(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2017-0077285
(43) 공개일자 2017년07월05일

(51) 국제특허분류(Int. Cl.)
H04N 21/2362 (2011.01) *H04H 60/40* (2008.01)
H04L 29/06 (2006.01) *H04L 5/00* (2006.01)
H04L 5/22 (2006.01) *H04N 21/2365* (2011.01)
H04N 21/242 (2011.01) *H04N 21/43* (2011.01)
H04N 21/845 (2011.01) *H04N 21/8547* (2011.01)
(52) CPC특허분류
H04N 21/2362 (2013.01)
H04H 60/40 (2013.01)
(21) 출원번호 10-2017-7017510(분할)
(22) 출원일자(국제) 2014년03월12일
심사청구일자 없음
(62) 원출원 특허 10-2015-7024271
원출원일자(국제) 2014년03월12일
심사청구일자 2016년08월10일
(85) 번역문제출일자 2017년06월26일
(86) 국제출원번호 PCT/JP2014/056574
(87) 국제공개번호 WO 2014/142203
국제공개일자 2014년09월18일
(30) 우선권주장
JP-P-2013-052559 2013년03월14일 일본(JP)
JP-P-2013-100774 2013년05월10일 일본(JP)

(71) 출원인
소니 주식회사
일본국 도쿄도 미나토쿠 코난 1-7-1
(72) 발명자
츠카고시 이쿠오
일본 1080075 도쿄도 미나토쿠 코난 1-7-1 소니 주식회사 내
(74) 대리인
장수길, 이중희

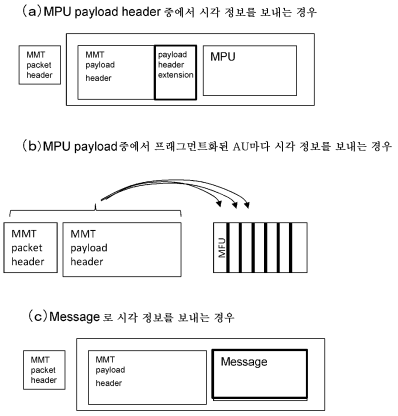
전체 청구항 수 : 총 13 항

(54) 발명의 명칭 **송신 장치, 송신 방법, 수신 장치 및 수신 방법**

(57) 요 약

차세대 방송용의 트랜스포트 구조에 있어서, 디코드 시각 및/또는 표시 시각을 취득하기 위한 시각 정보를 수신측으로 양호하게 전송 가능하게 한다. 페이로드에 전송 미디어를 포함하는 제1 전송 패킷과, 페이로드에 전송 미디어에 관한 정보를 포함하는 제2 전송 패킷이 시분할적으로 다중화된 전송 스트림을 생성한다. 이 전송 스트림을 RF 전송로, 통신 네트워크 전송로 등의 소정의 전송로를 통해서 수신측으로 송신한다. 제1 전송 패킷 또는 제2 전송 패킷에, 수신측에서 디코드 시각 및/또는 표시 시각을 취득하기 위한 시각 정보를 삽입한다.

대 표 도 - 도18



(52) CPC특허분류
H04L 5/0044 (2013.01)
H04L 5/22 (2013.01)
H04L 65/607 (2013.01)
H04N 21/2365 (2013.01)
H04N 21/242 (2013.01)
H04N 21/4307 (2013.01)
H04N 21/8456 (2013.01)
H04N 21/8547 (2013.01)

명 세 서

청구범위

청구항 1

페이로드에 전송 미디어를 포함하는 제1 전송 패킷과, 페이로드에 상기 전송 미디어에 관한 정보를 포함하는 제2 전송 패킷이 시분할적으로 다중화된 전송 스트림을 생성하는 전송 스트림 생성부와,

상기 전송 스트림을 소정의 전송로를 통해서 수신측으로 송신하는 전송 스트림 송신부와,

상기 제2 전송 패킷에, 상기 제1 전송 패킷에 포함되는 전송 미디어의 표시 시각의 값을 삽입하는 시각 정보 삽입부를 구비하는, 송신 장치.

청구항 2

제1항에 있어서,

상기 제1 전송 패킷에 포함되는 전송 미디어는 하나 이상의 액세스 유닛으로 구성되고,

상기 시각 정보 삽입부는, 상기 하나 이상의 액세스 유닛의 최초의 액세스 유닛에 대응한 표시 시각의 값을 삽입하는, 송신 장치.

청구항 3

제2항에 있어서,

상기 시각 정보 삽입부는, 각 액세스 유닛에 각각 대응한 디코드 시각으로부터 표시 시각까지의 오프셋값을 삽입하는, 송신 장치.

청구항 4

제3항에 있어서,

상기 오프셋값은, 절대 오프셋값에 대응한 상대 오프셋값이며,

상기 시각 정보 삽입부는, 상기 상대 오프셋값을 상기 절대 오프셋값으로 변환하기 위한 변환 정보를 삽입하는, 송신 장치.

청구항 5

제4항에 있어서,

상기 시각 정보 삽입부는, 상기 상대 오프셋값을 가변 길이 부호화한 후에 삽입하는, 송신 장치.

청구항 6

제1항에 있어서,

상기 제2 전송 패킷의 페이로드부는, 페이로드 헤더부와 페이로드 본체부를 포함하고,

상기 시각 정보 삽입부는, 상기 페이로드 본체부에, 상기 표시 시각의 값을 삽입하는, 송신 장치.

청구항 7

제1항에 있어서,

상기 소정의 전송로는, RF 전송로 또는 통신 네트워크 전송로인, 송신 장치.

청구항 8

제1항에 있어서,

상기 제1 전송 패킷 및 상기 제2 전송 패킷은, MMT 패킷인, 송신 장치.

청구항 9

페이로드에 전송 미디어를 포함하는 제1 전송 패킷과, 페이로드에 상기 전송 미디어에 관한 정보를 포함하는 제2 전송 패킷이 시분할적으로 다중화된 전송 스트림을 생성하는 전송 스트림 생성 스텝과,

상기 전송 스트림을 소정의 전송로를 통해서 수신측으로 송신하는 전송 스트림 송신 스텝과,

상기 제2 전송 패킷에, 상기 제1 전송 패킷에 포함되는 전송 미디어의 표시 시각의 값을 삽입하는 시각 정보 삽입 스텝

을 갖는, 송신 방법.

청구항 10

페이로드에 전송 미디어를 포함하는 제1 전송 패킷과, 페이로드에 상기 전송 미디어에 관한 정보를 포함하는 제2 전송 패킷이 시분할적으로 다중화된 전송 스트림을, 송신측으로부터 소정의 전송로를 통해서 수신하는 전송 스트림 수신부를 구비하고,

상기 제2 전송 패킷에는, 상기 제1 전송 패킷에 포함되는 전송 미디어의 표시 시각의 값이 삽입되어 있으며,

상기 전송 스트림으로부터 취출된 상기 전송 미디어를, 상기 표시 시각의 값에 기초하여 처리하는 전송 미디어 처리부를 더 구비하는, 수신 장치.

청구항 11

제10항에 있어서,

상기 제1 전송 패킷에 포함되는 전송 미디어는 하나 이상의 액세스 유닛으로 구성되고,

상기 표시 시각의 값은, 상기 하나 이상의 액세스 유닛의 최초의 액세스 유닛에 대응한 표시 시각의 값이며, 상기 제2 전송 패킷에는, 각 액세스 유닛에 각각 대응한 디코드 시각으로부터 표시 시각까지의 절대 오프셋값에 대응한 상대 오프셋값이 삽입되어 있으며,

상기 상대 오프셋값을 상기 절대 오프셋값으로 변환하는 오프셋값 변환부를 더 구비하는, 수신 장치.

청구항 12

페이로드에 전송 미디어를 포함하는 제1 전송 패킷과, 페이로드에 상기 전송 미디어에 관한 정보를 포함하는 제2 전송 패킷이 시분할적으로 다중화된 전송 스트림을, 송신측으로부터 소정의 전송로를 통해서 수신하는 전송 스트림 수신 스텝을 갖고,

상기 제2 전송 패킷에는, 상기 제1 전송 패킷에 포함되는 전송 미디어의 표시 시각의 값이 삽입되어 있으며,

상기 전송 스트림으로부터 취출된 상기 전송 미디어를, 상기 표시 시각의 값에 기초하여 처리하는 전송 미디어 처리 스텝을 더 갖는, 수신 방법.

청구항 13

제1항에 있어서,

상기 제2 전송 패킷에 포함되는 전송 미디어에 관한 정보는, 전송 미디어에 관한 제어 정보인, 송신 장치.

발명의 설명

기 술 분 야

[0001] 본 기술은, 송신 장치, 송신 방법, 수신 장치 및 수신 방법에 관한 것으로, 상세하게는, 비디오, 오디오 등의 전송 미디어를 RF 통신로나 통신 네트워크 통신로를 통해서 송신하는 송신 장치 등에 관한 것이다.

배 경 기 술

[0002] 종래, 예를 들어 특허문헌 1에 기재된 바와 같이, 방송용의 트랜스포트 구조로서 MPEG2-TS가 사용되고 있다.

선행기술문헌

특허문헌

[0003] (특허문헌 0001) 일본 특허공개 제2011-217161호 공보

발명의 내용

해결하려는 과제

[0004] 최근 들어, 차세대 방송용의 트랜스포트 구조로서 MMT(MPEG Media Transport) 구조가 점점 각광을 받고 있다. 이 MMT 구조는, IP 네트워크와의 공존이 주된 특징이다.

[0005] 본 기술의 목적은, 차세대 방송용의 트랜스포트 구조에 있어서, 디코드 시각 및/또는 표시 시각을 취득하기 위한 시각 정보를 수신측으로 양호하게 전송 가능하게 하는 데 있다.

과제의 해결 수단

[0006] 본 기술의 개념은,

[0007] 페이로드에 전송 미디어를 포함하는 제1 전송 패킷과, 페이로드에 상기 전송 미디어에 관한 정보를 포함하는 제2 전송 패킷이 시분할적으로 다중화된 전송 스트림을 생성하는 전송 스트림 생성부와,

[0008] 상기 전송 스트림을 소정의 전송로를 통해서 수신측으로 송신하는 전송 스트림 송신부와,

[0009] 상기 제1 전송 패킷 또는 상기 제2 전송 패킷에, 상기 수신측에서 디코드 시각 및/또는 표시 시각을 취득하기 위한 시각 정보를 삽입하는 시각 정보 삽입부를 구비하는 송신 장치에 있다.

[0010] 본 기술에 있어서, 전송 스트림 생성부에 의해, 전송 스트림이 생성된다. 이 전송 스트림은, 페이로드에 전송 미디어를 포함하는 제1 전송 패킷과, 페이로드에 전송 미디어에 관한 정보를 포함하는 제2 전송 패킷이 시분할적으로 다중화된 것이다. 예를 들어, 제1 전송 패킷 및 제2 전송 패킷은 MMT 패킷이도록 되어도 된다. 전송 스트림 송신부에 의해, 전송 스트림은, 소정의 전송로를 통하여, 수신측으로 송신된다. 예를 들어, 소정의 전송로는, RF 전송로 또는 통신 네트워크 전송로이도록 되어도 된다.

[0011] 시각 정보 삽입부에 의해, 제1 전송 패킷 또는 제2 전송 패킷에, 수신측에서 디코드 시각 및/또는 표시 시각을 취득하기 위한 시각 정보가 삽입된다. 예를 들어, 제1 전송 패킷에 포함되는 전송 미디어는 하나 이상의 액세스 유닛으로 구성되고, 시각 정보 삽입부에서 삽입되는 시각 정보는, 하나 이상의 액세스 유닛에 각각 대응한 디코드 시각 및/또는 표시 시각을 취득하기 위한 정보이도록 되어도 된다.

[0012] 이 경우, 시각 정보 삽입부에서 삽입되는 시각 정보는, 하나 이상의 액세스 유닛의 최초의 액세스 유닛에 대응한 디코드 시각 혹은 표시 시각의 값과, 각 액세스 유닛에 각각 대응한 디코드 시각으로부터 표시 시각까지의 오프셋값을 갖도록 되어도 된다. 이와 같이 오프셋값을 사용함으로써 시각 정보의 전송을 효율적으로 행하는 것이 가능하게 된다. 또한, 이 경우, 시각 정보 삽입부에서 삽입되는 시각 정보는, 하나 이상의 액세스 유닛의 각각에 대응한 표시 시각, 혹은 표시 시각 및 디코드 시각이도록 되어도 된다. 이와 같이 표시 시각, 혹은 표시 시각 및 디코드 시각 그 자체를 전송함으로써, 수신측에 있어서의 처리 부하의 경감이 가능하게 된다.

[0013] 그리고, 이 경우, 오프셋값은, 절대 오프셋값에 대응한 상대 오프셋값이며, 시각 정보 삽입부에서 삽입되는 시각 정보에는, 상대 오프셋값을 절대 오프셋값으로 변환하기 위한 변환 정보가 부가되도록 되어도 된다. 이와 같이 상대 오프셋값을 사용함으로써 오프셋값을 수신측에 효율적으로 전송하는 것이 가능하게 된다. 예를 들어, 시각 정보 삽입부는, 상대 오프셋값을 가변 길이 부호화한 후에 삽입하도록 되어도 된다. 이와 같이 가변 길이 부호화를 이용함으로써, 시각 정보의 전송 용량의 저감을 도모하는 것이 가능하게 된다.

[0014] 예를 들어, 제1 전송 패킷의 페이로드는, 페이로드 헤더부와 페이로드 본체부를 포함하며, 시각 정보 삽입부는, 페이로드 헤더부에, 시각 정보를 삽입하도록 되어도 된다. 또한, 예를 들어 제1 전송 패킷의 페이로드는, 페이로드 헤더부와 페이로드 본체부를 포함하고, 페이로드 본체부에는, 전송 미디어가 프래그먼트화된 하나 이상의

액세스 유닛을 각각 포함하는 프래그먼트 페이로드가 프래그먼트 헤더를 수반하여 배치되고, 시각 정보 삽입부는, 프래그먼트 헤더 혹은 프래그먼트 페이로드에, 대응하는 액세스 유닛의 시각 정보를 삽입하도록 되어도 된다. 또한, 예를 들어 제2 전송 패킷의 페이로드부는, 페이로드 헤더부와 페이로드 본체부를 포함하고, 시각 정보 삽입부는, 페이로드 본체부에, 시각 정보를 삽입하도록 되어도 된다.

[0015] 이와 같이 본 기술에 있어서는, 제1 전송 패킷 또는 제2 전송 패킷에, 수신측에서 디코드 시각 및 표시 시각을 취득하기 위한 시각 정보가 삽입되는 것이며, 차세대 방송용의 트랜스포트 구조에 있어서, 디코드 시각 및/또는 표시 시각을 취득하기 위한 시각 정보를 수신측으로 양호하게 전송 가능하게 된다.

[0016] 또한, 본 기술의 다른 개념은,

[0017] 페이로드에 전송 미디어를 포함하는 제1 전송 패킷과, 페이로드에 상기 전송 미디어에 관한 정보를 포함하는 제2 전송 패킷이 시분할적으로 다중화된 전송 스트림을, 송신측으로부터 소정의 전송로를 통해서 수신하는 전송 스트림 수신부를 구비하고,

[0018] 상기 제1 전송 패킷 또는 상기 제2 전송 패킷에는, 디코드 시각 및/또는 표시 시각을 취득하기 위한 시각 정보가 삽입되어 있으며,

[0019] 상기 전송 스트림으로부터 취출된 상기 전송 미디어를, 상기 시각 정보에 기초하여 취득된 상기 디코드 시각 및/또는 상기 표시 시각을 사용해서 처리하는 전송 미디어 처리부를 더 구비하는, 수신 장치에 있다.

[0020] 본 기술에 있어서, 전송 스트림 수신부에 의해, 소정의 전송로를 통하여, 송신측으로부터 전송 스트림이 수신된다. 이 전송 스트림은, 페이로드에 전송 미디어를 포함하는 제1 전송 패킷과, 페이로드에 전송 미디어에 관한 정보를 포함하는 제2 전송 패킷이 시분할적으로 다중화된 것이다. 제1 전송 패킷 또는 제2 전송 패킷에는, 디코드 시각 및/또는 표시 시각을 취득하기 위한 시각 정보가 삽입되어 있다. 전송 미디어 처리부에 의해, 전송 스트림으로부터 취출된 전송 미디어가, 시각 정보에 기초하여 취득된 디코드 시각 및/또는 표시 시각을 사용해서 처리된다.

[0021] 예를 들어, 제1 전송 패킷에 포함되는 전송 미디어는 하나 이상의 액세스 유닛으로 구성되고, 시각 정보는, 하나 이상의 액세스 유닛의 최초의 액세스 유닛에 대응한 디코드 시각 혹은 표시 시각의 값과, 각 액세스 유닛에 각각 대응한 디코드 시각으로부터 표시 시각까지의 절대 오프셋값에 대응한 상대 오프셋값이며, 이 상대 오프셋값을 절대 오프셋값으로 변환하는 오프셋값 변환부를 더 구비하도록 되어도 된다.

[0022] 이와 같이 본 기술에 있어서는, 제1 전송 패킷 또는 제2 전송 패킷에 디코드 시각 및/또는 표시 시각을 취득하기 위한 시각 정보가 삽입되어 있으며, 이 시각 정보에 기초하여 디코드 시각 및/또는 표시 시각을 취득하고, 전송 스트림으로부터 취출된 전송 미디어의 처리를 양호하게 행할 수 있다.

발명의 효과

[0023] 본 기술에 의하면, 차세대 방송용의 트랜스포트 구조에 있어서, 디코드 시각 및/또는 표시 시각을 취득하기 위한 시각 정보를 수신측으로 양호하게 전송할 수 있다. 또한, 본 명세서에 기재된 효과는 어디까지나 예시로서 한정되는 것이 아니라, 또한 부가적인 효과가 있어도 된다.

도면의 간단한 설명

[0024] 도 1은, 실시 형태로서의 송수신 시스템의 구성예를 나타내는 블록도이다.

도 2는, MMT 페이로드(MMT payload)의 구성을 개략적으로 나타내는 도면이다.

도 3은, MMT 파일과 실제로 전송되는 MMT 패킷의 대응 관계의 일례를 나타내는 도면이다.

도 4는, MMT 파일과 실제로 전송되는 MMT 패킷의 대응 관계의 다른 일례를 나타내는 도면이다.

도 5는, MMT 파일과 실제로 전송되는 MMT 패킷의 대응 관계의 또 다른 일례를 나타내는 도면이다.

도 6은, 프래그먼트화된 MFU를 포함하는 MMT 패킷을 전송 패킷으로 하는 경우로서, 그 MMT 패킷이 랜덤 액세스의 선두(GOP의 선두)의 패킷인 경우를 설명하기 위한 도면이다.

도 7은, 프래그먼트화된 MFU를 포함하는 MMT 패킷을 전송 패킷으로 하는 경우로서, 그 MMT 패킷이 랜덤 액세스의 비(非) 선두의 패킷인 경우를 설명하기 위한 도면이다.

도 8은, 프래그먼트화된 MFU를 포함하는 MMT 패킷을 전송 패킷으로 하는 경우로서, 그 MMT 패킷이 랜덤 액세스의 선두(GOP의 선두)의 패킷인 경우를 설명하기 위한 도면이다.

도 9는, MMT 패킷의 구성을 트리 형식으로 나타내는 도면이다.

도 10은, 트랜스포트 패킷 송신 장치 및 트랜스포트 패킷 수신 장치의 개념도이다.

도 11은, 「NTP short 포맷」을 나타내는 도면이다.

도 12는, 미디어 데이터가 비디오이며, 초기값이 디코드 시각인 경우에 있어서의 시각 정보를 설명하기 위한 도면이다.

도 13은, 미디어 데이터가 비디오이며, 초기값이 표시 시각인 경우에 있어서의 시각 정보를 설명하기 위한 도면이다.

도 14는, 미디어 데이터가 오디오이며, 초기값이 디코드 시각인 경우에 있어서의 시각 정보를 설명하기 위한 도면이다.

도 15는, 디코드 출력 처리부의 구성예를 나타내는 블록도이다.

도 16은, 제어부에 있어서의 AV 동기 재생 제어의 일례를 나타내는 타이밍 차트이다.

도 17은, MFU에 프래그먼트되어 있는 패킷이 전송 상에서 드롭된 경우에 대하여 설명하기 위한 도면이다.

도 18은, MMT 패킷에 시각 정보를 삽입하는 방법의 일례를 개략적으로 나타내는 도면이다.

도 19는, 송수신 지연을 낮게 하기 위해서, 각 액세스 유닛에 대응하여 시각 정보(타임 스탬프)를 삽입하는 경우의 패킷 구성의 일례를 나타내는 도면이다.

도 20은, MMT 패킷 전체의 구조예를 나타내는 도면이다.

도 21은, MMT 패킷 헤더(mmtp_header())의 구조예를 나타내는 도면이다.

도 22는, MMT 페이로드 헤더(mmtp_payload_header())의 구조예를 나타내는 도면이다.

도 23은, MMT 페이로드 헤더 익스텐션(mmtp_payload_header_extension())의 구조예를 나타내는 도면이다.

도 24는, MMT 페이로드 헤더 익스텐션의 구조예의 주요한 정보의 내용을 나타내는 도면이다.

도 25는, MMT 페이로드 헤더 익스텐션(mmtp_payload_header_extension())의 다른 구조예를 나타내는 도면이다.

도 26은, MMT 페이로드 헤더 익스텐션의 다른 구조예의 주요한 정보의 내용을 나타내는 도면이다.

도 27은, MPU 페이로드 중에서 프래그먼트화된 액세스 유닛마다 시각 정보를 보내는 경우에 대하여 설명하기 위한 도면이다.

도 28은, MFU의 구조예(Syntax)를 나타내는 도면이다.

도 29는, MFU 헤더(mfu_header())의 구조예를 나타내는 도면이다.

도 30은, MFU 헤더(mfu_header())의 구조예를 나타내는 도면이다.

도 31은, MFU 헤더의 구조예의 주요한 정보의 내용을 나타내는 도면이다.

도 32는, MFU 헤더 익스텐션(mfu_header_extension())의 구조예를 나타내는 도면이다.

도 33은, MFU 헤더 익스텐션의 구조예의 주요한 정보의 내용을 나타내는 도면이다.

도 34는, MFU 페이로드에 시각 정보를 삽입하는 경우에 있어서의 MFU 헤더(mfu_header())의 구조예를 나타내는 도면이다.

도 35는, MFU 페이로드에 시각 정보를 삽입하는 경우에 있어서의 MFU 헤더(mfu_header())의 구조예를 나타내는 도면이다.

도 36은, MFU 페이로드에 시각 정보를 삽입하는 경우에 있어서의 MFU 헤더의 구조예의 주요한 정보의 내용을 나타내는 도면이다.

도 37은, MFU 페이로드(mfu_payload())의 구조예 및 그 주요한 정보의 내용을 나타내는 도면이다.

도 38은, MFU 페이로드에 시각 정보를 삽입하는 경우에 있어서의 MFU 헤더(mfu_header())의 다른 구조예를 나타내는 도면이다.

도 39는, MFU 페이로드에 시각 정보를 삽입하는 경우에 있어서의 MFU 헤더(mfu_header())의 다른 구조예를 나타내는 도면이다.

도 40은, MFU 페이로드에 시각 정보를 삽입하는 경우에 있어서의 MFU 헤더의 다른 구조예의 주요한 정보의 내용을 나타내는 도면이다.

