



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2022-0068636
(43) 공개일자 2022년05월26일

(51) 국제특허분류(Int. Cl.)

H04N 21/2362 (2011.01) H04N 21/2368

(2011.01)

H04N 21/434 (2011.01) H04N 21/438 (2011.01)

H04N 21/6437 (2011.01)

(52) CPC특허분류

H04N 21/2362 (2013.01)

H04N 21/2368 (2013.01)

(21) 출원번호 10-2020-0155623

(22) 출원일자 2020년11월19일

심사청구일자 2020년11월19일

(71) 출원인

경희대학교 산학협력단

경기도 용인시 기흥구 덕영대로 1732 (서천동, 경희대학교 국제캠퍼스내)

(72) 발명자

서덕영

경기도 성남시 분당구 수내로 174, 201동 202호(수내동, 푸른마을벽산아파트)

(74) 대리인

성병기

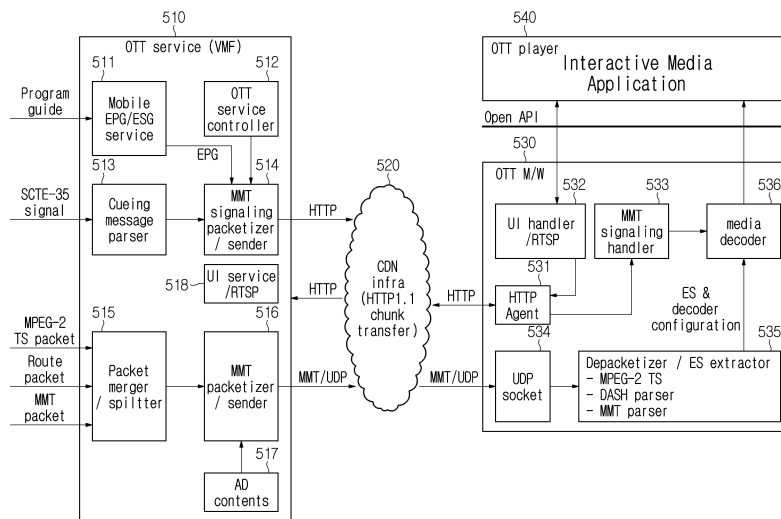
전체 청구항 수 : 총 13 항

(54) 발명의 명칭 초저지연 OTT 서비스를 제공하는 시스템 및 동작 방법

(57) 요약

본 발명은 OTT(over-the-top) 서비스를 제공하는 시스템 및 동작 방법에 관한 것으로서, OTT 서비스를 제공하는 서버의 동작 방법은, ATSC 3.0 서버로부터 상기 OTT 서비스의 프로그램에 관한 정보를 수신하는 단계; 제1 네트워크를 통해 상기 OTT 서비스의 프로그램에 관한 정보를 OTT 클라이언트에게 전송하는 단계; 상기 ATSC 3.0 서버로부터 미디어 데이터 패킷들을 수신하는 단계; 제2 네트워크를 통해 상기 미디어 데이터 패킷들을 상기 OTT 클라이언트에게 전송하는 단계; 및 상기 OTT 클라이언트에게 MPU 생성 요청 메시지를 전송하는 단계를 포함할 수 있다.

대표도



(52) CPC특허분류

H04N 21/4341 (2013.01)

H04N 21/4382 (2013.01)

H04N 21/6437 (2013.01)

H04N 21/64707 (2013.01)

이 발명을 지원한 국가연구개발사업

과제고유번호	1711103234
과제번호	IITP-2020-2015-0-00742-006
부처명	과학기술정보통신부
과제관리(전문)기관명	정보통신기획평가원
연구사업명	Grand ICT연구센터지원사업
연구과제명	라이프 컴패니온쉽 경험을 위한 지능형 인터랙션 융합 연구
기 여 율	1/1
과제수행기관명	경희대학교 산학협력단
연구기간	2020.01.01 ~ 2020.12.31

명세서

청구범위

청구항 1

OTT(over-the-top) 서비스를 제공하는 서버의 동작 방법으로서,
 ATSC 3.0 서버로부터 상기 OTT 서비스의 프로그램에 관한 정보를 수신하는 단계;
 제1 네트워크를 통해 상기 OTT 서비스의 프로그램에 관한 정보를 OTT 클라이언트에게 전송하는 단계;
 상기 ATSC 3.0 서버로부터 미디어 데이터 패킷들을 수신하는 단계;
 제2 네트워크를 통해 상기 미디어 데이터 패킷들을 상기 OTT 클라이언트에게 전송하는 단계; 및
 상기 OTT 클라이언트에게 MPU 생성 요청 메시지를 전송하는 단계를 포함하는, 동작 방법.

청구항 2

청구항 1에 있어서,
 상기 미디어 데이터 패킷 중 적어도 일부를 기초로 MPU(media processing unit) 헤더 데이터를 생성하는 단계를 더 포함하고,
 상기 미디어 데이터 패킷들을 상기 OTT 클라이언트에게 전송하는 단계는,
 상기 MPU 헤더를 더 전송하는, 동작 방법.

청구항 3

청구항 1에 있어서,
 상기 미디어 데이터 패킷을 UDP(user datagram protocol) 패킷 형태의MFU(media fragment unit)로 변환하는 단계를 더 포함하고,
 상기 미디어 데이터 패킷들을 상기 OTT 클라이언트에게 전송하는 단계는,
 상기 변환된 UDP 패킷 형태의 MFU를 전송하는, 동작 방법.

청구항 4

청구항 1에 있어서,
 상기 미디어 데이터 패킷 중 적어도 일부를 기초로 MPU(media processing unit) 테일 데이터를 생성하는 단계를 더 포함하고,
 상기 미디어 데이터 패킷들을 상기 OTT 클라이언트에게 전송하는 단계는,
 상기 MPU 테일을 더 전송하는, 동작 방법.

청구항 5

청구항 1에 있어서,
 상기 ATSC 3.0 서버로부터 미디어 데이터 패킷들을 수신하는 단계는,
 복수의 오디오 데이터 패킷 및 복수의 비디오 데이터 패킷을 교대로 수신하고,
 상기 미디어 데이터 패킷들을 상기 OTT 클라이언트에게 전송하는 단계는,
 상기 미디어 데이터 패킷 형태로 조합된 상기 복수의 오디오 데이터 패킷 및 상기 복수의 비디오 데이터 패킷들 각각을 인터리빙(interleaving) 방식으로 전송하는, 동작 방법.

청구항 6

청구항 1에 있어서,

상기 ATSC 3.0 서버로부터 미디어 데이터 패킷들을 수신하는 단계는,

복수의 오디오 데이터 패킷을 순차적으로 수신하는 단계; 및

복수의 비디오 데이터 패킷을 순차적으로 수신하는 단계를 포함하고,

상기 미디어 데이터 패킷들을 상기 OTT 클라이언트에게 전송하는 단계는,

상기 미디어 데이터 패킷 형태로 조합된 상기 복수의 오디오 데이터 패킷 및 상기 복수의 비디오 데이터 패킷들 각각을 비인터리빙(non-interleaving) 방식으로 전송하는, 동작 방법.

청구항 7

OTT(over-the-top) 서비스를 제공하는 클라이언트의 동작 방법으로서,

OTT 서버로부터 미디어 데이터를 포함하는 MFU(media fragment unit)들을 수신하는 단계;

상기 MFU들로부터 ES(elementary stream)들을 추출하는 단계;

상기 ES들을 디코딩함으로써 미디어 데이터를 획득하는 단계; 및

상기 획득한 미디어 데이터를 재생하는 단계를 포함하는, 동작 방법.

청구항 8

청구항 7에 있어서,

상기 OTT 서버로부터 MPU 생성 요청 메시지를 수신하는 단계;

상기 추출한 ES들을 병합함으로써, MPU를 생성하는 단계를 더 포함하고,

상기 MPU는,

CMAF(common media application format) 기반의 MPU인, 동작 방법.

청구항 9

청구항 7에 있어서,

상기 MFU들은,

MPU 헤더 데이터를 포함하는 제1 MFU; 및

MPU 테일 데이터를 포함하는, 제2 MFU를 포함하는, 동작 방법.

청구항 10

청구항 7에 있어서,

OTT 서버로부터 미디어 데이터를 포함하는 MFU들을 수신하는 단계는,

복수의 오디오 데이터 패킷 및 복수의 비디오 데이터 패킷들이 조합된 상기 MFU들 각각을 인터리빙(interleaving) 방식으로 수신하는, 동작 방법.

청구항 11

청구항 7에 있어서,

OTT 서버로부터 미디어 데이터를 포함하는 MFU들을 수신하는 단계는,

복수의 오디오 데이터 패킷 및 복수의 비디오 데이터 패킷들이 조합된 상기 MFU들 각각을 비인터리빙(interleaving) 방식으로 수신하는, 동작 방법.

청구항 12

OTT(over-the-top) 서비스를 제공하는 클라이언트에 있어서,
 송수신기; 및
 상기 송수신기에 결합된 프로세서를 포함하고, 상기 프로세서는,
 OTT 서버로부터 미디어 데이터를 포함하는 MFU(media fragment unit)들을 수신하고;
 상기 MFU들로부터 ES(elementary stream)들을 추출하고;
 상기 ES들을 디코딩함으로써 미디어 데이터를 획득하고; 그리고
 상기 획득한 미디어 데이터를 재생하는, 클라이언트.

청구항 13

청구항 12에 있어서,
 상기 프로세서는,
 상기 OTT 서버로부터 MPU 생성 요청 메시지를 수신하고; 그리고
 상기 추출한 ES들을 병합함으로써, MPU를 생성하고,
 상기 MPU는,
 CMAF(common media application format) 기반의 MPU인, 클라이언트.

