

# TV 동영상 화면 캡처 서버-클라이언트 애플리케이션 구현

서강대학교 영상대학원 미디어공학과 디지털 방송랩  
박연선, 김정환, 남윤석, 정문열

## Server-Client applications for Capturing and Sending images in TV video

Digital Broadcasting Lab, Dept of Media Technology of GSMC at Sogang Univ.  
Youn-Sun Park, Jung-Hwan Kim, Yoon-Suk Nam, Moon-Ryul Jung

### 요 약

본 논문은 시청자가 TV를 시청하다가 마음에 드는 화면이 있으면, 이를 캡처해서 친구들에게 메시지로 보내는 서비스를 구현하는 방법을 기술한다. 이를 위해서는 셋톱박스에서 실행되는 Xlet 애플리케이션에서 비디오 스트림의 특정 화면을 캡처할 수 있어야 하는데, 현재 미들웨어는 이런 기능을 제공하고 있지 않다. 따라서, 본 논문에서는 Xlet application 에서 시청자가 지정한 화면의 시점값을 방송국 서버로 전송하고, 서버에서 해당 화면을 캡처하여 이를 시청자가 원하는 수신자에게 보내주는 방법을 사용한다. 미들웨어 API 표준에서 현재의 NPT (Normal Play Time) 을 알려주는 API 를 제공하므로, 애플리케이션에서 시청자가 원하는 시점의 NPT 를 구해 이를 방송국 서버로 보내고, 서버에서는 이 NPT 에 해당되는 동영상의 미디어 타임을 구해, 그 시간에 해당되는 프레임을 계산하여 화면을 캡처한다. 본 기술을 활용하면 서버에서 캡처된 화면을 Mobile, Web 등의 매체에도 전송할 수 있으므로 유무선통합, 방통융합형 서비스로 확장해 나갈 수 있다.

### I. 서 론

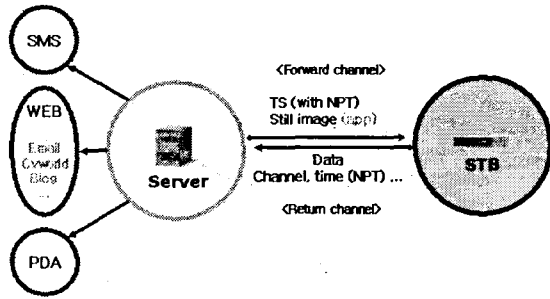
디지털 방송이 제공되기 시작하면서 다채널, 고화질 서비스 외에 다양한 형태의 데이터를 비디오/오디오와 함께 송출하는 데이터 방송이 가능해 졌다. 여기서 데이터란, 화면상에 표시되는 텍스트, 이미지뿐만 아니라 셋탑박스 상에서 실행 가능한 어플리케이션 프로그램을 총칭한다. 이를 바탕으로 디지털 방송 환경에서 양방향 서비스를 제공하기 위한 다양한 서비스 모델이 등장하였지만, 현실적으로 이를 지원하기 위한 하드웨어 및 소프트웨어의 개발이 부족한 실정이라, 시청자의 눈과 귀를 사로잡을 만한 서비스(어플리케이션)의 개발이 원활하게 이루어지지 못하고 있다. 특히, 이러한 데이터 방송 어플리케이션의 실행 환경을 지원하는 셋탑 박스의 성능이 PC환경에

비하여 상당히 제한적이기 때문에, 기존의 PC 환경에서의 동영상 화면 캡처 방식을 그대로 셋탑 박스 상에서 실행하기 어렵다.

또한, 동영상 화면을 직접 캡처하는 API나 캡처하고자 하는 동영상 화면의 시간값으로 쉽게 활용가능한 PTS, STC 등과 같은 값을 얻는 API 등을 미들웨어에서 제공하고 있지 않으므로, 이러한 기능을 활용한 서비스를 개발하는 것이 쉽지 않다.[4]

따라서, 본 논문에서는 동영상 캡처 기능을 클라이언트(셋탑박스)가 아닌 서버(방송국 송출 서버) 단에서 담당하도록 하는 대신, 아래 그림 1과 같이 시청자가 임의의 시점에 해당하는 방송 영상 화면을 캡처 하기 원할 경우, 클라이언트 애플리케이션에서 캡처하고자 하는 화면의 시간값을 리턴채널을 통해 서버로 전송하여, 서버에서

는 전송된 시간값에 해당하는 동영상의 프레임을 찾아내어 화면 캡처 기능을 수행 하는 방안을 제시할 것이다.



