



(19) 대한민국특허청(KR)  
(12) 등록특허공보(B1)

(45) 공고일자 2021년06월09일  
(11) 등록번호 10-2262813  
(24) 등록일자 2021년06월03일

- (51) 국제특허분류(Int. Cl.)  
H04L 29/06 (2006.01)
- (52) CPC특허분류  
H04L 65/80 (2013.01)  
H04L 65/1069 (2013.01)
- (21) 출원번호 10-2019-7037172
- (22) 출원일자(국제) 2018년06월11일  
심사청구일자 2019년12월16일
- (85) 번역문제출일자 2019년12월16일
- (65) 공개번호 10-2020-0007947
- (43) 공개일자 2020년01월22일
- (86) 국제출원번호 PCT/EP2018/065381
- (87) 국제공개번호 WO 2018/234080  
국제공개일자 2018년12월27일
- (30) 우선권주장  
62/522,422 2017년06월20일 미국(US)
- (56) 선행기술조사문헌  
KR1020030061806 A\*  
US20130219074 A1\*  
US20160088326 A1\*  
\*는 심사관에 의하여 인용된 문헌

- (73) 특허권자  
텔레폰악티에블라갯엘엠에릭슨(펍)  
스웨덴왕국 스톡홀름 에스이-164 83
- (72) 발명자  
로마르 토르스텐  
독일 52072 아헨 티타츠펠드 29  
이브라힘 모하메드  
독인 52066 아헨 비스마르크스트라췌 199  
(뒷면에 계속)
- (74) 대리인  
서장찬, 박병석

전체 청구항 수 : 총 14 항

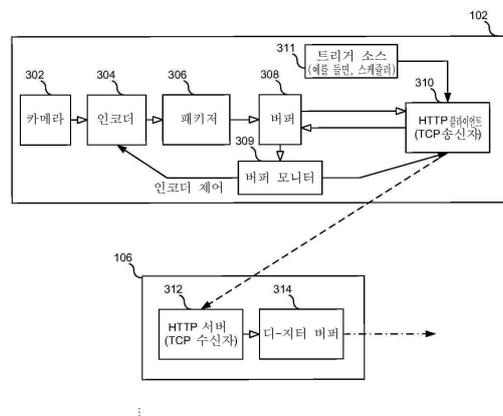
심사관 : 박보미

(54) 발명의 명칭 라이브 업링크 적응 스트리밍을 위한 장치 및 방법

(57) 요약

라이브 미디어 피드를 서버에 업스트리밍하기 위해 클라이언트에 의해 실행되는 방법이 제공된다. 그 방법은 클라이언트가 서버와 운송 계층 연결을 설정하는 단계; 헤더 및 바디를 포함하는 HTTP 요청 메시지를 서버에 전송하는 단계; 및 미디어 데이터가 발생될 때 라이브 미디어 피드에 대응하는 미디어 데이터를 전송 버퍼에 저장하 (뒷면에 계속)

대표도



는 단계를 포함한다. 헤더는 바디의 크기를 나타내지 않는다. 품질 설정은 미디어 데이터를 발생하는데 사용된다. 바디를 전송하는 단계는:

- 1) 미디어 데이터의 적어도 일부를 서버에 전송하는 단계;
- 2) 미디어 데이터의 상기 적어도 일부를 전송 버퍼에서 제거하는 단계;
- 3) 클라이언트가 미디어 데이터를 발생하는데 사용되는 품질 설정을 수정하여야 하는가 여부를 결정하는 단계; 및

4) 전송 중지 트리거가 검출될 때까지 단계(1), (2), 및 (3)을 반복하는 단계를 포함한다.

(52) CPC특허분류

*H04L 65/4084* (2013.01)

*H04L 65/602* (2013.01)

*H04L 65/604* (2013.01)

*H04L 65/607* (2013.01)

*H04L 65/608* (2013.01)

**마그너 라파엘로**

독일 40221 뒤셀도르프 볼머스웨터 에스티알. 236

(72) 발명자

**엘 에셀리 알리**

독일 52072 아헨 티타츠펠드 108

## 명세서

### 청구범위

#### 청구항 1

미디어 소스에 의해 생성되는 라이브 미디어 (오디오 및/또는 비디오) 피드를 서버(106)에 업스트리밍하기 위해 클라이언트 장치("클라이언트")(102)에 의해 실행되는 방법으로서,

상기 클라이언트(102)가 상기 서버(106)와 TCP 연결과 같은 운송 계층 연결을 설정하는 단계;

상기 클라이언트(102)가 헤더 및 바디를 포함하는 하이퍼텍스트 전달 프로토콜(HTTP) 요청 메시지를 상기 운송 계층 연결을 통해 상기 서버(106)에 전송하는 단계로, 여기서 상기 HTTP 요청 메시지의 상기 헤더는 상기 HTTP 요청 메시지의 상기 바디의 크기를 나타내지 않는 단계; 및

미디어 데이터가 발생될 때 상기 클라이언트(102)가 상기 라이브 미디어 피드에 대응하는 상기 미디어 데이터를 전송 버퍼(308)에 저장하는 단계로, 여기서 품질 설정은 상기 미디어 데이터를 발생하는데 사용되는 단계를 포함하고,

상기 운송 계층 연결을 통해 상기 서버(106)에 상기 HTTP 요청 메시지의 상기 바디를 전송하는 단계는:

1) 상기 클라이언트(102)가 상기 전송 버퍼(308)에 현재 저장된 상기 미디어 데이터의 적어도 일부를 상기 운송 계층 연결을 통해 상기 서버(106)에 전송하는 단계로, 미디어 데이터의 전송된 일부는 공통 미디어 애플리케이션 포맷(Common Media Application Format, CMAF) 청크의 포맷을 갖는 단계;

2) 상기 클라이언트(102)가 상기 미디어 데이터의 상기 적어도 일부를 상기 전송 버퍼(308)에서 제거하는 단계;

3) 전송 버퍼의 업로딩 진행을 모니터링함으로써 상기 라이브 미디어 피드에 대응하는 상기 미디어 데이터를 발생하는데 사용되는 상기 품질 설정을 상기 클라이언트(102)가 수정하여야 하는가 여부를 상기 클라이언트(102)가 결정하는 단계; 및

4) 전송 중지 트리거가 검출될 때까지 상기 클라이언트(102)가 상기 단계(1), (2), 및 (3)을 반복하는 단계를 포함하는 방법.

#### 청구항 2

제1항에 있어서,

상기 미디어 소스는 카메라(302)인 방법.

#### 청구항 3

제2항에 있어서,

상기 카메라(302)는 상기 클라이언트(102)의 필수 부분인 방법.

#### 청구항 4

제2항에 있어서,

상기 카메라(302)는 상기 클라이언트(102)로부터 떨어져 있는, 블루투스 연결과 같은 링크를 통해 상기 클라이언트(102)에 연결된 드론 상의 카메라(302)인 방법.

#### 청구항 5

제1항 내지 제4항 중 임의의 한 항에 있어서,

상기 트리거는 지속 시간을 지정하는 스케줄을 기반으로 하고,

상기 클라이언트(102)는 상기 지속 시간이 경과했음을 검출한 결과로 상기 트리거를 검출하는 방법.

**청구항 6**

제1항 내지 제4항 중 임의의 한 항에 있어서,

상기 클라이언트(102)가 상기 미디어 소스에 의해 발생된 제1 세트의 미디어 프레임을 인코딩하여 제1 세트의 인코딩 미디어 프레임을 생성하는데 제1 품질 설정을 사용하는 단계로, 여기서 상기 전송 버퍼(308)에 저장된 상기 미디어 데이터는 상기 제1 세트의 인코딩 미디어 프레임을 포함하는 단계를 더 포함하고;

상기 HTTP 요청 메시지의 상기 바디를 전송하는 상기 단계는 상기 클라이언트(102)가 제1 코덱 구성 정보를 전송하는 단계를 더 포함하고, 여기서 상기 제1 코덱 구성 정보는 상기 제1 품질 설정을 나타내는 정보를 포함하고;

상기 클라이언트(102)가 상기 전송 버퍼(308)에 저장된 데이터의 양을 모니터하는 단계를 더 포함하고;

상기 클라이언트(102)가 상기 전송 버퍼(308)에 저장된 상기 데이터의 양이 한계치를 초과함을 결정하는 단계를 더 포함하고;

상기 전송 버퍼(308)에 저장된 상기 데이터의 양이 상기 한계치를 초과함을 결정한 결과로, 상기 클라이언트(102)가 상기 미디어 소스에 의해 발생된 제2 세트의 미디어 프레임을 인코딩하여 제2 세트의 인코딩 미디어 프레임을 생성하는데 제2 품질 설정을 사용하는 단계를 더 포함하고;

상기 클라이언트(102)가 상기 라이브 미디어 피드에 대응하는 또 다른 미디어 데이터를 상기 전송 버퍼(308)에 저장하는 단계로, 여기서 상기 또 다른 미디어 데이터는 상기 제2 세트의 인코딩 미디어 프레임을 포함하는 단계를 더 포함하고,

또한, 상기 HTTP 요청 메시지의 상기 바디를 전송하는 상기 단계는 상기 클라이언트(102)가 제2 코덱 구성 정보를 전송하는 단계를 더 포함하고, 여기서 상기 제2 코덱 구성 정보는 상기 제2 품질 설정을 나타내는 정보를 포함하는 방법.