도 41은, MFU 페이로드(mfu_payload())의 구조예를 나타내는 도면이다.

도 42는, MFU 페이로드의 구조예의 주요한 정보의 내용을 나타내는 도면이다.

도 43은, 타임 스탬프 메시지(Timestamp Message)의 구조예(Syntax)를 나타내는 도면이다.

도 44는, 타임 스탬프 테이블(timestamp_table())의 구조예(Syntax)를 나타내는 도면이다.

도 45는, 패키지 액세스 메시지(Package Access message)의 구조예(Syntax)를 나타내는 도면이다.

도 46은, 액세스 유닛 AU(n)의 디코드 시각 D(n)와, 표시 시각 R(n)의 대응 관계의 일례를 나타내는 도면이다.

도 47은, 시계열의 오프셋값을 가변 길이 부호화하기 위한 가변 길이 부호 테이블의 일례를 나타내는 도면이다.

도 48은, 리오더 거리 M이 커질수록 가변 길이 부호화에 의한 전송 용량의 저감 효율이 높아지는 것을 나타내는 도면이다.

발명을 실시하기 위한 구체적인 내용

[0025] 이하, 발명을 실시하기 위한 형태(이하, 「실시 형태」로 함)에 대하여 설명한다. 또한, 설명을 이하의 순서로 행한다.

[0026] 1. 실시 형태

[0027] 2. 변형예

[0028] <1. 실시 형태>

[0029] [송수신 시스템의 구성예]

[0030] 도 1은, 실시 형태로서의 송수신 시스템(10)의 구성예를 나타내고 있다. 이 송수신 시스템(10)은, 트랜스포트 패킷 송신 장치(100)와, 트랜스포트 패킷 수신 장치(200)에 의해 구성되어 있다.

[0031] 송신 장치(100)는, MMT 구조(ISO/IEC CD 23008-1 참조)의 트랜스포트 패킷, 즉 MMT 패킷이 포함되는 전송 스트림을 생성하고, 이 전송 스트림을 RF 전송로 혹은 통신 네트워크 전송로를 통하여, 수신측으로 송신한다. 이 전송 스트림에는, 페이로드에 비디오나 오디오의 전송 미디어를 포함하는 제 1MMT 패킷과, 페이로드에 전송 미디어에 관한 정보를 포함하는 제 2MMT 패킷이, 적어도 프래그먼트된 패킷의 크기로 시분할적으로 다중화되어 있다. 본 실시 형태에 있어서, 제 1MMT 패킷 또는 제 2MMT 패킷에, 수신측에서 디코드 시각 및 표시 시각을 취득하기 위한 시각 정보가 삽입된다.

[0032] 수신 장치(200)는, 송신측으로부터, RF 전송로 혹은 통신 네트워크 전송로를 통하여, 전술한 전송 스트림을 수신한다. 수신 장치(200)는, 전송 스트림으로부터 취출한 전송 미디어를, 시각 정보에 기초하여 취득된 디코드 시각 및/또는 표시 시각을 사용해서 처리하고, 화상을 표시함과 함께 음성을 출력한다.

[0033] 도 2는, MMT 페이로드(MMT payload)의 구성을 개략적으로 나타내고 있다. 도 2에 있어서, MMT 패키지(MMT Package)는 MMT의 논리적인 개념이며, 전송 소재를 의미하고 있다. 이 MMT 패키지에는, 미디어인 에셋(Asset), 에셋의 전송 특성(Asset Delivery Characteristics), 패키지에 부수되는 메시지군(Package Access), MMT 패킷 테이블에 관한 정보(MPT Packet Table), 콤퍼지션 인포메이션(Composition Information) 등이 들어 있다. 콤퍼지션 인포메이션은, 미디어의 표시 제어를 담당하는 정보이다. 이 예에서는, 에셋 1(Asset1)이 비디오 1(Video1) 데이터이며, 에셋 2(Asset2)가 오디오 1(Audio1) 데이터이며, 에셋 3(Asset3)이 비디오 2(Video2) 데이터이다.

[0034] 도 2에는, MMT 패키지가 실제로 MMT 파일(MMT File)이 되어 있는 경우의 파일 구성의 일례를 나타내고 있다. 이 파일 구성은, 기본적으로는, MP4의 파일 구성과 거의 동등하다. 처음에 "styp"의 박스(Box)가 있다. 계속해서, 세그먼트 인포메이션으로서의 "sidx"의 박스가 있다. 계속해서, MMT 독자의 "mmpu"의 박스가 있다. 계속해서, 파일 전체의 메타데이터로서의 "moov"의 박스가 있다. 또한, 계속해서, "moof" 및 "mdat"의 박스가 있다. 이 "mdat"의 박스에는, 비디오(Video), 오디오(Audio), 자막 등의 실제 데이터가 들어 있다. 또한, "mdat"가 프래그먼트된 경우, 프래그먼트마다 "moof"의 박스가 있다.

[0035] MMT 패키지가 전송(Delivery)되는 경우, 도 2에 도시한 바와 같이, MPU(Media Processing Unit)를 단위로 하여 전송된다. 이 MPU는, 랜덤 액세스 포인트(RAP: Random Access Pint)로부터 시작되는 것이며, 1개 또는 복수의 액세스 유닛(AU:Access Unit)을 포함하는 것이다. 구체적으로는, 예를 들어 하나의 GOP(Group Of Picture)의 픽처가, 하나의 MPU의 구성으로 되는 경우가 있다. 이 MPU는, 에셋별로 정의되게 되어 있다. 따라서, 비디오의 에셋으로부터는 비디오 데이터만을 포함하는 비디오의 MPU가 작성되고, 오디오의 에셋으로부터는 오디오 데이터만을 포함하는 오디오의 MPU가 작성된다.

[0036] 도 2에 도시한 바와 같이, MPU는, 메시지(Message)와 함께, MMT 페이로드를 구성하게 된다. 메시지(Message)에는, 전술한 콤퍼지션 인포메이션 등의 정보가 포함된다. MFU(MMT Fragment Unit)는 MPU가 세분화, 즉 프래그먼트(Fragment)화된 것이다. 예를 들어, 비디오인 경우, 이 MFU를 하나의 NAL 유닛에 상당하도록 설정할 수 있다. 또한, 예를 들어 통신 네트워크 전송로에서 보내는 경우, 이 MFU를 1개 또는 복수의 MTU 사이즈(MTU size)로 구성할 수도 있다.

[0037] 도 2에 도시한 바와 같이, MMT 페이로드는, MMT 패킷에 의해 전송된다. 이 MMT 패킷은, MMT 패킷 헤더(MMT packet header)와, MMT 패킷 페이로드(MMT packet payload)에 의해 구성된다. 그리고, 이 MMT 패킷 페이로드는, MMT 페이로드 헤더(MMT payload header)와, MMT 페이로드 데이터(MMT payload data)에 의해 구성된다. 이 MMT 페이로드 데이터에, MPU나 메시지(Message)가 삽입된다.

[0038] 각 에셋의 MPU를 프래그먼트화하여 얻어진 MFU는, 도 2에 도시한 바와 같이, 시분할적으로 다중된다. MPU로서는, 예를 들어 15프레임분, 혹은 30프레임분이라는 어느 정도 긴 시간 단위의 것이다. 각 MPU를 프래그먼트화하여 시분할적으로 다중하지 않는 경우에는, 비디오 데이터를 보내고 있는 어느 정도 긴 시간 동안에, 오디오 데이터를 보낼 수 없게 된다. 그로 인해, 시간 맞춤을 위해 큰 버퍼 용량이 필요해짐과 함께, 화상, 음성의 출력까지의 지연이 커진다. 각 MPU를 프래그먼트화하여 시분할적으로 다중함으로써, 그와 같은 문제를 해결할 수 있다.

[0039] 도 3, 도 4, 도 5는, MMT 파일과, 실제로 전송되는 MMT 패킷의 대응 관계의 일례를 나타내고 있다. MMT 파일은, 전술한 바와 같이, "styp", "sidx", "mmpu", "moov"," moof", "mdat" 등의 박스를 갖고 있다.

[0040] MMT 패킷은, MMT 패킷 헤더(MMT Hdr)와, MMT 페이로드(MMT Payload)로 구성된다. MMT 헤더에는, 패킷 ID(packet_id), 패킷 시퀀스 넘버(packet_sequence_number), 트랜스미션 타임 스탬프(transmission_timestamp), 트랜스미션 프라이오리티(transmission_priority), 프라이빗 유저 데이터(private_user_data) 등이 포함된다.

[0041] 패킷 ID는, 비디오, 오디오의 에셋(Asset)이나, 컨트롤(Control)계인 메시지를 식별하기 위한 식별자이다. 패킷 시퀀스 넘버는, 패킷순을 나타내는 번호이다. 트랜스미션 타임 스탬프는, 전송을 위한 타입 스탬프, 즉 MMT 패킷이 송신측으로부터 나갈 때의 시각이다.

[0042] 트랜스미션 프라이오리티는, 전송로의 대역이 좁아진 경우에 어느 MMT 패킷을 우선적으로 통과할지를 판단하기 위한 지표가 되는 우선도이다. 프라이빗 유저 데이터는, 유저가 어떠한 방송 응용에 의해 사적으로 삽입할 수 있는 데이터이다. MMT 페이로드(MMT Payload)는 MMT 페이로드 헤더(MMT Pl_hdr)와, MMT 페이로드 데이터로 구성된다. MMT 페이로드 헤더에는, 페이로드 헤더 익스텐션(payload header extension)을 포함할 수 있다.

[0043] MMT 페이로드 헤더에는, 페이로드 길이(payload length), 페이로드 타입(payload type), 프래그먼트 타입(fragmentation_indicator), 프래그먼트 카운트(fragment_count), 애그리게이션 인포 플래그(aggregation_info_flag), RAP 플래그(random_access_point_flag) 등이 포함된다.

[0044] 또한, MMT 페이로드 헤더에는, 데이터 오프셋(data_offset), 데이터 유닛 넘버(numDU), 데이터 유닛 오프셋(DU_offset), 페이로드 시퀀스 넘버(payload_sequence_number), 헤더 익스텐션 필드 플래그(header_extension_field_flag) 등이 포함된다.

[0045] 페이로드 길이는, 페이로드 전체의 사이즈 정보이다. 페이로드 타입은, 페이로드가 MPU인지, 컨트롤계(메시지)인지를 나타내는 정보이다. 하나의 페이로드에는, 최대 64kbyte의 데이터를 넣을 수 있다. 프래그먼트 타입은, 하나의 페이로드 중에서 MPU가 완결되는지 여부를 나타내는 정보이다.

[0046] 예를 들어, MPU가 완결되는 경우에는 "0"이 삽입되고, 완결되지 않는 경우, 즉 소정수의 MFU로 세분화되어 있는 경우에는, "1", "2", "3" 중 어느 하나로 된다. "1"은 이 MMT 패킷에는 최초의 프래그먼트가 존재함을 나타낸다. "2"는 이 MMT 패킷에는 최초, 최후 중 어느 것도 아니며, 중간의 프래그먼트가 존재함을 나타낸다. "3"은 이 MMT 패킷에는 최후의 프래그먼트가 존재함을 나타낸다.

[0047] 프래그먼트 카운트는, MFU의 카운트 정보이다. 애그리게이션 인포 플래그는, 페이로드에 복수의 MPU가 들어 있는지 여부를 나타내는 플래그 정보이다. "0"은, 페이로드에 하나의 MPU만이 포함되는 것을 나타낸다. "1"은, 페이로드에 복수의 MPU가 포함되는 것을 나타낸다. RAP 플래그는, 이 MMT 패킷에 랜덤 액세스 포인트, 즉 GOP의 선두 픽처에 상당하는 액세스 유닛이 존재하는지 여부를 나타내는 정보이다.

[0048] 데이터 오프셋은, 페이로드 선두 위치로부터의 페이로드 데이터의 최초의 위치까지의 사이즈, 즉 페이로드 헤더의 크기를 나타내는 정보이다. 데이터 유닛 넘버는, 페이로드 내에 존재하는 MPU 데이터 유닛의 개수를 나타낸다. 데이터 유닛 오프셋은, 각 데이터 유닛의 페이로드 데이터의 선두 위치로부터의 오프셋 정보이다. 페이로드 시퀀스 넘버는, 이 MMT 패킷의 페이로드 시퀀스 넘버이다. 헤더 익스텐션 필드 플래그는, 페이로드 헤더 익스텐션이 존재하는지 여부를 나타내는 플래그 정보이다.

[0049] 도 3은, MMT 패킷화의 일례를 나타내고 있다. MMT 페이로드의 하나의 MPU 데이터 유닛에 MMT 파일의 "styp", "sidx", "mmpu", "moov", "moof"의 각 박스의 메타데이터가 삽입되고, MMT 패킷이 생성된다. 이 경우, MMT 페이로드 헤더에 있어서, 「fragmentation_indicator」는 "0", 「fragment_count」는 "0", 「aggregation_info_flag」는 "0", 「RAP flag」는 "1"로 된다.

[0050] 또한, 이 경우, MMT 페이로드의 하나의 MPU 데이터 유닛에 MMT 파일의 "mdat"에 존재하는 하나의 MPU가 삽입되고, MMT 패킷이 생성된다. 이 경우, MMT 페이로드 헤더에 있어서, 「fragmentation_indicator」는 "0", 「fragment_count」는 "0", 「aggregation_info_flag」는 "0", 「RAP_flag」는 "1"로 된다.

[0051] 도 4는, MMT 패킷화의 다른 일례를 나타내고 있다. 이 경우, 도 3의 예와 마찬가지로, MMT 페이로드의 하나의 MPU 데이터 유닛에 MMT 파일의 "styp", "sidx", "mmpu", "moov"의 각 박스의 메타데이터가 삽입되고, MMT 패킷이 생성된다. 또한, 이 경우, MMT 페이로드의 복수, 이 예에서는 3개의 MPU 데이터 유닛에 MMT 파일의 "mdat"에 존재하는 3개의 MPU가 삽입되고, MMT 패킷이 생성된다. 이 경우, MMT 페이로드 헤더에 있어서, 「fragmentation_indicator」는 "0", 「fragment_count」는 "0", 「aggregation_info_flag」는 "1", 「RAP_flag」는 "1", 「numDU」는 "3"으로 되고, 「DU offset」가 3개 존재한다.

[0052] 도 5는, MMT 패킷화의 또 다른 일례를 나타내고 있다. 이 경우, 도 3의 예와 마찬가지로, MMT 페이로드의 하나의 MPU 데이터 유닛에 MMT 파일의 "styp", "sidx", "mmpu", "moov", "moof"의 각 박스의 메타데이터가 삽입 혹은 소정의 정보의 변환이 이루어져서, MMT 패킷이 생성된다.

[0053] 또한, 이 경우, MMT 페이로드의 하나의 MPU 데이터 유닛에 MMT 파일의 "mdat"에 존재하는 하나의 액세스 유닛(AU)이 MFU 헤더(MFU Hdr)와 함께 삽입되어 MMT 패킷이 생성된다. 이 경우, MMT 페이로드 헤더에 있어서, 「fragmentation_indicator」는 "1", 「fragment_count」는 MFU의 카운트값인 "N", 「aggregation_info_flag」는 포함되는 MFU에 따라서 "1" 내지 "3" 중 어느 하나, 「RAP_flag」는 "1" 혹은 "0"으로 된다. 또한, MFU 헤더에는, MFU의 시퀀스 넘버(sequence_number) 등의 정보가 포함된다.

[0054] 도 6, 도 8은, 프래그먼트화된 MFU를 포함하는 MMT 패킷을 전송 패킷으로 하는 경우로서, 그 MMT 패킷이 랜덤 액세스(Random Access)의 선두(GOP의 선두)의 패킷인 경우를 나타내고 있다. 이 경우, 이 MMT 패킷은, MMT 파일의 "styp", "sidx", "mmpu", "moov", "moof"의 각 박스의 메타데이터가 삽입 혹은 소정의 정보의 변환이 이루어진 MMT 패킷과 함께 전송된다.

[0055] 예를 들어, "mmpu"의 영역에는, MFU의 데이터가 리얼 타임계 데이터인지 여부를 식별하는 정보가 포함되어 있지만, 수신측에서는, 랜덤 액세스(Random Access)의 선두에서, 예를 들어 이 식별 정보에 의해 MFU의 데이터가 리얼 타임계 데이터인지 여부의 식별이 가능하게 된다.

[0056] 도 7은, 프래그먼트화된 MFU를 포함하는 MMT 패킷을 전송 패킷으로 하는 경우로서, 그 MMT 패킷이 랜덤 액세스(Random Access)의 비(非) 선두의 패킷인 경우를 나타내고 있다. 이 경우, 이 MMT 패킷은, 전술한 선두 패킷과

는 달리, 최소 구성의 MMT 패킷으로 전송된다.

[0057] 도 9는, MMT 패킷의 구성을 트리 형식으로 나타낸 것이다. MMT 패킷은, 전술한 바와 같이, MMT 패킷 헤더(MMT Packet Header)와, MMT 페이로드 헤더(MMT Payload Header)와, MMT 페이로드(MMT Payload)에 의해 구성된다. MMT 페이로드에는, 메시지(Message), MPU(Media Processing Unit), FEC 수정 심볼(FEC Repair Symbol) 등이 포함되고, 이들 시그널링은 MMT 페이로드 헤더에 포함되는 페이로드 타입(payload_type)에 의해 행해진다.

[0058] 또한, 메시지에는, 각종 메시지 내용이 테이블 형식으로 삽입된다. 또한, MPU는, 프래그먼트화되어, MFU(MMT Fragment Unit)로 세분화되는 경우도 있다. 그 경우, 각 MFU의 선두에는, MFU 헤더(MFU Header)가 부가된다. MMT 페이로드에는, 비디오나 오디오의 미디어 데이터에 관한 MPU, 나아가, 메타데이터에 관한 MPU가 존재한다. 각 MPU를 포함하는 MMT 패킷은, MMT 패킷 헤더에 존재하는 패킷 ID(Packet_ ID)로 식별 가능하게 된다.

[0059] [송신 장치, 수신 장치의 개념]

[0060] 도 10은, 트랜스포트 패킷 송신 장치(100) 및 트랜스포트 패킷 수신 장치(200)의 개념도를 나타내고 있다. 송신 장치(100)는, 에셋 발생부(101)와, 인코더(102)와, 트랜스포트 패킷화부(103)와, 클럭 발생부(104)와, IP 송신부(105)와, RF 송신부(106)를 갖고 있다.

[0061] 에셋 발생부(101)는, 미디어 데이터로서의 비디오나 오디오의 데이터를 발생한다. 이 에셋 발생부(101)는, HDD(Hard Disk Drive)나 반도체 메모리 등을 포함하는 데이터 스토리지, 혹은 비디오 카메라, 마이크로폰 등이다. 인코더(102)는, 에셋 발생부(101)에서 발생되는 비디오나 오디오의 데이터에 대하여 부호화 처리를 행해 송신 데이터를 생성한다.

[0062] 트랜스포트 패킷화부(103)는, 인코더(102)에서 생성되는 송신 데이터를 MPU 단위, 혹은 그것을 세분화한 MFU 단위로 페이로드에 삽입한, 미디어 데이터를 포함하는 MMT 패킷을 생성하는 외에, 전술한 메타데이터를 포함하는 MMT 패킷이나 메시지를 포함하는 MMT 패킷을 생성한다. 트랜스포트 패킷화부(103)는, MMT 패킷을 생성할 때, 수신측에서 디코드 시각(Decode_Timestamp) 및 표시 시각(Display_Timestamp)을 취득하기 위한 시각 정보를, 그 MMT 패킷에 삽입한다. 이 MMT 패킷에의 시각 정보 삽입의 상세에 대해서는, 후술한다.

[0063] IP 송신부(105)는, 통신 네트워크 전송로를 사용하는 경우, 트랜스포트 패킷화부(103)로부터 순차 출력되는 MMT 패킷을 포함하는 전송 스트림을 IP 패킷화하고, 통신 네트워크 전송로를 통해 수신측으로 보낸다. RF 송신부(106)는, RF 전송로를 사용하는 경우, 트랜스포트 패킷화부(103)로부터 순차 출력되는 MMT 패킷을 포함하는 전송 스트림을 적절한 어댑테이션 레이어를 개재하는 등으로 하여 RF 변조하고, RF 전송로를 통해 수신측으로 보낸다. 여기서, 송신 장치(100)는, 클럭 발생부(104)로부터 출력되는 시스템 타임 클럭 STC에 기초하여, 송신 데이터에 송신 타임 스탬프를 부가한다.

[0064] 수신 장치(200)는, RF 수신부(201)와, IP 수신부(202)와, 클럭 재생부(203)와, 트랜스포트 언패킷화부(204)와, 디코드 출력 처리부(205)를 갖고 있다.

[0065] RF 수신부(201)는, 송신측으로부터 RF 전송로를 통해서 송신되는 MMT 패킷의 연속으로 이루어지는 전송 스트림을 어댑테이션 레이어의 해석을 거쳐 수신하고, 트랜스포트 언패킷화부(204)로 출력한다. 또한, IP 수신부(202)는, 송신측으로부터 통신 네트워크 전송로를 통해서 송신되는 MMT 패킷의 연속으로 이루어지는 전송 스트림을 수신하고, 트랜스포트 언패킷화부(204)로 출력한다.

[0066] 클럭 발생부(203)는, 시스템 타임 클럭 STC를 발생하고, 트랜스포트 언패킷화부(204) 등에 공급한다. 클럭 발생부(203)는, 송신 데이터에 부가되어 있는 송신 타임 스탬프값(NTP값)에 기초하여 클럭 세트를 행한다. 또는, NTP 패킷에서 공급되는 타임 스탬프값(NTP값)에 기초하여 클럭 세트를 행한다. 이 경우, 클럭 발생부(203)는, 발생하는 시스템 타임 클럭 STC와 타입 스탬프값(NTP값)의 차가 어떤 범위에 들어가도록, 발생하는 시스템 타임 클럭 STC의 보정을 행한다.

[0067] NTP가 MMT 패킷 헤더의 송신 시 타임 스탬프가 아니라, NTP 서버 등으로부터 취득되는 NTP 패킷에 의해 공급되는 경우, NTP의 포맷은, 도 11에 도시한 바와 같은 「NTP short 포맷」으로 한정되지 않는다. 도시는 생략하였지만, 64비트의 「NTP Timestamp Format(초의 정수 부분 32bits+초의 소수 부분 32bits)」으로 공급되는 경우도 있다(IETF RFC 5905 참조). 그 경우, 수신기 내에서는, 수신한 NTP의 값으로서 수신기의 디코더 클럭을 세트하고, 미디어의 타임 스탬프값과 비교된다. 이 경우, 미디어의 표시 혹은 디코드의 타임 스탬프와 수신기 클럭 사이의 정밀도의 차이는, 비교를 행할 때 수신기측에서 고려된다.

[0068] 트랜스포트 언패킷화부(204)는, RF 수신부(201) 혹은 IP 수신부(202)로부터 순차 공급되는 MMT 패킷의 언패킷화

를 행하여, 미디어 데이터로서의 수신 데이터를 얻는 외에, 메타데이터나 메시지를 얻는다. 디코드 출력 처리부(205)는, 트랜스포트 언패킷화부(204)에서 얻어지는 수신 데이터의 복호화를 행하여 비디오나 오디오의 데이터를 취득하고, 메타데이터나 메시지에 기초하여, 비디오 표시나 오디오 출력을 행한다.