발명의 설명

기술 분야

[0001] 이하의 설명은 OTT(over-the-top) 서비스 제공 방법 및 시스템에 관한 것으로, OTT 서비스에서 제공되는 미디어 파일을 MPU(media processing unit) 형식으로 변환하는 방법 및 장치에 관한 것이다.

배경 기술

[0002] 새로운 TV 표준인 ATSC 3.0에서는 MPEG-DASH과 MPEG MMT 등 인터넷 기반 멀티미디어 서비스 간의 연동을 용이하게 한다. OTT 서비스를 구현하기 위해, ATSC 3.0 TV 채널을 통해 전송된 TV 프로그램이나, 카메라에서 실시간으로 입력되는 스트림을 변환하고, 변환된 스트림을 MPEG-DASH 및/또는 MPEG MMT 표준에 따르는 클라이언트에서 소비할 수 있게 하는 중계 시스템이 필요하다.

발명의 내용

해결하려는 과제

[0003] 본 개시의 기술적 목적은 OTT(over-the-top) 서비스를 통해 제공되는 패킷들을 MPU(media processing unit) 형식에 따라 조합하고, MPU를 디코딩함으로써 획득한 미디어 데이터를 재생하는 클라이언트를 포함하는 OTT 서비스를 제공하는 시스템 및 동작 방법을 제공함에 있다.

[0004] 본 개시의 기술적 목적은 OTT 서비스를 제공함에 있어, 데이터의 특성 및 QoS(quality of service) 정보 등을 기초로 적응적으로 OTT 서비스를 제공하는 시스템 및 OTT 서비스 제공 방법을 제공함에 있다.

[0005] 본 개시에서 이루고자 하는 기술적 목적들은 이상에서 언급한 사항들로 제한되지 않으며, 언급하지 않은 또 다른 기술적 과제들은 이하 설명할 본 개시의 실시 예들로부터 본 개시의 기술 구성이 적용되는 기술분야에서 통상의 지식을 가진 자에 의해 고려될 수 있다.

과제의 해결 수단

[0006] 본 발명은 OTT 서비스를 제공함에 있어, 전송 지연을 최소화하는 것을 특징으로 한다.

[0007] 본 개시의 일 예로서, OTT(over-the-top) 서비스를 제공하는 서버의 동작 방법은, ATSC 3.0 서버로부터 상기

OTT 서비스의 프로그램에 관한 정보를 수신하는 단계; 제1 네트워크를 통해 상기 OTT 서비스의 프로그램에 관한 정보를 OTT 클라이언트에게 전송하는 단계; 상기 ATSC 3.0 서버로부터 미디어 데이터 패킷들을 수신하는 단계; 제2 네트워크를 통해 상기 미디어 데이터 패킷들을 상기 OTT 클라이언트에게 전송하는 단계; 및 상기 OTT 클라이언트에게 MPU 생성 요청 메시지를 전송하는 단계를 포함할 수 있다.

[0008] 본 개시의 일 예로서, 동작 방법은, 상기 미디어 데이터 패킷 중 적어도 일부를 기초로 MPU(media processing unit) 헤더 데이터를 생성하는 단계를 더 포함하고, 상기 미디어 데이터 패킷들을 상기 OTT 클라이언트에게 전송하는 단계에서는, 상기 MPU 헤더를 더 전송할 수 있다.

[0009] 본 개시의 일 예로서, 동작 방법은 상기 미디어 데이터 패킷을 UDP(user datagram protocol) 패킷 형태의 MFU(media fragment unit)로 변환하는 단계를 더 포함하고, 상기 미디어 데이터 패킷들을 상기 OTT 클라이언트에게 전송하는 단계에서는, 상기 변환된 UDP 패킷 형태의 MFU를 전송할 수 있다.

[0010] 본 개시의 일 예로서, 동작 방법은, 상기 미디어 데이터 패킷 중 적어도 일부를 기초로 MPU(media processing unit) 테일 데이터를 생성하는 단계를 더 포함하고, 상기 미디어 데이터 패킷들을 상기 OTT 클라이언트에게 전송하는 단계에서는, 상기 MPU 테일을 더 전송할 수 있다.

[0011] 본 개시의 일 예로서, 상기 ATSC 3.0 서버로부터 미디어 데이터 패킷들을 수신하는 단계에서는, 복수의 오디오 데이터 패킷 및 복수의 비디오 데이터 패킷을 교대로 수신하고, 상기 미디어 데이터 패킷들을 상기 OTT 클라이언트에게 전송하는 단계에서는, 상기 미디어 데이터 패킷 형태로 조합된 상기 복수의 오디오 데이터 패킷 및 상기 복수의 비디오 데이터 패킷들 각각을 인터리빙(interleaving) 방식으로 전송할 수 있다.

[0012] 본 개시의 일 예로서, 상기 ATSC 3.0 서버로부터 미디어 데이터 패킷들을 수신하는 단계에서는, 복수의 오디오 데이터 패킷을 순차적으로 수신하는 단계; 및 복수의 비디오 데이터 패킷을 순차적으로 수신하는 단계를 포함하고, 상기 미디어 데이터 패킷들을 상기 OTT 클라이언트에게 전송하는 단계에서는, 상기 미디어 데이터 패킷 형태로 조합된 상기 복수의 오디오 데이터 패킷 및 상기 복수의 비디오 데이터 패킷들 각각을 비인터리빙(non-interleaving) 방식으로 전송할 수 있다.

[0013] 본 개시의 일 예로서, OTT(over-the-top) 서비스를 제공하는 클라이언트의 동작 방법으로서, OTT 서버로부터 미디어 데이터를 포함하는 MFU(media fragment unit)들을 수신하는 단계; 상기 MFU들로부터 ES(elementary stream)들을 추출하는 단계; 상기 ES들을 디코딩함으로써 미디어 데이터를 획득하는 단계; 및 상기 획득한 미디어 데이터를 재생하는 단계를 포함할 수 있다.

[0014] 본 개시의 일 예로서, 상기 OTT 서버로부터 MPU 생성 요청 메시지를 수신하는 단계에서는, 상기 추출한 ES들을 병합함으로써, MPU를 생성하는 단계를 더 포함하고, 상기 MPU는, CMAF(common media application format) 기반의 MPU일 수 있다.

[0015] 본 개시의 일 예로서, 상기 MFU들은, MPU 헤더 데이터를 포함하는 제1 MFU; 및 MPU 테일 데이터를 포함하는, 제2 MFU를 포함할 수 있다.

[0016] 본 개시의 일 예로서, OTT 서버로부터 미디어 데이터를 포함하는 MFU들을 수신하는 단계에서는, 복수의 오디오 데이터 패킷 및 복수의 비디오 데이터 패킷들이 조합된 상기 MFU들 각각을 인터리빙(interleaving) 방식으로 수신할 수 있다.

[0017] 본 개시의 일 예로서, OTT 서버로부터 미디어 데이터를 포함하는 MFU들을 수신하는 단계에서는, 복수의 오디오 데이터 패킷 및 복수의 비디오 데이터 패킷들이 조합된 상기 MFU들 각각을 비인터리빙(interleaving) 방식으로 수신할 수 있다.

[0018] 본 개시의 일 예로서, OTT(over-the-top) 서비스를 제공하는 클라이언트는, 송수신기; 및 상기 송수신기에 결합된 프로세서를 포함하고, 상기 프로세서는, OTT 서버로부터 미디어 데이터를 포함하는 MFU(media fragment unit)들을 수신하고; 상기 MFU들로부터 ES(elementary stream)들을 추출하고; 상기 ES들을 디코딩함으로써 미디어 데이터를 획득하고; 그리고 상기 획득한 미디어 데이터를 재생할 수 있다.

[0019] 본 개시의 일 예로서, 상기 프로세서는, 상기 OTT 서버로부터 MPU 생성 요청 메시지를 수신하고; 그리고 상기 추출한 ES들을 병합함으로써, MPU를 생성하고, 상기 MPU는, CMAF(common media application format) 기반의 MPU일 수 있다.

발명의 효과

- [0020] 본 개시에 기초한 실시 예들에 의해 하기와 같은 효과가 있을 수 있다.
- [0021] 본 개시에 따르면, OTT(over-the-top) 서비스를 제공하는 OTT 서버는 MPU(media processing unit)를 구성하는 미디어 데이터 수신하는 즉시 별도의 인코딩 동작을 수행하지 아니하고, OTT 클라이언트에 전송할 수 있으며, 따라서, OTT 서버는 OTT 서비스를 제공함에 있어, 전송 지연을 최소화할 수 있다.
- [0022] 본 개시의 실시 예들에서 얻을 수 있는 효과는 이상에서 언급한 효과들로 제한되지 않으며, 언급하지 않은 또 다른 효과들은 이하의 본 개시의 실시 예들에 대한 기재로부터 본 개시의 기술 구성이 적용되는 기술분야에서 통상의 지식을 가진 자에게 명확하게 도출되고 이해될 수 있다. 즉, 본 개시에서 서술하는 구성을 실시함에 따른 의도하지 않은 효과들 역시 본 개시의 실시 예들로부터 당해 기술분야의 통상의 지식을 가진 자에 의해 도출될 수 있다.

