리밍 기반 생중계 기술에서 중요한 세부 기술 중의 하나가 실시간 인코딩 기술이다. 서비스 영상 품질과 지연 시간은 인코딩의 복잡도와 소요시간에 따라 크게 좌우되기 때문이다.

가장 관심을 끄는 코덱 중의 하나는 VP 계열의 코덱이다. 2008년 9월에 On2 사가 발표한 VP8 코덱은 2010년 구글이 On2를 인수 후 공개한 오픈 비디오 압축 포맷이다. 영상 품질 자체에는 다양한 의견이 있지만[9], 모든 CPU 자원을 이용하여 인코딩 속도를 재생 속도와 거의 같게 할 수 있으며, 지연 없이 가능한 최대 품질의

영상을 만들 수 있다는 점은[10] 생중계 서비스에 필요한 VP8의 장점이다. 2003년 10월에 발표된 VP6은 다수의 인코더와 디코더가 지원하고 있지만[11]¹ 주요 모바일 플랫폼인 안드로이드에서 지원하지 않는다. VP8은 안드로이드 2.3.3버전 이상에서 지원된다[12].

2003년 5월에 표준이 작성된 H.264 AVC 코덱은 HD급 비디오의 저장, 압축, 배포를 위한 포맷으로 Blu-ray, YouTube, iTunes, Flash Player, Silverlight, DVB, SBTVD 등에서 널리 사용되고 있다[13]. 아주 낮은 비트레이트를 사용하여 고품질의 영상을 제작하기 위하여 설계된 포맷이기 때문에 인코딩-디코딩 지연이 다소 있지만[14], 최근에는 비디오 컨퍼런스, 영상 통화 등의 영향으로 본 코덱을 사용하더라도 스트리밍 지연을 최소화하는 노력이 이루어지고 있다[15].

WebM은 HTML5와 같이 사용하고 로열티 없는 공개 비디오 압축 포맷을 제공하기 위해 설계된 오디오/비디오 포맷이며, 개발을 담당하는 WebM 프로젝트는 2010년 5월 시작되었으며 구글이 개발을 지원하고 있다. 안드로이드 2.3 버전부터 WebM을 지원하고 있으며, 포맷에 대한 지원이 확대되고 있다[16].

아도비(Adobe)가 개발하여 무료로 배포하고 있는 Flash Media Live Encoder는 일부 테스트에서 0.96초의 인코딩 지연이 보고되고 있다[8]. 반응성에 대한 요구 수준이 높은 게임 스트리밍 분야에서 사용하기 어렵지만, 시연 시스템에서는 최대 1초 정도의 지연은 허용될 수 있다. 유사한 서비스 중의 하나인 비디오 컨퍼런싱은 최대 1초 정도의 지연을 허용한다. 지원하는 코덱은 H.264와 VP6이다. 이밖에도 VLC 미디어 플레이어 등이 인코더로서 사용되고 있다.

1.2 실시간 스트리밍 기술

실시간 스트리밍을 위한 프로토콜은 RTSP(Real Time Streaming Protocol), RTMP(Real Time Message Protocol), HTTP Progressive Streaming, HTTP Live Streaming, HTTP Dynamic Streaming 등이 있다.

애플에서 개발한 HTTP Live Streaming 기술은 HTTP 기반의 미디어 스트리밍 통신 규약으로서 Quicktime X와 아이폰 소프트웨어의 일부로 구현되었

다. 대용량의 미디어 파일을 작은 크기의 HTTP 파일로 나누고 이를 단말기에 다운로드하여 재생한다. HTTP 프로토콜을 이용하기 때문에 방화벽이나 프락시 서버(proxy server)에 대한 제약이 없는 특징이 있다. 2009년 6월 IETF의 표준으로 제안되었으며 Internet-Draft 상태이다. Wowza Media Server, VLC 미디어 플레이어 등 다수의 스트리밍 서버가 지원하고 있으며 안드로이드 3.0, iOS 3.0 등의 클라이언트가 지원하고 있다.

아도비가 개발한 HTTP Dynamic Streaming 기술은 애플의 HTTP Live Streaming처럼 작은 크기의 스트림을 HTTP를 통하여 전달함으로써 방화벽이나 프락시 서버의 제약을 피할 수 있다. Adobe Flash Access 2를 사용하여 강력하고 유연하게 콘텐츠를 보호할 수 있다[17].

마크로미디어(Macromedia)에 의해서 개발된 RTMP는 Flash 플레이어와 서버 간의 오디오, 비디오 데이터를 스트리밍하기 위하여 개발되었다[18]. 낮은 지연, 빠른 재생 시작, 동적인 버퍼링, 스트림 암호화 등의 특징이 있다[19].