[0069] 이 경우, 디코드 출력 처리부(205)는, 트랜스포트 언패킷화부(204)에서 추출되는 시각 정보에 기초하여, 액세스 유닛마다의 디코드 시각(Decode_Timestamp) 및/또는 표시 시각(Display_Timestamp)을 취득하고, 디코드 타이밍 및 표시 타이밍을 제어하고, 비디오 및 오디오의 동기 재생을 실현한다. 이 비디오 및 오디오의 동기 재생의 상세에 대해서는, 후술한다.

[0070] 도 10에 도시한 송신 장치(100) 및 수신 장치(200)의 동작을 간단히 설명한다. 처음에 송신 장치(100)의 동작을 설명한다. 에셋 발생부(101)에서 발생되는 비디오나 오디오의 데이터는 인코더(102)에 공급된다. 그리고, 이 인코더(102)에서는, 비디오나 오디오의 데이터에 대하여 부호화 처리가 행해져서 송신 데이터(부호화 데이터)가 생성된다.

[0071] 인코더(102)에서 생성된 송신 데이터는, 트랜스포트 패킷화부(103)에 공급된다. 그리고, 이 트랜스포트 패킷화부(103)에서는, 송신 데이터를 MPU 단위, 혹은 그것을 세분화한 MFU 단위로 페이로드에 삽입한, 미디어 데이터를 포함하는 MMT 패킷이 생성되는 외에, 전술한 메타데이터를 포함하는 MMT 패킷이나 메시지를 포함하는 MMT 패킷이 생성된다.

[0072] 트랜스포트 패킷화부(103)에서 순차 생성되는 MMT 패킷을 포함하는 전송 스트림은, IP 송신부(105) 혹은 RF 송신부(106)에 공급된다. IP 송신부(105)에서는, 통신 네트워크 전송로를 사용하는 경우, 전송 스트림이 IP 패킷화되고, 통신 네트워크 전송로를 통해서 수신측으로 보내진다. 또한, RF 송신부(106)에서는, RF 전송로를 사용하는 경우, 전송 스트림이 RF 변조되고, RF 전송로를 통해서 수신측으로 보내진다. 이때, 클럭 발생부(104)로부터 출력되는 시스템 타임 클럭 STC에 기초하여, 송신 데이터에 송신 타임 스탬프가 부가된다.

[0073] 다음으로, 수신 장치(200)의 동작을 설명한다. 송신측으로부터 보내져 오는 전송 스트림은, RF 수신부(201) 혹은 IP 수신부(202)에서 수신되고, MMT 패킷이 연속해서 포함되는 전송 스트림이 취득된다. 이 전송 스트림은, 트랜스포트 언패킷화부(204)에 공급된다. 이 트랜스포트 언패킷화부(204)에서는, MMT 패킷의 언패킷화가 행해지고, 미디어 데이터로서의 수신 데이터가 얻어지는 외에, 메타데이터나 메시지, 나아가 시각 정보 등이 얻어진다. 이들 데이터나 정보는, 디코드 출력 처리부(205)에 공급된다.

[0074] 디코드 출력 처리부(205)에서는, 트랜스포트 언패킷화부(204)에서 얻어지는 수신 데이터의 복호화가 행해져서 비디오나 오디오의 데이터가 취득되고, 메타데이터나 메시지에 기초하여, 비디오 표시나 오디오 출력이 행해진다. 이 경우, 디코드 출력 처리부(205)에는, 트랜스포트 언패킷화부(204)에서 MMT 패킷으로부터 추출되는 시각 정보가 공급된다. 그리고, 디코드 출력 처리부(205)에서는, 시각 정보에 기초하여, 액세스 유닛마다의 디코드 시각(Decode_Timestamp) 및/또는 표시 시각(Display_Timestamp)이 취득되고, 디코드 타이밍 및 출력 타이밍이 제어되고, 비디오 및 오디오의 동기 재생이 실현된다.

[0075] [MMT 패킷에 삽입하는 시각 정보]

[0076] 다음으로, MMT 패킷에 삽입되는 시각 정보에 대하여 설명한다. 이 시각 정보는, 전술한 바와 같이, 수신측에서 디코드 시각(Decode_Timestamp) 및/또는 표시 시각(Display_Timestamp)을 취득하기 위한 정보이다. MMT 패킷에 포함되는 미디어 데이터는 하나 이상의 액세스 유닛으로 구성되어 있다. MMT 패킷에 삽입되는 시각 정보는, 이 하나 이상의 액세스 유닛에 각각 대응한 시각 정보 및/또는 표시 시각을 취득하기 위한 정보이다.

[0077] 예를 들어, 제1 방법에서는, MMT 패킷에 삽입되는 시각 정보에는, 초기값, 즉 하나 이상의 액세스 유닛의 최초의 액세스 유닛에 대응한 디코드 시각 혹은 표시 시각의 값이 포함된다. 또한, 이 시각 정보에는, 각 액세스 유닛에 각각 대응한 디코드 시각으로부터 표시 시각까지의 오프셋값이 포함된다. 또한, 예를 들어 제2 방법에서는, MMT 패킷에 삽입되는 시각 정보에는, 표시 시각, 혹은 표시 시각 및 디코드 시각 그 자체가 포함된다.

[0078] 제1 방법에 대하여 설명한다. 오프셋값은, 절대 오프셋값에 대응한 상대 오프셋값으로 된다. 그로 인해, 시각 정보에는, 상대 오프셋값을 절대 오프셋값으로 변환하기 위한 변환 정보가 부가된다. 이 변환 정보에는, 「timestamp_type」, 「time tick」, 「au_rate_scale」 및 「division_factor」 등이 포함된다.

[0079] 「timestamp_type」은, 초기값이 디코드 시각의 초기값인지 표시 시각의 초기값인지를 나타내는 정보이다. 「time tick」은, 어떠한 클럭으로 제어할지를 나타내는 정보이다. 이 정보에 의해, 예를 들어 90㎑의 클럭으로 전부를 제어하는 것 등을 나타낸다. 「au_rate_scale」은, 비디오나 오디오의 액세스 유닛 레이트를 구하기 위

한 스케일 정보이다. 또한, 변환 정보로서, 또한 미디어 데이터가 비디오인지 오디오인지를 나타내는 플래그 정보 「Asset_type」도 포함된다.

[0080] 예를 들어, 비디오인 경우, 전술한 「time_tick」가 90㎑를 나타낼 때, 「au_rate_scale」은, 1500, 1800, 3000, 3600 등의 값으로 된다. 각각, 90㎑를 그 값으로서 나눔으로써, 60Hz, 50Hz, 30Hz, 25Hz 등의 비디오 레이트를 산출할 수 있다.

[0081] 또한, 「division_factor」는, 레이트를 미세 조정하기 위한 요소이다. 예를 들어, NTSC계의 비디오에 있어서, 30Hz인지 29.94Hz인지를 조정하기 위한 값이며, 「1」 혹은 「1.001」이다.

[0082] 여기서, 미디어 데이터가 비디오이며, 초기값이 디코드 시각인 경우에 대하여 설명한다. 도 12는, 0번째부터 6번째까지의 액세스 유닛(AU)에 대응한, 디코드 시각 「DTS(n)」, 오프셋값 「DLT(n)」, 표시 시각 「PTS(n)」의 대응 관계의 일례를 나타내고 있다. 이 예에서는, 디코드 시각의 초기값으로부터의 표시 시각의 초기값까지의 오프셋값(dlt_time)은 「1」로 되고, 상대 오프셋값으로 되어 있다.

[0083] 여기서, 0번째의 액세스 유닛은 I 픽처, 1번째의 액세스 유닛은 P 픽처, 2번째의 액세스 유닛은 B 픽처, 3번째의 액세스 유닛은 B 픽처, 4번째의 액세스 유닛은 P 픽처, 5번째의 액세스 유닛은 B 픽처, 6번째의 액세스 유닛은 B 픽처이다. 그로 인해, 디코드 시각에 대하여 표시 시각은 리오더(Reorder)되게 된다.

[0084] 이 경우, MMT 패킷에 삽입되어 송신되는 시각 정보는, 디코드 시각의 초기값인 TS0(여기에서는, TS0=0)과, 상대 오프셋값 DLT(n)으로 된다. 또한, 도 12에 있어서, 「Time」은 액세스 유닛마다의 시각을 상대값으로서 나타나 있다.

[0085] 이와 같은 시각 정보가 수신측으로 송신되는 경우, 수신측에서는, 이하와 같이, 각 액세스 유닛의 디코드 시각 및 표시 시각을, 변환 정보를 이용하여, 산출할 수 있다. 이 경우, 초기값 TS0은, 최초의 액세스 유닛의 디코드 시각 DTS(0)이다. 이 초기값 TS0으로부터의 오프셋 시간(dlt_time)은 이하의 수식 1로 구해진다. 예를 들어, 전형예로서, au_rate_scale=1500이고, division_factor=1.001인 경우, dlt_time=1500*1.001로 된다.

[0086] <식 1>

[0087] dlt_time=au_rate_scale*division_factor

[0088] 그 후의 액세스 유닛 디코드 시각 DTS(n)은, 이하의 수식 2에 나타낸 바와 같이, 오프셋 시간(dlt_time)을 「time_tick」로 나눈 값을, 직전의 디코드 시각 DTS(n-1)에 가산함으로써 구해진다. 단, DTS(0)=TS0이다.

[0089] <식 2>

[0090] DTS(n)=dlt_time/time_tick+DTS(n-1)

[0091] 또한, 각 액세스 유닛의 표시 시각 PTS(n)은, 이하의 수식 3에 나타낸 바와 같이, 오프셋값 DLT(n)에 「dlt_time/time_tick」를 곱한 값을, 디코드 시각 DTS(n)에 가산함으로써 구해진다.

[0092] <식 3>

[0093] PTS(n)=DLT(n)*(dlt_time/time_tick)+DTS(n)

[0094] 다음으로, 미디어 데이터가 비디오이며, 초기값이 표시 시각인 경우에 대하여 설명한다. 도 13은, 도 12와 마찬가지의, 0번째부터 6번째까지의 액세스 유닛(AU)에 대응한, 디코드 시각 「DTS(n)」, 오프셋값 「DLT(n)」, 표시 시각 「PTS(n)」의 대응 관계의 일례를 나타내고 있다. 이 예에서는, 디코드 시각의 초기값으로부터의 표시 시각의 초기값까지의 오프셋값(dlt_time)은 「1」로 되어 있다. 이 경우에도, 디코드 시각에 대하여 표시 시각은 리오더(Reorder)되게 된다.

[0095] 이 경우, MMT 패킷에 삽입되어 송신되는 시각 정보는, 표시 시각의 초기값인 TS0(여기에서는, TS0=1)과, 상대 오프셋값 DLT(n)으로 된다. 또한, 도 13에 있어서, 「Time」은 액세스 유닛마다의 시각을 상대값으로서 나타나 있다.

[0096] 이와 같은 시각 정보가 수신측으로 송신되는 경우, 수신측에서는, 이하와 같이, 각 액세스 유닛의 디코드 시각 및 표시 시각을, 변환 정보를 이용하여, 산출할 수 있다. 이 경우, 초기값 TS0은, 최초의 액세스 유닛의 표시 시각 PTS(0)이다. 그로 인해, 최초의 액세스 유닛 디코드 시각 DTS(0)은, 이하의 수식 4에 나타낸 바와 같이, PTS(0)으로부터, 오프셋 시간만큼, 즉 「dlt_time」을 「time_tick」으로 나눈 값으로 오프셋값 DLT(n)을 곱한

값을, 감산함으로써 구해진다.

[0097] <식 4>

[0098] DTS(0)=PTS(0)-(dlt_time/time_tick)*DLT(n)

[0099] 여기서, 「dlt_time」은, 이하의 수식 5에 나타낸 바와 같이, 「au_rate_scale」에 「division_factor」를 곱함으로써 구해진다.

[0100] <식 5>

[0101] dlt_time=au_rate_scale*division_factor

[0102] 그 후의 액세스 유닛 디코드 시각 DTS(n)은, 이하의 수식 6에 나타낸 바와 같이, 오프셋 시간(dlt_time)을 「time_tick」으로 나눈 값을, 직전의 디코드 시각 DTS(n-1)에 가산함으로써 구해진다.

[0103] <식 6>

[0104] DTS(n)=dlt_time/time tick+DTS(n-1)

[0105] 또한, 각 액세스 유닛의 표시 시각 PTS(n)은, 이하의 수식 7에 나타낸 바와 같이, 오프셋값 DLT(n)에 「dlt_time/time_tick」을 곱한 값을, 디코드 시각 DTS(n)에 가산함으로써 구해진다.

[0106] <식 7>

[0107] PTS(n)=DLT(n)*(dlt_time/time_tick)+DTS(n)

[0108] 다음으로, 미디어 데이터가 오디오이며, 초기값이 디코드 시각인 경우에 대하여 설명한다. 도 14는, 0번째부터 6번째까지의 액세스 유닛(AU)에 대응한, 디코드 시각 「DTS(n)」, 오프셋값 「DLT(n)」, 오디오 출력 시각 「PTS(n)」의 대응 관계의 일례를 나타내고 있다. 이 예에서는, 「dlt_time/time_tick」은 「1」로 되어 있다.

[0109] 또한, 오디오 액세스 유닛은, 복수의 오디오 샘플의 집합체이다. 여기서, 「dlt_time」은, 이하의 수식 8에 나타낸 바와 같이, 「au_rate_scale」에 「division_factor」를 곱함으로써 구해진다. 예를 들어, 전형예로서, 오디오 샘플링 주파수가 44.1㎑이고, 오디오 부호화 방식이 1024 샘플을 1개의 오디오 액세스 유닛으로 하는 경우에는, au_rate_scale=2089.8이고, division_factor=1이며, dlt_time=2089.8*1로 된다.

[0110] <식 8>

[0111] dlt_time =au_rate_scale*division_factor

[0112] 이 경우, MMT 패킷에 삽입되어 송신되는 시각 정보는, 오디오 출력 시각(표시 시각)의 초기값인 TS0(여기에서는, TS0=0)과, 상대 오프셋값 DLT(n)으로 된다. 또한, 도 14에 있어서, 「Time」은 액세스 유닛마다의 시각을 상대값으로서 나타나 있다. 통상의 경우, 리오더(Reorder)는 없으며, 디코딩한 후에는 그대로 출력 버퍼에 전송되므로, 디코드 시각 DTS(n)과 오디오 출력 시각 PTS(n)의 차분을 나타내는 DLT(n)은 「0」으로 된다.

[0113] 이와 같은 시각 정보가 수신측으로 송신되는 경우, 수신측에서는, 이하와 같이, 각 액세스 유닛의 디코드 시각 및 오디오 출력 시각을, 변환 정보를 이용하여, 산출할 수 있다. 이 경우, 초기값 TS0은, 최초의 액세스 유닛의 오디오 출력 시각 PTS(0)이지만, 그대로, 최초의 액세스 유닛의 디코드 시각 DTS(0)으로도 된다.

[0114] 그 후의 액세스 유닛 디코드 시각 DTS(n) 및 오디오 출력 시각 PTS(n)은, 이하의 수식 9에 나타낸 바와 같이, 오프셋 시간(dlt_time)을 「time_tick」로 나눈 값을, 직전의 오디오 출력 시각 PTS(n-1)에 가산함으로써 구해진다. 단, PTS(0)=TS0이다.

[0115] <식 9>

[0116] PTS(n)=DTS(n)=(dlt_time/time_tick)+PTS(n-1)

[0117] [수신 장치의 디코드 출력 처리부의 구성예]

[0118] 수신 장치(200)의 디코드 출력 처리부(205)에 대하여 더 설명한다. 도 15는, 디코드 출력 처리부(205)의 구성예를 나타내고 있다. 이 디코드 출력 처리부(205)는, 디멀티플렉서(301)와, 비디오 디코더(302)와, 비디오 표시부(303)와, 오디오 디코더(304)와, 오디오 출력부(305)와, 제어부(306)를 갖고 있다.

[0119] 디멀티플렉서(301)는, 트랜스포트 언패킷화부(204)의 출력으로부터 다양한 정보 혹은 데이터를 추출한다. 즉, 디멀티플렉서(301)는, 비디오의 액세스 유닛마다의 부호화 비디오 데이터를 추출함과 함께, 오디오의 액세스 유닛마다의 부호화 오디오 데이터를 추출한다.

[0120] 또한, 디멀티플렉서(301)는, 비디오, 오디오의 각 액세스 유닛의 디코드 시각 DTS(n) 및 표시 시각(오디오 출력 시각) PTS(n)을 취득하기 위한 시각 정보를 추출한다. 이 시각 정보에는, 전술한 바와 같이 초기값 TS0과, 상대 오프셋값 DLT(n)이 포함되어 있다.

[0121] 또한, 디멀티플렉서(301)는, 상대 오프셋값 DLT(n)을 절대 오프셋값으로 변환하기 위한 다양한 정보(변환 정보)를 추출한다. 이 변환 정보에는, 전술한 바와 같이, 「timestamp_type」, 「time_tick」, 「au_rate_scale」 및 「division_factor」 등이 포함되어 있다.

[0122] 비디오 디코더(302)는, 디멀티플렉서(301)에 의해 추출되는 비디오의 각 액세스 유닛의 부호화 비디오 데이터에 대해 디코드 처리를 행하여 각 액세스 유닛이 복호화된 비디오 데이터를 얻는다. 비디오 표시부(303)는, 비디오 디코더(302)에 의해 얻어진 각 액세스 유닛의 복호화 비디오 데이터에 기초하여 비디오 표시(화상 표시)를 행한다.

[0123] 오디오 디코더(304)는, 디멀티플렉서(301)에 의해 추출되는 오디오의 각 액세스 유닛의 부호화 오디오 데이터에 대하여 디코드 처리를 행하여 각 액세스 유닛이 복호화된 오디오 데이터를 얻는다. 오디오 출력부(305)는, 오디오 디코더(304)에 의해 얻어진 각 액세스 유닛의 복호화 오디오 데이터에 기초하여 오디오 출력(음성 출력)을 행한다.

[0124] 제어부(306)는, 디멀티플렉서(301)에 의해 추출되는 시각 정보 및 변환 정보에 기초하여, 비디오, 오디오의 각 액세스 유닛의 디코드 시각 DTS(n) 및 표시 시각(오디오 출력 시각) PTS(n)을 구한다. 그리고, 제어부(306)는, 이하와 같이, 오디오, 비디오의 동기 재생 제어를 행한다.

[0125] 즉, 제어부(306)는, 비디오의 각 액세스 유닛의 디코드 시각 DTS(n) 및 표시 시각 PTS(n)에 기초하여, 각 액세스 유닛의 디코드 타이밍 및 표시 타이밍을 제어한다. 이 경우, 각 액세스 유닛의 디코드를, 클럭 발생부(203)(도 7 참조)에서 발생되는 시스템 클럭 STC가 디코드 시각 DTS(n)으로 되는 타이밍에 개시하도록, 비디오 디코더(302)를 제어한다. 또한, 각 액세스 유닛에 의한 비디오 표시를, 이 시스템 클럭 STC가 표시 시각 PTS(n)으로 되는 타이밍에 개시하도록, 비디오 표시부(303)를 제어한다.

[0126] 또한, 제어부(306)는, 오디오의 각 액세스 유닛의 디코드 시각 DTS(n) 및 오디오 출력 시각 PTS(n)에 기초하여, 각 액세스 유닛의 디코드 타이밍 및 표시 타이밍을 제어한다. 이 경우, 각 액세스 유닛의 디코드를, 시스템 클럭 STC가 디코드 시각 DTS(n)으로 되는 타이밍에 개시하도록 오디오 디코더(304)를 제어한다. 또한, 각 액세스 유닛에 의한 오디오 출력을, 이 시스템 클럭 STC가 오디오 출력 시각 PTS(n)으로 되는 타이밍에 개시하도록, 오디오 출력부(305)를 제어한다.

[0127] 도 15에 도시한 디코드 출력 제어부(205)의 동작을 간단하게 설명한다. 트랜스포트 언패킷화부(204)의 출력은, 디멀티플렉서(301)에 공급된다. 이 디멀티플렉서(301)에서는, 비디오의 액세스 유닛마다의 부호화 비디오 데이터가 추출되고, 비디오 디코더(302)에 공급된다.

[0128] 비디오 디코더(302)에서는, 비디오의 각 액세스 유닛의 부호화 비디오 데이터에 대하여 디코드 처리가 행해지고, 각 액세스 유닛이 복호화된 비디오 데이터가 얻어진다. 이와 같이 복호화된 각 액세스 유닛의 비디오 데이터는, 비디오 표시부(303)에 공급된다. 비디오 표시부(303)에서는, 각 액세스 유닛의 비디오 데이터에 기초하여 비디오 표시(화상 표시)가 행해진다.

[0129] 또한, 디멀티플렉서(301)에서는, 오디오의 액세스 유닛마다의 부호화 오디오 데이터가 추출되고, 오디오 디코더(304)에 공급된다. 오디오 디코더(304)에서는, 오디오의 각 액세스 유닛의 부호화 오디오 데이터에 대하여 디코드 처리가 행해져서, 각 액세스 유닛이 복호화된 오디오 데이터가 얻어진다. 이와 같이 복호화된 각 액세스 유닛의 오디오 데이터는, 오디오 출력부(305)에 공급된다. 오디오 출력부(305)에서는, 각 액세스 유닛의 오디오 데이터에 기초하여 오디오 출력(음성 출력)이 행해진다.

[0130] 또한, 디멀티플렉서(301)에서는, 비디오, 오디오의 각 액세스 유닛의 디코드 시각 DTS(n) 및 표시 시각(오디오 출력 시각) PTS(n)을 취득하기 위한 시각 정보(초기값 TS0, 상대 오프셋값 DLT(n))가 추출된다. 또한, 디멀티플렉서(301)에서는, 상대 오프셋값 DLT(n)을 절대 오프셋값으로 변환하기 위한 다양한 변환 정보가 추출된다. 이 시각 정보 및 변환 정보는, 제어부(306)에 공급된다.

[0131] 제어부(306)에서는, 시각 정보 및 변환 정보에 기초하여, 비디오, 오디오의 각 액세스 유닛의 디코드 시각 DTS(n) 및 표시 시각(오디오 출력 시각) PTS(n)이 산출된다. 그리고, 제어부(306)에서는, 이 디코드 시각 DTS(n) 및 표시 시각(오디오 출력 시각) PTS(n)에 기초하여, 오디오, 비디오의 동기 재생 제어가 행해진다.

[0132] 이 경우, 제어부(306)에서는, 비디오의 각 액세스 유닛의 디코드 시각 DTS(n) 및 표시 시각 PTS(n)에 기초하여, 비디오의 각 액세스 유닛의 디코드 타이밍 및 표시 타이밍이 제어된다. 또한, 제어부(306)에서는, 오디오의 각 액세스 유닛의 디코드 시각 DTS(n) 및 오디오 출력 시각 PTS(n)에 기초하여, 오디오의 각 액세스 유닛의 디코드 타이밍 및 출력 타이밍이 제어된다.

