



(19) 대한민국특허청(KR)

(12) 등록특허공보(B1)

(45) 공고일자 2020년08월18일

(11) 등록번호 10-2145653

(24) 등록일자 2020년08월11일

(51) 국제특허분류(Int. Cl.)

H04L 29/06 (2006.01) H04N 21/2343 (2011.01)

H04N 21/262 (2011.01) H04N 21/442 (2011.01)

H04N 21/81 (2011.01) H04N 21/845 (2011.01)

(52) CPC특허분류

H04L 65/602 (2013.01)

H04L 65/4069 (2013.01)

(21) 출원번호 10-2016-7027894

(22) 출원일자(국제) 2015년03월24일

심사청구일자 2020년03월20일

(85) 번역문제출일자 2016년10월07일

(65) 공개번호 10-2016-0136343

(43) 공개일자 2016년11월29일

(86) 국제출원번호 PCT/US2015/022264

(87) 국제공개번호 WO 2015/148519

국제공개일자 2015년10월01일

(30) 우선권주장

61/969,777 2014년03월24일 미국(US)

14/665,252 2015년03월23일 미국(US)

(56) 선행기술조사문헌

US20130185398 A1

US20100107191 A1

(73) 특허권자

퀄컴 인코포레이티드

미국 92121-1714 캘리포니아주 샌 디에고 모어하우스 드라이브 5775

(72) 발명자

슈토크하머 토마스

미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

(74) 대리인

특허법인코리아나

전체 청구항 수 : 총 10 항

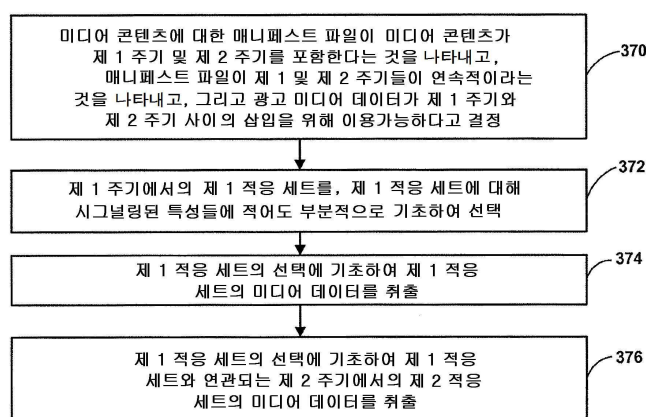
심사관 : 박보미

(54) 발명의 명칭 연속적인 멀티-주기 콘텐츠의 프로세싱

**(57) 요약**

하나의 예에서, 디바이스는 하나 이상의 프로세서들을 포함하고, 그 하나 이상의 프로세서들은 미디어 콘텐츠에 대한 매니페스트 파일이 미디어 콘텐츠가 제 1 주기 및 제 2 주기를 포함한다는 것을 나타내고, 매니페스트 파일이 제 1 및 제 2 주기들이 연속적이라는 것을 나타내고, 그리고 광고 미디어 데이터가 제 1 주기와 제 2 주기 사이의 삽입을 위해 이용가능하다고 결정

(뒷면에 계속)

**대표도** - 도8

사이의 삽입을 위해 이용가능하다고 결정하고, 제 1 주기에서의 제 1 적응 세트를, 제 1 적응 세트에 대해 시그널링된 특성들에 적어도 부분적으로 기초하여 선택하고, 제 1 적응 세트의 선택에 기초하여 제 1 적응 세트의 미디어 데이터를 추출하고, 광고 미디어 데이터를 추출하고, 그리고 제 1 적응 세트의 선택에 기초하여 제 1 적응 세트와 연관되는 매니페스트 파일이 나타내는 제 2 주기에서의 제 2 적응 세트의 미디어 데이터를 추출하고 추출된 미디어 데이터를 메모리에 저장하도록 구성된다.

(52) CPC특허분류

*H04N 21/23439* (2013.01)

*H04N 21/26258* (2013.01)

*H04N 21/44209* (2013.01)

*H04N 21/812* (2013.01)

*H04N 21/8456* (2013.01)

## 명세서

### 청구범위

#### 청구항 1

미디어 데이터를 추출하는 방법으로서,

미디어 콘텐츠에 대한 매니페스트 파일로부터, 상기 미디어 콘텐츠가 제 1 주기 및 제 2 주기를 포함하고, 상기 제 1 주기 및 상기 제 2 주기가 연속적이고, 그리고 광고 미디어 데이터가 상기 제 1 주기와 상기 제 2 주기 사이의 삽입을 위해 이용가능하다고 결정하는 단계 (370);

상기 제 1 주기에서의 복수의 적응 세트들 중에서 상기 제 1 주기에서의 제 1 적응 세트를, 상기 제 1 적응 세트에 대해 상기 매니페스트 파일에서 시그널링된 렌더링 특성들에 적어도 부분적으로 기초하여 선택하는 단계 (372)로서, 각각의 적응 세트는 미디어 콘텐츠의 상기 제 1 주기의 개별의 복수의 표현들을 포함하고, 상기 개별의 복수의 표현들의 각각은 동일한 렌더링 특성들을 공유하지만 상이한 비트레이트를 갖는, 상기 제 1 적응 세트를 선택하는 단계;

상기 제 1 적응 세트의 상기 선택에 기초하여 상기 제 1 적응 세트의 미디어 데이터를 추출하는 단계 (374);

상기 광고 미디어 데이터가 상기 제 1 주기와 상기 제 2 주기 사이의 삽입을 위해 이용가능하다는 결정 및 상기 제 1 적응 세트의 상기 선택에 기초하여, 상기 제 2 주기에서의 제 2 적응 세트가 상기 제 1 적응 세트와 연관된다고 결정하는 단계로서, 상기 제 2 적응 세트는 상기 제 2 주기에서의 복수의 적응 세트들 중 하나의 적응 세트이고, 각각의 적응 세트는 미디어 콘텐츠의 상기 제 2 주기의 개별의 복수의 표현들을 포함하고, 상기 개별의 복수의 표현들의 각각은 동일한 렌더링 특성들을 공유하지만 상이한 비트레이트를 갖고, 상기 제 2 적응 세트가 상기 제 1 적응 세트와 연관된다고 결정하는 단계는, 상기 매니페스트 파일이 상기 제 1 적응 세트 및 상기 제 2 적응 세트의 애셋 식별자들에 대해 동일한 값을 시그널링하고 상기 매니페스트 파일이 상기 제 1 적응 세트 및 상기 제 2 적응 세트가 개별의 "@id" 속성들에 대해 동일한 값을 갖는다는 것을 나타낸다고 결정하는 단계를 포함하는, 상기 제 2 적응 세트가 상기 제 1 적응 세트와 연관된다고 결정하는 단계;

상기 제 2 적응 세트의 미디어 데이터를 추출하는 단계 (376); 및

상기 제 1 적응 세트에서의 표현들의 @id 속성이 상기 제 2 적응 세트에서의 표현들의 @id 속성과 동일한 것에 기초하여, 상기 제 1 적응 세트의 표현의 초기화 세그먼트를 상기 제 2 적응 세트의 대응하는 표현의 초기화 세그먼트로서 이용하는 단계

를 포함하는, 미디어 데이터를 추출하는 방법.

#### 청구항 2

제 1 항에 있어서,

상기 매니페스트 파일은 미디어 프리젠테이션 디스크립션 (MPD) 을 포함하는, 미디어 데이터를 추출하는 방법.

#### 청구항 3

제 1 항에 있어서,

상기 매니페스트 파일로부터, 프리젠테이션 지속기간을 특정하는 @presentationTimeDuration 선택스 엘리먼트에 대한 값을 시그널링하는 데이터 또는 상기 제 1 적응 세트에 대해 "urn:mpeg:dash:period-continuity:2014" 로 설정된 @scheme\_id\_URI 엘리먼트를 가진 보충적 디스크립터를 시그널링하는 데이터 중 적어도 하나를 이용하여 상기 제 1 적응 세트의 표현에 대한 상기 프리젠테이션 지속기간을 결정하는 단계를 더 포함하는, 미디어 데이터를 추출하는 방법.

#### 청구항 4

제 3 항에 있어서,

상기 제 1 적응 세트는 상기 매니페스트 파일에서 기술되는 주기의 "@id" 엘리먼트의 값, 적응 세트 식별자를 표현하는 "AdaptationSet@id" 엘리먼트에 대한 값, 및 상기 제 1 적응 세트에 대한 "@presentationTimeOffset"의 값에 매칭하는 패킷 식별자 (PID)의 "@value" 엘리먼트를 더 포함하는, 미디어 데이터를 추출하는 방법.

#### 청구항 5

제 1 항에 있어서,

상기 제 2 적응 세트가 상기 제 1 적응 세트와 공통의 하나 이상의 특성들을 갖는다고 결정하는 단계를 더 포함하고, 공통의 상기 특성들은 언어, 미디어 컴포넌트 타입, 픽처 에스펙트비, 역할 속성, 액세스가능성 속성, 또는 뷰포인트 속성 중 하나 이상을 포함하는, 미디어 데이터를 추출하는 방법.

#### 청구항 6

제 1 항에 있어서,

상기 제 1 적응 세트 및 상기 제 2 적응 세트가 연관되고;

"@presentationTimeOffset"이 상기 매니페스트 파일에 존재하거나 또는 양자의 적응 세트들에서의 모든 표현들에 대해 0 으로서 추론될 수 있고;

상기 매니페스트 파일이 상기 제 1 적응 세트 및 상기 제 2 적응 세트 중 적어도 하나에 대해, "@presentationTimeOffset"의 값이 모든 표현들에 대해 동일하다는 것을 나타내고; 그리고

상기 "@presentationTimeOffset"의 값과 상기 제 1 적응 세트에서의 모든 표현들의 프리젠테이션 지속기간의 합이 상기 제 2 적응 세트의 "@presentationTimeOffset"의 값과 동일한

경우, 상기 제 2 적응 세트가 상기 제 1 적응 세트와 주기-연속적이라고 결정하는 단계를 더 포함하는, 미디어 데이터를 추출하는 방법.

#### 청구항 7

미디어 데이터를 추출하기 위한 디바이스로서,

미디어 데이터를 저장하도록 구성된 메모리; 및

하나 이상의 하드웨어-기반 프로세서들

을 포함하고, 상기 하나 이상의 하드웨어-기반 프로세서들은 :

미디어 콘텐츠에 대한 매니페스트 파일로부터, 상기 미디어 콘텐츠가 제 1 주기 및 제 2 주기를 포함하고, 상기 제 1 주기 및 상기 제 2 주기가 연속적이고, 그리고 광고 미디어 데이터가 상기 제 1 주기와 상기 제 2 주기 사이의 삽입을 위해 이용가능하다고 결정하고 (370),

상기 제 1 주기에서의 복수의 적응 세트들 중에서 상기 제 1 주기에서의 제 1 적응 세트를, 상기 제 1 적응 세트에 대해 상기 매니페스트 파일에서 시그널링된 렌더링 특성들에 적어도 부분적으로 기초하여 선택하는 것으로서 (372), 각각의 적응 세트는 미디어 콘텐츠의 상기 제 1 주기의 개별의 복수의 표현들을 포함하고, 상기 개별의 복수의 표현들의 각각은 동일한 렌더링 특성들을 공유하지만 상이한 비트레이트를 갖는, 상기 제 1 적응 세트를 선택하고,

상기 제 1 적응 세트의 상기 선택에 기초하여 상기 제 1 적응 세트의 미디어 데이터를 추출하고 (374),

상기 광고 미디어 데이터가 상기 제 1 주기와 상기 제 2 주기의 삽입을 위해 이용가능하다는 결정 및 상기 제 1 적응 세트의 상기 선택에 기초하여, 상기 제 2 주기에서의 제 2 적응 세트가 상기 제 1 적응 세트와 연관된다고 결정하는 것으로서, 상기 제 2 적응 세트는 상기 제 2 주기에서의 복수의 적응 세트들 중 하나의 적응 세트이고, 각각의 적응 세트는 미디어 콘텐츠의 상기 제 2 주기의 개별의 복수의 표현들을 포함하고, 상기 개별의 복수의 표현들의 각각은 동일한 렌더링 특성들을 공유하지만 상이한 비트레이트를 갖고, 상기 제 2 적응 세트가 상기 제 1 적응 세트와 연관된다고 결정하는 것은, 상기 매니페스트 파일이 상기 제 1 적응 세트 및 상기 제 2 적응 세트의 애셋 식별자들에 대해 동일한 값을 시그널링하고 상기 매니페스트 파일이 상기 제 1 적응 세트 및 상기 제 2 적응 세트가 개별의 "@id" 속성들에 대해 동일한 값을 갖는다는 것을 나타낸다고 결정하는 것을 포함하는, 상기 제 2 적응 세트가 상기 제 1 적응 세트와 연관된다고 결정하고;

상기 제 2 적응 세트의 미디어 데이터를 추출하고 (376) 추출된 상기 미디어 데이터를 상기 메모리에 저장하고; 그리고

상기 제 1 적응 세트에서의 표현들의 @id 속성이 상기 제 2 적응 세트에서의 표현들의 @id 속성과 동일한 것에 기초하여, 상기 제 1 적응 세트의 표현의 초기화 세그먼트를 상기 제 2 적응 세트의 대응하는 표현의 초기화 세그먼트로서 이용하도록

구성되는, 미디어 데이터를 추출하기 위한 디바이스.

#### 청구항 8

제 7 항에 있어서,

상기 매니페스트 파일은 미디어 프리젠테이션 디스크립션 (MPD) 을 포함하는, 미디어 데이터를 추출하기 위한 디바이스.

#### 청구항 9

제 7 항에 있어서,

상기 하나 이상의 하드웨어-기반 프로세서들은 상기 매니페스트 파일로부터, 상기 제 2 적응 세트가 상기 제 1 적응 세트와 공통의 하나 이상의 특성들을 갖는다고 결정하도록 구성되고, 공통의 상기 특성들은 언어, 미디어 컴포넌트 타입, 픽처 에스펙트비, 역할 속성, 액세스가능성 속성, 또는 뷰포인트 속성 중 하나 이상을 포함하는, 미디어 데이터를 추출하기 위한 디바이스.

#### 청구항 10

제 7 항에 있어서,

상기 하나 이상의 하드웨어-기반 프로세서들은 :

상기 제 1 적응 세트 및 상기 제 2 적응 세트가 연관되고;

"@presentationTimeOffset" 이 상기 매니페스트 파일에 존재하거나 또는 양자의 적응 세트들에서의 모든 표현들에 대해 0 으로서 추론될 수 있고;

상기 매니페스트 파일이 상기 제 1 적응 세트 및 상기 제 2 적응 세트 중 적어도 하나에 대해, "@presentationTimeOffset" 의 값이 모든 표현들에 대해 동일하다는 것을 나타내고; 그리고

상기 "@presentationTimeOffset" 의 값과 상기 제 1 적응 세트에서의 모든 표현들의 프리젠테이션 지속기간의 합이 상기 제 2 적응 세트의 "@presentationTimeOffset" 의 값과 동일한

경우, 상기 제 2 적응 세트가 상기 제 1 적응 세트와 주기-연속적이라고 결정하도록 구성되는, 미디어 데이터를 추출하기 위한 디바이스.

#### 청구항 11

삭제

#### 청구항 12

삭제

#### 청구항 13

삭제

#### 청구항 14

삭제

#### 청구항 15

삭제

청구항 16

삭제

청구항 17

삭제

청구항 18

삭제

청구항 19

삭제

청구항 20

삭제

청구항 21

삭제

청구항 22

삭제

청구항 23

삭제

청구항 24

삭제

청구항 25

삭제

청구항 26

삭제

청구항 27

삭제

청구항 28

삭제

청구항 29

삭제

청구항 30

삭제

**발명의 설명**

## 기술 분야

[0001] 본 출원은 2014년 3월 24일자로 출원된 미국 가출원 제61/969,777호의 이익을 주장하고, 그 전체 내용은 여기에 참조에 의해 통합된다.

[0002] 기술 분야

[0003] 본 개시는 미디어 데이터의 전송, 예를 들어, 네트워크를 통한 미디어 데이터의 스트리밍에 관한 것이다.

## 배경 기술

[0004] 디지털 비디오 능력들은 디지털 텔레비전들, 디지털 직접 브로드캐스트 시스템들, 무선 브로드캐스트 시스템들, 개인 휴대 정보 단말기들 (PDA들), 랩톱 또는 데스크톱 컴퓨터들, 디지털 카메라들, 디지털 레코딩 디바이스들, 디지털 미디어 플레이어들, 비디오 게이밍 디바이스들, 비디오 게임 콘솔들, 셀룰러 또는 위성 라디오 전화기들, 비디오 화상회의 디바이스들 등을 포함하는, 광범위한 디바이스들에 통합될 수 있다. 디지털 비디오 디바이스들은 디지털 비디오 정보를 보다 효율적으로 송신 및 수신하기 위해, MPEG-2, MPEG-4, ITU-T H.263 또는 ITU-T H.264/MPEG-4, 파트 10, AVC (Advanced Video Coding), ITU-T H.265/MPEG-H 파트 2 에 의해 정의된 표준들, 및 이러한 표준들의 확장들에서 설명된 것들과 같은 비디오 압축 기법들을 구현한다.

[0005] 비디오 압축 기법들은 비디오 시퀀스들에 내재하는 리던던시를 감소 또는 제거하기 위해 공간 예측 및/또는 시간 예측을 수행한다. 블록-기반 비디오 코딩에 대해, 비디오 프레임 또는 슬라이스는 매크로블록들로 파티셔닝될 수도 있다. 각각의 매크로블록은 추가 파티셔닝될 수 있다. 인트라-코딩된 (I) 프레임 또는 슬라이스에서의 매크로블록들은 이웃하는 매크로블록들에 대하여 공간 예측을 이용하여 인코딩된다. 인터-코딩된 (P 또는 B) 프레임 또는 슬라이스에서의 매크로블록들은 동일한 프레임 또는 슬라이스에서의 이웃하는 매크로블록들에 대하여 공간 예측을 이용하거나 또는 다른 레퍼런스 프레임들에 대하여 시간 예측을 이용할 수도 있다.

[0006] 비디오 데이터 (및/또는 다른 미디어 데이터, 이를 테면 오디오 및/또는 타임드 (timed) 텍스트 데이터) 가 인코딩된 후, 미디어 데이터는 송신 또는 저장을 위해 패킷화될 수도 있다. 패킷화된 미디어 데이터는 유닉캐스트 프로토콜, 이를 테면 하이퍼텍스트 전송 프로토콜 (HTTP), 또는 브로드캐스트 또는 멀티캐스트 프로토콜, 이를 테면 향상된 멀티미디어 브로드캐스트 멀티캐스트 서비스 (Enhanced Multimedia Broadcast Multicast Service; eMBMS) 를 이용하여 전송될 수도 있다.

## 발명의 내용

### 해결하려는 과제

### 과제의 해결 수단

[0007] 일반적으로, 본 개시는 플레이아웃 및 스트리밍 경우에서, 기존 세트의 미디어 데이터로의 추가적인 미디어 데이터 (예를 들어, 광고 미디어 데이터) 가 주기 경계를 시그널링함으로써, 예를 들어, 콘텐츠를 이용하는 전송 기에서의 판정에 의해 또는 콘텐츠를 수신하는 수신기에 의해 스플라이싱 및 추가될 수도 있지만, 동시에 어떤 추가적인 미디어 콘텐츠도 추가되지 않을 때에는, 수신기가 수신기에서의 콘텐츠의 디코딩 및 플레이아웃이 주기 경계에 걸쳐 연속적이라는, 즉 주기 연속성을 유지하고 있다는 충분한 정보를 갖도록 적응 스트리밍을 위해 이용되는 미디어 콘텐츠를 준비하기 위한 기법들을 설명한다. 미디어 콘텐츠는 별개의 주기들로 분리될 수도 있고, 여기서 각각의 주기는 각각이 하나 이상의 표현 (representation) 들을 포함하는 하나 이상의 적응 세트들을 포함한다. 일반적으로, 동일한 적응 세트 내의 표현들은 예를 들어, 네트워크 대역폭을 변화시키는 것에 적응하기 위해, 클라이언트 디바이스가 표현들 사이에서 스위칭할 수도 있도록, 서로에 대한 대안들인 것으로 고려될 수도 있다. 주기 내의 적응 세트들은 이러한 스위칭가능한 표현들, 즉 공통 코딩 및 렌더링 특성들을 공유하지만 비트레이트들에 있어서 상이한 표현들을 포함할 수도 있다.

[0008] 본 개시는 추가적인 미디어 데이터가 연속적인 주기들 사이에 삽입될 수 있도록, 하나의 주기에서 제공된 콘텐츠를 2 개의 연속적인 주기들로 스플리팅하기 위한 기법들을 설명한다. 이 방식으로, 연속적인 주기들의 표현들에 대한 특성들이 추가된 미디어 데이터에 후속하여 재시그널링될 필요는 없다. 그 대신, 하나의 주기

에서의 표현들의 데이터에 대한 동일한 시그널링된 특성들은 (추가적인 미디어 데이터가 연속적인 주기들 사이에 삽입될 수도 있는) 다음의 연속적인 주기에서의 표현들의 데이터에 대해 이용될 수도 있다. 이 방식으로, 미디어 콘텐츠의 미디어 데이터 및 추가된 미디어 데이터는 연속적인 방식으로 제시될 수도 있다.

예를 들어, 클라이언트 디바이스는 제 1 주기의 적응 세트의 선택에 기초하여 (예를 들어, 제 2 주기의 적응 세트의 특성들을 재분석하지 않고 및 클라이언트에서의 미디어 파싱 및 디코딩 엔티티를 재초기화하지 않고), 제 1 주기의 적응 세트에 대응하는 제 2 주기의 적응 세트를 이용할 수도 있다.