도면의 간단한 설명

- [0023] 이하에 첨부되는 도면들은 본 개시에 관한 이해를 돕기 위한 것으로, 상세한 설명과 함께 본 개시에 대한 실시 예들을 제공할 수 있다. 다만, 본 개시의 기술적 특징이 특정 도면에 한정되는 것은 아니며, 각 도면에서 개시하는 특징들은 서로 조합되어 새로운 실시 예로 구성될 수 있다. 각 도면에서의 참조 번호(reference numerals)들은 구조적 구성요소(structural elements)를 의미할 수 있다.
- 도 1은 본 개시에서 적용되는 ATSC 3.0에 따른 OTT(over-the-top) 중계 시스템의 예시를 도시한 도면이다.
- 도 2는 본 개시에서 적용되는 CMAF(common media application format) 기반의 MPU(media processing unit)의 구조를 도시한 도면이다.
- 도 3a는 본 개시에서 적용되는 OTT 시스템에서 CMAF 기반의 MPU를 생성하는 동작의 제1 실시예를 도시한 도면이다.
- 도 3b는 본 개시에서 적용되는 OTT 시스템에서 CMAF 기반의 MPU를 생성하는 동작의 제2 실시예를 도시한 도면이다.
- 도 4는 본 개시에서 적용되는 서버와 클라이언트 간의 RTSP(real time streaming protocol)를 이용한 스트리밍 서비스 동작을 도시한 도면이다.
- 도 5는 본 개시에서 적용되는 OTT 서버와 OTT 클라이언트 간의 미디어 데이터 송수신 동작의 제1 실시예를 도시한 도면이다.
- 도 6은 본 개시에서 적용되는 OTT 서버와 OTT 클라이언트 간의 미디어 데이터 송수신 동작의 제2 실시예를 도시한 도면이다.
- 도 7은 본 개시에서 적용되는 OTT 서버의 MFU 전송 방식의 제1 실시예를 도시한 도면이다.
- 도 8은 본 개시에서 적용되는 OTT 서버의 MFU 전송 방식의 제2 실시예를 도시한 도면이다.

발명을 실시하기 위한 구체적인 내용

- [0024] 본 발명은 다양한 변경을 가할 수 있고 여러 가지 실시예를 가질 수 있는 바, 특정 실시예들을 도면에 예시하고 상세한 설명에 상세하게 설명하고자 한다. 그러나, 이는 본 발명을 특정한 실시 형태에 대해 한정하려는 것이 아니며, 본 발명의 사상 및 기술 범위에 포함되는 모든 변경, 균등물 내지 대체물을 포함하는 것으로 이해되어야 한다. 각 도면을 설명하면서 유사한 참조부호를 유사한 구성요소에 대해 사용하였다.
- [0025] 제1, 제2 등의 용어는 다양한 구성요소들을 설명하는데 사용될 수 있지만, 상기 구성요소들은 상기 용어들에 의해 한정되어서는 안 된다. 상기 용어들은 하나의 구성요소를 다른 구성요소로부터 구별하는 목적으로만 사용된다. 예를 들어, 본 발명의 권리 범위를 벗어나지 않으면서 제1 구성요소는 제2 구성요소로 명명될 수 있고, 유사하게 제2 구성요소도 제1 구성요소로 명명될 수 있다. 및/또는 이라는 용어는 복수의 관련된 기재된 항목들의 조합 또는 복수의 관련된 기재된 항목들 중의 어느 항목을 포함한다.
- [0026] 어떤 구성요소가 다른 구성요소에 "연결되어" 있다거나 "접속되어" 있다고 언급된 때에는, 그 다른 구성요소에 직접적으로 연결되어 있거나 또는 접속되어 있을 수도 있지만, 중간에 다른 구성요소가 존재할 수도 있다고 이해되어야 할 것이다. 반면에, 어떤 구성요소가 다른 구성요소에 "직접 연결되어" 있다거나 "직접 접속되어" 있

다고 언급된 때에는, 중간에 다른 구성요소가 존재하지 않는 것으로 이해되어야 할 것이다.

[0027] 본 출원에서 사용한 용어는 단지 특정한 실시예를 설명하기 위해 사용된 것으로, 본 발명을 한정하려는 의도가 아니다. 단수의 표현은 문맥상 명백하게 다르게 뜻하지 않는 한, 복수의 표현을 포함한다. 본 출원에서, "포함하다" 또는 "가지다" 등의 용어는 명세서상에 기재된 특징, 숫자, 단계, 동작, 구성요소, 부품 또는 이들을 조합한 것이 존재함을 지정하려는 것이지, 하나 또는 그 이상의 다른 특징들이나 숫자, 단계, 동작, 구성요소, 부품 또는 이들을 조합한 것들의 존재 또는 부가 가능성을 미리 배제하지 않는 것으로 이해되어야 한다.

[0028] 이하, 첨부한 도면들을 참조하여, 본 발명의 실시예들을 상세하게 설명한다. 이하, 도면상의 동일한 구성요소에 대해서는 동일한 참조부호를 사용하고 동일한 구성요소에 대해서 중복된 설명은 생략한다.

[0029] 새로운 TV 표준인 ATSC 3.0에서는 MPEG-DASH과 MPEG MMT 등 인터넷 기반 멀티미디어 서비스 간의 연동을 용이하게 한다. OTT 서비스를 구현하기 위해, ATSC 3.0 TV 채널을 통해 전송된 TV 프로그램이나, 카메라에서 실시간으로 입력되는 스트림을 변환하고, 변환된 스트림을 MPEG-DASH 및/또는 MPEG MMT 표준에 따르는 클라이언트에서 소비할 수 있게 하는 중계 시스템이 필요하다.

[0030] OTT 서비스를 이용한 프로그램은 기존 ATSC 3.0 채널을 직접 받는 프로그램에 비해, 3초가량의 지연이 발생할 수 있다. 이는 TV 프로그램 시청에 있어서 충분히 만족스러운 정도이다. 그러나, 아래의 표 1에서 보듯이 미디어 서비스에 따라, 허용되는 지연 한계가 매우 다를 수 있다.

표 1

서비스 카테고리	네트워크 게임		허용되는 지연 한계
주문형 서비스	VOD 서비스		3초
대화형 서비스	화상전화		130ms
	화상회의		150ms
인터랙티브 서비스	네트워크 게임	RTS	500ms
		racing	50ms
		FPS	50ms
		simulation	2초
	4차 산업혁명 관련 서비스	스마트 공장	10ms
		(의료) 수술	20ms
		자율 자동차	10ms

[0031]

[0032] 표 1에서 특히 주목할 것은 게임 서비스이다. 햅틱 감각은 시각에 비해 지연에 훨씬 민감하므로, 종래의 무선 네트워크를 통해서만 게임 서비스 등을 제공하기 위해 요구되는 전송 지연 조건을 충족하기 어렵다.

[0033] 5G 무선 네트워크 표준 이후부터는 무선 네트워크를 통한 데이터의 송수신으로 인한 전송 지연을 1ms 정도로 단축시킬 수 있다. 그러므로, 5G 네트워크와 그 이후의 네트워크에서 서비스를 제공할 수 있으며, 네트워크를 통한 게임 등의 서비스의 제공 방법이 발전할 것이다. ATSC 3.0을 이용한 OTT 서비스에서도 기존의 주문형 비디오 서비스에 국한하지 말고, 더 지연에 민감한 서비스를 요구하는 시장의 흐름을 고려해야 할 것이다.

[0034] 본 개시에서의 OTT 시스템의 OTT 서버는 ATSC 3.0 서버로부터 수신한 비트 스트림의 변환 과정을 최소화한 상태로 전송하고, OTT 클라이언트에서 비트 스트림을 변환하는 프로세스를 진행하게 하여 지연을 최소화한다.

[0036] 도 1은 본 개시에서 적용되는 ATSC 3.0에 따른 OTT(over-the-top) 중계 시스템의 예시를 도시한 도면이다.

[0037] 도 1을 참조하면, OTT 중계 시스템은, ATSC 3.0 서버(102), ATSC 3.0 채널(104), OTT 서버(106), 인터넷(108) 및 OTT 클라이언트(110)를 포함할 수 있다.