**청구항 7**

제1항 내지 제4항 중 임의의 한 항에 있어서,

상기 클라이언트(102)가 상기 미디어 소스에 의해 발생된 제1 세트의 미디어 프레임을 인코딩하여 제1 세트의 인코딩 미디어 프레임을 생성하는 단계로, 여기서 상기 전송 버퍼(308)에 저장된 상기 미디어 데이터는 상기 제1 세트의 인코딩 미디어 프레임을 포함하는 단계;

상기 클라이언트(102)가 상기 전송 버퍼(308)에 저장된 데이터의 양을 모니터하는 단계;

상기 클라이언트(102)가 상기 전송 버퍼(308)에 저장된 상기 데이터의 양이 한계치를 초과함을 결정하는 단계; 및

상기 전송 버퍼(308)에 저장된 상기 데이터의 양이 상기 한계치를 초과함을 결정한 결과로, 상기 클라이언트(102)가 상기 제1 세트의 미디어 프레임의 적어도 한 서브세트를 상기 전송 버퍼(308)로부터 드로핑하여 상기 드로핑된 프레임이 상기 서버(106)로 전송되지 않게 하는 단계를 더 포함하는 방법.

**청구항 8**

제7항에 있어서,

상기 전송 버퍼(308)는 미디어 프레임의 상기 서브세트를 포함하고 미디어 프레임의 상기 서브세트에 대한 메타 데이터를 더 포함하는 미디어 프래그먼트를 저장하고,

미디어 프레임의 상기 서브세트를 상기 전송 버퍼(308)로부터 드로핑하는 상기 단계는 상기 전송 버퍼(308)로부터 상기 미디어 프래그먼트를 드로핑하여 상기 미디어 프래그먼트가 상기 서버(106)에 전송되지 않게 하는 단계를 포함하는 방법.

**청구항 9**

제1항 내지 제4항 중 임의의 한 항에 있어서,

상기 클라이언트(102)가 상기 미디어 소스에 의해 발생된 비압축 비디오 프레임을 획득하는 단계;  
 상기 클라이언트(102)가 상기 비압축 비디오 프레임을 인코딩하여 인코딩 비디오 프레임을 생성하는 단계;  
 상기 클라이언트(102)가: i) 상기 인코딩 비디오 프레임 및 ii) 상기 인코딩 비디오 프레임에 관련된 메타-데이터를 포함하는 비디오 프래그먼트를 발생시키는 단계; 및  
 상기 클라이언트(102)가 상기 프래그먼트를 상기 전송 버퍼(308)에 저장하는 단계를 더 포함하는 방법.

**청구항 10**

제9항에 있어서,  
 상기 비디오 프래그먼트는: i) ISO-BMFF 비디오 프래그먼트 및 ii) 공통 미디어 애플리케이션 포맷(CMAF) 비디오 프래그먼트 중 하나인 방법.

**청구항 11**

제1항 내지 제4항 중 임의의 한 항에 있어서,  
 상기 라이브 미디어 피드는 라이브 배급을 위한 수신 엔터티(상기 서버(106))에 의해 더 처리될 수 있는 방법.

**청구항 12**

제1항 내지 제4항 중 임의의 한 항에 있어서,  
 상기 클라이언트(102)가 상기 전송 버퍼(308)에 현재 저장된 상기 미디어 데이터의 적어도 일부를 상기 운송 계층을 통해 상기 서버(106)에 전송하는 단계는 상기 전송 버퍼(308)에 현재 저장된 상기 미디어 데이터의 적어도 일부를 포함하는 데이터 체크를 전송하는 단계를 포함하는 방법.

**청구항 13**

제1항 내지 제4항 중 임의의 한 항에 있어서,  
 상기 라이브 미디어 피드에 대응하는 상기 미디어 데이터를 발생시키는데 사용되는 상기 품질 설정을 상기 클라이언트(102)가 수정하여야 하는가 여부를 결정하는 상기 단계는 전송 버퍼 레벨 및/또는 시간에 걸친 상기 전송 버퍼 레벨에서의 변화와 같은 버퍼 레벨 경사도를 기반으로 하는 방법.

**청구항 14**

미디어 소스에 의해 생성되는 라이브 미디어 (오디오 및/또는 비디오) 피드를 서버(106)에 업스트리밍하기 위한 이동 디바이스 또는 사용자 장비(UE) 상의 HTTP 클라이언트(102)로서,

상기 HTTP 클라이언트(102)는:

상기 서버(106)와 TCP 연결과 같은 운송 계층 연결을 설정하고;

헤더 및 바디를 포함하는 하이퍼텍스트 전달 프로토콜(HTTP) 요청 메시지를 상기 운송 계층 연결을 통해 상기 서버(106)에 전송하고, 여기서 상기 HTTP 요청 메시지의 상기 헤더는 상기 HTTP 요청 메시지의 상기 바디의 크기를 나타내지 않고; 또한

미디어 데이터가 발생될 때 상기 라이브 미디어 피드에 대응하는 상기 미디어 데이터를 전송 버퍼(308)에 저장하고, 여기서 품질 설정은 상기 미디어 데이터를 발생하는데 사용되도록 적용되고,

상기 운송 계층 연결을 통해 상기 서버(106)에 상기 HTTP 요청 메시지의 상기 바디를 전송하는 단계는:

- 1) 상기 HTTP 클라이언트(102)가 상기 전송 버퍼(308)에 현재 저장된 상기 미디어 데이터의 적어도 일부를 상기 운송 계층 연결을 통해 상기 서버(106)에 전송하는 단계로, 미디어 데이터의 전송된 일부는 공통 미디어 애플리케이션 포맷(Common Media Application Format, CMAF) 체크의 포맷을 갖는 단계;
- 2) 상기 HTTP 클라이언트(102)가 상기 미디어 데이터의 상기 적어도 일부를 상기 전송 버퍼(308)에서 제거하는 단계;
- 3) 전송 버퍼의 업로딩 진행을 모니터링함으로써 상기 라이브 미디어 피드에 대응하는 상기 미디어 데이터를 발생

하는데 사용되는 상기 품질 설정을 상기 HTTP 클라이언트(102)가 수정하여야 하는가 여부를 상기 HTTP 클라이언트(102)가 결정하는 단계; 및

4) 전송 중지 트리거가 검출될 때까지 상기 HTTP 클라이언트(102)가 상기 단계(1), (2), 및 (3)을 반복하는 단계를 포함하는 HTTP 클라이언트(102).

**청구항 15**

삭제

**청구항 16**

삭제

**청구항 17**

삭제

**청구항 18**

삭제

**청구항 19**

삭제

**청구항 20**

삭제

**발명의 설명**

**기술 분야**

[0001] 하이퍼텍스트 전달 프로토콜(Hypertext Transfer Protocol, HTTP)을 사용하는 라이브 업링크 적응 스트리밍에 관련된 실시예가 설명된다.

**배경 기술**

[0002] 업링크 스트리밍을 (즉, 클라이언트로부터 서버로의 스트리밍) 위한 가장 일반적인 스트리밍 프로토콜은 실시간 메시징 프로토콜(Real Time Messaging Protocol, RTMP)로, 이는 Adobe의 소유이다 (www.adobe.com/devnet/rtmp.html을 참조). 초기에, RTMP는 서버로부터 클라이언트로 플래쉬 비디오(Flash Video)를 스트리밍하기 위해 개발되었다. 그러나, 오늘날의 RTMP는 주로 업링크 스트리밍에 사용된다. RTMP는 확실한 업링크 스트리밍을 위해 전송 제어 프로토콜(Transmission Control Protocol, TCP)를 사용한다. RTMP는 HTTP와 비교하여 완전한 대체 실현이다. RTMP 스트림은 RTMP 프로토콜 핸들러 구조(rtmp://)에 의해 식별될 수 있으므로, rtmp://ex.com/live.swf 형태의 URL이 애플리케이션에 의해 해석될 수 있다.

**발명의 내용**

**해결하려는 과제**

[0003] RTMP에는 몇가지 단점이 있다. 예를 들면, 규격이 공개 표준화 기구 또는 공동체에 의해 제어되지 않기 때문에, 비디오, 오디오, 또는 자막 해결법을 부가할 규칙이나 방법이 없다.

[0004] RTMP의 또 다른 결점은 HTTP 만큼 널리 사용되지 않는다는 점이다. 따라서, HTTP를 사용하여 라이브 업링크 스트리밍을 실행하는 것이 유리하다. HTTP의 추가 이점은 HTTP 또는 소스 인증과 같은 모든 HTTP 지정 보안 기능이 재사용될 수 있다는 점이다.

