

디지털 방송 수신기를 위한 효율적인 트랜스포트 스트림 패킷 매니저

고상원^o, 전제민, 원재훈, 김세창, 김정선
 한양대학교 컴퓨터공학과

funkcode^o@gmail.com, luvjim@naver.com, jhwon@cse.hanyang.ac.kr, sckim@cse.hanyang.ac.kr, jskim@cse.hanyang.ac.kr

Efficient Transport Stream Packet Manager for Digital Broadcast Receiver

Sang Won Ko^o, Je Min Jeon, Jae Hoon Won, Seh Chang Kim, Jung Sun Kim
 Department of Computer Science & Engineering, Hanyang University

요 약

기존의 아날로그 방송에서 디지털 방송으로의 전환이 가속화 되고 있으며, 디지털 방송은 아날로그 방송과는 다르게 다양한 서비스를 사용자에게 제공 할 수 있다는 장점을 가지고 있다. 디지털 방송에서는 트랜스포트 스트림을 이용하여 비디오/오디오 이외에 데이터를 전달하여 방송과 관련된 정보나 어플리케이션을 다룰 수 있다. 트랜스포트 스트림을 방송환경에서 사용 할 경우 응답시간이나 중복되는 패킷들에 대해서 어떻게 고려하느냐에 따라서 시스템 전체에 성능을 영향을 줄 수 있다. 본 논문에서는 데이터 방송을 위하여 패킷 단위로 전송되어온 트랜스포트 스트림을 효과적으로 처리 및 분류하고 선택적으로 상위 계층으로 전달하는 패킷 매니저를 설계 및 구현하였다.

1. 서 론

디지털 방송은 기존의 아날로그 방송보다 더 좋은 화질과 음향, 그리고 더 많은 채널과 데이터 방송으로 인한 프로그램 가이드, 게임, 양방향 응용 프로그램과 같은 다양한 서비스를 제공 할 수 있다는 장점을 가지고 있다. 이와 같은 디지털 방송은 비디오와 오디오, 그리고 데이터들이 섞인 비트 스트림 형태로 전달되게 되는데, 이와 같은 비트 스트림은 MPEG-2 TS(Transport Stream) 형식으로 인코딩되어 전송된다. 디지털 데이터가 시청자에게 직접 전달이 가능하고 사용자의 선택이 콘텐츠 제공자에게 전달되는 대화형 서비스도 가능하며, 이제는 단순한 텔레비전 방송 시청만이 중요한 것이 아니라 시청자의 요청에 대한 응답 또한 중요하게 되었다[2]. 본 논문은 디지털 방송이 MPEG-2 TS으로 전송 될 때, 실시간 방송 환경이라는 특성과 트랜스포트 스트림의 특성을 고려하여 데이터 방송과 관련된 패킷들을 관리 할 수 있고 데이터를 필요로 하는 상위 계층으로 효율적으로 전달 할 수 있도록 설계하고 구현하였다.

2. 트랜스포트 스트림(Transport Stream)

방송을 하기 위해서 비디오, 오디오 콘텐츠 그리고 데이터를 포함하여 방송에 적합하도록 MPEG-2 압축을 하여 방송에 적합한 형태인 트랜스포트 스트림으로 만든다. 트랜스포트 스트림은 188, 192, 204 byte 단위로 구

```

transport_packet(){
    sync_byte
    transport_error_indicator
    payload_mit_start_indicator
    transport_priority
    PID
    transport_scrambling_control
    adaptation_field_control
    continuity_counter
    if(adaptation_field_control == '10' || adaptation_field_control == '11'){
        adaptation_field()
    }
    if(adaptation_field_control == '01' || adaptation_field_control == '11'){
        for(i = 0; i < N; i++){
            data_byte
        }
    }
}
    
```