[0009] 하나의 예에서, 미디어 데이터를 추출 (retrieving) 하는 방법은 미디어 콘텐츠에 대한 매니페스트 파일 (manifest file) 이 미디어 콘텐츠가 제 1 주기 및 제 2 주기를 포함한다는 것을 나타내고, 매니페스트 파일이 제 1 주기 및 제 2 주기가 연속적이라는 것을 나타내고, 그리고 광고 미디어 데이터가 제 1 주기와 제 2 주기 사이의 삽입을 위해 이용가능하다고 결정하는 단계, 제 1 주기에서의 제 1 적응 세트를, 제 1 적응 세트에 대해 시그널링된 특성들에 적어도 부분적으로 기초하여 선택하는 단계, 제 1 적응 세트의 선택에 기초하여 제 1 적응 세트의 미디어 데이터를 추출하는 단계, 광고 미디어 데이터를 추출하는 단계; 및 제 1 적응 세트의 선택에 기초하여 제 1 적응 세트와 연관되는 제 2 주기에서의 제 2 적응 세트의 미디어 데이터를 추출하는 단계를 포함한다.

[0010] 다른 예에서, 미디어 데이터를 추출하기 위한 디바이스는 미디어 데이터를 저장하도록 구성된 메모리, 및 하나 이상의 하드웨어-기반 프로세서들을 포함하고, 그 하나 이상의 하드웨어-기반 프로세서들은 미디어 콘텐츠에 대한 매니페스트 파일이 미디어 콘텐츠가 제 1 주기 및 제 2 주기를 포함한다는 것을 나타내고, 매니페스트 파일이 제 1 주기 및 제 2 주기가 연속적이라는 것을 나타내고, 그리고 광고 미디어 데이터가 제 1 주기와 제 2 주기 사이의 삽입을 위해 이용가능하다고 결정하고, 제 1 주기에서의 제 1 적응 세트를, 제 1 적응 세트에 대해 시그널링된 특성들에 적어도 부분적으로 기초하여 선택하고, 제 1 적응 세트의 선택에 기초하여 제 1 적응 세트의 미디어 데이터를 추출하고, 광고 미디어 데이터를 추출하고, 그리고 제 1 적응 세트의 선택에 기초하여 제 1 적응 세트와 연관되는 매니페스트 파일이 나타내는 제 2 주기에서의 제 2 적응 세트의 미디어 데이터를 추출하고 추출된 미디어 데이터를 메모리에 저장하도록 구성된다.

[0011] 다른 예에서, 미디어 데이터를 전송하는 방법은 미디어 서버에 의해, 제 1 주기 및 제 2 주기를 포함하는 복수의 주기들로 미디어 데이터의 메인 콘텐츠를 스플리팅하는 단계로서, 제 1 주기 및 제 2 주기는 시간적으로 순차적인, 상기 미디어 데이터의 메인 콘텐츠를 스플리팅하는 단계; 미디어 서버에 의해, 제 1 주기와 제 2 주기 사이에 세컨더리 미디어 콘텐츠를 삽입하는 단계, 미디어 서버에 의해, 복수의 주기들의 각각에 대한 시간 표시자를 시그널링하는 단계로서, 시간 표시자는 복수의 주기들의 각각에 대한 하나 이상의 시간 특성들을 정의하는, 상기 시간 표시자를 시그널링하는 단계, 및 미디어 서버에 의해, 복수의 주기들, 세컨더리 미디어 콘텐츠, 및 시간 표시자들을 포함하는 미디어 스트림을 추출하기 위한 DASH (dynamic adaptive streaming over HTTP) 클라이언트로부터의 요청을 수신하는 단계를 포함한다.

[0012] 다른 예에서, 미디어 데이터를 전송하기 위한 디바이스는 미디어 데이터를 저장하도록 구성된 메모리, 및 하나 이상의 하드웨어-기반 프로세서들을 포함하고, 그 하나 이상의 하드웨어-기반 프로세서들은 시간적으로 순차적인 제 1 주기 및 제 2 주기를 포함하는 복수의 주기들로 미디어 데이터의 메인 콘텐츠를 스플리팅하고, 제 1 주기와 제 2 주기 사이에 세컨더리 미디어 콘텐츠를 삽입하고, 복수의 주기들의 각각에 대한 시간 표시자를 시그널링하는 것으로서, 시간 표시자는 복수의 주기들의 각각에 대한 하나 이상의 시간 특성들을 정의하는, 상기 시간 표시자를 시그널링하고, 그리고 복수의 주기들, 세컨더리 미디어 콘텐츠, 및 시간 표시자들을 포함하는 미디어 스트림을 추출하기 위한 DASH (dynamic adaptive streaming over HTTP) 클라이언트로부터의 요청을 수신하도록 구성된다.

[0013] 하나 이상의 예들의 상세들은 첨부한 도면들 및 이하의 설명에 기재된다. 다른 특징들, 목적들, 및 이점들은 설명 및 도면들로부터, 및 청구항들로부터 명백할 것이다.

## 도면의 간단한 설명

[0014] 도 1 은 네트워크를 통해 미디어 데이터를 스트리밍하기 위한 기법들을 구현하는 일 예의 시스템을 예시하는 블록 다이어그램이다.

도 2 는 예의 멀티미디어 콘텐츠의 엘리먼트들을 예시하는 개념적 다이어그램이다.

도 3 은 본 개시의 기법들을 구현할 수도 있는 다른 예의 시스템을 예시하는 블록 다이어그램이다.



도 4 내지 도 6 은 본 개시의 다양한 양태들에 따른 하나 이상의 기법들을 예시하는 플로우 다이어그램들이다.

도 7 은 미디어 서버에 의해 본 개시의 소정의 기법들을 수행하는 일 예의 방법을 예시하는 플로우차트이다.

도 8 은 클라이언트 디바이스에 의해 본 개시의 소정의 기법들을 수행하는 일 예의 방법을 예시하는 플로우차트이다.

### 발명을 실시하기 위한 구체적인 내용

- [0015] 일반적으로, 본 개시는 연속적인, 멀티-주기 스트림에의 광고 (ad) 삽입을 위한 기법들을 설명한다. 이들 기법들은 예를 들어, 유니캐스트 서비스 (이를 테면 HTTP), 브로드캐스트, 또는 멀티캐스트 서비스, 이를 테면 향상된 멀티미디어 브로드캐스트 멀티캐스트 서비스 (eMBMS) 에 따라, 미디어 데이터를 스트리밍할 때 이용될 수도 있다. 예를 들어, 본 개시의 기법들은 MBMS 개선들-향상된 MBMS 동작 (MBMS Improvements-Enhanced MBMS Operation; MI-EMO) 의 기법들과 함께, 또는 그 기법들을 증강시키는데 이용될 수도 있다. MI-EMO 는 예를 들어, [http://www.3gpp.org/ftp/Information/WORK\\_PLAN/Description\\_Releases/Rel-12\\_description\\_20131224.zip](http://www.3gpp.org/ftp/Information/WORK_PLAN/Description_Releases/Rel-12_description_20131224.zip) 에서 입수가 가능한, 3GPP Release 12 V0.1.1, December 2013 의 개관에서 설명된다.
- [0016] 광고 데이터는 미디어 데이터, 즉 프라이머리 미디어 데이터의 프리젠테이션의 하나 이상의 짧은 인터미션들 동안에 사용자에게 제시되는 데이터라는 것이 이해되어야 한다. 추가적인 미디어 데이터 (예를 들어, 광고 미디어 데이터) 는 일반적으로 프라이머리 미디어 데이터와 실질적으로 동일한 방식으로 제시된다. 예를 들어, 프라이머리 미디어 데이터가 오디오 데이터와 비디오 데이터 양자를 포함하면, 광고 미디어 데이터는 필시 오디오 데이터 및 비디오 데이터를 또한 포함할 것이다. 따라서, 광고 데이터는 사용자에게 일반적으로 제시되지 않는 클라이언트 디바이스에 대한 서버 광고와 혼동되지 않아야 한다.
- [0017] 본 개시의 기법들은 스트리밍 네트워크 프로토콜, 이를 테면 DASH (Dynamic Adaptive Streaming over HTTP) 의 맥락에서 적용될 수도 있다. DASH 에서, 미디어 콘텐츠는 표현들의 세트로 분할되고, 여기서 각각의 표현은 특성들, 예를 들어, 코딩 특성들 (이를 테면 CODEC, 프로파일, 레벨 등), 렌더링 특성들 (예를 들어, 뷰 각도, 픽처 디멘전들, 프레임 레이트 등), 및 비트 레이트의 특정한 조합에 대응한다. 미디어 콘텐츠는 적응 세트들을 더 포함할 수도 있고, 여기서 각각의 적응 세트는 서로에 대한 스위칭가능한 대안들로서의 역할을 하는 복수의 표현들을 포함하여, 그 표현들은 동일한 렌더링 특성들을 공유하지만 예를 들어, 대역폭 적응의 목적을 위해 비트레이트에 의해 상이하다. 표현들의 이들 특성들 뿐만 아니라 적응 세트들의 표시들은 통상 매니페스트 파일에서 시그널링된다. 더욱이, 각각의 적응 세트는 미디어 콘텐츠의 특정한 주기에 대응할 수도 있고, 여기서 각각의 주기는 하나 이상의 적응 세트들을 포함할 수도 있다. 하나의 주기는 일반적으로 주기에 대한 재생 시작 시간 (starting playback time) 에서 후속 주기에 대한 재생 시작 시간까지에 의해 정의되는 미디어 콘텐츠의 섹션에 대응한다. 미디어 콘텐츠는 하나 이상의 주기들을 포함할 수도 있다.
- [0018] 본 개시는 미디어 프리젠테이션 디스크립션 (Media Presentation Description; MPD) 과 같은 매니페스트 파일, 및 온-디맨드 (On-Demand) 또는 라이브 콘텐츠로서 생성된 콘텐츠의 주기 연속성을 제공하는 다수의 주기들을 가진 MPD (또는 다른 매니페스트 파일) 로의 변환에 관한 것이다. 다수의 주기들을 가진 콘텐츠는 상이한 이유들을 위해 생성될 수도 있다. 예를 들어, 다수의 주기들은 예를 들어 ad 삽입을 위한, 콘텐츠의 스폰라 이성을 인에이블하기 위해 생성될 수도 있다. 다른 예에서, 다수의 주기들은 세그먼트 넘버링의 드리프트를 회피하기 위한 동기화 포인트를 제공하기 위해 생성될 수도 있다. 다른 예에서, 다수의 주기들은 적응 세트에서 소정의 표현들을 제거 또는 추가하기 위해 생성될 수도 있다. 다른 예에서, 다수의 주기들은 소정의 적응 세트들을 제거 또는 추가하기 위해 생성될 수도 있다. 다른 예에서, 다수의 주기들은 소정의 콘텐츠 분배 네트워크 (content distribution network; CDN) 들 상에서 제공된 콘텐츠를 추가 또는 제거하기 위해 생성될 수도 있다. 다른 예에서, 다수의 주기들은 인코더에 의해 생성되면, 더 짧은 세그먼트들의 시그널링을 인에이블하기 위해 생성될 수도 있다. 이들 예의 기법들은 단독으로 또는 임의의 조합으로 이용될 수도 있다.
- [0019] 주기들은 큐 (cue) 들로도 또한 알려진 광고 삽입을 위한 기회들을 제공한다. 그러나 ad들은 전송기 또는 클라이언트가 행한 판정들에 기초하여 추가될 수도 있고 또는 추가되지 않을 수도 있다. 따라서, 어떤 ad 도 삽입되지 않은 콘텐츠는 통상적으로 미디어 파싱 및 렌더링의 관점에서 연속적이다. 주기들이 연속적이면, 클라이언트에 대한 연속적인 플레이아웃이 시그널링되어, 클라이언트는 재초기화할 필요가 없고 그리고 주기 경계에서의 글리치의 어떤 가능한 결과도 회피할 필요가 없다. 연속적인 플레이아웃을 다루기 위해, 본

개시는 다양한 기법들을 설명한다. 하나의 예에서, 콘텐츠 제공자는 규칙적인 온-디맨드 콘텐츠를 다수의 주기들로 스플리팅하고 주기 연속성 시그널링을 제공한다. 다른 예에서, 콘텐츠 제공자는 규칙적인 라이브 콘텐츠를 다수의 주기들로 스플리팅하고 주기 연속성 시그널링을 제공한다. 다른 예에서, 콘텐츠 제공자는 주기 연속성을 인에이블하기 위해 콘텐츠를 시그널링한다. 다른 예에서, 콘텐츠 제공자는 미디어 콘텐츠, 즉 표현들을 수정하지 않고도 단지 MPD 정보에 기초하여 다수의 주기들을 시그널링한다.

[0020] 일부 예들에서, 브로드캐스트 또는 멀티캐스트를 이용하여 미디어 콘텐츠를 수신할 때, MBMS 클라이언트 또는 eMBMS 클라이언트는 미디어 콘텐츠를 수신할 수도 있고, 그 후 미디어 콘텐츠를 스트리밍 클라이언트, 이를 테면 DASH 클라이언트에 이용가능하게 할 수도 있다. DASH 클라이언트는 예를 들어, HTTP 추출 동작들을 이용하여 MBMS 클라이언트로부터 미디어 콘텐츠를 추출할 수도 있다. HTTP 스트리밍, 이를 테면 DASH 에서, 빈번히 이용된 동작들은 HEAD, GET, 및 부분 GET 을 포함한다. HEAD 동작은 URL (uniform resource locator) 또는 URN (uniform resource name) 과 연관된 페이로드를 추출하지 않고, 주어진 URL 또는 URN 과 연관된 파일의 헤더를 추출한다. GET 동작은 주어진 URL 또는 URN 과 연관된 전체 파일을 추출한다. 부분 GET 동작은 입력 파라미터로서 바이트 범위를 수신하고 수신된 바이트 범위에 대응하는 파일의 연속적인 수의 바이트들을 추출한다. 따라서, 무비 프래그먼트들은 부분 GET 동작이 하나 이상의 개개의 무비 프래그먼트들을 입수할 수 있기 때문에 HTTP 스트리밍을 위해 제공될 수도 있다. 무비 프래그먼트에서, 상이한 트랙들의 여러 트랙 프래그먼트들이 존재할 수 있다. HTTP 스트리밍에서, 미디어 프리젠테이션은 클라이언트에 액세스가능할 수도 있는 데이터의 구조화된 컬렉션일 수도 있다. 클라이언트는 사용자에게 스트리밍 서비스를 제시하기 위해 미디어 데이터 정보를 요청 및 다운로드할 수도 있다.

[0021] HTTP 스트리밍을 이용하여 3GPP 데이터를 스트리밍하는 예에서, 멀티미디어 콘텐츠의 비디오 및/또는 오디오 데이터에 대한 다수의 표현들이 존재할 수도 있다. 이하 설명한 바와 같이, 상이한 표현들은 상이한 코딩 특성들 (예를 들어, 비디오 코딩 표준의 상이한 프로파일들 또는 레벨들), 상이한 코딩 표준들 또는 코딩 표준들의 확장들 (이를 테면 멀티뷰 및/또는 스케일러블 확장들), 또는 상이한 비트레이트들에 대응할 수도 있다. 이러한 표현들의 매니페스트는 미디어 프리젠테이션 디스크립션 (MPD) 데이터 구조에서 정의될 수도 있다. 미디어 프리젠테이션은 HTTP 스트리밍 클라이언트 디바이스에 액세스가능할 수도 있는 데이터의 구조화된 컬렉션에 대응할 수도 있다. HTTP 스트리밍 클라이언트 디바이스는 클라이언트 디바이스의 사용자에게 스트리밍 서비스를 제시하기 위해 미디어 데이터 정보를 요청 및 다운로드할 수도 있다. 미디어 프리젠테이션은 MPD 데이터 구조에서 설명될 수도 있고, 이는 MPD 의 업데이트들을 포함할 수도 있다.

[0022] 미디어 프리젠테이션은 하나 이상의 주기들의 시퀀스를 포함할 수도 있다. 주기들은 MPD 에서 주기 엘리먼트에 의해 정의될 수도 있다. MPD 는 각각의 주기에 대한 *start* 속성 및 *availableStartTime* 속성을 포함할 수도 있다. 라이브 서비스들에 대해, 주기의 *start* 속성과 MPD 속성 *availableStartTime* 의 합은 UTC 포맷으로 주기의 이용가능성 시간, 특히 대응하는 주기에서의 각각의 표현의 제 1 미디어 세그먼트를 특정할 수도 있다. 온-디맨드 서비스들에 대해, 제 1 주기의 *start* 속성은 0 일 수도 있다. 임의의 다른 주기에 대해, *start* 속성은 제 1 주기의 시작 시간에 대한 대응하는 주기의 시작 시간 간의 시간 오프셋을 특정할 수도 있다. 각각의 주기는 다음 주기의 시작까지, 또는 마지막 주기의 경우에서의 미디어 프리젠테이션의 종료까지 연장할 수도 있다. 주기 시작 시간들은 정확할 수도 있다. 그들은 모든 이전 주기 (prior period) 들의 미디어를 플레이하는 것으로부터 기인하는 실제 타이밍을 반영할 수도 있다.

[0023] 각각의 주기는 동일한 미디어 콘텐츠에 대한 하나 이상의 표현들을 포함할 수도 있다. 표현은 오디오 또는 비디오 데이터의 다수의 대안적 인코딩된 버전들 중 하나일 수도 있다. 표현들은 인코딩 타입들에 의해, 예를 들어, 비디오 데이터에 대한 비트레이트, 레졸루션, 및/또는 코덱 및 오디오 데이터에 대한 비트레이트, 언어, 및/또는 코덱에 의해 상이할 수도 있다. 용어 표현은 특정한 방식으로 인코딩되고 멀티미디어 콘텐츠의 특정한 주기에 대응하는 인코딩된 오디오 또는 비디오 데이터의 섹션을 지칭하는데 이용될 수도 있다.

[0024] 특정한 주기의 표현들은 그 표현들이 속하는 적응 세트를 표시하는 MPD 에서의 속성에 의해 나타낸 그룹에 할당될 수도 있다. 동일한 적응 세트에서의 표현들은, 예를 들어, 대역폭 적응을 수행하기 위해, 클라이언트 디바이스가 이들 표현들 간에 동적으로 및 심리스로 스위칭할 수 있다는 점에서, 일반적으로 서로에 대한 대안들인 것으로 고려된다. 예를 들어, 특정한 주기에 대한 비디오 데이터의 각각의 표현은 동일한 적응 세트에 할당될 수도 있어, 표현들 중 임의의 것은 대응하는 주기에 대한 멀티미디어 콘텐츠의 비디오 데이터 또는 오디오 데이터와 같은 미디어 데이터를 제시하기 위한 디코딩을 위해 선택될 수도 있다. 하나의 주기 내의 미디어 콘텐츠는 존재한다면, 그룹 0 으로부터의 하나의 표현, 또는 일부 예들에서, 각각의 년-제로 그룹으로부터의 많아 봐야 하나의 표현의 조합 중 어느 하나에 의해 표현될 수도 있다. 주기의 각각의 표현에 대한 타이밍

데이터는 주기의 시작 시간에 대해 표현될 수도 있다.

- [0025] 표현은 하나 이상의 세그먼트들을 포함할 수도 있다. 각각의 표현은 초기화 세그먼트를 포함할 수도 있고, 또는 표현의 각각의 세그먼트는 셀프-초기화하고 있을 수도 있다. 존재할 경우, 초기화 세그먼트는 표현에 액세스하기 위한 초기화 정보를 포함할 수도 있다. 일반적으로, 초기화 세그먼트는 미디어 데이터를 포함하지 않는다. 세그먼트는 식별자, 이를 테면 URL (uniform resource locator), URN (uniform resource name), 또는 URI (uniform resource identifier) 에 의해 고유하게 참조될 수도 있다. MPD 는 각각의 세그먼트에 대한 식별자들을 제공할 수도 있다. 일부 예들에서, MPD 는 또한, URL, URN, 또는 URI 에 의해 액세스가능한 파일 내의 세그먼트에 대한 데이터에 대응할 수도 있는 *range* 속성의 형태로 바이트 범위들을 제공할 수도 있다.
- [0026] 상이한 표현들은 상이한 타입들의 미디어 데이터에 대한 실질적으로 동시 추출을 위해 선택될 수도 있다. 예를 들어, 클라이언트 디바이스는 세그먼트들을 추출할 오디오 표현, 비디오 표현, 및 타임드 텍스트 표현을 선택할 수도 있다. 일부 예들에서, 클라이언트 디바이스는 대역폭 적응을 수행하기 위한 특정한 적응 세트들을 선택할 수도 있다. 즉, 클라이언트 디바이스는 비디오 표현들을 포함하는 적응 세트, 오디오 표현들을 포함하는 적응 세트, 및/또는 타임드 텍스트를 포함하는 적응 세트를 선택할 수도 있다. 대안적으로, 클라이언트 디바이스는 소정 타입들의 미디어 (예를 들어, 비디오) 에 대한 적응 세트들을 선택하고, 곧바로 다른 타입들의 미디어 (예를 들어, 오디오 및/또는 타임드 텍스트) 에 대한 표현들을 선택할 수도 있다.
- [0027] 통상적으로, 콘텐츠 제공의 관점에서 주기 경계에는 어떤 연속성도 필요하지 않다. 콘텐츠는 상이한 코덱들, 언어 속성들, 콘텐츠 보호 등으로 제공될 수도 있다. 클라이언트는 콘텐츠를 주기들에 걸쳐 연속적으로 플레이하지만, 플레이아웃을 완전 연속적이며 그리고 심리스하게 하기 위한 구현들의 관점에서 암시 (implication) 들이 존재할 수도 있다는 것이 예상된다.
- [0028] 일반적으로, 주기 경계들에 걸쳐 스위칭하는 오디오/비디오는 심리스가 아닐 수도 있다. ISO/IEC 23009-1, 섹션 7.2.1 에 따르면, 새로운 주기의 처음에는, 미디어 콘텐츠 컴포넌트들의 플레이아웃 절차는 선행하는 주기의 마지막에 표현과의 작은 오버랩들 또는 갭들이 존재할 수도 있기 때문에 새로운 주기의 periodStart 시작에 매칭하기 위해 선행하는 주기의 마지막에 조정될 필요가 있을 수도 있다. 오버랩들 (각각 갭들) 은 주기 지속기간에 의해 나타낸 것보다 더 긴 (각각 더 짧은) 미디어 스트림의 실제 프리젠테이션 지속기간을 가진 미디어 세그먼트들로부터 기인할 수도 있다. 또한, 주기의 시작부에서, 표현의 임의의 액세스 유닛의 가장 이른 프리젠테이션 시간이 @presentationTimeOffset 에서 시그널링된 프리젠테이션 시간 오프셋과 동일하지 않을 수도 있다면, 플레이아웃 절차들은 이에 따라 조정될 필요가 있다.
- [0029] 그러나, 소정의 조건들 하에서, 주기들에 걸친 프리젠테이션의 심리스 연속은 특히 콘텐츠가 적절히 컨디셔닝되면, 달성될 수도 있다. 이 섹션은 주기들에 걸쳐 콘텐츠를 다운로드 및 제시할 때 클라이언트 거동 및 주기들에 걸친 콘텐츠 제공 요건들 및 권고들에 대한 개관을 제공한다.
- [0030] 이하의 논의는 정적 및 동적 미디어 프리젠테이션들 양자에 적용하고, 이하에 설명된 기법들은 xlink 의 사용에 독립적일 수도 있다는 것에 주목한다. 이하에 설명된 규칙들은 원격 엘리먼트들이 역참조된 후의 케이스를 위해 적용할 수도 있다.
- [0031] 표현의 프리젠테이션 지속기간은 표현의 종료 프리젠테이션 시간과 표현의 가장 이른 프리젠테이션 시간 사이의 차이일 수도 있다. 프리젠테이션 시간 지속기간은 예를 들어, 프리젠테이션 시간 오프셋, 즉 @timescale 과 동일한 유닛을 가질 수도 있고, 표현의 정확한 프리젠테이션 지속기간을 표현한다.
- [0032] 프리젠테이션 지속기간은 2 개의 다음의 예의 기법들, 또는 유사한 기법들 중 하나에 의해 시그널링될 수도 있다. 하나의 예의 기법은 @timescale 속성의 시간 스케일에서의 포함된 표현의 프리젠테이션 지속기간을 특정하는 새로운 옵션적 속성 @presentationTimeDuration 을 수반한다. 제 2 예의 기법은 MPD 에 포함될 수도 있는 주기의 @id 의 값, AID 인 AdaptationSet@id 의 값, 및 이 적응 세트들에 대한 @presentationTimeOffset 의 값에 매칭하는, 디스크립터, PID 의 @value 를 가진 적응 세트에 대해 제공될 수도 있는, "urn:mpeg:dash:period\_continuity:2014" 로 설정된 @scheme\_id\_URI 를 가진 보충적 디스크립터를 수반한다.
- [0033] 이 신호가 존재하면, PID 인 peiroad@id 의 값을 가진 주기에 대해 및 AID 인 AdaptationSet@id 를 가진 적응 세트에 대해, 이 적응 세트에서의 각각의 표현의 프리젠테이션 지속기간은 PTO 마이너스 @presentationTimeOffset 의 값의 차이로서 획득될 수도 있다.

