

데이터 방송 환경에서 서비스 정보 추출을 위한 효율적인 SI Manager 구현

최현석*, 전제민*, 김두호*, 김정선*
*한양대학교 컴퓨터공학과
e-mail: tksanga@gmail.com

An Efficient SI Management for Extracting Service information in Data Broadcasting

Hyeon-Seok Choi*, Jae-Min Jeon*, Doo-Ho Kim, JungSun Kim*
*Dept of Computer Science, Han-Yang University

요 약

데이터 방송에 사용되는 주요한 정보로는 SI(Service Information)가 있다. SI는 수신기와 사용자에게 유용한 정보를 제공하는 여러 개의 테이블들로 구성되어 있으며, 각각이 트랜스포트 스트림을 통하여 수신기로 전송된다. 수신기에서 SI 테이블을 처리하기 위한 작업은 모니터링과 파싱의 2가지로 나누어지며, 실시간 처리를 위하여 파싱 작업은 각각의 쓰레드를 생성하여 처리한다. 하지만 SI는 빈번하게 전송되고, 그 크기가 작기 때문에 쓰레드의 생성과 소멸이 반복해서 발생하게 된다. 이러한 문제는 대부분의 수신기가 자원이 부족한 임베디드 환경임을 고려하였을 때 효율적이지 못하다. 본 논문에서는 이러한 문제를 해결하기 위한 효율적인 SI Manager의 구조를 제안한다.

1. 서론

데이터 방송이 처음 도입된 이후로 오늘날의 데이터 방송은 단순한 텍스트 정보 제공을 위주로 하는 기초적인 수준에서 벗어나 프로그램 가이드, 날씨, 교통 정보 등의 다양한 부가 정보를 제공한다.

MPEG-2 TS(Transport Stream)에서는 이러한 부가 정보를 전달하기 위한 표준 형식으로 SI(Service Information) Table을 이용한다. SI Table은 포함하고 있는 정보에 따라 다양한 종류로 나눌 수 있으며, MPEG-2에서 정의한 4개의 기본적인 Table 외에 MHP, OCAP, ACAP 등 다양한 데이터 방송 표준 규격에서 해당 방송 환경에 특화된 SI Table들을 추가로 정의하고 있다.

일반적으로 SI Table을 처리하기 위한 과정은 현재 들어오고 있는 스트림에서 SI Table을 모니터링 하는 것과 Table을 파싱하는 2개의 과정으로 나누어지며, 파싱하는 과정에서 발생할 수 있는 데이터의 손실을 최소화하기 위하여, 이 두 과정을 별도의 쓰레드로 처리하는 방법이 제안되었다 [4].

하지만 제한된 자원을 갖고 있는 데이터 방송 수신기의 특성상, SI 파싱 처리를 위한 쓰레드를 반복적으로 생성, 소멸시키는 것은 매우 비효율적이다. 따라서 본 논문에서

는 파싱 속도와 자원의 소모량을 고려하여 데이터 방송 환경에 최적화된 SI table 파싱 쓰레드 풀 모델을 제안하였다.

본 논문의 구성은 다음과 같다. 먼저 2장에서는 SI와 관련된 표준과 SI 처리를 위한 관련 연구에 대하여 살펴본다. 3장에서는 본 논문에서 제안하는 SI Manager의 구조를 설명한다. 마지막 4장에서는 결론 및 향후 연구과제에 대하여 기술한다.

2. 관련연구

2.1. 데이터 방송을 위한 SI 표준

모든 데이터 방송 표준의 기반이 되는 MPEG-2에서는 SI를 위하여 PSI(Program Specific Information)를 사용한다[1]. PSI는 PAT(Program Association Table), PMT(Program Map Table), NIT(Network Information Table), CAT(Conditional Access Table), TSDDT(Transport Stream Description Table)로 5개의 테이블로 구성되어 있다. PAT는 TS에 포함되어 있는 프로그램 각각에 관련된 PMT 패킷 정보로 구성이 되며, PMT는 한 프로그램을 구성하는 비디오, 오디오, 데이터 등을 가지고 있는 패킷에 대한 정보로 구성된다. CAT는 프로그램을 시청할 권리를 가진 사람만 볼 수 있게 하는 접근 제어 정보로 구성된다. NIT는 주어진 네트워크 속성들에 대한 정보와 네트워크를 통하여 운반되어 지는 TS들에 대한

이 논문은 2008년도 정부(교육과학기술부)의 재원으로 한국과학재단의 지원을 받아 수행된 연구임(No. R01-2008-000-11234-0(2008))

물리적인 구조에 관한 정보로 구성되며, TSDT는 트랜스포트 스트림에 대한 정보 기술자로 구성이 되며, 선택적으로 사용된다.