RTSP는 1998년 IETF의 MMUSIC(Multi-party Multimedia Session Control) WG에 의해 개발되고 표준화되었다. 미디어 데이터 자체의 전송을 위해 RTP를 사용하고, 메타 정보를 위해 RTCP를 사용한다. 미디어 스트리밍 서버를 제어하고 엔터테인먼트나 통신 시스템에서 사용하기 위하여 설계된 네트워크 제어 프로토콜이며, 클라이언트와 서버 간의 세션을 구성하고 관리한다[20].

이러한 미디어 스트리밍 프로토콜을 지원하는 실시간 미디어 스트리밍 서버로는 Flash Media Server, Wowza Media Server 등이 있다.

2. 실시간 스트리밍 기반 생중계 기술의 응용

2.1 게임 스트리밍

실시간 스트리밍 기술을 이용하는 응용 서비스 중의 하나가 게임 스트리밍이다. 대규모의 클라우드 노드를 구성하고, 컴퓨터 게임을 각 클라우드 노드 상에서 실행한다. 노드의 게임 영상은 미디어 스트림 형태로 유

무선 네트워크를 통하여 원격 단말기에 전송한다. 게이머는 원격 단말기에 전송된 게임 영상을 보고 게임을 플레이하고, 사용자 입력은 클라우드 노드에 즉시 전송되어 게임에 반영된다[21].

대표적인 서비스는 미국 OnLive 사의 OnLive 서비스이다[그림 3][22]. 2010년 6월에 미국 내에서 상업 서비스를 시작하였으며, 2011년 9월에는 영국에서 서비스를 시작할 예정이다[23]. 이외에도 GaiKai[24], Games@Large[25] 등의 기술이 게임 스트리밍 서비스를 위해서 개발되었다.

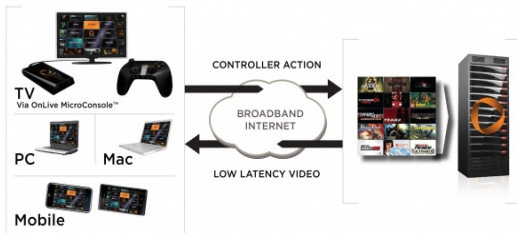


그림 3. OnLive 서비스의 개념도

2.2 텔레로보틱스(Telerobotics)

텔레로보틱스는 무선 통신을 이용해서 원격지의 로봇을 제어하기 위한 로보틱스(robotics) 분야의 기술이다. 모든 동작을 자체적으로 수행하는 로봇(robot)과 달리, 장비의 동작을 운영자가 지시한다. 이러한 장비를 텔레로봇(telerobot)이라고 부르며, 운영자는 텔레로봇에 설치된 카메라에서 전송되어오는 영상 이미지를 참고로 원격 조정 장치를 제어하여 텔레로봇을 조정한다. [그림 4]는 사람에게 치명적인 위험이 될 수 있는 화재 현장에서 사용되는 소방 로봇[26]과 아주 높은 정밀도를 필요로 하는 수술 로봇[27]의 예이다.

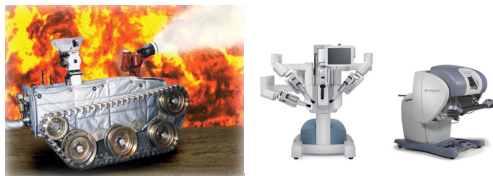


그림 4. 소방 로봇과 수술 로봇의 예

2.3 스마트폰 기반 원격 제어 시스템

스마트폰의 보급이 확대됨에 따라 스마트폰을 활용하여 PC를 제어하는 어플리케이션이 등장하고 있다. 헝가리의 LogMeIn 사는 자사의 RDP(remote desktop protocol)을 적용한 LogMeIn이라는 원격 제어 솔루션을 상용화하였다[그림 5]. 원격제어 이외에도 협업, 기술 지원 기능을 포함하고 있으며 전용 어플리케이션이나 웹 브라우저를 통하여 원격 제어가 이루어진다. 독일의 TeamViewer사는 동명의 원격제어 소프트웨어를 개발하여 출시하였으며, 유무선 인터넷을 통해 다른 플랫폼을 제어하는 기능을 PC, Mac, Linux 뿐만 아니라 스마트폰에서도 제공하고 있다. 비상업적 용도인 경우 무료이다. 원격 제어 이외에도 협업과 발표의 기능을 제공한다.



