

일반논문 (Regular Paper)

방송공학회논문지 제25권 제2호, 2020년 3월 (JBE Vol. 25, No. 2, March 2020)

<https://doi.org/10.5909/JBE.2020.25.2.252>

ISSN 2287-9137 (Online) ISSN 1226-7953 (Print)

MPEG-2 TS로부터 MMTP 스트림으로의 변환기 개발

박 민 규^{a)}, 김 용 한^{b)†}

A Development of MPEG-2 TS-to-MMTP Stream Converter

MinKyu Park^{a)} and Yong Han Kim^{b)†}

요 약

2017년 5월 31일 세계 최초로 우리나라에서 지상파 UHD 방송이 상용화되었다. 기존 HDTV 방송은 압축된 미디어와 부가 데이터를 다중화하여 전달하기 위해 MPEG-2 TS(Transport Stream) 표준을 사용하였으나, 지상파 UHD 방송 규격에서는 차세대 표준인 MMT(MPEG Media Transport)가 채택되었다. 그러나 UHD 콘텐츠 제작 비용이 고가이므로, 현재 전체 방송시간 중 일부만 UHD로 방송되고 있으며 점차로 그 시간을 증가시켜 가고 있는 중이다. 한편 MMT가 채택된 ATSC 3.0 표준의 경우에도 북미에서 본격적인 상용화 단계에 이르지 못하고 있다. 이로 인해 MMT 방송 장비는 아직도 고가의 초기 단계에 있다. 본 논문에서는 기존 MPEG-2 TS 다중화기의 출력을 입력으로 받아 MMTP(MMT Protocol) 패킷 스트림으로 변환하는 저렴한 장비를 개발하는 데에 활용할 수 있는 소프트웨어를 다중 쓰레드(multi-thread) 기반으로 일반 PC 상에서 구현하고 그 기능을 실험을 통해 검증하였다.

Abstract

Korea has launched the world-wide first terrestrial UHD broadcast services on May 31, 2017. While the existing HDTV broadcast services use MPEG-2 TS (Transport Stream) standard for multiplexing and delivering compressed media with additional data, the terrestrial UHD broadcast services use MMT (MPEG Media Transport) standard, which is the next-generation standard beyond MPEG-2 TS. However, the production cost of UHD contents is so high that only a part of the total broadcast time is filled with UHD contents and the UHD time portion is planned to be gradually increased. On the other hand, the ATSC 3.0 standard that uses MMT is not yet used in full-fledged broadcast services in North America. Hence MMT broadcast equipment is still at an early stage with high prices. In this paper we implemented a multi-thread software running on an ordinary PC that can be utilized to realize a low-cost converter that converts the output of an existing MPEG-2 TS multiplexer to an MMTP (MMT Protocol) packet stream. We also verified the functionality of the software through experiments.

Keywords : MMT, MPEG-2 TS, Broadcast protocol conversion

a) LG전자(LG Electronics)

b) 서울시립대학교 일반대학원 전자전기컴퓨터공학과(Department of Electrical and Computer Engineering, Graduate School, University of Seoul)

† Corresponding Author : 김용한(Yong Han Kim)

E-mail: yhkim@uos.ac.kr

Tel: +82-2-6490-2330

ORCID: <https://orcid.org/0000-0001-9470-6060>

※ 이 논문은 2016년도 서울시립대학교 교내학술연구비에 의하여 지원되었음.(This work was supported by the 2016 Research Fund of the University of Seoul.)

· Manuscript received February 3, 2020; Revised March 16, 2020; Accepted March 16, 2020.

Copyright © 2020 Korean Institute of Broadcast and Media Engineers. All rights reserved.

“This is an Open-Access article distributed under the terms of the Creative Commons BY-NC-ND (<http://creativecommons.org/licenses/by-nc-nd/3.0>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited and not altered.”

I. 서론

현재 DTV 및 HDTV 방송에는 압축된 미디어를 부가 데이터와 함께 다중화하여 전송하기 위해 대부분 MPEG-2 TS(Transport Stream) 표준^[1]이 채택되어 사용되고 있다. UHDTV 시대가 도래하면서, MPEG(Moving Picture Experts Group)에서는 MPEG-2 TS 표준의 다음 세대 미디어 다중화 및 전달 표준으로서 MMT(MPEG Media Transport) 표준^[2]을 완성하였고, 현재 국내에서 세계 최초로 상용화된 지상파 UHDTV 표준^[3]과 미국의 ATSC 3.0 표준^[4]에 MMT가 채택되어 있다.

비록 우리나라가 세계 최초로 2017년 5월 31일 지상파 UHD 방송을 개시하였다고는 하나 현재까지도 콘텐츠 제작 비용 문제로 인해 전체 방송 시간 중 일부만 UHD로 방송하고 있으며 이 시간을 점차 늘려가고 있는 단계이고, 북미의 경우에는 아직 ATSC 3.0 표준에 의한 상용화가 본격화되지 않고 있다. 따라서 아직도 UHD 방송 및 관련 산업은 초기 단계를 벗어나고 있지 못하여 관련 장비도 고가에 출시되고 있는 상황이다. 아직도 하드웨어로 구현된 MMT 방송 장비는 고가이므로 실험용이나 소규모 방송에 사용할 수 있는 저렴한 MMT 방송 장비를 구현하기 위해서는 기존 MPEG-2 TS 다중화기의 출력을 MMTP(MMT Protocol) 패킷 스트림으로 변환할 수 있는 저렴한 장비가 필요하다.

본 논문에서는 MPEG-2 TS 파일로부터 TS 패킷들을 읽어 들여 MMTP 패킷 스트림으로 변환하여 새로운 파일에 저장하는 방법을 제안하고, 이를 검증하기 위한 변환 소프트웨어를 PC 상에서 구현한 후, 실험을 통해 검증을 시행하였다. 본 논문에서 구현한 변환 소프트웨어는 일반 PC에서 실행되므로 기존 MPEG-2 TS 다중화기의 출력을 입력으로 받아 MMTP 패킷 스트림으로 변환하여 UDP/IP로 전송하는 장비를 최소한의 비용으로 구성하는 데에 쉽게 활용될 수 있다.

본 논문의 구성은 다음과 같다. II장 및 III장에서는 각기 MPEG-2 TS 표준과 MMT 표준의 개요에 대해 본 논문의 내용을 이해할 수 있는 정도의 수준으로 설명하고, IV장에서는 MPEG-2 TS 파일로부터 MMTP 패킷 스트림 파일로의 변환 방법에 대해 설명하며, V장에서는 이를 구현하여 실험을 통해 검증한 결과를 설명한 후, IV장에서 결론 및

향후 과제를 제시한다.

II. MPEG-2 TS 표준의 개요

MPEG-2 TS(이하 TS라 줄임)는 MPEG-2 시스템에서 AV(Audio and Video) 데이터 및 기타 데이터를 다중화하여 하나의 스트림(stream)으로 전송하기 위한 통신 규약이다. MPEG-2 TS는 188바이트의 고정길이의 패킷을 사용하며, 방송 환경에서 주로 사용된다. TS 내에서 서로 다른 패킷 사이에는 빈 공간이 없다. 즉, 앞선 패킷에 이어서 바로 다음 패킷이 나오는 형태의 스트림이다.