그림 1 TS packet

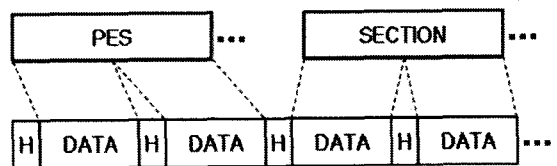


그림 2 TS packet 과 PES, Section

성 될 수 있으며, 본 논문에서는 188 byte를 기준으로 하여 설명한다. 트랜스포트 스트림에서 각각 패킷의 시작과 끝의 구분은 항상 8비트 값인 '0x47'(0100 0111) 값을 가지는 sync_byte를 이용하여 알 수 있다. transport_error_indicator는 항상 0의 값을 가지며, 에러

가 발생할 경우 1이 되어 문제가 생겼음을 알 수 있게 한다. payload_unit_start_indicator는 '1'일 경우 패킷들의 집합의 첫 번째 패킷임을 나타내며, '0'일 경우에는 나머지 패킷임을 나타내기 때문에 여러 개의 패킷들이 모여서 PES 패킷이나 하나의 Section을 만들 때 중요한 역할을 한다. transport_priority는 같은 PID가 들어 왔을 경우, 이 값이 '1'인 TS의 우선순위가 높음을 나타낼 수 있다. PID는 해당 패킷의 ID를 나타내며, 각 패킷을 구분할 수 있게 해주는 식별자(identifier) 역할을 한다.

Value	Description
0x0000	Program Association Table
0x0001	Conditional Access Table
0x0002	Transport Stream Description Table
0x0003 - 0x000F	Reserved
0x00010 ... 0x1FFE	May be assigned as network_PID, Program_map_PID, elementary_PID, or for other purposes
0x1FFF	Null packet

표 1 PID Table

transport_scrambling_control은 해당 패킷의 스크램블링 모드를 알려주며, 원하지 않는 데이터가 들어오는 것을 피하기 위해 데이터의 순서를 섞는다. '00'인 경우에는 'not scrambled'를 의미하며 나머지 '01', '10', '11'은 User-defined로 되어 있다. adaptation_field_control은 해당 패킷의 헤더 다음에 나오는 데이터가 어댑테이션 필드인지 페이로드인지 또는 실제 페이로드의 위치가 어디에 있는지 알게 해준다.

Value	Description
00	Reserved for future use by ISO/IEC
01	No adaptation_field, payload only
10	Adaptation_field only, no payload
11	Adaptation_field followed by payload

표 2 Adaptation field control values

continuity_counter는 같은 PID를 가진 패킷에서 각 패킷의 순서를 알려주는 역할을 하며, 0에서 15까지의 값을 가지게 된다. 만약 패킷의 개수가 16개 이상이면 모

둘라 연산을 통하여 0부터 다시 시작한다. 만약 adaptation_field_control이 '00' 또는 '01'일 경우에는 이 값이 증가되지 않는다. 지금까지 설명한 섹터를 기준으로 패킷이 분류되어 상위 레이어의 PES나 섹션(Section)형태로 만들어지게 되는데, PES의 경우 비디오와 오디오 신호가 들어있는 ES가 패킷으로 나누어진 것이며, Section은 비디오, 오디오 데이터와 같이 TS로 전달되는 데이터가 일정한 크기로 나누어진 것이다[1]. 본 논문에서는 각 패킷의 PID에 대한 정보를 담고 있는 Table이 섹션으로 나눈 다음 패킷 형태로 전송되고 있기 때문에 Section에 대해서만 다루도록 하겠다.

2.1 Section

패킷이 모여서 섹션 형태가 되면 크게 두 가지로 분류가 가능하다. TS에 대한 정보를 담고 있는 SI(Service Information) 섹션과 방송환경을 위한 파일 시스템인 DSM-CC 섹션이 있다. SI 섹션 같은 경우에는 경우에 따라 하나 이상의 섹션이 모여서 하나의 테이블 역할을 하게 되는데, 각각의 SI 테이블은 고유의 역할을 가지고 그에 맞는 정보를 가지고 있다. TS에서 패킷마다 가지고 있는 정보가 무엇인지 PID를 기준으로 알 수 있도록 하는 정보도 SI 테이블에서 가지고 있다. 그러므로 고정적으로 PID를 부여받은 PAT와 같은 테이블 외에는 특정 순서에 맞게 파싱(Parsing)과정을 거쳐야 유동적인 PID를 가진 패킷들이 의미가 있는 패킷이 된다.