<그림 1. 클라이언트-서버 어플리케이션 개요도>

비디오와 어플리케이션을 동기화 시키기 위해서 일반적으로 스트림 이벤트를 사용하는데, 이는 스트림 이벤트를 방송 스트림에 포함시켜 전송하여, 이를 셋탑 박스상에서 해당 스트림 이벤트에 명시된 시각에 이벤트를 발생시키는 방법을 의미한다. 따라서, 특정 방송스트림(비디오)과 스트림 이벤트 간의 동기화를 설정해야 한다. 이때 스트림 이벤트의 동기화 시각은 절대적인 시간이 아닌 상대적인 시간, 즉 해당 프로그램이 시작된 후 일정 시간이 경과 한 후 이벤트가 발생 하도록 정해져야 한다. 이는 특정 방송 프로그램이 시작 되는 시간이 항상 일정하지 않기 때문이다. 따라서, 데이터 방송 스트림 이벤트의 이벤트 처리 시각은 NPT(Normal Play Time)를 사용하는 것이 일반적이다. NPT는 앞서 말한 바와 같이, 특정 프로그램(이벤트)마다 존재하는 이벤트 내부의 국지시간을 나타낸다.

즉, 프로그램 전송 스트림에 NPT 참조값을 삽입하여 이를 송출하면, 셋탑박스 상에서 이벤트(사용자의 캡처 요청) 발생시 이 시간 값 (현재의 NPT 값)을 getNPT()[7] 라는 API를 통해 해당 어플리케이션이 알아내게 된다. 이 NPT 값을 리턴 채널을 통해 방송국으로 전송하면, 이를 서버에서 미디어 타임으로 변환하여 캡처하고자 하는 화면을 얻어낼 수 있다.

이 어플리케이션의 기능을 활용하면 시청자가 원하는 임의의 방송 화면을 캡처 하여, 해당 화면을 핸드폰, PDA 등의 모바일 기기나 개인 홈페이지에 전송 할 수 있을 것

이다, 또한, 클라이언트가 아닌 서버상에서 캡처 기능이 이루어지기 때문에, 셋탑박스 상의 자원 점유 문제나, 캡처 화면에 대한 저작권 및 과금 문제의 해결 역시 용이하리라 기대된다.

본 논문은 2장에서는 클라이언트의 디지털/데이터 방송 규약, 서버상의 DirectX를 이용한 캡처 기능에 대하여 설명하고, 3장에서는 2장의 내용을 바탕으로 한 클라이언트와 서버 어플리케이션으로 구성된 동영상 캡처 어플리케이션을 설명하고, 4장에서 결론을 맺는다.

## II. 비디오 캡처 서비스를 위한 디지털/데이터 방송 기술

본 장에서는 본 논문을 이해하는데 필요한 디지털/데이터 방송의 기본 개념, 스트림 이벤트, 이벤트 처리시각의 기준인 NPT, 데이터 방송 어플리케이션 Java Xlet과 이를 바탕으로 제작하는 연동형 어플리케이션 구조를 설명하고, DirectX를 이용한 서버상에서의 동영상 캡처 방법에 대해 설명한다.

### 1. 서비스와 이벤트, 스트림 이벤트

디지털 방송에서 방송의 기본단위를 서비스(Service)라 하며, 하나의 서비스에는 보통 하나의 채널에서 방송하는 방송물 전체를 가리킨다. 일반적으로 하나의 서비스에는 복수개의 스트림 (오디오/비디오/데이터 스트림)을 포함하고 있으며, 하나의 데이터 방송 어플리케이션과 이 어플리케이션에서 사용하는 데이터 또한 데이터 스트림을 통해 전송 하고 있는 서비스의 구성 요소라 할 수 있다. 이벤트는 우리가 일반적으로 하나의 방송 프로그램으로 취급하는 9시 뉴스, 축구경기 전반전등의 프로그램을 의미한다. 디지털 방송에서는 복수의 서비스를 하나의 전송 스트림 (Transport Stream, TS)에 실어 보내며, 보통 복수의 전송 스트림을 송출 한다.