그림 5. LogMeIn의 실행 예

이외에도 많은 스마트폰 어플리케이션이 출시되고 있지만 발표 현장에서 S/W 시스템을 시연하는 경우, 시스템에 따라서 많은 사용자 인터랙션이 요구되는 경우가 있어서 사용하기 쉽지 않거나 발표 진행에 방해가 될 수 있다. 발표와 동시에 시연을 효율적으로 진행하기 위해서는 사용자 입력을 저장하여 재생하는 기능과 텍스트 입력을 간편하게 할 수 있는 기능 등이 필요하다.

III. 스마트 데모 시스템의 설계

1. 요구사항 분석

1.1 기능적 요구사항

스마트 데모 시스템을 구현하기 위하여 다음과 같은 기능이 요구된다.

- ① **시연 환경 설정** - 시연 대상이 되는 어플리케이션의 명칭, 실행 파일 경로, 실행 매개변수 값, 호스트 시스템 이름 및 IP 주소 등을 설정하여 시연 환경을 준비한다.
- ② **어플리케이션의 구동과 종료** - 모바일 단말기에서 사용자 입력을 통하여 윈도우 시스템의 어플리케이션을 구동하거나 종료시킨다. 원격 단말기에 사용자가 등록한 어플리케이션 정보, 실행 파일 경로, 실행 매개변수 값 등을 호스트 시스템에 전달하여 어플리케이션을 구동하도록 한다.
- ③ **사용자 입력의 캡처** - 모바일 단말기에서 발생하는 사용자 입력 및 명령을 캡처한다. 어플리케이션의 사용자 입력은 주로 마우스, 키보드에서 입력이 발생하지만 스마트폰에서는 터치, 터치 이동, 버튼 키, 소프트 키보드가 불리는 가상 키보드의 키 누름 등이 있다.
- ④ **사용자 입력의 저장 및 재생** - 모바일 단말기 상에서 발생하는 사용자 입력을 DB의 테이블로 저장하고 필요에 따라 재생할 수 있다.
- ⑤ **사용자 입력의 전송** - 시연 프로그램을 제어하기 위하여 모바일 단말기 상에서 발생하는 사용자 입력을 원격 시스템으로 무선 네트워크를 통하여 전송한다.
- ⑥ **사용자 입력의 변환** - 모바일 단말기의 사용자 입력은 어플리케이션 및 호스트 시스템의 구동에 필요한 사용자 입력과 많은 차이가 있다. 모바일 단말기의 사용자 입력이 어플리케이션 및 호스트 시스템에 유효한 사용자 입력이 되도록 변환한다.
- ⑦ **사용자 입력의 어플리케이션 반영** - 어플리케이션 및 호스트 시스템에 맞게 변환된 사용자 입력을 적용한다.
- ⑧ **어플리케이션 영상 캡처 및 스트리밍 서비스** - 어플리케이션의 영상을 실시간으로 캡처하고 영상을 스트림 패키지로 모바일 단말기에 전송한다.
- ⑨ **스트림 영상 재생** - 호스트 시스템에서 전송받은 영상 스트림을 모바일 단말기에서 재생한다. 시연자는 재생된 영상을 참조하여 어플리케이션을 직접 제어할 수 있다.

- ⑩ **시나리오 구성** - 모바일 단말기에 시나리오를 구성하여 저장하고 최소한의 사용자 인터랙션을 활용하여 시연을 진행할 수 있도록 한다.

1.2 성능적 요구사항

네트워크를 통하여 서비스되는 모든 시스템은 네트워크 지연이 서비스 성공의 결정적 요인이다. 게임 스트리밍인 경우 200ms(5fps)의 지연이 게임을 즐기기 위한 최소한의 지연으로 여겨지며, 즉각적 반응성이 필수적인 레이싱 게임이나 FPS(First Person Shooting) 게임의 경우, 80ms(12.5fps)가 최소한의 지연으로 여겨지고 있다[28]. 본 시스템은 1장의 1.1절에서 언급된 것처럼 최대 1msec의 지연을 목표로 한다.

스마트폰과 스마트패드는 다양한 해상도를 지원하기 때문에 영상 스트리밍 서비스도 다양한 해상도를 지원해야 한다.

2. 시스템의 구성

스마트 데모 시스템은 [그림 6]과 같이 어플리케이션 호스트 서버와 모바일 시연 클라이언트로 이루어진다. 어플리케이션 호스트 서버는 어플리케이션 영상의 실시간 스트림을 담당하는 기능(스트리밍 서버), 어플리케이션의 실행, 정지 등을 담당하는 기능(어플리케이션 관리 모듈)을 갖는다. 모바일 시연 클라이언트는 전송된 어플리케이션의 스트림 영상을 재생하는 기능(스트림 영상 재생 모듈), 어플리케이션 제어 기능(어플리케이션 제어 모듈), 사용자 입력 캡처 및 전송 기능(사용자 입력 캡처 모듈)을 갖는다.