[0005] C.Lottermann 등의 "업링크 적응 HTTP 스트리밍을 위한 네트워크-인식 비디오 레벨 인코딩(Network-Aware Video Level Encoding for Uplink Adaptive HTTP Streaming)". 2015 IEEE International Conference on Communications (ICC)는 (이후 "Lottermann"이라 칭하여지는) HTTP를 사용한 업링크 스트리밍을 설명한다. 그

러나, Lottermann에서 설명된 시스템에서는 비디오 패키저(video packager)가 원격 HTTP 클라이언트로부터 요청을 수신하고 그 요청에 응답하여 원격 HTTP 클라이언트에게 비디오를 스트리밍하는 기능을 갖춘 HTTP 서버와 함께 위치한다. 이러한 실현은 인프라구조에서 HTTP 클라이언트 위치를 찾고 클라이언트 디바이스 측에 (즉, 카메라 측에) HTTP 서버를 배치한다.

[0006] Lottermann에서 설명된 시스템의 문제점은 업링크 비디오 스트림을 생성하기를 원하는 많은 사용자가 방화벽에 의해, 또는 예를 들어, 동적 호스트 구성 프로토콜(Dynamic Host Configuration Protocol, DHCP)를 사용하여 할당된 동적 IP 어드레스와 결합한 네트워크 어드레스 변환기(network address translator, NAT) 사용과 같은 다른 보안 특징에 의해 차단되고 보호된다는 점이다. 예를 들면, 가정용 네트워크 내의 디바이스는 일반적으로 가정용 네트워크 외부의 클라이언트가 그 디바이스에 대한 TCP 연결을 설정하지 못하게 하는 (즉, 예를 들어 TCP 또는 HTTP 연결을 사용하여 외부 디바이스로부터 가정 내부의 디바이스에 도달하는 것이 금지된) 방화벽에 의해 차폐된다. 이는 가정용 네트워크 외부에 있는 해커로부터 디바이스를 보호한다. 이 단점은 또한 이동 환경에서도 주어진다. 일반적으로, 운영자는 인터넷-지향 트래픽으로부터 이동 디바이스를 (예를 들면, 스마트폰) 차폐 및 보호하여 해킹, 서비스 거부, 및 스팸 공격으로부터 이동 디바이스를 보호하게 된다. 사용자만이 (즉, 이동 디바이스) 서버와의 세션을 초기화하거나 설정할 수 있다 (다른 방법은 없이).

**과제의 해결 수단**

[0007] 본 내용은 이러한 단점을 극복한 실시예를 설명한다. 실시예는 제안된 바와 같이 카메라 측에서 HTTP 서버를 위치시키는 것과 반대로 HTTP 클라이언트가 클라이언트 디바이스 측에 (즉, 카메라 측) 위치하는 새로운 라이브 업링크 비디오 해결법을 제공한다. 이는 기존에 널리 이용가능한 HTTP 라이브러리 및 인프라구조의 사용을 여전히 가능하게 하면서, 상기에 설명된 방화벽 문제점을 해결한다. TCP를 통한 HTTP는 확실하고 탄력적인 운송 메커니즘을 제공한다. 탄력성은 세션 처리량이 이용가능한 링크 비트비율에 가능한한 빨리 조정되도록 보장한다. 라이브 소스는 미디어 프레임을 연속적으로 생성하므로, 가능한한 빨리 업로드되어야 한다.

[0008] 여기서 설명되는 실시예에서, 클라이언트 디바이스에서 운행되는 (또는 클라이언트 디바이스에 비교적 근접하여 위치하는) HTTP 클라이언트는 HTTP 요청을 사용하여 HTTP 취합 서버에 연결을 (예를 들면, HTTPS 연결) 설정한다. 라이브 업링크 미디어는 (오디오 및/또는 비디오) 이때 HTTP 요청의 HTTP 바디 내에 제공된다. HTTP 클라이언트는 HTTP 1.0 원칙을 사용하여 미디어 콘텐츠를 직접 HTTP 바디에 파이핑(piping)하거나, HTTP 1.1 청크 전달 인코딩(Chunked Transfer Encoding)을 사용할 수 있다. HTTP 청크 전달 인코딩은 클라이언트가 이어지는 HTTP 트랜잭션을 위해 설정된 TCP 연결을 (지속적인 TCP 연결) 재사용하도록 허용한다. 따라서, 여기서 설명되는 실시예는 HTTP 클라이언트로부터 HTTP 서버로의 라이브 업링크 스트림을 위한 HTTP 요청 바디의 사용에 관련된다. 무엇보다도, 이는 전송 비트비율을 이용가능한 링크 비트비율로 조정하도록 HTTP 세션의 신뢰성 및 탄력성을 활용할 수 있게 한다.

[0009] 실시예는 탄력적인 비율 적응 구조의 진행을 모니터링하는 새로운 비율 적응 구조를 제공하고, 클라이언트 전송-버퍼가 지연 제한 내에서 라이브 비디오 피드(feed)를 업로딩하고 또한/또는 너무 많은 지터(jitter)를 부가하지 않게 보장한다.

[0010] 실시예는 새로운 라이브 비디오 해결법을 수용할 수 있는 새로운 HTTP 서버 기능을 제공한다. 실시예에서, 클라이언트는 HTTP 요청 내에서 라이브 업링크 비디오를 시작할 수 있다. HTTP 서버는 이때 수신된 데이터 청크를 후처리 기능에 파이핑 할 수 있다 (예를 들면, 이들이 수신된 대로 모든 청크를 수신하기 전에).

[0011] 이후 설명되는 실시예는 라이브 업링크 스트리밍을 수행하도록 프래그먼트(fragment)화된 MP4(fragmented MP4, fMP4) 및 HTTP 청크 전달 인코딩과 결합되어 HTTP를 사용하는 방법을 설명한다. 실시예는 비트비율 적응 라이브 업링크 해결법을 제공하고, 이는 TCP-레벨 재전송 (즉, 별도의 UDP 재전송 서버가 필요하지 않은), 비율 제어, 및 혼잡 방지 원칙을 이용할 수 있지만, 여전히 미디어 비트비율 적응을 수행할 수 있다.

[0012] 실시예는 MPEG 운송 스트림(MPEG Transport Stream, MPEG2-TS), MPEG 미디어 운송(MPEG Media Transport, MMT), 또는 독립적인 RTMP와 같은 접근법에 대한 대안으로서 동작할 수 있다.

[0013] 또한, 디지털(digit) 버퍼 및 ABR 트랜스코더/패키저와 같이, 임의의 후처리 기능으로 요청으로부터의 임의의 수신 데이터 청크를 파이핑하기 시작하도록 (즉, 요청으로부터 모든 데이터 청크를 수신하기 전에) 기존 HTTP 서버를 확장하는 것에 관련된 실시예가 설명된다.

[0014] 그에 따라, 한 측면으로 미디어 소스에 (예를 들면, 카메라) 의해 생성되는 라이브 미디어 (오디오 및/또는 비

디오) 피드를 서버에 업스트리밍하기 위해 클라이언트 장치에 ("클라이언트") 의해 실행되는 방법이 제공된다. 한 실시예에서, 그 방법은 클라이언트가 서버와 운송 계층 연결을 (예를 들면, TCP 연결) 설정하는 단계를 포함한다. 방법은 또한 클라이언트가 헤더(header) 및 바디(body)를 포함하는 하이퍼텍스트 전달 프로토콜(HTTP) 요청 메시지를 운송 계층 연결을 통해 서버에 전송하는 단계를 포함하고, 여기서 HTTP 요청 메시지의 헤더는 HTTP 요청 메시지의 바디의 크기를 나타내지 않는다. 방법은 또한 미디어 데이터가 발생될 때 클라이언트가 라이브 미디어 피드에 대응하는 미디어 데이터를 전송 버퍼에 저장하는 단계를 포함하고, 품질 설정은 미디어 데이터를 발생하는데 사용된다. 운송 계층 연결을 통해 서버에 HTTP 요청 메시지의 바디를 전송하는 단계는: 1) 클라이언트가 전송 버퍼에 현재 저장된 미디어 데이터의 적어도 일부를 운송 계층 연결을 통해 서버에 전송하는 단계; 2) 클라이언트가 미디어 데이터의 상기 적어도 일부를 전송 버퍼에서 제거하는 단계; 3) 라이브 미디어 피드에 대응하는 미디어 데이터를 발생하는데 사용되는 품질 설정을 클라이언트가 수정하여야 하는가 여부를 클라이언트가 결정하는 단계; 및 4) 전송 중지 트리거가 검출될 때까지 클라이언트가 단계(1), (2), 및 (3)을 반복하는 단계를 포함한다.

[0015] 일부 실시예에서, 미디어 소스는 카메라이다. 일부 실시예에서, 카메라는 클라이언트의 필수 부분이고; 다른 실시예에서, 카메라는 클라이언트로부터 떨어져있다 (예를 들면, 블루투스 연결과 같은 링크를 통해 클라이언트에 연결된 드론 상의 카메라). 실시예에서, 트리거는 지속 시간을 지정하는 스케줄을 기반으로 하고, 클라이언트는 지속 시간이 경과했음을 검출한 결과로 트리거를 검출한다.

