

논문-04-09-1-09

동기화된 데이터방송을 위한 근사적인 NPT 재구성 기법

정 문 열*, 김 용 한**, 백 두 원***

An Approximate Reconstruction of NPT for Synchronized Data Broadcasting

Moon-Ryul Jung*, Yong Han Kim** and Doowon Paik***

요 약

DVB-MHP에서는 NPT(normal play time)를 스트림 이벤트의 시각으로 쓰기를 권하고 있다. NPT는 특정 이벤트(프로그램) 내의 국지시간이다. 현재 상용으로 나와 있는 전송스트림(TS) 생성기와 TV 미들웨어는 아직 NPT를 지원하지 못하고 있다. 특히 전송스트림 생성기가 셋톱박스에서 NPT를 재구성하는데 필요한 NPT참조서술자를 생성하지 않고 있다. 이로 인해 PP가 연동형 애플리케이션(Xlet)의 아이디어를 실험하는 것이 불가능하다. 이에 우리는 TS에 NPT 참조서술자를 삽입하는 스트림 생성기와 NPT를 근사적으로 재구성하는 MyGetNPT API를 구현하였는데, 본 논문은 그 방법을 기술한다. NPT 재구성 API를 구현하기 위해서는 STC(system time clock) 값을 알 필요가 있으나, Xlet에서는 STC를 읽을 수 없다. 따라서 본 연구에서는 TS를 통해 전송되는 PCR(program clock reference) 과 Java 시스템 타임을 이용하여 STC를 근사적으로 계산하는 방법을 제안한다. 이 방법에서 전송스트림 생성기는 이미 존재하는 TS로부터 PCR 들을 추출한 후, 이를 TS의 null 패킷에 Xlet에서 읽을 수 있는 MPEG 섹션의 형태로 삽입한다. 이때, PCR이 TS 내의 원래 위치에서 이동하여 다른 위치에 삽입되므로, PCR 값은 TS내의 원래 위치와 새로운 위치간의 시간 차이를 고려하여 수정한다. 구현한 TS 생성기와 MyGetNPT API를 이용하여 그래픽 이미지의 디스플레이가 목적인 스트림 이벤트를 가진 연동형 애플리케이션을 구현하여 실험을 하였다. 그 결과 그래픽 이미지들이 원래 의도된 시점으로부터 240ms 이내에 비디오와 동기화되는 것을 확인하였다. 이 시간은 기존의 연구에서 발견된 그래픽 이미지와 비디오간의 동기화 오차 허용 한계이다.

Abstract

DVB-MHP recommends that NPT(normal play time) be used as the times of stream events. NPT is the local time within an event(TV program). But we found that commercial transport stream (TS) generators and middlewares for DVB-MHP settop boxes are not ready to support the use of NPT by applications. In particular, TS generators do not create NPT reference descriptors needed to reconstruct NPT at the TV receiver. This situation is undesirable because program providers cannot experiment with the idea of synchronized applications. So we have implemented a TS generator that inserts NPT reference descriptors to TS and MyGetNPT API to approximately reconstruct NPT. STC (system time clock) is needed to reconstruct NPT, but Xlets are not allowed to read it. So, we approximate STC by using PCR (program clock reference) and the Java system time. In this method, the stream generator extracts PCRs from an existing TS and inserts them into null TS packets in the form of MPEG sections, which can be read by Xlets. Because PCRs are displaced into new positions in TS, their values should be adjusted based on the time intervals between the original positions and the new positions. We implemented a synchronized application by using our TS generator and MyGetNPT API, where the task of stream events are to display graphic images. We found that graphic images are displayed within 240 ms from their intended time, where 240ms is a human tolerance for the synchronization skew between graphic image and video.

Keywords : Synchronized applications, stream events, NPT, NPT Reference Descriptors, STC, PCR, MPEG sections.

I. 서론

데이터 방송은 데이터를 비디오와 함께 방송하여 시청자에게 전달하는 것을 의미하는데, 여기서 데이터라 함은 텍스트나 이미지 파일과 같은 자료 외에 TV 수신기에서 실행 가능한(executable) 애플리케이션 프로그램도 포함된다. 비디오내의 특정시간에 특정행위를 하는 데이터 방송 애플리케이션을 연동형 애플리케이션이라 하는데, 이를 통해 새로운 형식의 방송물제작이 가능하다. 데이터 방송 표준으로는 북미의 ATSC-DASE^[1]와 OCAP^[2], 유럽의 DVB-MHP^[3] 등이 있다. 데이터 방송 애플리케이션에 대한 포괄적이고 심도 있는 설명은 문헌 [4], [8]을 참고하라.

우리나라의 디지털 위성방송 표준인 DVB-MHP^[3]은 비디오와 애플리케이션을 동기화시키기 위해 스트림 이벤트를 사용하도록 권하고 있다. 이 방법에서는 스트림 이벤트를 방송 중간에 전송스트림에 포함시켜 전송하고 수신기의 애플리케이션이 이벤트에 명시된 시각에 이 이벤트의 의도대로 특정행위를 수행한다^{[3][8]}. 연동형 애플리케이션을 제작하려면 프로그램 제작자(program provider, PP)가 해당 방송 프로그램과 스트림 이벤트간의 동기화를 설정해야 한다. 이때 스트림 이벤트의 동기화 시각은 해당 프로그램이 방송되는 시각과 관계없이 설정되어야 한다. 해당 프로그램이 언제 방송될지 모르기 때문이다.