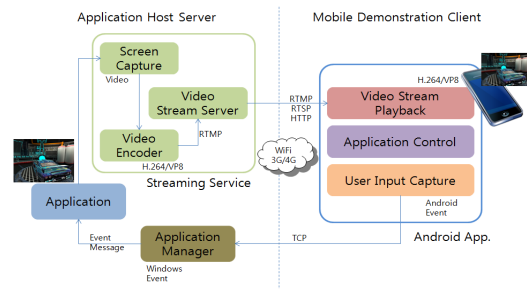


그림 6. 모바일 기반 시연 시스템의 구성

2.1 어플리케이션 호스트 서버

스트리밍 서버를 세부 모듈로 구분하면 어플리케이션 창에 대한 영상 캡처 모듈, 실시간 비디오 인코딩 모듈, 스트림 서비스 모듈로 구성된다. 비디오 영상 캡처 모듈은 스크린 또는 어플리케이션 창의 영상을 확보하여 실시간 비디오 인코딩 모듈에 전달하며, 실시간 비디오 인코딩 모듈은 대용량의 비디오 영상을 네트워크로 전송하기 위하여 적절한 크기로 압축하는 기능을 담당한다. 스트림 서비스 모듈은 원격 단말기가 끊김 없이 실시간으로 비디오 영상을 재생할 수 있도록 비디오 스트림을 작은 크기로 패키징하고 전송하는 기능을 담당한다.

어플리케이션 관리 모듈은 모바일 단말기로부터 전송받은 사용자 입력을 호스트 시스템과 어플리케이션에 적절한 사용자 입력으로 변환하는 기능과 호스트 시스템의 IPC(Inter-Process Communication)를 통하여 사용자 입력 또는 명령을 어플리케이션에 반영하는 기능을 수행한다. 모바일 단말기의 사용자 입력과 호스트 시스템의 사용자 입력은 그 종류와 양에서 많은 차이가 있다.

2.2 모바일 시연 클라이언트

모바일 시연 클라이언트는 [그림 6]처럼 어플리케이션 호스트 서버에서 전송되어온 비디오 스트림을 안드로이드 단말기에서 재생하는 비디오 영상 재생 모듈, 어플리케이션의 구동, 정지, 일시 정지 기능을 담당하는 어플리케이션 제어 모듈, 모바일 단말기의 사용자의 입력 이벤트를 캡처하여 어플리케이션 관리 모듈에 전송하는 사용자 입력 캡처 모듈로 구성된다. 사용자 입력 캡처 모듈은 사용자의 입력을 DB에 저장하고 재생하는 기능을 포함한다.

사용자 입력 전송 모듈은 모바일 단말기에서 발생하는 사용자 입력을 전송하기 위하여 TCP 소켓을 이용한다.

IV. 스마트 데모 시스템의 구현

모바일 클라이언트는 안드로이드 플랫폼을 활용하여

구현한다. 아직은 사용자 만족도나 앱 종류 면에서 아이폰 플랫폼에 미흡하지만 2010년 4분기 기준으로 전세계 스마트폰 시장의 33%를 차지함으로써 가장 많은 시장 점유율을 기록할 정도로 일반적인 플랫폼이 되었다[29]. 일반적으로 활용 가능한 2.2 버전을 기반으로 구현한다. 호스트 서버는 윈도우 플랫폼 상에서 구현한다. 대다수의 어플리케이션이 개발되는 플랫폼이고, 대부분의 동영상 인코더와 미디어 서버가 지원하는 플랫폼이다.

윈도우 운영체제 시스템을 비롯한 많은 시스템은 비동기 방식의 TCP 소켓을 지원하나 안드로이드 플랫폼은 비동기 방식의 TCP 소켓을 지원하지 않기 때문에 TCP 소켓을 이용하여 비동기 통신을 하기 위해서는 안드로이드 플랫폼의 서비스 컴포넌트를 활용하여 이와 유사한 서비스를 구현하여야 한다. 액티비티가 동기화 소켓을 처리하는 경우, 일정 시간 액티비티가 정지될 수 있고, 일정 시간이 지나면 강제 종료되는 상황이 벌어질 수 있다. 따라서, 상기의 목적을 달성하기 위하여 구현되는 네트워크 서비스는 스레드(Thread) 객체인 러너블(Runnable) 객체를 구현하여 사용한다.