- [0034] 콘텐츠가 다수의 주기들로 제공될 수도 있으면, 콘텐츠 제공자는 실제 미디어 프리젠테이션 시간이 실제 주기 지속기간에 가능한 근접할 수도 있도록 콘텐츠를 제공할 수 있다. 주기 지속기간은 주기에 포함된 모든 표현들의 프리젠테이션 지속기간의 최대치일 수도 있다는 것이 권고될 수도 있다.
- [0035] 소정의 환경들에서, 콘텐츠 제공자는 이전의 주기에서의 콘텐츠의 연속일 수도 있는 다음 주기에서, 가능하게는 바로 후속하는 주기에서 또는 추후의 주기에서 콘텐츠를 제공한다. 후자의 케이스는 예를 들어, 광고 주기가 삽입된 후에 적용한다. 콘텐츠 제공자는 2 개의 상이한 주기들에서 2 개의 적응 세트들에 포함된 미디어 컴포넌트들이 양자의 주기들에 등가의 애셋 식별자들을 할당함으로써 및 속성 @id 에 대해 동일한 값을 가진 양자의 적응 세트들을 식별함으로써 연관된다는 것을 표현할 수도 있다.
- [0036] 2 개의 상이한 주기들에서의 적응 세트들이 연관되면, 후속하는 파라미터들 중 임의의 것 또는 전부는 2 개의 적응 세트들에 대해 동일할 수도 있다: @lang 속성에 의해 기술한 바와 같은 언어, @contentType 속성에 의해 기술된 미디어 컴포넌트 타입, @par 속성에 의해 기술한 바와 같은 픽처 애스펙트비, 역할 엘리먼트들에 의해 기술한 바와 같은 임의의 역할 속성, 액세스가능성 엘리먼트들에 의해 기술한 바와 같은 임의의 액세스가능성 속성, 및 뷰포인트 엘리먼트들에 의해 기술한 바와 같은 임의의 뷰포인트 속성. 하나의 예에서, DASH 표준은 2 개의 상이한 주기들에서의 적응 세트들이 연관되는데, 상기 파라미터들의 각각이 2 개의 적응 세트들에 대해 동일하다는 것을 요구하기 위해 본 개시의 기법들에 따라 수정될 수도 있다.
- [0037] 더욱이, 하나의 MPD 에서의 2 개의 적응 세트들은 다음의 조건들의 각각이 2 개의 적응 세트들에 대해 참이면 주기-연속적이라고 할 수도 있다:
- [0038] • 적응 세트들은 연관되는 것으로서 나타내지고, 및/또는 연관되는 것으로 결정된다,
- [0039] • @presentationTimeOffset 은 존재할 수도 있거나 또는 양자의 적응 세트들에서의 모든 표현들에 대해 0 으로서 추론될 수 있다,
- [0040] • 하나의 적응 세트 내에서, @presentationTimeOffset 의 값은 모든 표현들에 대해 동일할 수도 있다,
- [0041] • @presentationTimeOffset 의 값과 하나의 적응 세트에서의 모든 표현들의 프리젠테이션 지속기간의 합은 다른 적응 세트의 @presentationTimeOffset 의 값과 동일하다, 그리고
- [0042] • 양자의 적응 세트들에서의 표현들이 @id 에 대해 동일한 값을 가지면, 그들은 기능적으로 등가인 초기화 세그먼트들을 갖고, 즉, 하나의 표현의 초기화 세그먼트는 다른 표현의 플레이-아웃을 연속하기 위해 이용될 수도 있다.
- [0043] 콘텐츠 저자는 상기 도입된 2 개의 예의 기법들 중 하나를 이용하여 (예를 들어, "urn:mpeg:dash:period:continuity:2014" 로 설정된 @scheme\_id\_URI 를 가진 보충적 디스크립터 및/또는 @presentationTimeDuration 속성과 같은 프리젠테이션 지속기간 시그널링을 이용하여) 프리젠테이션 지속기간을 시그널링함으로써 주기-연속적 적응 세트들을 시그널링할 수 있다. 콘텐츠 저자는 MPD 가 동일한 애셋 식별자들을 가진 주기들을 포함하면 주기-연속적 적응 세트들을 가진 MPD 를 제공할 수 있다. 더욱이, 하나의 MPD 에서의 2 개의 적응 세트들이 주기-연속적이고 제 2 적응 세트가 보충적 디스크립터를 @schemeIDURI="urn:mpeg:dash:period-switchable" 과 연관시켰다면, 제 1 적응의 임의의 표현의 초기화 세그먼트 없는 제 2 적응 세트로부터의 임의의 표현과의 연접은 제 1 적응 세트에서 시그널링한 바와 같이 사용 시에 미디어 포맷들에 따르는 비트스트림 포맷을 초래한다. 이 속성은 주기-연접가능한 적응 세트 (period-concatenable adaptation set) 들로 지칭될 수도 있다. 클라이언트들에서의 단순화된 프로세싱을 위해, 콘텐츠 저자는 MPD 가 동일한 애셋 식별자들을 가진 주기들을 포함하면 주기-연접가능한 적응 세트들을 가진 MPD 를 제공할 수도 있다.
- [0044] 통상적으로, 콘텐츠 제공의 관점에서는, 어떤 연결성도 주기 경계에서 보장되지 않는다. 콘텐츠는 다른 것들 중에서도 상이한 코덱들, 언어 속성들, 및 상이한 보호로 제공될 수도 있다. 클라이언트는 콘텐츠를 주기들에 걸쳐 연속적으로 플레이할 수 있지만, 완전 연속적이고 심리스한 플레이아웃을 제공하기 위한 구현의 관점에서 암시들이 존재할 수도 있다. 그것은 (예를 들어, 어떤 연속성도 없는) 주기 경계들에서, 프리젠테이션 엔진은 예를 들어, 포맷들, 코덱들 또는 다른 속성들의 변화들로 인해 재초기화될 수도 있는 케이스일 수도 있다. 이것은 재초기화 지연을 초래할 수도 있다. 이러한 재초기화 지연은 감소되거나 또는 최소화될



수 있다.

- [0045] MPD 가 타입 정적이라면, 실제 미디어 프리젠테이션은 초기화 지연들에 의해 확장될 수 있다. 그러나, 동적 서비스들의 케이스에서, 플레이아웃에의 재초기화 지연의 추가는 인코더와 프리젠테이션 엔진 사이에 드리프트를 초래할 수도 있다. 따라서, 플레이아웃은 MPD 에서 기록된 시간과 실제 플레이아웃 사이의 드리프트를 추가하지 않고 연속적인 프리젠테이션을 제공하기 위해 각각의 주기의 마지막에 조정될 수 있다, 즉 실제 플레이아웃 시간과 주기 시작 시간 사이의 차이는 일정한 상태일 수 있다.
- [0046] 클라이언트가 하나의 주기에서 소정의 적응 세트의 미디어 컴포넌트들을 제시하면, 그리고 후속하는 주기가 동일한 애셋 식별자를 할당했다면, 클라이언트는 연관된 주기를 식별할 수 있고, 다른 정보의 부재 시에는, 연관된 적응 세트에서의 콘텐츠를 계속 플레이할 수 있다.
- [0047] 더욱이 적응 세트들이 주기-연속적이면, 즉, 프리젠테이션 시간들이 연속적이고 이것이 MPD 에서 시그널링될 수도 있다면, 클라이언트는 콘텐츠를 주기 경계에 걸쳐 심리스로 플레이할 수도 있다. 가장 적합하게는, 클라이언트는 동일한 @id 를 가진 적응 세트에서 표현을 계속 플레이할 수도 있지만, 이 표현이 이용가능할 수도 있다는 어떤 보장도 없을 수도 있다. 이 경우에, 클라이언트는 적응 세트에서 임의의 다른 표현으로 심리스로 스위칭해야 한다.
- [0048] 입력 포맷은 온-디맨드 프로파일에 따르는 MPD 일 수도 있다. MPD 의 관련 엘리먼트들 및 속성들은 이하의 표에 제공된다. 이하의 예에서, MPD 는 2 개의 적응 세트들을 포함한다.

표 1

표 1—온-디맨드 프로파일에 대한 관련 정보

MPD 정보	상태	코멘트
<b>MPD@type</b>	의무적, "정적" 으로 설정됨	미디어 프리젠테이션의 타입은 정적일 수도 있다, 즉, 세그먼트들은 이용가능성 시작 시간에서 이용가능해진다.
<b>MPD@profiles</b>	의무적, "urn:mpeg:dash:profile:isoff-on-demand:2011" 로 설정됨	MPD 프로파일은 온-디맨드로 설정될 수도 있다
<b>AdaptationSet@subsegment Alignment</b>	참으로 설정되는 것이 의무적임	
<b>AdaptationSet@subsegmentStarts WithSAP</b>	1 또는 2 로 설정되는 것이 의무적임	
<b>representation@bandwidth</b>		
<b>representation.BaseURL</b>		
<b>SegmentBase@timescale</b>	옵션적 디폴트	미디어 프리젠테이션의 프리젠테이션 시간 오프셋.
<b>SegmentBase@presentationTime Offset</b>	옵션적 디폴트	미디어 프리젠테이션의 프리젠테이션 시간 오프셋.
<b>SegmentBase@presentation Duration</b>	옵션적	업데이트에서 DASH 사양에 추가될 예정임
<b>SegmentBase@indexRange</b>	DASH-IF/264 에 존재하는 것이 의무적임	

[0049]

[0050]

각각의 표현에 대해, 세그먼트 인덱스는 다음과 같이 init 으로서 표시된 SegmentBase@indexRange 의 값 및 baseURL 로서 표시된 BaseURL 에 제공된 표현의 URL 로 다운로드될 수도 있다:

# GET baseURL HTTP/1.1

Range: bytes=0-(init-1)

[0051]

[0052]

그 후, 수신된 바이트 범위는 세그먼트 인덱스에 대해 파싱될 수도 있다. 제 1 세그먼트 인덱스 박스로부터, 정보가 추출될 수도 있다. 트랙의 타임스케일은 세그먼트 인덱스 타임스케일로부터 추출될 수도 있고 타임스케일로서 표시될 수도 있다. 서브세그먼트들의 총 수 Nsub 는 세그먼트 인덱스로부터 reference\_count 로서 추출될 수도 있다. 가장 이른 프리젠테이션 시간 ept 는 세그먼트 인덱스로부터

earliest\_presentation\_time 으로서 추출될 수도 있다. 제 1 오프셋 fo 는 세그먼트 인덱스로부터 first\_offset 으로서 추출될 수도 있다. 각각의 리프 (leaf) 서브세그먼트 i 에 대해, 각각의 리프 서브세그먼트의 세그먼트 인덱스에서의 서브세그먼트 지속기간은 sdur[i] 로서, 서브세그먼트의 사이즈는 size[i] 로서, 그리고 다른 서브세그먼트 관련 정보는 ssi[i] 로서 표시된다.

[0053] 온-디맨드 스트림의 하나의 예에서, 프리젠테이션은 주기 시작 시간에 대해 표현된 미디어 프리젠테이션 시간 tsplit 에서 가장 이른 2 개의 주기들로 스플리팅될 수도 있다. 2 개의 적응 세트들이 존재할 수도 있고 각각의 적응 세트는 하나의 표현을 가질 수도 있다. 일부 예들에서, 적응 세트들의 서브세그먼트 경계의 정렬이 적용될 수도 있다.

[0054] 다음의 정보는 단일의 주기 콘텐츠에 대해 이용가능한 것으로 가정될 수도 있다.

- [0055] • period@start ps
- [0056] • MPD@mediaPresentationDuration mpDUR
- [0057] • BaseURL bu1, bu2
- [0058] • SegmentBase@indexRange ir1, ir2
- [0059] • SegmentBase@timeScale ts1, ts2
- [0060] • SegmentBase@presentationDuration pd1, pd2
- [0061] • 표현의 세그먼트 인덱스
  - [0062] ◦ Nsub1, Nsub2
  - [0063] ◦ ept1, ept2
  - [0064] ◦ fo1, fo2
  - [0065] ◦ sdur1[i], sdur2[i]
  - [0066] ◦ fo1[i], fo2[i]
  - [0067] ◦ ssi1[i], ssi2[i]

[0068] 새로운 미디어 프리젠테이션을 생성하기 위하여, 다음의 정보가 생성될 수도 있다:

[0069] 각각의 적응 세트의 각각의 표현에 대한 제 1 주기에서의 속성들 및 엘리먼트들의 업데이트:

- [0070] a. **period@start** ps1
- [0071] b. add **period@duration** pdur1 = tsplit
- [0072] c. **BaseURL** bu1, bu2
- [0073] d. **SegmentBase@indexRange** ir11, ir12
- [0074] e. **SegmentBase@presentationTimeOffset** pto11, pto12
- [0075] f. **SegmentBase@presentationDuration** pd11, pd12
- [0076] g. 표현의 세그먼트 인덱스
  - [0077] i. Nsub11, Nsub12
  - [0078] ii. ept11, ept12
  - [0079] iii. fo11, fo12

- [0080]           iv.       sdur11[i], sdur12[i]
- [0081]           v.       ssi11[i], ssi12[i]
- [0082] h.       표현 자체는 단축될 수도 있다
- [0083] 각각의 표현에 대한 새로운 제 2 주기에서의 엘리먼트들 및 속성들의 업데이트 및 추가:
- [0084] a.       add **period**@duration   pdur2
- [0085] b.       **BaseURL** bu21, bu22
- [0086] c.       **SegmentBase**@indexRange ir21, ir22
- [0087] d.       **SegmentBase**@presentationTimeOffset pto21, pto22
- [0088] e.       **SegmentBase**@presentationDuration pd21, pd22
- [0089] f.       표현의 세그먼트 인덱스
- [0090]           i.       Nsub21, Nsub22
- [0091]           ii.      ept21, ept22
- [0092]           iii.     fo21, fo22
- [0093]           iv.      sdur21[i], sdur22[i]
- [0094]           v.       ssi21[i], ssi22[i]
- [0095] 표현 자체가 생성될 수도 있다.
- [0096] 새로운 정보는 세그먼트 경계와 정렬하는 제 1 표현에 대해 다음과 같이 생성될 수도 있다.       먼저 예를 들어, 다음의 알고리즘에 따라, tsplit 에 대응하는 Nsub11 이 발견될 수도 있다:
- [0097]       •       time = ept1 - pto1
- [0098]       •       Nsub11 = 0
- [0099]       •       while ( time < tsplit\*ts1 && Nsub11 < Nsub1 ) )
- [0100]       •       do
- [0101]           o       time += sdur[Nsub11]
- [0102]           o       Nsub11++
- [0103]       •       done
- [0104] 이 알고리즘에 의해, Nsub11 이 결정될 수도 있다.       Nsub11 은 Nsub1 보다 더 작을 수도 있다.       Nsub11 이 Nsub1 보다 더 작지 않다면, 어떤 주기 스플릿도 반드시 행해지는 것은 아니다.
- [0105] 표현 1 에 대한 스플리팅 시간은 정확한 세그먼트 경계에서의 스플릿이 요망될 수도 있기 때문에 tsplit1 = time/ts1 >= tsplit 일 수도 있다.       공표된 스플릿은 tsplit 일 수도 있지만, 표현 레벨에서, 단지 세그먼트 경계들만이 스플리팅된다는 것에 주목한다.       새로운 세그먼트 인덱스 사이즈는 이전의 것보다 더 작을 수도 있고, 즉, 새로운 세그먼트 인덱스 사이즈는 (Nsub1 - Nsub11)\*12 바이트 더 작을 수도 있다.       이것은 제 1 오프셋으로부터 공제될 수도 있다.
- [0106] 시간은 프리젠테이션 지속기간 pd11 일 수도 있다.       주기 1 의 period@start 는 유지될 수도 있다. period@duration 은 tsplit 으로 설정될 수도 있다.       다음의 알고리즘은 새로운 정보가 주기 1 에서 표현 1 에 대해 생성될 수도 있는 방법의 하나의 예이다.
- [0107]       •       bul1 = bul\_period1



[0108]       •        $ir11 = ir1 - (Nsub1 - Nsub11)*12$

[0109]       •        $pto11 = ptol$

[0110]       •        $pd11 = time$

[0111]       •        $ept11 = ept1$

[0112]       •        $fo11 = fol - (Nsub1 - Nsub11)*12$

[0113]       •        $i = 0$

[0114]       •       while (while (i < Nsub11) )

[0115]       •       do

[0116]           ◦        $ssi11[i] = ssil[i]$

[0117]           ◦        $i++$

[0118]       •       done

[0119]       그 후 예를 들어, 다음의 알고리즘에 따라  $tsplit$  에 대응하는  $Nsub12$  가 발견될 수도 있다:

[0120]       •        $time = ept2 - pto2$

[0121]       •        $Nsub12 = 0$

[0122]       •       while ( time <  $tsplit*ts2$  &&  $Nsub12 < Nsub2$  ) )

[0123]       •       do

[0124]           ◦        $time += sdur[Nsub12]$

[0125]           ◦        $Nsub12++$

[0126]       •       done

[0127]       표현 2 에 대한 실제 스플리팅 시간은  $tsplit2 = time/ts2 \geq tsplit$  일 수도 있고, 정확한 서브세그먼트 경계에서의 스플릿이 요망될 수도 있다.       새로운 세그먼트 인덱스 사이즈는 이전의 것보다 더 작을 수도 있고, 예를 들어, 그것은  $(Nsub2 - Nsub12)*12$  바이트 더 작을 수도 있다.       이것은 제 1 오프셋으로부터 공제될 수도 있다.

[0128]       시간은 프리젠테이션 지속기간  $pd21$  일 수도 있다.       이하의 알고리즘은 새로운 정보가 주기 1 에서 표현 2 에 대해 생성될 수도 있는 방법의 하나의 예이다.

[0129]       •        $bu12 = bu2\_period1$

[0130]       •        $ir12 = ir2 - (Nsub2 - Nsub12)*12$

[0131]       •        $pto12 = pto2$

[0132]       •        $pd12 = time$

[0133]       •        $ept12 = ept2$

[0134]       •        $fo12 = fo2 - (Nsub2 - Nsub12)*12$

[0135]       •       i = 0

[0136]       •       while (while (i < Nsub12) )

[0137]       •       do

[0138]           ◦       ssi12[i] = ssi2[i]

[0139]           ◦       i++

[0140]       •       done

[0141] 새로운 주기 2 가 생성될 수도 있다.       특히, 하나의 예에서, 서버 디바이스는 새로운 주기 2 를 생성할 수도 있다.       무엇보다도, 서버 디바이스 (60) 는 주기 1 로부터의 정보 (또는 애셋 식별자, 및 id들을 가진 적응 세트들 등) 를 카피할 수도 있다.       새로운 주기의 period@start 는 제공될 필요가 없다.       새로운 주기의 period@duration pdur2 는 mpDUR - tsplit 일 수도 있다.       다음의 알고리즘은 새로운 정보가 주기 2 에서 표현 1 에 대해 생성될 수도 있는 방법의 일 예이다.

[0142]       •       Nsub12 = Nsub1 - Nsub11

[0143]       •       bu21 = bu1\_period2

[0144]       •       ir21 = ir1 - (Nsub1 - Nsub12)\*12

[0145]       •       pto21 = (tsplit1 - tsplit)\*ts1

[0146]       •       pd21 = pd1 - pd11

[0147]       •       ept21 = pd11

[0148]       •       fo21 = fo1 - (Nsub1 - Nsub12)\*12

[0149]       •       i = 0

[0150]       •       while (while (i < Nsub12) )

[0151]       •       do

[0152]           ◦       ssi21[i] = ssi1[i+Nsub11]

[0153]           ◦       i++

[0154]       •       done

[0155] 다음의 알고리즘은 새로운 정보가 주기 2 에서 표현 2 에 대해 생성될 수도 있는 방법의 하나의 예이다.