PAT (PID = 0)

program number	PMT PID
1	23
3	13
⋮	

PMT (PID = 13)

stream	stream-type	PID	app-type
1	video	51	null
2	audio	16	null
⋮			

그림 3 PAT와 PMT

PMT 같은 경우에 PAT를 통해서 PMT를 PID를 알 수 있으며, PMT를 통해서 해당 프로그램에 대한 다른 패킷들의 PID를 알 수 있다[4]. PMT 이외에도 다른 테이블

역시 이와 같은 문제가 있으며, 그에 대한 설명은 관련 표준을 찾아보길 권한다[1].

3. 패킷 매니저(Packet Manager)

패킷 매니저는 방송 환경의 특성을 고려하여 TS에서 패킷 단위로 관리하여 상위 계층으로 원하는 데이터를 전달하는데 주목적이 있다. 오디오와 비디오의 경우는 디멀티플렉서를 거쳐서 디코더로 바로 전달되는 것이 일반적이므로 본 패킷 매니저에서는 디지털 방송에서 중요한 역할을 하는 데이터에 대해서 다루고 있다. 패킷 매니저에서는 방송 환경이라는 특수한 환경에서 중요한 특징인 latency 문제를 고려하고 있는데, 일반적인 컴퓨터 환경에서의 latency는 저장장치에서 원하는 데이터를 읽어내는데 걸리는 시간이지만, 방송 환경에서는 데이터가 TS를 통하여 반복적으로 전달되는 주기가 latency가 된다 [3]. 그러므로 만약 상위 계층에서 특정 데이터에 대한 패킷을 요청할 경우에 최악의 경우 해당 데이터의 전송 주기만큼 기다려야 하는 경우가 발생한다. 그리고 TS에서 패킷은 반복적으로 전달되도록 구성되어 있기 때문에, 같은 버전을 가진 패킷이 여러 차례에 걸쳐서 전달되고 있다. 방송 환경이라는 특수한 상황이기 때문에 반복적으로 전달되는 것이지만, 수신기에서 해당 데이터를 정확하게 수신하여 상위 계층으로 전달하였다면 굳이 같은 버전의 동일한 데이터를 다시 반복하여 처리할 이유가 없어진다. 본 논문에서 개발한 패킷 매니저는 위에서 언급한 방송 환경의 특성들을 고려하였다.

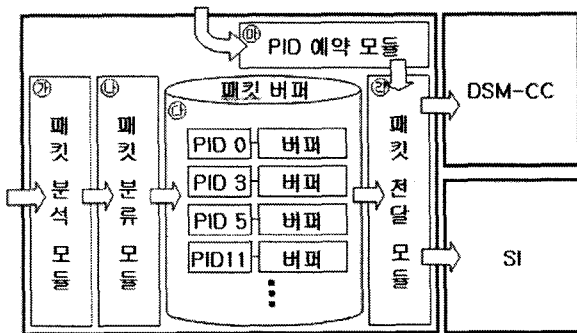


그림 4 패킷 매니저(Packet Manager) 구조

3.1 패킷 매니저의 구조

본 논문에서 제안한 패킷 매니저는 그림 4와 같은 구조로 되어 있다. ㉑ 패킷 분석 모듈은 TS에서 들어오는 하나의 완전한 하나의 패킷에 대해 파싱(Parsing)을 하여 패킷의 섹스들을 얻어 올 수 있도록 한다. TS 패킷은