1. 어플리케이션 호스트 서버의 구현

1.1 스트림 서비스 모듈

안드로이드 플랫폼의 레퍼런스 문서에 따르면 플랫폼이 지원하는 비디오 포맷은 인코더와 디코더에 따라 다르며, 동영상 재생에 필요한 디코더 포맷은 [표 1]과 같이 H.263, H.264 AVC, MPEG4 SP, VP8이며 VP8은 안드로이드 2.3.3이상에서 지원된다. VP6가 1초 미만의 지연을 갖기 때문에 최소한의 지연이 필수조건인 스마트 데모 시스템에 적절한 포맷이다. 하지만, 안드로이드 플랫폼에서 VP6를 지원하지 않기 때문에 H.264 코덱을 사용한다. 안드로이드 플랫폼이 지원하는 전송 프로토콜은 RTSP, HTTP Progressive Streaming, HTTP Live Streaming이다. 애플에서 개발한 HTTP Live Streaming은 안드로이드 3.0이상에서만 지원한다. HTTP Progressive Streaming과 HTTP Live Streaming은 공히 HTTP 프로토콜을 이용하기 때문에

방화벽과 프락시 서버의 제약 없이 동영상 스트림을 원격 단말기에 전송할 수 있다는 장점이 있다. 스마트 데모 시스템을 비롯한 생중계 시스템은 지연 시간이 최소화되어야 하기 때문에 HTTP Live Streaming을 사용해야 한다. 하지만 HTTP Live Streaming은 안드로이드 3.0이상에서 지원하고 안드로이드 3.0 이상을 탑재한 단말기는 아직 보급이 미진한 상태이므로 스마트 데모 시스템은 RTSP 프로토콜을 이용하여 구현한다.

RTSP 전송 프로토콜과 MP4 파일 포맷을 동시에 지원하는 스트림 서버는 Wowza Media Server, Evostream Media Server, Darwin Streaming Server, Helix Universal Server 등이 있다. 현 스마트 데모 구현의 목적이 자체적인 스트림 서버를 구현하는 것이 아니고 또한 각 스트림 서버의 성능을 비교 분석하는 것이 아니기 때문에 위에 열거한 Wowza Media Server를 사용하여 구성하였다.

표 1. 안드로이드 플랫폼의 코덱

코덱	인코더	디코더	파일 포맷
H.263	지원	지원	3GPP(.3gp) MPEG-4(.mp4)
H.264 AVC	3.0 이상	지원	3GPP(.3gp) MPEG-4(.mp4) MPEG-TS(.ts)
MPEG-4 SP		0	3GPP(.3gp)
VP8		2.3.3 이상	WebM(.webm)

스크린 캡처는 SplitMediaLabs 사의 VHSrCap이라는 툴을 사용한다.

1.2 어플리케이션 관리 모듈

모바일 클라이언트에서 전송되어진 사용자 입력은 호스트 시스템과 어플리케이션이 필요로 하는 사용자 입력과 다르기 때문에 이를 변환하는 과정이 [표 2]와 같이 필요하다. 구현에 사용되는 플랫폼과 어플리케이션의 종류에 따라 사용하는 입력 이벤트의 종류와 변환 이벤트가 달라질 수 있다. 윈도우 플랫폼을 사용하기 때문에 윈도우 메시지를 어플리케이션에 전달하는 방법을 채택하였다.

스마트 데모는 윈도우 운영체제 기반의 시스템에서 동작하는 어플리케이션을 안드로이드 단말기 시스템을 이용하여 원격 구동하는 것을 목표로 한다. 두 시스템의 UI는 많은 면에서 상이하기 때문에, 시연 대상 어플리케이션의 동작에 필요한 UI 이벤트를 안드로이드 단말기의 UI 이벤트로 매핑하는 작업이 필요하다. 해당 UI 이벤트가 안드로이드 단말기에서 발생할 때, 이를 윈도우 시스템의 UI 이벤트로 변환하여 어플리케이션에 적용한다.

표 2. 사용자 입력 이벤트의 변환

Window System				Android System	
사용자 입력	Window Event	WPARAM	LPARAM	사용자 입력	Android Event
Left Mouse Down	WM_LBUTTONDOWN	MK_LBUTTON	x y<<16	Touch	ACTION_DOWN
Left Mouse Drag	WM_MOUSEMOVE	MK_LBUTTON	x y<<16	Drag	ACTION_MOVE
Key	WM_KEYDOWN	key 값	0	Key	KEYCODE * -
Key	WM_KEYDOWN	1개 이상의 key 값	0	Virtual Key	1개 이상의 key 값

모바일 단말기는 대부분 물리적 키보드가 부착되어 있지 않기 때문에 어플리케이션에서 필요한 키보드 입력을 위해서는 소프트웨어 키보드(Soft Keyboard)라고 불리는 온스크린 버추얼 키보드(On-Screen Virtual Keyboard)를 사용해야 한다. 키보드 입력을 줄일 수 있는 자동 단어 완성 기능을 지원해야 하며 시나리오 기반의 시연 시스템 구현을 위해서는 모바일 단말기 상의 키입력을 저장하고 불러 사용하는 과정이 필요하다.