[0156]       •       Nsub22 = Nsub2 - Nsub12

[0157]       •       bu22 = bu2\_period2

[0158]       •       ir22 = ir2 - (Nsub2 - Nsub22)\*12

[0159]       •       pto22 = (tsplit2 - tsplit)\*ts2

[0160]       •       pd22 = pd2 - pd12

[0161]       •       ept22 = pd12

[0162]       •       fo22 = fo2 - (Nsub2 - Nsub22)\*12

[0163]       •       i = 0

[0164]       •       while (while (i < Nsub22) )

[0165]       •       do

[0166]           ◦       ssi22[i] = ssi2[i+Nsub12]

[0167]           ◦       i++

[0168]       •       done

[0169]       하나의 예에서, tsplit1 은 0 인 pto21 을 갖도록 하기 위하여 스플릿 시간으로서 이용될 수도 있다.       스플리팅이 IDR 프레임에서 행해질 수도 있을 때, 이것이 그 케이스일 수도 있다.

[0170]       상기 정보는 동일한 정보를 가진 완전한 MPD 에 다시 맵핑될 수도 있지만, 상기 논의된 알고리즘에 의해, 시간 tsplit 에서의 스플라이스 포인트가 생성될 수도 있다.       임의의 주기는 스플라이스 포인트에 삽입될 수 있다.       상기 노력은 MPD@mediaPresentationDuration 을 제거하고 상기 기록한 바와 같이 추가 및 대체할 수도 있다.

[0171]       본 개시의 다른 예의 기법에서, 라이브 스트림이 이용될 수도 있다.       라이브 프로파일 정보에 대해, 정보는 MPD 및 세그먼트들로부터 이용가능할 수도 있다.       트랙의 타임스케일은 세그먼트 인덱스 타임스케일로부터 추출될 수도 있고 타임스케일로서 표시될 수도 있다.       세그먼트들의 총 수 Nseg 는 세그먼트 정보로부터 추출될 수도 있다.       가장 이른 프리젠테이션 시간 ept 는 세그먼트 인덱스로부터 earliest\_presentation\_time 으로서 추출될 수도 있다.       제 1 오프셋 @startNumber 는 MPD 로부터 추출될 수도 있다.       각각의 세그먼트 I 에 대해, 각각의 리프 서브세그먼트의 세그먼트 인덱스에서의 세그먼트 지속기간은 sdur[i] 로서 그리고 다른 세그먼트 관련 정보는 si[i] 로서 표시된다.

[0172]       하나의 예의 프리젠테이션에서, 프리젠테이션은 주기 시작 시간에 대해 표현된 미디어 프리젠테이션 시간 tsplit 에서 가장 이른 2 개의 주기들로 스플리팅될 수도 있다.       이 스플리팅은 제 2 적응 세트의 세그먼트 경계와 정렬하지 않을 수도 있다.       2 개의 적응 세트들이 존재할 수도 있고 각각의 적응 세트는 하나의 표현을 가질 수도 있다.

[0173]       다음의 정보는 단일의 주기 콘텐츠에 대해 이용가능한 것으로 가정될 수도 있다.

[0174]       •       **period**@start ps

[0175]       •       **MPD**@mediaPresentationDuration mpDUR

[0176]       •       **SegmentTemplate**@media mt1, mt2

[0177]       •       **SegmentTemplate**@timecale ts1, ts2

[0178]       •       **SegmentTemplate**@presentationDuration pd1, pd2

[0179]       •       세그먼트 정보

[0180]           ◦       Nseg1, Nseg2

[0181]           ◦       ept1, ept2

[0182]           ◦       sn1, sn2

[0183]           ◦       sdur1[i], sdur2[i]

- [0184]           ◦        si1[i], si2[i]
- [0185] 새로운 미디어 프리젠테이션을 생성하기 위하여, 다음의 정보가 생성될 수도 있다:
- [0186]       • 서버 디바이스는 다음과 같이, 각각의 적응 세트의 각각의 표현에 대해 제 1 주기에서의 속성들 및 엘리먼트들을 업데이트할 수도 있다:
- [0187]           ◦        **period**@start ps1
- [0188]           ◦        add **period**@duration   pdur1 = tsplit
- [0189]           ◦        **SegmentTemplate**@media mt11, mt12
- [0190]           ◦        **SegmentTemplate**@presentationTimeOffset pto11, pto12
- [0191]           ◦        **SegmentTemplate**@presentationDuration pd11, pd12
- [0192]           ◦        표현의 세그먼트 정보
- [0193]               ▪        Nseg11, Nseg12
- [0194]               ▪        ept11, ept12
- [0195]               ▪        sn11, sn12
- [0196]               ▪        sdur11[i], sdur12[i]
- [0197]               ▪        si11[i], si12[i]
- [0198]           ◦        표현 자체는 단축될 수도 있다
- [0199]       • 서버 디바이스는 다음과 같이, 각각의 표현에 대한 새로운 제 2 주기에서의 엘리먼트들 및 속성들을 업데이트 및 추가할 수도 있다:
- [0200]           ◦        add **period**@duration   pdur2
- [0201]           ◦        **SegmentTemplate**@media mt21, mt22
- [0202]           ◦        **SegmentTemplate**@presentationTimeOffset pto21, pto22
- [0203]           ◦        **SegmentTemplate**@presentationDuration pd21, pd22
- [0204]           ◦        표현의 세그먼트 정보
- [0205]               ▪        Nseg21, Nseg22
- [0206]               ▪        ept21, ept22
- [0207]               ▪        sn21, sn22
- [0208]               ▪        sdur21[i], sdur22[i]
- [0209]               ▪        si21[i], si22[i]
- [0210]           ◦        표현 자체가 생성될 수도 있다
- [0211] 서버 디바이스는 세그먼트 경계와 정렬하는 제 1 표현에 대해 다음과 같이 새로운 정보를 생성할 수도 있다.

먼저, 다음의 알고리즘에 따라 `tsplit` 에 대응하는 `Nseg11` 이 발견될 수도 있다:

```
[0212] •      time = ept1 - ptol
[0213] •      Nseg11 = 0
[0214] •      while ( time < tsplit*ts1 && Nseg11 < Nseg1 ) )
[0215] •      do
[0216]   ◦      time += sdur[Nseg11]
[0217]   ◦      Nseg11++
[0218] •      done
```

[0219] 이 알고리즘에 의해, 서버 디바이스는 `Nseg11` 을 결정할 수도 있다. `Nseg11` 은 `Nseg1` 보다 더 작을 수도 있다. `Nseg11` 이 `Nseg1` 보다 더 작지 않다면, 어떤 주기 스플릿도 행해지지 않을 수도 있다. 시간은 프리젠테이션 지속기간 `pd11` 일 수도 있다. 표현 1 에 대한 스플리팅 시간은 정확한 세그먼트 경계에서의 스플릿이 요망될 수도 있기 때문에 `tsplit1 = time/ts1 >= tsplit` 일 수도 있다. 공표된 스플릿은 `tsplit` 일 수도 있지만, 표현 레벨에서, 단지 세그먼트 경계들이 스플리팅될 수도 있다는 것에 주목한다. 주기 1 의 `period@start` 는 유지될 수도 있다. `period@duration` 은 `tsplit` 으로 설정될 수도 있다. 다음의 알고리즘은 새로운 정보가 주기 1 에서 표현 1 에 대해 생성될 수도 있는 방법의 일 예이다.

```
[0220] •      mt11 = mt1
[0221] •      ptol1 = ptol
[0222] •      pd11 = time
[0223] •      ept11 = ept1
[0224] •      sn11 = sn1
[0225] •      i = 0
[0226] •      while (while (i < Nseg11) )
[0227] •      do
[0228]   ◦      sil1[i] = sil[i]
[0229]   ◦      i++
[0230] •      done
```

[0231] 그 후 예를 들어, 다음의 예의 알고리즘에 따라, `tsplit` 에 대응하는 `Nseg12` 가 발견될 수도 있다:

```
[0232] •      time = ept2 - pto2
[0233] •      Nseg12 = 0
[0234] •      while ( time < tsplit*ts2 && Nseg12 < Nseg2 ) )
[0235] •      do
[0236]   ◦      time += sdur[Nseg12]
```

[0237]           o       Nseg12++

[0238]       •       done

[0239]       표현 2 에 대한 실제 스플리팅 시간은 정확한 서브세그먼트 경계에서의 스플릿이 요망될 수도 있기 때문에  $tsplit2 = time/ts2 \geq tsplit$  일 수도 있다.       시간은 프리젠테이션 지속기간  $pd12$  일 수도 있다.       다음의 알고리즘은 새로운 정보가 주기 1 에서 표현 2 에 대해 생성될 수도 있는 방법의 일 예이다.

[0240]       •        $mt12 = mt2$

[0241]       •        $pto12 = pto2$

[0242]       •        $pd12 = time$

[0243]       •        $ept12 = ept2$

[0244]       •        $sn12 = sn2$

[0245]       •        $i = 0$

[0246]       •       while (while (  $i < Nseg12$  ) )

[0247]       •       do

[0248]           o        $si12[i] = si2[i]$

[0249]           o        $i++$

[0250]       •       done

[0251]       새로운 주기 2 가 생성될 수도 있다.       무엇보다도, 주기 1 로부터의 정보 (또한 애셋 식별자, 및 id들을 가진 적응 세트들 등) 가 카피될 수도 있다.       새로운 주기의  $peirod@start$  는 제공되지 않을 수도 있다. 새로운 주기의  $period@duration$   $pdur2$  는  $mpDUR - tsplit$  일 수도 있다.       다음의 알고리즘은 새로운 정보가 주기 2 에서 표현 1 에 대해 생성될 수도 있는 방법의 일 예이다.

[0252]       •        $Nseg12 = Nseg1 - Nseg11$

[0253]       •        $mt21 = mt1$

[0254]       •        $pto21 = (tsplit1 - tsplit)*ts1$

[0255]       •        $pd21 = pd1 - pd11$

[0256]       •        $ept21 = pd11$

[0257]       •        $sn21 = sn1 + Nseg11$

[0258]       •        $i = 0$

[0259]       •       while (while (  $i < Nseg12$  ) )

[0260]       •       do

[0261]           o        $si21[i] = si1[i+Nseg11]$

[0262]           o        $i++$

- [0263]       •       done
- [0264]       다음의 알고리즘은 새로운 정보가 주기 2 에서 표현 2 에 대해 생성될 수도 있는 방법의 일 예이다.
- [0265]       •       Nseg22 = Nseg2 - Nseg12
- [0266]       •       mt22 = mt2
- [0267]       •       pto22 = (tsplit1 - tsplit)\*ts2
- [0268]       •       pd22 = pd2 - pd12
- [0269]       •       ept22 = pd12
- [0270]       •       sn22 = sn2 + Nseg12
- [0271]       •       i = 0
- [0272]       •       while (while (i < Nseg22) )
- [0273]       •       do
- [0274]           ◦       si22[i] = si2[i+Nseg12]
- [0275]           ◦       i++
- [0276]       •       done
- [0277]       일부 예들에서, tsplit1 은 0 인 pto21 을 갖도록 하기 위하여 스플릿 시간으로서 이용될 수도 있다.       스플리팅이 IDR 프레임에서 행해질 수도 있을 때, 이것이 그 케이스일 수도 있다.
- [0278]       상기 정보는 동일한 정보를 가진 완전한 MPD 에 다시 맵핑될 수 있지만, 이것에 의해 시간 tsplit 에서의 스플라이스 포인트가 생성될 수도 있다.       임의의 주기가 여기에 삽입될 수 있다.       상기의 효과는 MPD@mediaPresentationDuration 을 제거하고 상기 기록된 정보를 추가 및 대체할 수도 있다.
- [0279]       현재의 개시의 기법들을 이용하여, 온-디맨드 프로파일이 최적화될 수도 있다.       온-디맨드 프로파일 제공에 의한 이슈는 ad 삽입을 위한 주기 구성을 행하기 위하여, 파일을 물리적으로 수정해야 한다는 것일 수도 있다.       이것은 2 개의 결과들을 갖는다.       첫째로, 수정들은 파일 레벨 수정들을 요구하고 상대적으로 복잡하다.       이것은 ad 삽입을 위해, 콘텐츠가 변경될 수도 있다는 것을 의미한다.       둘째로, 상이한 ad 스플라이싱 포인트들이 고려되면, 상이한 콘텐츠 및 상이한 URL들이 이용될 수도 있다.       이것은 동일한 콘텐츠에 대해, 저장 및 특히 캐싱 (caching) 효율이 감소될 수도 있다는 것을 의미한다.       따라서, MPD-레벨 정보에만 기초하여 주기 및 ad 삽입을 인에이블하는 것이 다음에서 제안될 수도 있다.       이것은 MPD 의 몇몇 수정들을 요구하고 이들 틀들은 xlink 를 포함하는 새롭게 개발된 ad 삽입 프로파일에 추가되어야 한다.
- [0280]       온-디맨드 스트림의 하나의 예에서, 프리젠테이션은 주기 시작 시간에 대해 표현된 미디어 프리젠테이션 시간 tsplit 에서 가장 이른 2 개의 주기들로 스플리팅될 수도 있다.       2 개의 적응 세트들이 존재할 수도 있고 각각의 적응 세트는 하나의 표현을 가질 수도 있다.       일부 예들에서, 적응 세트들의 서브세그먼트 경계의 정렬은 적용될 수도 있다.       다음의 정보는 단일의 주기 콘텐츠에 대해 이용가능한 것으로 가정될 수도 있다.
- [0281]       •       **period**@start ps
- [0282]       •       **MPD**@mediaPresentationDuration mpDUR
- [0283]       •       **BaseURL** bu1, bu2
- [0284]       •       **SegmentBase**@indexRange ir1, ir2

- [0285]       •       **SegmentBase@timeScale** ts1, ts2
- [0286]       •       **SegmentBase@presentationDuration** pd1, pd2
- [0287]       •       표현의 세그먼트 인덱스
  - [0288]           ◦       Nsub1, Nsub2
  - [0289]           ◦       ept1, ept2
  - [0290]           ◦       fo1, fo2
  - [0291]           ◦       sdur1[i], sdur2[i]
  - [0292]           ◦       fo1[i], fo2[i]
  - [0293]           ◦       ssi1[i], ssi2[i]
- [0294]       새로운 미디어 프리젠테이션을 생성하기 위하여, 다음의 정보가 생성될 수도 있다:
- [0295]       •       각각의 적응 세트의 각각의 표현에 대한 제 1 주기에서의 속성들 및 엘리먼트들의 업데이트:
  - [0296]           ◦       **period@start** ps1
  - [0297]           ◦       add **period@duration** pdur1 = tsplit
  - [0298]           ◦       **SegmentBase@indexRange**
  - [0299]           ◦       **SegmentBase@presentationTimeOffset** pto11, pto12
  - [0300]           ◦       **SegmentBase@presentationDuration** pd11, pd12
- [0301]       •       각각의 표현에 대한 새로운 제 2 주기에서의 엘리먼트들 및 속성들의 업데이트 및 추가:
  - [0302]           ◦       add **period@duration** pdur2
  - [0303]           ◦       **SegmentBase@indexRange**
  - [0304]           ◦       **SegmentBase@presentationTimeOffset** pto21, pto22
  - [0305]           ◦       **SegmentBase@presentationDuration** pd21, pd22

[0306]       pto 및 pd 에 대한 상기 정보가 온-디맨드 프로파일에 대해서와 동일한 방식으로 생성될 수도 있다.       단지 차이는 세그먼트 인덱스에서의 정보를 이용하여, 단지 pto 와 pd 사이의 시간만이 이 표현에 대해 플레이될 수도 있다고 표현에서 명시적으로 말할 수도 있다는 것일 수도 있다.

[0307]       다음의 예는 도 3 을 참조하고 있다.       단일의 주기 프리젠테이션을 포함하는 CDN 상에서 온-디맨드 콘텐츠를 가리키는 MPD 가 이용가능할 수도 있다.       MPD 는 쿼들(ad 삽입 기회들) 에 기초하여, 서버 상에서 콘텐츠를 변경하지 않고 ad 다수의 주기들로 수정될 수도 있다.       각각의 기회에 대해, Ad 관정 서버에 대한 xlink 가 제공될 수도 있다.       MPD 는 DASH 클라이언트에 제공될 수도 있다.       MPD 에서의 각각의 xlink 에 대해, ad 들이 사용자에게 추가, 가능하게는 타겟팅되는지가 체크될 수도 있다.       만약 아니라면, 주기는 0 으로 레졸빙(resolving) 될 수도 있다.       만약 그렇다면, 다른 콘텐츠가 추가될 수도 있다.       DASH 클라이언트는 콘텐츠를 연속적으로 플레이하고, 0 으로 레졸빙되면, 미디어 엔진은 새롭게 초기화되지 않을 수도 있다.

[0308]       도 1 은 네트워크를 통해 미디어 데이터를 스트리밍하기 위한 기법들을 구현하는 일 예의 시스템 (10) 을 예시하는 블록 다이어그램이다.       이 예에서, 시스템 (10) 은 콘텐츠 준비 디바이스 (20), 서버 디바이스 (60), 및 클라이언트 디바이스 (40) 를 포함한다.       클라이언트 디바이스 (40) 및 서버 디바이스 (60) 는 인터넷을



포함할 수도 있는 네트워크 (74) 에 의해 통신가능하게 커플링된다. 일부 예들에서, 콘텐츠 준비 디바이스 (20) 및 서버 디바이스 (60) 는 또한, 네트워크 (74) 또는 다른 네트워크에 의해 커플링될 수도 있고, 또는 직접 통신가능하게 커플링될 수도 있다. 일부 예들에서, 콘텐츠 준비 디바이스 (20) 및 서버 디바이스 (60) 는 동일한 디바이스를 포함할 수도 있다.

[0309] 콘텐츠 준비 디바이스 (20) 는 도 1 의 예에서, 오디오 소스 (22) 및 비디오 소스 (24) 를 포함한다. 오디오 소스 (22) 는 예를 들어, 오디오 인코더 (26) 에 의해 인코딩될 캡처된 오디오 데이터를 표현하는 전기 신호들을 생성하는 마이크로폰을 포함할 수도 있다. 대안적으로, 오디오 소스 (22) 는 이전에 레코딩된 오디오 데이터를 저장하는 저장 매체, 컴퓨터화된 합성기와 같은 오디오 데이터 생성기, 또는 오디오 데이터의 임의의 다른 소스를 포함할 수도 있다. 비디오 소스 (24) 는 비디오 인코더 (28) 에 의해 인코딩될 비디오 데이터를 생성하는 비디오 카메라, 이전에 레코딩된 비디오 데이터로 인코딩된 저장 매체, 컴퓨터 그래픽스 소스와 같은 비디오 데이터 생성 유닛, 또는 비디오 데이터의 임의의 다른 소스를 포함할 수도 있다. 콘텐츠 준비 디바이스 (20) 는 반드시 모든 예들에서 서버 디바이스 (60) 에 통신가능하게 커플링되는 것은 아니고, 서버 디바이스 (60) 에 의해 관독되는 별도의 매체에 멀티미디어 콘텐츠를 저장할 수도 있다.

[0310] 원시 (raw) 오디오 및 비디오 데이터는 아날로그 또는 디지털 데이터를 포함할 수도 있다. 아날로그 데이터는 오디오 인코더 (26) 및/또는 비디오 인코더 (28) 에 의해 인코딩되기 전에 디지털화될 수도 있다. 오디오 소스 (22) 는 화자 참가자가 발화하고 있는 동안에 화자 참가자로부터 오디오 데이터를 획득할 수도 있고, 비디오 소스 (24) 는 화자 참가자의 비디오 데이터를 동시에 획득할 수도 있다. 다른 예들에서, 오디오 소스 (22) 는 저장된 오디오 데이터를 포함하는 컴퓨터 관독가능 저장 매체를 포함할 수도 있고, 비디오 소스 (24) 는 저장된 비디오 데이터를 포함하는 컴퓨터 관독가능 저장 매체를 포함할 수도 있다. 이 방식으로, 본 개시에서 설명된 기법들은 라이브, 스트리밍, 실시간 오디오 및 비디오 데이터에 또는 아카이브된, 사전-레코딩된 오디오 및 비디오 데이터에 적용될 수도 있다.

[0311] 비디오 프레임들에 대응하는 오디오 프레임들은 일반적으로 비디오 프레임들 내에 포함되는 비디오 소스 (24) 에 의해 캡처된 (또는 생성된) 비디오 데이터와 동시 발생적으로 오디오 소스 (22) 에 의해 캡처 (또는 생성) 되었던 오디오 데이터를 포함하는 오디오 프레임들이다. 예를 들어, 화자 참가자가 일반적으로 발화함으로써 오디오 데이터를 생성하지만, 오디오 소스 (22) 는 오디오 데이터를 캡처하고, 비디오 소스 (24) 는 동시에, 즉, 오디오 소스 (22) 가 오디오 데이터를 캡처하고 있는 동안에 화자 참가자의 비디오 데이터를 캡처한다. 이런 이유로, 오디오 프레임은 하나 이상의 특정한 비디오 프레임들에 시간적으로 대응할 수도 있다. 이에 따라, 비디오 프레임에 대응하는 오디오 프레임은 일반적으로, 오디오 데이터 및 비디오 데이터가 동시에 캡처되었고 오디오 프레임 및 비디오 프레임이 동시에 캡처되었던 오디오 데이터 및 비디오 데이터를 각각 포함하는 상황에 대응한다.

[0312] 일부 예들에서, 오디오 인코더 (26) 는 인코딩된 오디오 프레임에 대한 오디오 데이터가 레코딩되었던 시간을 표현하는 각각의 인코딩된 오디오 프레임에서의 타임스탬프를 인코딩하고, 유사하게, 비디오 인코더 (28) 는 인코딩된 비디오 프레임에 대한 비디오 데이터가 레코딩되었던 시간을 표현하는 각각의 인코딩된 비디오 프레임에서의 타임스탬프를 인코딩할 수도 있다. 이러한 예들에서, 비디오 프레임에 대응하는 오디오 프레임은 타임스탬프를 포함하는 오디오 프레임 및 동일한 타임스탬프를 포함하는 비디오 프레임을 포함할 수도 있다. 콘텐츠 준비 디바이스 (20) 는 오디오 인코더 (26) 및/또는 비디오 인코더 (28) 가 타임스탬프들을 생성할 수도 있고, 또는 오디오 소스 (22) 및 비디오 소스 (24) 가 오디오 및 비디오 데이터를 각각 타임스탬프와 연관시키기 위해 이용할 수도 있는 내부 클록을 포함할 수도 있다.