한 패킷은 크게 4바이트의 고정 헤더, 확장 헤더에 해당하는 적응 필드(adaptation field), 그리고 유료부하(payload)로 구성된다. 패킷 헤더에는 값이 0x47인 8비트의 동기 바이트(sync byte)와 13비트의 패킷 아이디(Packet Identifier, PID)가 포함되며, 그 외 1비트의 유료부하 단위 시작 지시자(payload_unit_start_indicator, PUSI), 2비트의 적응 필드 제어(adaptation field control) 등이 포함된다. 동기 바이트는 TS에서 프레임링(framing) 기능을 제공한다. PID는 TS 패킷의 유료부하 내용을 쉽게 식별하는 데 사용된다. 즉 TS의 각 테이블(table)이나 기초 스트림(Elementary Stream, ES)을 식별하는 데 사용된다. PSUI는 해당 TS 패킷의 유료부하가 테이블을 실어 나르는 섹션(section) 또는 ES를 실어 나르는 PES(Packetized Elementary Stream) 패킷의 첫 부분임을 나타낸다. 적응 필드 제어 값에 따라 적응 필드와 유료부하는 독립적으로 생략 가능하다. 적응 필드에는 27Mhz의 PCR(Program Clock Reference)이 포함되며, 이는 수신 측에서 한 프로그램을 구성하는 ES들을 복호 후 동기화하여 재생할 때, 즉 속칭 AV 립싱크(lip-sync)를 수행할 때 사용하는 시스템 타임 클록(System Time Clock, STC)을 복원하는 데 필요한 클록 참조값이다. PCR은 실제로 송신 측 STC를 0.1초 이내의 주기로 주기적으로 샘플링하여 전송되는 값으로서, 수신 측에서는 이를 이용하여 송신 측 STC와 동일한 빠르기를 갖는 클록을 복원하여 AV 동기화를 수행하는 데 사용한다.

TS에서는 수신기에서 역다중화하기 위한 필요 정보를 테이블 형식으로 정의하였으며, 이를 프로그램 지정 정

보(Program Specific Information, PSI)라고 한다. 대표적인 PSI 테이블로는 프로그램 연결 테이블(Program Association Table, PAT), 프로그램 맵 테이블(Program Map Table, PMT), 접근 제어 정보 테이블(Conditional Access Table, CAT), 네트워크 정보 테이블(Network Information Table, NIT) 등이 있다. CAT에는 유료수신을 위한 수신제어 정보를 넣을 수 있고, NIT에는 TS가 실릴 물리적 채널에 대한 정보를 넣을 수 있는데, 본 논문에서는 스트림 다중화와 AV 동기화 등의 TS 기본 기능에 대한 MMTP로의 변환을 목표로 하므로 PAT와 PMT만을 다룬다. 테이블은 섹션이라는 구조에 담은 후, TS 패킷의 유료부하에 실어 전송한다. 물론 한 섹션의 크기가 한 TS 패킷의 유료부하 크기보다 클 때에는, 여러 TS 패킷에 나누어 싣는다. 만약 하나의 테이블의 크기가 섹션의 최대 크기인 1,024바이트보다 크다면 이를 여러 섹션에 나누어 싣어야 한다. 테이블의 크기가 1,024바이트보다 작더라도 필요에 따라 이를 여러 섹션으로 나눌 수도 있다. 또한 하나의 TS 패킷에 한 섹션 끝머리와 또 다른 섹션의 첫머리를 함께 넣는 형태로, 서로 다른 두 섹션의 데이터를 하나의 TS 패킷에 실을 수도 있는데, 이때는 두 섹션의 데이터를 쉽게 분리할 수 있도록 해당 TS 패킷의 유료부하 중 첫 바이트를 포인터 필드(pointer field)로서 사용하며 여기에 두 번째 섹션의 첫 바이트 위치에 대한 바이트 오프셋(offset)을 넣어 준다.

TS는 오디오, 비디오 그리고 부가 데이터로 구성되는 프로그램 단위로 구성되며, TS 내에는 하나 또는 그 이상의 프로그램들이 포함될 수 있다. PAT에는 해당 TS에 들어있는 모든 프로그램에 대해 각기 프로그램 번호(program number)와 PMT PID 정보가 들어 있다. 여기서 PMT PID란 PMT를 실어 나르는 TS 패킷의 PID를 말한다. PAT PID 즉, PAT를 실어 나르는 TS 패킷의 PID는 0번으로 TS 표준에서 고정되어 있다. 이는 수신 측에서 TS에 들어 있는 프로그램에 대한 정보를 얻기 위해서 가장 먼저 해독해야 하는 정보가 PAT에 들어 있기 때문에 별도의 시그널링(signaling) 없이도 수신 측에서 이를 획득할 수 있게 하기 위함이다. 수신 측에서 TS 내의 특정 프로그램을 추출하고자 할 때, 우선 PAT를 수신하여 해독함으로써 해당 프로그램의 PMT PID를 알아내고 다시 이를 이용하여 PMT를 수신하여 해독하게 된다.

PMT에는 한 프로그램을 구성하는 모든 ES에 대한 정보가 들어 있다. PMT에는 우선 PCR PID가 들어 있는데, 이는 해당 프로그램에 적용될 PCR을 실어 나르는 TS 패킷의 PID를 지정한다. 그 다음 PMT의 구문(syntax)에는 2개의 구문 루프(loop)가 존재하는데 첫 번째는 프로그램 전체에 대한 메타데이터 서술자(descriptor)들의 구문 루프이고, 두 번째 구문 루프에 해당 프로그램에 포함된 ES들의 정보가 들어 있다. 이러한 ES 정보는, 각 ES에 대해, 이를 실어 나르는 TS 패킷의 PID와 ES의 종류 및 압축부호화 방식을 나타내는 스트림 타입(stream type), 추가적인 메타데이터 서술자들을 포함할 수 있는 구문 루프 등을 포함한다.

수신기에서는 오디오와 비디오를 재생할 때 동기화를 달성하기 위한 시간 정보가 필요하다. ES를 TS 패킷에 싣기 전에, 우선 PES 패킷화를 시행한다. 이 때 PES 패킷 헤더에는 해당 PES 패킷의 유료부하에 담긴 ES 데이터를 복호할 시간과 표출할 시간을 지정하는 DTS(Decoding Time Stamp)와 PTS(Presentation Time Stamp)를 넣는다. 만약 DTS와 PTS가 동일하면, DTS는 보통 생략한다. 실제로 DTS와 PTS가 적용되는 ES 데이터 단위는 AU(Access Unit)이며, 이에 상응하는 비압축 원 데이터는 PU(Presentation Unit)이다. 예를 들어 비디오의 경우, 한 프레임은 PU이며 이를 압축하여 얻은 데이터는 하나의 AU에 해당하며, 오디오의 경우, 일정 개수의 일련의 오디오 샘플들이 PU가 되고 이를 압축하여 얻은 데이터를 하나의 AU로 간주한다. DTS와 PTS의 기준은 STC이다. AV 압축 전송 시스템에 있어 AV 동기화의 목표는 여러 미디어 데이터 별로 송신 측 부호기에 입력되는 시점에서부터 수신 측 복호기에서 출력되는 시점까지의 경과 시간을 모두 동일하게 유지하는 것이다. TS에서는 이를 위해 송신 측의 STC를 수신 측에서 복원할 수 있도록 PCR을 주기적으로 전송하고, 여러 미디어의 PU가 송신 측 부호기로 입력될 때 STC 값을 샘플링하여 해당 AU의 PTS로서 수신 측으로 전송한다. 수신 측에서는 PCR을 주기적으로 수신하여 송신 측의 STC와 동일한 빠르기를 갖는 클록을 복원하고, 수신된 AU로부터 복원한 PU를 PTS가 지정하는 시점에서 플레이함으로써 AV 동기화를 달성한다. PES 패킷에는 여러 AU가 함께 포함될 수 있으며, 이 경우 PES 패킷 헤더의 PTS는 해당 PES 패킷 유료부하에서 그 시작 부분이 첫 번째로 등장하는 AU에

적용되며, 두 번째 이후에 등장하는 AU의 PTS는 비디오 프레임 및 오디오 샘플들을 활용하여 첫 번째 AU의 PTS로부터 계산해 낸다.