유럽 디지털 방송 표준인 DVB-MHP는 SI를 위하여 DVB-SI를 사용한다. 이 DVB-SI는 방송 프로그램에 대한 개별 정보를 전송하기 위하여 PSI 이외에 NIT를 포함한 3개의 필드 테이블과 6개의 선택적인 테이블을 추가하였다[2].

ATSC(American Television Systems Committee)는 SI를 위하여 PSIP(Program and System Information Protocol)를 사용한다. PSIP는 DVB-SI와 마찬가지로 방송 프로그램에 대한 개별 정보를 전송하기 위하여 PSI 이외에 9개의 테이블을 추가하였다[3].

2.2. Separate-filtering

Lin는 모니터링을 위한 쓰레드와 파싱을 위한 쓰레드를 분리한 모델인 Separate-filtering 기법을 제안하였다[4]. 이것은 SI 테이블이 파싱되는 동안 발생하는 패킷의 손실을 줄여주는 기법이다.

하지만 여러 개의 SI 테이블이 빈번하게 전송되기 때문에 파싱을 위한 쓰레드는 생성과 소멸을 반복해야 한다. 이는 수신기가 임베디드 시스템을 고려할 때, 쓰레드 생성과 소멸에 따르는 오버헤드는 무시할 수 없게 된다. 또한 다른 상위 어플리케이션과 미들웨어에서 많은 쓰레드가 사용 중일 경우 SI 테이블에 대한 처리가 느려지는 문제점이 발생한다.

3. SI 추출

3.1. SI 테이블 구조

<표 1> table_id assignment values[1]

Value	description
0x00	program_association_section
0x01	conditional_access_section
0x02	TS_program_map_section
0x03	TS_description_section
0x04	ISO_IEC_14496_scene_description_section
0x05	ITU-T Rec. H.222.0/ISO/IEC 13818-1 reserved
0x06-0x37	Defined in ISO/IEC 13818-6
0x40-0xFE	User private
0xFF	forbidden

SI 테이블의 크기는 TS 패킷의 크기보다 크기 때문에 여러 개의 패킷으로 나누어 전송된다. 수신기에서는 나누어진 패킷을 합하여 섹션을 만들고, 생성된 섹션의 table_id를 기반으로 테이블 정보를 알 수 있다. PAT는 0x00, PMT는 0x02, CAT는 0x01, TSDT는 0x03의 table_id를 가진다. 표 1은 DVB-SI에서 table_id와 SI 테

이들의 관계를 보여준다.

섹션은 테이블을 전송하기 위한 단위로서 테이블은 여러 개의 섹션으로 나누어져 전송된다. 각 섹션의 구조는 포함된 테이블에 따라 그 구조가 다르지만, 공통적인 몇 개의 필드를 갖는다. 그 중 last_section_number는 테이블을 구성하는 마지막 섹션의 ID를 가지고 있으므로, 이를 분석함으로써 해당 테이블이 몇 개의 섹션으로 구성되어야 할지를 알 수 있다. 다른 공통된 필드로는 section_length, section_number 등이 있으며, 각각 섹션의 크기와 ID를 나타낸다. 표 2는 테이블 중 NIT를 가지고 있는 섹션의 구조이다.

<표 2> Network Information 섹션의 구조

Syntax	Number of bits	Identifier
network_information_section(){		
table_id	8	uimbsf
section_syntax_indicator	1	bslbf
reserved_future_use	1	bslbf
reserved	2	bslbf
section_length	12	uimbsf
network_id	16	uimbsf
reserved	2	bslbf
version_number	5	uimbsf
current_next_indicator	1	bslbf
section_number	8	uimbsf
last_section_number	8	uimbsf
reserved_future_use	4	bslbf
network_descriptors_length	12	uimbsf
for(i=0;i<N;i++){		
descriptor()		
}		
reserved_future_use	4	bslbf
transport_stream_loop_length	12	uimbsf
for(i=0;i<N;i++){		
transport_stream_id	16	uimbsf
original_network_id	16	uimbsf
reserved_future_use	4	bslbf
transport_descriptors_length	12	uimbsf
for(j=0;j<N;j++){		
descriptor()		
}		
}		
CRC_32	32	rpchf
}		