[0313] 일부 예들에서, 오디오 소스 (22) 는 오디오 데이터가 레코딩되었던 시간에 대응하는 데이터를 오디오 인코더 (26) 로 전송할 수도 있고, 비디오 소스 (24) 는 비디오 데이터가 레코딩되었던 시간에 대응하는 데이터를 비디오 인코더 (28) 로 전송할 수도 있다. 일부 예들에서, 오디오 인코더 (26) 는 인코딩된 오디오 데이터의 상대적 시간 순서를 나타내기 위해 (그러나 오디오 데이터가 레코딩되었던 절대적 시간을 반드시 나타내는 것은 아님) 인코딩된 오디오 데이터에서의 시퀀스 식별자를 인코딩할 수도 있고, 그리고 유사하게, 비디오 인코더 (28) 는 인코딩된 비디오 데이터의 상대적 시간 순서를 나타내기 위해 시퀀스 식별자들을 또한 이용할 수도 있다. 유사하게, 일부 예들에서, 시퀀스 식별자는 타임스탬프와 맵핑되거나 또는 다르게는 타임스탬프와 상관될 수도 있다.

[0314] 오디오 인코더 (26) 는 일반적으로 인코딩된 오디오 데이터의 스트림을 생성하는 한편, 비디오 인코더 (28) 는 인코딩된 비디오 데이터의 스트림을 생성한다. 데이터의 각각의 개개의 스트림 (오디오든 비디오든) 은 기

본 스트림으로 지칭될 수도 있다. 기본 스트림은 표현의 단일의, 디지털로 코딩된 (가능하게는 압축된) 컴포넌트이다. 예를 들어, 표현의 코딩된 비디오 또는 오디오 부분은 기본 스트림일 수 있다. 기본 스트림은 비디오 파일 내에 캡슐화되기 이전에 패킷화된 기본 스트림 (packetized elementary stream; PES) 으로 변환될 수도 있다. 동일한 표현 내에서, 스트림 ID 는 하나의 기본 스트림에 속하는 PES-패킷들을 다른 것과 구별하는데 이용될 수도 있다. 기본 스트림의 데이터의 기본적인 유닛은 패킷화된 기본 스트림 (PES) 패킷이다. 따라서, 코딩된 비디오 데이터는 일반적으로 기본 비디오 스트림들에 대응한다. 유사하게, 오디오 데이터는 하나 이상의 각각의 기본 스트림들에 대응한다.

[0315] 많은 비디오 코딩 표준들, 이를 테면 ITU-T H.264/AVC 및 다가오는 고효율 비디오 코딩 (HEVC) 표준은 에러 없는 (error-free) 비트스트림들에 대한 신택스, 시맨틱스, 및 디코딩 프로세스를 정의하고, 이들 중 임의의 것은 소정의 프로파일 또는 레벨에 따른다. 비디오 코딩 표준들은 통상적으로 인코더를 특정하지 않지만, 그 인코더는 생성된 비트스트림들이 디코더에 대해 표준-순응적임을 보장하는 업무를 담당한다. 비디오 코딩 표준들의 맥락에서, "프로파일" 은 알고리즘들, 피처들, 또는 툴들 및 그들에 적용하는 제약들의 서브세트에 대응한다. 예를 들어, H.264 표준에 의해 정의한 바와 같이, "프로파일" 은 H.264 표준에 의해 특정되는 전체 비트스트림 신택스의 서브세트이다. "레벨" 은 픽처들의 레졸루션, 비트 레이트, 및 블록 프로세싱 레이트에 관련되는, 예를 들어, 디코더 메모리 및 컴퓨테이션과 같은, 디코더 리소스 컴퓨테이션의 제한들에 대응한다. 프로파일은 profile\_idc (프로파일 표시자) 값으로 시그널링될 수도 있는 한편, 레벨은 level\_idc (레벨 표시자) 값으로 시그널링될 수도 있다.

[0316] H.264 표준은 예를 들어, 주어진 프로파일의 신택스에 의해 부과된 한계 내에서, 디코딩된 픽처들의 특정된 사이즈와 같은 비트스트림에서의 신택스 엘리먼트들에 의해 취해진 값들에 의존하여 인코더들 및 디코더들의 성능의 큰 변동을 요구하는 것이 여전히 가능하다는 것을 인지한다. H.264 표준은 또한, 많은 애플리케이션들에서, 특정된 프로파일 내에서 신택스의 모든 가정적인 사용들을 다루는 것이 가능한 디코더를 구현하는 것은 현실적이지도 경제적이지도 않다는 것을 인지한다. 이에 따라, H.264 표준은 비트스트림에서의 신택스 엘리먼트들의 값들에 부과된 제약들의 특정된 세트로서 "레벨" 을 정의한다. 이들 제약들은 값들에 대한 단순한 제한들일 수도 있다. 대안적으로, 이들 제약들은 값들의 산술적 조합들 (예를 들어, 픽처 폭 곱하기 픽처 높이 곱하기 초당 디코딩된 픽처들의 수) 에 대한 제약들의 형태를 취할 수도 있다. H.264 표준은 또한, 개개의 구현들이 각각의 지원된 프로파일에 대해 상이한 레벨을 지원할 수도 있다는 것을 규정한다.

[0317] 프로파일에 따르는 디코더는 프로파일에서 정의된 모든 피처들을 보통 지원한다. 예를 들어, 코딩 피처로서, B-픽처 코딩은 H.264/AVC 의 베이스라인 프로파일에서는 지원되지 않고 H.264/AVC 의 다른 프로파일들에서는 지원된다. 레벨에 따르는 디코더는 레벨에서 정의된 제한들을 넘는 리소스들을 요구하지 않는 임의의 비트스트림을 디코딩하는 것이 가능할 수 있다. 프로파일들 및 레벨들의 정의들은 해석가능성을 위해 도움이 될 수도 있다. 예를 들어, 비디오 송신 동안에, 한 쌍의 프로파일 및 레벨 정의들은 전체 송신 세션에 대해 협상되고 합의가 될 수도 있다. 보다 구체적으로, H.264/AVC 에서, 레벨은 프로세싱될 필요가 있는 매크로블록들의 수, 디코딩된 픽처 버퍼 (DPB) 사이즈, 코딩된 픽처 버퍼 (CPB) 사이즈, 수직 모션 벡터 범위, 2 개의 연속적인 MB들 당의 모션 벡터들의 최대 수, 및 B-블록이 8x8 픽셀들 미만의 서브-매크로블록 파티션들을 가질 수 있는 여부에 대한 제한들을 정의할 수도 있다. 이 방식으로, 디코더는 디코더가 비트스트림을 적절히 디코딩하는 것이 가능한지 여부를 결정할 수도 있다.

[0318] 도 1 의 예에서, 콘텐츠 준비 디바이스 (20) 의 캡슐화 유닛 (30) 은 비디오 인코더 (28) 로부터의 코딩된 비디오 데이터를 포함하는 기본 스트림들 및 오디오 인코더 (26) 로부터의 코딩된 오디오 데이터를 포함하는 기본 스트림들을 수신한다. 일부 예들에서, 비디오 인코더 (28) 및 오디오 인코더 (26) 는 인코딩된 데이터로부터 PES 패킷들을 형성하기 위한 패킷화기 (packetizer) 들을 각각 포함할 수도 있다. 다른 예들에서, 비디오 인코더 (28) 및 오디오 인코더 (26) 는 인코딩된 데이터로부터 PES 패킷들을 형성하기 위한 각각의 패킷화기들과 각각 인터페이스할 수도 있다. 여전히 다른 예들에서, 캡슐화 유닛 (30) 은 인코딩된 오디오 및 비디오 데이터로부터 PES 패킷들을 형성하기 위한 패킷화기들을 포함할 수도 있다.

[0319] 비디오 인코더 (28) 는 다양한 비트레이트들에서 그리고 다양한 특성들, 이를 테면 픽셀 레졸루션들, 프레임 레이트들, 다양한 코딩 표준들에 대한 순응, 다양한 코딩 표준들에 대한 다양한 프로파일들 및/또는 프로파일들의 레벨들에 대한 순응, (예를 들어, 2 차원 또는 3 차원 재생을 위한) 하나 또는 다수의 뷰들을 갖는 표현들, 또는 다른 이러한 특성들로 멀티미디어 콘텐츠의 상이한 표현들을 생성하기 위해, 다양한 방식으로 멀티미디어 콘텐츠의 비디오 데이터를 인코딩할 수도 있다. 표현은 본 개시에서 사용한 바와 같이, 오디오 데이터, 비디오 데이터, 텍스트 데이터 (예를 들어, 폐쇄 자막을 위한), 또는 다른 이러한 데이터 중 하나를 포함할 수도 있

다. 표현은 기본 스트림, 이를 태면 오디오 기본 스트림 또는 비디오 기본 스트림을 포함할 수도 있다. 각각의 PES 패킷은 PES 패킷이 속하는 기본 스트림을 식별하는 stream\_id 를 포함할 수도 있다. 캡슐화 유닛 (30) 은 기본 스트림들을 다양한 표현들의 비디오 파일들 (예를 들어, 세그먼트들) 에 어셈블리하는 것을 담당한다.

[0320] 캡슐화 유닛 (30) 은 오디오 인코더 (26) 및 비디오 인코더 (28) 로부터의 표현의 기본 스트림들에 대한 PES 패킷들을 수신하고 PES 패킷들로부터 대응하는 네트워크 추상화 계층 (NAL) 유닛들을 형성한다. H.264/AVC (Advanced Video Coding) 의 예에서, 코딩된 비디오 세그먼트들은 비디오 텔레포니, 스토리지, 브로드캐스트, 또는 스트리밍과 같은 애플리케이션들을 어드레싱하는 "네트워크-친화형" 비디오 표현을 제공하는 NAL 유닛들로 편성된다. NAL 유닛들은 비디오 코딩 계층 (VCL) NAL 유닛들 및 비-VCL NAL 유닛들로 카테고리화될 수 있다. VCL 유닛들은 코어 압축 엔진을 포함할 수도 있고 블록, 매크로블록, 및/또는 슬라이스 레벨 데이터를 포함할 수도 있다. 다른 NAL 유닛들은 비-VCL NAL 유닛들일 수도 있다. 일부 예들에서, 보통 프라이머리 코딩된 픽처로서 제시된, 하나의 시간 인스턴스에서의 코딩된 픽처는 하나 이상의 NAL 유닛들을 포함할 수도 있는 액세스 유닛에 포함될 수도 있다.

[0321] 비-VCL NAL 유닛들은 다른 것들 중에서도, 파라미터 세트 NAL 유닛들 및 SEI NAL 유닛들을 포함할 수도 있다. 파라미터 세트들은 (시퀀스 파라미터 세트들 (SPS) 에서의) 시퀀스-레벨 헤더 정보 및 (픽처 파라미터 세트들 (PPS) 에서의) 드물게 변화하는 픽처-레벨 헤더 정보를 포함할 수도 있다. 파라미터 세트들 (예를 들어, PPS 및 SPS) 의 경우, 드물게 변화하는 정보는 각각의 시퀀스 또는 픽처에 대해 반복될 필요가 없고, 이런 이유로 코딩 효율은 개선될 수도 있다. 더욱이, 파라미터 세트들의 사용은 중요한 헤더 정보의 대역의 송신을 가능하게 하여, 에러 내성을 위한 중복 송신들의 필요성을 회피할 수도 있다. 대역의 송신 예들에서, 파라미터 세트 NAL 유닛들은 SEI NAL 유닛들과 같은 다른 NAL 유닛들과는 상이한 채널을 통해 송신될 수도 있다.

[0322] 보충 강화 정보 (Supplemental Enhancement Information; SEI) 는 VCL NAL 유닛들로부터의 코딩된 픽처들 샘플들을 디코딩하는데 필요하지 않은 정보를 포함할 수도 있지만, 디코딩, 디스플레이, 에러 내성, 및 다른 목적들에 관련된 프로세스들을 보조할 수도 있다. SEI 메시지들은 비-VCL NAL 유닛들에 포함될 수도 있다. SEI 메시지들은 일부 표준 사양들의 규범적인 부분이고, 따라서 표준 순응 디코더 구현에 대해 항상 의무적인 것은 아니다. SEI 메시지들은 시퀀스 레벨 SEI 메시지들 또는 픽처 레벨 SEI 메시지들일 수도 있다. 일부 시퀀스 레벨 정보는 SVC 의 예에서의 스케일러빌리티 정보 SEI 메시지들 및 MVC 에서의 뷰 스케일러빌리티 정보 SEI 메시지들과 같은 SEI 메시지들에 포함될 수도 있다. 이들 예의 SEI 메시지들은 예를 들어, 동작 포인트들의 추출 및 그 동작 포인트들의 특성들에 대한 정보를 운반할 수도 있다. 또한, 캡슐화 유닛 (30) 은 표현들의 특성들을 기술하는 미디어 프리젠테이션 디스크립터 (MPD) 와 같은 매니페스트 파일을 형성할 수도 있다. 캡슐화 유닛 (30) 은 XML (extensible markup language) 에 따라 MPD 를 포맷팅할 수도 있다.

[0323] 캡슐화 유닛 (30) 은 멀티미디어 콘텐츠의 하나 이상의 표현들에 대한 데이터를 매니페스트 파일 (예를 들어, MPD) 과 함께 출력 인터페이스 (32) 에 제공할 수도 있다. 출력 인터페이스 (32) 는 네트워크 인터페이스 또는 저장 매체에 기입하기 위한 인터페이스, 이를 태면 범용 직렬 버스 (USB) 인터페이스, CD 또는 DVD 라이터 또는 버너, 자기 또는 플래시 저장 매체들에 대한 인터페이스, 또는 미디어 데이터를 저장 또는 송신하기 위한 다른 인터페이스들을 포함할 수도 있다. 캡슐화 유닛 (30) 은 멀티미디어 콘텐츠의 표현들의 각각의 데이터를 출력 인터페이스 (32) 에 제공할 수도 있고, 그 출력 인터페이스 (32) 는 그 데이터를 네트워크 송신 또는 저장 매체들을 통해 서버 디바이스 (60) 로 전송할 수도 있다. 도 1 의 예에서, 서버 디바이스 (60) 는 각각의 매니페스트 파일 (66) 및 하나 이상의 표현들 (68A 내지 68N) (표현들 (68)) 을 각각 포함하는 다양한 멀티미디어 콘텐츠들 (64) 을 저장하는 저장 매체 (62) 를 포함한다. 일부 예들에서, 출력 인터페이스 (32) 는 또한, 네트워크 (74) 로 직접 데이터를 전송할 수도 있다.

[0324] 일부 예들에서, 표현들 (68) 은 적응 세트들로 분리될 수도 있다. 즉, 표현들 (68) 의 다양한 서브세트들은 코덱, 프로파일 및 레벨, 레졸루션, 뷰들의 수, 세그먼트들에 대한 파일 포맷, 예를 들어, 스피커들에 의해, 디코딩 및 제시될 오디오 데이터 및/또는 표현으로 디스플레이될 텍스트의 언어 또는 다른 특성들을 식별할 수도 있는 텍스트 타입 정보, 적응 세트에서의 표현들에 대한 장면의 카메라 각도 또는 실세계 카메라 관점을 기술할 수도 있는 카메라 각도 정보, 특정한 청중들에 대한 콘텐츠 적합성을 기술하는 레이팅 정보 등과 같은 특성들의 각각의 공통 세트들을 포함할 수도 있다.

[0325] 매니페스트 파일 (66) 은 특정한 적응 세트들에 대응하는 표현들 (68) 의 서브세트들을 표시하는 데이터 뿐만 아니라 적응 세트들에 대한 공통 특성들을 포함할 수도 있다. 매니페스트 파일 (66) 은 또한, 적응 세트들

의 개개의 표현들에 대한, 비트레이트들과 같은 개개의 특성들을 표현하는 데이터를 포함할 수도 있다. 이 방식으로, 적응 세트는 단순화된 네트워크 대역폭 적응을 허용할 수도 있다. 적응 세트에서의 표현들은 매니페스트 파일 (66) 의 적응 세트 엘리먼트의 지식 엘리먼트들을 이용하여 나타내질 수도 있다. 본 개시의 기법들에 따르면, 서버 디바이스 (60) 는 멀티미디어 콘텐츠 (64) 가 광고 콘텐츠가 삽입될 수도 있는 2 개 이상의 연속적인 주기들을 포함한다는 것을 나타내기 위해 매니페스트 파일 (66) 을 수정할 수도 있다.

[0326] 특히, 서버 디바이스 (60) 는 추가적인 콘텐츠 (이를 테면 광고 콘텐츠) 가 연속적인 주기들 사이에 삽입될 수 있도록, 멀티미디어 콘텐츠 (64) 에 있어서 연속적인 주기들이 존재한다는 것을 나타내기 위해, 상기 설명된 다양한 기법들, 또는 유사한 기법들 중 임의의 것을 이용하여 매니페스트 파일 (66) 을 구성할 수도 있다. 예를 들어, 서버 디바이스 (60) 는 2 개의 연속적인 주기들로 스플리팅되는 각각의 주기에 대해 매니페스트 파일 (66) 에 Period@duration 속성을 추가할 수도 있다. 마찬가지로, 서버 디바이스 (60) 는 Period@duration 엘리먼트들, baseURL 엘리먼트들, SegmentBase@indexRange 엘리먼트들, SegmentBase@presentationTimeOffset 엘리먼트들, 및 SegmentBase@presentationDuration 엘리먼트들과 같은 새로운 연속적인 주기를 표시하는 데이터를 매니페스트 파일 (66) 에 추가할 수도 있다. 더욱이, 서버 디바이스 (60) 는 상기 논의된 다양한 세그먼트 인덱스 엘리먼트들을 시그널링하고, 기존의 표현들로부터 새로운 주기에 대한 새로운 표현들을 생성할 수도 있다.

[0327] 이에 따라, 일부 예들에서, 서버 디바이스 (60) 는 제 1 주기 및 제 2 주기를 포함하는 복수의 주기들로 미디어 데이터의 메인 콘텐츠를 스플리팅할 수도 있고, 여기서 제 1 주기 및 제 2 주기는 시간적으로 순차적이다. 즉, 제 1 및 제 2 주기들은 연속적으로 플레이 아웃되도록 의도되는 메인 콘텐츠에 대한 데이터를 포함한다. 서버 디바이스 (60) 는 제 1 주기와 제 2 주기 사이에 세컨더리 미디어 콘텐츠 (예를 들어, 광고 미디어 콘텐츠) 를 삽입할 수도 있다. 더욱이, 서버 디바이스 (60) 는 복수의 주기들의 각각에 대한 하나 이상의 시간 특성들을 정의하는 복수의 주기들의 각각에 대한 시간 표시자를 시그널링할 수도 있다. 시간 표시자는 주기들에 대해 매니페스트 파일 (66) 에 예를 들어 Period@duration 속성들을 포함할 수도 있다. 서버 디바이스 (60) 는 그 후, 주기들, 세컨더리 미디어 콘텐츠, 및 시간 표시자들을 포함하는 미디어 스트림을 추출하기 위한 예를 들어 클라이언트 디바이스 (40) 로부터의 요청을 수신할 수도 있다. 요청은 클라이언트 디바이스 (40) 로부터 미디어 스트림의 데이터를 추출하기 위한 복수의 요청들 중 하나일 수도 있다. 예를 들어, 복수의 요청들은 HTTP GET 또는 부분 GET 요청들일 수도 있다.

[0328] 상기의 기법들은 서버 디바이스 (60) 에 대하여 설명되지만, 콘텐츠 준비 디바이스 (20) 는 또한 (즉, 추가로 또는 대안으로) 2 개의 연속적인 주기들로 주기를 스플리팅하고 그리고 연속적인 주기들 사이에 광고 미디어 데이터를 삽입하도록 구성될 수도 있다는 것이 이해되어야 한다.

[0329] 서버 디바이스 (60) 는 요청 프로세싱 유닛 (70) 및 네트워크 인터페이스 (72) 를 포함한다. 일부 예들에서, 서버 디바이스 (60) 는 복수의 네트워크 인터페이스들을 포함할 수도 있다. 더욱이, 서버 디바이스 (60) 의 특징들 중 임의의 것 또는 전부는 라우터들, 브릿지들, 프록시 디바이스들, 스위치들, 또는 다른 디바이스들과 같은 콘텐츠 전달 네트워크의 다른 디바이스들 상에 구현될 수도 있다. 일부 예들에서, 콘텐츠 전달 네트워크의 중간 디바이스들은 멀티미디어 콘텐츠 (64) 의 데이터를 캐싱하고, 서버 디바이스 (60) 의 컴포넌트들에 실질적으로 따르는 컴포넌트들을 포함할 수도 있다. 일반적으로, 네트워크 인터페이스 (72) 는 네트워크 (74) 를 통해 데이터를 전송 및 수신하도록 구성된다.

[0330] 요청 프로세싱 유닛 (70) 은 저장 매체 (62) 의 데이터에 대해, 클라이언트 디바이스들, 이를 테면 클라이언트 디바이스 (40) 로부터 네트워크 요청들을 수신하도록 구성된다. 예를 들어, 요청 프로세싱 유닛 (70) 은 RFC 2616, "Hypertext Transfer Protocol - HTTP/1.1", by R. Fielding et al, Network Working Group, IETF, June 1999 에서 설명한 바와 같이, 하이퍼텍스트 전송 프로토콜 (HTTP) 버전 1.1 을 구현할 수도 있다. 즉, 요청 프로세싱 유닛 (70) 은 HTTP GET 또는 부분 GET 요청들을 수신하고 요청들에 응답하여 멀티미디어 콘텐츠 (64) 의 데이터를 제공하도록 구성될 수도 있다. 요청들은 예를 들어, 세그먼트의 URL 을 이용하여 표현들 (68) 중 하나의 세그먼트를 특정할 수도 있다. 일부 예들에서, 요청들은 또한, 세그먼트의 하나 이상의 바이트 범위들을 특정하여, 부분 GET 요청들을 포함할 수도 있다. 요청 프로세싱 유닛 (70) 은 또한, 표현들 (68) 중 하나의 세그먼트의 헤더 데이터를 제공하기 위해 HTTP HEAD 요청들을 서비스하도록 구성될 수도 있다. 어느 경우나, 요청 프로세싱 유닛 (70) 은 클라이언트 디바이스 (40) 와 같은 요청 디바이스에 요청된 데이터를 제공하기 위한 요청들을 프로세싱하도록 구성될 수도 있다.