[0133] 또한, 전술한 설명에서는, MMT 패킷에 삽입되는 시각 정보가 제1 방법에 의한 경우, 즉 시각 정보가, 디코드 시각 혹은 표시 시각의 초기값과, 각 액세스 유닛에 대응한 오프셋값인 경우를 설명하였다. 상세 설명은 생략하지만, MMT 패킷에 삽입되는 시각 정보가 제2 방법에 의한 경우, 즉 시각 정보가, 각 액세스 유닛의 표시 시각, 혹은 표시 시 시각 및 디코드 시각 그 자체인 경우도 생각된다. 그 경우, 제어부(306)에서는, 그들 시각이 그대로 사용된다.

[0134] 도 16은, 제어부(306)에 있어서의 AV 동기 재생 제어의 일례를 나타내는 타이밍차트이다. 도 16의 (a)는, 클럭 발생부(203)에서 발생되는 시스템 클럭 STC를 나타내고 있다. 이 시스템 클럭 STC는, 전술한 바와 같이, 송신측으로부터 보내져 오는 송신 타임 스탬프(NTP값), 혹은 NTP 패킷에서 공급되는 타임 스탬프값(NTP값)에 의해 클럭 세트가 행해진다.

[0135] 도 16의 (b)에 도시한 바와 같이, 시스템 클럭 STC가 비디오의 각 액세스 유닛의 디코드 시각 DTS(n)으로 될 때마다, 비디오의 각 액세스 유닛의 디코드가 개시된다. 그리고, 도 16의 (c)에 도시한 바와 같이, 시스템 타임 클럭 STC가 비디오의 각 액세스 유닛의 표시 시각 PTS(n)으로 될 때마다, 비디오의 각 액세스 유닛의 비디오 표시(화상 표시)가 개시된다.

[0136] 도 16의 (d)에 도시한 바와 같이, 시스템 클럭 STC가 오디오의 각 액세스 유닛의 디코드 시각 DTS(n)으로 될 때마다, 오디오의 각 액세스 유닛의 디코드가 개시됨과 함께, 오디오 출력(음성 출력)이 개시된다. 도 16의 (e)는, 오디오 샘플 출력을 나타내고 있다.

[0137] 또한, MFU에 프래그먼트화되어 있는 패킷이, 전송 상에서 드롭된 경우, 혹은, 수신 장치(200) 내에서 표시에 사용되지 않는다고 판단된 경우, 수신측에서 재생되는 타임 스탬프 테이블의 액세스 유닛 카운트와, 실제로 수신하는 액세스 유닛과 관계는, 도 17에 도시되게 된다. MFU 헤더의 시퀀스 넘버(sequence_number)에 의해 소트된 MFU 데이터로서 수신하는 액세스 유닛(AU)을 타임 스탬프 테이블 위에서 검사하고, 압축 버퍼로부터는 DTS(n)의 타이밍에 디코더로 전송한다. 또한, 디코드 버퍼로부터 PTS(n)의 타이밍에 표시용 프로세스로 전송한다. 이와 같이 하여, 수신하지 않는 액세스 유닛, 혹은 표시시키지 않는다고 판단한 액세스 유닛에 대한, 타임 스탬프(Timestamp)를 참조할지 여부를 판단한다.

[0138] [파일로부터 전송 패킷으로 변환할 때의 시각 정보의 생성 방법]

[0139] 파일의 상태로부터 전송 패킷으로 변환할 때, MMT 파일로부터 이하와 같이 취득된 정보에 기초하여, MMT 패킷에 삽입할 시각 정보(수신측에서 디코드 시각 및/또는 표시 시각을 취득하기 위한 시각 정보)가 생성된다. 파일 내의 정보로부터, MMT 패킷에 삽입하는 시각 정보의 생성은, 트랜스포트 패킷화부(103)에서 행해진다.

[0140] 처음에, 파일 전체에 대하여 「moov」 정보를 이용하는 경우에 대하여 설명한다. 이 경우, "Moov" 박스의 관리하의 'stts', 'ctts'의 박스이고, 파일 내에 포함되는 전체 샘플의 디코드 시각, 표시 시각이 공급된다. 이때, 'stts'(decoding time to sample)의 박스에 의해, 디코드 시각이 부여된다. 또한, 'ctts'(composition time to sample)의 박스에 의해 디코드 시각과 표시 시각의 차분이 부여된다. 또한, 'stss'(sync sample table)의 박스에서는, 랜덤 액세스의 샘플 위치가 나타난다.

[0141] 다음으로, 프래그먼트마다 「moof」 정보를 이용하는 경우에 대하여 설명한다. 이 경우, "Moof" 박스의 관리하의, 'trun', 'tfra'의 박스에서 프래그먼트마다의 샘플의 디코드 시각, 표시 시각이 공급된다. 이때, 'trun'(track fragment run)의 박스에 의해, 프래그먼트의 선두 위치로부터의 오프셋 위치와, 샘플의 디코드 시각, 표시 시각과의 오프셋(비디오인 경우에는 프레임 수, 오디오인 경우에는 오디오 샘플 수)이 공급된다. 또한, 'tfra'(track fragment random access)의 박스에 의해, 랜덤 액세스 위치와, 그 샘플의 디코드 시각이 공급되고, 또한 샘플마다의 디코드 시각과 표시 시각의 차분값으로부터, 표시 시각도 알 수 있다.

[0142] [시각 정보의 삽입 위치]

[0143] 다음으로, 시각 정보의 삽입 방법에 대하여 설명한다. 도 18은, MMT 패킷에 시각 정보를 삽입하는 방법을 개략적으로 나타내고 있다. 도 18의 (a)는, MPU 페이로드 헤더(MPU payload header) 중에서 시각 정보를 보내는 경우를 나타내고 있다. 이 경우, MPU의 페이로드가 복수의 액세스 유닛(AU)을 포함할 때에는, 페이로드 헤더 익스텐션(payload header extension)에, 그 복수의 액세스 유닛만큼의 시각 정보를 삽입한다. 한편, MPU 페이로드에 하나의 액세스 유닛을 갖는 구성에서는, 페이로드 헤더 익스텐션에, 개개의 액세스 유닛에 대하여 시각 정보를 삽입한다.

[0144] 도 18의 (b)는, MPU 페이로드 중에서 프래그먼트화된 액세스 유닛(AU)마다 시각 정보를 보내는 경우를 나타내고 있다. 이 경우, MFU에 하나의 액세스 유닛을 갖는 구성으로서, MFU 헤더 익스텐션(MFU header extension)에, 개개의 액세스 유닛에 대하여 시각 정보를 삽입한다. 또는, MFU 페이로드(MFU payload)에, 개개의 액세스 유닛에 대하여 시각 정보를 삽입한다. 또한, 도 18의 (c)는, 메시지(Message)로 시각 정보를 보내는 경우를 나타내고 있다. 이 경우, 적어도, RAP(랜덤 액세스 포인트) 단위로, 대상의 패킷 ID(packet_id)에서 관련짓는 미디어에 대한 시각 정보를 갖는 메시지를 전송한다.

[0145] 도 19는, 송수신 지연을 낮게 하기 위해서, 각 액세스 유닛에 대응하여 시각 정보(수신측에서 디코드 시각 및/또는 표시 시각을 취득하기 위한 시각 정보)를 삽입하는 경우의 패킷 구성의 일례를 나타내고 있다. 도 19의 (a)는, MMT 페이로드 헤더에 시각 정보를 삽입하는 경우의 패킷 구성예이다. 도 19의 (b)는, MFU 내에 시각 정보를 삽입하는 경우의 패킷 구성예이다. 또한, 도 19의 (c)는, MMT 메시지에 시각 정보를 삽입하는 경우의 패킷 구성예이다.

[0146] 또한, 각 패킷 구성예에 있어서, 대상의 액세스 유닛을 갖는 MMT 패킷이 랜덤 액세스의 선두 패킷, 즉 랜덤 액세스 포인트의 패킷인 경우에는, MMT 파일의 "styp", "sidx", "mmpu", "moov"의 각 박스의 메타데이터가 삽입된 MMT 패킷과 함께 전송된다. 한편, 이 MMT 패킷이 랜덤 액세스의 비 선두 패킷, 즉 비 랜덤 액세스 포인트의 패킷인 경우에는, 전술한 메타데이터가 삽입된 MMT 패킷은 전송되지 않는다.

[0147] MPU 페이로드 헤더(MPU payload header) 중에서 시각 정보를 보내는 경우에 대하여 더 설명한다. 도 20은, MMT 패킷 전체의 구조예(Syntax)를 나타내고 있다. MMT 패킷(mmt packet())에는, MMT 패킷 헤더(mmtp_header())와, MMT 페이로드(mmtp_payload())가 포함되어 있다. 또한, MMT 페이로드에는, MMT 페이로드 헤더(mmtp_payload_header())와, MMT 페이로드 데이터(mmtp_payload_data())가 포함되어 있다.

[0148] 도 21은, MMT 패킷 헤더(mmtp_header())의 구조예(Syntax)를 나타내고 있다. 상세 설명은 생략하지만, 이 MMT 패킷 헤더에는, 전술한 바와 같이, 패킷 ID(packet_id), 패킷 시퀀스 넘버(packet_sequence_number), 트랜스미션 타임 스탬프(transmission_timestamp), 트랜스미션 프라이오리티(transmission Priority), 프라이빗 유저 데이터(private user_data) 등이 포함되어 있다.

[0149] 도 22는, MMT 페이로드 헤더(mmtp_payload_header())의 구조예(Syntax)를 나타내고 있다. 상세 설명은 생략하지만, 이 MMT 페이로드 헤더에는, 전술한 바와 같이, 페이로드 길이(payload_length), 페이로드 타입(payload_type), 프래그먼트 타입(fragment_type), 프래그먼트 카운트(fragment_count), 애그리게이션 인포 플래그(aggregation_info_flag), RAP 플래그(random_access_point_flag), 데이터 오프셋(data_offset), 데이터 유닛 넘버(numDU), 데이터 유닛 오프셋(DU_offset), 페이로드 시퀀스 넘버(payload_seq_number), 헤더 익스텐션 필드 플래그(header_extension_field_flag) 등이 포함되어 있다.

[0150] 그리고, 헤더 익스텐션 필드 플래그가 "1"일 때, 이 MMT 페이로드 헤더에는, MMT 페이로드 헤더 익스텐션(mmtp_payload_header_extension())이 더 포함된다.

[0151] 도 23은, MMT 페이로드 헤더 익스텐션(mmtp_payload_header_extension())의 구조예(Syntax)를 나타내고 있다. 이 구조예는, MPU 페이로드 헤더(MPU payload header) 중에서 시각 정보를 보내는 경우에 대응하고 있다. 도 24는, 이 구조예의 주요한 정보의 내용(Semantics)을 나타내고 있다.

[0152] 「payload_header_extension_type」의 16비트 필드는, MMT 페이로드 헤더 익스텐션의 타입을 나타낸다. 예를 들어, "0×0001"은, 처리를 위한 시각 정보(타입 스탬프)를 제공함을 나타낸다. 「payload_header_extension_length」의 16비트 필드는, MMT 페이로드 헤더 익스텐션의 사이즈를 나타낸다. 「asset_type」 2비트 필드는, 에셋 타입을 나타낸다. 예를 들어, "01"은 비디오를 나타내고, "10"은 오디오를 나타낸다.

[0153] 「time_tick」의 2비트 필드는, 어떠한 클럭으로 제어하는지를 나타낸다. "01"은 90㎑의 정밀도의 클럭값을 나

타낸다. "10"은 「NTP short time」임을 나타낸다. 「au_rate_scale」의 3비트 필드는, 비디오나 오디오의 액세스 유닛 레이트를 구하기 위한 스케일 정보이다.

[0154] 에셋 타입이 비디오인 경우, "000"은 3750의 값을 나타내고, 이 값에 의해 레이트로서 24Hz를 구할 수 있다. 또한, "001"은 3600의 값을 나타내고, 이 값에 의해 레이트로서 25Hz를 구할 수 있다. 또한, "010"은 3000의 값을 나타내고, 이 값에 의해 레이트로서 30Hz를 구할 수 있다.

[0155] 또한, "011"은 1800의 값을 나타내고, 이 값에 의해 레이트로서 50Hz를 구할 수 있다. 또한, "100"은 1500의 값을 나타내고, 이 값에 의해 레이트로서 60Hz를 구할 수 있다. 또한, "101"은 900의 값을 나타내고, 이 값에 의해 레이트로서 100Hz를 구할 수 있다. 또한, "110"은 750의 값을 나타내고, 이 값에 의해 레이트로서 120Hz를 구할 수 있다. 또한, "111"은 리저브로 된다.

[0156] 한편, 에셋 타입이 오디오인 경우, "000"은 1920의 값을 나타내고, 이 값에 의해 레이트로서 48㎑*1024를 구할 수 있다. 또한, "001"은 2089.8의 값을 나타내고, 이 값에 의해 레이트로서 44.1㎑*1024를 구할 수 있다. 또한, "010"은 2880의 값을 나타내고, 이 값에 의해 레이트로서 32㎑*1024를 구할 수 있다. 또한, 그 밖에는 리저브로 된다.

[0157] 「division_factor」의 2비트 필드는, 레이트를 미세 조정하기 위한 요소이다. 에셋 타입이 비디오인 경우, "00"은 1을 나타내고, "01"은 1.001을 나타낸다. 또한, 에셋 타입이 오디오인 경우, "00"은 1을 나타내고, "01"은 2를 나타낸다.

[0158] 「timestamp_type」의 1비트 필드는, 초기값이 디코드 시각의 초기값인지 표시 시각의 초기값인지를 나타내는 정보이다. "1"은 디코드 시각(decode_timestamp)임을 나타내고, "0"은 표시 시각(display_timestamp)임을 나타낸다.

[0159] 「timestamp_for_processing」의 32비트 필드는, 초기값을 나타낸다. 이 경우, 「time_tick」가 "01"일 때는, 90㎑ 정밀도의 것으로 된다. 또한, 「time_tick」가 "10"일 때는, RFC5059로 규정되는 「NTP short timestamp」로 된다.

[0160] 「au_count_in_mpu」의 10비트 필드는, MPU에 포함되는 액세스 유닛(AU)의 개수를 나타낸다. 「delta_sequence_type」의 1비트 필드는, 오프셋값이 가변 길이 부호인지, 8비트 고정 길이인지를 나타낸다. "1"은 가변 길이 부호임을 나타내고, "0"은 8비트 고정 길이임을 나타낸다. 또한, 오프셋값의 가변 길이 부호화에 대해서는 후술한다.

[0161] 「delta_fixed_length_code」의 8비트 필드는, 8비트 고정 길이의 오프셋값을 삽입하는 영역이다. 「delta_variable_length_code」의 가변 길이 영역은, 가변 길이 부호로서의 오프셋값을 삽입하는 영역이다. 또한, 「tralling_filler()」는, MMT 페이로드 헤더 익스텐션(mmtp_payload_header_extension())의 사이즈가 바이트 얼라인되도록, 「delta_variable_length_code」의 누적에 대하여, 적절히 1 내지 7비트의 연속되는 "0"을 삽입하는 것이다.

[0162] 또한, 도 24의 내용(Semantics)에 추가하는 것으로, 비디오인 경우, time_tick=60Hz, au_rate_scale=1을 할당하고, 오디오인 경우, time_tick=44.1㎑, au_rate_scale=1을 할당할 수도 있다.

[0163] 또한, 전술에서는, MMT 페이로드 헤더 익스텐션에, 디코드 시각 혹은 표시 시각과, 오프셋값을 삽입하는, 제1 방법에 대응한 구조예를 나타내었다. 이 MMT 페이로드 헤더 익스텐션에 표시 시각, 혹은 표시 시각 및 디코드 시각을 삽입하는, 제2 방법에 대응한 구조예도 생각된다.

[0164] 도 25는, 그 경우에 있어서의 MMT 페이로드 헤더 익스텐션(mmtp_payload_header_extension())의 구조예(Syntax)를 나타내고 있다. 이 구조예도, MPU 페이로드 헤더(MPU payload header) 중에서 시각 정보를 보내는 경우에 대응하고 있다. 도 26은, 이 구조예의 주요한 정보의 내용(Semantics)을 나타내고 있다.

[0165] 「payload_header_extension_type」의 16비트 필드는, MMT 페이로드 헤더 익스텐션의 타입을 나타낸다. 예를 들어, "0×01"은, NTP short time 포맷의 표시 타임 스탬프(표시 시각)를 공급함을 나타낸다. "0×02"는, NTP short time 포맷의 표시 타임 스탬프와, 디코드 타임 스탬프(디코드 시각)를 공급함을 나타낸다. "0×03"은, 90㎑ 정밀도에서의 표시 타임 스탬프를 공급함을 나타낸다. "0×04"는, 90㎑ 정밀도에서의 표시 타임 스탬프와, 디코드 타임 스탬프를 공급함을 나타낸다.

[0166] 「payload_header_extension_length」의 16비트 필드는, MMT 페이로드 헤더 익스텐션의 사이즈를 나타낸다.

「presentation_timestamp」의 32비트 필드는, 표시 타임 스탬프(표시 시각)의 값을 나타낸다. 「decoding_timestamp」의 32비트 필드는, 디코드 타임 스탬프(디코드 시각)의 값을 나타낸다.

[0167] 다음으로, MPU 페이로드 중에서 프래그먼트화된 액세스 유닛(AU)마다 시각 정보를 보내는 경우에 대하여 더 설명한다. 이 경우, 예를 들어 도 27에 도시한 바와 같이, 액세스 유닛(AU)마다, 표시 시각(표시 타임 스탬프) PTS와, 오프셋값 DLT를, MFU 헤더 익스텐션(mfu_header_extension)의 영역에 삽입하여 전송한다. 또한, 도시를 생략하였지만, 표시 시각(표시 타임 스탬프) PTS 대신에 디코드 시각(디코드 타임 스탬프) DTS를 전송하는 것도 가능하다.

[0168] 도 28은, MFU의 구조예(Syntax)를 나타내고 있다. MFU에는, MFU 헤더(mfu_header())와, MFU 미디어 데이터(mfu_media_data())가 포함되어 있다. 도 29, 도 30은, MFU 헤더(mfu_header())의 구조예(Syntax)를 나타내고 있다. 도 31은, 이 구조예의 주요한 정보의 내용(Semantics)을 나타내고 있다.

[0169] 「sequence_number」의 32비트 필드는, MPU 내의 MFU의 시퀀스 번호이다. 「trackref_index」의 8비트 필드는, 미디어 트랙의 번호이다. 「sample_number」의 32비트 필드는, 당 MFU가 속하는 샘플(액세스 유닛)의 moof 박스 내의 순서를 나타낸다. 「priority」의 8비트 필드는, MPU 내의 당 MFU의 우선도를 나타낸다. 「dependency_counter」의 8비트 필드는, 당 MFU에 디코드가 의존하고 있는 MFU의 수를 나타낸다.

[0170] 「offset」의 16비트 필드는, mdat 박스로부터의 오프셋을 나타낸다. 「length」의 32비트 필드는, MFU의 크기를 나타낸다. 「multiLayer_flag」의 1비트 필드는, 멀티레이어 정보의 유무를 나타낸다. 「mfu_header_extension_flag」의 1비트 필드는, MFU 익스텐션(mfu_extension)의 유무를 나타낸다.

[0171] 「dependency_id」, 「depth_flag」, 「temporal_id」, 「quality_id」, 「priority_id」, 「view_id」, 「layer_id」는, MFU끼리의 다양한 의존 관계를 나타내는 ID 군이다. 「Item_ ID」는, 파일의 ID를 나타낸다. MFU 헤더 익스텐션 플래그(mfu_header_extension_flag)가 "1"일 때, MFU 헤더 익스텐션(mfu_header_extension())이 존재한다.

[0172] 도 32는, MFU 헤더 익스텐션(mfu_header_extension())의 구조예(Syntax)를 나타내고 있다. 도 33은, 이 구조예의 주요한 정보의 내용(Semantics)을 나타내고 있다. 「asset_type」 2비트 필드는, 에셋 타입을 나타낸다. 예를 들어, "01"은 비디오를 나타내고, "10"은 오디오를 나타낸다.

[0173] 「time_tick」의 2비트 필드는, 어떠한 클럭으로 제어하는지를 나타낸다. "01"은 90㎑의 정밀도의 클럭으로 전부를 제어함을 나타낸다. "10"은 「NTP short time」임을 나타낸다. 「au_rate_scale」의 10비트 필드는, 비디오나 오디오의 액세스 유닛 레이트를 구하기 위한 스케일 정보이다.

[0174] 에셋 타입이 비디오인 경우, "000"은 3750의 값을 나타내고, 이 값에 의해 레이트로서 24㎐를 구할 수 있다. 또한, "001"은 3600의 값을 나타내고, 이 값에 의해 레이트로서 25㎐를 구할 수 있다. 또한, "010"은 3000의 값을 나타내고, 이 값에 의해 레이트로서 30㎐를 구할 수 있다.

[0175] 또한, "011"은 1800의 값을 나타내고, 이 값에 의해 레이트로서 50㎐를 구할 수 있다. 또한, "100"은 1500의 값을 나타내고, 이 값에 의해 레이트로서 60㎐를 구할 수 있다. 또한, "101"은 900의 값을 나타내고, 이 값에 의해 레이트로서 100㎐를 구할 수 있다. 또한, "110"은 750의 값을 나타내고, 이 값에 의해 레이트로서 120㎐를 구할 수 있다. 또한, "111"은 리저브로 된다.

[0176] 한편, 에셋 타입이 오디오인 경우, "000"은 1920의 값을 나타내고, 이 값에 의해 레이트로서 48㎑*1024를 구할 수 있다. 또한, "001"은 2089.8의 값을 나타내고, 이 값에 의해 레이트로서 44.1㎑*1024를 구할 수 있다. 또한, "010"은 2880의 값을 나타내고, 이 값에 의해 레이트로서 32㎑*1024를 구할 수 있다. 또한, 그 밖에는 리저브로 된다.

[0177] 「division_factor」의 2비트 필드는, 레이트를 미세 조정하기 위한 요소이다. 에셋 타입이 비디오인 경우, "00"은 1을 나타내고, "01"은 1.001을 나타낸다. 또한, 에셋 타입이 오디오인 경우, "00"은 1을 나타내고, "01"은 2를 나타낸다.

[0178] 「timestamp_for_processing」의 32비트 필드는, 대응하는 액세스 유닛의 시각을 나타낸다. 이 경우, 「time_tick」가 "01"일 때는, 90㎑ 정밀도의 것으로 된다. 또한, 「time_tick」가 "10"일 때는, RFC5059로 규정되는 「NTP short timestamp」로 된다.

[0179] 「timestamp_type」의 1비트 필드는, 전술한 「timestamp_for_processing」으로 나타내는 시각이 디코드 시각

인지 표시 시각인지를 나타낸다. "1"은 디코드 시각(decode_timestamp)임을 나타내고, "0"은 표시 시각(display_timestamp)임을 나타낸다. 「delta_fixed_length_code」의 8비트 필드는, 8비트 고정 길이의 오프셋 값을 삽입하는 영역이다.

[0180] 또한, 전술에서는, MFU 헤더 익스텐션에, 시각 정보(예를 들어, 표시 시각 PTS 혹은 디코드 시각 DTS와, 오프셋 값 DLT)를 삽입하는 구조예를 나타내었다. MFU 페이로드에 시각 정보를 삽입하는 구조예도 고려된다. 도 34, 도 35는, 그 경우에 있어서의 MFU 헤더(mfu_header())의 구조예(Syntax)를 나타내고 있다. 도 36은, 이 구조예의 주요한 정보의 내용(Semantics)을 나타내고 있다.

