

## 데이터 방송 시스템 시뮬레이션을 위한 Virtual Set Top Box

전제민<sup>○</sup>, 고상원, 김세창, 원재훈, 김정선  
한양대학교 컴퓨터공학과 HPC&OT 연구실  
luvjjm@naver.com<sup>○</sup>, funkcode@gmail.com, {sckim, jhwon, jskim}@cse.hanyang.ac.kr

### The Virtual Set Top Box for Data Broadcasting System

Je-Min Jeon<sup>○</sup>, Sang-Won Ko, Se-Chang Kim, Jae-Hun Won, Jung-Sun Kim  
HPC&OT Lab, Dept. of Computer Eng., Hanyang University

#### 요 약

최근 방송 통신 기술의 눈부신 발전에 힘입어 등장한 데이터 방송(Data Broadcasting)은 시청자에게 영상과 음성 뿐만 아니라 기존에는 접할 수 없었던 다양한 서비스들을 제공하고 있다. 이런 데이터 방송을 수신하기 위해서는 TS로부터 데이터를 추출하고 처리할 수 있는 미들웨어(middleware)가 포함된 셋탑박스(set-top box)가 필요하다. 그리하여 MHP, OCAP, ACAP 등 여러 데이터 방송 시스템의 미들웨어 표준 규격이 정의되었다. 그 중 MHP의 경우, OpenMHP라는 오픈 소스 프로젝트를 통해 PC 상에서 간단하게나마 MHP API를 테스트 해 볼 수 있는 환경을 제공해 주고 있다.[1] 하지만 셋탑박스 내에서 하드웨어를 이용하는 작업들은 거의 구현되지 못 했기 때문에 테스트에 있어 극히 제한적일 수 밖에 없었다. 따라서 본 논문에서는 소프트웨어적으로 셋탑박스 환경을 에뮬레이션 해주는 Virtual Set-Top Box(VSTB)를 설계, 구현하였다. 셋탑박스가 없는 환경이나 간단한 시뮬레이션이 필요한 경우, 실시간 스트림 역다중화와, 파싱, 필터링 작업등이 지원되는 가상 셋탑박스 시스템을 통해 MHP 미들웨어 API를 테스트 하는데 유용한 역할을 할 수 있을 것이다.

#### 1. 서 론

데이터 방송(Data Broadcasting)이란 디지털 신호를 이용하여 음성과 영상 이외에 다양한 정보를 부가적으로 전달하는 새로운 형태의 방송을 지칭한다. 데이터 방송의 등장으로 인해 시청자들은 TV를 통해 프로그램 가이드를 받아보거나 콘텐츠와 동기화된 응용 프로그램을 실행하는 등 기존의 방송 시스템에서는 불가능했던 여러 가지 유용한 서비스들을 제공 받을 수 있게 되었다.

일반적으로 데이터 방송을 수신하기 위해서는 트랜스포트 스트림(TS, Transport Stream)을 역다중화(demultiplex) 하여 데이터를 추출하고 처리 할 수 있는 셋탑박스(STB, Set-Top Box)가 필요하다. 이런 셋탑박스 내에는 데이터 방송 프로토콜을 지원하고 스트림에 포함된 응용 프로그램을 실행시키기 위한 미들웨어(middleware)가 포함된다. 그동안 데이터 방송 미들웨어 시스템의 표준 규격을 정의하기 위해 많은 작업들이 이루어져 왔으며 그 중 대표적인 것으로는 DVB의 MHP(Multimedia Home Platform)와 CableLabs의 OCAP(OpenCable Applications Platform)등이 있다.[2]