디지털 방송에서 제공하는 스트림 이벤트는 두 가지 종류로 나눌 수 있는데, 하나는 "do it now" 스트림 이벤트 이고, 다른 하나는 "scheduled" 스트림 이벤트이다.[7] "do it now" 스트림 이벤트는 전송 스트림 내에 이벤트가 도착하자마자 어플리케이션이 이벤트를 처리하는 방법이고, "scheduled" 스트림 이벤트는 스트림 이벤트의 처리시각(EventNPT)을 명시하고 스트림 이벤트 서

술자(stream event descriptor)에 삽입하여 어플리케이션이 그 시각을 읽어 현재의 NPT와 비교하여 같으면 이벤트를 처리하는 방법이다. “do it now” 스트림 이벤트가 별도의 시간 계산이 필요 없는 대신 원하는 시간에 이벤트를 발생시키기 어렵다. 그에 반해 “scheduled” 이벤트의 경우 NPT를 이용하여 명시된 시각에 이벤트를 발생시킬 수 있어, 비디오와 정확하게 동기화된 시각에 어플리케이션의 실행이 가능하다.

## 2. NPT (Normal Play Time)

NPT는 “9시 뉴스” 처럼 특정시각에 시작하여 특정시각에 끝나는 단위 프로그램내에서의 시간이다. 즉, 하나의 방송 프로그램(이벤트)의 시작 시간을 0으로 세팅하고 90Khz 단위로 연속적으로 증가하는 내부의 진행 시간이다. NPT는 STC와 달리 특정 이벤트의 일부가 추가 혹은 삭제되더라도 그 시간 값이 변하지 않는다. (STC와 NPT의 관계는 참조 논문 [1],[2],[3]에 자세히 설명되어 있다) 결국 NPT값은 언제나 해당 프로그램내의 동일한 위치를 가리키므로 NPT는 스트림 이벤트 처리시각을 표시하는 시간 기준으로 적합하다.

현재의 NPT 값을 계산하는 아래의 식은 MPEG-2 DSMCC 스펙[6]에 명시되어 있으며, 그 값을 계산하기 위해서는 STCR, NPTR 참조 시간값들이 필요한데 이 시간값들은 NPT Reference Descriptor에 명시하여 이 서술자가 포함된 스트림으로부터 읽어들이게 된다. 그러나, 현재 이러한 스트림을 생성하기 어렵고, 스트림을 생성하였다 하더라도 현재 구현된 미들웨어에서 이를 바르게 구현해 놓지 않았으므로, 정확한 NPT 값을 계산하기 위해서는 어플리케이션이 STC 값을 알아야 한다. 하지만, 어플리케이션은 STC 값을 읽어낼 방법이 없으므로, 참고 논문 [1], [2]를 통하여 임의적으로 스트림을 생성하는 생성기와 현재의 STC 값을 구하는 방법을 통해 현재의 NPT를 재구성하는 임의의 getNPT 함수를 이용하여 구할 수 있다. 자세한 방법은 두 논문을 참조하기 바라며, 본 논문은 NPT를 생성하고 재구성하는 것이 주제가 아니므로 미들웨어가 정상적으로 구현되어 있고, NPT 참조 서술자를 포함한 정상적인 스트림이 존재한다는 가정하에 서술하도록 한다.

$$NPT = (STC/300 - STC\_Reference) + NPT\_Reference$$

본 논문에서는 화면을 캡처하고자 하는 시간에 어플리케이션에서 getNPT() 함수를 호출하는 시간을 이벤트 발생시점이라 가정하고, 해당 시간의 현재 NPT 값을 얻어 이를 서버상의 동영상 캡처에 활용하고자 한다.

## 3. Java Xlet을 이용한 연동형 어플리케이션

플랫폼별로 데이터 방송 규약은 ACAP,OCAP,DVB-MHP로 나뉘어져 있지만, 데이터 방송 규약을 실제로 구현하는 데이터 방송용 어플리케이션은 Java 언어의 Xlet을 사용하고 있다. 그리고, 연동형 어플리케이션은 현재 방송중인 채널의 비디오 혹은 오디오에 연동하여 실행되는 어플리케이션을 의미한다. 본 논문에서 제작할 클라이언트(셋탑박스)용 데이터 방송 어플리케이션 역시 특정 비디오 방영시 TV를 보면서 해당 화면을 캡처하는 기능을 가진 연동형 어플리케이션의 형태라 할 수 있으며, 직접 화면을 캡처하는 대신 캡처하고자 하는 시간의 NPT 값을 getNPT() 함수를 통해 얻어내어 리턴 채널을 통해 서버로 전송하는 Java Xlet 이다.