[0181] 「timestamp_type」의 2비트 필드는, MPU 페이로드의 선두에 타임 스탬프가 공급되는지 여부를 나타낸다. "00"은, 타임 스탬프는 공급되지 않음을 나타낸다. 이 경우, MPU 페이로드의 최초의 바이트(1st 바이트)가 프래그먼트된 MPU 페이로드의 선두 바이트와 일치한다. "01"은, MPU 페이로드의 선두에 표시 타임 스탬프가 공급됨을 나타낸다. 이 경우, 표시 타임 스탬프의 직후에 프래그먼트된 MPU 페이로드의 선두 바이트가 공급된다. "10"은, MFU 페이로드의 선두에 디코드 타임 스탬프와 표시 타임 스탬프의 양쪽이 공급됨을 나타낸다. 이 경우, 타임 스탬프의 직후에 프래그먼트된 MPU 페이로드의 선두 바이트가 공급된다.

[0182] 「time_tick」의 1비트 필드는, 타임 스탬프값의 정밀도를 나타낸다. "0"은, 90000Hz 정밀도임을 나타낸다. "1"은, NTP short time 포맷임을 나타낸다.

[0183] 도 37은, MFU 페이로드(mfu_payload())의 구조예(Syntax) 및 그 주요한 정보의 내용(Semantics)을 나타내고 있다. 「presentation_timestamp」의 32비트 필드는, 표시 타임 스탬프(표시 시각)의 값을 나타낸다. 「decoding_timestamp」의 32비트 필드는, 디코드 타임 스탬프(디코드 시각)의 값을 나타낸다. MFU 헤더에 있어서의 「timestamp_type 」이 "01"일 때, 페이로드 데이터(payload_data)의 직전에 「presentation_timestamp」의 32비트 필드가 존재한다. 또한, MFU 헤더에 있어서의 「timestamp_type」이 "10"일 때, 페이로드 데이터(payload_data)의 직전에 「presentation_timestamp」의 32비트 필드 및 「decoding_timestamp」의 32비트 필드가 존재한다.

[0184] 도 38, 도 39는, MFU 페이로드에 시각 정보를 삽입하는 경우에 있어서의, MFU 헤더(mfu_header())의 다른 구조예(Syntax)를 나타내고 있다. 도 40은, 이 구조예의 주요한 정보의 내용(Semantics)을 나타내고 있다. 「payload_header_flag」의 1비트 플래그 정보는, MFU 페이로드의 선두에, MFU 페이로드 헤더가 존재하는지 여부를 나타낸다. "0"은, MFU 페이로드 헤더는 존재하지 않음을 나타낸다. 이 경우, MFU 페이로드의 최초의 바이트(1st 바이트)는 프래그먼트된 MPU 페이로드의 선두 바이트와 일치한다. "1"은, MFU 페이로드 헤더가 존재함을 나타낸다. 이 경우, MFU 페이로드 헤더의 직후에 프래그먼트된 MPU 페이로드의 선두 바이트가 공급된다.

[0185] 도 41은, MFU 페이로드(mfu_payload())의 구조예(Syntax)를 나타내고 있다. 도 42는, 이 구조예의 주요한 정보의 내용(Semantics)을 나타내고 있다. 「mfu_payload_header_type」의 8비트 필드는, MFU 페이로드 헤더의 타입을 나타낸다. 예를 들어, "0×01"은, NTP short time 포맷의 표시 타임 스탬프(표시 시각)를 공급함을 나타낸다. "0×02"는, NTP short time 포맷의 표시 타임 스탬프와, 디코드 타임 스탬프(디코드 시각)를 공급함을 나타낸다. "0×03"은, 90㎑ 정밀도에서의 표시 타임 스탬프를 공급함을 나타낸다. "0×04"는, 90㎑ 정밀도에서의 표시 타임 스탬프와, 디코드 타임 스탬프를 공급함을 나타낸다.

[0186] 「mfu_payload_header_length」의 8비트 필드는, 본 엘리먼트 이후의 MFU 페이로드 헤더의 사이즈를 바이트 수로 나타낸다. 「presentation_timestamp」의 32비트 필드는, 표시 타임 스탬프(표시 시각)의 값을 나타낸다. 「decoding_timestamp」의 32비트 필드는, 디코드 타임 스탬프(디코드 시각)의 값을 나타낸다.

[0187] 다음으로, 메시지(Message)에서 시각 정보를 보내는 경우에 대하여 더 설명한다. 도 43은, 타임 스탬프 메시지(Timestamp Message)의 구조예(Syntax)를 나타내고 있다. 이 타임 스탬프 메시지는, MPU에 포함되는 각 액세스 유닛(AU)의 디코드 시각(decode_timestamp), 표시 시각(display_timestamp)을 수신측에서 취득하기 위해서, 수신측의 시각 정보를 공급하기 위한 메시지이다.

[0188] 이 타임 스탬프 메시지에는, 메시지 내용이 배치되는 타임 스탬프 테이블(timestamp_table())이 포함된다. 도 44는, 이 타임 스탬프 테이블(timestamp_table())의 구조예(Syntax)를 나타내고 있다. 상세 설명은 생략하지만, 이 타임 스탬프 테이블에는, 전술한 MMT 페이로드 헤더 익스텐션(mmtp_payload_header_extension())(도 23, 도 24 참조)과 마찬가지의 정보(시각 정보, 변환 정보)가 포함된다.

[0189] 이 타임 스탬프 메시지에는, 패킷 ID(acket_ ID)가 포함되고, 에셋(asset) 레벨에서의 관련짓기가 행해진다.

그 밖에, 이 타임 스탬프 메시지에는, 종래의 테이블(table) 구성에 따라서, 테이블 ID(table_id), 버전(version), 길이(length) 등의 정보가 추가되어 있다. 여기서, 테이블 ID나 버전은, 시스템에 의해 고유하게 할당되는 것이다. 또한, 길이는, 당 테이블 전체의 사이즈를 나타낸다.

[0190] 이 타임 스탬프 테이블(timestamp_table)은, 메시지(message)군을 묶는 메시지이다. 패키지 액세스 메시지(Package Access message) 중 메시지 페이로드(message payload())로 전송하는 것도 가능하다. 도 45는, 패키지 액세스 메시지(Package Access message)의 구조예(Syntax)를 나타내고 있다.

[0191] 그 경우, 패키지 액세스 메시지로서는, 도시한 바와 같이, 헤더 부분에 패키지 액세스 메시지를 나타내는 「message_id」로부터 시작해서, 「version」, 「length」가 계속된다. 그리고,「number_of_tables」에서 테이블(table)의 수가 나타난 후, 그 테이블의 수만큼 루프(loop)가 있고, 각 루프 중에는 「table_id」로부터 시작해서, 「table_version」, 「table_length」가 계속된다. 「table_id」는, 타임 스탬프 테이블(timestamp_table) 중 「table_id」와 일치한다. 「table_length」는, 「timestamp_table」를 포함하고, 다른 「table」을 포함한 사이즈를 나타낸다. 그 후, 「message_payload」 중에, 「timestamp_table」이 그대로 공급된다.

[0192] [오프셋값의 가변 길이 부호화]

[0193] 다음으로, 오프셋값 DLT(n)의 가변 길이 부호화에 대하여 설명한다. 전술한 바와 같이, 액세스 유닛(AU)마다의 디코드 시각 및 표시 시각을 수신측에서 재현하기 위해서, 각 액세스 유닛의 오프셋값을 시계열적으로 전송하는 것이 행해진다. 이 오프셋값은, 배신순(=디코드순)의 계열을 표시순으로 정렬할 때 유효해지는 것이며, 주로 비디오에 관한 픽처 리오더(picture reorder)를 정확하게 행하고, 또한 동기 표시를 확보할 목적으로 참조된다.

[0194] 도 46은, 액세스 유닛 AU(n)의 디코드 시각 D(n)과, 표시 시각 R(n)의 대응 관계의 일례를 나타내고 있다. 이 예의 경우, 리오더 거리 M=3이며, 이하의 값을 액세스 유닛의 수만큼 부호화하게 된다.

[0195] D(0)→R(0)의 오프셋값 DLT=+1

[0196] D(1)→R(1)의 오프셋값 DLT=+3

[0197] D(2)→R(2)의 오프셋값 DLT=0

[0198] D(3)→R(3)의 오프셋값 DLT=0

[0199] 따라서, 오프셋값 DLT의 시계열 표현은, +1, +3, 0, 0, +3, 0, 0, 3, …으로 된다. 이것은, 리오더 거리 M에 의존하는 것이며, 리오더 거리 M이 커지면, "0"을 넣는 비율이 증대된다. 본 실시 형태에서는, 이것을 이용하여, 0 발생 확률에 따라서 유효한 부호 사이즈를 확보하여, 전송 용량에 맞출 목적으로 가변 길이 부호화를 행한다.

[0200] 도 47은, 시계열의 오프셋값을 가변 길이 부호화하기 위한 가변 길이 부호 테이블의 일례를 나타내고 있다. 고확률로 발생하는 "0"에 대하여 쇼트 워드(short word)가 할당되고, 리오더 거리 M의 단위로 발생하는 값으로는, 롱 워드(long word)가 할당된다. 이러한 가변 길이 부호화에 의해, 시계열의 오프셋값 전송 용량을 효율적으로 저감할 수 있다.

[0201] 또한, 리오더 거리 M이 커질수록 B 픽처의 비율이 커지므로, 이러한 가변 길이 부호화에 의한 전송 용량의 저감 효율은 크게 되어있다. 도 48은, 이러한 점을 나타내고 있다. 예를 들어, 리오더 거리 M이 3인 경우, 「IPBB」의 4개의 액세스 유닛의 오프셋값 전송 용량은 10비트로 되고, 평균하면, 10/4비트로 된다. 이에 반하여, 예를 들어 리오더 거리 M이 6인 경우, 「IPBBBBB」의 7개의 액세스 유닛의 오프셋값 전송 용량은 13비트로 되고, 평균하면, 13/7비트로 된다.

[0202] 전술한 바와 같이, 도 1에 도시한 송수신 시스템(10)에 있어서는, 송신 장치(100)로부터 수신 장치(200)로 보내지는 MMT 패킷에, 수신측에서 각 액세스 유닛의 디코드 시각 및/또는 표시 시각을 취득하기 위한 시각 정보가 삽입되는 것이다. 따라서, 예를 들어 수신측에서 디코드 시각 및/또는 표시 시각을 취득하기 위한 시각 정보의 전송을 양호하게 행할 수 있다.

[0203] 또한, 도 1에 도시한 송수신 시스템(10)에 있어서는, 시각 정보는, 예를 들어 하나 이상의 액세스 유닛의 최초의 액세스 유닛에 대응한 디코드 시각 혹은 표시 시각의 값(초기값)과, 각 액세스 유닛에 각각 대응한 디코드 시각으로부터 표시 시각까지의 오프셋값으로 되는 것이다. 또한, 이 오프셋값은 절대 오프셋값이 아니라, 상대 오프셋값으로 된다. 따라서, 시각 정보의 전송을 효율적으로 행할 수 있다.

[0204] 또한, 도 1에 도시한 송수신 시스템(10)에 있어서는, 시계열의 오프셋을 가변 길이 부호화하여 전송하는 것이다. 따라서, 시각 정보의 전송 용량의 저감을 도모할 수 있다.

[0205] <2. 변형예>

[0206] 또한, 전술한 실시 형태에 있어서는, 파일 내의 시각 정보를 MMT 패킷에 반영하는 방법을 기술(記述)하였지만, "moov" 내, 혹은 "moof" 내의 시각 정보를 그대로 MMT 페이로드 헤더 혹은 MFU 내의 영역, 또는 메시지에서 공급하는 것으로 해도 유사한 효과가 얻어진다.

[0207] 또한, 전술한 실시 형태에 있어서는, 본 기술을, MMT 구조의 트랜스포트 패킷, 즉 MMT 패킷이 포함되는 전송 스트림을 RF 전송로 혹은 통신 네트워크 전송로를 통해서 전송하는 송수신 시스템(10)에 적용한 것이다. 본 기술은, 마찬가지의 전송 스트림을 RF 전송로 혹은 통신 네트워크 전송로를 통해서 전송하는 송수신 시스템에도 마찬가지로 적용할 수 있음은 물론이다.

[0208] 또한, 본 기술은, 이하와 같은 구성을 취할 수도 있다.

[0209] (1) 페이로드에 전송 미디어를 포함하는 제1 전송 패킷과, 페이로드에 상기 전송 미디어에 관한 정보를 포함하는 제2 전송 패킷이 시분할적으로 다중화된 전송 스트림을 생성하는 전송 스트림 생성부와,

[0210] 상기 전송 스트림을 소정의 전송로를 통해서 수신측으로 송신하는 전송 스트림 송신부와,

[0211] 상기 제1 전송 패킷 또는 상기 제2 전송 패킷에, 상기 수신측에서 디코드 시각 및/또는 표시 시각을 취득하기 위한 시각 정보를 삽입하는 시각 정보 삽입부를 구비하는, 송신 장치.

[0212] (2) 상기 제1 전송 패킷에 포함되는 전송 미디어는 하나 이상의 액세스 유닛으로 구성되고,

[0213] 상기 시각 정보 삽입부에서 삽입되는 시각 정보는, 상기 하나 이상의 액세스 유닛에 각각 대응한 디코드 시각 및/또는 표시 시각을 취득하기 위한 정보인, 상기 (1)에 기재된 송신 장치.

[0214] (3) 상기 시각 정보 삽입부에서 삽입되는 시각 정보는, 상기 하나 이상의 액세스 유닛의 최초의 액세스 유닛에 대응한 디코드 시각 혹은 표시 시각의 값과, 각 액세스 유닛에 각각 대응한 디코드 시각으로부터 표시 시각까지의 오프셋값을 갖는, 상기 (2)에 기재된 송신 장치.

[0215] (4) 상기 오프셋값은, 절대 오프셋값에 대응한 상대 오프셋값이며,

[0216] 상기 시각 정보 삽입부에서 삽입되는 시각 정보에는, 상기 상대 오프셋값을 상기 절대 오프셋값으로 변환하기 위한 변환 정보가 부가되는, 상기 (3)에 기재된 송신 장치.

[0217] (5) 상기 시각 정보 삽입부는, 상기 상대 오프셋값을 가변 길이 부호화한 후에 삽입하는, 상기 (4)에 기재된 송신 장치.

[0218] (6) 상기 시각 정보 삽입부에서 삽입되는 시각 정보는, 상기 하나 이상의 액세스 유닛의 각각에 대응한 표시 시각, 혹은 표시 시각 및 디코드 시각인, 상기 (2)에 기재된 송신 장치.

[0219] (7) 상기 제1 전송 패킷의 페이로드는, 페이로드 헤더부와 페이로드 본체부를 포함하고,

[0220] 상기 시각 정보 삽입부는, 상기 페이로드 헤더부에, 상기 시각 정보를 삽입하는, 상기 (1) 내지 (6) 중 어느 하나에 기재된 송신 장치.

[0221] (8) 상기 제1 전송 패킷의 페이로드는, 페이로드 헤더부와 페이로드 본체부를 포함하고,

[0222] 상기 페이로드 본체부에는, 상기 전송 미디어가 프래그먼트화된 하나 이상의 액세스 유닛을 각각 포함하는 프래그먼트 페이로드가 프래그먼트 헤더를 수반하여 배치되고,

[0223] 상기 시각 정보 삽입부는, 상기 프래그먼트 헤더에, 대응하는 액세스 유닛의 시각 정보를 삽입하는, 상기 (1) 내지 (6) 중 어느 하나에 기재된 송신 장치.

[0224] (9) 상기 제1 전송 패킷의 페이로드는, 페이로드 헤더부와 페이로드 본체부를 포함하고,

[0225] 상기 페이로드 본체부에는, 상기 전송 미디어가 프래그먼트화된 하나 이상의 액세스 유닛을 각각 포함하는 프래그먼트 페이로드가 프래그먼트 헤더를 수반하여 배치되고,

[0226] 상기 시각 정보 삽입부는, 상기 프래그먼트 페이로드에, 대응하는 액세스 유닛의 시각 정보를 삽입하는, 상기

(1) 내지 (6) 중 어느 하나에 기재된 송신 장치.

[0227] (10) 상기 제2 전송 패킷의 페이로드부는, 페이로드 헤더부와 페이로드 본체부를 포함하고,

[0228] 상기 시각 정보 삽입부는, 상기 페이로드 본체부에, 상기 시각 정보를 삽입하는, 상기 (1) 내지 (6) 중 어느 하나에 기재된 송신 장치.

[0229] (11) 상기 소정의 전송로는, RF 전송로 또는 통신 네트워크 전송로인, 상기 (1) 내지 (10) 중 어느 하나에 기재된 송신 장치.

[0230] (12) 상기 제1 전송 패킷 및 상기 제2 전송 패킷은, MMT 패킷인, 상기 (1) 내지 (11) 중 어느 하나에 기재된 송신 장치.

[0231] (13) 페이로드에 전송 미디어를 포함하는 제1 전송 패킷과, 페이로드에 상기 전송 미디어에 관한 정보를 포함하는 제2 전송 패킷이 시분할적으로 다중화된 전송 스트림을 생성하는 전송 스트림 생성 스텝과,

[0232] 상기 전송 스트림을 소정의 전송로를 통해서 수신측으로 송신하는 전송 스트림 송신 스텝과,

[0233] 상기 제1 전송 패킷 또는 상기 제2 전송 패킷에, 상기 수신측에서 디코드 시각 및/또는 표시 시각을 취득하기 위한 시각 정보를 삽입하는 시각 정보 삽입 스텝을 구비하는, 송신 방법.

[0234] (14) 페이로드에 전송 미디어를 포함하는 제1 전송 패킷과, 페이로드에 상기 전송 미디어에 관한 정보를 포함하는 제2 전송 패킷이 시분할적으로 다중화된 전송 스트림을, 송신측으로부터 소정의 전송로를 통해서 수신하는 전송 스트림 수신부를 구비하고,

[0235] 상기 제1 전송 패킷 또는 상기 제2 전송 패킷에는, 디코드 시각 및/또는 표시 시각을 취득하기 위한 시각 정보가 삽입되어 있으며,

[0236] 상기 전송 스트림으로부터 취출된 상기 전송 미디어를, 상기 시각 정보에 기초하여 취득된 상기 디코드 시각 및/또는 상기 표시 시각을 사용해서 처리하는 전송 미디어 처리부를 더 구비하는, 수신 장치.

[0237] (15) 상기 제1 전송 패킷에 포함되는 전송 미디어는 하나 이상의 액세스 유닛으로 구성되고,

[0238] 상기 시각 정보는, 상기 하나 이상의 액세스 유닛의 최초의 액세스 유닛에 대응한 디코드 시각 혹은 표시 시각의 값과, 각 액세스 유닛에 각각 대응한 디코드 시각으로부터 표시 시각까지의 절대 오프셋값에 대응한 상대 오프셋값이며,

[0239] 상기 상대 오프셋값을 상기 절대 오프셋값으로 변환하는 오프셋값 변환부를 더 구비하는, 상기 (14)에 기재된 수신 장치.

[0240] (16) 페이로드에 전송 미디어를 포함하는 제1 전송 패킷과, 페이로드에 상기 전송 미디어에 관한 정보를 포함하는 제2 전송 패킷이 시분할적으로 다중화된 전송 스트림을, 송신측으로부터 소정의 전송로를 통해서 수신하는 전송 스트림 수신 스텝을 구비하고,

[0241] 상기 제1 전송 패킷 또는 상기 제2 전송 패킷에는, 디코드 시각 및/또는 표시 시각을 취득하기 위한 시각 정보가 삽입되어 있으며,

[0242] 상기 전송 스트림으로부터 취출된 상기 전송 미디어를, 상기 시각 정보에 기초하여 취득된 상기 디코드 시각 및/또는 상기 표시 시각을 사용해서 처리하는 전송 미디어 처리 스텝을 더 구비하는, 수신 방법.

[0243] 본 기술의 주된 특징은, MMT 패킷에, 복수의 액세스 유닛의 최초의 액세스 유닛에 대응한 디코드 시각 혹은 표시 시각의 값과, 각 액세스 유닛에 각각 대응한 디코드 시각으로부터 표시 시각까지의 상대 오프셋값을 삽입함으로써, 시각 정보의 전송을 효율적으로 행할 수 있도록 한 것이다(도 15, 도 18 참조).