- [0038] ATSC 3.0 서버(102)는 ATSC 3.0 채널(103)을 통해 OTT 서비스를 위한 데이터 및 OTT 서비스를 통해 제공할 미디어 데이터들을 전송할 수 있다. 여기서 ATSC 3.0 서버(102)에 의해 전송되는 미디어 데이터는 MPEG-2 TS, ROUTE, MMT 형식에 따를 수 있다.
- [0039] OTT 서버(106)는 ATSC 3.0 서버(102)로부터 ATSC 3.0 채널(104)을 통해 OTT 서비스를 위한 데이터 및 OTT 서비스를 통해 제공할 미디어 데이터들을 수신할 수 있다. 그리고 OTT 서버(106)는 인터넷(108)의 HTTP 프로토콜 및 MMT/UDP 프로토콜을 통해 데이터들을 전송할 수 있다.
- [0040] OTT 클라이언트(110)는 인터넷(108)을 통해 OTT 서버(106)로부터 OTT 서비스를 위한 데이터 및 미디어 데이터들을 수신할 수 있다. 그리고 OTT 클라이언트(110)는 수신한 미디어 데이터들을 디코딩하고, 미디어를 재생할 수 있다. OTT 클라이언트(110)는 미디어 데이터를 MPU(media processing unit) 형태로 조합할 수 있으며, 조합한 MPU를 디코딩하여 미디어를 재생할 수 있다. 여기서, MPU는 CMAF(common media application format) 기반의 MPU일 수 있다. CMAF기반 MPU에 포함된 정보들은 아래에 설명하는 바와 같다.
- [0042] 도 2는 본 개시에서 적용되는 CMAF(common media application format) 기반의 MPU(media processing unit)의 구조를 도시한 도면이다.
- [0043] 도 2를 참조하면, CMAF 기반의 MPU는 CMAF 헤더(header), CMAF 세그먼트(segment), CMAF 청크(chunk) 및 CMAF 트랙 파일(track file)을 포함할 수 있다.
- [0044] CMAF 헤더는 특정한 트랙을 초기화하기 위한 정보들을 포함한다. CMAF 세그먼트는 동일한 트랙으로부터의 적어도 하나 이상의 연속적인 프래그먼트들의 시퀀스를 포함한다. CMAF 청크는 하나의 프래그먼트에 포함된 연속적인 샘플들의 서브셋을 포함한다. 그리고, CMAF 트랙 파일은 하나의 ISO_BMFF 파일에 포함된 완전한 트랙을 포함한다.
- [0045] OTT 시스템에서 CMAF 기반의 MPU를 송수신하는 동작은 아래에 설명하는 바와 같다.
- [0047] 도 3(a)는 본 개시에서 적용되는 OTT 시스템에서 CMAF 기반의 MPU를 생성하는 동작의 제1 실시예를 도시한 도면이다.
- [0048] 도 3(a)를 참조하면, OTT 시스템은 ATSC 3.0 서버, OTT 서버 및 OTT 수신기를 포함하고, 여기서, OTT 서버는 CMAF 핸들러 및 송신기를 포함할 수 있다.
- [0049] S301a 단계에서, ATSC 3.0 서버는 ATSC 3.0 채널을 통해 패킷들(예를 들어, MPEG-2 TS 패킷, ROUTE 패킷 및/또는 MMT 패킷)을 전송할 수 있다. S303a 단계에서, ATSC 3.0 서버로부터 패킷들을 수신한 OTT 서버의 CMAF 핸들러는 수신한 패킷들을 조합함으로써, CMAF 기반의 MPU를 생성할 수 있다. S305a 단계에서, OTT 서버의 CMAF 핸들러는 송신기에게 CMAF 기반의 MPU의 전송을 요청할 수 있다. S307a 단계에서, OTT 서버는 송신기를 이용하여, OTT 클라이언트의 수신기에 CMAF 기반의 MPU를 전송할 수 있다. OTT 클라이언트는 수신기를 통해 CMAF 기반의 MPU를 수신할 수 있다. S309a 단계에서, OTT 클라이언트는 수신한 CMAF 기반의 MPU를 디코딩하고, 디코딩 결과 획득한 미디어를 재생할 수 있다.
- [0051] 기존의 방법에 따르면, 하나의 MPU(MMT Processing Unit)를 생성하기 위해서, OTT 서버는 하나의 MPU에 해당하는 데이터를 전부 수신하여야 한다. 즉, OTT 서버는 MPU의 길이만큼 MPU의 전송을 대기하여야 한다. 따라서, 2 GOP(group of picture) 길이 가량의 지연이 발생할 수 있다.
- [0052] 반면, 본 개시에 따르면, OTT 서버는 MPU에 해당하는 데이터의 수신 완료 시까지 대기하지 아니하고, ATSC 3.0 서버로부터 수신한 데이터를 즉시 UDP(user datagram protocol) 패킷 형태의 MFU로 변환하고, UDP 패킷을 전송한다. OTT 클라이언트는 UDP 패킷 형태의 MFU를 수신할 수 있다. 그리고, OTT 클라이언트의 OTT 플레이어는 수신한 UDP 패킷 형태의 MFU를 미디어 데이터를 소비할 수 있는 형태인 MPU, CMAF 기반의 MPU 및/또는 ES(elementary stream) 형태로 변환할 수 있다. 이에 따른 OTT 서비스를 지원하는 OTT 서버 및 OTT 클라이언트의 동작은 아래에 설명한 바와 같다.

- [0054] 도 3(b)는 본 개시에서 적용되는 OTT 시스템에서 CMAF 기반의 MPU를 생성하는 동작의 제2 실시예를 도시한 도면이다.
- [0055] 도 3(b)를 참조하면, OTT 시스템은 ATSC 3.0 서버, OTT 서버 및 OTT 수신기를 포함하고, 여기서, OTT 서버는 CMAF 핸들러 및 송신기를 포함할 수 있다.
- [0056] S301b 단계에서, ATSC 3.0 서버는 ATSC 3.0 채널을 통해 패킷들(예를 들어, MPEG-2 TS 패킷, ROUTE 패킷 및/또는 MMT 패킷)을 전송할 수 있다. S303b 단계에서, ATSC 3.0 서버로부터 패킷들을 수신한 OTT 서버의 CMAF 핸들러는 수신한 패킷을 송신기에게 전달할 수 있다. S305b 단계에서, OTT 서버는 송신기를 이용하여, OTT 클라이언트의 수신기에 패킷들을 전송할 수 있다. OTT 클라이언트는 수신기를 통해 패킷들을 수신할 수 있으며, S307b 단계에서, 수신한 패킷들을 조합함으로써, CMAF 기반의 MPU를 생성할 수 있다. 그리고 S309b 단계에서, OTT 클라이언트는 생성한 CMAF 기반의 MPU를 디코딩하고, 디코딩 결과 획득한 미디어를 재생할 수 있다.
- [0057] 본 개시에 따르면, OTT 서버는 미디어 데이터를 MPU(예를 들어, CMAF 기반의 MPU) 형태로 조합하지 않고 ATSC3.0 채널에서 수신하자마자 바로 전송하며, OTT 클라이언트가 미디어 데이터를 MPU(예를 들어, CMAF 기반의 MPU) 형태로 조합하고, 조합된 미디어 데이터를 재생한다. 위와 같은 OTT 서버 및 OTT 클라이언트의 특징으로 인해, OTT 시스템은 전송 지연을 1 GOP 길이 가량으로 단축시킬 수 있다.
- [0058] OTT 서버는 RTSP를 이용하여 OTT 클라이언트에 스트리밍 서비스를 제공할 수 있다. RTSP를 이용하여 스트리밍 서비스를 제공하는 OTT 서버 및 OTT 서버로부터 수신한 데이터를 기초로 스트리밍을 수행하는 OTT 클라이언트의 동작은 아래에 설명한 바와 같을 수 있다.
- [0060] 도 4는 본 개시에서 적용되는 서버와 클라이언트 간의 RTSP(real time streaming protocol)를 이용한 스트리밍 서비스 동작을 도시한 도면이다.
- [0061] 도 4를 참조하면, OTT 클라이언트는 웹 브라우저 등의 클라이언트 및 미디어 플레이어를 포함할 수 있고, OTT 서버는 웹 서버 및 미디어 서버를 포함할 수 있다.
- [0062] S401 단계에서, 클라이언트는 미디어 데이터의 http 주소를 요청할 수 있다. 그리고 S403 단계에서, 웹 서버는 미디어 데이터의 http 주소 요청에 대한 응답을 전달할 수 있다.
- [0063] S405 단계에서, 미디어 플레이어는 획득한 http 주소를 이용하여, 미디어 서버에 RSTP 접속을 요청할 수 있다. 그리고 S407 단계에서, 미디어 플레이어는 미디어 서버로부터 콘텐츠 정보를 전달받을 수 있다.
- [0064] S409 단계에서, 미디어 서버는 미디어 플레이어와 RTSP 셋업 동작을 수행할 수 있다. S411 단계에서, 미디어 서버는 미디어 플레이어에게 RTSP 재생을 명령할 수 있다. S413 단계에서, 미디어 서버는 미디어 플레이어에게 RTP(real-time transport protocol) 오디오 데이터를 전송할 수 있으며, S415 단계에서, 미디어 서버는 미디어 플레이어에게 RTP 오디오 데이터를 전송할 수 있다. 미디어 서버로부터 RTSP 재생 명령을 수신한 미디어 플레이어는 수신한 RTP 오디오 및 RTP 비디오 데이터를 재생할 수 있다.
- [0065] S417 단계에서, 미디어 플레이어는 미디어 서버에게 RTC(RTP control) 정보를 전송할 수 있다. S419 단계에서, 미디어 서버는 미디어 플레이어에게 RTSP 일시 정지를 명령할 수 있다. 그리고 S421 단계에서 미디어 서버는 미디어 플레이어에게 미디어 데이터의 전송을 중단하고 세션을 해제하도록 RTSP 티어다운(teardown)을 명령할 수 있다. 미디어 서버로부터 RTSP 일시 정지 및 RTSP 티어다운 명령을 수신한 미디어 플레이어는 미디어 데이터의 수신을 중단하고 세션을 해제할 수 있다.
- [0066] OTT 서버는 RTSP를 이용하여 OTT 클라이언트에 스트리밍 서비스를 제공할 수 있다. 여기서, 본 개시에서의 OTT 서버 및 OTT 클라이언트에서 수행되는 RTSP를 이용한 OTT 서비스를 위한 미디어 데이터 송수신 동작은 아래에 설명한 바와 같을 수 있다.
- [0068] 도 5는 본 개시에서 적용되는 OTT 서버와 OTT 클라이언트 간의 미디어 데이터 송수신 동작의 제1 실시예를 도시한 도면이다.
- [0069] 도 5를 참조하면, OTT 서버(510)는 모바일 EPG/ESG 서비스부(511), OTT 서비스 제어부(512), 큐잉 메시지 파서(513), MMT 시그널링 패킷화/전송부(514), 패킷 병합/분할부(515), MMT 패킷화/전송부(516), 광고 콘텐츠

(AD(advertisement) contents) 저장부(517) 및 UI 서비스/RTSP부(518)를 포함한다. 그리고 OTT 클라이언트는 OTT M/W(middleware)(530), OTT 플레이어(540)를 포함한다. OTT M/W(530)는 HTTP 에이전트(531), UI 핸들러/RTSP부(532), MMT 시그널링 핸들러(533), UDP 소켓(534), 패킷 분해/ES 추출부(535), 및 미디어 디코더(536)를 포함한다.

[0070] OTT 서버(510)는 ATSC 3.0 채널을 통해 수신한 정보 및/또는 신호를 UDP 패킷으로 변환할 수 있다. OTT 서버(510)는 MMT/UDP 프로토콜을 이용하여 실시간 미디어 데이터를 전송할 수 있다. 그리고 OTT 서버(510)는 대체로 HTTP/TCP 프로토콜을 이용하여 EPG와 MMT 신호를 전송한다. OTT 서버(510)는 MMT/UDP 프로토콜을 이용하여 실시간 미디어 데이터와 같이 실시간으로 전송되어야 하는 시그널링을 전송한다.

[0071] 구체적으로, EPG/ESG 서비스부(511)는 프로그램 가이드 발생기로부터 획득한 정보를 HTTP/TCP 프로토콜을 이용하여 CDN 인프라(520)를 통해 전송할 수 있고, SCTE-35 신호를 MMT 형식에 따라 패킷화함으로써, 패킷화된 SCTE-35 신호를 HTTP/TCP 프로토콜을 이용하여 CDN 인프라(520)를 통해 전송한다.