따라서 DVB-MHP는 스트림 이벤트는 이벤트 처리시각으로 NPT(Normal Play Time)을 사용하도록 규정하고 있다. NPT가 이벤트(프로그램)마다 존재하는 이벤트 내부의 국지시간이다. NPT를 사용하는 스트림 이벤트를 이용하여 연동형 애플리케이션을 제작하기 위해서는 Xlet에서 해당 이벤트내의 NPT를 알아야 한다. 이를 위해 DVB-MHP는 getNPT()라는 API를 정의하고 있다. 하지만 현재 상용화되어 있는 전송스트림 다중화기와 수신기 미들웨어는 데이터 방송 표준에 규정되어 있는 NPT 기능을 사용할 수 있도록 구성되어 있지 않다. 현재 국내에서 상용으로 쓰고 있는 전송 스트림 생성기는 애플리케이션에서 NPT를 재구성하여 사용하는데 필요한 NPT 참조서술자 스트림을 생성하

지 않는다. 그리고 현재 스카이 라이프에서 상용으로 사용하고 있는 수신기 미들웨어도 getNPT()를 구현했다고는 하나 NPT참조서술자 스트림을 생성하는 TS 생성기가 없는 상황에서 이 API가 충분히 테스트되었는지도 미지수이다.

그러나 연동형 데이터 방송을 준비하는 PP들의 스튜디오에서는 연동형 애플리케이션을 실험적으로 제작할 필요가 있다. 이를 위해서는 NPT 참조서술자 스트림을 생성하는 것이 필요하고, 애플리케이션이 NPT를 사용할 수 있도록 NPT참조 서술자를 해석하여 현재 순간의 NPT를 재구성하는 것이 필요하다. 본 논문은 이 NPT 재구성 문제를 해결하는 방법을 기술한다. 연동형 애플리케이션 실험을 하려면 NPT 참조서술자 스트림과 관련 스트림들을 생성하는 스트림 생성기가 필요하나, 본 논문은 이에 대해서는 별도로 다루지 않는다. 이것은 문헌 [9]를 참조하기 바란다. 본 논문은 명시된 시각에 이벤트를 처리하는 Xlet 애플리케이션을 구현하여 본 논문의 방법을 검증하였다. 비디오와 동기화된 Xlet 애플리케이션을 구현하는 방법은 문헌 [11]에 기술되어 있다.

본 논문의 구성은 다음과 같다. 제 2장에서는 본 논문의 이해를 돕기 위해 관련된 디지털/데이터방송 기술을 간략히 설명하고 제 3장에서는 수신기에서 NPT를 재구성하는 방법을 기술하며 제 4장에서는 NPT를 재구성하는 방법의 검증을 위한 실험방법 및 결과를 기술하고 제 5장에서 결론을 맺는다.

II. 디지털/데이터 방송 개요

본 장에서는 본 논문을 이해하는데 필요한 디지털/데이터 방송의 기본 개념, 스트림 이벤트와 이벤트 처리시각의 기준시간이 되는 NPT를 설명한다.

1. 서비스, 이벤트, 애플리케이션

디지털 방송에서는 방송의 기본단위를 서비스(Service)라고 한다. 하나의 서비스는 보통 하나의 채널에서 방송하는 방송물 전체를 가리킨다. 하나의 서비스는 보통 여러 개의 성분 스트림(Elementary Stream, 비디오/오디오/데이터 스트림)을 포함한다^{[3][4][5]}. 애플리케이션과 이것이 사용하는 데이터 스트림도 하나의 서비스의 구성요소이다. 그리고 우

* 서강대학교 영상대학원 미디어공학과
Dept. of Media Technology, Sogang University

** 서울시립대학교 공과대학 전기전자컴퓨터공학부
Dept. of Electrical and Computer Eng., University of Seoul

*** 송실대학교 미디어학부
School of Media, Soongsil University

리가 흔히 말하는 방송 프로그램 (9시 뉴스, 한일전의 전반전)을 이벤트(Event)라고 한다. 특정 프로그램 중에 삽입되는 광고도 하나의 이벤트이다. 서비스는 정해진 시작과 끝이 따로 없다. 서비스 안에 포함된 특정 이벤트(프로그램, 예: 9시 뉴스, 8시 드라마)는 시작과 끝이 있다. 이벤트는 서비스를 시간대별로 구분한 것이므로 한 이벤트는 다수의 성분 스트림을 포함할 수 있다. "이벤트" 는 "방송 프로그램" 이라는 의미 외에 보다 일반적으로 "사건" 이라는 의미로 쓰인다. 본 논문에서도 두 가지 의미로 사용하는데, "이벤트"가 프로그램을 가리킨다는 것을 확실히 하고자 할 때는 "이벤트 (프로그램)" 이라고 표기한다. 디지털 방송에서는 복수의 서비스를 하나의 전송스트림 (Transport Stream, TS)에 실어 보내며, 보통 복수의 전송 스트림을 방송한다.