부호의 설명

[0244] 10: 송수신 시스템

100: 트랜스포트 패킷 송신 장치

101: 에셋 발생부

102: 인코더

103: 트랜스포트 패킷화부

104: 클럭 발생부

105: IP 송신부

106: RF 송신부

200: 트랜스포트 패킷 수신 장치

201: RF 수신부

202: IP 수신

203: 클럭 발생부

204: 트랜스포트 언패킷화부

205: 디코드 출력 처리부

301: 디멀티플렉서

302: 비디오 디코더

303: 비디오 표시부

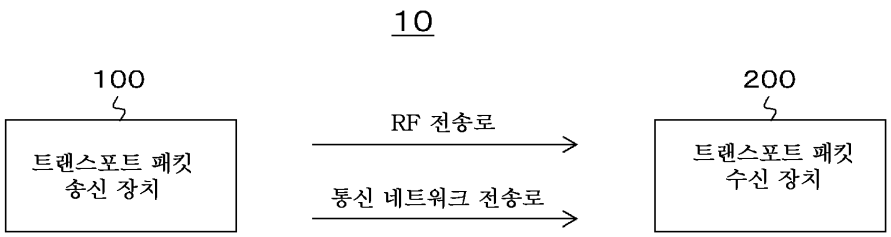
304: 오디오 디코더

305: 오디오 출력부

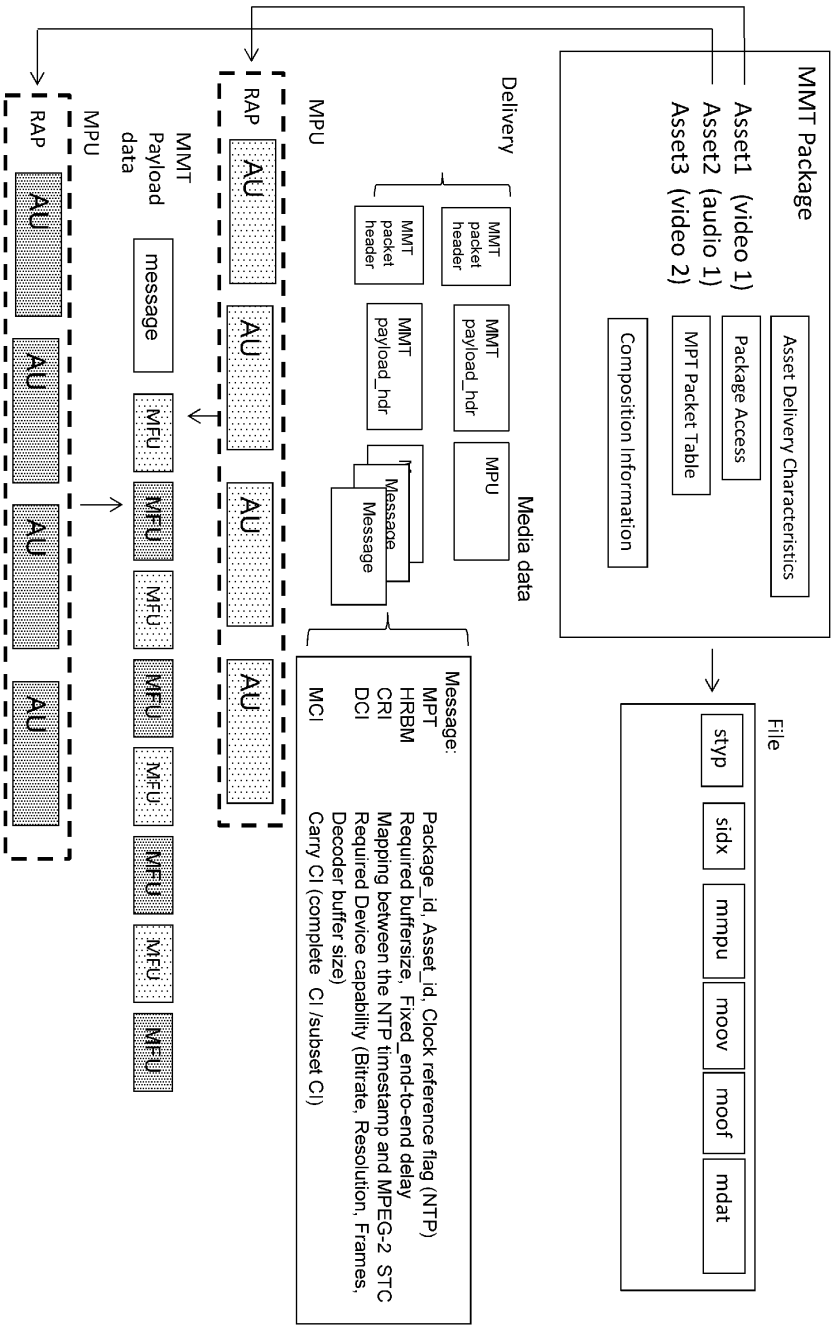
306: 제어부

도면

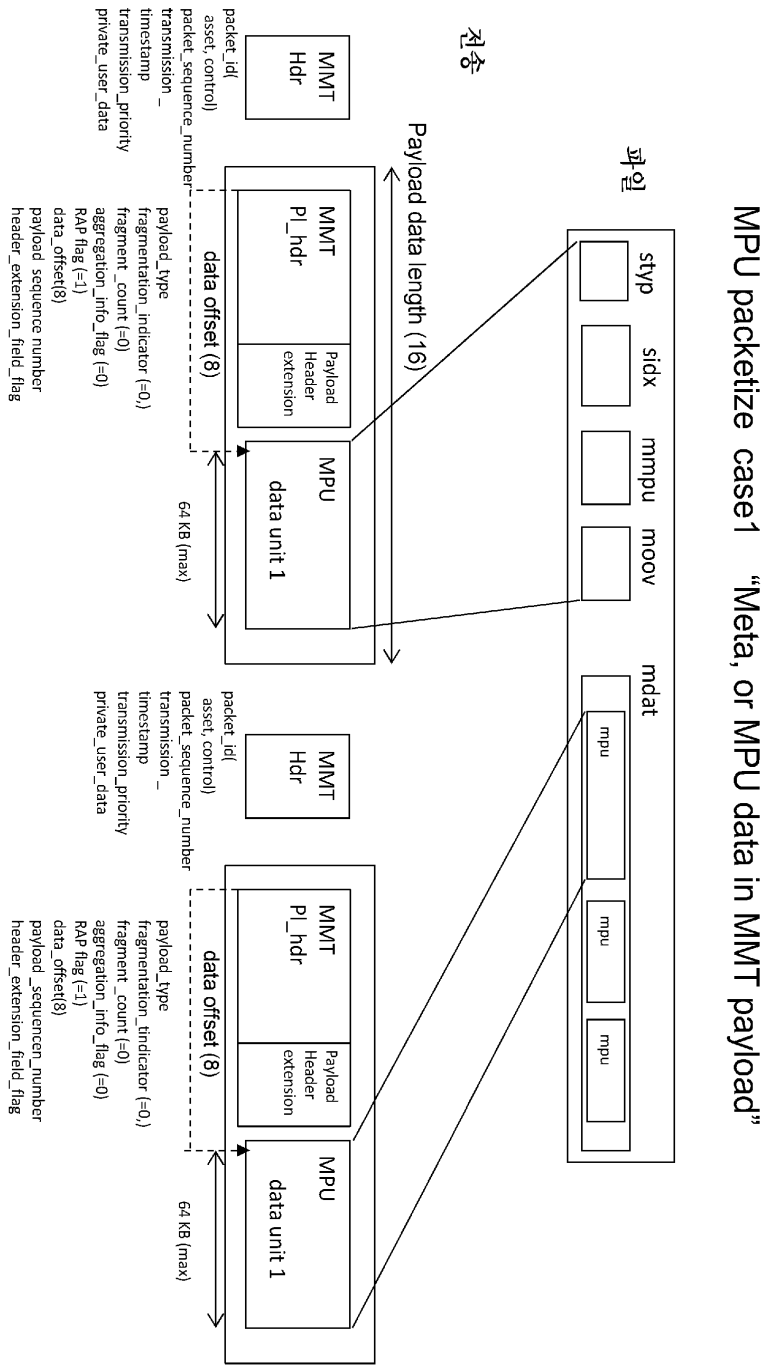
도면1



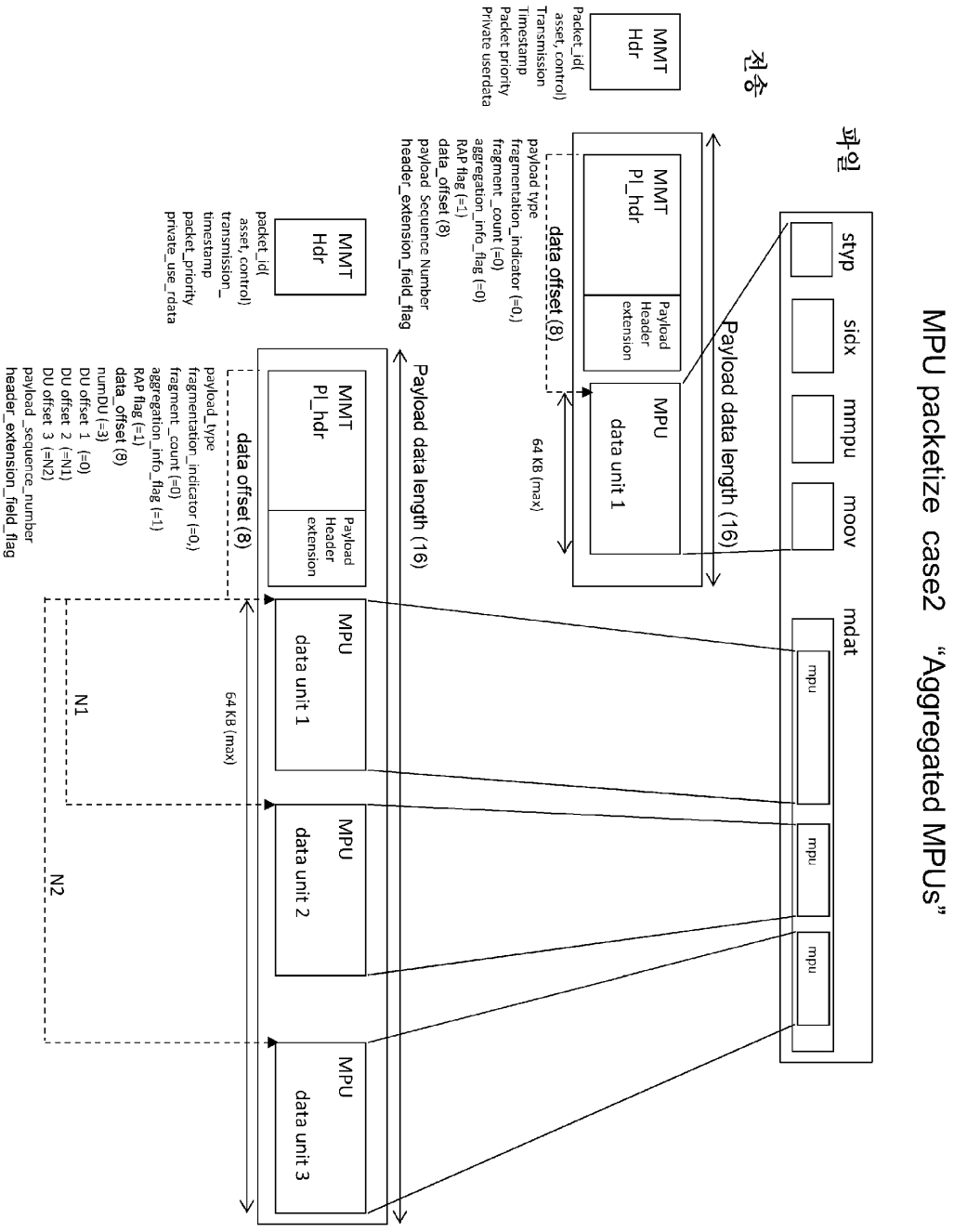
도면2


MMT Package
Asset1 (video 1)
Asset2 (audio 1)
Asset3 (video 2)
Asset Delivery Characteristics
Package Access
MPT Packet Table
Composition Information
File
styp
sidx
mmpu
moov
moof
mdat
Delivery
Media data
MMT packet header
MMT payload_hdr
MPU
Message
Message:
MPT
HRBM
CRI
DCI
MCI
Package_id, Asset_id, Clock reference flag (NTP)
Required buffersize, Fixed_end-to-end delay
Mapping between the NTP timestamp and MPEG-2 STC
Required Device capability (Bitrate, Resolution, Frames,
Decoder buffer size)
Carry CI (complete CI /subset CI)
MPU
RAP
AU
MMT Payload data
message
MFU

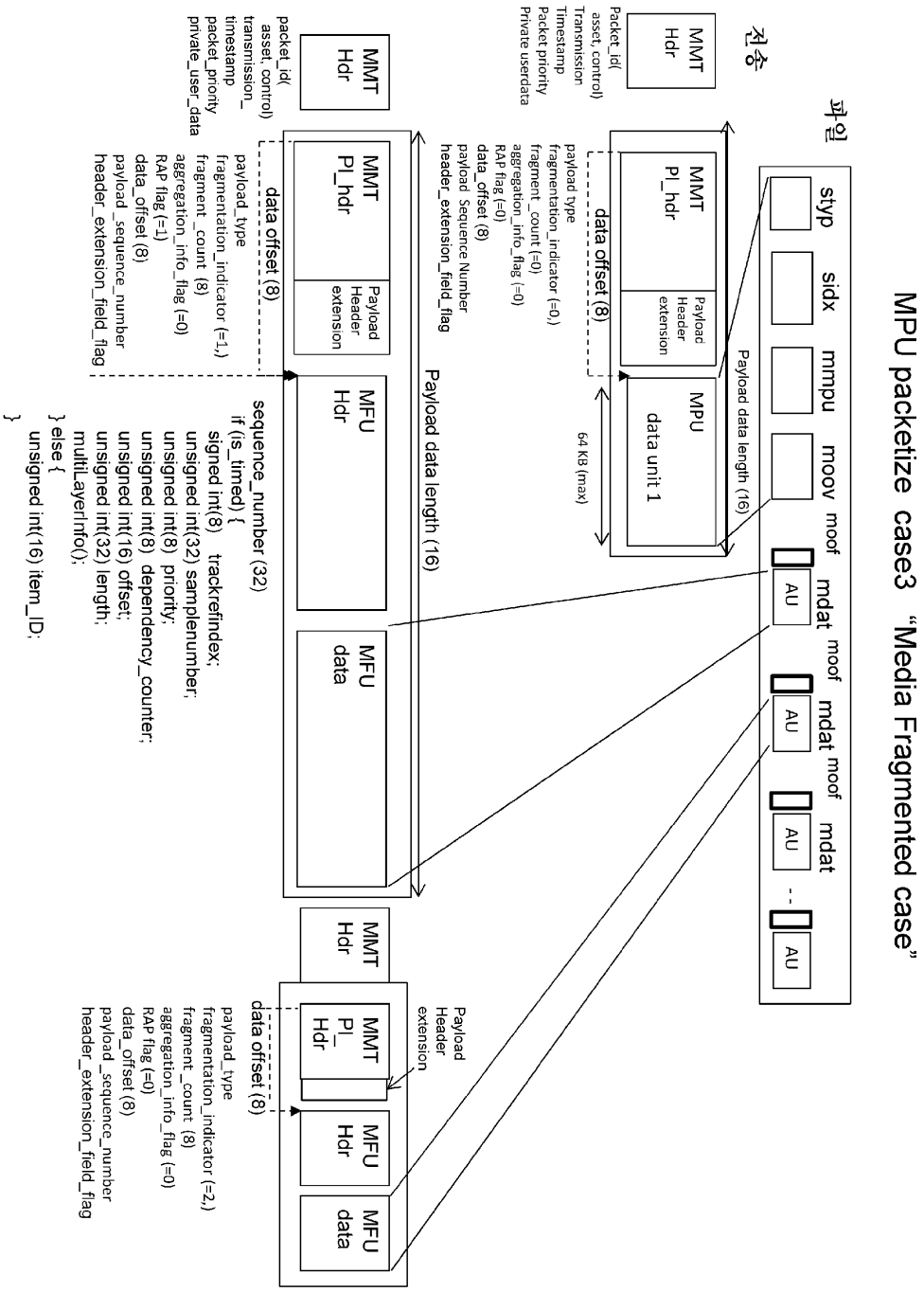
도면3


MPU packetize case1 "Meta, or MPU data in MMT payload"
파일
styp
sidx
mmpu
moov
mdat
mpu
mpu
mpu
전송
Payload data length (16)
MMT Hdr
MMT PI_hdr
Payload Header extension
MPU data unit 1
data offset (8)
64 KB (max)
packet_id(
asset, control)
packet_sequence_number
transmission _ timestamp
transmission_priority
private_user_data
payload_type
fragmentation_indicator (=0,)
fragment _count (=0)
aggregation_info_flag (=0)
RAP flag (=1)
data_offset(8)
payload_sequencen_number
header_extension_field_flag

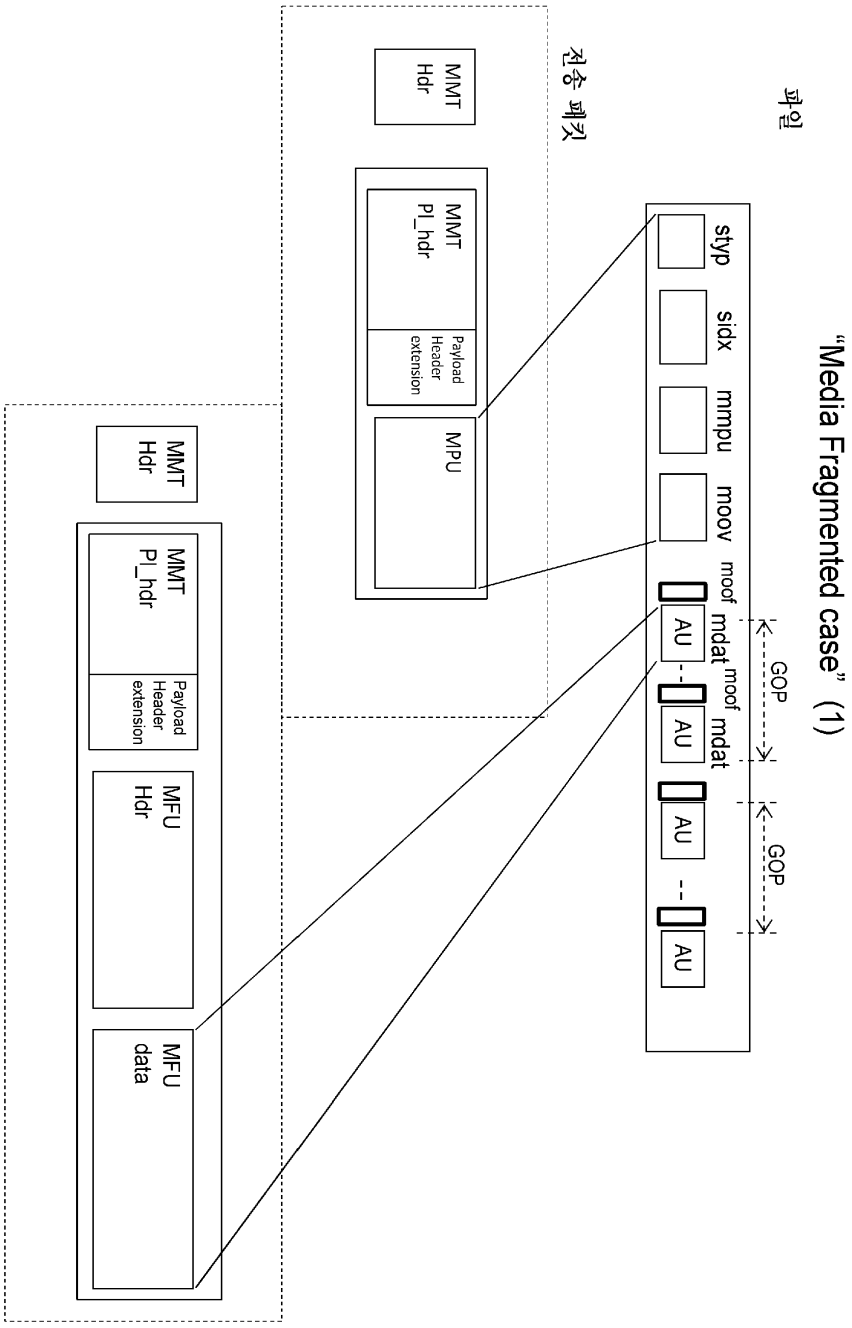
도면4


MPU packetize case2 "Aggregated MPUs"
파일
styp
sidx
mmpu
moov
mdat
mpu
mpu
mpu
Payload data length (16)
전송
MMT Hdr
Packet_id(
asset, control)
Transmission Timestamp
Packet priority
Private userdata
MMT PI_hdr
Payload Header extension
MPU data unit 1
data offset (8)
payload type
fragmentation_indicator (=0,)
fragment_count (=0)
aggregation_info_flag (=0)
RAP flag (=1)
data_offset (8)
payload Secuence Number
header_extension_field_flag
64 KB (max)
MMT Hdr
packet_id(
asset, control)
transmission_
timestamp
packet_priority
private_use_rdata
MMT PI_hdr
Payload Header extension
Payload data length (16)
data offset (8)
payload_type
fragmentation_indicator (=0,)
fragment_count (=0)
aggregation_info_flag (=1)
RAP flag (=1)
data_offset (8)
numDU (=3)
DU offset 1 (=0)
DU offset 2 (=N1)
DU offset 3 (=N2)
payload_sequence_number
header_extension_field_flag
MPU data unit 1
MPU data unit 2
MPU data unit 3
64 KB (max)
N1
N2

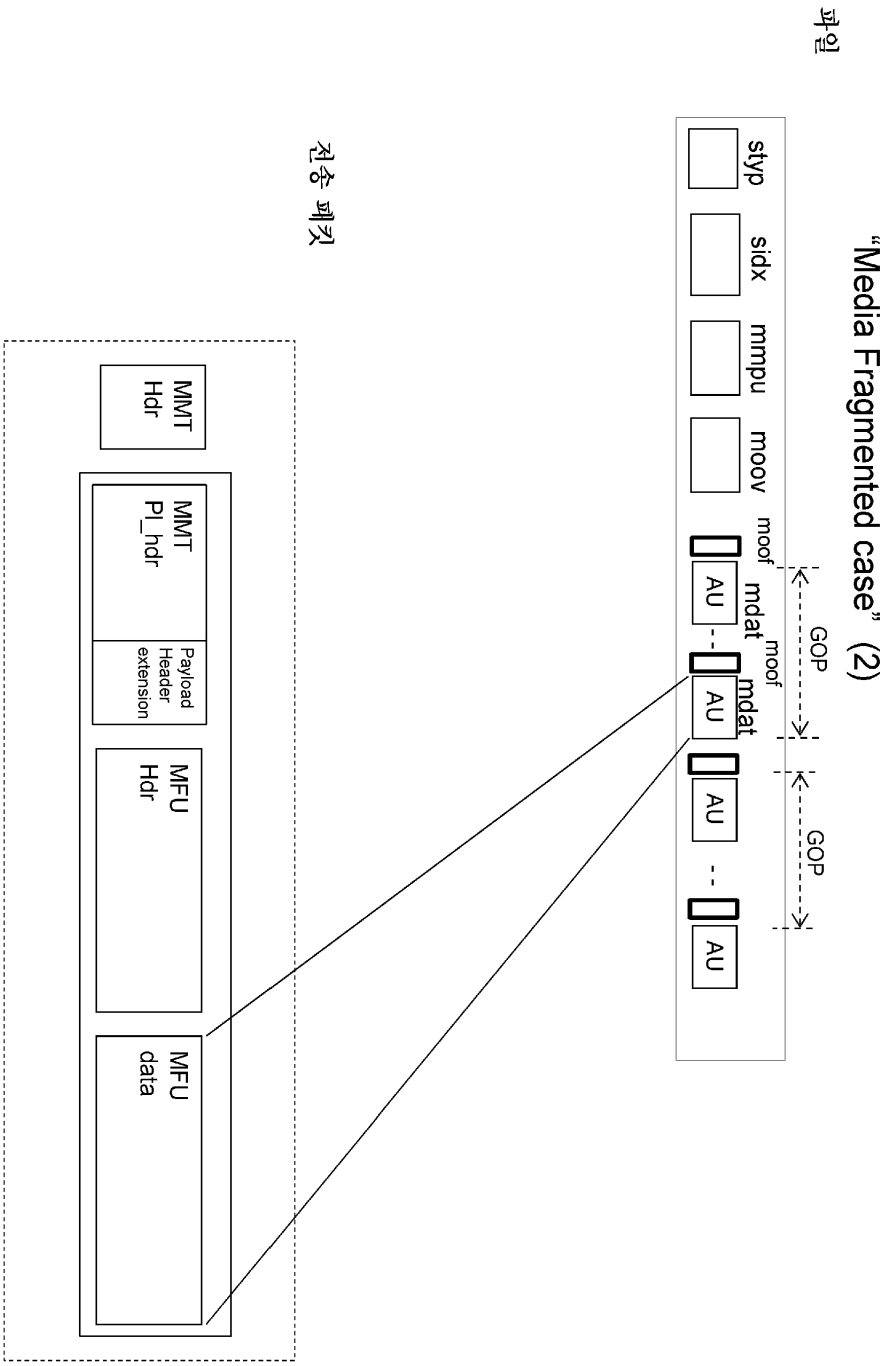
도면5


MPU packetize case3 "Media Fragmented case"
파일
styp
sidx
mmpu
moov
moof
mdat
moof
mdat
moof
mdat
AU
AU
AU
AU
AU
Payload data length (16)
64 KB (max)
MPU
data unit 1
MMT Pl_hdr
Payload Header extension
data offset (8)
전송
MMT Hdr
Packet_id(asset, control)
Transmission Timestamp
Packet priority
Private userdata
payload type
fragmentation_indicator (=0,)
fragment_count (=0)
aggregation_info_flag (=0)
RAP flag (=0)
data_offset (8)
payload Sequence Number
header_extension_field_flag
Payload data length (16)
MMT Hdr
MMT Pl_hdr
Payload Header extension
data offset (8)
MFU Hdr
MFU data
packet_id(asset, control)
transmission_ timestamp
packet_priority
private_user_data
payload_type
fragmentation_indicator (=1,)
fragment_count (8)
aggregation_info_flag (=0)
RAP flag (=1)
data_offset (8)
payload_sequence_number
header_extension_field_flag
sequence_number (32)
if (is_timed) {
signed int(8) trackrefindex;
unsigned int(32) samplenumber;
unsigned int(8) priority;
unsigned int(8) dependency_counter;
unsigned int(16) offset;
unsigned int(32) length;
multiLayerInfo();
} else {
unsigned int(16) item_ID;
}
MMT Hdr
MMT Pl_ Hdr
Payload Header extension
data offset (8)
MFU Hdr
MFU data
payload_type
fragmentation_indicator (=2,)
fragment_count (8)
aggregation_info_flag (=0)
RAP flag (=0)
data_offset (8)
payload_sequence_number
header_extension_field_flag