PES 패킷의 최대 길이는 그 헤더의 구문 요소 중 PES_packet_length가 표현 가능한 최대 길이인 64KB이다. 비디오 AU에 한하여 만약 그 길이가 이보다 클 경우에는 PES_packet_length의 값을 0으로 설정하여 64KB보다 큰 비디오 AU를 하나의 PES 패킷으로 구성할 수 있도록 하였다. 오늘날 고화질 비디오 AU의 크기는 대부분 64KB보다 크기 때문에, PES_packet_length의 값을 0으로 설정하여 PES 패킷으로 구성할 때가 많다. 이 경우, 패킷 길이 정보가 없으므로, PES 패킷 계층의 하위 계층인 TS 패킷 계층의 도움을 받아야 해당 PES 패킷의 끝을 인지할 수 있다. TS 패킷 계층에 이를 위한 기능이 있음을 아래에서 설명하기로 한다. 이렇게 구성된 PES 패킷은 보통 여러 TS 패킷의 유효부하에 나뉘어 실리게 된다. 단, 한 개의 TS 패킷에는 하나의 PES 패킷으로부터 유래된 데이터만 실도록 표준에 규정되어 있다. 이에 따라, 한 PES 패킷의 첫 바이트는 어떤 TS 패킷의 유효부하의 첫 바이트에 실어야 하고, 한 PES 패킷의 마지막 바이트는 어떤 TS 패킷의 유효부하의 마지막 바이트에 해당되도록 한다. 후자의 규칙은 TS 패킷의 헤더와 유효부하 사이에 있는 적응 필드 내에 필요한 수만큼 채워넣기(stuffing) 바이트(0xFF)를 넣음으로써 항상 지킬 수 있다. 전자의 경우, TS 패킷 헤더 내의 PUSI를 1로 설정하게 함으로써, 상기 후자의 규칙과 함께 사용하면, TS 패킷 계층에서 TS 패킷 단위로 PES 패킷들의 시작과 끝을 구분할 수 있게 하였다.

III. MMT 표준의 개요

MMT는 UHDTV 및 인터넷 시대에 맞춰 HEVC(High Efficiency Video Coding)를 지원하며, All-IP(All-Internet Protocol) 네트워크를 사용하여 데이터를 전송할 수 있도록 설계되었다.

MMT 표준에서는 미디어 처리 단위(Media Processing Unit, MPU) 포장 형식, 시그널링 메시지(signaling message), 그리고 전달 프로토콜을 규정하고 있다. 이와 함께

프레젠테이션 정보(Presentation Information, PI)를 제공함으로써 미디어 구성 요소들 간의 시공간적 관계를 정의할 수 있게 하였다. 이 PI로는 MMT와 같이 표준화된 MMT CI(Composition Information)^[5]를 HTML5(Hyper Text Markup Language 5)와 함께 사용하면 가장 효율적이나, 필요시 다른 형식의 화면 구성 정보도 사용할 수 있도록 설계되었다. MMT CI는 HTML5가 공간적인 요소만 정의하고 있기 때문에 시간적인 정보도 함께 지원하고자 MPEG에서 새롭게 정의하였다.

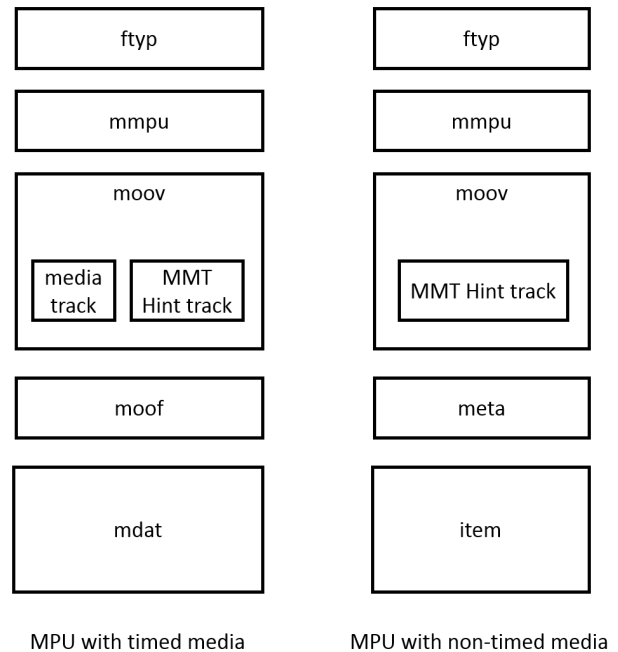


그림 1. MPU의 구조^[2]
Fig. 1. The structure of MPU^[2]

MMT는 단일 포장(encapsulation) 포맷으로 ISO 기본 미디어 파일 포맷(ISO Base Media File Format, ISO/BMFF)^[6] 기반의 MPU를 채택하였다. 그림 1은 MPU의 구조를 나타낸다. 어떤 파일 내에 포장된 MPU는 전 세계적으로 유일무이하게 식별할 수 있도록, MPU의 고유번호와 그것이 속한 애셋(asset)의 애셋 ID가 'mmpu' 박스 내에 제공된다. 또한 시간 지정 미디어(timed media)의 경우, MPU를 구성하는 무비 프래그먼트(movie fragment)에 대한 순위를 제공하기 위해, 포장된 MPU 내에 'sidx' 박스가 포함될 수 있

다. 포장된 MPU 내의 ‘moov’ 박스는 MPU에 포함된 모든 미디어 데이터를 복호하여 출력하는 데에 필요한 코덱 정보를 포함하며, 단일 미디어 트랙으로 구성된다. 포장된 MPU를 MMTP 유료부하(payload)로 변환하고, MMTP 패킷화하는 데 필요한 정보들은 MMT 힌트 트랙(hint track)에 포함된다.

그림 2는 MPU를 MMTP 유료부하에 실는 방법을 나타낸다. 하나의 MPU는 MPU 메타데이터(MPU metadata), 프래그먼트 메타데이터(fragment metadata), MFU(Media Fragment Unit) 유형의 데이터 단위(data unit) 등으로 구성된다. MMTP의 유료부하에는 한 가지 유형의 데이터 단위만 실을 수 있다. MFU 유형의 MMTP 유료부하에는 하나 또는 둘 이상의 MFU들이 실릴 수 있다. 그림 2와 같이 하나의 MPU는 한 개의 MPU 메타데이터와 여러 개의 프래그먼트 메타데이터들과 여러 개의 MFU들로 구성된다. 프래그먼트 메타데이터에는 moov 박스와 mdat 박스의 앞부분(박스 이름 및 박스 크기를 포함)이 포함된다. 그리고 MFU의 앞부분에는 해당 미디어 샘플 또는 슬라이스에 대한 힌트 샘플(hint sample)이 추가된다. 힌트 샘플은 MFU들의 경계를 알려주는 역할을 하며, MMT 힌트 트랙(hint track)에 포함되어 있다. 즉, MMT 송신 측에서 MPU로부터 MFU들을 구분해 내는 데에 필요한 정보를 MMT 힌트 트랙에서 제공하며, 이를 바탕으로 MPU 내의 미디어 데이터는 전달될 시점에서 MPU로부터 추출되어 힌트 샘플과 함께 MMTP 유료부하에 실리게 된다. MPU의 이런 기능은

미디어의 전달 포맷과 저장 포맷간의 변환을 쉽게 하므로, 네트워크 자원을 보다 더 효율적으로 사용할 수 있다.

MMT에서는 미디어 서비스에 대한 다중화 정보 및 제어 정보를 시그널링 메시지를 통해 전달한다. 시그널링 메시지는 이진(binary) 형식 또는 XML(Extensible Markup Language)의 두 가지 다른 형태로 정의된다. 각각의 형태는 장단점이 있기 때문에 상황에 맞게 사용하면 된다. 그림 3은 패킷 소비를 위한 시그널링 메시지와 이에 포함될 수 있는 테이블들을 나타낸다. 시그널링 메시지에는 PA(Package Access), MPI(Media Presentation Information), MPT(MMT Package Table), DCI(Device Capability Information), CRI(Clock Relation Information) 메시지 등이 존재하며, 테이블에는 메시지와 마찬가지로 PA, MPI, MP, DCI, CRI 테이블 등이 존재한다.

다음은 패키지 소비를 위해서 정의된 다섯 가지 유형의 시그널링 메시지이다.