4byte의 헤더를 가지고 있으며, 기존에 패킷 분석을 할 경우에 TS 패킷의 헤더에서만 필요한 정보를 가져와 분석을 하였다. 하지만 본 패킷 분석 모듈에서는 payload_unit_start_indicator가 1인 경우에는 헤더를 제외한 payload 부분에 동일 PID 패킷들이 합쳤을 경우 섹션의 헤더 부분에 해당하는 부분을 포함하고 있기 때문에, 섹션의 헤더 부분에 해당하는 섹스들의 정보들인 table_id나 version_number, 그리고 section_number 등을 미리 얻어 올 수 있도록 하였다. ㉒ 패킷 분류 모듈은 해당 패킷의 PID를 기준으로 하여 분류하게 되며, 여러 가지 동일한 섹스들이 동일한 PID를 사용하는 경우에는 table_id등 다른 섹스를 이용한다. 그리고 해당 PID에 대한 continuity_counter를 이용하여 몇 개의 패킷이 각 PID 별 버퍼에 저장되고 있는지 확인하며, version_number를 통하여 동일한 버전에 대해서 불필요하게 중복된 과정을 거치지 않도록 한다. ㉓ 패킷 버퍼는 패킷 매니저로 들어온 패킷마다 분류 기준이 되는 정보와 버퍼를 할당하여 해당 PID에 대한 패킷들을 버퍼에 저장한다. 저장하면서 continuity_counter를 증가하여 버퍼에 저장된 패킷들에 대한 카운터를 유지한다. ㉔ 패킷 전달 모듈은 패킷 버퍼에 있는 패킷 중에서 상위 계층에서 전달 받기를 원하는 PID에 대해서 패킷들이 모두 모였는지 확인을 한 다음에 요청이 들어온 상위 계층으로 전달한다. 해당 PID에 대해서 모든 패킷들이 모였는지에 대해서는 payload_unit_start_indicator가 1인 패킷에서 섹션의 헤더 부분에 해당하는 section_length 섹스에 대해 값을 얻어와 현재 버퍼에 들어와 있는 패킷들의 크기와 비교하여 알 수 있다. ㉕ PID 예약 모듈은 상위 계층에서 원하는 데이터에 해당하는 예약하여, 선별적으로 상위 계층으로 패킷들을 전달 할 수 있도록 한다. 선별적인 패킷 전달은 상위 계층에서 사용하지 않음에도 불구하고 전달되어 불필요한 데이터 처리를 하게 되는 경우를 없애준다.

3.2 패킷 관리 전략

패킷들을 관리하기 위해 각각 패킷들을 분류하는 것을 우선으로 한다. 패킷을 분류하기 위해 기준이 되는 섹스로는 PID, table_id, version_number, continuity_counter, section_number, section_length가 있다. 단순히 PID만으로 분류하지 않는 이유는 상위 계층에서 같은 PID로 여러 종류의 섹션이 전달되는 경우가 있기 때문이다. version_number는 동일한 버전의 섹션을 중복하여 처리하게 되는 경우를 줄여줄 것이며, section_number는 동일한 PID와 table_id 그리고 version_number를 가진 다른 섹션을 구분하기 위함이다. section_length와 continuity_counter는 하나의 섹션을 구성하기 위한 패킷

들인지 확인하고 필요한 패킷이 모두 모였는지 확인하는 데 사용된다. 그리고 패킷을 관리하는데 있어서 중점적인 사항은 앞에서 얘기한 바와 같이 latency와 불필요한 연산의 최소화라고 할 수 있다. 방송 환경에서 latency 같은 경우에는 사용자가 원하는 패킷들을 다시 전송되어 오기까지 기다리는 시간인데, 수신자 사용자가 원하는 서비스를 제공하기 위해서 해당 서비스에 대한 패킷들의 PID를 알려면 일련의 과정을 거쳐야 한다. 예를 들어 특정 서비스의 어플리케이션을 실행시키기를 원하는 경우에는 PAT를 통해 사용자가 원하는 서비스에 대한 정보를 가진 PMT의 PID를 찾은 다음에 어플리케이션에 대한 정보를 가진 AIT의 PID를 PMT를 통해서 알아야 한다. 이와 같은 경우에 특정 테이블에 증속되어진 형태를 가지고 있으므로, 사용자가 원하는 정보에 대해서 즉각적으로 처리가 불가능하다. 본 논문의 패킷 매니저는 이 시간을 최소화하기 위해서 패킷 매니저는 입력되는 모든 패킷을 PID별로 분류를 하여 후보 패킷들로서 미리 패킷 버퍼에 저장을 한다. 그리고 모여진 패킷들을 무조건 상위 계층으로 전달하는 것이 아니라 사용자의 요청이 들어오거나 특정 순서에 따라서 필요한 패킷들을 찾을 때, 패킷 버퍼를 먼저 검색하여 필요한 패킷이 모두 모여 있는 경우에는 바로 패킷 전달 모듈을 통하여 패킷을 요청한 상위 계층으로 전달하도록 한다. 만약에 원하는 PID에 해당하는 패킷들이 모두 없을 경우는 기다렸다가 패킷 전달 모듈을 통하여 상위 계층으로 전달하도록 한다.

