

동기화된 데이터 방송 애플리케이션 제작을 위한 스트림 생성기 개발

김 세 훈 정 문 룰

서강대학교 영상대학원 미디어공학과

skim@ganchannel.com moon@sogang.ac.kr

The Stream Generator for Synchronized Data Broadcasting Applications

Se hoon Kim Moon Ryul Jung

Dept of MediaTechnology, Graduate School of Media Communications, Sogang Univ

요 약

본 논문은 DVB-MHP 규약을 따르는 동기화된 데이터 방송 애플리케이션을 제작하기 위한 스트림 생성기의 구현을 기술한다. 동기화된 애플리케이션은 비디오와 공유된 시간기준을 가지며 특정시간에 발생한 이벤트(스케줄드 스트림 이벤트)를 처리할 수 있는 애플리케이션이다. 동기화된 애플리케이션을 위해서 스트림 생성기는 다음 작업을 수행한다. (1) NPT(Normal Play Time)를 수신기로 보내기 위해 NPT 참조서술자를 전송 스트림 내에 MPEG2 섹션으로 포장하여 넣는다. (2) 스케줄드 스트림 이벤트 들을 생성하여 전송 스트림내의 적당한 위치에 MPEG2 섹션으로 포장하여 넣는다. (3) 애플리케이션 프로그램과 스트림 이벤트에 대한 참조(reference)를 포함하는 객체 카로셀을 생성한다. DVB-J 애플리케이션은 DVB-MHP API중에 섹션필터 관련 API를 이용하여 전송된 NPT 참조 서술자를 읽어들이고 이로 부터 NPT를 재구성하며, 스트림 이벤트 관련 API를 이용하여 스트림 이벤트를 처리하도록 구현한다. 스트림 생성기가 생성한 MPEG2 전송 스트림으로 Interactive Game Show 애플리케이션을 제작하여 비디오와 애플리케이션이 사용하는 이미지 디스플레이 간의 동기화를 테스트하였다.

1. 서 론

연동형 데이터 방송은 동영상과 데이터방송이 결합된 형태로서 인터랙티브 TV 프로그램을 구현할 수 있는 방송 콘텐츠 형식이다. 연동형 콘텐츠 중의 하나로 한 프로그램에 속하는 비디오의 시간과 동기화 된 데이터 방송 콘텐츠를 생각할 수 있다. 동기화 된 데이터 방송 콘텐츠는 방송의 내용과 흐름, 시간에 따라 알맞은 데이터 방송을 실시된다. 동기화 된 데이터 콘텐츠는 방송 프로그램을 보완하고 시청자의 자발적 참여를 유도한다.

본 논문은 DVB-MHP 표준에 따르는 동기화된 데이터 방송 애플리케이션이 포함된 방송스트림을 생성하는 생성기의 구현을 기술하는 것을 주목적으로 한다. 동기화된 데이터 방송용 애플리케이션을 제작하기 위해서는 애플리케이션과 비디오가 공유하는 시간값가져야 하며 실행환경과 전송대역을 고려하여 동기화된 데이터를 전송하여야 한다[1]. 동기화 방법으로 스트림 이벤트를 이용할 수 있다[2]. 스트림 이벤트는 전송 스트림에 이벤트를 전송하고 이벤트가 셋탑박스로 도착한 시점에 이벤트를 발생 시킨다. 스트림 이벤트는 애플리케이션에서 이벤트가 셋탑박스에 도착한 시점이 아니라 이벤트가 도착한 후 특정 시각에 이벤트가 발생하도록 할 수 있는데 이를 스케줄드 스트림 이벤트 라고 한다. 스케줄드 스트림 이벤트 는 스트림 이벤트 서술자에 이벤트

NPT를 포함시켜서 전송한다. 스케줄드 스트림 이벤트 가 도착하면 NPT 참조값 서술자에 전송 되는 NPT값과 비교하여 정해진 이벤트NPT 시간에 이벤트를 발생시킨다. NPT는 비디오의 기준시간을 통해서 생성되며 NPT 참조값 서술자에 의해 MPEG2 섹션으로 전송된다[3]. NPT는 하나의 DVB-이벤트에서 연속되는 시간기준이므로 이벤트에 따라 생성된 애플리케이션이 이용하기 적합하다.