- 패키지 접근(Package Access, PA) 메시지: PA 테이블을 포함한다. PA 테이블은 어떤 패키지에 접근하기 위해 요구되는 모든 시그널링 테이블에 대한 정보를 갖고 있다. 신속한 패키지 정보 획득을 위해 이 테이블들 자체를 PA 메시지에 모두 포함한다.
- 미디어 프레젠테이션 정보(Media Presentation Information, MPI) 메시지: PI 문서 전체 또는 일부(계층화 전달 시)를 포장하는 MPI 테이블을 포함한다. 또한 신

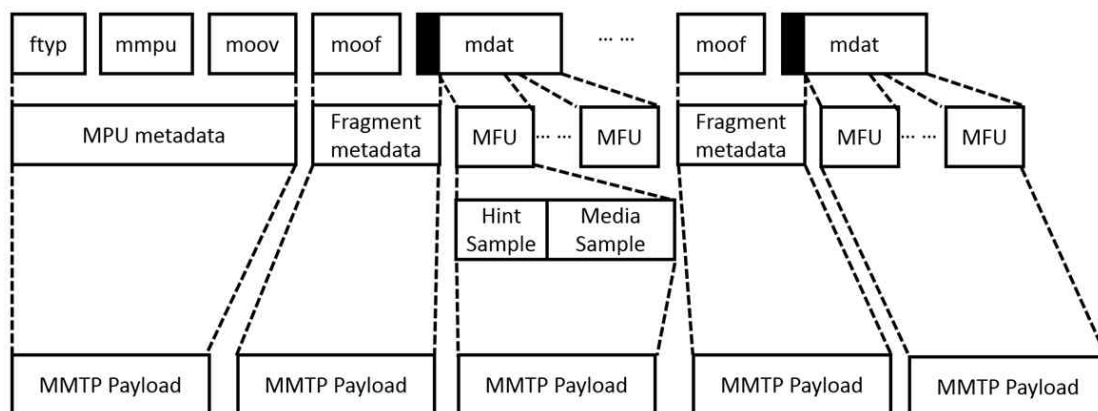


그림 2. MPU를 MMTP 유료부하에 실는 방법^[2]

Fig. 2. The method of carrying an MPU as MMTP payloads^[2]

속한 패키지 정보 획득을 위해 MPI 테이블에 상응하는 MP 테이블을 포함할 수 있다.

- MMT 패키지 테이블(MMT Package Table, MPT) 메시지: MP 테이블을 포함한다. MP 테이블은 하나의 패키지를 소비하기 위해 요구되는 모든 정보 또는 일부 정보(계층화 전달 시)를 포함한다.
- 클럭 관계 정보(Clock Relation Information, CRI) 메시지: CRI 테이블을 포함한다. CRI 테이블은 NTP (Network Time Protocol) 타임스탬프(timestamp)와

MPEG-2 STC를 짝짓기하는 데 사용되는 클럭 관계 정보를 포함한다. CRI 메시지와 CRI 테이블은 MPEG-2 시스템의 ES를 MMT 패키지의 애셋으로서 사용할 때만 사용한다.

- 기기 능력 정보(Device Capability Information, DCI) 메시지: 이 유형의 메시지는 DCI 테이블을 포함한다. DCI 테이블은 패키지를 소비하기 위해 요구되는 수신기 능력에 대한 정보를 제공한다.

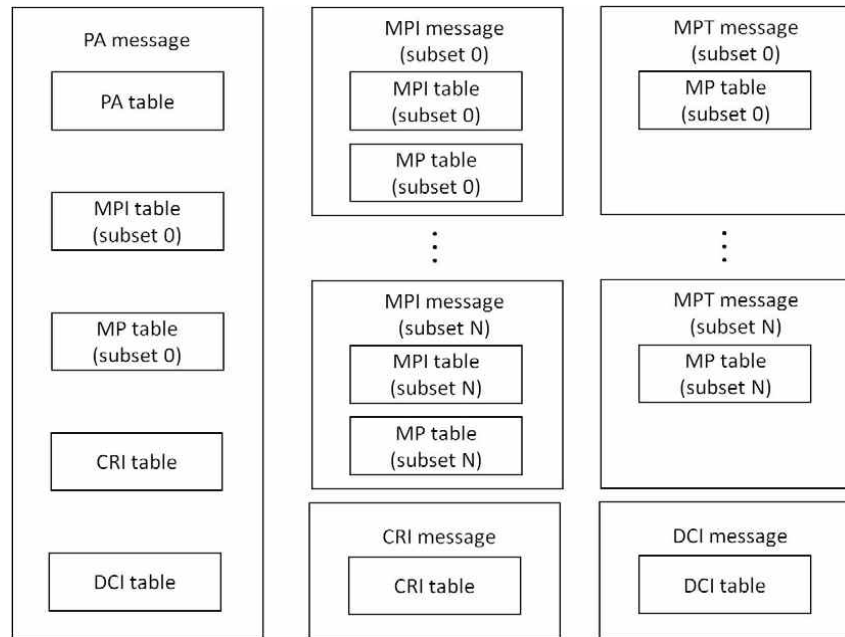


그림 3. 패키지 소비를 위한 MMT 시그널링 메시지와 테이블의 구조^[2]

Fig. 3. Structure of the MMT signaling messages and tables for package consumption^[2]

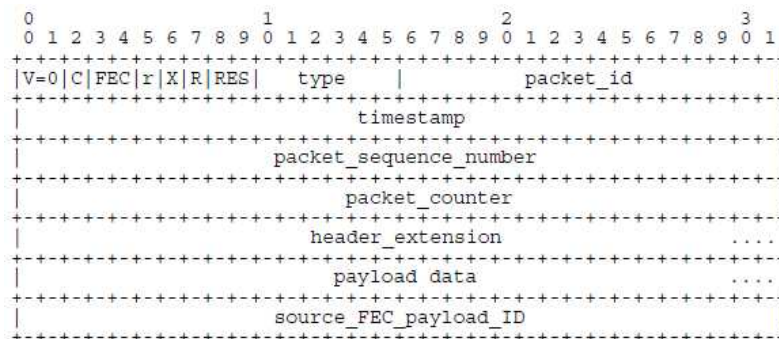


그림 4. MMTP 패킷의 구조(버전: 0)^[2]

Fig. 4. Structure of the MMTP packet (version: 0)^[2]

MMT는 패킷 헤더 압축 기능을 사용하는 패킷과 그렇지 않는 패킷에 대하여 정의하였다. 그림 4는 패킷 헤더 압축 기능을 사용하지 않는 MMTP 패킷의 구조를 나타낸다.

현재 MMTP 패킷의 유료부하가 어떤 유형인지에 따라서 타입(type) 필드가 정의된다. 가능한 유형으로는 시그널링 메시지, MPU, GFD(Generic File Delivery)가 있다. 여기서 GFD는 일반 파일을 MMTP 패킷 스트림에 포함하여 전달할 때에 사용하는 것으로 본 논문에서 다루지 않는다.

그림 5와 그림 6은 패킷 헤더 압축 기능을 사용하는 MMTP 패킷의 구조이다. MMT에서 패킷 헤더 압축 기능을 사용하기 위해서는 그림 5의 최대 크기 헤더(full size header)와 그림 6의 축소한 헤더(reduced header)를 동시에 사용한다. 최대 크기 헤더는 MMTP 패킷 헤더의 모든 정보

를 담고 있으며, 이는 수신기에서 축소한 헤더를 수신하여 복원하는 데 기준이 되는 참조 데이터 역할을 한다. 따라서 MMT에서는 최대 크기 헤더를 주기적으로 전송하도록 규정하고 있다.