모바일 단말기의 크기 및 성능은 타겟 시스템에 비해서 상당히 제한적이다. 특히, 사용자 입력은 소프트웨어 키보드를 이용한 입력을 제외한다면 클릭, 긴 클릭, 터치(드래그 이동 포함) 등 3가지로 제한되어 있다. 이들을 활용하여 어플리케이션을 제어하려면 [표 2]와 같이 사용자 입력을 시연에 필요한 타겟 시스템의 사용자 입력으로 변환하는 시스템이 필요하거나 시연 대상 어플리케이션에 필요한 사용자 입력으로 변환하는 시스템이

필요하다. 안드로이드 플랫폼의 터치(Touch) 입력은 세부적으로 MotionEvent의 ACTION_DOWN(터치인 경우)나 ACTION_MOVE(드래그)로 구분되고, 이러한 터치 입력은 윈도우 플랫폼에서 각각 WM_LBUTTONDOWN 이벤트와 WM_MOUSE_MOVE 이벤트로 변환되어 어플리케이션에 적용된다. 터치 입력의 위치는 윈도우 메시지에 LPARAM 매개변수 값으로 반영된다.

사용자 입력 처리 모듈은 어플리케이션을 제어하고 윈도우 메시지를 전달하는 역할을 담당하기 때문에 데몬이나 서비스 형태로 구현될 수 있는데, 이번 구현에서는 [그림 7과 같이 MFC의 Dialog 형태로 구현하였으며, 메시지 창을 두어 동작 상황을 쉽게 파악할 수 있도록 하였다. 사용자 입력 이벤트에 대한 네트워크 패킷을 수령하기 위하여 MFC CAsyncSock 클래스를 활용하여 TCP 소켓을 구현하였다.

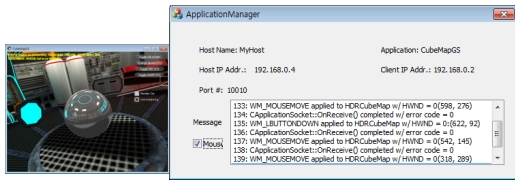


그림 7. 어플리케이션 관리 모듈의 실행 예

2. 모바일 시연 클라이언트의 구현

모바일 단을 구성하는 스트림 영상 재생 모듈, 어플리케이션 제어 모듈, 사용자 입력 캡처 및 전송 모듈은 [그림 8]과 같이 구현하였다.

구체적으로 어플리케이션의 시작과 종료 명령을 내리고 네트워크 세션을 관리하는 주 액티비티(Main Window)로 구성된다[그림 9]. 액티비티는 안드로이드 플랫폼에서 UI 창에 해당하는 컴포넌트이다. 뷰 액티비티(View Window)는 사용자의 명령과 입력을 캡처하고 네트워크 서비스를 통하여 어플리케이션 호스트 서버에 전달한다.