스트림 이벤트를 이용하기 위해서는 애플리케이션이 객체 카로셀 구조로 전송되고 객체 카로셀 구조안에 스트림 이벤트에 대한 참조값을 가지고 있어야 한다[2]. 본 논문에서는 객체 카로셀 프로토콜로 애플리케이션을 전송하는 스트림 생성기를 개발하여 실제 애플리케이션에 스케줄드 스트림 이벤트를 적용하고 동기화 방법을 테스트하여 스트림 생성기를 검증하였다. 본 스트림 생성기를 통한 동기화된 애플리케이션의 제작은 데이터방송을 보다 풍부하고 시청자를 끌어당길 수 있는 인터랙티브 콘텐츠로 만들어 줄 것이다.

2. 배경이론

본 장에서는 DVB-MHP 규약에 따르는 동기화된 애플리케이션 제작을 위한 스트림 생성기 개발에 필요한 배경이론에 대해 서술하겠다.

2.1 데이터 전송 프로토콜

디지털 방송에서는 비디오와 오디오, 데이터를 수신기로 전송하기 위해서 MPEG2 System[6]규약에서 정의된 MPEG2 전송 스트림을 사용한다. DVB-MHP 규약에서는 애플리케이션의 전송을 위해서는 DVB 객체 카로셀을 사용한다. DVB 객체 카로셀은 DSMCC User to User 객체 카로셀을 기본으로 해서 정의된다. 그림 1에서 보듯이 애플리케이션은 DVB 객체 카로셀 구조로 인코딩된 후에 데이터 카로셀로 인코딩되고 다시 MPEG2 섹션으로 인코딩된후에 MPEG2 전송 스트림으로 패킷화 된다.

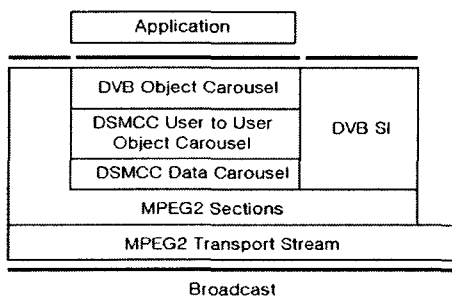


그림 1 : 데이터 전송 프로토콜(Broadcast Protocol stack)

2.2 스트림 이벤트

스트림 이벤트는 스트림 이벤트 서술자의 구조로 MPEG2 섹션에 담겨져서 MPEG2 전송 스트림에 전송된다. 스트림 이벤트를 이용한 애플리케이션은 스트림 상의 특정 시간에 이벤트를 발생시킬 수가 있어 비디오와 동기화할 수 있다. 애플리케이션이 스트림 이벤트를 이용하는 것은 MPEG2 전송 스트림상에 스트림 이벤트 서술자가 포함되는 것만으로 가능하지는 않다. 애플리케이션이 전송되는 프로토콜인 오브젝트 카로셀 구조안에 스트림 이벤트의 주소에 대한 참조값 정보를 담고 있는 스트림 이벤트 오브젝트가 포함되어야 한다. 스케줄드 스트림 이벤트는 스트림 이벤트 서술자내에 예정된 데이터 처리시간(이벤트NPT)을 명시하고 애플리케이션이 그 시간에 데이터를 처리한다. 이러한 동기화를 위해서는 송신단과 수신단에서 동일한 시간인 NPT가 전송되며 애플리케이션에서도 그 시간을 이용할 수 있어야 한다. 또한 DVB-J 애플리케이션에서 스트림 이벤트를 이용하기 위해서는 애플리케이션에서 org.dvb.dsmcc.DSMCCStreamEvent 객체를 생성하여야 한다[5].