2. 스트림 이벤트 (Stream Event)

방송국에서 TS를 통해 스트림 이벤트를 전송하면 애플리케이션이 이를 처리하게 함으로써 비디오 방송과 애플리케이션을 동기화할 수 있다. DVB-MHP에서는 스트림 이벤트를 이용하는 두 가지 방법을 제시하였는데 하나는 "do it now" 스트림 이벤트이고 다른 하나는 "scheduled" 스트림 이벤트 이다[3]. "do it now" 스트림 이벤트는 전송 스트림 내에 이벤트가 도착하자마자 애플리케이션이 이벤트를 처리하는 방법인데, 별도로 시간계산을 하지 않고 이벤트가 발생하였을 때 마다 처리를 한다는 장점이 있지만 원하는 시간에 이벤트를 발생시키기 어렵다. 반면 "scheduled" 스트림 이벤트는 스트림 이벤트의 처리시간 (NPT)을 명시하고 애플리케이션이 그 시각에 이벤트의 데이터를 처리하는 방법인데, 명시된 시각에 이벤트를 발생시킬 수 있어 비디오와 정확하게 동기화된 시각에 애플리케이션 실행을 가능하게 한다.

3. NPT (Normal Play Time)

NPT는 "9시 뉴스"처럼 특정시각에 시작하여 특정시각에 끝나는 단위 프로그램 내에서의 시간이다. 여기서 단위 프로그램이라고 하는 것은 DVB에서 "event" 라고 부르는 것이다. 이벤트는 하나의 채널에서 방송하는 서비스를 시간대로 분할하여 특정시간부터 특정시간까지 진행되는 하나의 단위 프로그램을 가리킨다. NPT는 바로 이 이

벤트내부의 진행시간이다. NPT는 프로그램의 시작 시각에 시작하여 90KHz로 프로그램 내에서 연속적으로 증가한다.

NPT는 그림 1에서 보듯이 해당 이벤트 (프로그램)의 일부가 편집되어 없어지거나, 중간에 광고 등의 다른 이벤트(프로그램)가 삽입되는 경우에도 그 시간 값이 변하지 않는다. NPT값은 언제나 해당 프로그램내의 동일한 위치를 가리키므로 NPT는 스트림 이벤트 처리시각을 표시하는 시간기준으로 적합하다. 그림 1은 전역시간이라고 볼 수 있는 STC와 국지시간이라고 볼 수 있는 NPT의 차이를 설명하고 있다. 그림 1 (a)은 이벤트(프로그램) P와 P를 구성하고 있는 P1, P2, P3의 시작과 끝점에서의 NPT를 나타낸 그림이다. 예를 들어 P1의 시작점에서의 NPT값은 0 이고 P1이 끝나는 시점에서의 NPT값은 2 이다. 그림 1 (b)은 이벤트(프로그램) P의 사이에 다른 프로그램인 E와 F가 삽입되어 기존의 서비스가 변경된 것을 보여주고 있다. 이 경우에도 P의 NPT는 E와 F가 삽입됨에 관계없이 전과 동일한 값을 가짐을 보여준다. 반면

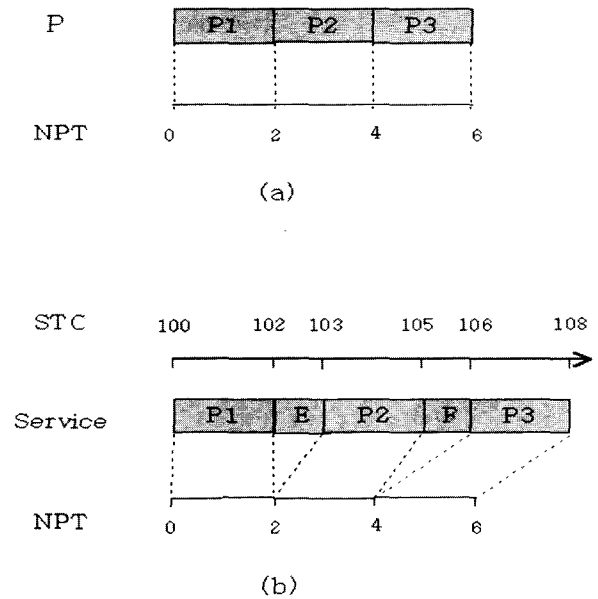


그림 1. NPT는 이벤트 (단위 프로그램)의 국지시간이고 STC 는 시작과 끝이 없는 특정 서비스의 전역시간이다. 그림 (b)에서 이벤트 P안에 다른 이벤트 E, F 가 삽입되어도 이벤트 P의 시간인 NPT는 그림 (a) 와 차이가 없으나, STC는 E, F 의 진행시간까지 고려하여 진행된다.

Fig. 1. NPT(Normal Play Time)

STC(System Time Clock)는 E와 F가 삽입된 서비스 내에서 연속적으로 증가한다. 비디오와 애플리케이션의 동기화는 이벤트(프로그램) 내에서 이루어져야 하므로 동기화의 기준시간으로 STC가 아닌 NPT를 이용한다. STC의 주파수는 27 MHz이고 NPT의 주파수는 90 KHz 인데 표시를 용이하게 하기위해 그림 1은 이들을 스케일된 값으로 표현하였다.