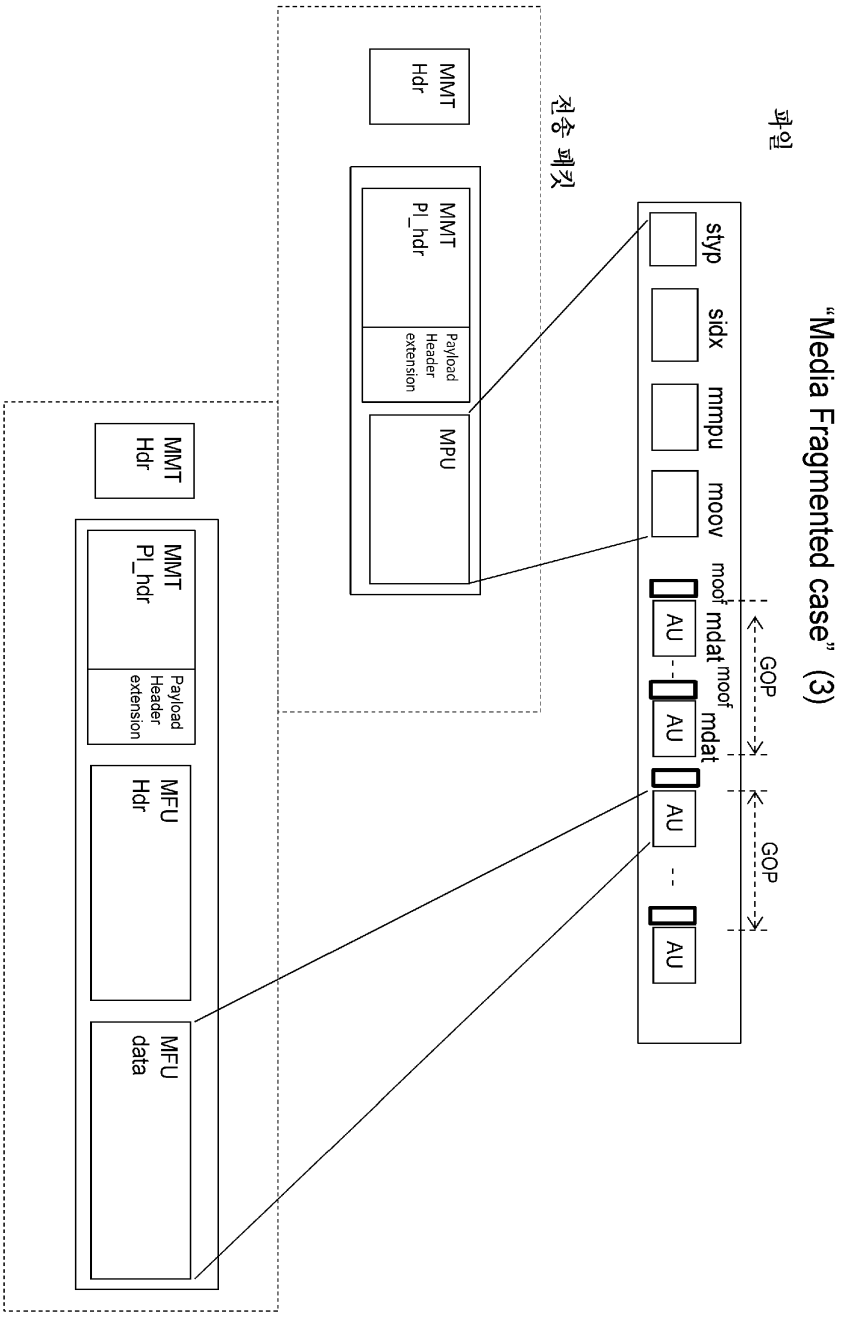
도면6


"Media Fragmented case" (1)
파일
styp
sidx
mmpu
moov
moof
mdat
moof
mdat
AU
AU
AU
AU
GOP
GOP
전송 패킷
MMT Hdr
MMT PI_hdr
Payload Header extension
MPU
MMT Hdr
MMT PI_hdr
Payload Header extension
MFU Hdr
MFU data

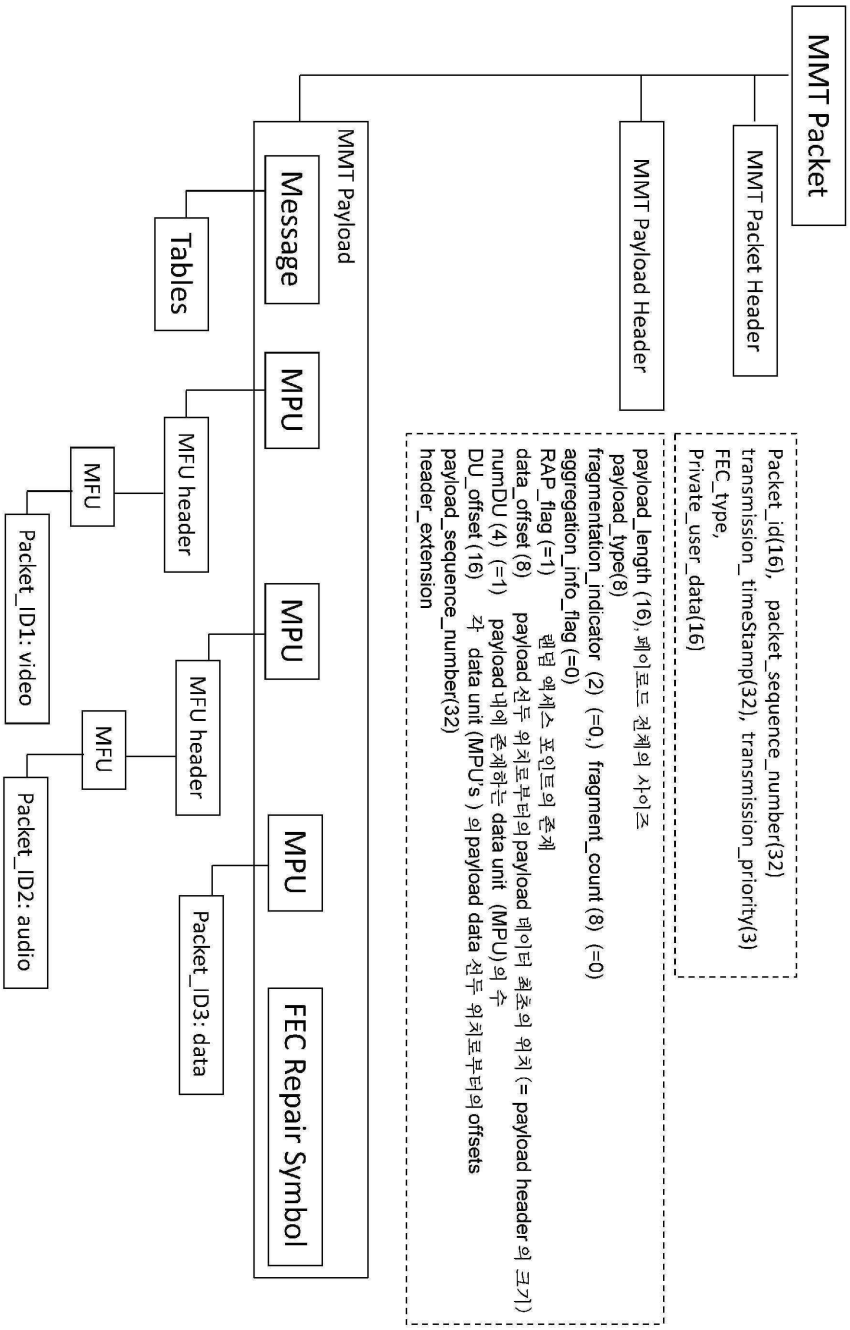
도면7


"Media Fragmented case" (2)
파일
styp
sidx
mmpu
moov
moof
AU
mdat
moof
mdat
GOP
GOP
AU
AU
AU
전송 패킷
MMT Hdr
MMT PI_hdr
Payload Header extension
MFU Hdr
MFU data

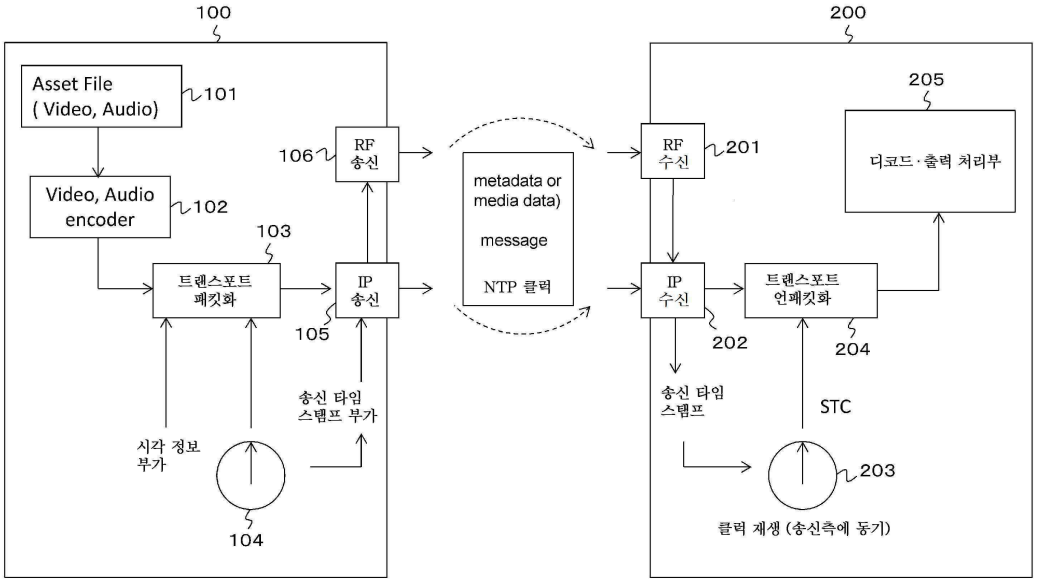
도면8


"Media Fragmented case" (3)
파일
styp
sidx
mmpu
moov
moof
mdat
moof
mdat
AU
AU
AU
AU
GOP
GOP
전송 패킷
MMT Hdr
MMT PI_hdr
Payload Header extension
MPU
MMT Hdr
MMT PI_hdr
Payload Header extension
MFU Hdr
MFU data

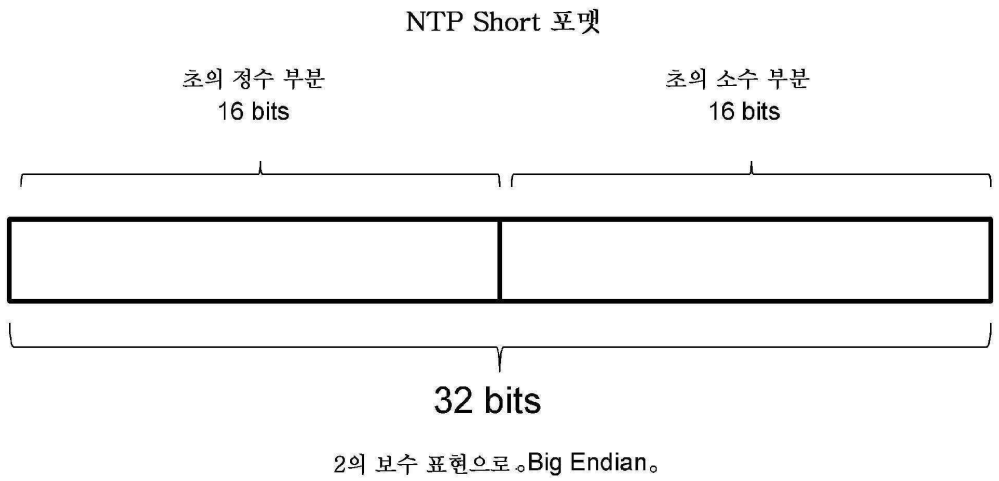
도면9


MMT Packet
MMT Packet Header
MMT Payload Header
MMT Payload
Message
MPU
MPU
MPU
FEC Repair Symbol
Tables
MFU header
MFU
MFU header
MFU
Packet_ID1: video
Packet_ID2: audio
Packet_ID3: data
Packet_id(16), packet_sequence_number(32)
transmission_ timeStamp(32), transmission_priority(3)
FEC_type,
Private_user_data(16)
payload_length (16),페이로드 전체의 사이즈
payload_type(8)
fragmentation_indicator (2) (=0,) fragment_count (8) (=0)
aggregation_info_flag (=0)
RAP_flag (=1) 랜덤 액세스 포인트의 존재
data_offset (8) payload 선두 위치로부터의payload 데이터 최초의 위치 (= payload header의 크기)
numDU (4) (=1) payload 내에 존재하는 data unit (MPU)의 수
DU_offset (16) 각 data unit (MPU's) 의payload data 선두 위치로부터의 offsets
payload_sequence_number(32)
header_extension

도면10

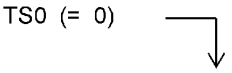


도면11



2의 보수 표현으로 。Big Endian。

도면12



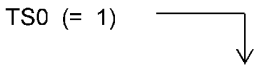
Time	0	1	2	3	4	5	6	7
Dec(AU)	0	1	2	3	4	5	6	7
Display(AU)		0	2	3	1	5	6	4

← Reorder 의 예

AU#	DTS(n)	DLT(n)	PTS(n)
0[I]	0	1	1
1[P]	1	3	4
2[B]	2	0	2
3[B]	3	0	3
4[P]	4	3	7
5[B]	5	0	5
6[B]	6	0	6

(dlt_time / time_tick = 1로 한 경우의 예)

도면13



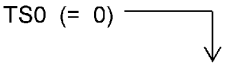
Time	0	1	2	3	4	5	6	7
Dec(AU)	0	1	2	3	4	5	6	7
Display(AU)		0	2	3	1	5	6	4

← Reorder 의 예

AU#	DTS(n)	DLT(n)	PTS(n)
0[I]	0	1	1
1[P]	1	3	4
2[B]	2	0	2
3[B]	3	0	3
4[P]	4	3	7
5[B]	5	0	5
6[B]	6	0	6

(dlt_time / time_tick = 1로 한 경우의 예)

도면14

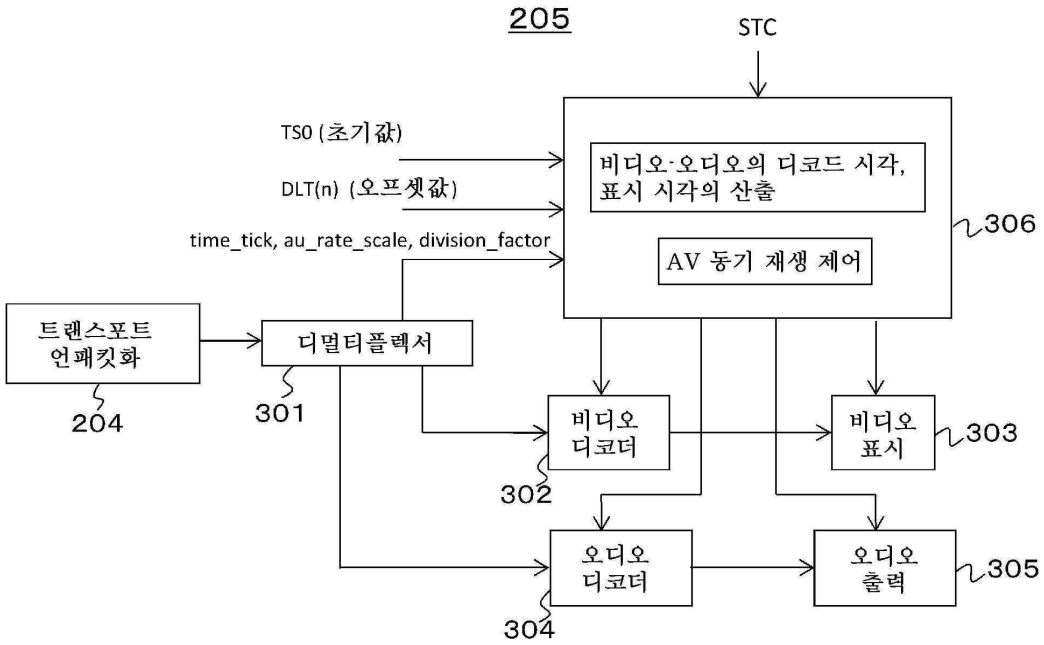


Time	0	1	2	3	4	5	6	7
Dec(AU)	0	1	2	3	4	5	6	7
Display(AU)	0	1	2	3	4	5	6	7

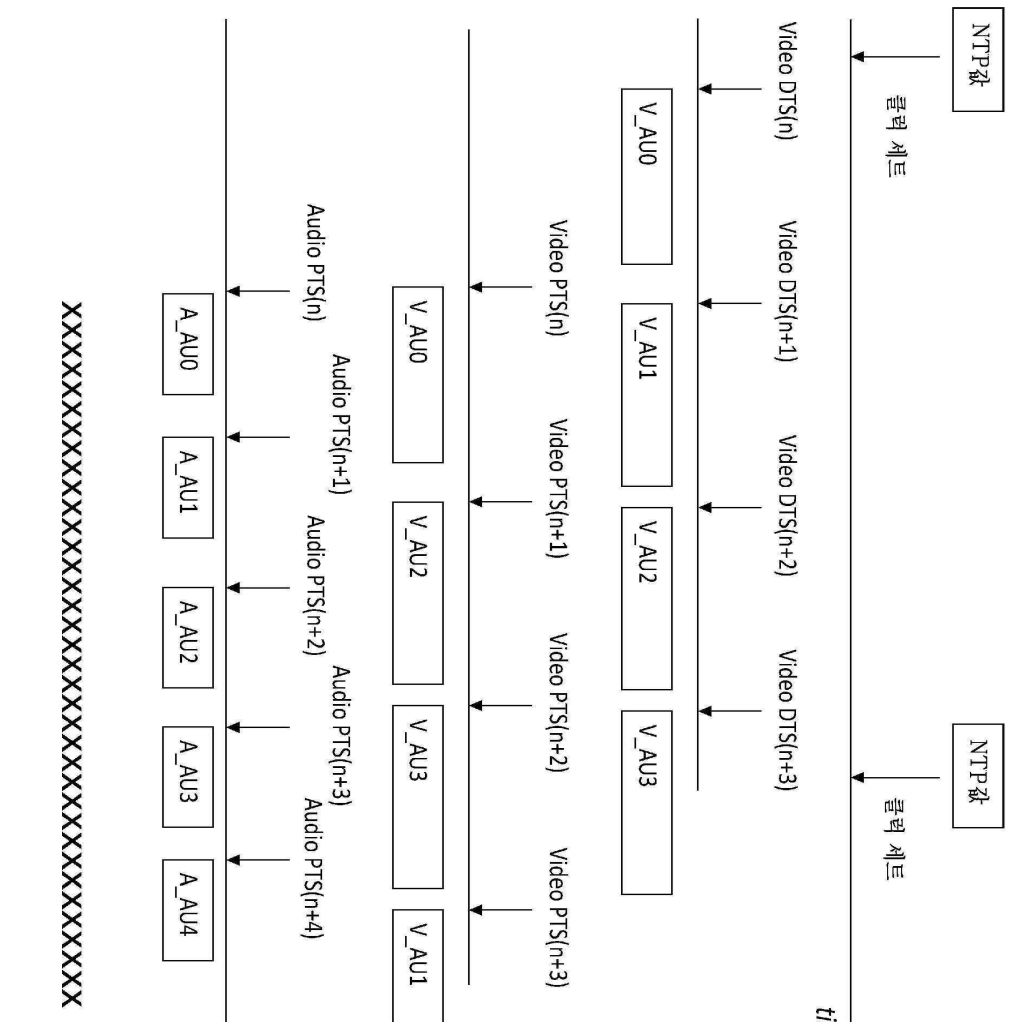
AU#	DTS(n)	DLT(n)	PTS(n)
0	0	0	0
1	1	0	1
2	2	0	2
3	3	0	3
4	4	0	4
5	5	0	5
6	6	0	6

(dlt_time / time_tick = 1로 한 경우의 예)

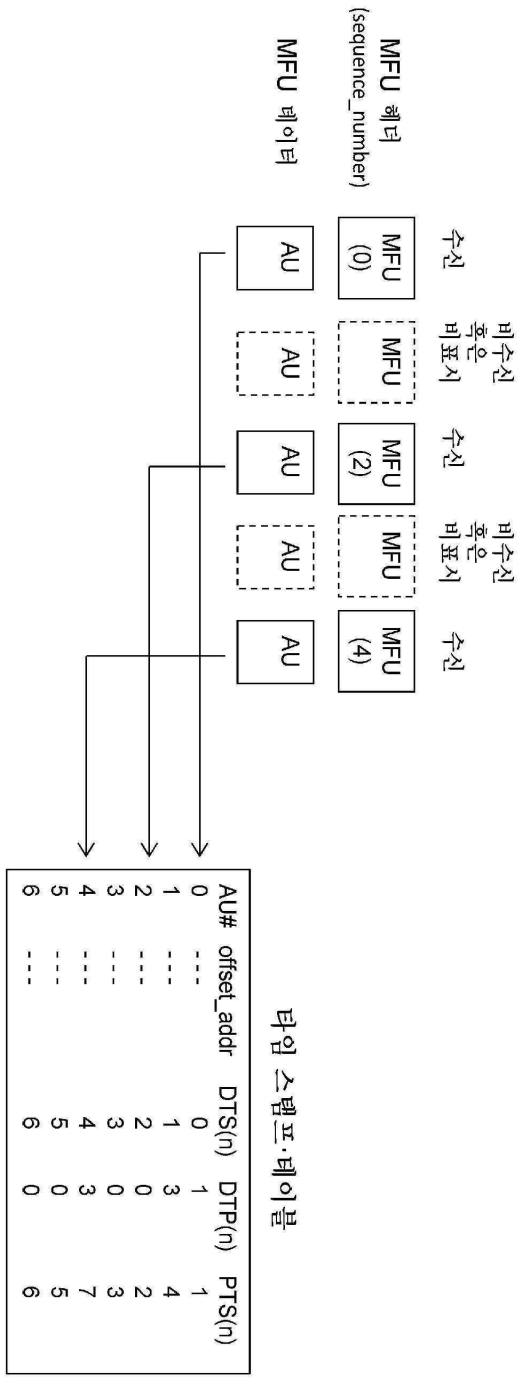
도면15


205
STC
TS0 (초기값)
DLT(n) (오프셋값)
time_tick, au_rate_scale, division_factor
비디오-오디오의 디코드 시각, 표시 시각의 산출
AV 동기 재생 제어
306
트랜스포트 언패킷화
204
디멀티플렉서
301
비디오 디코더
302
비디오 표시
303
오디오 디코더
304
오디오 출력
305