[0072] 그리고 OTT 서버(510)는 미디어 데이터를 MMT/UDP 프로토콜을 통해 전송할 수 있다. OTT 서버(510)는 ATSC 3.0 채널을 통해 MPEG-2 TS 패킷, ROUTE 패킷, 및/또는 MMT 패킷을 수신할 수 있다. OTT 서버(510)의 패킷 병합/분할부(515)는 수신한 패킷들을 병합하거나 분할할 수 있다. 그리고 OTT 서버(510)의 MMT 패킷화/전송부(516)는 수신한 패킷의 유형을 구분하지 아니하고 MMT 패킷 형태로 패킷화할 수 있으며, MMT 패킷을 MMT/UDP 프로토콜을 통해 전송할 수 있다. 광고가 삽입되는 시간 동안, OTT 서버는 ATSC 3.0 채널을 통해 수신한 데이터 대신에 광고 콘텐츠 저장부(517)에 미리 저장된 광고 미디어 데이터를 전송한다. 여기서, 광고가 삽입되는 시간은 SCTE-35 시그널에 의해 지시될 수 있다.

[0073] OTT 서버(510)는 ATSC 3.0 채널을 통해 수신한 데이터를 데이터의 프레임이나 GOP(group of picture)의 구분과 상관없이 정해진 시간 간격마다 MMT 패킷의 형태로 전송한다. 같은 프로그램(예를 들어, 같은 드라마 또는 중계 방송)은 하나의 ATSC 3.0 채널에 포함된 여러 개의 서브 채널을 통해 8k, 4k, HD, SD 등의 서로 다른 품질로 시멀캐스트(simulcast)될 수 있다. 여기서, OTT 서버는 QoS 측정 시스템을 통해서 적절하다고 판단된 서브 채널을 통해 미디어 데이터를 전송한다. 또는 OTT 서버는 트랜스 코딩을 통해서 미디어 데이터를 다른 품질로 변환하여 보낼 수 있다.

[0074] OTT M/W(530)의 HTTP 에이전트(531)는 HTTP 프로토콜을 통해 프로그램 가이드 정보 및 SCTE-35 신호를 획득할 수 있다. OTT M/W(530)의 MMT 시그널링 핸들러(533)는 HTTP 에이전트(531)로부터 획득한 프로그램 가이드 정보 및 SCTE-35 신호를 기초로 MMT 신호를 생성할 수 있으며, 생성한 MMT 신호를 미디어 디코더(536)에게 전달한다.

[0075] 그리고 OTT M/W(530)의 UDP 소켓(534)는 MMT/UDP 프로토콜을 통해 미디어 패킷(예를 들어, MPEG-2 TS 패킷, ROUTE 패킷, 및/또는 MMT 패킷)을 수신할 수 있다. OTT M/W의 구조는 이용할 수 있는 API의 ES 디코딩 특성에 따라서, 다르게 구현될 수 있다. API가 ES를 바로 디코딩하는 경우, OTT M/W는 패킷분해/ES 추출부(535)를 이용하여, 미디어를 디코딩할 수 있도록 ES를 추출하고, 추출한 ES를 미디어 디코더(536)에게 전달한다. 미디어가 비디오인 경우, ES의 기본 단위는 하나의 프레임이다. MMT에서는 MFU(media fragment unit)이며, 역시 하나의 프레임이다. 미디어를 디코딩하기 위해 사용되는 코덱이나, 미디어가 재생되는 화면 크기 정보 등을 포함하는 디코더 정보는 패킷분해/ES 추출부(535)에 의해 추출되어 전달되거나, MMT 시그널링 핸들러(533)에 의해 MMT 신호를 통해 전달될 수 있다. 만일 네트워크에서 지연과 지터가 거의 없는 경우라면(즉, 수십 ms 이내) ATSC 3.0 서버에서부터의 시간 지연은 수십 ms 이내일 수 있다. 여기서, OTT M/W는 UDP 소켓에 지터의 크기보다 큰 리시버 버퍼를 포함할 수 있다.

[0077] 도 6은 본 개시에서 적용되는 OTT 서버와 OTT 클라이언트 간의 미디어 데이터 송수신 동작의 제2 실시예를 도시한 도면이다.

[0078] 도 6을 참조하면, OTT 서버(610)는 모바일 EPG/ESG 서비스부(611), OTT 서비스 제어부(612), 큐잉 메시지 파서(613), MMT 시그널링 패킷화/전송부(614), 패킷 병합/분할부(615), MMT 패킷화/전송부(616), 광고 콘텐츠(AD(advertisement) contents) 저장부(617) 및 UI 서비스/RTSP부(618)를 포함한다. 그리고 OTT 클라이언트는 OTT M/W(middleware)(630), OTT 플레이어(640)를 포함한다. OTT M/W(630)는 HTTP 에이전트(631), UI 핸들러/RTSP부(632), MMT 시그널링 핸들러(633), UDP 소켓(634), 패킷분해/ES 추출부(635), CMAF 세그먼트/MPU 생성부(636) 및 미디어 핸들러(637)를 포함한다.

[0079] 앞서 도 5에 관한 설명과 동일하게, OTT 서버(610)는 ATSC 3.0 채널을 통해 수신한 정보 및/또는 신호를 UDP 패

킷으로 변환할 수 있다. OTT 서버(610)는 MMT/UDP 프로토콜을 이용하여 CDN 인프라(620)을 통해 실시간 미디어 데이터를 전송한다. 그리고 OTT 서버(610)는 HTTP/TCP 프로토콜을 이용하여 CDN 인프라(620)을 통해 EPG와 MMT 신호를 전송한다.

- [0080] OTT M/W(630)는 HTTP 프로토콜을 이용하여, 프로그램 가이드 정보 및 SCTE-35 신호를 수신할 수 있다. 그리고 OTT M/W(630)는 MMT/UDP 프로토콜을 이용하여, 미디어 패킷(예를 들어, MPEG-2 TS 패킷, ROUTE 패킷, 및/또는 MMT 패킷)을 수신할 수 있다. OTT M/W(630)는 수신한 미디어 패킷을 분해하고, ES를 추출할 수 있다.
- [0081] OTT M/W(630)의 구조는 이용할 수 있는 API의 ES 디코딩 특성에 따라서, 다르게 구현될 수 있다. API가 CMAF나 MPU 단위로 ES를 디코딩하는 경우, OTT M/W(630)는 ES들을 CMAF 및/또는 MPU 포맷에 따라 조합해야 하는 CMAF 세그먼트/MPU 생성부(636)를 더 포함할 수 있으며, ES를 CMAF 및/또는 MPU 포맷에 따라 조립할 수 있다.
- [0082] 예를 들어, 스마트폰의 경우, CMAF를 재생하는 API 만을 포함하고 있을 수 있어, OTT M/W(630)는 ES들을 CMAF 및/또는 MPU 포맷에 따라 조합해야 한다. 그리고, OTT 시스템이 DASH(Dynamic Adaptive Streaming over HTTP)를 지원하는 경우, OTT M/W(630)는 미디어 핸들러에 미디어 데이터를 GOP 단위로 입력할 수 있다.
- [0083] 따라서, 하나의 CMAF 세그먼트 및/또는 하나의 MPU에 해당하는 시간만큼의 지연이 발생한다. 지연 시간은 대개 0.5초 정도 가량이나, 2초 가량까지 증가할 수 있다.
- [0085] 도 7은 본 개시에서 적용되는 OTT 서버의 MFU 전송 방식의 제1 실시예를 도시한 도면이다.
- [0086] 도 7을 참조하면, OTT 서버는 ATSC 3.0 서버로부터 미디어 데이터를 수신할 수 있으며, 미디어 데이터는 GOP 형식의 데이터일 수 있다. GOP는 GOP 헤더 및 복수의 프레임들을 포함할 수 있고, 복수의 프레임들 각각은 비디오 데이터 및 오디오 데이터를 포함할 수 있다.
- [0087] 여기서, 각각의 비디오 프레임 데이터들과 비디오 프레임 데이터들 각각에 동기화된 오디오 데이터들은 인터리빙 방식으로 전송될 수 있다. 즉, 복수의 프레임들 각각의 비디오 데이터들 및 오디오 데이터들은 번갈아 가며 전송될 수 있다.
- [0089] 즉, OTT 서버는 복수의 오디오 데이터 패킷들 및 복수의 비디오 데이터 패킷들을 교대로 수신할 수 있다. 그리고 OTT 서버는 하나의 프레임의 오디오 데이터 및 비디오 데이터를 전부 수신한 이후에, 프레임의 오디오 데이터 및 비디오 데이터를 조합하여 MFU를 생성하고 전송할 수 있다. 따라서, OTT 서버는 MFU들을 주기적으로 인터리빙 방식으로 전송할 수 있다.
- [0091] 도 8은 본 개시에서 적용되는 OTT 서버의 MFU 전송 방식의 제2 실시예를 도시한 도면이다.
- [0092] 도 8을 참조하면, OTT 서버는 ATSC 3.0 서버로부터 미디어 데이터를 수신할 수 있으며, 미디어 데이터는 GOP 형식의 데이터일 수 있다. GOP는 GOP 헤더 및 복수의 프레임들을 포함할 수 있고, 복수의 프레임들 각각은 비디오 데이터 및 오디오 데이터를 포함할 수 있다.
- [0093] 여기서, 복수의 프레임들 각각의 비디오 데이터들과 오디오 데이터들은 비인터리빙 방식으로 전송될 수 있다. 즉, 복수의 프레임들 각각의 오디오 데이터들은 연속적으로 전송될 수 있으며, 오디오 데이터들이 먼저 연속적으로 전송된 이후에, 비디오 데이터들이 연속적으로 전송될 수 있다. 오디오 데이터의 크기가 비디오 데이터의 크기에 비해서 매우 작으므로, 비인터리빙 방식으로 전송하는 경우에 GOP 전체의 오디오 데이터를 먼저 전송하고, 다음에 비디오 프레임을 전송하면, 비디오 프레임을 수신하자마자 디코딩할 수 있으므로 전체적인 지연을 줄일 수 있다.
- [0095] 본 개시의 OTT 서비스 방식에 따르면, OTT 서버는 입력된 데이터를 MFU 단위로 OTT 클라이언트에게 전송할 수 있으며, OTT 클라이언트가 썬클라이언트인 경우를 가정해서, 많은 연산량을 요구하는 동작은 OTT 서버에 의해 수행된다. 본 개시의 OTT 서비스 방식에 따르면, 도 7에 도시된 바와 같이 인터리빙 방식으로 전송될 때 유용한 방식일 수 있다.
- [0096] 즉, OTT 서버는 MPEG-2 TS나 ROUTE/DASH, 또는 MMTP MFU를 MPU로 변환하는 동작 및 필요시 CMAF 기반의 MPU로