2.3 NPT(Notmal Play Time)

DVB-MHP에서 사용하는 NPT는 STC를 이용하여 생성되고 NPT 참조값 서술자에는 STC와

NPT의 샘플링 값인 STC 참조값과 NPT 참조값이 전송된다. NPT는 하나의 미디어 혹은 DVB-이벤트에서의 시간이다[3]. NPT 참조서술자에서 보내지는 참조값 값들은 STC의 재구성에 이용되지 않고 NPT 참조값의 사이값을 보장하는것에 이용된다. DSMCC 규약에서 정의된 전송된 NPT의 재구성은 다음 식과 같다.

$$NPT_k = STC_k - STC_{reference} + NPT_{reference}$$

실제 구현에서는 시스템의 STC를 얻어올 수 없으므로 규약과 다른 방법으로 NPT를 재구성이 필요하다. STC가 수신단에서 재구성되기 위해서 전송 스트림에 PCR을 넣어서 보내듯이[4] NPT는 NPT 참조값을 통하여 NPT 참조서술자내에 넣어서 보내지며 수신기에서 재구성한다.

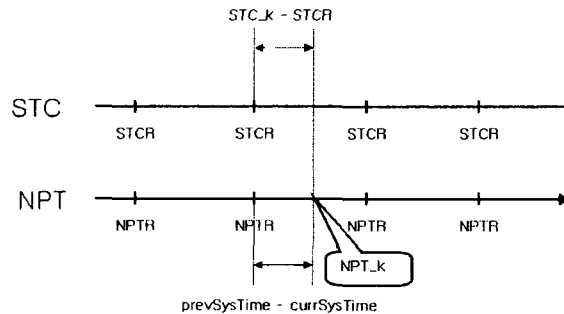


그림 2 : NPT 재구성(NPT reconstruction)

DVB-J 애플리케이션은 매 전송된 NPT 참조값 서술자에서 NPT 참조값을 섹션필터를 통하여 추출한다. NPT를 재구성하기 위해서는 디코더인 셋탑박스의 시간기준 STC값을 읽을 수 있어야 한다. 본 논문에서는 NPT 참조값 보정을 위한 시스템시간으로 일반 Java API에서 제공하는 MilliSecond 단위의 시간을 사용하였다. 그림 2에서 보듯이 k라는 시점의 NPT값은 바로전의 NPT참조값에서 다음 NPT참조값이 도착하기 전의 시각이다. NPT는 STC를 이용해서 k라는 시점의 NPT값을 보정하게 되는 데 다른 시스템의 시간값으로 그 차이값을 보정할 수 있다면 굳이 STC를 이용하지 않아도 k라는 시점의 NPT값이 보정된 NPT라고 할 수 있다. DVB-MHP에서 NPT의 단위는 millisecond이다. NPT를 구하기 위한 식은 다음과 같다.

$$NPT = (curSysTime - prevSysTime) + NPTR/90$$

curSysTime은 NPT를 구하는 함수가 호출되었을 때의 시스템시간이며 prevSysTime은 NPT참

조값을 선택필터를 통하여 읽어들이었을 때의 시스템 시간이다. 위의 식으로 NPT는 getNPT() 함수로 구현되며 애플리케이션에서 재구성되고 동기화에 이용된다.

3. 스트림 생성기 개발

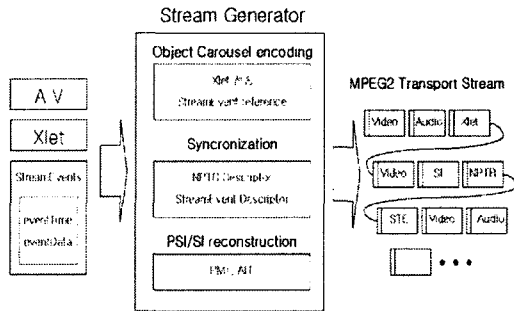


그림 3 : 스트림 생성기의 구조(structure of stream generator)