## 4. DirectX를 이용한 서버상에서의 동영상 캡처

서버 어플리케이션은 클라이언트에서 보내준 시간값(NPT 값)을 아래의 MPEG-2 DSMCC 스펙[6]에 명시된 계산식을 이용하여 미디어 타임으로 계산한 후 해당 프레임을 구하여 사용자가 원하는 시점의 장면을 캡처한 뒤 이를 클라이언트(셋탑박스) 어플리케이션에서 화면에 표현할 수 있는 이미지로 전환하여 이를 다시 전송해주거나, 원하는 플랫폼(휴대폰,PDA,웹 등)으로 보내는 역할을 수행한다.

$$NPT\_seconds = NPT / (System\_Clock\_Frequency / 300)$$

$$NPT\_microseconds = ((NPT \times 1000000) / (System\_Clock\_Frequency / 300)) - (NPT\_seconds \times 1000000)$$

여기서, 서버 어플리케이션의 핵심기능인 TS의 특정 채널의 특정 시점의 영상을 캡처 하는 기능을 수행하기 위해서 마이크로소프트사의 멀티미디어 API인 DirectX를 사용한다. 즉, DirectX의 라이브러리 중 하나인 DirectShow를 이용하여 클라이언트에서 받은 NPT 값을 위의 식을 통해 미디어 타임으로 계산한 후, 아래의 식을

통해 미디어 타임에 해당하는 프레임 수를 계산하여 시청자가 원하는 특정 채널의 동영상에서 해당 프레임의 화면을 캡처한다.

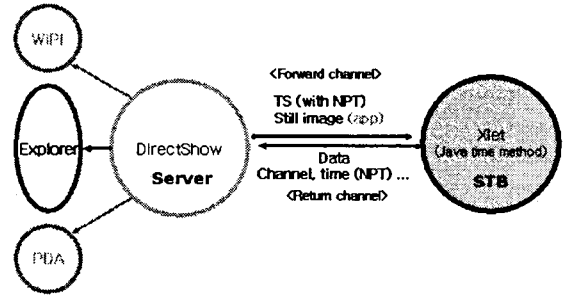
$$\text{Frames}[i] = \text{NPT\_seconds} \times \text{Frames/sec}$$

본 식에서  $\text{Frames}[i]$  는 클라이언트에서 캡처하고자 하는 시간의 현재 NPT 값  $i$ 에 해당하는 프레임 수로 해당 동영상의 시작 프레임을 0이라고 가정했을 때  $\text{Frames}[i]$  가 300이라면, 300번째 프레임을 캡처하면 된다.  $\text{NPT\_seconds}$ 는 클라이언트에서 `getNPT` 함수를 통해 캡처하고자 하는 시점의 현재 NPT 값을 미디어 타임 단위인 초단위로 계산한 값이며,  $\text{Frames/sec}$ 은 해당 동영상의  $\text{Frames/sec}$ 의 평균값을 디렉트쇼에서 제공하는 API를 이용하여 얻어낸 값이다. 본 논문은 이  $\text{Frames/sec}$ 의 평균값을 오차 범위 허용치라 가정하고 사용하도록 한다.

### III. 동영상 화면 캡처 서버-클라이언트 어플리케이션 구현

클라이언트 어플리케이션은 셋탑박스의 미들웨어상에서 실행되는 Java 언어로 짜여진 Xlet의 형태를 띤 데이터 방송 어플리케이션이다. 클라이언트 어플리케이션은 시청자의 입력에 따라 특정 화면에서 리모콘상의 미리 설정해둔 특정 버튼을 눌러 해당 화면을 캡처할 것을 요청하면, 어플리케이션은 이때의 현재 NPT 값을 `getNPT()` 함수를 통해 알아내고, 해당 비디오가 담겨있는 TS와 서비스의 정보를 찾아 내어, 이 값들을 그림 2에서 보듯이 리턴 채널을 통해 서버에게 송신하게 된다.