변환하는 동작을 수행한다. 단, OTT 서버는 하나의 MPU에 속한 데이터의 전부 입력되는 것을 기다리지 않을 수 있으며, OTT 서버는 MPU 헤더를 먼저 전송한 후, MPU에 속한 프레임 데이터를 획득하는 즉시 MFU로 변환하고, MFU를 OTT 클라이언트에게 전송한다. 만일 OTT 클라이언트가 CMAF 기반의 MPU 만을 재생하는 API를 포함하는 경우, OTT 클라이언트는 MFU를 CMAF 파일로 변환한다.

- [0097] ATSC 3.0 채널을 통한 입력 데이터가 MPEG-2 TS인 경우, OTT 서버는 이미 수신한 PMT, PAT 정보 및 GOP의 개시 시점 이후 처음으로 수신한 정보들을 이용하여 MPU 헤더를 생성한다. 또는 OTT 서버는 필요한 경우, CMAF 기반의 MPU의 헤더를 생성할 수 있다.
- [0098] 하나의 GOP의 처음 프레임에 해당하는 비디오와 오디오 데이터를 수신한 OTT 서버는 MPU 헤더 및/또는 CMAF 기반의 MPU 헤더를 생성하고, MPU 헤더 및/또는 CMAF 기반의 MPU 헤더와 첫 프레임에 해당하는 데이터를 같이 전송한다. 이후, OTT 서버는 다음 프레임에 해당하는 데이터를 수신하는 즉시, MFU로 변환한 후, MMTP 표준에 따라서 전송할 수 있다. 여기서, MMTP 표준은 RTP(Realtime Transport Protocol, RFC 3550) 표준으로 대체할 수 있다. 이때, OTT 서버는 각각 MMT QoS 제어 알고리즘 또는 RTCP(Realtime Control Protocol, RFC 3605 또는 RFC5760)으로 네트워크의 상태(패킷 손실율, 왕복 지연 등)를 측정하여 적응적으로 전송한다.
- [0099] 그리고 OTT 서버는 필요한 경우, 하나의 GOP의 마지막 프레임 데이터에 MPU 테일 또는 CMAF 기반의 MPU 테일을 붙여서 전송할 수 있다.
- [0100] OTT 클라이언트는 디지터링 버퍼(dejittering buffer)를 두어, 정해진 시간 간격에 맞추어서 프레임마다 디코딩하고 디코딩한 프레임을 재생할 수 있다. 만일 OTT 클라이언트가 하나의 GOP에 해당하는 CMAF 기반의 MPU를 입력으로 하는 플레이어 API를 포함하는 경우, OTT 클라이언트는 하나의 GOP에 해당하는 데이터를 전부 수신한 후, 미디어를 재생할 수 있다.
- [0101] ATSC 3.0 채널을 통한 입력이 ROUTE/DASH인 경우, 미디어 데이터 각각의 세그먼트(예를 들어, 하나의 GOP)는 순차적인 비트스트림 형태로 전송될 수 있다. 즉, OTT 서버는 하나의 세그먼트를 전부 수신하기 전에도 수신한 부분의 데이터를 전송할 수 있다. OTT 서버는 각각의 세그먼트의 헤더 정보를 이용해서 MPU 헤더를 생성할 수 있으며, 필요한 경우에 CMAF 헤더를 생성할 수 있다.
- [0102] 도 7과 같이 미디어 데이터가 인터리빙 방식으로 전송되는 경우, OTT 서버는 하나의 GOP의 처음 프레임에 해당하는 비디오 데이터와 오디오 데이터를 분리할 수 있다. 첫 번째 프레임에 해당하는 비디오 데이터와 오디오 데이터를 수신한 OTT 서버는 MPU 헤더와 CMAF 헤더를 생성하고, 생성한 MPU 헤더 및/또는 CMAF 헤더를 첫 프레임에 해당하는 데이터와 같이 전송할 수 있다.
- [0103] 이후, OTT 서버는 다음 하나의 프레임에 해당하는 데이터를 수신하는 즉시 MFU를 생성하고, 생성한 MFU를 MMTP 표준에 맞추어 전송한다. MMTP 표준은 RTP(Realtime Transport Protocol, RFC 3550) 표준으로 대체할 수 있다. 이때, OTT 서버는 MMT QoS 제어 알고리즘 또는 RTCP(Realtime Control Protocol, RFC 3605 또는 RFC5760)을 이용하여, 네트워크의 상태(패킷 손실율, 왕복지연 등)에 따라서 데이터를 적응적으로 전송한다. OTT 서버는 필요한 경우, 하나의 GOP의 마지막 프레임 데이터에 MPU 테일 또는 CMAF 테일을 붙여서 전송한다.
- [0104] 그러나, 도 8에 도시된 바와 같이 미디어 데이터가 비인터리빙 방식으로 전송되는 경우, OTT 서버는 하나의 세그먼트의 데이터를 전부 수신한 이후에 첫 프레임에 해당하는 데이터를 전송할 수 있다. 하나의 GOP의 오디오 데이터가 먼저 전송되고, 비디오 데이터가 전송되는 구조인 경우, OTT 서버는 비디오 프레임 데이터들 각각을 수신하는 즉시 이전에 수신한 오디오 데이터와 비디오 데이터를 같이 전송함으로써, 파이프라인 구조에 의한 지연 감소 효과를 거둘 수 있다.
- [0105] OTT 클라이언트는 디지터링 버퍼(dejittering buffer)를 두어, 정해진 시간 간격에 맞추어서 프레임별로 디코딩하고 리플레이 한다. 만일 한 GOP에 해당하는 CMAF 파일을 입력으로 하는 플레이어 API를 사용해야 하는 경우에는 한 GOP에 해당하는 데이터가 다 도착한 후 플레이한다.
- [0106] ATSC 3.0 채널을 통한 입력이 MMTP의 경우에는 전송되는 패킷을 그대로 전송한다. 다만, OTT 서버와 OTT 클라이언트간 네트워크의 특성에 따라, 패킷 크기가 달라지는 경우, OTT 서버는 네트워크 특성을 반영함으로써, MFU와 MMTP 패킷을 수정한다. 또한, OTT 서버는 필요시 MPU 헤더 정보를 이용하여 CMAF 헤더를 만들어 MPU 헤더와 같이 전송한다.
- [0107] 도 7과 같이 미디어 데이터가 인터리빙 방식으로 전송된다면, 한 GOP의 처음 프레임에 해당하는 비디오와 오디오 데이터가 분리가능하다. 그 데이터가 수신되면, MPU 헤더와 CMAF 헤더를 만들어서 첫 프레임에 해당하는 데

이터와 같이 전송한다. 이후 OTT 서버는 다음 하나의 프레임에 해당하는 데이터를 수신하는 즉시 MFU를 생성할 수 있으며, 생성한 MFU를 MMTP 표준에 맞추어 전송할 수 있다. MMTP 표준은 RTP(Realtime Transport Protocol, RFC 3550) 표준으로 대체할 수 있다. 이때, OTT 서버는 MMT QoS 제어 알고리즘 또는 RTCP(Realtime Control Protocol, RFC 3605 또는 RFC5760)으로 네트워크의 상태(패킷 손실율, 왕복지연 등)를 측정하여 적응적으로 전송한다. OTT 서버는 필요한 경우, 하나의 GOP의 마지막 프레임 데이터에 MPU 테일 또는 CMAF 테일을 붙여서 전송한다.

[0108] 그러나, 도 8에 도시된 바와 같이 미디어 데이터가 비인터리빙 방식으로 전송되는 경우, OTT 서버는 하나의 GOP에 해당하는 데이터를 전부 수신한 이후에 첫 번째 프레임에 해당하는 데이터를 전송할 수 있다. 이때에, 하나의 GOP의 오디오 데이터를 먼저 전송하고, 비디오 데이터를 전송한다. OTT 클라이언트 비디오 프레임을 수신하는 즉시, 이전에 수신한 오디오 데이터와 같이 재생할 수 있어서, 파이프라인 구조에 의한 지연 감소 효과를 거둘 수 있다.

[0109] OTT 클라이언트는 디지터링 버퍼(dejittering buffer)를 두어, 정해진 시간 간격에 맞추어서 프레임별로 디코딩하고 리플레이 한다. 만일 OTT 클라이언트가 하나의 GOP에 해당하는 CMAF 파일을 입력으로 하는 플레이어 API를 사용하는 경우, OTT 클라이언트의 플레이어 API는 하나의 GOP에 해당하는 데이터를 전부 수신한 후에, 미디어를 재생할 수 있다.