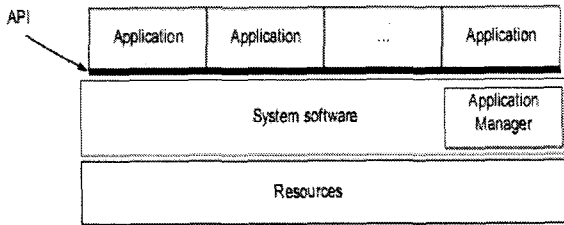
본 논문에서는 MHP 미들웨어 API를 윈도우즈(Windows) 기반의 PC 상에서 테스트 할 수 있도록 지원해주는 Virtual Set Top Box(이하 VSTB)를 제안한다. VSTB는 셋탑박스 내에서 하드웨어적으로 이루어지는 역다중화(demultiplexing), 패킷 파싱(packet parsing), 섹션 필터링(section filtering), 테이블 캐싱(table caching)등의 작업을 PC 상에서 에뮬레이션(emulation) 함으로써 MHP 미들웨어 개발자들이 손쉽게 자신들이 구현한 API를 테스트 할 수 있도록 도와준다.

본 논문의 구성은 다음과 같다. 먼저 2장에서는 MHP에 대한 전반적인 설명을 하고 방송 중인 트랜스포트 스트림이 셋탑박스 내에서 처리되는 과정을 보인다. 3장에서는 VSTB의 구조와 구성 요소들에 대해 설명하고 4장에서 구현상의 이슈에 대해서 논한다. 마지막으로 5장에서는 결론 및 향후 과제에 대하여 기술한다.

2. MHP(Multimedia Home Platform)

2.1 개요

MHP는 DVB(Digital Video Broadcasting) project에 의해 제안된 개방형 미들웨어 시스템으로써, 상호작용 가능한 응용 프로그램과 이 응용 프로그램이 실행되는 단말기를 연결하는 일반화된 인터페이스를 제공한다.[3]



[그림 1] MHP의 기본 아키텍처

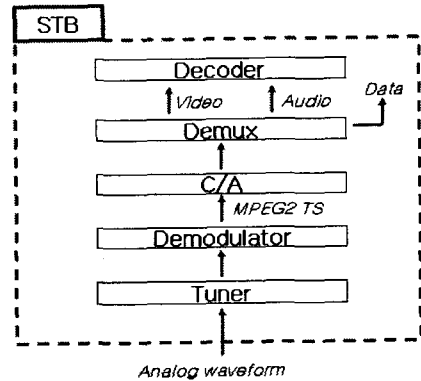
MHP의 아키텍처는 그림 1과 같이 크게 세 개의 계층으로 정의되어 있다.[4] 가장 하위에 자리하고 있는 자원(resources) 계층은 MPEG 처리 장치, I/O 장치, 그래픽 시스템과 같은 하드웨어 개체들을 포함한다. 시스템 소프트웨어(System Software) 계층은 응용 프로그램에게 추상화된 뷰(MHP API)를 제공함으로써 응용 프로그램이 직접적으로 자원에 접근하지 못하도록 제한한다. 응용 프로그램(Application)은 한가지 이상의 하드웨어 개체 상에서 동작하는 대화형 서비스를 소프트웨어로 구현한 것을 말한다.[4][5]

2.2 MHP 시스템에서의 TS 처리 과정

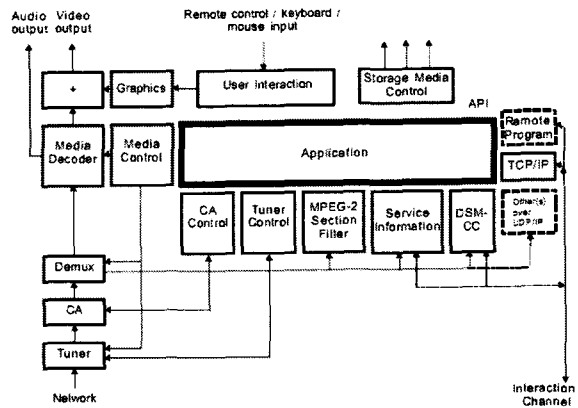
데이터 방송 환경에서는 기존의 영상, 음성 정보 외에 데이터 정보를 추가로 전송하는데 이를 위해 MPEG2 인코딩 방식이 이용된다. 인코딩이 된 각각의 스트림들은 다중화(multiplexing) 과정을 거쳐 수신단으로 보낼 하나의 비트 스트림을 생성하게 되는데 이를 트랜스포트 스트림(TS, Transport Stream)이라 한다. 수신단인 셋탑박스에서는 다시 일련의 과정을 통해 전달 받은 TS를 영상과 음성, 그리고 데이터로 분리한다.