[0016] 일부 실시예에서, 방법은 클라이언트가 미디어 소스에 의해 발생된 제1 세트의 미디어 프레임을 인코딩하여 제1 세트의 인코딩 미디어 프레임을 생성하는데 제1 품질 설정을 사용하는 단계를 더 포함한다. 전송 버퍼에 저장된 미디어 데이터는 제1 세트의 인코딩 미디어 프레임을 포함하고; HTTP 요청 메시지의 바디를 전송하는 단계는 클라이언트가 제1 코덱 구성 정보를 전송하는 단계를 더 포함하고, 여기서 제1 코덱 구성 정보는 제1 품질 설정을 나타내는 정보를 포함한다. 방법은 클라이언트가 전송 버퍼에 저장된 데이터의 양을 모니터링하는 단계; 클라이언트가 전송 버퍼에 저장된 데이터의 양이 한계치를 초과함을 결정하는 단계; 및 전송 버퍼에 저장된 데이터의 양이 한계치를 초과함을 결정한 결과로, 클라이언트가 미디어 소스에 의해 발생된 제2 세트의 미디어 프레임을 인코딩하여 제2 세트의 인코딩 미디어 프레임을 생성하는데 제2 품질 설정을 사용하는 단계를 더 포함한다. 방법은 클라이언트가 라이브 미디어 피드에 대응하는 미디어 데이터를 전송 버퍼에 저장하는 단계를 더 포함하고, 여기서 미디어 데이터는 제2 세트의 인코딩 미디어 프레임을 더 포함한다. HTTP 요청 메시지의 바디를 전송하는 단계는 클라이언트가 제2 코덱 구성 정보를 전송하는 단계를 더 포함하고, 여기서 제2 코덱 구성 정보는 제2 품질 설정을 나타내는 정보를 포함한다.

[0017] 일부 실시예에서, 방법은 클라이언트가 미디어 소스에 의해 발생된 제1 세트의 미디어 프레임을 인코딩하여 제1 세트의 인코딩 미디어 프레임을 생성하는 단계로, 여기서 전송 버퍼에 저장된 미디어 데이터는 제1 세트의 인코딩 미디어 프레임을 포함하는 단계; 클라이언트가 전송 버퍼에 저장된 데이터의 양을 모니터링하는 단계; 클라이언트가 전송 버퍼에 저장된 데이터의 양이 한계치를 초과함을 결정하는 단계; 및 전송 버퍼에 저장된 데이터의 양이 한계치를 초과함을 결정한 결과로, 클라이언트가 제1 세트의 미디어 프레임의 적어도 한 서브세트를 전송 버퍼로부터 드로핑(dropping)하여 드로핑된 프레임이 서버로 전송되지 않게 하는 단계를 더 포함한다. 일부 실시예에서, 상기 전송 버퍼는 미디어 프레임의 상기 서브세트를 포함하고 미디어 프레임의 서브세트에 대한 메타 데이터를 더 포함하는 미디어 프래그먼트(media fragment)를 저장하고, 미디어 프레임의 상기 서브세트를 전송 버퍼로부터 드로핑하는 단계는 전송 버퍼로부터 미디어 프래그먼트를 드로핑하여 미디어 프래그먼트가 서버에 전송되지 않게 하는 단계를 포함한다.

[0018] 일부 실시예에서, 방법은 클라이언트가 미디어 소스에 의해 발생된 비압축 비디오 프레임을 획득하는 단계; 클라이언트가: i) 인코딩된 비디오 프레임 및 ii) 인코딩된 비디오 프레임에 관련된 메타-데이터를 포함하는 비디오 프래그먼트를 발생시키는 단계; 및 클라이언트가 프래그먼트를 전송 버퍼에 저장하는 단계를 더 포함한다. 실시예에서, 비디오 프래그먼트는: i) ISO-BMFF 비디오 프래그먼트 및 ii) 공통 미디어 애플리케이션 포맷(Common Media Application Format, CMAF) 비디오 프래그먼트 중 하나이다.

[0019] 일부 실시예에서, 라이브 피드는 라이브 배급을 위한 수신 엔터티(서버)에 의해 더 처리될 수 있다. 실시예에서, 클라이언트가 전송 버퍼에 현재 저장된 미디어 데이터의 적어도 일부를 운송 계층을 통해 서버에 전송하는 단계는 전송 버퍼에 현재 저장된 미디어 데이터의 적어도 일부를 포함하는 데이터 청크를 전송하는 단계를 포함한다. 실시예에서, 클라이언트가 라이브 미디어 피드에 대응하는 미디어 데이터를 발생시키는데 사용되는 품질 설정을 수정하여야 하는가 여부를 결정하는 상기 단계는 전송 버퍼 레벨 및/또는 시간에 걸친 전송 버퍼

레벨에서의 변화를 (예를 들면, 버퍼 레벨 경사도) 기반으로 한다.

[0020] 또 다른 측면으로, 미디어 소스에 의해 생성되는 라이브 미디어 (오디오 및/또는 비디오) 피드를 클라이언트로부터 수신하기 위해 서버에 의해 실행되는 방법이 제공된다. 그 방법은 클라이언트와의 운송 계층 연결을 (예를 들면, TCP 연결) 통해 바디를 포함하는 HTTP 요청 메시지에 대한 하이퍼텍스트 전달 프로토콜(HTTP) 헤더를 수신하는 단계를 포함한다. HTTP 헤더는 HTTP 요청 메시지의 바디의 크기를 나타내지 않는다. 방법은 HTTP 헤더를 수신한 이후에, HTTP 요청 메시지의 바디를 수신하는 단계를 더 포함한다. HTTP 요청 메시지의 바디를 수신하는 단계는 HTTP 클라이언트로부터 제1 데이터 청크를 수신하는 단계; HTTP 클라이언트로부터 제2 데이터 청크를 수신하는 단계; 및 제1 데이터 청크를 수신한 이후이고 제2 데이터 청크를 수신하기 이전에, 스트리밍 비디오를 배급하기 위한 배급 시스템에 제1 데이터 청크를 전달하는 단계를 포함한다.

[0021] 또 다른 측면으로, 미디어 소스에 의해 생성되는 라이브 미디어 (오디오 및/또는 비디오) 피드를 서버에 업스트리밍하기 위한 HTTP 클라이언트가 (예를 들면, 이동 디바이스 또는 사용자 장비(UE)에서) 제공된다. HTTP 클라이언트는 서버와의 운송 계층 연결을 (예를 들면, TCP 연결) 설정하고; 서버에 운송 계층 연결을 통해 헤드 및 바디를 포함하는 하이퍼텍스트 전달 프로토콜(HTTP) 요청 메시지를 전송하고, 여기서 HTTP 요청 메시지의 헤더는 HTTP 요청 메시지의 바디의 크기를 나타내지 않고; 또한 미디어 데이터가 발생될 때 라이브 미디어 피드에 대응하는 미디어 데이터를 전송 버퍼에 저장하도록 적용된다. 품질 설정은 미디어 데이터를 발생시키는데 사용된다. HTTP 요청 메시지의 바디를 서버에 운송 계층 연결을 통해 전송하는 단계는: 1) 클라이언트가 전송 버퍼에 현재 저장된 미디어 데이터의 적어도 일부를 운송 계층 연결을 통해 서버에 전송하는 단계; 2) 클라이언트가 미디어 데이터의 상기 적어도 일부를 전송 버퍼로부터 제거하는 단계; 3) 라이브 미디어 피드에 대응하는 미디어 데이터를 발생하는데 사용되는 품질 설정을 클라이언트가 수정하여야 하는가 여부를 클라이언트가 결정하는 단계; 및 4) 전송 중지 트리거가 검출될 때까지 클라이언트가 단계(1), (2), 및 (3)을 반복하는 단계를 포함한다.

[0022] 또 다른 측면으로, 미디어 소스에 의해 생성되는 라이브 미디어 (오디오 및/또는 비디오) 피드를 서버에 업스트리밍하기 위한 HTTP 클라이언트가 (예를 들면, 이동 디바이스 또는 사용자 장비(UE)에서) 제공된다. HTTP 클라이언트는 서버와의 운송 계층 연결을 (예를 들면, TCP 연결) 설정하도록 구성된 운송 모듈; 서버에 운송 계층 연결을 통해 헤드 및 바디를 포함하는 하이퍼텍스트 전달 프로토콜(HTTP) 요청 메시지를 전송하도록 구성된 전송 모듈, 여기서 HTTP 요청 메시지의 헤더는 HTTP 요청 메시지의 바디의 크기를 나타내지 않고; 또한 미디어 데이터가 발생될 때 라이브 미디어 피드에 대응하는 미디어 데이터를 전송 버퍼에 저장하도록 구성된 데이터 저장 모듈을 포함한다. 품질 설정은 미디어 데이터를 발생시키는데 사용된다. HTTP 요청 메시지의 바디를 서버에 운송 계층 연결을 통해 전송하는 단계는: 1) 클라이언트가 전송 버퍼에 현재 저장된 미디어 데이터의 적어도 일부를 운송 계층 연결을 통해 서버에 전송하는 단계; 2) 클라이언트가 미디어 데이터의 상기 적어도 일부를 전송 버퍼로부터 제거하는 단계; 3) 라이브 미디어 피드에 대응하는 미디어 데이터를 발생하는데 사용되는 품질 설정을 클라이언트가 수정하여야 하는가 여부를 클라이언트가 전송 버퍼 레벨 및/또는 시간에 걸친 전송 버퍼 레벨에서의 변화를 (예를 들면, 버퍼 레벨 경사도) 기반으로 결정하는 단계; 및 4) 전송 중지 트리거가 검출될 때까지 클라이언트가 단계(1), (2), 및 (3)을 반복하는 단계를 포함한다.