4. NPT 참조서술자 (NPT Reference Descriptor)

DVB-MHP에서는 `getNPT()`라는 API를 정의하고 있는데 이것은 특정 이벤트의 현재 순간의 NPT를 계산하는 메소드이다. 이 메소드가 제대로 작동을 하려면 방송국에서 NPT를 샘플링 한 값(NPT Reference, NPTR)과 STC를 샘플링 한 값(STC Reference, STCR)을 연결시킨 NPT 참조서술자(NPT reference descriptor)를 TS에 삽입해서 보내야 한다. NPT참조 서술자는 DSMCC 규약에서 정의하고 있는 스트림서술자의 일종이다. NPT 참조서술자를 1초에 1번 이상 보내므로 이것도 하나의 스트림을 구성한다^[3]. NPT 참조서술자가 가지고 있는 (*STCR_i*, *NPTR_i*) 쌍은 전역시간인 *STCR_i*와 특정 이벤트 (프로그램) 내의 국지 시간인 *NPTR_i*이 동일한 시각을 나타낸다는 것을 의미한다. 예를 들어 NPT 가 그림 1과 같이 진행된다면, NPT 참조서술자의 (*STCR_i*, *NPTR_i*)는 (100,0), (102,2), (103,2), (105,4), (106,4), (108,6) 등의 값을 가지게 된다. STC 는 TS를 다중화할 때 결정되고 STCR(주파수 90KHz)은 STC (주파수 27MHz)를 1/300 로 샘플링한 값이므로 이것 역시 TS를 다중화할 때 결정된다. 따라서 NPT를 사용하게 하려면 TS 다중화기는 NPT 참조서술자를 생성하는 기능이 있어야 한다. 그러나 현재 상용으로 쓰고 있는 다중화기에는 이런 기능이 아직 없다.

본 논문에서는 NPT 참조서술자를 생성하기 위해 이미 생성된 TS로부터 PCR값(주파수 27MHz, STC와 주파수 같음)을 추출하고 이를 1/300으로 샘플링 하여 STCR로 사용한다. 물론 이때 STCR로 사용하고자 하는 PCR 에 대응되는 NPTR를 정해주어야 하는데, 이것은 사용자가 TS를 플레이하면서 해당 이벤트의 시작을 지정하고, 이때의 NPTR을 0으로 삼은 후, NPTR의 빈도수를 정하면 스트림 생성기가 NPTR 값들에 대응되는 PCR값을 읽어서 NPTR에 대응되는 STCR을 구한다.

본 논문에서 제시한 NPT 참조서술자 스트림 생성 방법은 매우 큰 제약을 가지고 있다. 이것은 실제 방송에서는 사용될 수 없고, PP의 스튜디오에서 연동형 애플리케이션을 제작하여 테스트하기 위해서만 사용할 수 있다. 실제 방송에서 각 이벤트 (프로그램)에 대한 NPT참조서술자를 생성하여 TS에 삽입하려면 PP의 방송서버와 플랫폼 사업자(예, 스카이라이프)의 서버 (다중화기)간의 긴밀한 실시간 통신이 요구된다. PP에서는 각 이벤트(프로그램) 별로 동영상을 보관하고 있다가, 해당 이벤트가 방송되어야 할 시점이 도달하면 그 순간, 해당 프로그램의 동영상을 실시간적으로 플랫폼 사업자의 서버로 전송하고, 플랫폼 사업자의 서버는 해당 이벤트의 동영상이 도착하는 순간, 그 순간의 NPTR을 0으로 설정하고 그 순간의 STCR 값을 샘플링 하여 (NPTR, STCR) 쌍을 만들어서 이벤트 시작점에 대한 NPT 참조서술자를 생성해야 한다. 플랫폼 사업자 서버는 모든 PP의 서버와 실시간적 통신으로 연결되어 있으므로, 모든 채널 (서비스), 모든 이벤트에 대해 위와 같은 방식으로 NPT참조서술자를 생성하여, TS에 삽입할 수 있다.

그러나 본 방법은 TS를 실시간으로 생성하지 않는다. 이미 생성된 TS를 이용하여 그 안에 이미 들어 있는 PCR을 추출하고, 이를 STCR로 재사용하여 각 이벤트의 NPT 참조서술자를 생성한다. 실제 방송환경에서 사용하려면, 위에서 설명한 대로 특정 이벤트(프로그램)가 시작되는 시점을 다중화기가 알고, 그 시점에 STCR을 샘플링 해야 한다. 그리고 이것을 실시간적으로 해야 한다. 그러나 본 방법에서는 이미 이벤트들이 구분 없이 연결되어 있는 TS로부터 제작자가 육안으로 이벤트의 시작을 확인하고, 그 시점의 STCR을 읽어낸다. 따라서 본 방법은 실제 방송환경에서는 사용할 수 없고, PP의 스튜디오에서 연동형 애플리케이션을 실험적으로 제작하여 테스트할 때만 사용할 수 있다.