도면16


(a)
시스템 클럭 STC
NTP값
클럭 세트
NTP값
클럭 세트
time
(b)
STC = DTS 의 타이밍
Video 디코드
Video DTS(n)
Video DTS(n+1)
Video DTS(n+2)
Video DTS(n+3)
V_AU0
V_AU1
V_AU2
V_AU3
(c)
STC = PTS 의 타이밍
Video 표시
Video PTS(n)
Video PTS(n+1)
Video PTS(n+2)
Video PTS(n+3)
V_AU0
V_AU2
V_AU3
V_AU1
(d)
STC = PTS (= DTS) 의 타이밍
Audio PTS(n)
Audio PTS(n+1)
Audio PTS(n+2)
Audio PTS(n+3)
Audio PTS(n+4)
A_AU0
A_AU1
A_AU2
A_AU3
A_AU4
(e)
Audio Sample 출력
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

도면17

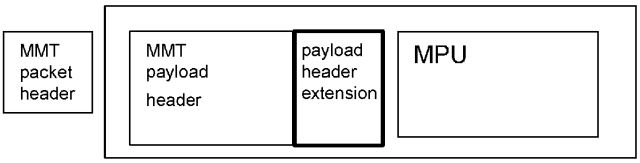


타임 스탬프·테이블

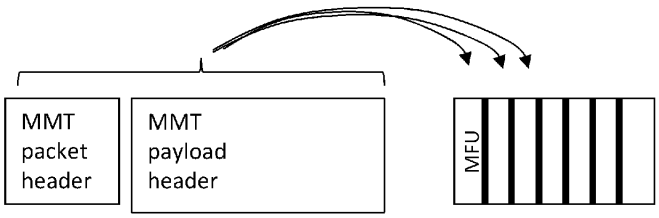
AU#	offset_addr	DTS(n)	DTP(n)	PTS(n)
0	---	0	1	1
1	---	1	3	4
2	---	2	0	2
3	---	3	0	3
4	---	4	3	7
5	---	5	0	5
6	---	6	0	6

도면18

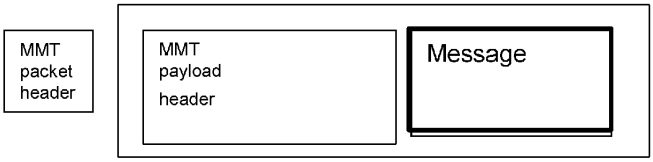
(a) MPU payload header 중에서 시각 정보를 보내는 경우



(b) MPU payload 중에서 프래그먼트화된 AU마다 시각 정보를 보내는 경우



(c) Message 로 시각 정보를 보내는 경우



도면19

(a) MMT 페이로드 헤더에 시각 정보를 넣는 경우의 패킷 구성
랜덤 액세스 포인트
MMT 패킷 헤더
MMT 페이로드 헤더
MPU (Meta)
MMT 헤더
MMT 페이로드 헤더
MFU
MMT 헤더
MMT 페이로드 헤더
MFU
비 랜덤 액세스 포인트
(b) MFU 내에 시각 정보를 넣는 경우의 패킷 구성
랜덤 액세스 포인트
MMT 패킷 헤더
MMT 페이로드 헤더
MPU (Meta)
MMT 헤더
MMT 페이로드 헤더
MFU
MMT 헤더
MMT 페이로드 헤더
MFU
비 랜덤 액세스 포인트
(c) MMT 메시지에 시각 정보를 넣는 경우의 패킷 구성
랜덤 액세스 포인트
MMT 패킷 헤더
MMT 페이로드 헤더
Message
MMT 패킷 헤더
MMT 페이로드 헤더
MPU (Meta)
MMT 패킷 헤더
MMT 페이로드 헤더
MFU
MMT 패킷 헤더
MMT 페이로드 헤더
Message
MMT 패킷 헤더
MMT 페이로드 헤더
MFU
비 랜덤 액세스 포인트

도면20

MMT packet overall

Syntax	No. of Bits	Format
mmt packet( ) {		
mmtp_header()		
mmtp_payload() {		
mmtp_payload_header()		
mmtp_payload_data()		
}		
}		

도면21

MMT packet header

Syntax	No. of Bits	Format
mmtp_header(){		
packet_id	16	uimsbf
packet_sequence_number	32	uimsbf
transmission_ timestamp	32	uimsbf
QoS_classifier_flag	1	bslbf
flow_identifier_flag	1	bslbf
private_usesr_data_flag	1	bslbf
FEC_type	2	bslbf
reserved	3	bslbf
type_of_bitrate	3	bslbf
delay_sensitivity	3	bslbf
reliability_flag	1	bslbf
transmission_priority	3	bslbf
reserved	1	bslbf
flow_label	7	bslbf
extension_flag	1	bslbf
reserved	5	bslbf
private_user_data	16	bslbf
}		

도면22

MMT payload header

Syntax	No. of Bits	Format
mmtp_payload_header(){		
payload_length	16	uimsbf
payload_type	8	bslbf
fragmentation_indicator	2	bslbf
fragment_count_flag	1	bslbf
aggregation_info_flag	1	bslbf
random_access_point_flag	1	bslbf
payload_sequence_number_flag	1	bslbf
header_extension_field_flag	1	bslbf
reserved	1	bslbf
data_offset	8	uimsbf
if(fragment_count_flag == 1)		
fragment_count	8	uimsbf
numDU	8	uimsbf
if (aggregation_info_flag == 1){		
for (i = 0 ; i < numDU ; i++)		
DU_offset	16	uimsbf
}		
if (payload_sequence_number_flag == 1)		
payload_sequence_number	32	uimsbf
if (header_extension_field_flag == 1)		
mmtp_payload_header_extension()		
}		

도면23

MMT payload header extension

Syntax	No. of Bits	Format
mmtp_payload_header_extension() {		
payload_header_extension_type	16	bslbf
payload_header_extension_length	16	uimsbf
asset_type	2	bslbf
delta_sequence_type	1	bslbf
au_count_in_mpu	10	uimsbf
au_rate_scale	3	uimsbf
division_factor	2	bslbf
time_tick	2	bslbf
timestamp_type	1	bslbf
reserved	3	bslbf
timestamp_for_processing	32	uimsbf
if(delta_sequence_type) {		
for (i = 0; i < au_count_in_mpu ; i++)		
delta_variable_length_code	variable	bslbf
trailing_filler()		
} else {		
for (i = 0; i < au_count_in_mpu ; i++)		
delta_fixed_length_code	8	uimsbf
}		
}		

도면24

semantics

payload_header_extension_type	16bits
0x0001	providing timestamp for processing
payload_header_extension_length	16bits
asset_type	2bits
01	video
10	audio
others	reserved
time_tick	2bits
01	90000 Hz 정밀도의 값(LSB)
10	NTP shorttime
others	reserved
au_rate_scale 3bits	
if(asset_type == 01) //video	
000	3750 24Hz
001	3600 25Hz
010	3000 30Hz
011	1800 50Hz
100	1500 60Hz
101	900 100Hz
110	750 120Hz
111	reserved
else if (asset_type == 10) // audio	
000	1920 48KHz * 1024
001	2089.8 44.1KHz * 1024
010	2880 32KHz * 1024
others	reserved

division_factor	2bits
if(asset_type == 01) //video	
00	1
01	1.001
others	reserved
else if (asst_type == 10) // audio	
00	1
01	2
others	reserved
timestamp_type	1bit
1 timestamp is decode_timestamp	
0 timestamp is display_timestamp	
timestamp_for_processing	32bits
if time_tick is “01”,	
timestamp is in 90KHz precision.	
if time_tick is “10”,	
timestamp is NTP short timestamp	
specified in RFC5059.	
au_count_in_mpu	10bits
delta_sequence_type	1bit
1 delta value in variable length coding	
0 delta value in fixed length coding	
delta_fixed_length_code	8bits
delta_variable_length_code	variable

도면25

MMT payload header extension

Syntax	No. of Bits	Format
mmtp_payload_header_extension() {		
payload_header_extension_type	16	bslbf
payload_header_extension_length	16	uimsbf
if (payload_header_extension_type == 0x01)		
presentation_timestamp /* in NTP format*/	32	uimsbf
else if (mfu_payload_header_type == 0x02){		
decoding_timestamp /* in NTP format*/	32	uimsbf
presentation_timestamp /* in NTP format*/	32	uimsbf
}		
else if (mfu_payload_header_type == 0x03)		
presentation_timestamp /* in 90KHz */	32	uimsbf
else if (mfu_payload_header_type == 0x04){		
decoding_timestamp /* in 90KHz */	32	uimsbf
presentation_timestamp /* in 90KHz */	32	uimsbf
}		
else{		
}		
}		

도면26

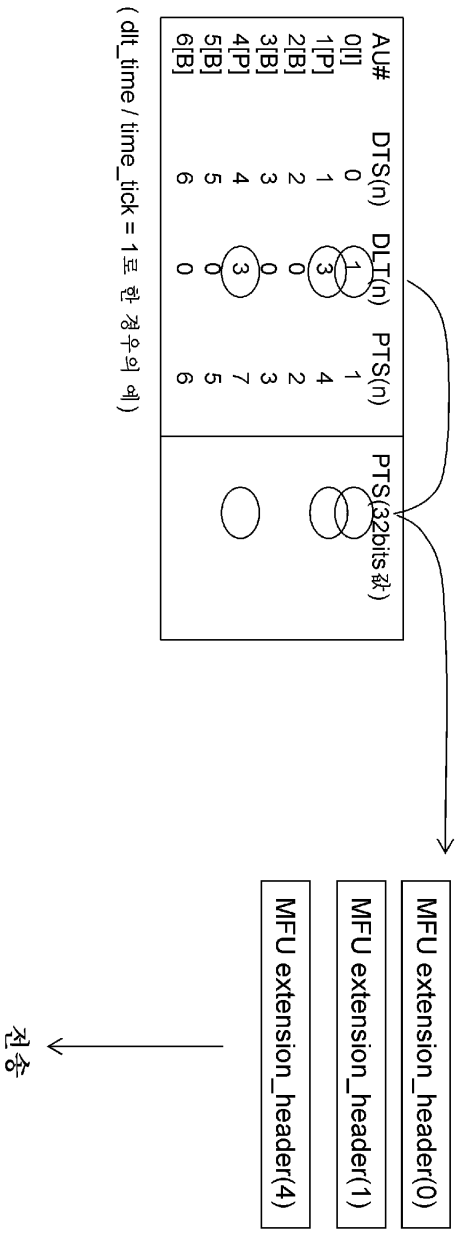
semantics

payload_header_extension_type **(16bits)**
MMT payload header extension 의 타입을 나타낸다

0x01 NTP short time 포맷의 표시 타임 스탬프를 공급한다
0x02 NTP short time 포맷의 표시 타임 스탬프와 디코드 타임 스탬프를 공급한다
0x03 90KHz 정밀도에서의 표시 타임 스탬프를 공급한다
0x04 90KHz 정밀도에서의 표시 타임 스탬프와 디코드 타임 스탬프를 공급한다
others reserved

presentation_timestamp (32bits)
표시 타임 스탬프의 값
decoding_timestamp (32bits)
디코드 타임 스탬프의 값

도면27



도면28

MFU

Syntax	No. of Bits	Format
mfu(){		
mfu_header()		
mfu_media_data()		
}		

도면29

MFU header

Syntax	No. of Bits	Format
mfu_header(){		
sequence_number	32	uimsbf
if (is_timed) {		
trackref_index	8	uimsbf
sample_number	32	uimsbf
priority	8	bslbf
dependency_counter	8	bslbf
offset	16	uimsbf
length	32	uimsbf
multiLayer_flag	1	bslbf
mfu_header_extension_flag	1	bslbf
reserved	6	bslbf
if(multilayer_flag == 1){		
dependency_id	3	bslbf
depth_flag	1	bslbf
reserved	4	bslbf
temporal_id	4	bslbf
quality_id	4	bslbf
priority_id	6	bslbf
view_id	10	bslbf

continued

도면30

Continued from previous page

Syntax	No. of Bits	Format

} else {
layer_id
temporal_id
reserved
}
if (mfu_header_extension_flag == 1) {
mfu_header_extension()
}
} else {
item_ID
}
}
	
	6
	3
	23
	
	16
	
		bslbf
		bslbf
		bslbf
		
		bslbf

도면31

Semantics

sequence_number	MPU 내의 MFU의 시퀀스 번호
trackref_index	미디어 트랙의 번호
sample_number	당 MFU가 속하는 샘플(액세스 유닛)의 moof box 내의 순번
priority	MPU 내의 당 MFU의 우선도
dependency_counter	당 MFU에 디코드가 의존하고 있는 MFU의 수
offset	mdat box로부터의 오프셋
length	MFU의 크기
multiLayer_flag	멀티 레이어 정보의 유무
mfu_header_extension_flag	mfu_extension의 유무
dependency_id	MFU끼리의 다양한 의존 관계를 나타내는 ID군
depth_flag	
temporal_id	
quality_id	
priority_id	
view_id	
layer_id	
Item_ID	파일의 ID

도면32

Syntax	No. of Bits	Format
mfu_header_extension(){		
mfu_header_extenstion_type	8	bslbf
mfu_extension_length	8	uimsbf
asset_type	2	bslbf
au_rate_scale	3	uimsbf
division_factor	2	bslbf
time_tick	2	bslbf
timestamp_type	1	bslbf
reserved	6	bslbf
timestamp_for_processing	32	uimsbf
delta_fixed_length_code	8	uimsbf
}		

도면33

semantics

asset_type	2bits
01	video
10	audio
others	reserved
Time tick	2bits
01	90000 Hz
10	NTP shorttime
others	reserved
AU_rate_scale	3bits
if(asset_type == 01) //video	
000	3750 24Hz
001	3600 25Hz
010	3000 30Hz
011	1800 50Hz
100	1500 60Hz
101	900 100Hz
110	750 120Hz
111	reserved
else if (asset_type == 10) // audio	
000	1920 48KHz * 1024
001	2089.8 44.1KHz * 1024
010	2880 32KHz * 1024
others	reserved
others	reserved

division_factor	2bits
if(asset_type == 01) //video	
00	1
01	1.001
others	reserved
else if (asst_type == 10) // audio	
00	1
01	2
timestamp_type	1bit
1 timestamp is decode_timestamp	
0 timestamp is display_timestamp	
timestamp_for_processing	32bits
if time_tick is "01",	
timestamp is in 90KHz precision.	
if time_tick is "10",	
timestamp is NTP short timestamp	
specified in RFC5059.	
delta_fixed_length_code	8bits

MFU header

Syntax	No. of Bits	Format
mfu_header(){		
sequence_number	32	uimsbf
if (is_timed) {		
trackref_index	8	bslbf
sample_number	32	uimsbf
priority	8	bslbf
dependency_counter	8	bslbf
offset	16	uimsbf
length	32	uimsbf
multiLayer_flag	1	bslbf
timestamp_type	2	bslbf
time_tick	1	bslbf
reserved	4	bslbf
if(multilayer_flag == 1){		
dependency_id	3	bslbf
depth_flag	1	bslbf
reserved	4	bslbf
temporal_id	4	bslbf
quality_id	4	bslbf
priority_id	6	bslbf
view_id	10	bslbf

continued

도면34

도면35

Continued from previous page

Syntax	No. of Bits	Format
} else {		
layer_id	6	bslbf
temporal_id	3	bslbf
reserved	23	bslbf
}		
} else {		
item_ID	16	bslbf
}		
}		

도면36

semantics

Timestamp_type (2bits)	MFU 페이로드의 선두에 타임 스탬프가 공급되는지 여부를 나타낸다.
00	타임 스탬프는 공급되지 않는다. MFU 페이로드의 1st 바이트가 프래그먼트된 MPU 페이로드의 선두 바이트와 일치한다
01	MFU 페이로드의 선두에 표시 타임 스탬프가 공급된다. 표시 타임 스탬프의 직후에 프래그먼트된 MPU 페이로드의 선두 바이트가 공급된다
10	MFU 페이로드의 선두에 디코드 타임 스탬프와 표시 타임 스탬프의 양쪽이 공급된다
11	타임 스탬프의 직후에 프래그먼트된 MPU 페이로드의 선두 바이트가 공급된다 reserved
Time tick (1bit)	타임 스탬프값의 정밀도를 기술한다
0	90000 Hz 정밀도
1	NTP short time 포맷

도면37

Syntax	No. of Bits	Format
mfu_payload(){		
if(timestamp_type == 01)		
presentation_timestamp	32	uimsbf
else if(timestamp_type == 10){		
decoding_timestamp	32	uimsbf
presentation_timestamp	32	uimsbf
}		
else{		
}		
payload_data		
}		

semantics

presentation_timestamp (32bits)
표시 타임 스탬프의 값
decoding_timestamp (32bits)
디코드 타임 스탬프의 값

도면38

Syntax	No. of Bits	Format
Mfu_header(){		
sequence_number	32	uimsbf
if (is_timed) {		
trackref_index	8	bslbf
sample_number	32	uimsbf
priority	8	bslbf
dependency_counter	8	bslbf
offset	16	uimsbf
length	32	uimsbf
multiLayer_flag	1	bslbf
payload_header_flag	1	bslbf
reserved	6	bslbf
if(multilayer_flag == 1){		
dependency_id	3	bslbf
depth_flag	1	bslbf
reserved	4	bslbf
temporal_id	4	bslbf
quality_id	4	bslbf
priority_id	6	bslbf
view_id	10	bslbf

Continued to next page

도면39

Continued from previous page

Syntax	No. of Bits	Format

} else {
layer_id
temporal_id
reserved
}
} else {
item_ID
}
}
	
	6
	3
	23
	
	16
	
		bslbf
		bslbf
		bslbf
		
		bslbf

도면40

semantics

Payload_header_flag (1bit)	MFU 페이로드의 선두에 MFU 페이로드 헤더가 존재하는지 여부를 나타낸다.
0	MFU 페이로드 헤더는 존재하지 않는다. MFU 페이로드의 1st 바이트는 프래그먼트된 MPU 페이로드의 선두 바이트와 일치한다
1	MFU 페이로드 헤더가 존재한다. 프래그먼트된 MPU 페이로드의 선두 바이트가 MFU 페이로드 헤더의 직후에 공급된다

도면41

Syntax	No. of Bits	Format
mfu_payload(){		
if (payload_header_flag == 1){		
mfu_payload_header_type	8	bslbf
mfu_payload_header_length	8	uimsbf
if (mfu_payload_header_type == 0x01)		
presentation_timestamp /* in NTP format*/	32	uimsbf
else if (mfu_payload_header_type == 0x02){		
decoding_timestamp /* in NTP format*/	32	uimsbf
presentation_timestamp /* in NTP format*/	32	uimsbf
}		
else if (mfu_payload_header_type == 0x03)		
presentation_timestamp /* in 90KHz */	32	uimsbf
else if (mfu_payload_header_type == 0x04){		
decoding_timestamp /* in 90KHz */	32	uimsbf
presentation_timestamp /* in 90KHz */	32	uimsbf
}		
else{		
}		
}		
payload_data		
}		

도면42

semantics

mfu_payload_header_type (8bits)
MFU 페이로드 헤더의 타입을 나타낸다.

0x01	NTP short time 포맷의 표시 타임 스탬프를 공급한다
0x02	NTP short time 포맷의 표시 타임 스탬프와 디코드 타임 스탬프를 공급한다
0x03	90KHz 정밀도에서의 표시 타임 스탬프를 공급한다
0x04	90KHz 정밀도에서의 표시 타임 스탬프와 디코드 타임 스탬프를 공급한다
others	reserved

mfu_payload_header_length (8bits)
본 엘리먼트 이후의 mfu 페이로드 헤더의 사이즈를 byte 수로 나타낸다.

presentation_timestamp (32bits)
표시 타임 스탬프의 값

decoding_timestamp (32bits)
디코드 타임 스탬프의 값

도면43

Timestamp Message

Syntax	No. of Bits	Format
timestamp_message() {		
message_id	16	
version	8	
length	16	
extension {		
timestamp_table()		
}		
}		

도면44

Syntax	No. of Bits	Format
Timestamp_table() {		
table_id	8	uimsbf
version	8	uimsbf
length	16	uimsbf
packet_id	16	uimsbf
asset_type	2	bslbf
delta_sequence_type	1	bslbf
au_count_in_mpu	10	uimsbf
au_rate_scale	3	uimsbf
division_factor	2	bslbf
time_tick	2	bslbf
timestamp_type	1	bslbf
reserved	3	bslbf
timestamp_for_processing	32	uimsbf
if(delta_sequence_type) {		
for (i = 0; i < au_count_in_mpu ; i++)		
delta_variable_length_code	variable	
trailing_filler()		
} else{		
for (i = 0; i < au_count_in_mpu ; i++)		
delta_fixed_length_code	8	uimsbf
}		
}		

도면45

Package Access Message

Syntax	No. of Bits	Format
Package_Access_message() {		
message_id	16	uimsbf
version	8	uimsbf
length	32	uimsbf
extension {		
number_of_tables ; N1	8	uimsbf
for (i=0; I < N1; i++) {		
table_id	8	uimsbf
table_version	8	uimsbf
table_length	16	uimsbf
}		
message_payload {		
for (i=o; I < N1; i++){		
table()		
}		
}		
}		

도면46

Time	0	1	2	3	4	5	6	7
Dec(AU)	D(0)	D(1)	D(2)	D(3)	D(4)	D(5)	D(6)	D(7)
Display(AU)		R(0)	R(2)	R(3)	R(1)	R(5)	R(6)	R(4)
DLT(AU)	1	3	0	0	3	0	0	3

도면47

가변 길이 부호 테이블

부호화 입력값	가변 길이 부호	부호어 길이 (bit)
0 ::	1	1
1 ::	010	3
2 ::	011	3
3 ::	00100	5
4 ::	00101	5
5 ::	00110	5
6 ::	00111	5
7 ::	0001000	7
8 ::	0001001	7
9 ::	0001010	7
10 ::	0001011	7
11 ::	0001100	7
12 ::	0001101	7
13 ::	0001110	7
14 ::	0001111	7
15 ::	000010000	9

“0”의 연속하는 수가 부호 길이를 나타낸다.
“0”이 없으면 부호 길이는 1. “0”이 1개이면 부호 길이는 3.
“0”이 2개이면, 부호 길이는 5. “0”이 3개이면 부호 길이는 7.

도면48

부호 길이의 견적

GOP length (M)	총 bit 길이	4bit 고정의 언어 길이
M = 3 (I P B B) I B B P B_probability: 2/4	3 + 5 + 2 = 10 (3+7) bits	16 bits
M = 4 (I P B B B) I B B B P B_probability: 3/5	3 + 5 + 3 = 11 (3+8) bits	20bits
M = 5 (I P B B B B) I B B B B P B_probability: 4/6	3 + 5 + 4 = 12 (3+9) bits	24bits
M = 6 (I P B B B B B) I B B B B B P B_probability: 5/7	3 + 5 + 5 = 13 (3+10) bits	28bits
M = 7 (I P B B B B B B) I B B B B B B P B_probability: 6/8	3 + 7 + 6 = 16 (3+13) bits	32bits
M = 8 (I P B B B B B B B) I B B B B B B B P B_probability: 7/9	3 + 7 + 7 = 17 (3+14) bits	36bits
M = 15 (I P B B B B B B B B B B B B B B) I B B B B B B B B B B B B B B P B_probability: 14/16	3 + 9 + 14 = 26 (3+23) bits	64bits