[0023] 또 다른 측면으로, 미디어 소스에 의해 생성되는 라이브 미디어 (오디오 및/또는 비디오) 피드를 서버에 업스트리밍하기 위한 HTTP 클라이언트가 (예를 들면, 이동 디바이스 또는 사용자 장비(UE)에서) 제공된다. HTTP 클라이언트는 수신기; 전송기; 데이터 저장 시스템; 및 프로세서를 포함하는 데이터 프로세싱 장치를 포함한다. 데이터 프로세싱 장치는 데이터 저장 시스템, 전송기, 및 수신기에 연결되고, 데이터 프로세싱 장치는: 서버와의 운송 계층 연결을 (예를 들면, TCP 연결) 설정하고; 서버에 운송 계층 연결을 통해 헤드 및 바디를 포함하는 하이퍼텍스트 전달 프로토콜(HTTP) 요청 메시지를 전송하고, 여기서 HTTP 요청 메시지의 헤더는 HTTP 요청 메시지의 바디의 크기를 나타내지 않고; 또한 미디어 데이터가 발생될 때 라이브 미디어 피드에 대응하는 미디어 데이터를 전송 버퍼에 저장하도록 구성된다. 품질 설정은 미디어 데이터를 발생시키는데 사용된다. HTTP 요청 메시지의 바디를 서버에 운송 계층 연결을 통해 전송하는 단계는: 1) 클라이언트가 전송 버퍼에 현재 저장된 미디어 데이터의 적어도 일부를 운송 계층 연결을 통해 서버에 전송하는 단계; 2) 클라이언트가 미디어 데이터의 상기 적어도 일부를 전송 버퍼로부터 제거하는 단계; 3) 라이브 미디어 피드에 대응하는 미디어 데이터를 발생하는데 사용되는 품질 설정을 클라이언트가 수정하여야 하는가 여부를 클라이언트가 결정하는 단계; 및 4) 전송 중지 트리거가 검출될 때까지 클라이언트가 단계(1), (2), 및 (3)을 반복하는 단계를 포함한다.

[0024] 또 다른 측면으로, 미디어 소스에 의해 생성되는 라이브 미디어 (오디오 및/또는 비디오) 피드를 클라이언트로부터 수신하는 HTTP 서버가 제공된다. HTTP 서버는: 클라이언트와의 운송 계층 연결을 (예를 들면, TCP 연결)

통해 바디를 포함하는 HTTP 요청 메시지에 대한 하이퍼텍스트 전달 프로토콜(HTTP) 헤더를 수신하도록 적용된다. HTTP 헤더는 HTTP 요청 메시지의 바디의 크기를 나타내지 않는다. HTTP 서버는 HTTP 헤더를 수신한 이후에, HTTP 요청 메시지의 바디를 수신하도록 더 적용된다. HTTP 요청 메시지의 바디를 수신하는 단계는: HTTP 클라이언트로부터 제1 데이터 청크를 수신하는 단계; HTTP 클라이언트로부터 제2 데이터 청크를 수신하는 단계; 및 제1 데이터 청크를 수신한 이후이고 제2 데이터 청크를 수신하기 이전에, 스트리밍 비디오를 배급하기 위한 배급 시스템에 제1 데이터 청크를 전달하는 단계를 포함한다.

[0025] 또 다른 측면으로, 미디어 소스에 의해 생성되는 라이브 미디어 (오디오 및/또는 비디오) 피드를 클라이언트로부터 수신하기 위한 HTTP 서버가 제공된다. HTTP 서버는 클라이언트와의 운송 계층 연결을 (예를 들면, TCP 연결) 통해 바디를 포함하는 HTTP 요청 메시지에 대한 하이퍼텍스트 전달 프로토콜(HTTP) 헤더를 수신하도록 구성된 제1 수신 모듈을 포함하고, 여기서 HTTP 헤더는 HTTP 요청 메시지의 바디의 크기를 나타내지 않고; 제1 수신 모듈에 의해 HTTP 헤더를 수신한 이후에, HTTP 요청 메시지의 바디를 수신하도록 구성된 제2 수신 모듈을 포함한다. HTTP 요청 메시지의 바디를 수신하는 단계는: HTTP 클라이언트로부터 제1 데이터 청크를 수신하는 단계; HTTP 클라이언트로부터 제2 데이터 청크를 수신하는 단계; 및 제1 데이터 청크를 수신한 이후이고 제2 데이터 청크를 수신하기 이전에, 스트리밍 비디오를 배급하기 위한 배급 시스템에 제1 데이터 청크를 전달 모듈에 의해 전달하는 단계를 포함한다.

[0026] 또 다른 측면으로, 미디어 소스에 의해 생성되는 라이브 미디어 (오디오 및/또는 비디오) 피드를 클라이언트로부터 수신하기 위한 HTTP 서버가 제공된다. HTTP 서버는 수신기; 전송기; 데이터 저장 시스템; 및 프로세서를 포함하는 데이터 프로세싱 장치를 포함하고, 여기서 데이터 프로세싱 장치는 데이터 저장 시스템에 연결된다. 전송기, 수신기, 및 데이터 프로세싱 장치는: 클라이언트(102)와의 운송 계층 연결을 (예를 들면, TCP 연결) 통해 바디를 포함하는 HTTP 요청 메시지에 대한 하이퍼텍스트 전달 프로토콜(HTTP) 헤더를 수신하고, 여기서 HTTP 헤더는 HTTP 요청 메시지의 바디의 크기를 나타내지 않고; 또한 HTTP 헤더를 수신한 이후에, HTTP 요청 메시지의 바디를 수신하도록 구성된다. HTTP 요청 메시지의 바디를 수신하는 단계는: HTTP 클라이언트로부터 제1 데이터 청크를 수신하는 단계; HTTP 클라이언트로부터 제2 데이터 청크를 수신하는 단계; 및 제1 데이터 청크를 수신한 이후이고 제2 데이터 청크를 수신하기 이전에, 스트리밍 비디오를 배급하기 위한 배급 시스템에 제1 데이터 청크를 전달하는 단계를 포함한다.

[0027] 이후에는 상기 및 다른 실시예가 설명된다.

**도면의 간단한 설명**

[0028] 본 명세서에 포함되고 그 일부를 형성하는 첨부 도면은 다양한 실시예를 설명한다.

- 도 1은 일부 실시예에 따른 네트워크 설계를 설명한다.
- 도 2는 일부 실시예에 따라 프래그먼트로 분할된 미디어 스트림을 설명한다.
- 도 3은 일부 실시예에 따른 네트워크 설계를 설명한다.
- 도 4는 일부 실시예에 따라 버퍼 크기와 인코딩 품질 사이의 관계를 도시하는 그래프이다.
- 도 5는 일부 실시예에 따른 프로세스를 설명하는 흐름도이다.
- 도 6은 일부 실시예에 따른 프로세스를 설명하는 흐름도이다.
- 도 7은 일부 실시예에 따른 프로세스를 설명하는 흐름도이다.
- 도 8은 일부 실시예에 따라 HTTP 클라이언트의 기능적 모듈을 도시하는 도면이다.
- 도 9는 일부 실시예에 따라 HTTP 서버의 기능적 모듈을 도시하는 도면이다.
- 도 10은 일부 실시예에 따른 HTTP 클라이언트의 블록도이다.
- 도 11은 일부 실시예에 따른 HTTP 서버의 블록도이다.

**발명을 실시하기 위한 구체적인 내용**

[0029] 도 1은 예시적인 실시예의 종단간 설계를 설명한다. 도시된 바와 같이, 클라이언트 장치(102)는 (또는 줄여서 "클라이언트(102)") HTTP 서버(106)를 (또한 취합 서버(ingest server)라 칭하여지는) 포함하는 미디어 프로세

싱 유닛(104)과 통신한다. 이는 기여 링크 (일명 "취합 링크"), 즉 클라이언트(102)로부터 (즉 미디어 소스) 미디어 프로세싱 유닛(104)으로의 (즉 미디어 싱크(media sink)) 링크를 형성한다. 미디어 프로세싱 유닛(104)은 트랜스코더-패키저(transcoder-packager)(108)를 (예를 들면, 적응 비트비율(Adaptive Bitrate, ABR) 트랜스코더) 더 포함할 수 있다. 선(107)으로 나타내진 바와 같이, HTTP 서버(106)(진입부)는 트랜스코더-패키저(108)(출구)로부터 분리될 수 있다; 즉, 미디어 프로세싱 유닛(104)은 일부 실시예에서 단일 물리적 유닛이 될 수 있고, 일부 실시예에서는 유닛(104)이 분리된 물리적 디바이스 상에 놓이는 서버 및 트랜스코더-패키저 기능을 포함할 수 있다. 트랜스코더-패키저(108)는 패키지화된 미디어를 단말 사용자 수신기(112)에 더 전달하는 콘텐츠 배급 네트워크(content distribution network, CDN)(110)와 통신할 수 있다. 트랜스코더-패키저(180), CDN(110), 및 수신기(112)는 배급 시스템을 포함한다.