III. NPT 재구성

정해진 시각에 이벤트가 정의하는 일을 수행하기위해서 Xlet 애플리케이션은 현재의 NPT를 알아야 하는데 이를 위해 MHP에서 정의한 API가 `getNPT()`이다. `getNPT()`를 사용하기 위해서는 방송국에서 NPT 참조서술자 스트림을 방송 스트림에 삽입해서 전송해야 한다^[3]. 그러나 현재

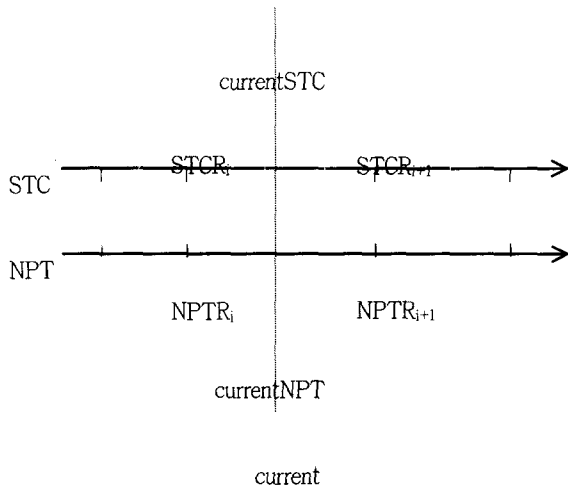


그림 2. NPT의 재구성
Fig. 2. Reconstruction of NPT

상용으로 나와 있는 다중화기는 NPT참조 서술자 스트림을 생성하는 기능이 포함되어 있지 않다. 그리고 국내에서 상용으로 사용되고 있는 미들웨어에서 구현하고 있는 getNPT() API는 NPT참조서술자 스트림 생성의 미비로 아직 충분히 테스트한 상태가 아니다.

1. NPT 재구성식

NPT 참조서술자가 가지고 있는 (STCR_i, NPTR_i) 쌍을 Xlet이 읽었을 때, 이것의 의미는 전역시간인 STCR_i와 국지시간인 NPTR_i이 동일한 시각을 나타낸다는 것을 의미한다^[6]. STCR_i은 현재순간의 STC (currentSTC)와 동일할 필요는 없고 그 전후의 값을 가지면 된다. 그러나 currentSTC로부터 너무 멀리 떨어져 있으면 좋지 않다. 현재순간의 NPT를 재구성하는 식이 방금 읽은 STCR_i와 currentSTC 간의 차이를 방금 읽은 NPTR_i에 더하는 것이기 때문이다 (그림 2, 식 (1) 참조).

currentNPT를 구하는 식은 다음과 같다^[6].

$$CurrentNPT = NPTR_i + (CurrentSTC/300 - STCR_i) \quad (1)$$

STCR의 단위는 STC 단위의 1/300이다. 식 (1)의 currentNPT 보간식은 NPTR_i가 도착한 후, NPT값이 연

속적으로 증가한다는 가정에 기초하고 있다. 따라서 프로그램이 편집되어 NPT값이 연속적이지 않게 되면, 불연속이 발생하는 시점에 NPT참조서술자를 생성해 주어야 한다. NPT참조서술자를 이렇게 전송하는 것은 NPT 관련 표준 규약에서 권고하고 있는 것이다^[3].

Xlet이 식 (1)을 이용하여 currentNPT를 재구성하기 위해서는 Xlet이 currentSTC를 읽을 수 있어야 한다. 하지만 DVB-MHP API에서는 STC가 전적으로 미들웨어 내부적인 문제라고 생각하기 때문에 애플리케이션에서 STC를 읽을 수 있는 API가 정의되어 있지 않다. 즉, 식 (1)은 미들웨어 내부에서 사용하는 식이며 애플리케이션에서는 사용할 수 없다.

따라서 본 논문에서는 Xlet에서 TS를 통해 전송되는 PCR을 읽어서 이를 이용하여 STC를 근사적으로 계산하는 방법을 제안한다. Xlet에서 PCR을 어떻게 읽을 것인가는 아래에 상세히 설명한다. Xlet이 가장 최근에 읽은 PCR을 PCR_j, 그 순간의 Java system time을 SysTime_j라고 하고, 현재 순간의 Java system time을 CurrSysTime이라고 하자. 그러면 currentSTC는 PCR_j + (CurrSysTime - SysTime_j)로 근사시킬 수 있다. 자바 시스템 시간의 단위는 밀리 세컨드이고 PCR의 단위는 27MHz (1/27000 millisecond)이므로 단위보정을 하면 currentSTC는 다음과 같이 계산할 수 있다.

$$currentSTC = PCR_j + 27000 * (CurrSysTime - SysTime_j) \quad (2)$$

(2)를 (1)에 대입하면, currentNPT는 다음과 같이 얻을 수 있다.

$$currentNPT = NPTR_i + [(90 * (CurrSysTime - SysTime_j) + PCR_j / 300) - STCR_i] \quad (3)$$

2. Xlet에서 PCR 읽기

이제 Xlet에서 TS에 포함되어 있는 PCR을 어떻게 읽을 것인가의 문제를 생각해 보자. 원래 PCR은 TS packet의 adaptation filed에 들어가 있고, DVB-MHP 데이터 방송 표준은 이 값 역시 미들웨어 내부적인 문제라고 생각하여 이 값을 읽을 수 있는 API를 제공하고 있지 않다. 물론

PCR을 읽는 새로운 함수를 만들면 되지만, 그것을 미들웨어를 수정하는 것을 의미한다. 본 논문은 미들웨어를 수정하지 않고 본 문제를 해결하는 방법을 제공한다.