IV. MPEG-2 TS로부터 MMTP 패킷 스트림으로의 변환 방법

본 절에서는 MPEG-2 TS로부터 MMTP 패킷 스트림으로의 변환 방법을 제안하고, 이를 검증할 소프트웨어 구현에 대해 설명한다. 설명의 편의상, 소프트웨어 구현 내용과 병행하여 제안한 방법을 함께 설명하도록 한다. 본 논문

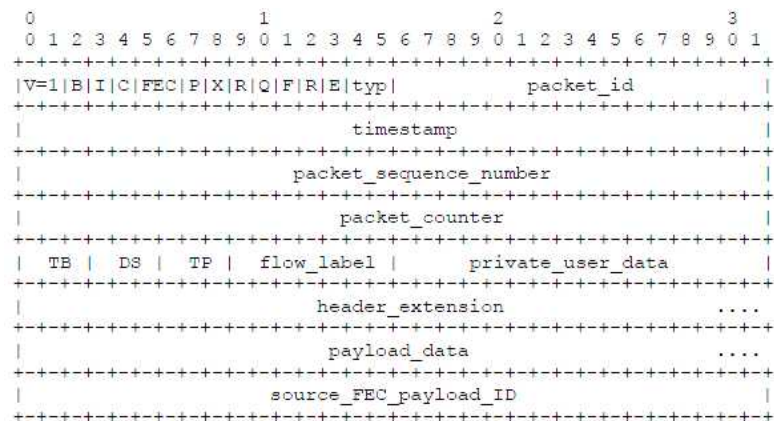


그림 5. MMTP 패킷의 구조(버전: 1, B: 0)^[2]

Fig. 5. Structure of the MMTP packet (version: 1, B: 0)^[2]

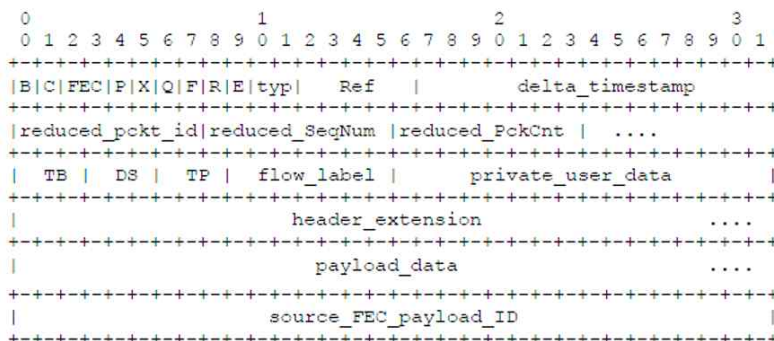


그림 6. MMTP 패킷의 구조(버전: 1, B: 1)^[2]

Fig. 6. Structure of the MMTP packet (version: 1, B: 1)^[2]

서 구현한 MPEG-2 TS로부터 MMTP 패킷 스트림으로의 변환 소프트웨어(이하 ‘변환 소프트웨어’로 줄임)는 윈도우 7 64비트 운영체제 하에서 C++언어로 구현하였다. 개발 도구로 마이크로소프트 비주얼 스튜디오 2012를 사용하였으며, QT 프레임워크^[7] 기반으로 개발하였다.

본 논문에서 구현한 변환 소프트웨어는 MMT의 시그널링 테이블들을 하나의 PA 메시지에 담아 전달하며, MMTP 헤더 압축 기능도 지원한다. 또한 TS에 여러 프로그램이 들어 있는 경우에는 그 중 한 프로그램을 사용자가 지정하여 MMTP 패킷 스트림 파일로 변환 저장할 수 있다.

그림 7은 변환 소프트웨어에 대한 구조를 보여준다. 본 논문에서 구현한 변환 소프트웨어는 MPEG-2 TS 파일을 읽어 들이는 것부터 MMTP 패킷 스트림(이하 ‘MMTP 스트림’으로 줄임) 파일을 생성하기까지의 일련의 과정들을 모듈화하였다. 그리고 각각의 모듈들을 쓰레드(thread)로서 인스턴스(instance)화하고 관련된 쓰레드들을 버퍼(buffer)로 연결하였다. 변환 과정은 순차적으로 진행되므로 반드시 이와 같이 다중 쓰레드(multi-thread) 형태로 구성할 필요는 없으나 이렇게 함으로써 소프트웨어적으로 동시에 공정한 처리가 가능하게 하여 시스템 개발의 효율을 높일도록 하였다. 이런 쓰레드들의 생성과 관리는 사용자가 제공하는 설정 정보 파일(Configuration Information File)에 포함된 정보로부터 대부분 이루어진다.

변환 소프트웨어는 설정 정보 파일을 입력으로 받아 TS에서 MMTP 스트림으로의 변환을 시작한다. 설정 정보에는 TS 파일 위치와 HTML5 문서 및 CI 문서 위치가 포함된다. 더불어 MMT 메시지 구성에 필요한 정보와 소스(source) 데이터가 되는 TS의 ES들을 애셋으로 변환할 때 필요한 정보 그리고 이런 정보들에 대한 관계 정보가 포함된다. 또한 미리 TS 파일을 분석하여 그 결과를 반영한 정보도 함께 포함된다. 이런 정보의 예로는 비디오 및 오디오 ES에 대한 asset_type, mime_type, 비디오 해상도(resolution), 비트율(bitrate), 버퍼 크기, 오디오 MPU의 지속 시간(duration) 등이 있다. 비디오 MPU의 지속 시간은 MPEG-2 TS 비디오 ES의 GOP(Group of Picture) 시간으로 고정하였다. 오디오 MPU의 지속 시간은 보유한 지상파 HDTV 샘플 MPEG-2 TS 파일의 비디오 GOP 지속 시간이 대부분 0.5초이기 때문에 기본 값으로 0.5초로 설정하였다. 그 이유는 비디오와 오디오의 MPU 지속 시간이 비슷해야 효율적으로 MMT 클라이언트가 MMTP 스트림을 전송받아 미디어 서비스를 진행할 수 있기 때문이다.

설정 정보 파일은 XML 형식으로 규칙에 맞게 작성하였으며, 그림 8은 설정 정보 파일의 예로서 그 내용은 다음과 같다. TS 파일(‘./res/rec_10310723pm_2.ts’)을 입력으로 받아 이를 MMTP 스트림 파일로 변환하여 저장한다. TS 파일에서 비디오와 오디오 ES를 추출하며, 이를 참고하는 애