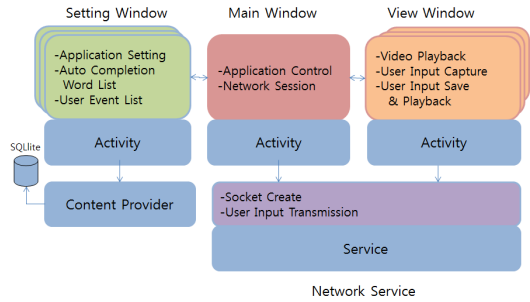


그림 8. 모바일 시연 클라이언트의 구현

어플리케이션이 호스트 서버에서 실행되면 어플리케이션의 영상 스트림을 뷰 액티비티는 SurfaceView를 활용하여 화면에 표시한다[그림 10]. 다수의 어플리케이션을 동시에 시연하거나 시나리오 장면마다 별도의 액티비티를 구현하는 경우, 다수의 뷰 액티비티를 갖는다. 네트워크 서비스는 서비스 컴포넌트를 활용하여 구현하며 서비스 요청을 주 액티비티와 뷰 액티비티로부터 받아 TCP 소켓을 통하여 호스트 시스템에 전송하는 작업을 담당한다. 어플리케이션의 이름, 실행 파일 경로, 실행 변수 등을 저장하고 설정하기 위하여 하나의 액티비티로 설정 창(Setting Window)을 구현하며, 단어 자동 완성 기능을 위한 단어 목록 설정용 액티비티와 사용자 입력 확인 및 편집을 위한 액티비티를 추가로 구현한다[그림 11][그림 12]. 뷰 액티비티는 사용자 입력을 저장하고 재생하는 기능을 가지며 [그림 12]는 저장된 사용자 입력을 리스트 형태로 확인하는 화면의 예이다. 기존 원격 제어 시스템과 달리 스마트 데모 시스템은 사용자 인터랙션을 최소화할 수 있도록 사용자 입력을 저장하고 재생하는 기능을 구현하였고, 단어 자동 완성용 단어 등록 기능을 추가로 구현하였다.

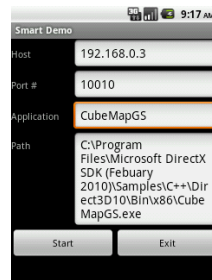


그림 9. 모바일 시연 클라이언트의 주 화면

V. 결론 및 향후 연구 계획

본 연구는 기존의 IT 서비스를 모바일 단말기 상에서 구현한 다른 모바일 앱과 달리 스마트폰이 출현되면서 새롭게 가능해진 모바일 기반 시연 서비스의 개념을 구체화하고 이를 구현했다는 의의가 있다. 기존의 시연 방법과 달리 발표자의 메시지를 효과적으로 전달할 수 있다.

이렇게 구현된 모바일 기반 시연 시스템은 다음과 같은 장점이 있다.

않고 발표와 함께 시연을 진행할 수 있다.

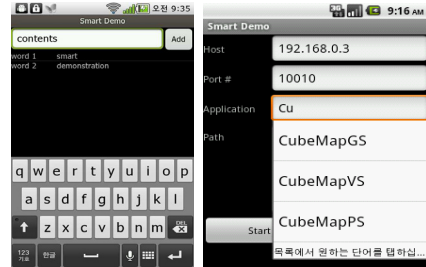


그림 11. 자동 완성용 단어 목록 화면



그림 10. 모바일 시연 클라이언트



그림 12. 사용자 입력 이벤트 목록

- ① **청중의 관심 증대** - 발표자의 메시지를 전달하는 중간에 시연을 간결하게 진행함으로써, 청중의 관심을 집중시킬 수 있고 전달하고자 하는 메시지를 효과적으로 전달할 수 있다.
- ② **발표자 주도적/임기응변 가능** - 발표자가 모바일 단말기를 사용하여 직접 시연을 진행하므로 발표자가 상황에 따라 속도를 조절하여 발표자의 메시지를 효과적으로 전달할 수 있다. 자세한 설명이 필요한 경우에는 많은 시간을 할애하여 설명하거나 다수의 장면을 재 시연하는 것도 가능하다.
- ③ **별도의 시연 세션 불필요** - 많은 시간이 소요되는 시연은 비교적 짧게 주어지는 발표 시간에 진행하지 못하고 별도의 시연 세션을 갖는 경우가 있다. 일반적으로 많은 시간이 소요되는 시연 과정은 발표자가 원하는 메시지를 간결하면서도 효율적으로 전달하는데 지장을 주기 때문이다. 빠른 시연이 가능하기 때문에 별도의 시연 세션을 갖지

- ④ **추가적인 시연 인력/훈련 불필요** - 다소 복잡한 어플리케이션 시스템을 시연하기 위해서는 다수의 시연 인력이 필수적이다. 하지만, 하나의 모바일 단말기에서 다수의 호스트와 어플리케이션을 한꺼번에 제어할 수 있기 때문에, 1인 시연이 가능하다. 어플리케이션 시스템 시연을 진행하거나 돕기 위하여 별도의 시연 인력이 필요하거나 시연을 위한 별도의 훈련이 필요하지 않다.
- ⑤ **동영상 제작/화질 걱정 불필요** - 시스템을 실제 상황에서 실시간에 시연하기 위해서는 시간이 많이 소요되지만 비교적 짧게 주어지는 시연 시간에 시연을 맞추기 위하여 주요 장면을 동영상으로 제작하여 실시간 시연을 대체하는 경우가 있다. 스마트 데모 시스템을 사용한다면 사용자 입력을 빠른 속도로 진행하여 동영상을 활용하지 않고도 짧은 시간에 실시간 시연을 진행할 수 있는 장점이 있다. 실제 시연이 동영상보다 신뢰도가 높아 전

달하고자하는 메시지의 전달에 도움이 된다.