그림 셋탑박스에 전달된 방송 신호가 실제로 어떤 흐름을 통해 본래의 미디어 스트림과 데이터 스트림으로 분리가 되는지 살펴보도록 하겠다. 먼저 튜너를 통해 해당하는 주파수 신호를 선택한다. 그렇게 나온 결과는 디모듈레이터(demodulator)를 거쳐 비트 스트림 형태인 TS로 변환된다. 이 TS는 사용자의 권한을 점검하는 수신 제한 시스템

(CAS, Conditional Access System)을 거친 후, 역다중화기(demultiplexer)를 통해 미디어 재생을 위한 영상, 음성 신호와 부가적인 데이터 신호로 분리 된다.[6] 이렇게 분리된 영상, 음성 신호는 적합한 디코더를 이용해 재생이 되고, 데이터 신호는 시스템 프로그램 또는 MHP 시스템을 기반으로 하는 응용 프로그램에 의해 적절히 처리 된다. [그림 2,3]



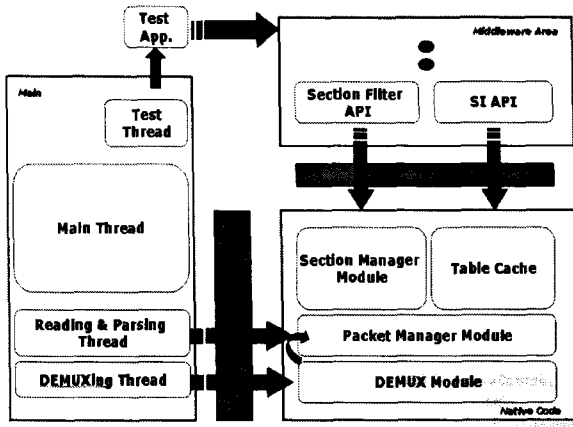
[그림 2] 수신단에서의 데이터 흐름도



[그림 3] MHP 시스템과 응용 프로그램 간의 인터페이스[3]

3. VSTB의 구조

본 논문에서 제안하는 VSTB는 실제로 방송이 이루어지는 환경을 염두하여 기존 셋탑박스 내에서 하드웨어적으로 처리되던 역다중화, 패킷 파싱, 섹션 필터링, SI 테이블 캐싱 등의 작업을 소프트웨어로 대체하는 것을 목적으로 하였다. 그림 4는 VSTB 컴포넌트들의 전체적인 구조를 나타내고 있다.



[그림 4] VSTB 컴포넌트 아키텍처

VSTB 는 다음과 같이 두 개의 주요 컴포넌트로 이루어진다.

### 3.1 main 컴포넌트

main 컴포넌트는 총 4개의 쓰레드(thread)로 구성된다. 첫 번째는 main UI 쓰레드로써 재생시킬 TS 파일을 추가하고, VSTB를 켜고 끄는 등의 유저 인터페이스를 담당한다. 그리고 demuxing 쓰레드와 reading&parsing 쓰레드가 존재하는데, 이들은 각각 native 컴포넌트 내의 demux 모듈과 packet manager 모듈을 작동시킨다. 이렇게 쓰레드를 따로 분리시킨 이유는 실시간으로 스트림이 흘러들어오는 셋탑박스 내의 환경을 최대한 유사하게 재현하기 위해서이다. 마지막으로 test 쓰레드는 사용자가 'test' 버튼을 눌렀을 시 미리 정의해놓은 테스트 응용 프로그램 코드를 수행하는 역할을 한다.