3.3 구현 결과

본 논문에서 제안한 패킷 매니저는 MS Windows XP, Visual C++ 6.0 환경에서 개발하였다. 패킷 매니저는 C++를 이용하여 구현하였으며, 특정 플랫폼에 종속되지 않도록 코드를 작성하였다. 테스트 및 패킷 매니저의 사용은 실제 수신기를 가정하여 JNI를 이용하여 JAVA에서

이루어졌다. JNI를 이용하게 되면 상위 계층에서 미들웨어 등에서 실제 하드웨어에 종속적인 부분을 PC환경에서 가상으로 시뮬레이션 할 경우에도 명확히 파악 및 제시 할 수 있다는 장점이 있다.

4. 결론 및 향후 과제

본 논문에서는 디지털 방송을 위한 MPEG TS 패킷 중에서 데이터 방송과 관련된 패킷들을 효율적으로 관리 및 전달하도록 하였다. 제안된 패킷 매니저를 통하여 상위 계층으로 전달된 패킷들은 해당 계층에 적합한 정책에 맞추어 관리가 될 것이다. 모든 섹션은 패킷들의 집합이므로 패킷 매니저에서의 우선적인 관리로 인하여 부담이 줄어들게 된다. 그리고 패킷 매니저는 디지털 방송 표준인 DVB, OCAP의 차이점에 상관없이 동일하게 적용이 가능하도록 하였다. 이러한 패킷 매니저는 하드웨어의 영향이 적은 PC상에서 방송 환경을 시뮬레이션하거나 특정 패킷에 대한 분석 및 모니터링 등으로 연동 및 확장되어 활용될 수 있을 것이다.

참고문헌

- [1] ISO/IEC International Standard 13818-1, "Information technology - Generic coding of moving pictures and associated audio information: Systems", 2000.
- [2] Steven Morris, "Interactive TV Standards", 2005.
- [3] Hongguang Zhang, Tianpu Jiang, Zhiqi Gu, and Shibao Zheng, "Design and Implementation of Broadcast File System Based on DSM-CC Data Carousel Protocol", International Conference on Consumer Electronics, 2004.
- [4] C.Peng and P.Vuorimaa, "DIGITAL TELEVISION APPLICATION MANAGER", 2001 IEEE International Conference on Multimedia and Expo.
- [5] John Cosmas, Alan Lucas, Kannan Krishnapillai and Mohammed Akhtar, "STORAGE MANAGER SYSTEM FOR DVB TERMINALS", PG Net Conference 2001.

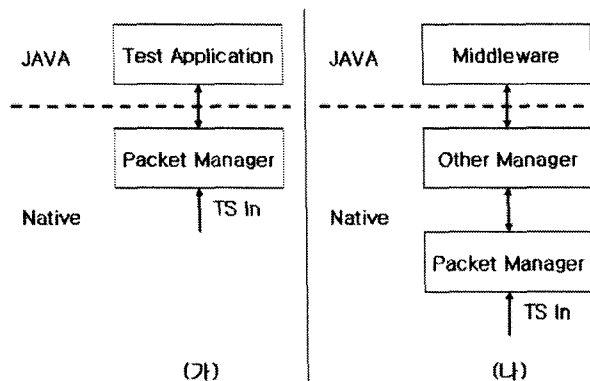


그림 5 (가) 테스트 환경 (나) 패킷 매니저 적용 환경