- ⑥ **시나리오 사전 구성 및 시행** - 시연 구성에 필요한 다수의 모바일 UI 화면을 미리 제작된 시나리오를 바탕으로 구성할 수 있다.

이번 스마트 데모 시스템의 구현은 안드로이드 2.2 버전을 기반으로 이루어졌다. 안드로이드 3.0 버전 이상의 단말기 보급이 일반적이지 않기 때문이다. 실시간 스트리밍과 관련하여 안드로이드 플랫폼은 3.0 버전 이상에서 HTTP Live Streaming, WebM, VP8 등에 대한 지원이 추가 되었다. 향후, 안드로이드 3.0 버전의 단말기를 활용하여 구현할 예정이고, 우수한 어플리케이션 영상의 스트림을 재생하면서 동시에 스트리밍 지연을 대폭 감소시킬 수 있을 것으로 기대된다. 현재의 구현은 3초 정도의 스트리밍 지연이 존재한다. 코드와 UI를 자동으로 생성하는 편집 툴을 개발한다면 모바일 플랫폼의 프로그램 개발에 익숙하지 않은 사용자도 시나리오 기반 모바일 시연 시스템의 구축이 가능하다.

참 고 문 헌

- [1] Smartphone Definition from PC Magazine Encyclopedia, www.pcmag.com
- [2] smartphone, www.phonescoop.com
- [3] 2011 Trends Report: Smartphone Application Development, www.focus.com
- [4] 이고은, 이종우, “스마트폰 상에서의 웹 응용프로그램 개발 환경 비교”, 한국콘텐츠학회논문지, 제10권, 제12호, pp.155-163, 2010.
- [5] 오형용, 민병원, “스마트폰 앱 디자인 스타일 및 사용성 분석”, 한국콘텐츠학회논문지, 제10권, 제12호, pp.129-136, 2010.
- [6] 윤이진, “라이브 스트리밍 비디오 시장 동향”, KT경제경영연구소, 2011.
- [7] <http://www.adobe.com/products/flashmediaserver/flashmediaencoder/faq/>
- [8] <http://www.iupui.edu/~nmstream/live/findings.php>
- [9] J. Ozer, “VP8 vs. H.264”, www.springmedia.com, 2011.
- [10] <http://en.wikipedia.org/wiki/VP8>
- [11] <http://en.wikipedia.org/wiki/VP6>
- [12] <http://developer.android.com/guide/appendix/media-formats.html>
- [13] http://en.wikipedia.org/wiki/H.264/MPEG-4_AVC
- [14] http://www.axis.com/products/video/about_networkvideo/compression.htm
- [15] <http://x264dev.multimedia.cx/archives/249>
- [16] http://en.wikipedia.org/wiki/HTML5_video
- [17] http://www.adobe.com/products/httpdynamics_treaming/
- [18] http://en.wikipedia.org/wiki/Real_Time_Messaging_Protocol
- [19] http://www.adobe.com/products/httpdynamics_treaming/
- [20] <http://en.wikipedia.org/wiki/Rtsp>
- [21] 임충규, 김성수, 김경일, 원중호, 박창준, “클라우드 컴퓨팅 기반의 게임 스트리밍 기술 동향”, 전자통신동향분석, 제26권, 제1호, pp.47-56, 한국전자통신연구원(ETRI), 2011.
- [22] www.onlive.com
- [23] <http://onlivefans.com/news/onlives-cloud-video-game-service-launches-in-the-uk-on-22nd-september/2075/>
- [24] www.gaikai.com
- [25] Y. Tzruya, Games@Large-a new platform for ubiquitous gaming and multimedia, Broadband Europe Conference 2006, 2006.
- [26] <http://robotstocknews.blogspot.com/2007/04/firefighting-robots-from-irobot.html>
- [27] www.intuitivesurgical.com
- [28] <http://news.bbc.co.uk/2/hi/technology/7976206.stm>
- [29] <http://www.canalysis.com/pr/2011/r2011013.html>

저 자 소 개

임 충 규(Choong-Gyoo Lim)

정회원



- 1988년 2월 : 서울대학교 수학교육과(이학사)
- 1990년 2월 : 서울대학교 대학원 수학교육과(교육학석사)
- 1998년 5월 : 美 루이지애나 주립대 대학원 Computer Science

(Ph.D)

- 1999년 2월 ~ 2011년 2월 : 한국전자통신연구원 책임연구원
- 2011년 3월 ~ 현재 : 성공회대학교 멀티미디어시스템공학과 교수

<관심분야> : 기하 모델링, 컴퓨터 그래픽스, 컴퓨터 게임