[0111] OTT 클라이언트는 UI(User Interface)를 통해 채널 변경 여부를 결정하고, 결정한 채널 변경 여부에 관한 정보를 OTT 서버에게 전송한다. 예를 들어, 클라이언트는 OTT 서버에게 프로그램 가이드 및/또는 SCTE-35 신호가 전송되는 채널의 변경을 요청할 수 있다. 도 5 및 도 6을 참조하면, 프로그램 가이드 및/또는 SCTE-35 신호는 HTTP/TCP 채널로 전송되었으나, 지연에 민감한 서비스라면, MMT/UDP 채널을 통해 전송될 수 있다. 도 5 및 도 6을 참조하면, 채널 변경이 요청된 경우, OTT 서버는 ATSC 3.0 신호를 받는 부분에서 해당하는 채널 신호가 들어오도록 제어할 수 있다. 그리고 OTT 서버는 변경된 채널(예를 들어, MMT/UDP 채널)을 통해 프로그램 가이드 및/또는 SCTE-35를 전송할 수 있다.

[0112] OTT 클라이언트는 UDP 소켓 모듈을 이용하여, QoS를 측정할 수 있다. 그리고 OTT 클라이언트는 측정한 QoS를 기초로, 지터와 가용한 비트율 및 패킷 손실율을 측정한다. MMT/UDP를 통한 데이터 전송 도중에 데이터의 손실율이 미리 설정된 값을 초과하는 경우, OTT 클라이언트는 피드백 채널을 통해 손실율을 OTT 서버로 전송할 수 있다. 그리고, OTT 서버는 FEC(Forward Error Correction)의 레벨을 정하여, 잉여 패킷(redundant packet 또는 FEC packet)을 전송한다. 손실율이 약 10%인 경우, OTT 서버는 약 20% 정도의 잉여 패킷을 전송할 수 있다. 예를 들어, OTT 서버는 80개의 미디어 패킷을 전송할 때 20개의 잉여 패킷을 추가로 전송한다.

[0114] 본 개시의 예시적인 방법들은 설명의 명확성을 위해서 동작의 시리즈로 표현되어 있지만, 이는 단계가 수행되는 순서를 제한하기 위한 것은 아니며, 필요한 경우에는 각각의 단계가 동시에 또는 상이한 순서로 수행될 수도 있다. 본 개시에 따른 방법을 구현하기 위해서, 예시하는 단계에 추가적으로 다른 단계를 포함하거나, 일부의 단계를 제외하고 나머지 단계를 포함하거나, 또는 일부의 단계를 제외하고 추가적인 다른 단계를 포함할 수도 있다.

[0115] 본 개시의 다양한 실시 예는 모든 가능한 조합을 나열한 것이 아니고 본 개시의 대표적인 양상을 설명하기 위한 것이며, 다양한 실시 예에서 설명하는 사항들은 독립적으로 적용되거나 또는 둘 이상의 조합으로 적용될 수도 있다.

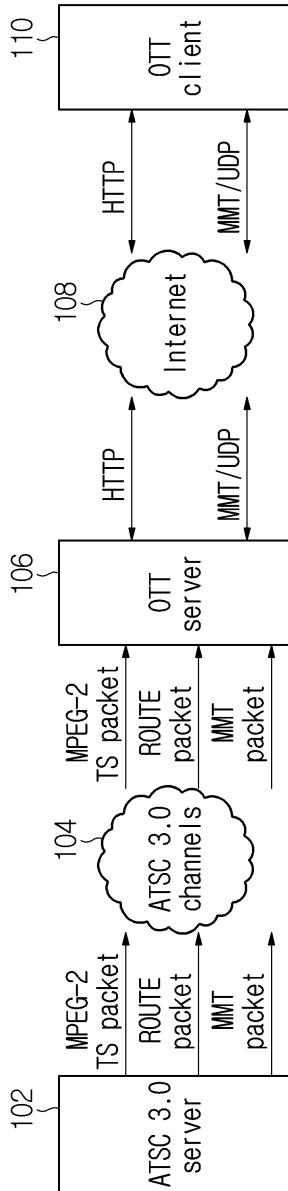
[0116] 또한, 본 개시의 다양한 실시 예는 하드웨어, 펌웨어(firmware), 소프트웨어, 또는 그들의 결합 등에 의해 구현될 수 있다. 하드웨어에 의한 구현의 경우, 하나 또는 그 이상의 ASICs(Application Specific Integrated Circuits), DSPs(Digital Signal Processors), DSPDs(Digital Signal Processing Devices), PLDs(Programmable Logic Devices), FPGAs(Field Programmable Gate Arrays), 범용 프로세서(general processor), 컨트롤러, 마이크로 컨트롤러, 마이크로 프로세서 등에 의해 구현될 수 있다.

[0117] 본 개시의 범위는 다양한 실시 예의 방법에 따른 동작이 장치 또는 컴퓨터 상에서 실행되도록 하는 소프트웨어 또는 머신-실행가능한 명령들(예를 들어, 운영체제, 애플리케이션, 펌웨어(firmware), 프로그램 등), 및 이러한 소프트웨어 또는 명령 등이 저장되어 장치 또는 컴퓨터 상에서 실행 가능한 비-일시적 컴퓨터-판독가능 매체

(non-transitory computer-readable medium)를 포함한다.

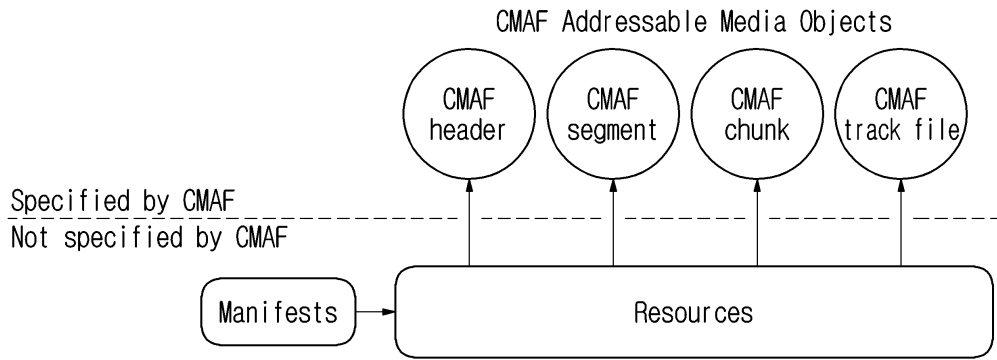
도면

도면1

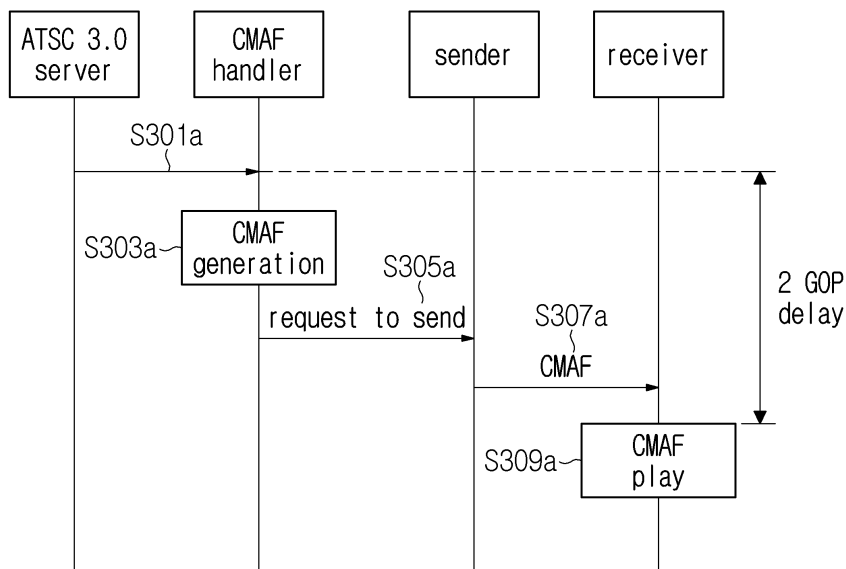


도면2

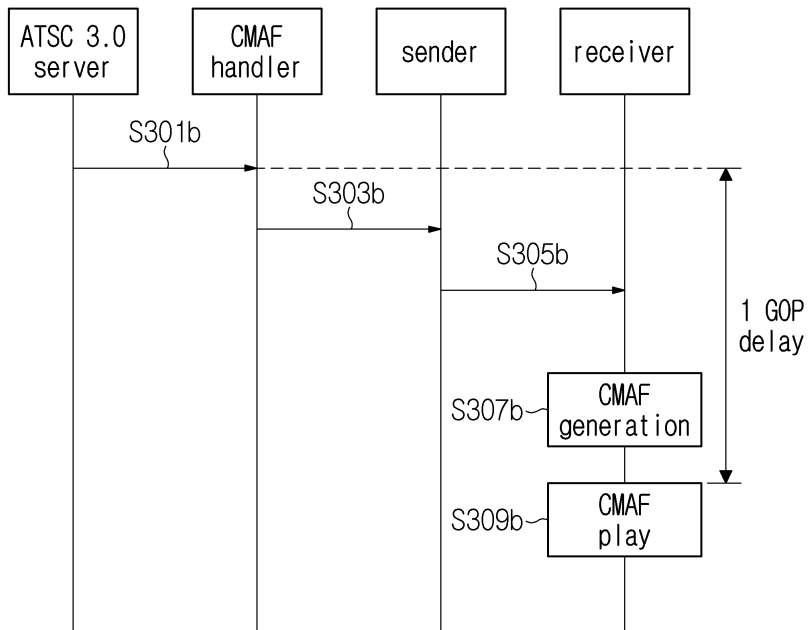
CMAF Header : Headers contain information that includes information for initializing a track.
 CMAF Segment : A sequence of one or more consecutive fragments from the same track.
 CMAF Chunk : A chunk contain a sequential subset of samples from a fragment.
 CMAF Track File : A complete track in one ISO_BMFF file.