[0331] 추가적으로 또는 대안적으로, 요청 프로세싱 유닛 (70) 은 eMBMS 와 같은 브로드캐스트 또는 멀티캐스트 프로토



콜을 통해 미디어 데이터를 전달하도록 구성될 수도 있다. 콘텐츠 준비 디바이스 (20) 는 설명한 것과 실질적으로 동일한 방식으로 DASH 세그먼트들 및/또는 서브-세그먼트들을 생성할 수도 있지만, 서버 디바이스 (60) 는 eMBMS 또는 다른 브로드캐스트 또는 멀티캐스트 네트워크 전송 프로토콜을 이용하여 이들 세그먼트들 또는 서브-세그먼트들을 전달할 수도 있다. 예를 들어, 요청 프로세싱 유닛 (70) 은 클라이언트 디바이스 (40) 로부터 멀티캐스트 그룹 조인 요청을 수신하도록 구성될 수도 있다. 즉, 서버 디바이스 (60) 는 특정한 미디어 콘텐츠 (예를 들어, 라이브 이벤트의 브로드캐스트) 와 연관된, 클라이언트 디바이스 (40) 를 포함하는 클라이언트 디바이스들에 멀티캐스트 그룹과 연관된 인터넷 프로토콜 (IP) 어드레스를 광고할 수도 있다. 클라이언트 디바이스 (40) 는 차례로, 멀티캐스트 그룹을 조인하기 위한 요청을 제출할 수도 있다. 이 요청은 라우터들이 멀티캐스트 그룹과 연관된 IP 어드레스를 목적지로 한 트래픽을 클라이언트 디바이스 (40) 와 같은 가입 클라이언트 디바이스들로 지향하도록 야기되는 네트워크 (74), 예를 들어, 네트워크 (74) 를 구성하는 라우터들 전체에 전파될 수도 있다.

[0332] 도 1 의 예에서 예시한 바와 같이, 멀티미디어 콘텐츠 (64) 는 미디어 프리젠테이션 디스크립션 (MPD) 에 대응할 수도 있는 매니페스트 파일 (66) 을 포함한다. 매니페스트 파일 (66) 은 상이한 대안적 표현들 (68) (예를 들어, 상이한 품질들을 가진 비디오 서비스들) 의 디스크립션들을 포함할 수도 있고 디스크립션은 예를 들어, 표현들 (68) 의 코덱 정보, 프로파일 값, 레벨 값, 비트레이트, 및 다른 기술적 특성들을 포함할 수도 있다. 클라이언트 디바이스 (40) 는 표현들 (68) 의 세그먼트들에 액세스하는 방법을 결정하기 위해 미디어 프리젠테이션의 MPD 를 추출할 수도 있다.

[0333] 특히, 추출 유닛 (52) (이는 본 개시의 기법들을 구현할 수도 있음) 은 비디오 디코더 (48) 의 디코딩 능력들 및 비디오 출력 (44) 의 렌더링 능력들을 결정하기 위해 클라이언트 디바이스 (40) 의 구성 데이터 (미도시) 를 추출할 수도 있다. 구성 데이터는 또한, 클라이언트 디바이스 (40) 의 사용자에게 의해 선택된 언어 선호도, 클라이언트 디바이스 (40) 의 사용자에게 의해 설정된 깊이 선호도들에 대응하는 하나 이상의 카메라 관점들, 및/또는 클라이언트 디바이스 (40) 의 사용자에게 의해 선택된 레이팅 선호도 중 임의의 것 또는 전부를 포함할 수도 있다. 추출 유닛 (52) 은 예를 들어, HTTP GET 및 부분 GET 요청들을 제출하도록 구성된 웹 브라우저 또는 미디어 클라이언트를 포함할 수도 있다. 추출 유닛 (52) 은 클라이언트 디바이스 (40) 의 하나 이상의 프로세서들 또는 프로세싱 유닛들 (미도시) 에 의해 실행된 소프트웨어 명령들에 대응할 수도 있다. 일부 예들에서, 추출 유닛 (52) 에 대하여 설명된 기능성의 전부 또는 부분들은 하드웨어, 또는 하드웨어, 소프트웨어, 및/또는 펌웨어의 조합으로 구현될 수도 있고, 여기서 필수 하드웨어는 소프트웨어 또는 펌웨어에 대한 명령들을 실행하기 위해 제공될 수도 있다.

[0334] 추출 유닛 (52) 은 클라이언트 디바이스 (40) 의 디코딩 및 렌더링 능력들과 매니페스트 파일 (66) 의 정보에 의해 나타난 표현들 (68) 의 특성들을 비교할 수도 있다. 추출 유닛 (52) 은 표현들 (68) 의 특성들을 결정하기 위해 매니페스트 파일 (66) 의 적어도 부분을 초기에 추출할 수도 있다. 예를 들어, 추출 유닛 (52) 은 하나 이상의 적응 세트들의 특성들을 기술하는 매니페스트 파일 (66) 의 부분을 요청할 수도 있다. 추출 유닛 (52) 은 클라이언트 디바이스 (40) 의 코딩 및 렌더링 능력들에 의해 충족될 수 있는 특성들을 갖는 표현들 (68) 의 서브세트 (예를 들어, 적응 세트) 를 선택할 수도 있다. 추출 유닛 (52) 은 적응 세트에서의 표현들에 대한 비트레이트들을 결정하고, 현재 이용가능한 네트워크 대역폭의 양을 결정하고, 그리고 네트워크 대역폭에 의해 충족될 수 있는 비트레이트를 갖는 표현들 중 하나로부터 세그먼트들을 추출할 수도 있다.

[0335] 일반적으로, 더 높은 비트레이트 표현들은 더 높은 품질의 비디오 재생을 산출할 수도 있지만, 더 낮은 비트레이트 표현들은 이용가능한 네트워크 대역폭이 감소할 때 충분한 품질의 비디오 재생을 제공할 수도 있다. 이에 따라, 이용가능한 네트워크 대역폭이 상대적으로 높을 때, 추출 유닛 (52) 은 상대적으로 높은 비트레이트 표현들로부터 데이터를 추출할 수도 있는 반면, 이용가능한 네트워크 대역폭이 낮을 때, 추출 유닛 (52) 은 상대적으로 낮은 비트레이트 표현들로부터 데이터를 추출할 수도 있다. 이 방식으로, 클라이언트 디바이스 (40) 는 네트워크 (74) 를 통해 멀티미디어 데이터를 스트리밍할 수도 있는 동시에 네트워크 (74) 의 네트워크 대역폭 이용가능성의 변화에도 또한 적응할 수도 있다.

[0336] 추가적으로, 또는 대안적으로, 추출 유닛 (52) 은 eMBMS 또는 IP 멀티캐스트와 같은 브로드캐스트 또는 멀티캐스트 네트워크 프로토콜에 따라 데이터를 수신하도록 구성될 수도 있다. 이러한 예들에서, 추출 유닛 (52) 은 특정한 미디어 콘텐츠와 연관된 멀티캐스트 네트워크 그룹을 조인하기 위한 요청을 제출할 수도 있다. 멀티캐스트 그룹을 조인한 후에, 추출 유닛 (52) 은 서버 디바이스 (60) 또는 콘텐츠 준비 디바이스 (20) 에 이 슈된 추가 요청들 없이 멀티캐스트 그룹의 데이터를 수신할 수도 있다. 추출 유닛 (52) 은 예를 들어, 재생을 정지하거나 또는 상이한 멀티캐스트 그룹에의 채널들을 변화시키기 위해, 멀티캐스트 그룹의 데이터가 더 이

상 필요하지 않은 경우에 멀티캐스트 그룹을 떠나기 위한 요청을 제출할 수도 있다.

[0337] 더욱이, 본 개시의 기법들에 따르면, 추출 유닛 (52) 은 멀티미디어 콘텐츠 (64) 의 2 개 이상의 주기들이 연속적이라는 것을 매니페스트 파일 (66) 이 나타내는지 여부를 결정하도록 구성될 수도 있다. 예를 들어, 추출 유닛 (52) 은 2 개의 주기들의 2 개의 적응 세트들이 적응 세트들이 연관되는 경우 주기-연속적이고, 매니페스트 파일 (66) 이 @presentationTimeOffset 엘리먼트를 포함하거나, 또는 적응 세트들에서의 모든 표현들에 대해, 0 으로서 추론될 수 있고, 하나의 적응 세트 내에서 @presentationTimeOffset 의 값이 모든 표현들에 대해 동일하고, @presentationTimeOffset 의 값과 하나의 적응 세트에서의 모든 표현들의 프리젠테이션 지속기간의 합이 다른 적응 세트의 @presentationTimeOffset 의 값과 동일하다고 결정할 수도 있다.

[0338] 더욱이, 추출 유닛 (52) 은 하나의 주기의 하나의 적응 세트에서의 하나의 표현의 초기화 세그먼트를 다른 주기의 다른 적응 세트의 표현에 대한 초기화 세그먼트로서 이용할 수도 있고, 여기서 적응 세트들은 표현들이 @id 에 대해 동일한 값을 갖는다면 주기-연속적이다. 마찬가지로, 2 개의 적응 세트들이 주기-연속적이고 추후의 적응 세트가 @schemeIDURI="urn:mpeg:dash:period-switchable" 을 가진 연관된 보충 디스크립터를 갖는 경우, 추출 유닛 (52) 은 적응 세트들 중 하나에서의 표현들이 다른 적응 세트의 초기화 세그먼트 없이 다른 적응 세트로부터의 임의의 표현과 연결될 수 있다고 결정할 수도 있다. 이 방식으로, 추출 유닛 (52) 은 제 1 및 제 2 주기들이 연속적이라는 것을 가정하여, 제 1 주기의 적응 세트의 선택에 기초하여 (예를 들어, 제 2 주기의 적응 세트의 특성들을 추가로 분석하지 않고도) 제 2 주기의 적응 세트를 선택할 수도 있다.

[0339] 이 방식으로, 추출 유닛 (52) 은 미디어 콘텐츠에 대한 매니페스트 파일이 미디어 콘텐츠가 제 1 주기 및 제 2 주기를 포함한다는 것을 나타내고, 매니페스트 파일이 제 1 및 제 2 주기들이 연속적이라는 것을 나타내고, 그리고 광고 미디어 데이터가 제 1 주기와 제 2 주기 사이의 삽입을 위해 이용가능하다고 결정하도록 구성될 수도 있다. 추출 유닛 (52) 은 또한, 제 1 주기에서의 제 1 적응 세트를, 제 1 적응 세트에 대해 시그널링된 특성들에 적어도 부분적으로 기초하여 선택하도록 구성될 수도 있다. 이러한 특성들은 예를 들어, 코덱, 프로파일, 레벨, 픽처 디멘전들, 프레임 레이트 등과 같은 코딩 및 렌더링 특성들을 포함할 수도 있다. 적응 세트를 선택하기 위해, 추출 유닛 (52) 은 클라이언트 디바이스 (40) 의 코딩 및 렌더링 능력들과 다양한 이용가능한 적응 세트들의 특성들을 비교하고, 클라이언트 디바이스 (40) 에 의해 코딩 및 렌더링될 수 있는 적응 세트를 선택할 수도 있다.

[0340] 추출 유닛 (52) 은 그 후 제 1 적응 세트의 선택에 기초하여 제 1 적응 세트의 미디어 데이터를 추출할 수도 있다. 특히, 추출 유닛 (52) 은 표현에 대한 비트레이트가 이용가능한 네트워크 대역폭을 초과하지 않도록, 예를 들어, 적응 세트의 표현들에 대한 비트레이트들 및 이용가능한 네트워크 대역폭에 기초하여, (하나보다 더 많은 표현이 이용가능하다면) 적응 세트의 표현을 선택할 수도 있다. 추출 유닛 (52) 은 또한 광고 미디어 데이터를 추출할 수도 있다. 더욱이, 추출 유닛 (52) 은 제 1 적응 세트의 선택에 기초하여 제 1 적응 세트와 연관되는 제 2 주기에서의 제 2 적응 세트의 미디어 데이터를 추출할 수도 있다. 따라서, 추출 유닛 (52) 은 제 2 적응 세트가 제 1 적응 세트와 주기-연속적이라고 단순히 결정하고, 제 1 적응 세트에 대해 행한 대로 제 2 주기의 적응 세트들의 특성들을 분석하기 보다는, 제 2 적응 세트는 이미 선택되었던 제 1 적응 세트와 주기-연속적이기 때문에 제 2 적응 세트를 단순히 선택할 수도 있다.

[0341] 네트워크 인터페이스 (54) 는 선택된 표현의 세그먼트들의 데이터를 수신하여 추출 유닛 (52) 에 제공할 수도 있고, 그 추출 유닛 (52) 은 차례로 그 세그먼트들을 캡슐화해제 유닛 (50) 에 제공할 수도 있다. 캡슐화해제 유닛 (50) 은 비디오 파일의 엘리먼트들은 구성 PES 스트림들로 캡슐화해제하고, 인코딩된 데이터를 추출하기 위해 PES 스트림들을 패킷화해제하고, 그리고 예를 들어, 스트림의 PES 패킷 헤더들에 의해 나타낸 바와 같이, 인코딩된 데이터가 오디오 또는 비디오 스트림의 부분인지 여부에 의존하여, 인코딩된 데이터를 오디오 디코더 (46) 또는 비디오 디코더 (48) 중 어느 하나로 전송할 수도 있다. 오디오 디코더 (46) 는 인코딩된 오디오 데이터를 디코딩하고 디코딩된 오디오 데이터를 오디오 출력 (42) 으로 전송하는 한편, 비디오 디코더 (48) 는 인코딩된 비디오 데이터를 디코딩하고 스트림의 복수의 뷰들을 포함할 수도 있는 디코딩된 비디오 데이터를 비디오 출력 (44) 으로 전송한다.

[0342] 비디오 인코더 (28), 비디오 디코더 (48), 오디오 인코더 (26), 오디오 디코더 (46), 캡슐화 유닛 (30), 추출 유닛 (52), 및 캡슐화해제 유닛 (50) 각각은 적용가능한 대로, 하나 이상의 마이크로프로세서들, 디지털 신호 프로세서들 (DSP들), 주문형 집적 회로들 (ASIC들), 필드 프로그래밍가능 게이트 어레이들 (FPGA들), 이산 로직 회로, 소프트웨어, 하드웨어, 펌웨어 또는 그 임의의 조합들과 같은 다양한 적합한 프로세싱 회로 중 임의의 것으로서 구현될 수도 있다. 비디오 인코더 (28) 및 비디오 디코더 (48) 의 각각은 어느 하나가 결합된 비디

오 인코더/디코더 (CODEC) 의 일부로서 통합될 수도 있는 하나 이상의 인코더들 또는 디코더들에 포함될 수도 있다. 마찬가지로, 오디오 인코더 (26) 및 오디오 디코더 (46) 의 각각은 어느 하나가 결합된 CODEC 의 일부로서 통합될 수도 있는 하나 이상의 인코더들 또는 디코더들에 포함될 수도 있다. 비디오 인코더 (28), 비디오 디코더 (48), 오디오 인코더 (26), 오디오 디코더 (46), 캡슐화 유닛 (30), 취출 유닛 (52), 및/또는 캡슐화해제 유닛 (50) 을 포함하는 장치는 집적 회로, 마이크로프로세서, 및/또는 무선 통신 디바이스, 이를 테면 셀룰러 전화기를 포함할 수도 있다.

[0343] 클라이언트 디바이스 (40), 서버 디바이스 (60), 및/또는 콘텐츠 준비 디바이스 (20) 는 본 개시의 기법들에 따라 동작하도록 구성될 수도 있다. 예의 목적들을 위해, 본 개시는 클라이언트 디바이스 (40) 및 서버 디바이스 (60) 에 대하여 이들 기법들을 설명한다. 그러나, 콘텐츠 준비 디바이스 (20) 는 서버 디바이스 대신에 (또는 그것에 더하여) 이들 기법들을 수행하도록 구성될 수도 있다는 것이 이해될 수 있다.

[0344] 캡슐화 유닛 (30) 은 NAL 유닛이 속하는 프로그램을 식별하는 헤더 뿐만 아니라, 페이로드, 예를 들어, 오디오 데이터, 비디오 데이터, 또는 NAL 유닛이 대응하는 전송 또는 프로그램 스트림을 기술하는 데이터를 포함하는 NAL 유닛들을 형성할 수도 있다. 예를 들어, H.264/AVC 에서, NAL 유닛은 1-바이트 헤더 및 가변하는 사이즈의 페이로드를 포함한다. 그 페이로드에서의 비디오 데이터를 포함하는 NAL 유닛은 비디오 데이터의 다양한 입도 레벨들을 포함할 수도 있다. 예를 들어, NAL 유닛은 비디오 데이터의 블록, 복수의 블록들, 비디오 데이터의 슬라이스, 또는 비디오 데이터의 전체 픽처를 포함할 수도 있다. 캡슐화 유닛 (30) 은 기본 스트림들의 PES 패킷들의 형태로 비디오 인코더 (28) 로부터 인코딩된 비디오 데이터를 수신할 수도 있다. 캡슐화 유닛 (30) 은 각각의 기본 스트림을 대응하는 프로그램과 연관시킬 수도 있다.

[0345] 캡슐화 유닛 (30) 은 또한 복수의 NAL 유닛들로부터 액세스 유닛들을 어셈블리할 수도 있다. 일반적으로, 액세스 유닛은 비디오 데이터의 프레임 표현하기 위한 하나 이상의 NAL 유닛들을 포함할 수도 있고, 물론 오디오 데이터는 이러한 오디오 데이터가 이용가능할 때 프레임에 대응한다. 액세스 유닛은 일반적으로 하나의 출력 시간 인스턴스에 대한 모든 NAL 유닛들, 예를 들어, 하나의 시간 인스턴스에 대한 모든 오디오 및 비디오 데이터를 포함한다. 예를 들어, 각각의 뷰가 20 fps (frames per second) 의 프레임 레이트를 가지면, 각각의 시간 인스턴스는 0.05 초의 시간 간격에 대응할 수도 있다. 이 시간 간격 동안에, 동일한 액세스 유닛 (동일한 시간 인스턴스) 의 모든 뷰들에 대한 특정 프레임들은 동시에 렌더링될 수도 있다. 하나의 예에서, 액세스 유닛은 프라이어리 코딩 픽처로서 제시될 수도 있는, 하나의 시간 인스턴스에서의 코딩된 픽처를 포함할 수도 있다.

[0346] 이에 따라, 액세스 유닛은 공통 시간적 인스턴스의 모든 오디오 및 비디오 프레임들을 포함할 수도 있고, 예를 들어, 모든 뷰들은 시간  $X$  에 대응한다. 본 개시는 또한, 특정한 뷰의 인코딩된 픽처를 "뷰 컴포넌트" 로서 지칭한다. 즉, 뷰 컴포넌트는 특정한 시간에 특정한 뷰에 대한 인코딩된 픽처 (또는 프레임) 를 포함할 수도 있다. 이에 따라, 액세스 유닛은 공통 시간적 인스턴스의 모든 뷰 컴포넌트들을 포함하는 것으로서 정의될 수도 있다. 액세스 유닛들의 디코딩 순서는 출력 또는 디스플레이 순서와 반드시 동일할 필요는 없다.

[0347] 미디어 프리젠테이션은 상이한 대안적 표현들 (예를 들어, 상이한 품질들을 가진 비디오 서비스들) 의 디스크립션들을 포함할 수도 있는 미디어 프리젠테이션 디스크립션 (MPD) 을 포함할 수도 있고 디스크립션은 예를 들어, 코덱 정보, 프로파일 값, 및 레벨 값을 포함할 수도 있다. MPD 는 매니페스트 파일, 이를 테면 매니페스트 파일 (66) 의 하나의 예이다. 클라이언트 디바이스 (40) 는 다양한 프리젠테이션들의 무비 프래그먼트들에 액세스하는 방법을 결정하기 위해 미디어 프리젠테이션의 MPD 를 취출할 수도 있다. 무비 프래그먼트들은 비디오 파일들의 무비 프래그먼트 박스들 (moof 박스들) 에 위치될 수도 있다.

[0348] 매니페스트 파일 (66) (이는 예를 들어, MPD 를 포함할 수도 있다) 은 표현들 (68) 의 세그먼트들의 이용가능성을 광고할 수도 있다. 즉, MPD 는 표현들 (68) 중 하나의 제 1 세그먼트가 이용가능해지는 벽시계 시간을 나타내는 정보는 물론, 표현들 (68) 내의 세그먼트들의 지속기간들을 나타내는 정보를 포함할 수도 있다. 이 방식으로, 클라이언트 디바이스 (40) 의 취출 유닛 (52) 은 시작 시간 뿐만 아니라 특정한 세그먼트에 선행하는 세그먼트들의 지속기간들에 기초하여, 각각의 세그먼트가 이용가능할 때를 결정할 수도 있다.

[0349] 캡슐화 유닛 (30) 이 수신된 비디오에 기초하여 NAL 유닛들 및/또는 액세스 유닛들을 비디오 파일에 어셈블리한 후에, 캡슐화 유닛 (30) 은 비디오 파일을 출력을 위해 출력 인터페이스 (32) 로 전달한다. 일부 예들에서, 캡슐화 유닛 (30) 은 비디오 파일을 로컬로 저장하거나 또는 비디오 파일을 클라이언트 디바이스 (40) 에 바로 전송하는 것보다는, 비디오 파일을 출력 인터페이스 (32) 를 통해 원격 서버로 전송할 수도 있다. 출력 인터페이스 (32) 는 예를 들어, 송신기, 트랜시버, 데이터를 컴퓨터 판독가능 매체, 이를 테면 예를 들어, 광 드

라이브, 자기 미디어 드라이브 (예를 들어, 플로피 드라이브) 에 기입하기 위한 디바이스, 범용 직렬 버스 (USB) 포트, 네트워크 인터페이스, 또는 다른 출력 인터페이스를 포함할 수도 있다. 출력 인터페이스 (32) 는 비디오 파일을 컴퓨터 판독가능 매체 (34), 이를 테면 예를 들어 송신 신호, 자기 매체, 광 매체, 메모리, 플래시 드라이브, 또는 다른 컴퓨터 판독가능 매체에 출력한다.