데이터 방송 표준은 MPEG section안에 삽입되어 전송된 데이터를 읽을 수 있는 API (섹션필터라고 함)를 제공하고 있다. 따라서 본 연구에서는 송신단에서 PCR을 MPEG 섹션에 넣어 보내는 방법을 고안하였다. 이 방법은 주어진 TS를 분해하여 PCR이 들어 있는 TS packet으로부터 PCR 값을 읽어서 이를 MPEG 섹션으로 포장하여 현재 TS packet에서 가장 가까운 null 패킷에 삽입하는 것이다. PCR을 MPEG 섹션에 넣어서 보내는 것은 디지털 방송 규격에 어긋나는 것이다. 그러나 PCR을 읽을 수 있는 API가 없는 상황에서는 다른 방법이 없어 보인다.

PCR은 그것이 삽입된 TS내의 위치의 시점을 나타내도록 되어 있다. 즉 PCR을 읽는 순간의 시각과 읽은 PCR의 값이 같은 시각을 가리킨다. 그러나 위에서 기술한 대로 PCR을 원래 위치에서 뽑아 TS의 다른 위치로 이동하면 그 값의 의미를 상실한다. 따라서 PCR을 다른 TS packet으로 이동할 때, 그 이동거리만큼의 시간을 더하든지 빼든지 한 후 새로운 위치에 삽입해야 한다. TS는 고정된 비트 전송율을 가지고 있으므로 TS 내에서의 두 패킷간의 시간 간격은 쉽게 구할 수 있다. 그렇게 PCR이 들어가는 TS내의 위치가 바뀔 때 PCR의 값을 조정하면 PCR의 원래 성질, 즉 PCR을 읽는 순간의 시각과 그 PCR의 값은 동일하다는 성질을 만족시킬 수 있다.

한편, Xlet이 MPEG 섹션에 포함되어 전송되는 PCR 또는 NPT 참조서술자를 읽으려면 이들을 포함하는 MPEG 섹션들이 포함되어 있는 스트림의 PID를 알아야 한다. Xlet이 PID를 알고 있으면 이 PID를 가지는 MPEG 섹션들을 TS로부터 필터링할 수 있다. Xlet이 PID를 알 수 있는 가장 간단한 방법은 Xlet 애플리케이션이 PCR이 포함된 MPEG 섹션들의 PID를 미리 알고 있다는 가정하는 것이다. 그러나 PCR스트림이나 NPT 참조서술자 스트림의 PID를 포함하여 모든 PID는 TS 패킷들이 전송되는 스트림을 구분해주는 역할만 하기 때문에 전송스트림을 재다중화할 때 변경되지 않는다는 보장이 없다. 따라서 Xlet 코드가 PID를 직접 사용하면 전송 스트림을 재다중화할 때 마다, Xlet 코드를 수정해 주어야 한다. 이것은 매우 불편하고 타당하지 않은 방법이다. 이 문제는 Xlet이 특정 스트림의 PID를 알고 있다고 가정하는 대신, 그 스트림의 componentTag를 알고 있다고 가정하면 해결된다^[7]. 스트

림에 대한 componentTag는 스트림에 대한 식별번호로서 재다중화때 변경되는 값이 아니기 때문이다.

특정 componentTag가 가리키는 스트림이 TS에 삽입될 때 어떤 PID를 가진 TS 패킷들에 의해 전송될지는 다중화/재다중화시에 결정된다. 이를 표시하기 위해 스트림 생성기는 현재 선택된 서비스(PCR 이나 NPT참조서술자 스트림을 읽는 Xlet이 포함된)의 MAP (Program Map Table)에 특정 PID가 어떤 componentTag와 대응되는지를 나타내는 서술자를 삽입한다^[7]. Xlet은 알고 있는 componentTag를 이용하여 PMT에서 이에 대응되는 PID를 획득하고 이 PID를 이용하여 PCR 이나 NPT참조서술자 스트림을 읽는다.