동기화된 데이터 방송을 제작하기 위해서는 전송 스트림을 생성하고 생성된 스트림을 해석하는 애플리케이션을 개발하여야 한다. 본 논문에서는 동기화된 데이터 방송을 제작하기 위해 스트림 생성기를 개발하였다. 본 스트림 생성기는 애플리케이션을 오브젝트 카로셀 방식으로 전송하며 스케줄드 스트림 이벤트를 이용하기 위해서 오브젝트 카로셀내에 스트림 이벤트 참조값을 포함하고, 스트림 이벤트 서술자와 NPT 참조값 서술자를 생성한다. 그림 3에서 데이터전송 스트림 생성기 구조에 대해서 나타내었다. 스트림 생성기는 셋탑 박스에서 실행될 자바(Java) 애플리케이션인 엑슬릿[6]과 엑슬릿에서 실행될 데이터를 입력 받는다. 스트림 생성기는 DVB-MHP 규약에 따라 애플리케이션을 객체 카로셀 방식으로 MPEG2 전송 스트림에 넣으며 해당 데이터는 MPEG2 섹션의 스트림 이벤트 서술자에 넣는다. NPT 참조값 서술자에서 NPT는 DSMCC 규약[3]에 의해서 DVB-이벤트의 시작시간에서 연속적으로 증가하는 값을 생성된다. 사용자의 Xlet과 데이터를 입력으로 받은 스트림 생성기는 오브젝트 카로셀 구조로 포장된 Xlet, 데이터를 담고 있는 스트림 이벤트 서술자, 시간 참조값을 가지는 NPT 참조값 서술자를 포함하는 MPEG2 전송 스트림을 생성한다.

3.1 스케줄드 스트림 이벤트 애플리케이션 구현

스케줄드 스트림 이벤트를 이용하는 애플리케이션은 크게 세부분으로 구성된다. 첫째는 스트림 이벤트 객체를 생성하여 스트림에서 발생한 스트

림 이벤트를 가져오는 부분, 둘째는 선택필터 객체를 생성하여 MPEG2 섹션에 전송되는 NPT 참조 서술자를 읽어서 NPT를 재구성하는 부분, 셋째로 스케줄드 스트림 이벤트를 발생시키기 위하여 스트림 이벤트 상의 eventNPT와 현재 시각(NPT)을 비교하는 부분이다. 그림 4과 같이 애플리케이션은 스트림이벤트 객체를 통하여 스트림이벤트가 발생하였을 때 스트림 이벤트 서술자로 접근하여 해석한다. 이벤트NPT는 스케줄드 스트림 이벤트에서 이벤트가 도착한 후에 애플리케이션이 스트림 이벤트를 실행할 시각을 나타낸다. 스트림 이벤트 서술자의 이벤트데이터에 들어있는 데이터는 애플리케이션이 실제 사용할 데이터이다. 본 논문에서는 스트림 이벤트가 발생한 시각에 이미지 파일을 읽어 들이고 계획된 이벤트 NPT에 이미지를 디스플레이한다.

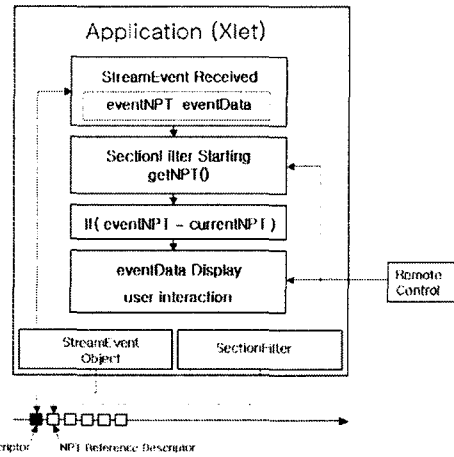


그림 4 : 스케줄드 스트림 이벤트 구현 (Implementation scheduled stream event)

4. Interactive Game Show 제작



그림 5 : 인터랙티브 게임쇼 실행화면 (Interacitve game show)

스케줄드 스트림 이벤트를 이용하여 Interactive Game Show 애플리케이션을 개발하였다. 시청자가 답을 입력할 수 있는 애플리케이션을 제작하고 입력 가능한 시간을 스케줄드 스트림 이벤트를 이용하여 제어하였다.

동영상과 이미지 디스플레이의 동기화는 보는 사람이 받아들일 수 있는 오차범위를 가지고 있다 [7]. 시청자는 동영상과 이미지가 240ms(millisecond) 범위내에서 다른 시간에 보여지더라도 같은 시각에 동기화된 것으로 인식한다. 본 논문에서는 수신기 화면에서 보여지는 동영상과 DVB-J 애플리케이션을 촬영한 후 촬영된 동영상을 프레임별로 점검하여 동기화를 확인하였다. 그림 6에서 7프레임은 Interactive Game Show에서 문제가 주어지고 답을 할 수 있는 시작 시점으로 그래픽 이미지가 동기화 되어야 할 지점이다. 실제로는 8프레임에서 그래픽 이미지는 디스플레이됨을 그림에서 확인할 수 있다. 그림 11에서 1프레임의 오차가 발생하였는데 이것은 40ms의 차이로서 오차범위내에 있다. 이와 같은 방법으로 동영상의 각 지점에서 동기화 정확도를 점검하였는데 모두 240ms 이내에 동기화가 됨을 확인하였다.