### 3.2 native 컴포넌트

native 컴포넌트는 셋탑박스 내에서 실제 하드웨어적으로 처리되는 작업들을 에뮬레이션 하는 역할을 맡는다. 먼저 demux 모듈은 선택된 TS로부터 데이터 패킷들만을 분리하여 packet manager 모듈에게 전달하는 역할을 한다. packet manager 모듈은 target pool 에 등록된 PID에 매칭되는 패킷들을 선별적으로 받아, PID별 패킷 리스트에 채우는 작업을 담당한다. 이렇게 관리되는 패킷 리스트들은 구문(syntax)에 명시된 패킷 개수에 도달하게 되면 section manager 모듈로 넘겨진다. 여기서 target pool 은 동적인 파싱 작업을 지원하기 위해 쓰이는 데이터 풀(data

pool)로써, VSTB 시스템이나 응용 어플리케이션에서는 언제든 필요할 때 이 곳에 파싱하길 원하는 PID를 등록할 수 있다. 현재는 native 컴포넌트가 실행되고 동시에 기본적으로 4개의 PSI 테이블(PAT, PMT, CAT, TSDT)과 9개의 DVB SI 테이블(NIT, SDT, BAT, EIT, RST, TDT, TOT, DIT, SIT), 추가로 AIT의 PID 가 등록되도록 구현되어 있다.[2][4][8] 다음으로 section manager 모듈은 packet manager 모듈로부터 넘겨받은 패킷 리스트를 해당 섹션의 구문에 맞게 파싱하여 섹션을 구성한다. 이 때, 파싱된 섹션이 위에 언급된 테이블 중의 하나이면 table cache 모듈에 갱신(update)을 요청한다. section manager 모듈로부터 요청을 받은 table cache 모듈은 기존의 테이블이 존재한다면 버전을 비교해 본 후, 업데이트 여부를 결정하게 된다. (버전을 확인하지 않는 테이블도 존재한다) 이렇게 저장된 테이블 정보는 MHP 미들웨어의 SI API를 이용하여 얻어올 수 있다.

## 4. VSTB의 구현

### 4.1 main 컴포넌트의 구현

main 컴포넌트는 Java 언어로 작성된 DVB-J application(Xlet)을 테스트 하기 용이하게끔 Java 코드로 작성하였다. 미들웨어 개발자는 test 버튼의 리스너(Listener) 메소드에 테스트 하려는 xlet 코드를 삽입함으로써 간단하게 해당 프로그램을 테스트 할 수 있다.

### 4.2 native 컴포넌트의 구현

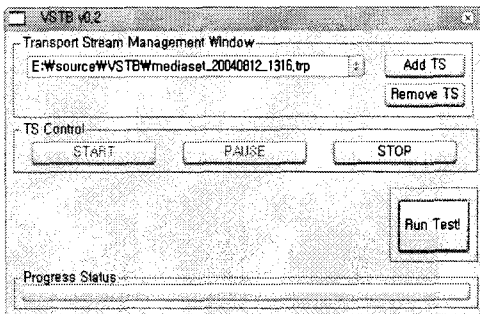
native 컴포넌트는 MFC를 이용하여 DLL 라이브러리 형태로 구현하였다. demux 모듈은 처음 실행될 때 읽어들이는 패킷이 비디오/오디오 스트림에 속하는지 혹은 데이터 스트림에 속하는지 구분할 수 없기 때문에, PMT를 이용하여 각 스트림의 PID 정보를 획득해야 한다. 따라서 PMT를 얻기 전까지는 demux 모듈에서 읽어들이는 모든 패킷들을 packet manager 모듈로 전송한다.[7]

한편, demux 모듈에서 패킷들을 전달하는 속도와 packet manager 모듈이 이를 받아 table cache 모듈까지 파싱하면서 처리해 나가는 속도의 격차가 크기 때문에, 패킷 유실(packet loss)이 많이 일어날 수 있다. 따라서 이를 막기 위한 효율적인 버퍼 전략이 필수적으로 요구된다. 본 논문에서는 그에 대한 해결책으로 demux 모듈과 packet manager 모듈 사이에 더블 버퍼(double buffer)를 삽입함으로써 패킷 유실을 최대한 방지하였다.