식 (3)은 Xlet에서 STC 재구성을 위한 PCR 스트림과 NPT 참조서술자 스트림을 각각 따로 읽는다는 것을 가정하고 있다. 각 스트림은 MPEG 섹션 필터링을 통해서 읽는다. 섹션 필터는 귀한 리소스이고 또한 섹션 필터링을 많이 하면 수신기 시스템에 부하가 걸린다. 따라서 본 논문에서는 NPT 참조서술자와 PCR을 하나의 섹션에 포함시켜 전송함으로써 수신기가 필터링을 두 번해야 되는 부담을 한번으로 줄였다. 이렇게 하는 한 방법은 NPT 참조서술자를 구성할 때, PCR 스트림에 들어가는 각 PCR_i에 해당되는 NPTR_i를 만들고 이를 PCR_i이 들어가는 섹션에 넣는 것이다. 그렇게 되면 식 (3)은 다음과 같이 바뀐다.

$$\begin{aligned} \text{currentNPT} &= \text{NPTR}_i + [(90 * (\text{CurrSysTime} - \text{SysTime}_i) + \text{PCR}_i / 300) - \text{PCR}_i / 300] \\ &= \text{NPTR}_i + 90 * (\text{CurrSysTime} - \text{SysTime}_i) \end{aligned} \quad (4)$$

식 (4)의 두 번째 식을 보면 PCR 값이 사용되지 않는다는 것을 알 수 있다. 그러나 이 식의 유도과정을 보면 NPTR_i이 특수한 성질을 만족하고 있다는 것을 알 수 있다. 즉 NPTR_i은 아무 TS 패킷에 들어가도 되는 것은 아니고, 반드시 "PCR 패킷", 즉 수신기에서 패킷을 읽는 순간의 시각과 그 패킷 안에 들어있는 시간 값이 동일한 시각을 나타내는 패킷에 포함되어 전송되어야 한다. 그리고 식 (4)는 또한 NPTR이 PCR 빈도수만큼 TS에 삽입된다는 가정을 하고 있다. 원래 NPTR은 그렇게 자주 전송될 필요는 없다. 그러나 식 (4)가 STC를 PCR 값을 이용하여 근사적으로 구한다는 것을 감안하면, NPTR을 자주 전송해서 NPT 재구성의 근사성을 줄이는 것은 타당한 접근 방법이다.

IV. 실험

본 연구에서 제안한 NPT의 재구성기법을 검증하기 위해서 명시된 시각에 스트림 이벤트를 처리하는 “인터랙티브 게임쇼” 라는 Xlet 애플리케이션을 구현하였다. 이 애플리케이션은 비디오의 특정시점에 특정 이미지를 디스플레이 하도록 되어 있다. 그림 3에서 보는 것처럼, 이 이미지는 퀴즈문제를 기술하고 있는 이미지이다. 실험의 구체적인 방법은 다음과 같다.

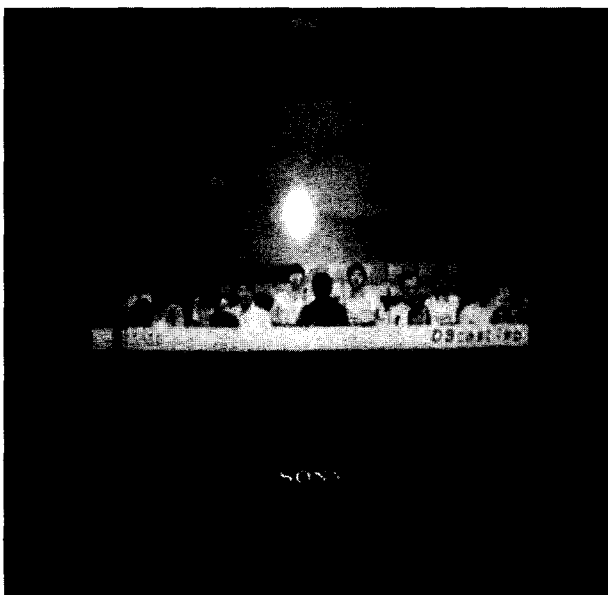


그림 3. 인터랙티브 게임쇼: 비디오의 특정 프레임에 특정퀴즈문제를 기술하는 이미지를 디스플레이 한다. 시청자는 리모컨으로 답을 선택한다. 동기화가 잘 이루어지지 않았다면 TV 수신기에서 이미지가 엉뚱한 프레임에 나타났을 것이다.

Fig. 3. Interactive game show

동기화된 데이터 스트림을 만들 때 주어진 비디오를 플레이하면서 특정 시점을 선택하여 스트림 이벤트의 시점을 정하고 이 스트림 이벤트에 이벤트 데이터 등을 삽입하여 스케줄드 스트림 이벤트를 생성, TS에 삽입한다. 스트림 이벤트가 포함된 TS는 본 연구팀에서 개발한 스트림 생성기를 사용하였다^[9]. 생성된 TS를 스트림 스테이션을 이용하여 셋톱박스에 전송하면, 셋톱박스에 방송이 디스플레이가 된다. 이때 화면을 캡처하고, 캡처된 동영상을 초당 25프레임단위로 나누어 프레임별로 점검하였다. 즉 이미지들이 디

스플레이 되는 시점이 TS를 생성할 때 명시한 특정 시점과 일치하는지를 확인했다. 그 결과 명시된 모든 지점에서 비디오와 이미지간의 동기화에 대한 사람의 식별오차범위인 240ms 이내^[10]에서 이미지가 비디오에 동기화됨을 확인하였다. 본 실험에서 측정된 오차의 원인은 크게 두 가지를 들 수 있다. 첫째는 NPT를 재구성하는데 필요한 PCR을 읽을 때 본 방법은 PCR이 셋톱박스에 도착하자마자 그 값을 읽을 수 있다는 가정을 하고 있으나, 실제로는 Xlet에서 PCR을 읽는데 걸리는 시간이 있고, 이것은 예측하기 불가능한 시간이다. 둘째는 NPT 재구성에서 사용하는 Java 시스템 타임이 원하는 정도만큼 정확한 것이 아니기 때문이다.