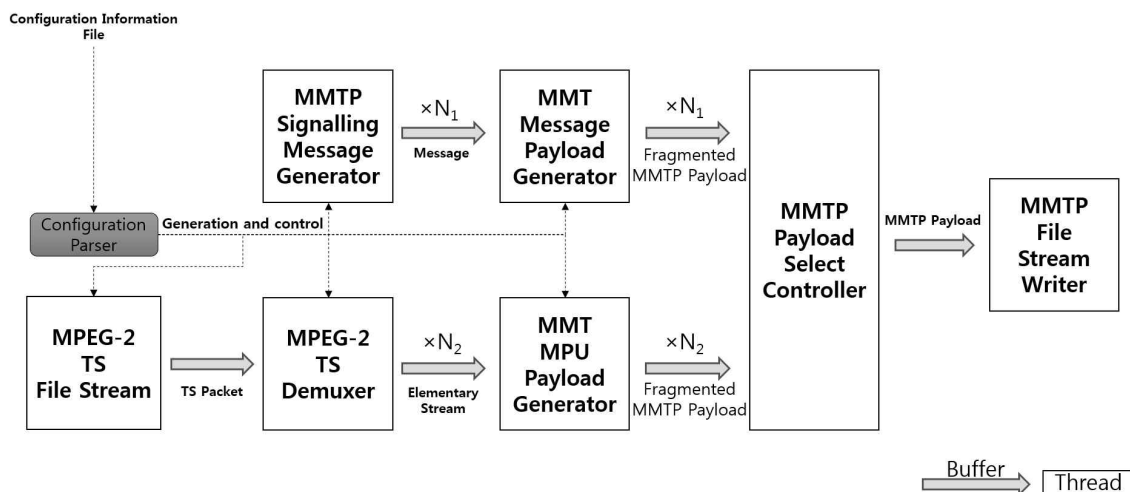


그림 7. MPEG-2 TS로부터 MMTP 스트림으로의 변환 소프트웨어에 대한 구조도
Fig. 7. Structural diagram for the MPEG-2 TS-to-MMTP stream conversion software

```

<?xml version="1.0"?>
<root>
  <Channels>
    <File target="ts" type="System" src="./res/rec_10310723pm_2.ts"/>
  </Channels>
  <MPEG2_System id="ts">
    <RefMessage ref="message_pa"/>
    <RefAsset type="Video" ref="VideoAsset"/>
    <RefAsset type="Audio" ref="AudioAsset"/>
  </MPEG2_System>
  <MMT_ASSET id="VideoAsset" asset_id_scheme="UUI" asset_id_byte="Video1" asset_type="mp4v"
mime_type="video/mp4" asset_clock_relation_flag="0">
    <MMT_general_location_info packet_id="40" location_type="0"/>
    <MMT_Codec_Complexity minimum_buffer_size="17400" temporal_resolution="0"
vertical_resolution="1920" horizontal_resolution="1080" maximum_bitrate="17400"
average_bitrate="17400"/>
  </MMT_ASSET>
  <MMT_ASSET id="AudioAsset" asset_id_scheme="UUI" asset_id_byte="Audio1" asset_type="mp4a"
mime_type="audio/mp4" asset_clock_relation_flag="0">
    <MMT_Codec_Complexity minimum_buffer_size="384" maximum_bitrate="384" average_bitrate="384"/>
    <MMT_general_location_info packet_id="50" location_type="0"/>
  </MMT_ASSET>
  <MMT_MESSAGE type="PA" id="message_pa" period="400" version="0" message_id="0">
    <MMT_general_location_info packet_id="0" location_type="0"/>
    <MMT_TABLE type="PA" id="table_0" version="0" private_extension_flag="0" table_id="0">
      <MMT_general_location_info location_type="7"/>
      <RefTable ref="table_16"/>
      <RefTable ref="table_32"/>
      <RefTable ref="table_34"/>
    </MMT_TABLE>
    <MMT_TABLE type="MPI" id="table_16" version="0" table_id="16" PI_mode="1">
      <MMT_general_location_info location_type="7"/>
      <MMT_PI_CONTENT refResource="html" PI_content_type_byte="application/html"
PI_content_name_byte="mux-1.html"/>
      <MMT_PI_CONTENT refResource="ci" PI_content_type_byte="application/mmt+xml"
PI_content_name_byte="mux-1.xml"/>
    </MMT_TABLE>
    <MMT_TABLE type="MP" id="table_32" version="0" table_id="32">
      <MMT_general_location_info location_type="7"/>
      <RefAsset ref="VideoAsset" identifier_type="0"/>
      <RefAsset ref="AudioAsset" identifier_type="0"/>
    </MMT_TABLE>
    <MMT_TABLE type="DCI" id="table_34" version="0" table_id="34">
      <MMT_general_location_info location_type="7"/>
      <RefAsset ref="VideoAsset"/>
      <RefAsset ref="AudioAsset"/>
    </MMT_TABLE>
  </MMT_MESSAGE>
  <Resources>
    <File src="./res/mux-1.html" id="html"/>
    <File src="./res/mux-1.xml" id="ci"/>
  </Resources>
</root>

```

그림 8. 설정 정보 파일의 예

Fig. 8. An example of the configuration information file

셋의 아이디는 'VideoAsset'과 'AudioAsset'이다. 'Video-Asset'을 전송하는 MMTP 패킷의 아이디는 '40'이고 'Audio-Asset'을 전송하는 MMTP 패킷의 아이디는 '50'이다. MMT 메시지는 PA 메시지만 생성되며, 이를 전송하는 MMTP 패킷의 아이디는 '0'이다. PA 메시지는 0.4초 주기로 생성되며, PA 테이블, MPI 테이블, MP 테이블, DCI 테이블 등이 이에 포함된다. MPI 테이블에는 두 개의 문서(HTML5 문서 및 CI)가 포함되며, 각각의 위치는 './res/mux-1.html'과 './res/mux-1.xml'이다. 만약 지상파 UHDTV 표준과 같이 MP 메시지만을 사용하도록 변환하기를 원한

다면, 'MMT_MESSAGE' 노드를 MP 메시지에 맞게 수정하여 사용한다.

그림 8의 설정 정보 파일을 예로 들어 그림 7의 각 쓰레드의 기능을 설명하면 다음과 같다. MMTP 시그널링 메시지 생성부(MMTP Signalling Message Generator) 쓰레드는 설정 정보 파일에 정의된 MMT 메시지 생성 정보에 따라 MMTP 시그널링 메시지를 생성한다. 그림 8의 설정 정보 파일에는 PA 메시지만 정의되어 있기 때문에 쓰레드 하나만 생성된다(그림 7의 N1=1). 따라서 뒤에 연결되는 MMT 메시지 유료부하 생성부(MMT Message Payload Genera-

tor) 쓰레드 역시 하나만 생성된다. 메시지 생성은 0.4초 주기로 이루어지며, 그 기준 시간은 MPEG-2 TS 역다중화부(MPEG-2 TS Demuxer) 쓰레드에서 PMT를 분석 완료한 시점의 STC에 상응하는 시점으로 하였다. 이는 MMTP 유료부하 선택 제어부(MMTP Payload Select Controller) 쓰레드가 자신에게 입력되는 여러 가지 MPU 데이터들 중에서 한 가지를 선택하여 버퍼로 출력할 때, 지정된 주기 이내에 MMT 메시지 유료부하를 우선적으로 출력할 수 있도록 하기 위함이다.

MMT 메시지 유료부하 생성부 쓰레드는 MMTP 시그널링 메시지 생성부 쓰레드로부터 전달된 MMT 메시지를 MMT 메시지 유료부하로 구성하는 역할을 한다. 설정 정보 파일에 정의된 MMTP 패킷 최대 크기에 따라 MMT 메시지 유료부하를 구성하며, 이보다 크기가 큰 MMT 메시지가 전달된다면 이를 자동으로 분할(fragmentation)한다. 여기서, MMTP 패킷 최대 크기 기본값은 1,472바이트로 설정하였는데, 이는 기본적인 MTU(Maximum Transmission Unit) 크기인 1,500바이트로부터 UDP/IP 패킷 헤더의 크기를 제외한 크기이다. 분할된 MMT 메시지 유료부하들의 시간 정보는 해당 MMT 메시지를 전달받을 때 함께 전달받은 시간 정보로 동일하게 설정하여 MMTP 유료부하 선택 제어부 쓰레드로 전달한다. 그 이유는 MMT 메시지가 중요한 데이터이기 때문에 뒷단 쓰레드인 MMTP 유료부하 선택 제어부 쓰레드에서 하나의 MMT 메시지로부터 분할된 MMT 메시지 유료부하들을 한꺼번에 우선적으로 선택하여 MMTP 파일 스트림 작성부(MMTP File Stream Writer) 쓰레드 측으로 전달하기 위함이다.

MPEG-2 TS 파일 스트림(MPEG-2 TS File Stream) 생성부 쓰레드는 설정 정보 파일에 정의된 TS 파일 위치 정보를 바탕으로 TS 파일로부터 TS 패킷들을 패킷 단위로 분리하여 뒷단인 MPEG-2 TS 역다중화부(Demuxer) 쓰레드로 전달하는 역할을 한다. 이 쓰레드의 기능은 MPEG-2 TS 역다중화부에 포함되어도 상관없지만, 차후에 있을 수 있는 소프트웨어 확장(다른 MPEG-2 TS 플랫폼과의 연동)을 위해서 기능을 분리하여 모듈화하였다.

MPEG-2 TS 역다중화부 쓰레드는 전달받은 TS 패킷들을 분석하여 그 유료부하로부터 추출한 ES 스트림을 미디어 프레임 단위로 뒷단인 MMT MPU 유료부하 생성부

(MMT MPU Payload Generator)로 전달하는 역할을 한다. 앞단 쓰레드에서 한 번에 전달하는 TS 패킷 수에 상관없이 동작하도록 구현하였으며, 패킷을 잃어버렸거나 패킷 오류가 발생하였을 경우도 대비하여 구현하였다. TS에 포함된 ES의 수에 따라 MMT MPU 유료부하 생성부 쓰레드가 생성된다(그림 7에서 N2로 표기). 이에 관련된 내용을 설정 정보 파일에 미리 기입하도록 하여, 설정 정보 파일의 내용에 따라 MPEG-2 TS 역다중화부 쓰레드와 MMT MPU 유료부하 생성부 쓰레드들을 버퍼를 통해 연결한다. 따라서 MPEG-2 TS 역다중화부 쓰레드의 출력 버퍼는 한 개 이상 존재하겠지만, TS에는 오디오와 비디오 ES만 존재하는 경우가 대부분이므로 대부분 두 개가 될 것이다. 각 미디어 프레임에 대한 표출 시간 정보(PTS)를 미디어 프레임을 전달할 때 함께 전달하도록 하여, 뒤에 위치한 쓰레드에서 MPU를 구성할 때 활용하도록 하였다.