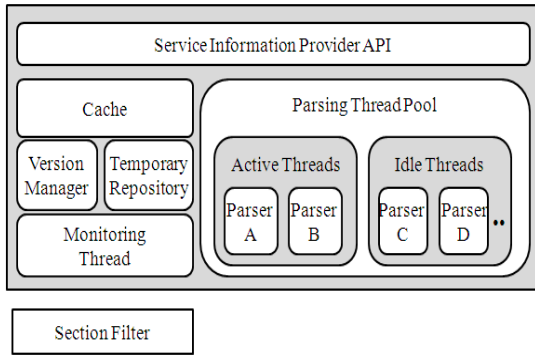
3.2. SI Manager

본 논문에서 제안하는 효율적인 SI Manager의 구조는 (그림 1)과 같다. Separation filter 기법을 도입하여 파싱 과정 중 발생할 수 있는 패킷 분실 현상을 최소화하였다.

모니터링 쓰레드는 섹션 필터를 이용하여 새로운 SI 테이블 섹션이 수신되었는지 모니터링 작업을 수행한다. 만약 SI 테이블 섹션이 수신되었다면, 모니터링 쓰레드는 테이블을 구성하기 위해 필요한 섹션의 개수를 검사한다. 만약 필요한 섹션의 개수가 1개라면 이 섹션은 곧바로 파싱 쓰레드로 전달되어 처리가 이루어진다. 반면, 2개 이상의 섹션을 필요로 할 경우에는 Temporary Repository에 필요한 섹션이 모두 모일 때까지 저장된다.

또한 효율적인 쓰레드 관리를 위하여 쓰레드 풀을 도입하였다. 쓰레드 풀은 요청이 있을 때 마다 쓰레드를 생성하는 것이 아니라, 미리 일정 개수만큼 생성하여 두어 필요시 이미 생성된 쓰레드를 사용하는 프로그래밍 기법으로 쓰레드의 생성과 소멸에 있어서 효율적인 방법을 제공

한다[5].



(그림 1) SI Manager

쓰레드 풀 모델을 사용할 경우, 풀 내의 쓰레드의 개수가 시스템의 성능에 많은 영향을 준다. 만약 쓰레드의 수가 충분하지 못할 경우, 파싱 작업이 원활히 진행되지 문제가 발생하고, 반대로 너무 많을 경우 자원이 낭비되기 때문에 쓰레드 개수의 조절이 필요하다.

본 논문에서는 풀 내의 쓰레드 개수를 유동성 있게 조절하기 위하여 초기에는 1개의 Idle쓰레드를 생성하였다. 만약, 동시에 처리해야할 파싱 작업이 2개 이상인 경우 부족한 만큼의 쓰레드를 생성하여 처리하고 Idle 상태로 변경되게 하였다. 이를 통해 빈번한 쓰레드 생성과 소멸에 따른 오버헤드를 줄였으며, 최대 쓰레드의 수도 현 시점에 동시에 처리해야 하는 파싱 작업의 수에 맞추어 변경되도록 하여 효율적으로 쓰레드를 관리하였다.

4. 결론 및 향후 연구 과제

본 논문에서는 데이터 방송 환경에서 SI 테이블을 파싱을 위한 효율적인 SI Manager를 제안하였다. 패킷 손실을 줄이기 위하여 모니터링 쓰레드와 파싱 쓰레드를 분리하였다. 또한 파싱 쓰레드가 빈번히 생성됨에 따른 오버헤드를 줄이기 위하여 쓰레드 풀을 도입하였고, 시스템에 따른 환경을 고려하여 풀 내의 쓰레드의 개수를 최적화하여 효율적인 SI Manager를 구현하였다.

이 논문에서 제안한 효율적인 SI 처리 모델은 MHP 뿐만 아니라 SI 처리를 필요로 하는 OCAP, ACAP 등의 다른 데이터 방송 환경에서도 이용될 수 있을 것이다.

참고문헌

[1] ISO/IEC International Standard 1318-1 "Generic coding of moving pictures and associated audio information: Systems", Second edition, 2000.
 [2] ETSI EN 300 468 V1.7.1 "Specification for Service Information (SI) in DVB systems", 2005
 [3] ATSC Standard A/65 "Program and System

Information Protocol". 2006

[4] Feng Lin, Jun Sun, "An Interactive Media Platform Scheme for DTV Receiver Compliant with MHP", IEEE Transaction on Consumer Electronics, Vol.53, NO.4, Nov. 2007.

[5] <http://www.chrisarndt.de/projects/threadpool>, "Easy to use object-oriented thread pool framework".

[6] Steven Morris, et al., "Interactive TV Standards". 2005.