[0350] 네트워크 인터페이스 (54) 는 네트워크 (74) 를 통해 NAL 유닛 또는 액세스 유닛을 수신하고 NAL 유닛 또는 액세스 유닛을 추출 유닛 (52) 을 통해 캡슐화해제 유닛 (50) 에 제공할 수도 있다. 캡슐화해제 유닛 (50) 은 비디오 파일의 엘리먼트들을 구성 PES 스트림들로 캡슐화해제하고, 인코딩된 데이터를 추출하기 위해 PES 스트림들을 패킷화해제하고, 그리고 예를 들어, 스트림의 PES 패킷 헤더들에 의해 나타낸 바와 같이, 인코딩된 데이터가 오디오 또는 비디오 스트림 중 일부인지 여부에 의존하여, 인코딩된 데이터를 오디오 디코더 (46) 또는 비디오 디코더 (48) 중 어느 하나로 전송할 수도 있다. 오디오 디코더 (46) 는 인코딩된 오디오 데이터를 디코딩하고 디코딩된 오디오 데이터를 오디오 출력 (42) 으로 전송하는 한편, 비디오 디코더 (48) 는 인코딩된 비디오 데이터를 디코딩하고 스트림의 복수의 뷰들을 포함할 수도 있는 디코딩된 비디오 데이터를 비디오 출력 (44) 으로 전송한다.

[0351] 하나의 예에서, 미디어 데이터를 추출하는 방법이 설명되고, 방법은 미디어 서버에 의해, 복수의 주기들로 미디어 데이터의 메인 콘텐츠를 스플리팅하는 단계, 미디어 서버에 의해, 제 1 주기와 제 2 주기 사이에 세컨더리 미디어 콘텐츠를 삽입하는 단계로서, 제 1 주기 및 제 2 주기는 시간적으로 순차적인, 상기 세컨더리 미디어 콘텐츠를 삽입하는 단계, 미디어 서버에 의해, 복수의 주기들의 각각에 대한 시간 표시자를 시그널링하는 단계로서, 시간 표시자는 복수의 주기들의 각각에 대한 하나 이상의 시간 특성들을 정의하는, 상기 시간 표시자를 시그널링하는 단계, 및 미디어 서버에 의해, DASH (dynamic adaptive streaming over HTTP) 클라이언트로부터, 복수의 주기들, 세컨더리 미디어 콘텐츠, 및 시간 표시자들을 포함하는 미디어 스트림을 추출하기 위한 요청을 수신하는 단계를 포함한다.

[0352] 다른 예에서, 미디어 데이터를 추출하는 방법이 설명되고, 방법은 미디어 콘텐츠에 대한 매니페스트 파일이 미디어 콘텐츠가 제 1 주기 및 제 2 주기를 포함한다는 것을 나타내고, 매니페스트 파일이 제 1 주기 및 제 2 주기가 연속적이라는 것을 나타내고, 그리고 광고 미디어 데이터가 제 1 주기와 제 2 주기 사이의 삽입을 위해 이용가능하다고 결정하는 단계, 제 1 주기에서의 제 1 적응 세트를, 제 1 적응 세트에 대해 시그널링된 특성들에 적어도 부분적으로 기초하여 선택하는 단계, 제 1 적응 세트의 선택에 기초하여 제 1 적응 세트의 미디어 데이터를 추출하는 단계, 광고 미디어 데이터를 추출하는 단계, 및 제 1 적응 세트의 선택에 기초하여 제 1 적응 세트와 연관되는 매니페스트 파일이 나타내는 제 2 주기에서의 제 2 적응 세트의 미디어 데이터를 추출하는 단계를 포함한다.

[0353] 다른 예에서, 미디어 데이터를 추출하는 방법이 설명되고, 방법은 미디어 콘텐츠가 제 1 주기 및 제 2 주기를 포함하는지 여부를 결정하는 단계, 제 1 주기 및 제 2 주기가 연속적인지 여부를 결정하는 단계, 광고 미디어 데이터가 제 1 주기와 제 2 주기 사이의 삽입을 위해 이용가능한지 여부를 결정하는 단계, 및 매니페스트 파일을 형성하는 단계로서, 매니페스트 파일은 미디어 콘텐츠가 제 1 주기 및 제 2 주기를 포함하는지 여부, 제 1 주기 및 제 2 주기가 연속적인지 여부, 및 광고 미디어 데이터가 제 1 주기와 제 2 주기 사이의 삽입을 위해 이용가능한지 여부의 표시들을 포함하는, 상기 매니페스트 파일을 형성하는 단계를 포함한다.

[0354] 도 2 는 예의 멀티미디어 콘텐츠 (102) 의 엘리먼트들을 예시하는 개념적 다이어그램이다. 멀티미디어 콘텐츠 (102) 는 멀티미디어 콘텐츠 (64) (도 1), 또는 메모리 (62) 에 저장된 다른 멀티미디어 콘텐츠에 대응할 수도 있다. 도 2 의 예에서, 멀티미디어 콘텐츠 (102) 는 미디어 프리젠테이션 디스크립션 (MPD) (104) 및 복수의 표현들 (110A 내지 110N) 을 포함한다. 표현 (110A) 은 옵션적 헤더 데이터 (112) 및 세그먼트들 (114A 내지 114N) (세그먼트들 (114)) 을 포함하는 한편, 표현 (110N) 은 옵션적 헤더 데이터 (122) 및 세그먼트들 (124A 내지 124N) (세그먼트들 (124)) 을 포함한다. 문자 N 은 편의상 표현들 (110A, 110N) 의 각각에서의 마지막 무비 프래그먼트를 지정하는데 사용된다. 일부 예들에서, 표현들 (110A, 110N) 사이에는 상이한 수들의 무비 프래그먼트들이 존재할 수도 있다.

[0355] MPD (104) 는 표현들 (110A 내지 110N) 과는 별도인 데이터 구조를 포함할 수도 있다. MPD (104) 는 도 1 의 매니페스트 파일 (66) 에 대응할 수도 있다. 마찬가지로, 표현들 (110A 내지 110N) 은 도 1 의 표현들 (68) 에 대응할 수도 있다. 일반적으로, MPD (104) 는 표현들 (110A 내지 110N) 의 특성들, 이를 테면 코딩 및 렌더링 특성들을 일반적으로 기술하는 데이터, 적응 세트들, MPD (104) 가 대응하는 프로파일, 텍스트 타임 정보, 카메라 각도 정보, 레이팅 정보, 트릭 모드 정보 (예를 들어, 시간적 서브-시퀀스들을 포함하는 표현



들을 표시하는 정보), 및/또는 (예를 들어, 재생 동안에 미디어 콘텐츠에의 타겟팅된 광고 삽입을 위한) 원격 주기들을 추출하기 위한 정보를 포함할 수도 있다.

[0356] 헤더 데이터 (112) 는 존재할 때, 세그먼트들 (114) 의 특성들, 예를 들어, 랜덤 액세스 포인트들 (RAP들, 스트림 액세스 포인트들 (SAP들) 로도 또한 지칭됨) 의 시간적 로케이션들, 세그먼트들 (114) 중 어느 것이 랜덤 액세스 포인트들을 포함하는지, 세그먼트들 (114) 내의 랜덤 액세스 포인트들에 대한 바이트 오프셋들, 세그먼트들 (114) 의 URL (uniform resource locator) 들, 또는 세그먼트들 (114) 의 다른 양태들을 기술할 수도 있다.

헤더 데이터 (122) 는 존재할 때, 세그먼트들 (124) 에 대한 유사한 특성들을 기술할 수도 있다. 추가적으로 또는 대안적으로, 이러한 특성들은 MPD (104) 내에 완전히 포함될 수도 있다.

[0357] 세그먼트들 (114, 124) 은 각각이 비디오 데이터의 프레임들 또는 슬라이스들을 포함할 수도 있는 하나 이상의 코딩된 비디오 샘플들을 포함한다. 세그먼트들 (114) 의 코딩된 비디오 샘플들의 각각은 유사한 특성들, 예를 들어, 높이, 폭, 및 대역폭 요건들을 가질 수도 있다. 이러한 특성들은 이러한 데이터가 도 2 의 예에서 예시되지는 않지만, MPD (104) 의 데이터에 의해 기술될 수도 있다. MPD (104) 는 본 개시에서 설명된 시그널링된 정보 중 임의의 것 또는 전부의 추가로, 3GPP 사양에 의해 기술한 바와 같은 특성들을 포함할 수도 있다.

[0358] 세그먼트들 (114, 124) 의 각각은 고유한 URL (uniform resource locator) 과 연관될 수도 있다. 따라서, 세그먼트들 (114, 124) 의 각각은 DASH 와 같은 스트리밍 네트워크 프로토콜을 이용하여 독립적으로 추출가능할 수도 있다. 이 방식으로, 목적지 디바이스, 이를 테면 클라이언트 디바이스 (40) 는 세그먼트들 (114 또는 124) 을 추출하기 위해 HTTP GET 요청을 이용할 수도 있다. 일부 예들에서, 클라이언트 디바이스 (40) 는 세그먼트들 (114 또는 124) 의 특정 바이트 범위들을 추출하기 위해 HTTP 부분 GET 요청들을 이용할 수도 있다.

[0359] 미디어 프리젠테이션 디스크립션 (104) 은 본 개시의 기법들에 따라, 표현들 (110A 내지 110N) 이 대응하는 주기가 후속 표현들 (미도시) 이 대응할 수도 있는 다른 주기와 연속적이라는 것을 나타낼 수도 있다. 이에 따라, 표현들 (110A 내지 110N) 을 포함하는 적응 세트는 다른 주기의 적응 세트와 연관될 수도 있다. 따라서, 표현들 (110A 내지 110N) 을 포함하는 적응 세트의 하나 이상의 특성들은 다른 적응 세트와 동일할 수도 있고, 여기서 이러한 특성들은 미디어 프리젠테이션 디스크립션 (104) 의 @lang 속성에 의해 기술한 바와 같은 언어, 미디어 프리젠테이션 디스크립션 (104) 의 @contentType 속성에 의해 기술된 미디어 컴포넌트 타입, 미디어 프리젠테이션 디스크립션 (104) 의 @par 속성에 의해 기술한 바와 같은 픽처 에스펙트비, 미디어 프리젠테이션 디스크립션 (104) 의 역할 엘리먼트들에 의해 기술한 바와 같은 임의의 역할 속성, 미디어 프리젠테이션 디스크립션 (104) 의 액세스가능성 엘리먼트들에 의해 기술한 바와 같은 임의의 액세스가능성 속성, 및/또는 미디어 프리젠테이션 디스크립션 (104) 의 뷰포인트 엘리먼트들에 의해 기술한 바와 같은 임의의 뷰포인트 속성 중 임의의 것 또는 전부를 포함할 수도 있다.

[0360] 이 방식으로, 클라이언트 디바이스 (40) 는 미디어 프리젠테이션 디스크립션 (104) 에서 시그널링된 특성들에 기초하여 표현들 (110A 내지 110N) 을 포함하는 적응 세트를 선택하고, 그 후 표현들 (110A 내지 110N) 을 포함하는 적응 세트의 선택에 기초하여 (후속의, 연속적인 주기의) 다른 적응 세트의 미디어 데이터를 추출할 수도 있다. 즉, 클라이언트 디바이스 (40) 는 (후속 주기의 적응 세트가 표현들 (110A 내지 110N) 을 포함하는 적응 세트와 연관된다는 미디어 프리젠테이션 디스크립션 (104) 에서의 표시에 기초하여) 이러한 특성들이 동일하기 때문에, 연관된 적응 세트의 특성들을 평가할 필요는 없다.

[0361] 도 3 은 본 개시의 기법들을 구현할 수도 있는 다른 예의 시스템 (200) 을 예시하는 블록 다이어그램이다. 도 3 의 시스템 (200) 의 엘리먼트들은 도 1 의 엘리먼트들에 일반적으로 대응할 수도 있다. 예를 들어, 시스템 (200) 은 광고 (ad) 관정 서버 (208), 콘텐츠 분배 시스템 (212), 및 클라이언트 디바이스 (206) 를 포함한다. 콘텐츠 분배 시스템 (212) 의 엘리먼트들은 도 1 의 콘텐츠 준비 디바이스 (20) 및/또는 서버 디바이스 (60) 에 일반적으로 대응할 수도 있는 한편, 클라이언트 디바이스 (206) 의 엘리먼트들은 도 1 의 클라이언트 디바이스 (40) 에 대응할 수도 있다. 일부 예들에서, 클라이언트 디바이스 (206) 의 엘리먼트들은 도 1 의 추출 유닛 (52) 에 대응할 수도 있다.

[0362] 이 예에서, 클라이언트 디바이스 (206) 는 미디어 엔진 (202) 및 DASH 액세스 클라이언트 (204) 를 포함한다. 콘텐츠 분배 시스템 (212) 은 MPD 생성기 (214), 패키저 (216), 및 콘텐츠 분배 네트워크 (CDN)/오리진 서버 (218) 를 포함한다. 오리진 서버 (218) 는 MPD (220), 메인 콘텐츠 (222A 내지 222C), 및 광고 데이터 (224A 내지 224C) 를 저장한다.

- [0363] 미디어 엔진 (202), DASH 액세스 클라이언트 (204), MPD 생성기 (214), 및 패키저 (216) 는 하드웨어 또는 소프트웨어로 구현될 수도 있다. 소프트웨어로 구현될 경우, 하나 이상의 프로세싱 유닛들 및 하나 이상의 컴퓨터 판독가능 저장 매체들과 같은 필수 하드웨어가 또한 제공된다는 것이 추정된다. 컴퓨터 판독가능 저장 매체들은 소프트웨어에 대한 명령들을 저장할 수도 있고, 프로세싱 유닛들은 상기 설명된 기능성을 수행하기 위한 명령들을 실행할 수도 있다.
- [0364] 단일의 주기 프리젠테이션을 포함하는 CDN 상의 온-디맨드 콘텐츠를 가리키는 MPD 가 이용가능하다. MPD 는, 큐들 (ad 삽입 기회들) 에 기초하여, 서버 상의 콘텐츠의 변화 없이 ad 다수의 주기들로 수정된다. 각각의 기회에 대해, Ad 판정 서버에 대한 xlink 가 제공된다. MPD 는 DASH 클라이언트에 제공된다. MPD 에서의 각각의 xlink 에 대해, ad들이 사용자에게 추가, 가능하게는 타겟팅되는지가 체크된다. 만약 그렇지 않다면, 주기는 0 으로 레졸빙된다. 만약 그렇다면, 다른 콘텐츠가 추가된다. DASH 클라이언트는 콘텐츠를 연속적으로 플레이하고, 0 으로 레졸빙되면, 미디어 엔진은 새롭게 초기화되지 않는다.
- [0365] 도 3 의 MPD 생성기 (214) 는 2 개 이상의 주기들이 연속적이라는 것을 나타내도록 구성될 수도 있다. 이에 따라, MPD 생성기 (214) 는 본 개시의 기법들에 따라, MPD (또는 다른 매니페스트 파일) 에서, 상기 논의한 바와 같은, 연관되는 주기들의 적응 세트들을 나타낼 수도 있다. 따라서, 클라이언트 디바이스 (206) 는 MPD 생성기 (214) 에 의해 생성된 MPD 에서 시그널링된 제 1 주기의 적응 세트의 특성들에 기초하여, 하나의 주기의 적응 세트 및 다른 연속적인 주기의 연관된 적응 세트를 선택할 수도 있다.
- [0366] 이 방식으로, 콘텐츠 분배 시스템 (212) 및 CDN/오리진 서버 (218) 는 미디어 데이터를 전송하기 위한 디바이스의 예들을 표현하고, 그 디바이스는 미디어 데이터를 저장하도록 구성된 메모리, 및 하나 이상의 하드웨어-기반 프로세서들을 포함하고, 그 하나 이상의 하드웨어-기반 프로세서들은 시간적으로 순차적인 제 1 주기 및 제 2 주기를 포함하는 복수의 주기들로 미디어 데이터의 메인 콘텐츠를 스플리팅하고, 제 1 주기와 제 2 주기 사이에 세컨더리 미디어 콘텐츠를 삽입하고, 복수의 주기들의 각각에 대한 시간 표시자를 시그널링하는 것으로서, 시간 표시자는 복수의 주기들의 각각에 대한 하나 이상의 시간 특성들을 정의하는, 상기 시간 표시자를 시그널링하고, 그리고 복수의 주기들, 세컨더리 미디어 콘텐츠, 및 시간 표시자들을 포함하는 미디어 스트림을 추출하기 위한 DASH (dynamic adaptive streaming over HTTP) 클라이언트로부터의 요청을 수신하도록 구성된다.
- [0367] 마찬가지로, 클라이언트 디바이스 (206) 는 미디어 데이터를 추출하기 위한 디바이스의 일 예를 표현하고, 디바이스는 미디어 데이터를 저장하도록 구성된 메모리; 및 하나 이상의 하드웨어-기반 프로세서들을 포함하고, 그 하나 이상의 하드웨어-기반 프로세서들은 미디어 콘텐츠에 대한 매니페스트 파일이 미디어 콘텐츠가 제 1 주기 및 제 2 주기를 포함한다는 것을 나타내고, 매니페스트 파일이 제 1 주기 및 제 2 주기가 연속적이라는 것을 나타내고, 그리고 광고 미디어 데이터가 제 1 주기와 제 2 주기 사이의 삽입을 위해 이용가능하다고 결정하고, 제 1 주기에서의 제 1 적응 세트를, 제 1 적응 세트에 대해 시그널링된 특성들에 적어도 부분적으로 기초하여 선택하고, 제 1 적응 세트의 선택에 기초하여 제 1 적응 세트의 미디어 데이터를 추출하고, 광고 미디어 데이터를 추출하고, 그리고 제 1 적응 세트의 선택에 기초하여 제 1 적응 세트와 연관되는 매니페스트 파일이 나타내는 제 2 주기에서의 제 2 적응 세트의 미디어 데이터를 추출하고 추출된 미디어 데이터를 메모리에 저장하도록 구성된다.
- [0368] 도 4 는 현재의 개시의 하나 이상의 기법들에 따른 미디어 데이터를 전송하는 방법을 예시하는 플로우 다이어그램이다. 이 예에서, 미디어 서버는 복수의 주기들로 미디어 데이터의 메인 콘텐츠를 스플리팅할 수도 있다 (300). 미디어 서버는 제 1 주기와 제 2 주기 사이에 세컨더리 미디어 콘텐츠를 삽입할 수도 있고 (302), 여기서 제 1 주기 및 제 2 주기는 시간적으로 순차적이다. 세컨더리 미디어 콘텐츠는 예를 들어, 메인 미디어 콘텐츠와 실질적으로 동일한 방식으로 제시될 광고 미디어 콘텐츠일 수도 있다. 미디어 서버는 복수의 주기들의 각각에 대한 시간 표시자를 시그널링할 수도 있고 (304), 여기서 시간 표시자는 복수의 주기들의 각각에 대한 하나 이상의 시간 특성들을 정의한다. 미디어 서버는 예를 들어, DASH (dynamic adaptive streaming over HTTP) 클라이언트로부터, 미디어 스트림을 추출하기 위한 요청을 수신할 수도 있다 (306). 미디어 스트림에 대한 요청은 복수의 주기들, 세컨더리 미디어 콘텐츠, 및 시간 표시자들을 포함하는 미디어 스트림을 추출하기 위한 요청에 대응할 수도 있다.
- [0369] 도 5 는 현재의 개시의 하나 이상의 기법들에 따른 미디어 데이터를 추출하는 방법을 예시하는 플로우 다이어그램이다. 이 예에서, 디바이스는 매니페스트 파일의 특성들을 결정한다 (310). 예를 들어, 디바이스는 미디어 콘텐츠에 대한 매니페스트 파일이 미디어 콘텐츠가 제 1 주기 및 제 2 주기를 포함한다는 것을

나타내고, 매니페스트 파일이 제 1 주기 및 제 2 주기가 연속적이라는 것을 나타내고, 그리고 광고 미디어 데이터가 제 1 주기와 제 2 주기 사이의 삽입을 위해 이용가능하다고 결정할 수도 있다. 디바이스는 제 1 적응 세트를 선택할 수도 있다 (312). 하나의 예에서, 제 1 적응 세트는 제 1 주기에 있을 수도 있고, 선택은 제 1 적응 세트에 대해 시그널링된 특성들에 적어도 부분적으로 기초할 수도 있다. 디바이스는 그 후 제 1 적응 세트의 선택에 기초하여 제 1 적응 세트의 미디어 데이터를 추출할 수도 있다 (314). 디바이스는 그 후 광고 미디어 데이터를 추출할 수도 있다 (316). 게다가, 디바이스는 제 1 적응 세트의 선택에 기초하여 제 2 적응 세트의 미디어 데이터를 추출할 수도 있다 (318). 제 2 적응 세트는 제 2 주기에 있을 수도 있고, 매니페스트 파일은 제 2 적응 세트가 제 1 적응 세트와 연관된다는 것을 나타낼 수도 있다.

[0370] 도 6 은 현재의 개시의 하나 이상의 기법들에 따른 미디어 데이터를 전송하는 방법을 예시하는 플로우 다이어그램이다. 이 예에서, 디바이스는 미디어 콘텐츠가 제 1 주기 및 제 2 주기를 포함하는지 여부를 결정할 수도 있다 (320). 디바이스는 제 1 주기 및 제 2 주기가 연속적인지 여부를 결정할 수도 있다 (322). 디바이스는 광고 미디어 데이터가 예를 들어, 제 1 주기와 제 2 주기 사이의 삽입을 위해 이용가능한지 여부를 결정할 수도 있다 (324). 디바이스는 매니페스트 파일을 형성할 수도 있으며 (326), 여기서 매니페스트 파일은 미디어 콘텐츠가 제 1 주기 및 제 2 주기를 포함하는지 여부, 제 1 주기 및 제 2 주기가 연속적인지 여부, 및 광고 미디어 데이터가 제 1 주기와 제 2 주기 사이의 삽입을 위해 이용가능한지 여부의 표시들을 포함한다.