MMT MPU 유료부하 생성부 쓰레드는 전달 받은 ES를 MPU로 구성하고, 이를 다시 MMT MPU 유료부하로 구성하여, 이를 MMTP 유료부하 선택 제어부 쓰레드로 전달하는 역할을 한다. MMT 메시지 유료부하 생성부 쓰레드에서 설명한 것과 동일하게 필요시 분할을 진행한다. 다만, MMTP 유료부하 선택 제어부 쓰레드로 전달할 때 설정하는 MMTP 유료부하에 대한 시간 정보는 MMT 설정 정보 파일에 기입된 비트율 및 전달받은 각 미디어 프레임들의 시간 정보를 종합적으로 계산하여 설정한다. 비디오 MPU는 기본적으로 GOP 단위로 구성하며, 오디오 MPU는 설정 정보 파일에 정의된 시간 단위(기본값 0.5초)로 구성한다. 이를 위해서 비디오 MPU인 경우 전달받은 미디어 프레임이 I(Intra) 프레임인지 P(Predicted) 프레임인지 또는 B(Bidirectional) 프레임인지 분석한다. 오디오 MPU의 경우 전달받은 미디어 프레임 정보와 설정 정보 파일에 기입된 정보를 바탕으로 각 미디어 프레임에 대한 지속 시간을 계산한다.

MMTP 유료부하 선택 제어부 쓰레드는 MMT 메시지 유료부하 생성부 쓰레드와 MMT MPU 유료부하 생성부 쓰레드로부터 전달받은 MMTP 유료부하들에 대해서 우선순위를 판단하여 한 MMTP 유료부하를 선택하여, 이를 MMTP 파일 스트림 작성부 쓰레드로 전달하는 역할을 한다. 우선순위를 판단할 때 사용한 방법으로 앞에서 설명한 MMTP 유료부하와 함께 전달된 시간 정보를 사용하였다.

MMTP 파일 스트림 작성부 쓰레드는 전달받은 MMTP 유류부하를 MMTP 패킷으로 구성하여, 파일로 저장하는 역할을 한다. 하지만 MMTP 패킷에는 프레이밍 기능 및 패킷 길이에 대한 정보가 없다. 따라서 본 논문에서는 저장하는 MMTP 패킷들을 구분하기 위해서 각 패킷 데이터 직전에 패킷 길이 정보를 추가로 포함하여 파일로 구성하였다. 이 형식은 변환된 MMTP 스트림을 실험 검증하기 위해 사용할 MMT 클라이언트^[8]에서 해석할 수 있도록 맞추었다.

V. 실험 결과

그림 9는 본 논문에서 구현한 변환 소프트웨어의 실행 화면을 캡처한 것으로서 TS를 MMTP 스트림으로 변환하고 있는 모습이다. 그림에서 보인 바와 같이, 설정 파일의 이름은 'mmt-8.xml'이며, 'mmt-8.mmt'라는 파일 이름으로 MMTP 스트림 파일을 생성한다. 그리고 MMTP 패킷의 최대 크기는 MTU 크기(보통 1,500바이트)에서 UDP/IP 헤더 크기를 뺀 1,472바이트로 설정하여 MMTP 스트림을 제작하고 있다.

변환된 MMTP 스트림에 대한 검증을 위해, [8]에서 개발한 MMT 클라이언트 프로그램을 이용하였다. 그림 10은 이

러한 MMT 클라이언트 프로그램의 구조도를 나타낸다. 본 논문에서는 그림 10에 표현된 MMT 클라이언트의 기능 중, 파일 형태로 저장된 MMTP 스트림으로부터 미디어를 읽어 재생하는 기능을 활용하였다. MMTP 스트림을 저장하는 파일의 포맷은 매우 단순하게 정의하였는데, 각 패킷을 저장하기 직전 해당 패킷의 바이트 길이를 먼저 저장하도록 하였다. 자세한 내용에 대해서는 [8]을 참조하도록 한다. 그림 11은 이 클라이언트를 활용하여 변환된 MMTP 스트림 파일을 성공적으로 재생하는 화면을 캡처한 것이다.

VI. 결 론

MMTP는 IP 친화적이기 때문에 오늘날과 같이 대부분의 네트워크가 IP화된 상황에서는 TS에 비해 여러 가지 네트워크에 대응하는 것이 매우 쉽다. 또한 IP의 경우 MTU의 크기가 TS의 패킷 크기보다 훨씬 크기 때문에 MMTP의 경우 패킷 헤더로 인한 유효전송률 손실을 줄일 수 있다. TS의 경우에는 AV 동기화를 위해 STC를 별도로 수신 측으로 전달하였으나, MMTP의 경우에는 NTP에 의한 월클록(wall clock)을 사용하여 AV 동기화를 달성하므로 AV 동기화 과정이 더 간단하다.

```

C:\Windows\system32\cmd.exe
Input file: "./mmt/mmt-8.xml"
Output file: "mmt-8.mmt"
Max MMTP Packet size: 1472
MMTP Maker start
count[ 10000 ] time[ 6.914 ] avg bitrate : 16.4886 Mbps
count[ 20000 ] time[ 13.197 ] avg bitrate : 17.2909 Mbps
count[ 30000 ] time[ 19.536 ] avg bitrate : 17.5219 Mbps
count[ 40000 ] time[ 25.811 ] avg bitrate : 17.6867 Mbps
count[ 50000 ] time[ 32.123 ] avg bitrate : 17.7684 Mbps
count[ 60000 ] time[ 38.441 ] avg bitrate : 17.8183 Mbps
count[ 70000 ] time[ 44.721 ] avg bitrate : 17.8703 Mbps
count[ 80000 ] time[ 51.04 ] avg bitrate : 17.8956 Mbps
count[ 90000 ] time[ 57.306 ] avg bitrate : 17.9368 Mbps
count[ 100000 ] time[ 63.65 ] avg bitrate : 17.9432 Mbps
count[ 110000 ] time[ 69.953 ] avg bitrate : 17.9584 Mbps
count[ 120000 ] time[ 76.256 ] avg bitrate : 17.9716 Mbps
count[ 130000 ] time[ 82.541 ] avg bitrate : 17.9877 Mbps
count[ 140000 ] time[ 88.849 ] avg bitrate : 17.9968 Mbps
count[ 150000 ] time[ 95.16 ] avg bitrate : 18.0038 Mbps
count[ 160000 ] time[ 101.458 ] avg bitrate : 18.0117 Mbps
count[ 170000 ] time[ 107.787 ] avg bitrate : 18.0138 Mbps
count[ 180000 ] time[ 114.078 ] avg bitrate : 18.0218 Mbps
count[ 190000 ] time[ 120.401 ] avg bitrate : 18.0242 Mbps
count[ 200000 ] time[ 126.697 ] avg bitrate : 18.0305 Mbps
count[ 210000 ] time[ 132.998 ] avg bitrate : 18.0352 Mbps
  
```

그림 9. MPEG-2 TS to MMTP 스트림 변환 소프트웨어 실행 화면

Fig. 9. A screen capture of the execution results of the MPEG-2 TS-to-MMTP stream conversion software