[0030] 전체 내용에 걸쳐 카메라라는 용어가 사용되지만, 임의의 미디어 소스가 또한 사용될 수 있음을 이해하여야 한다 (예를 들면, 마이크로폰).

[0031] 실시예는 카메라로부터 (예를 들면, 이동 카메라) 취합 서버로의 기여 링크에 대한 개선을 제공한다. 기여 링크는 콘텐츠 준비 파이프라인의 업링크이다. 개념적으로, 이는 예를 들어, CDN을 통한 DASH를 사용하여, 배급 준비를 위한 스크리닝(screening), ABR 트랜스코딩, 및 패키징 기능을 포함할 수 있다; 대안적으로 이들 기능 일부 또는 모두가 기여 링크로부터 분리되어 배급 시스템의 일부로 고려될 수 있다.

[0032] 취합을 위해, HTTP 취합 서버 해결법이 제안된다. HTTP 취합 서버(106)는 (임의의 다른 웹 서버와 같이) 입력을 청취한다. 일반적으로, HTTP 트래픽에 대해서는 TCP 포트 80 또는 443이 사용된다. HTTP 취합 서버(106)는 안전한 HTTPS 연결이 설정되기 이전에, 클라이언트의 인증 및 인증을 위한 보안 기능을 지원할 수 있다. 모든 HTTP 트랜잭션이 보안될 수 있으므로, 여기서 HTTP라는 용어는 HTTPS와 동의어로 사용된다.

[0033] HTTP 취합 서버(106)는 완전한 HTTP 요청이 서버에 의해 수신되기 이전에, 이어지는 프로세싱 기능으로 (예를 들면, ABR 트랜스코딩 및 패키징) 데이터를 (수신된 대로) 즉시 전달하는 방식으로 구현될 수 있다. 이 동작은 요청의 프로세싱이 트리거되기 이전에 HTTP 서버가 먼저 HTTP 요청의 모든 데이터를 수신함을 요구할 수 있는 종래 기술의 해결법과 대조적이다. 현재 설명되는 실시예에서, HTTP 진입부 서버(106)는 수신된 HTTP 청크를 ABR 트랜스코더-패키저(108)와 같은 이어지는 기능에 파이핑할 수 있다. 다른 프로세싱 기능이 또한 이 파이프라인(pipeline)에 사용될 수 있다. 예를 들면, ABR 트랜스코더로의 연속적인 입력을 보장하도록 ABR 트랜스코더 이전에 디-지터(de-jitter) 버퍼가 또한 위치할 수 있다 (도 3의 디-지터 버퍼(314)와 같이). 후처리 기능은 HTTP 취합 서버(106)와 함께 위치하거나 분산될 수 있다. 분산되는 경우, HTTP 취합 서버(106)는 예를 들어, HTTP 또는 다른 프로토콜을 사용하는 네트워크를 통해 후처리 기능으로 데이터를 파이핑할 수 있다.

[0034] 취합 이후에, ABR 트랜스코더 및 패키징 기능은 배급을 위한 콘텐츠를 준비한다. 트랜스코딩 매개변수는 클라이언트(102)에 의해 영향을 받을 수 있고 클라이언트로부터 취합까지의 업링크 품질에 의존할 수 있다.

[0035] HTTP 취합 서버(106)는 수신된 HTTP 요청 중 바디 부분을 처리함으로써 (예를 들면, 청크 전달 또는 비제한 바디 크기를 사용하여) 비디오를 취합할 수 있다. 비제한 바디 크기를 사용하는 경우, 클라이언트(102)는 업로드를 종료하기 위해 TCP 연결을 종료시킬 수 있다. 청크 전달을 사용하는 경우, 클라이언트(102)는 업로드를 중단하고 이어지는 트랜잭션을 위해 (예를 들면, 비디오의 해상도를 변경하기 위해) TCP 연결을 재사용할 수 있다.

[0036] 도 2는 비디오가 HTTP 취합 서버(106)에 의한 취합을 용이하게 하도록 분할될 수 있는 방법을 설명한다. 도시된 예에서, 클라이언트(102)는 인코딩 비디오 프레임의 연속적인 스트림을 생성하고 있다. 하나 이상의 이러한 프레임은 단일 프래그먼트로 (예를 들면, 공통 미디어 애플리케이션 포맷(Common Media Application Format, CMAF) 청크) (예를 들면, 프래그먼트(210, 212, 214, 216)) 수집될 수 있다. CMAF 청크에 인트라-프레임(intra-frame)이 포함될 필요는 없다. CMAF 청크는, 도시된 바와 같이, 적어도 moov 박스 및 mdat 박스를 포함한다. 코덱 구성을 위해, 초기화 세그먼트(202)가 (moov 박스를 포함하는) 선행된다.

[0037] 도 2에 도시된 바와 같이, 정의되지 않은 길이의 HTTP 바디를 갖는 HTTP 1.0 타입의 실현이 사용된다. 또 다른 실현은 HTTP 청크 전달 인코딩을 사용하는 것이다. HTTP 청크는 (예를 들어, HTTP 1.1 사양서에 의해 정의된 바와 같이) 하나 이상의 프래그먼트를 (또는 CMAF 청크) 포함할 수 있음을 주목하여야 한다. 일부 실시예에서 단일 프래그먼트가 단일 HTTP 청구에 포함되지만, HTTP 청크와 프래그먼트 사이에 의존성은 없다.

[0038] 도 3은 일부 실시예에 따른 클라이언트(102) 및 HTTP 취합 서버(106)를 설명한다.

[0039] 클라이언트(102)는 소정의 프레임 비율로 압축되지 않은 프레임의 연속적인 시퀀스를 제공하는 카메라(302)를

포함한다. 인코더(304)는 이어서 압축되지 않은 프레임의 시퀀스를 입력으로 취하여 압축된 프레임의 시퀀스를 출력으로 생성한다. 압축된 프레임은 화상 그룹(group of picture, GOP) 구조 내에서 구성된다. 패키지(306)는 이어서 하나 이상의 프레임들 (GOP 구조를 갖는) 수집하고 이들을 프래그먼트로 패키징한다. 프래그먼트는 CMAF 청크 (예를 들면, 적어도 하나의 무비 프래그먼트 박스(Movie Fragment Box)(moof) 및 하나의 미디어 데이터 박스(Media Data Box)(mdat) 섹션 (선택적인 다른 ISO-BMFF 박스를 갖는)), ISO-BMFF 프래그먼트, 또는 다른 적절한 구조가 될 수 있다. 결과로, 각 프래그먼트는 (특정한 시간의 (수 초의) 압축 미디어 데이터를 포함하는) 특정한 크기를 갖는다. 패키지(306)는 이어서 프래그먼트 내의 압축 미디어 데이터의 양에 따라 고정된 간격을 갖는 프래그먼트를 출력한다.

[0040] 패키지(306)는 프래그먼트를 (예를 들면, CMAF 청크) 버퍼(308)에 기록하고, 이는 버퍼 모니터(309)에 의해 모니터링 된다. HTTP 요청과 HTTP 연결을 오픈한 HTTP 클라이언트(310)는 HTTP 요청의 바디를 생성하기 위해 버퍼(308)로부터 프래그먼트의 바이트를 (예를 들면, CMAF 청크) 수신한다. 버퍼 모니터(309)는 버퍼(308)의 배출을 포함하여, 업로드 프로세스를 모니터링 한다. 버퍼 레벨이 (또는 크기) 증가되면 (즉, 버퍼 내의 데이터의 양이 한계치에 이를 때), 모니터(309)는 인코딩 프로세스를 수정하거나 다른 조치를 취하도록 제어 명령을 트리거한다. 도시된 바와 같이, 이는 인코더(304), 패키지(306), 및 모니터(309) 사이에서 피드백 루프의 형태를 취할 수 있다. HTTP 클라이언트(310)는 데이터를 (예를 들면, 바이트 또는 HTTP 청크) HTTP 취합 서버(106)에 송신한다.

[0041] 클라이언트(102)는 또한 트리거 소스(311)를 포함할 수 있다. 트리거 소스는 일부 실시예에서 스케줄러(scheduler)가 될 수 있다 (예를 들면, 미디어 피드의 시작 및 종단을 트리거하거나, 특정한 시간에 또한/또는 간격으로, 미디어 피드의 시작을 트리거하는). 일부 실시예에서, 트리거 소스(311)는 버퍼 모니터(309)로부터 올 수 있다 (예를 들면, 해상도가 변했거나 변할 것으로 기대된다는 표시). 비록 트리거 소스(311)가 클라이언트(102)의 일부로 도시되지만, 트리거 소스(311)는 또한 일부 실시예에서 클라이언트(102) 외부에 있을 수 있다.