V. 결론

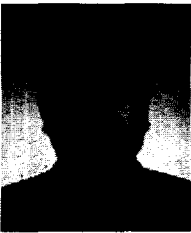
현재순간의 NPT를 정확히 재구성하는 것은 비디오와 애플리케이션의 동기화를 위해 반드시 필요하다. 그러나 현재 국내에서 상용으로 쓰고 있는 전송 스트림 생성기는 NPT를 사용할 수 있도록 NPT 참조서술자 스트림을 생성하고 있지 않고, 상용으로 사용하고 있는 수신기 미들웨어도 NPT를 재구성하는 API인 getNPT()를 구현했다고는 하나 NPT참조서술자 스트림의 미비 등으로 인해 충분히 테스트가 되어 있지 않다. 이런 상황에서도 PP들의 스튜디오에서는 연동형 데이터 방송을 제작, 실험할 필요가 있다. 따라서 본 논문에서는 이를 도와주기 위해 NPT를 수신기에서 근사적으로 재구성하는 방법을 제안하였다. 제안된 NPT 재구성 기법을 적용하여 명시된 시각에 이벤트를 처리하는 Xlet 애플리케이션과 스트림 이벤트가 포함된 스트림을 이용하여 실험한 결과 명시된 모든 지점에서 사람의 식별오차범위내에서 동기화가 됨을 확인하였다.

참고 문헌

- [1] ATSC. 2003. <http://www.atsc.org>
- [2] OCAP. 2003. <http://www.opencable.com>
- [3] ETSI TS 101 802 : Digital Video Broadcasting-Multimedia Home Platform 1.0.1, 2001.
- [4] Richard Chernock, "Data Broadcasting", McGraw-Hill, 2001.
- [5] ISO/IEC 13818-1 Generic Coding of Moving Picture and Associated Audio : Systems

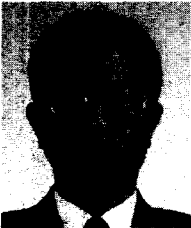
- [6] ISO/IEC 13818-6 Generic Coding of Moving Picture and Associated Audio : Digital Storage Media Command and Control
- [7] TR 101 202 DVB: Implementation guidelines for Data Broadcasting
- [8] Steven Morris, The Interactive TV Web. <http://www.mhp-interactive.org/tutorial> 2002.
- [9] 김세훈, 정문열, 백두원, 연동형 데이터 방송 애플리케이션을 위한 스트림 생성기 개발. 방송공학회 학술대회, pp. 267-270, 2003.
- [10] Ralf Steinmetz, "Human Perception of Jitter and Media Synchronization". IEEE Journal on Selected Areas in Communications, vol. 14, no. 1, Jan. 1996, pp. 61-72.
- [11] 정문열, 백두원, 연동형 데이터 방송 애플리케이션의 구조, 서강대 영상대학원 Technical Report, 2003. 방송공학회 논문지 제출.

저 자 소 개



정 문 열

- 서울대학교 계산통계학과 졸업 (BS)
- 카이스트 전산학과 졸업 (MS)
- University of Pennsylvania 전산학과 졸업 (Ph. D.)
- 일본 큐슈공과대학 정보공학부 조교수 역임
- 송실대학교 컴퓨터학부 부교수 역임
- 현 서강대학교 영상대학원 미디어공학과 부교수
- 주관심분야 : 컴퓨터 그래픽스 인터랙티브 방송



김 용 한

- 1982년 2월 : 서울대학교 공과대학 제어계측공학과 공학사
- 1984년 2월 : 서울대학교 대학원 제어계측공학과 공학석사
- 1990년 12월 : 렌슬리어공과대학 (Rensselaer Polytechnic Institute, 미국) 전기컴퓨터및시스템공학과 공학박사
- 1984년 3월 ~ 1996년 3월 : 한국전자통신연구원 책임연구원 (최종)
- 1996년 3월 ~ 현재 : 서울시립대학교 공과대학 전자전기컴퓨터공학부 부교수
- 1991년 10월 ~ 1992년 9월 : 일본 NTT 휴먼인터페이스연구소 객원연구원
- 1995년 8월 ~ 1996년 7월 : MPEG-Korea 의장
- 2002년 5월 ~ 현재 : 차세대디지털방송표준포럼 DMB분과위원장
- 주관심분야 : 멀티미디어 통신, 영상압축, 디지털TV, DMB



백 두 원

- 서울대학교 수학과 졸업 (BS)
- University of Minnesota 전산학과 졸업 (MS, Ph. D.)
- AT&T Bell Lab., Member of Technical Staff
- Cadence Design System, Member of Consulting Staff
- 현 송실대학교 미디어학부 부교수
- 주관심분야 : 디지털방송 컴퓨터 그래픽스 DRM 알고리즘