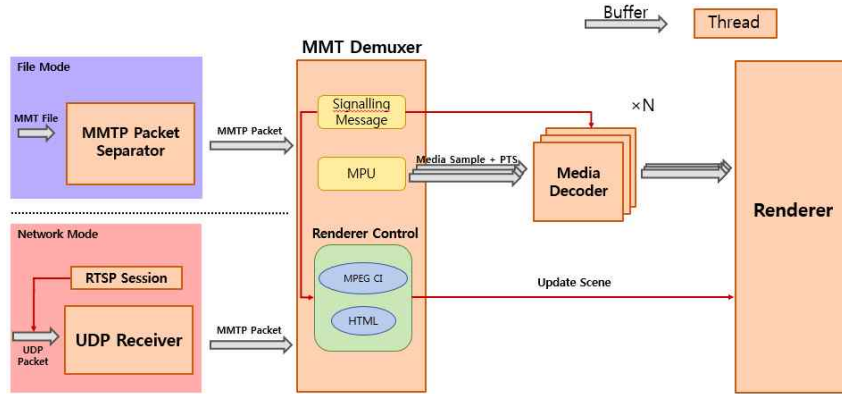


그림 10. [8]에서 개발된 MMT 클라이언트 구조도
Fig. 10. The MMT client architecture developed in [8]

본 논문에서는 기존 MPEG-2 TS 다중화기의 출력을 입력으로 받아 MMTP 패킷 스트림을 출력할 수 있는 저렴한 장비를 개발하는 데에 활용할 수 있는 방송 프로토콜 변환 방법을 제안하였다. 제안한 변환 방법을 PC 상에서 다중 쓰레드 기반으로 구현하고 그 기능을 실험을 통해 검증하였다.

본 논문에서 제안한 변환 방법에서는 변환 전후의 스트림 특성을 비슷하게 유지하기 위해 실험에 사용된 TS와 가능한 한 유사하게 MMTP의 인자들을 맞추도록 하였다. TS의 PAT 및 PMT와 유사한 정보인 MMTP의 PA 메시지에 대한 전송 주기는 0.4초로 함으로써 TS 표준에 적시된 0.5

초 이내 주기를 만족시키도록 하였다. 또한 MMTP MPU의 지속시간을 실험에 사용된 TS의 한 GOP에 해당하는 시간과 유사하게 0.5초로 맞추었다. 물론 이러한 인자들을 실제 입력된 TS의 그것에 맞게 자동으로 설정하는 기능은 향후 추가로 개발되어야 한다.

본 논문에서는 변환된 MMTP의 AV 동기화를 달성하기 위해 TS의 PTS를 NTP 타임스탬프로 변환하였다. TS의 첫 AU에 상응하는 NTP 타임스탬프를 현재 월클록 값으로 지정한 후, 이후의 AU들에 대해서는 PTS 값 차이를 월클록 시간 차이로 환산하여 순차적으로 NTP 타임스탬프에 더해

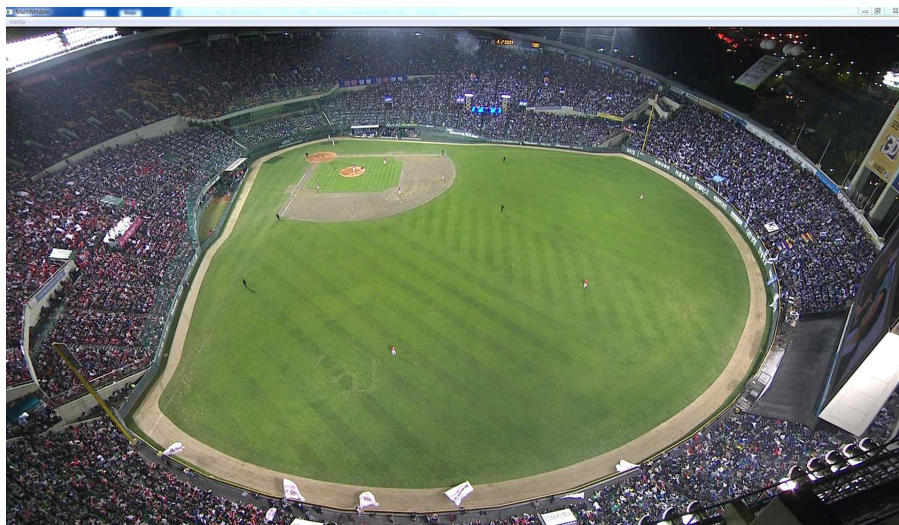


그림 11. 변환된 MMTP 스트림을 [8]에서 개발된 MMT 클라이언트로 재생한 화면
Fig. 11. A screen capture of the converted MMTP stream played by MMT client developed in [8]

중으로써 NTP 타임스탬프를 설정하도록 하였다.

본 논문에서 개발된 변환 소프트웨어는 MPEG-2 TS를 파일로부터 입력 받고 변환된 MMTP 패킷 스트림을 파일로 출력하는 형태로 구현되었다. 향후 실제 MPEG-2 TS 다중화기 출력을 입력 받아 본 논문에서 개발된 소프트웨어로 MMTP 패킷 스트림으로 변환한 후, UDP/IP 네트워크 인터페이스를 통해 출력하는 실제적인 장비 형태로 구현하는 과정이 추가로 필요하다.

참 고 문 헌 (References)

[1] ISO, *Information technology - Generic coding of moving pictures and*

associated audio information: Systems, Third Edition, ISO/IEC 13818-1, 2007.

- [2] ISO, *Information technology - High efficiency coding and media delivery in heterogeneous environments - Part 1: MPEG Media Transport*, ISO/IEC 23008-1, 2014.
- [3] TTA, *Transmission and Reception for Terrestrial UHDTV Broadcasting Service*, TTAK.KO-07.0127/R4, 2019.
- [4] ATSC, *Program and System Information Protocol for Terrestrial Broadcast and Cable*, ATSC Standard Document A/65, 2013.
- [5] ISO, *Information technology - High efficiency coding and media delivery in heterogeneous environments - Part 11: MPEG Media Transport Composition Information*, ISO/IEC 23008-11, 2015.
- [6] ISO, *Information technology - Coding of audio-visual objects - Part 12: ISO Base Media File Format*, ISO/IEC 14496-12, 2012.
- [7] <https://www.qt.io/> (accessed May 25, 2016).
- [8] J. Y. Jeong, *An implementation of Internet multimedia streaming by MMTP*, Master's Thesis of University of Seoul, Seoul, Rep. of Korea, 2015.

저 자 소 개



박 민 규

- 2010년 2월 : 서울시립대학교 공과대학 전자전기컴퓨터공학부 졸업 (공학사)
- 2012년 2월 : 서울시립대학교 대학원 전자전기컴퓨터공학과 졸업 (공학석사)
- 2016년 8월 : 서울시립대학교 대학원 전자전기컴퓨터공학과 (공학박사)
- 2016년 12월 ~ 2017년 6월 : (주)파넬피시큐어 선임연구원
- 2017년 7월 ~ 현재 : LG전자(주) CTO부문 A&B센터 스마트모빌리티연구소 선임연구원
- ORCID : <https://orcid.org/0000-0001-9254-9735>
- 주관심분야 : 멀티미디어 시스템



김 용 한

- 1982년 2월 : 서울대학교 제어계측공학과 (공학사)
- 1984년 2월 : 서울대학교 대학원 제어계측공학과 (공학석사)
- 1990년 12월 : 미국 렌슬리어공대(Rensselaer Polytechnic Institute, RPI) 전기컴퓨터시스템공학과 (Ph.D.)
- 1984년 3월 ~ 1996년 3월 : 한국전자통신연구원 책임연구원(최종)
- 1991년 10월 ~ 1992년 9월 : 일본 NTT 휴먼인터페이스연구소 객원연구원
- 1996년 3월 ~ 현재 : 서울시립대학교 전자전기컴퓨터공학부 교수
- 2017년 1월 ~ 2017년 12월 : 한국방송미디어공학회 회장
- 2014년 1월 ~ 현재 : 미래방송미디어표준포럼(구 차세대방송표준포럼) 의장
- 2000년 1월 ~ 현재 : MPEG뉴미디어포럼(구 MPEG포럼) 운영위원/자문위원
- ORCID : <https://orcid.org/0000-0001-9470-6060>
- 주관심분야 : 영상통신, 디지털 방송, 멀티미디어 부호화 및 전송