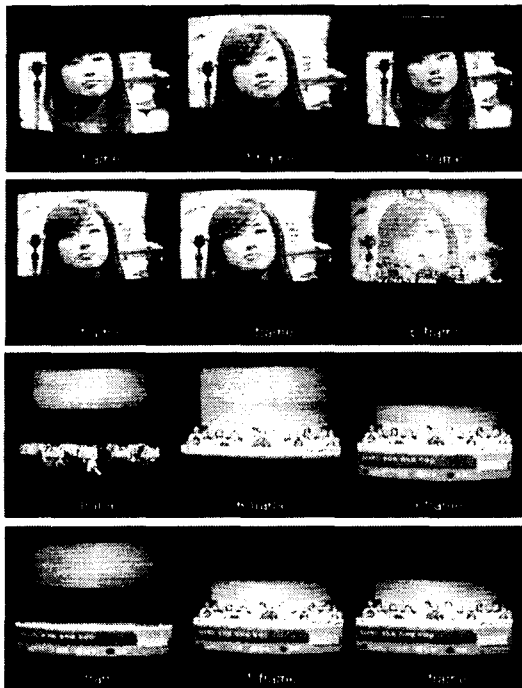


그림 6 : 프레임별 인터랙티브 게임쇼 (Interactive gameshow slide)

5. 결론

동기화된 데이터 방송 애플리케이션을 제작하기 위해서 스트림 생성기를 개발하고 실제 DVB-J 애플리케이션을 제작하여 동기화를 테스트하였다.

비디오와 애플리케이션의 동기화는 스케줄드 스트림 이벤트를 이용하였다. DVB-MHP 규약에 의거하여 객체 카로셀 구조안에 애플리케이션을 전송하고 스트림 이벤트 객체를 포함하며 추가적으로 MPEG2 섹션에 NPT 참조값 서술자, 스트림 이벤트 서술자가 포함된 MPEG2 전송 스트림을 생성해 주었다. 원활한 스트림 이벤트의 생성 및 동기화된 시점에 데이터의 전송을 위하여 본 논문에서 정의한 스트림 이벤트 서술 파일을 사용했다. 동기화된 스트림을 이용하는 DVB-J 애플리케이션은 스트림 이벤트 객체와 섹션 필터 객체를 통해서 MPEG2 전송스트림의 정보를 얻어오며 각 객체를 해석하는 기능을 갖추어야 한다. Interactive Game Show 애플리케이션을 제작하여 비디오와 이미지 디스플레이의 동기화를 점검한 결과 사람의 눈으로 허용할 수 있는 +-240ms이내에서 동기화가 이루어 졌다. 본 논문에서 개발한 스트림 생성기를 통한 동기화된 애플리케이션 제작방법은 안정적이며 효과적인 동기화 방법이다.

6. 참고문헌

- [1] A.Vogel et. al."Distributed Multimedia and QoS : A Survey", IEEE Multimedia, Summer, 1995, pp10 - 19
- [2] ETSI TS 101 802 : Digital Video Broadcasting-Multimedia Home Platform 1.0.1
- [3] ISO/IEC 13818-6 Generic Coding of Moving Picture and Associated Audio : Digital Storage Media Command and Control
- [4] ISO/IEC 13818-1 Generic Coding of Moving Picture and Associated Audio : Systems
- [5] John Jones "DVB-MHP/JavaTV Data Transport Mechanisms", International Conference on Technology of Object-Oriented Languages and Systems, Sydney, Australia, 2002
- [6] JavaTV Technical Overview <http://java.sun.com/products/javatv/>
- [7] Ralf Steinmetz, "Synchronization Properties in Multimedia Systems", IEEE Journal on Selected Areas in Communications, vol. 8, no. 3, April 1990, pp. 401 - 412