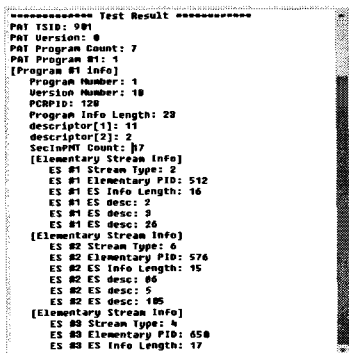
packet manager 모듈의 경우에는 addTargetPool 메서드를 두어 언제나 동적으로 파싱 대상을 선택할 수 있도록 구현하였다.

그리고 section manager 모듈과 table cache 모듈에서는 JNI 메소드를 제공함으로써 미들웨어 API에서 이들 메소드를 통해 원하는 섹션이나 테이블 정보를 얻을 수 있게 하였다.

마지막으로 멀티 쓰레딩 방식에서 일어날 수 있는 동기화 문제를 해결하기 위해 여러 쓰레드에서 동시에 접근할 수 있는 메소드에는 Win32 API에서 지원하는 EnterCriticalSection() 메소드와 LeaveCriticalSection() 메소드를 통해 동기화를 실현했다.[그림 5,6]



[그림 5] VSTB의 UI 화면



[그림 6] PAT를 추출하여 출력하는 테스트 프로그램

이선이 필요한 경우, 실시간 스트림 역다중화와, 파싱, 필터링 작업등이 지원되는 가상 셋탑박스 시스템이 유용한 역할을 할 수 있을 것으로 기대한다. 뿐만 아니라 다양한 파싱 전략과 캐싱 전략을 시뮬레이션하여 성능을 비교하는 작업에도 효율적으로 적용될 수 있을 것이다.

향후 과제로는 DSM-CC 섹션을 파싱할 수 있는 모듈을 추가하여 TS 내에 포함된 응용 프로그램을 실시간으로 실행할 수 있는 기능을 추가한다면 VSTB 시스템의 완성도를 높이는 데 크게 기여할 수 있으리라 생각한다. 이를 위해선 DSM-CC 섹션 파서 외에도, 이를 통해 추출한 응용 프로그램들의 라이프 싸이클(life cycle)을 관리해 줄 수 있는 어플리케이션 매니저(application manager)가 도입되어야 할 것이다. 또한 OCAP이나 ACAP 스펙에 명시되어 있는 테이블 파싱 작업을 추가하여 해당 미들웨어의 API를 테스트 해보는 용도로도 확장할 수 있을 것이다.

참 고 문 헌

- [1] openMHP, <http://www.openmhp.org>
- [2] Steven Morris et al., "Interactive TV Standards", Focal Press, 2005
- [3] DVB-MHP, <http://www.mhp.org>
- [4] MHP Specification 1.0.3: ETSI TS 101 812 V1.3.1, 2003-06
- [5] 서울대학교 컴퓨터신기술공동연구소, "대화형 멀티미디어 TV 방송의 기본기술 개발", 2000
- [6] 김도형, "똑똑한 TV '인터랙티브 DTV' 세상", 월간 마이크로소프트, 2002.1
- [7] 김무한, "MPEG-2 Transport Stream의 Software Demultiplexer 개발", 호서대학교 정보통신공학과, 2002
- [8] Specification for SI in DVB systems: ETSI EN 300 468 V.1.7.1, 2005-12

5. 결론 및 향후과제

본 논문에서는 윈도우즈 기반의 PC 상에서 가상의 셋탑박스 환경을 에뮬레이션 하기 위한 VSTB 시스템을 제안하고 이를 구현하였다. 셋탑박스가 없는 환경이나 간단한 시뮬레