[0042] 클라이언트(102)의 모듈(302 내지 311)은 동일한 하우징 내에 모두 수납되거나 분리된 하우징에 수납될 수 있다. 예를 들어, 모듈(302 내지 311)은 모두 이동 통신 디바이스(mobile communication device, MCD)의 (예를 들어 스마트폰, 태블릿, 랩탑 컴퓨터) 구성성분이고 MCD의 하우징 내에 수납될 수 있다. 또 다른 예로, 카메라(302)는 제1 디바이스의 (예를 들면, 드론) 일부가 되고, 다른 구성성분(304 내지 311)은 분리된 디바이스의 (예를 들어 스마트폰, 랩탑, 태블릿) 일부가 될 수 있다. 이러한 실시예에서, 무선 링크는 카메라(302)를 인코더(304)와 통신되게 연결할 수 있다. 일부 실시예에서, 클라이언트(102)는 패키지(306)를 포함하지 않는다.

[0043] 도시된 실시예에서, HTTP 취합 서버(106)는 HTTP 서버(312) 및 디-지터 버퍼(314)를 포함한다.

[0044] 도 4는 일부 실시예에서 버퍼 모니터(309)가 동작될 수 있는 방법의 예시적인 설명을 도시한다. 수직축을 따라, 인코딩 품질은 "프레임 드로핑(frame dropping)"에서 "저품질(low quality)"(LQ), "중간 품질(medium quality)"(MQ), 및 "고품질(high quality)"(HQ)로 나타난다. 수평축을 따라, 버퍼 크기가 (버퍼(308)에 저장된 미디어 신호를 millisecond로 표현하여) 0ms 내지 800ms의 범위를 나타낸다. 도시된 바와 같이, 버퍼(308) 크기가 제1 한계치 (여기서는 200ms) 이하이면, 인코딩 품질은 HQ이다. 버퍼(308)가 제1 한계치 이상이지만 제2 한계치 (여기서는 400ms) 이하로 채워지면, 인코딩 품질은 MQ로 저하될 수 있다. 버퍼(308)가 제2 한계치 이상이지만 제3 한계치 (여기서는 600ms) 이하로 채워지면, 인코딩 품질은 LQ로 더 저하될 수 있다. 버퍼(308)가 제3 한계치 이상으로 채워지면, 인코딩 품질은 프레임 드로핑을 사용함으로써 더 저하될 수 있다. 버퍼 모니터(309)는 만족스러운 라이브 스트리밍 QoS 매개변수를 유지하는데 특정한 인코딩 전략을 사용하도록 버퍼(308)의 크기를 모니터링한다. 버퍼(308)가 채워짐에 따라 인코딩의 품질이 저하되는 것에 부가하여, 반대의 프로세스가 (즉, 버퍼 크기가 감소함에 따라 인코딩의 품질이 개선되는) 또한 사용될 수 있다. 버퍼 모니터(309)는 현재의 버퍼(308)의 크기만을 고려할 수 있거나, 또는 시간의 함수로 버퍼 크기를 고려하여 적절한 인코딩 매개변수를 결정할 때 과거 버퍼 크기를 (또는 예측되는 미리 크기) 고려할 수 있다.

[0045] 방금 설명된 바와 같이, 고품질(HQ), 중간 품질(MQ), 및 저품질(LQ) 인코딩은 다른 코덱 구성을 참조하여 다른 시각적 품질을 제공할 수 있다. 일부 실시예에서, 버퍼 모니터(309)는 인코딩 비디오의 결과 비트비율을 변경하도록 인코더의 양자화 매개변수를 수정할 수 있다.

[0046] 버퍼 모니터(309)는, 일부 실시예에서, 버퍼(308)의 버퍼 레벨을 연속하여 체크한다. 버퍼(308)가 새로운 CMAF 청크를 버퍼에 추가할 때 비어 있으면, 모니터는 어떠한 동작도 취하지 않는다. 버퍼 레벨이 증가될 때 (압축

된 미디어 시간으로 나타내지는), 모니터는 제어 동작을 취한다. 제1 동작은 비디오 품질을 (예를 들면, 비트 비율) 감소시키는 것이 될 수 있다. 버퍼 레벨이 아직 감소되지 않을 때, 모니터는 버퍼(308)로부터 프레임 또는 CMAF 청크 또는 프래그먼트를 드로핑하기 시작할 수 있다. 각각의 프레임을 드로핑할 때, GOP의 끝부분으로부터의 프레임만이 드로핑되어야 한다. GOP는 화상 그룹(Group of Picture)으로, 여기서 종속적인 픽처의 세트가 독립적으로 디코딩 가능한 프레임과 함께 그룹화된다.

[0047] 각 프래그먼트가 단일 비디오 프레임까지의 (또는 ISO-BMFF 용어에서의 샘플) 몇 프레임과 같이, GOP의 일부만을 포함할 때, 버퍼 모니터(309)는 완전한 프래그먼트를 드로핑할 수 있다. 그렇게 할 때, 버퍼 모니터(309)는 또한 프레임 사이의 코딩 의존도를 고려할 수 있다 (드로핑 프레임이 랜덤하지 않지만, 선택적으로 또한 지능적으로 품질의 저하를 최소로 하게 실행되도록). 버퍼 모니터(309)는 또한 (예를 들어, 프래그먼트가 완전한 GOP를 포함하는 경우) 프래그먼트를 편집함으로써 (또는 재기록함으로써) 데이터를 제거할 수 있다. 예를 들어, 버퍼 모니터(309)는 미디어 데이터 박스(mdat)로부터 특정한 비디오 프레임의 데이터를 제거하고 그에 따라 박스가 줄어들 수 있다. 트랙 프래그먼트 런 박스(Track Fragment Run box, trun)는 샘플 타이밍 및 샘플 크기와 함께, 미디어 데이터 박스(mdat)에서의 모든 샘플에 (또는 비디오 프레임) 대한 엔트리를 포함한다. 샘플이 프래그먼트로부터 제거될 때, trun 박스는 예를 들어, trun 박스로부터 대응하는 샘플 엔트리를 제거하고 대응하는 양 만큼 trun 박스의 크기를 줄임으로서, 조정될 필요가 있다. 일부 실시예에서, 프래그먼트는 GOP의 일부만을 (GOP 전체가 아니라) 운반하여 버퍼 모니터(309)가 용이하게 완전한 프레임을 드로핑할 수 있도록 구성된다. 각 프래그먼트가 하나의 프레임만을 포함할 때, 버퍼 모니터(309)는 어떠한 프래그먼트 박스도 재기록하지 않고 각각의 프레임을 드로핑할 수 있다.

[0048] 다음의 GOP 구조를 갖는 것으로 가정한다 (GOP 크기 = 8 프레임): I1, B1, P1, B2, P2, B3, P3, B4. 이 표시에서, I1은 독립적으로 디코딩 가능한 화상이고, 일반적으로 I-프레임 또는 인트라(Intra)-프레임 또는 키(Key)-프레임 또는 스트림으로의 랜덤 액세스 포인트 또는 스트림으로의 서비스 액세스 포인트라 칭하여지고; P 화상은 예측된 화상으로, I-프레임 또는 다른 P 프레임에 의존하고; 또한 B-프레임은 양방향 예측 프레임으로, 적어도 두개의 다른 프레임에 의존한다 (I, P 또는 B든 상관없이). 다른 프레임 및 의존 프레임의 프레임 구성이 가능하다.

[0049] 프래그먼트가 단일 프레임을 (또는 화상) 포함할 때, 각 프레임은 프래그먼트로 패키징된다. 본 예에서, 버퍼 모니터(309)가 프레임을 드로핑하여야 하면, 프레임 B4가 먼저 드로핑되고, 이어서 각각 프레임 B3, B2, B1, P3, P2, P1으로 이어진다.

[0050] 다음의 GOP 구조를 갖는 것으로 가정한다 (GOP 크기 = 8 프레임): I1, P1, P2, P3, P4, P5, P6, P7.

[0051] 본 예에서, 버퍼 모니터(309)는 프레임 P7을 드로핑하여야 하고, P6, P5, P4, P3, P2, P1로 이어진다.

[0052] 일부 실시예에서, 프레임 드로핑은 프레임 의존도를 고려하여 좌측에서 우측으로 시작된다.

[0053] 제어 함수가 해상도 또는 다른 코딩 구조 매개변수를 변화시킬 때 (예를 들어, 시퀀스 매개변수 세트(Sequence Parameter Set, SPS) 또는 화상 매개변수 세트(Picture Parameter Set, PPS)가 수정되면), HTTP 클라이언트(310)는 하나의 업로드를 중단하고 새로운 초기 세그먼트로 시작될 필요가 있다.

[0054] 도 5는 HTTP 청크 전달 인코딩을 사용하는 연속적인 업로드에 대한 예시적인 호출 흐름도를 설명한다.

[0055] HTTP 클라이언트(310)는 요청 및 요청 헤더로 HTTP 세션을 오픈한다. 요청 바디에 대해, HTTP 클라이언트(310)는 HTTP 청크를 업로드하도록 버퍼(308)로부터 판독한다. 일부 실시예에서, 각 HTTP 청크는 CMAF 청크에 대응하지만, 기술적으로 정렬 요구사항은 없다.