[0371] 도 7 은 미디어 서버에 의해 본 개시의 소정의 기법들을 수행하는 일 예의 방법을 예시하는 플로우차트이다. 이 예의 방법은 미디어 서버에 의해, 제 1 주기 및 제 2 주기를 포함하는 복수의 주기들로 미디어 데이터의 메인 콘텐츠를 스플리팅하는 단계를 포함하고, 여기서 제 1 주기 및 제 2 주기는 시간적으로 순차적이다 (350). 이 예의 방법은 또한, 미디어 서버에 의해, 세컨더리 미디어 콘텐츠가 제 1 주기와 제 2 주기 사이의 삽입을 위해 이용가능하다는 것을 나타내는 정보를 시그널링하는 단계를 포함한다 (352). 이 예의 방법은 미디어 서버에 의해, 복수의 주기들의 각각에 대한 시간 표시자를 시그널링하는 단계를 더 포함하고, 여기서 시간 표시자는 복수의 주기들의 각각에 대한 하나 이상의 시간 특성들을 정의한다 (354).

[0372] 도 8 은 클라이언트 디바이스에 의해 본 개시의 소정의 기법들을 수행하는 일 예의 방법을 예시하는 플로우차트이다. 이 예의 방법은 미디어 콘텐츠에 대한 매니페스트 파일이 미디어 콘텐츠가 제 1 주기 및 제 2 주기를 포함한다는 것을 나타내고, 매니페스트 파일이 제 1 주기 및 제 2 주기가 연속적이라는 것을 나타내고, 그리고 광고 미디어 데이터가 제 1 주기와 제 2 주기 사이의 삽입을 위해 이용가능하다고 결정하는 단계를 포함한다 (370). 이 예의 방법은 또한, 제 1 주기에서의 제 1 적응 세트를, 제 1 적응 세트에 대해 시그널링된 특성들에 적어도 부분적으로 기초하여 선택하는 단계를 포함한다 (372). 이 예의 방법은 제 1 적응 세트의 선택에 기초하여 제 1 적응 세트의 미디어 데이터를 추출하는 단계를 더 포함한다 (374). 마지막으로, 이 예의 방법은 제 1 적응 세트의 선택에 기초하여 제 1 적응 세트와 연관되는 제 2 적응 세트에서의 제 2 적응 세트의 미디어 데이터를 추출하는 단계를 포함한다 (376).

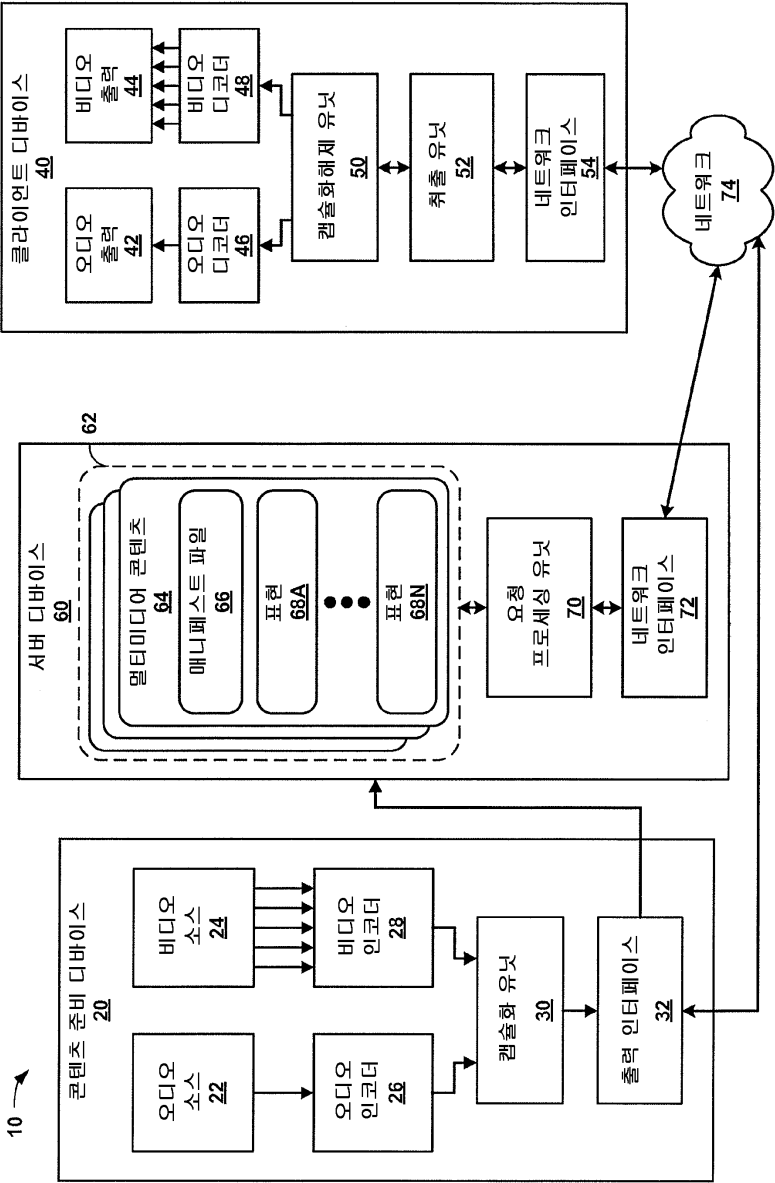
[0373] 일부 예들에서, 클라이언트 디바이스는 상기 설명된 제 1 예, 제 2 예, 및 제 3 예의 기법들 중 임의의 것 또는 전부를 수행하도록 구성될 수도 있다는 것이 이해되어야 한다. 예를 들어, 상이한 콘텐츠 분배 네트워크들은 타겟팅된 광고 삽입을 위한 상이한 메커니즘들을 지원할 수도 있고, 클라이언트 디바이스는 제 1 예, 제 2 예, 및/또는 제 3 예 중 임의의 것 또는 전부의 기법들을 구현할 수도 있다. 다른 예로서, 콘텐츠 분배 네트워크는 상기 설명된 제 1 예, 제 2 예, 및/또는 제 3 예의 기법들 중 임의의 것 또는 전부를 지원할 수도 있다. 더욱이, 상기 설명된 제 1 예, 제 2 예, 및/또는 제 3 예의 기법들은 임의의 조합으로 함께 수행될 수도 있다.

[0374] 하나 이상의 예들에서, 설명된 기법들은 하드웨어, 소프트웨어, 펌웨어, 또는 그 임의의 조합으로 구현될 수도 있다. 소프트웨어로 구현되면, 기능들은 컴퓨터 판독가능 매체 상에 하나 이상의 명령들 또는 코드로서 저장되거나 또는 이를 통해 송신되고 하드웨어-기반 프로세싱 유닛에 의해 실행될 수도 있다. 컴퓨터 판독가능 매체들은 데이터 저장 매체들과 같은 유형의 매체에 대응하는 컴퓨터 판독가능 저장 매체들, 또는 예를 들어, 통신 프로토콜에 따라 일 장소로부터 다른 장소로의 컴퓨터 프로그램의 전송을 용이하게 하는 임의의 매체를 포함하는 통신 매체들을 포함할 수도 있다. 이 방식으로, 컴퓨터 판독가능 매체들은 일반적으로 (1) 비일시적인 유형의 컴퓨터 판독가능 저장 매체들 또는 (2) 신호 또는 캐리어파와 같은 통신 매체에 대응할 수도 있다. 데이터 저장 매체들은 본 개시에서 설명된 기법들의 구현을 위한 명령들, 코드, 및/또는 데이터 구조들을 추출하기 위해 하나 이상의 컴퓨터들 또는 하나 이상의 프로세서들에 의해 액세스될 수 있는 임의의 이용가능한 매체들일 수도 있다. 컴퓨터 프로그램 제품은 컴퓨터 판독가능 매체를 포함할 수도 있다.

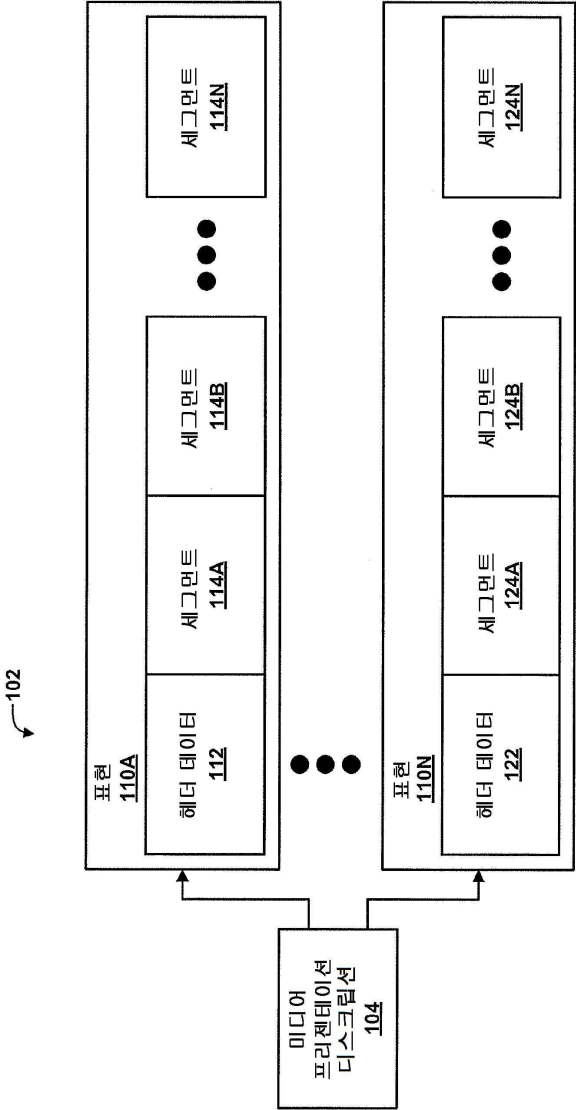
- [0375] 제한이 아닌 일 예로, 이러한 컴퓨터 판독가능 저장 매체들은 RAM, ROM, EEPROM, CD-ROM 또는 다른 광 디스크 스토리지, 자기 디스크 스토리지, 또는 다른 자기 저장 디바이스들, 플래시 메모리, 또는 명령들 또는 데이터 구조들 형태의 원하는 프로그램 코드를 저장하는데 이용될 수 있고 컴퓨터에 의해 액세스될 수 있는 임의의 다른 매체를 포함할 수 있다. 또한, 임의의 접속이 적절히 컴퓨터 판독가능 매체라 불리게 된다. 예를 들어, 명령들이 동축 케이블, 광섬유 케이블, 트위스티드 페어, 디지털 가입자 회선 (DSL), 또는 적외선, 무선, 및 마이크로파와 같은 무선 기술들을 이용하여 웹사이트, 서버, 또는 다른 원격 소스로부터 송신된다면, 매체의 정의에는 동축 케이블, 광섬유 케이블, 트위스티드 페어, DSL, 또는 적외선, 무선, 및 마이크로파와 같은 무선 기술들이 포함된다. 그러나, 컴퓨터 판독가능 저장 매체들 및 데이터 저장 매체들은 연결들, 캐리어파들, 신호들, 또는 다른 일시적 매체들을 포함하지 않고, 그 대신 비일시적 유형의 저장 매체들로 지향된다는 것이 이해될 수 있다. 디스크 (disk) 및 디스크 (disc) 는 본 명세서에서 사용한 바와 같이, 콤팩트 디스크 (CD), 레이저 디스크, 광 디스크, 디지털 다기능 디스크 (DVD), 플로피 디스크 및 블루-레이 디스크를 포함하고, 여기서 디스크 (disk) 들은 보통 데이터를 자기적으로 재생하는 한편, 디스크 (disc) 들은 레이저들로 데이터를 광학적으로 재생한다. 상기의 조합들이 또한 컴퓨터 판독가능 매체들의 범위 내에 포함될 수 있다.
- [0376] 명령들은 하나 이상의 프로세서들, 이를 테면 하나 이상의 디지털 신호 프로세서들 (DSP들), 범용 마이크로프로세서들, 주문형 집적 회로들 (ASIC들), 필드 프로그래밍가능 로직 어레이들 (FPGA들), 또는 다른 등가의 집적 또는 이산 로직 회로에 의해 실행될 수도 있다. 이에 따라, 용어 "프로세서" 는 본 명세서에서 사용한 바와 같이, 전술한 구조 또는 본 명세서에서 설명된 기법들의 구현에 적합한 임의의 다른 구조 중 임의의 것을 지칭할 수도 있다. 또한, 일부 양태들에서, 본 명세서에서 설명된 기능성은 인코딩 및 디코딩을 위해 구성되고 또는 결합 코텍에서 통합된 전용 하드웨어 및/또는 소프트웨어 모듈들 내에 제공될 수도 있다. 또한, 기법들은 하나 이상의 회로들 또는 로직 엘리먼트들에서 완전히 구현될 수 있다.
- [0377] 본 개시의 기법들은 무선 핸드셋, 집적 회로 (IC) 또는 IC들의 세트 (예를 들어, 칩 세트) 를 포함하는 광범위한 디바이스들 또는 장치들에서 구현될 수도 있다. 다양한 컴포넌트들, 모듈들, 또는 유닛들은 본 개시에서 개시된 기법들을 수행하도록 구성된 디바이스들의 기능적 양태들을 강조하기 위해 설명되지만 상이한 하드웨어 유닛들에 의한 실현을 반드시 요구하는 것은 아니다. 오히려, 상기 설명한 바와 같이, 다양한 유닛들은 코텍 하드웨어 유닛에서 결합되거나 또는 적합한 소프트웨어 및/또는 펌웨어와 함께 상기 설명한 바와 같은 하나 이상의 프로세서들을 포함하는 상호동작적 하드웨어 유닛들의 콜렉션에 의해 제공될 수도 있다.
- [0378] 다양한 예들이 설명되었다. 이들 및 다른 예들은 다음의 청구항들의 범위 내에 있다.

도면

도면1

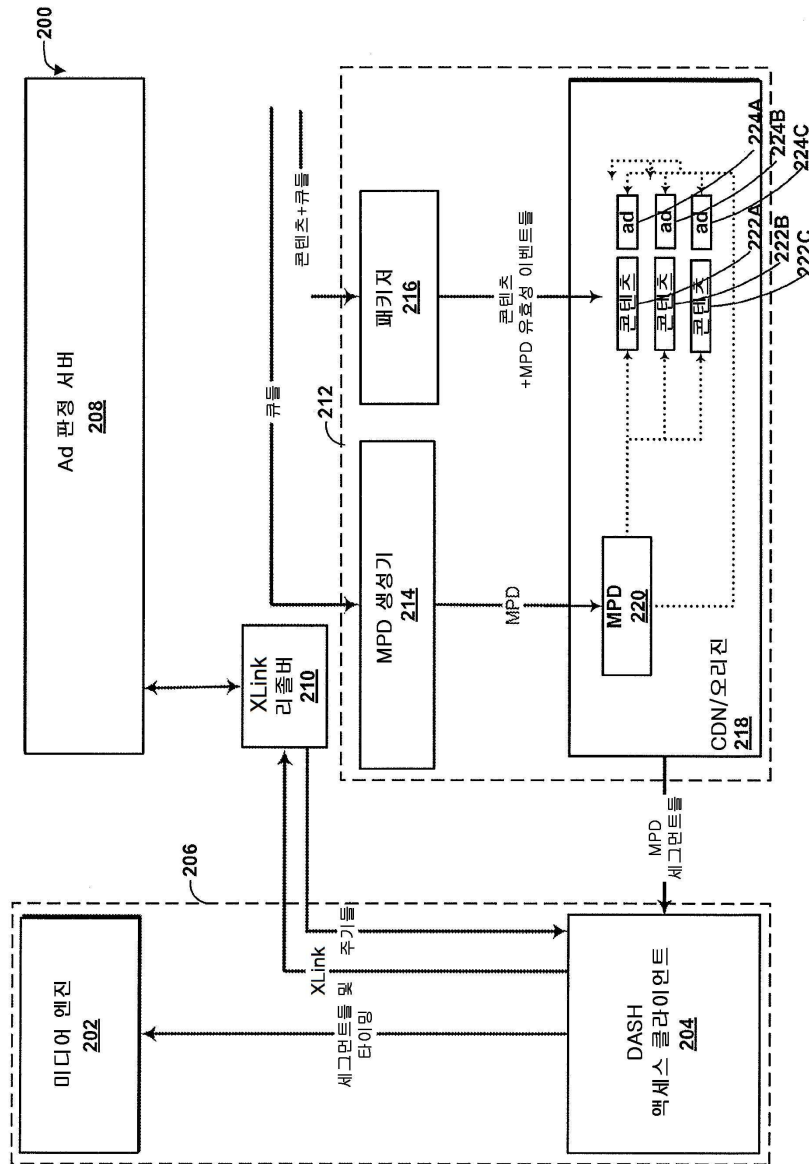


도면2

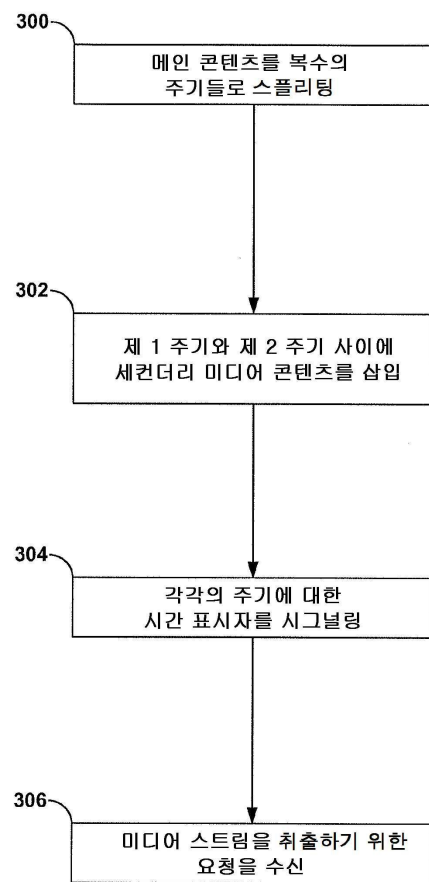




도면3

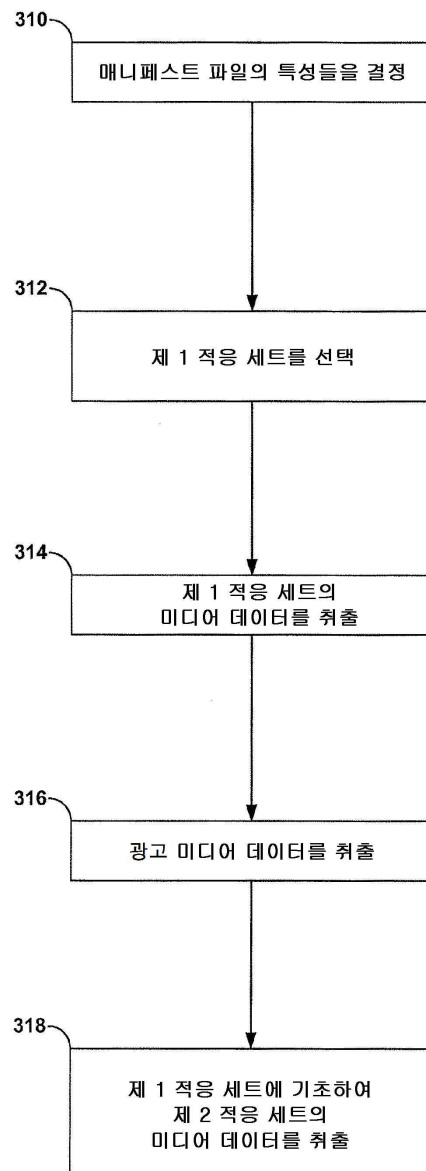


도면4

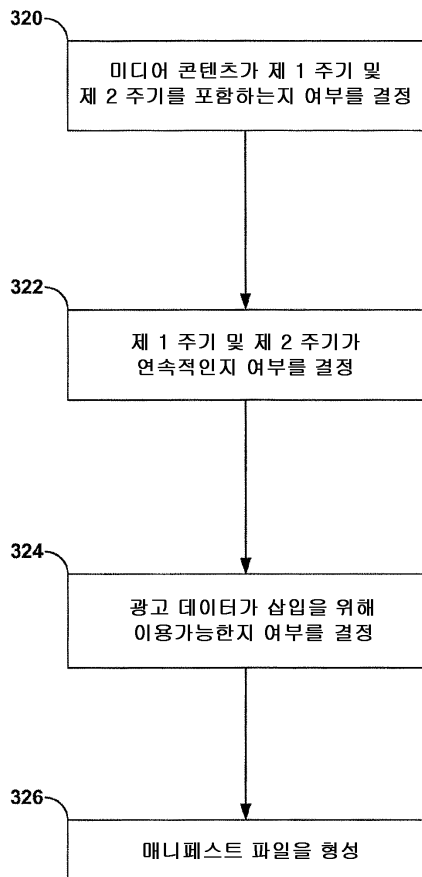




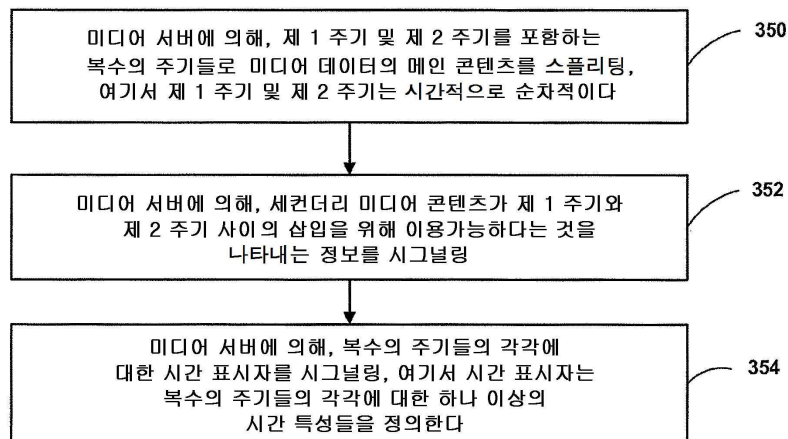
도면5



도면6



도면7



도면8