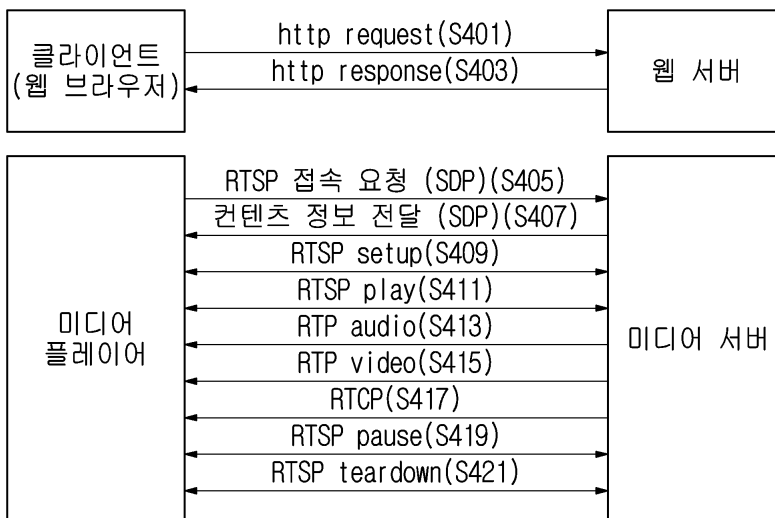
도면3a



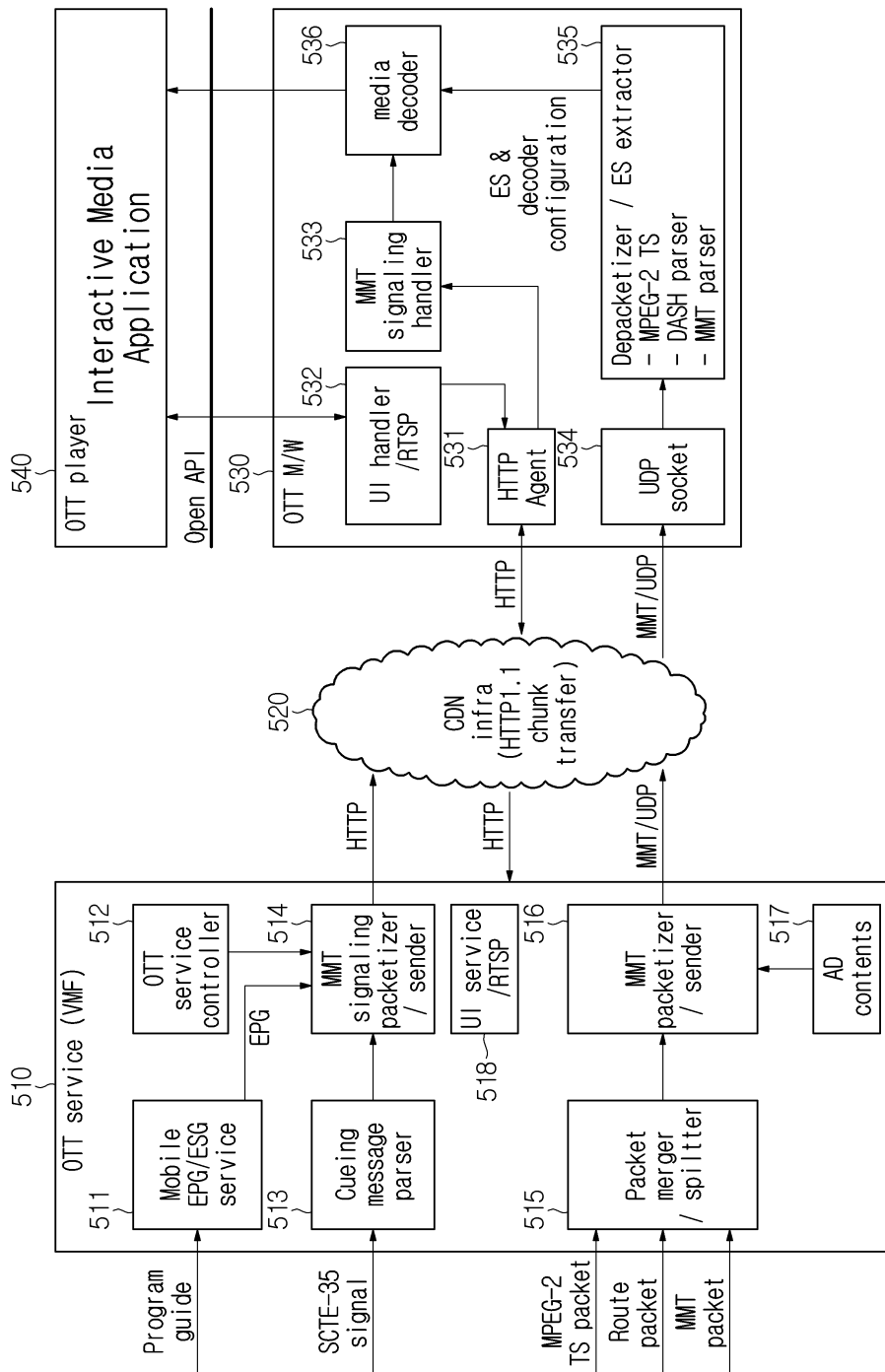
도면3b



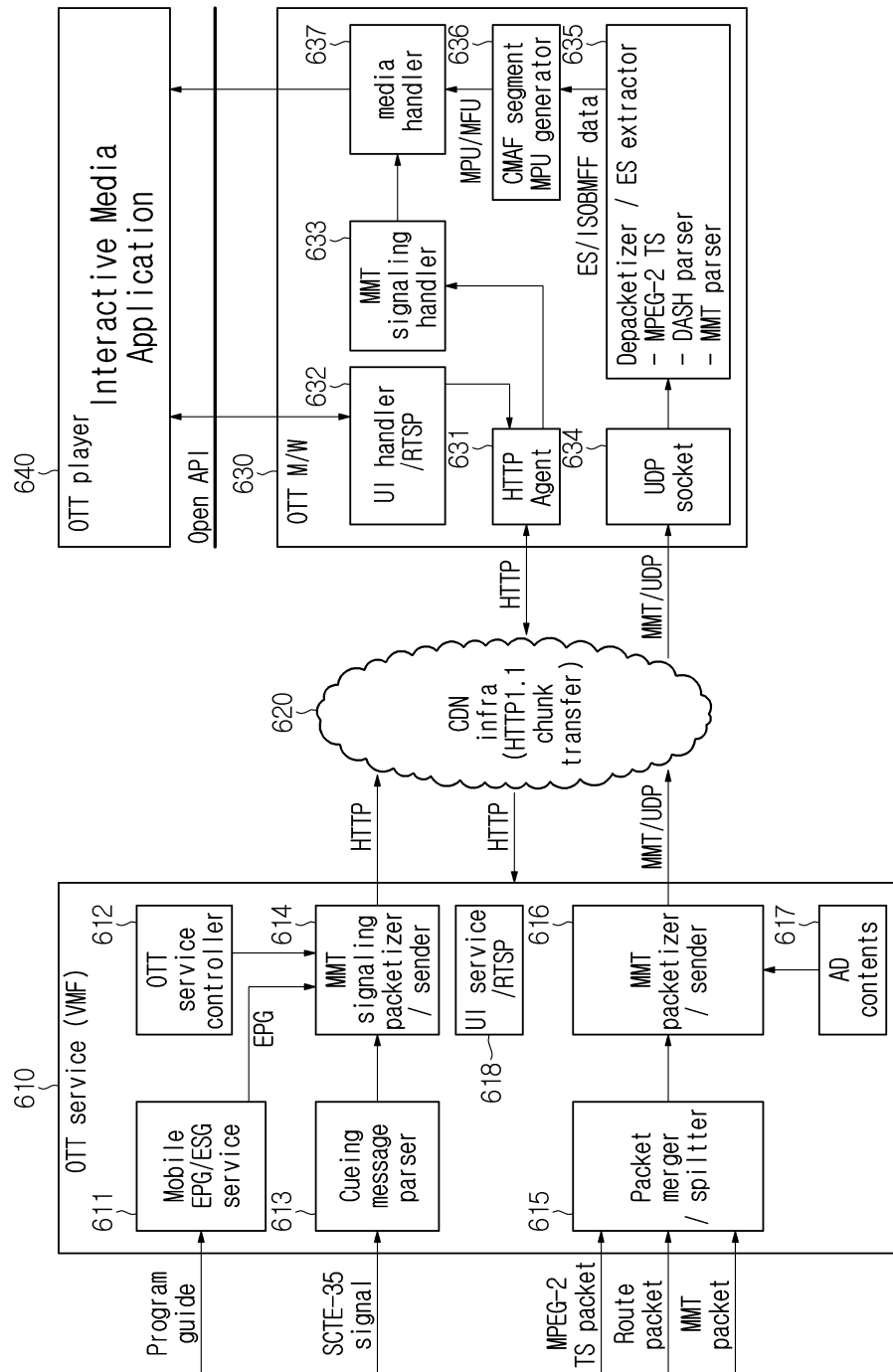
도면4



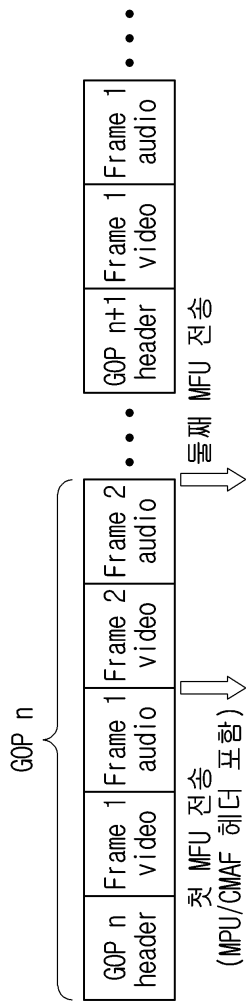
도면5



도면6



도면7



도면8