앞에서 설명한 “Scheduled” 이벤트 처리 방법에서 `EventNPT`를 현재 버튼을 누른 시간의 NPT로 보고, 버튼을 누른 행위를 하나의 이벤트처럼 처리하는 개념이다. 원래 `EventNPT`는 서버에서 클라이언트의 어플리케이션이 해당 시간에 이벤트를 발생시키는 시간을 계산하여 클라이언트로 그 정보를 전달하여 동영상과 이벤트의 동기화를 위해 사용되는 것이나, 본 논문에서는 반대로 이를 응용하여 클라이언트에서 이벤트 발생시점을 얻은 후 이를 서버로 전송하여 서버측의 동영상에서 해당 시간의 화면을 찾아내도록 구현한 것이다,



<그림 2 클라이언트-서버 어플리케이션 구조도>

클라이언트(셋탑박스)의 동기화 정보를 받아들인 서버는 시청자의 캡처 요구 화면이 어떤 채널의 어떤 서비스에 속한 동영상인지 찾아낸 뒤, 해당 동영상으로부터 사용자가 요청한 캡처 화면이 담긴 시각, 즉 NPT값을 통해 알아낸 프로그램 시작 후 상대적인 시간을 위에서 언급한 방법에 따라 해당 동영상의 프레임 수를 구한 후 `DirectShow` API를 사용하여 해당 프레임을 캡처한다.

서버는 캡처한 스틸 이미지를 사용자의 요구에 따라 셋탑박스에 재전송하여 사용자가 이를 감상할 수 있도록 하거나, 혹은 사용자의 휴대폰, PDA와 같은 모바일 기기로 전송 또는 인터넷을 비롯한 웹 영역에 올릴 수 있도록 재가공하여 전송하는 역할을 수행할 수 있다.

### IV. 결론 및 향후 과제

본 논문은 셋탑박스에서 실행되는 xlet 어플리케이션에서 비디오 스트림의 특정 화면을 캡처하는 기능이 현재 미들웨어에서 제공되지 않으므로, xlet에서는 미디어 타임으로 전환이 가능한 NPT 값을 얻어 이를 서버로 전송하여 캡처하는 기능을 서버에서 구현하도록 하였다.

MPEG-2 비디오 영상은 다른 포맷의 영상들과는 달리 프레임 단위로 계산되어지지 않으므로 본 논문의 서버 구현시 초당 프레임 수를 동영상의 평균치로 가정하고 계산하였으므로 약간의 오차를 허용하고 있다.

따라서, 향후 이러한 문제를 해결하기 위해 MPEG-2 영상을 프레임 단위로 계산 가능한 영상으로 다시 인코딩한 후 해당 시간의 프레임을 얻어내는 방법도 연구해 볼 수

있겠다. 혹은, 미들웨어에서 MPEG-2 영상의 PTS 값을 직접 얻어낼 수 있는 API를 제공해준다면 이를 활용하여 직접 MPEG-2 영상에서 PTS를 검색하여 해당 화면을 캡처하는 방법도 가능하리라 생각된다.

## 참고문헌

- [1] 정문열, 백두원. “연동형 데이터 방송 애플리케이션의 구조”. 방송공학회논문지. 제9권 제1호. 2004.
- [2] 정문열, 김용한, 백두원. “동기화된 데이터방송을 위한 근사적인 NPT 재구성 기법”. 방송공학회논문지. 제9권 제1호. 2004.
- [3] 김세훈, 동기화된 데이터 방송 애플리케이션 제작을 위한 스트림 생성기 개발, 석사논문, 2003.
- [4] 박연선, 디지털 방송에서 실시간 대화형 이미지 갤러리 구현, 석사논문, 2001
- [5] ISO/IEC 13818-1 Generic Coding of Moving Picture and Associated Audio : Systems.
- [6] ISO/IEC 13818-6 Generic Coding of Moving Picture and Associated Audio : Digital Storage Media Command and Control.
- [7] ETSI ES 201 812 v1.1.1 : Digital Video Broadcasting(DVB); Multimedia Home Platform(MHP) Specification v1.0.3
- [8] OC-SP-OCAP1.0-I13-041215, OCAP 1.0 Profile, 2004