[0056] HTTP 요청이 아직 진행중이므로, HTTP 서버(312)로부터 HTTP 응답이 없다. HTTP 클라이언트(310)가 0 크기의 HTTP 청크로 HTTP 바디를 종료할 때에만, 서버는 "200 OK" 또는 "201 Created"와 같은 HTTP 응답 코드로 응답하게 된다.

[0057] 청크 전달-인코딩을 사용하지 않는 실시예에서 (예를 들어, 여기서는 이용가능하지 않은 HTTP 1.0 트랜잭션), HTTP 클라이언트(310)는 TCP 연결을 종료할 필요가 있다.

[0058] 청크 전달-인코딩을 사용하는 실시예에서는 이어지는 HTTP 트랜잭션을 위해 기존 TCP 연결을 재사용하는 것이 가능하다 (도시된 바와 같이). HTTP 청크 전달-인코딩의 이점은 HTTP 바디가 HTTP 트랜잭션이 시작할 때 미지의 크기를 가질 수 있어 설정된 TCP 연결을 (지속적인 TCP 연결) 재사용하는 것이 가능하다는 점이다.

- [0059] 본 실시예에서, HTTP 클라이언트(310)가 라이브 소스로부터 ("무한적으로" 데이터를 생성하는) 입력 데이터를 수신하므로, HTTP 트랜잭션의 종료를 트리거하는 외부 이벤트가 필요하다. 외부 이벤트는 라이브 피드의 종료 시 또는 라이브 피드의 해상도 변경시 (예를 들면, 버퍼 크기가 너무 클 때 버퍼 모니터(309)에 의해 트리거될 때) 있을 수 있다. 해상도 변경의 경우, HTTP 클라이언트(310)는 새로운 코덱 정보 및 초기화 매개변수를 (예를 들어, 새로운 moov 박스) 송신할 필요가 있다. 이 경우, 버퍼 모니터(309)는 (외부 트리거로 작용하는) 코덱 구성을 (예를 들어, 비디오 해상도) 변경할 필요가 있음을 검출하게 되고, 인코딩 프로세스를 중단하고, HTTP 청크의 업링크 트래픽을 중단하도록 HTTP 클라이언트(310)를 트리거한다 (그러나, 설정된 TCP 연결을 계속 유지한다). 이때, 인코더 프로세스는 다시 시작되어야 하지만 (즉, 인코더(304)는 비디오 프레임의 생성을 재개하지만), 다른 해상도로 (또는 다른 코덱 구성 또는 다른 방법으로 다른 인코딩 매개변수로) 시작되어야 한다. 먼저, 인코더(304)는 코덱 구성을 포함하는 초기화 세그먼트를 (예를 들면, ISO BMFF moov 박스) 배출한다. 이어서, 인코더(304)는 패키지(306)를 통해 버퍼 모니터(309)의 업로드 버퍼에 압축된 비디오 프레임을 제공한다.
- [0060] 한 실시예에서, HTTP 클라이언트(310) 또는 버퍼 모니터(309)는 새로운 코덱 구성으로 인코더(304)를 재시작하기 이전에 버퍼(308)로부터 모든 데이터를 없앤다. 또 다른 실시예에서는 비디오 시간 라인 내의 갭을 방지하기 위해 버퍼(308)로부터의 오래된 데이터가 유지된다. 예를 들면, 업로드되었던 버퍼(308) 내의 임의의 데이터는 비디오 시퀀스 내의 프레임에 대응한다.
- [0061] 일부 인코더는 타겟 비트비율과 같은 코딩 구성에 대한 변경을 (예를 들면, 양자화 매개변수(quantization parameter, QP)의 변화를 통해) 허용하지 않을 수 있다. 그 경우, 클라이언트는 변경된 코덱 구성 매개변수로 인코더(304)를 시작하지만, 새로운 초기화 세그먼트를 (예를 들면, moov 박스) 생성하지는 않는다.
- [0062] 실시예는 사용-용도에 따라, 외부 트리거의 실현을 더 지원한다. 일부 사용-용도에서는 자체-비행 드론 또는 바디-캠 또는 감시 캠과 같이, 원격 작동자가 해상도 변경을 트리거할 수 있다 (예를 들어, 품질을 증가시키거나 버퍼 지연을 감소시키기 위해). 또 다른 외부 트리거는 프로그램된 스케줄이 될 수 있고, 여기서 카메라는 특정한 시간 동안만 또는 특정한 기간 동안만 기록하도록 프로그램된다 (즉, 중단 시간 트리거).
- [0063] 예를 들면, 혼잡 제어 및 재전송을 위해 서버로부터 TCP 레벨 트랜잭션이 있다.
- [0064] HTTP 청크 전달을 사용한 업링크 라이브 스트림에 대한 예시적인 HTTP 요청은 다음과 같다:
- [0065] POST /upload/xyz/live-session1.mp4 HTTP/1.1
- [0066] Host: 192.168.1.141:9030
- [0067] Transfer-Encoding: chunked
- [0068] Connection: keep-alive
- [0069] Accept-Encoding: gzip, deflate
- [0070] Accept: \*/\*
- [0071] User-Agent: python-requests/2.9.1
- [0072] 20
- [0073] ... ftypmp42....mp42mp41isomiso2
- [0074] 336
- [0075] ...6moov...lmvhd.....PKS.PKS.....  
 @.....Itrak...Wtkhd.....PKS.PK  
 S.....@.....@.....mdia...  
 mdhd.....PKS.PKS.....-  
 hdlr.....vide.....VideoHandler....7minf.....vmhd.....\$dinf.....dref.....url  
 .....stbl....stsd.....avc1.....@...H...  
 H.....lavC.d.....gd...AA.....B.`...h.....btr  
 t.....stts.....stsc.....stsz.....stco.....Yudta...Qmet  
 a.....!hdlr...mhlrmdir.....\$ilst....too....data.....x264...=udta...5meta.....!hdl

r...mhlrmdir.....ilst...<mvex...mehd..... trex.....

- [0076] 5dc
- [0077] ...Tmoof....mfhd.....
- [0078] 본 예에서, HTTP POST는 여기서 업링크 스트림을 나타내는 HTTP 방법으로 사용된다. 또 다른 해결법은 HTTP PUT을 사용하는 것이 될 수 있다. HTTP 요청에서의 HTTP URL은 (여기서, upload/xyz/live-session1.mp4) 업링크 스트리밍 세션의 타겟 채널을 나타낸다. URL은 계정별로 지정되고 보안이 유지되므로, 인증된 송신자만이 업로드할 수 있다. 계정 소유자는 어느 송신자가 데이터를 송신하도록 허용되었나를 인증한다.
- [0079] 본 예에서는 콘텐츠-길이 HTTP 헤더가 주어지지 않으므로, 미지의 HTTP 바디 크기를 나타낸다. HTTP 청크 전달 인코딩은 헤더에서 나타내지므로 (즉, "Transfer-Encoding: chunked"), HTTP 바디가 HTTP 청크의 시퀀스임을 나타낸다.
- [0080] 도 6은 프로세서(600)를 설명한다. 프로세서(600)는 예를 들어, 클라이언트에 (예를 들면, 클라이언트 장치 (102)) 의해 실행될 수 있다. 클라이언트는 서버와 운송 계층 연결을 (예를 들면, TCP 연결) 설정한다(단계 602). 클라이언트는 서버에 운송 계층 연결을 통해 헤더 및 바디를 포함하는 하이퍼텍스트 전달 프로토콜 (HTTP) 요청 메시지를 전송한다(단계 604). HTTP 요청 메시지의 헤더는 HTTP 요청 메시지의 바디의 크기를 나타내지 않는다. 클라이언트는 미디어 데이터가 발생될 때 라이브 미디어 피드에 대응하는 미디어 데이터를 전송 버퍼에 저장한다(단계 606). 품질 설정은 미디어 데이터를 발생하는데 사용된다. 운송 계층 연결을 통해 서버에 HTTP 요청 메시지의 바디를 전송하는 단계는: 1) 클라이언트가 전송 버퍼에 현재 저장된 미디어 데이터의 적어도 일부를 운송 계층 연결을 통해 서버에 전송하는 단계(단계 608); 2) 클라이언트가 미디어 데이터의 상기 적어도 일부를 전송 버퍼에서 제거하는 단계(단계 610); 3) 전송 버퍼 레벨 및/또는 시간에 걸친 전송 버퍼 레벨에서의 변화를 (예를 들면, 버퍼 레벨 경사도) 기반으로 라이브 미디어 피드에 대응하는 미디어 데이터를 발생하는데 사용되는 품질 설정을 클라이언트가 수정하여야 하는가 여부를 클라이언트가 결정하는 단계(단계 612); 및 4) 전송 중지 트리거가 검출될 때까지 클라이언트가 단계(1), (2), 및 (3)을 반복하는 단계(단계 614)를 포함한다.
- [0081] 일부 실시예에서, 미디어 소스는 카메라이고, 실시예에서 카메라는 클라이언트의 필수 부분이거나, 클라이언트로부터 떨어져있다 (예를 들면, 블루투스 연결과 같은 링크를 통해 클라이언트에 연결된 드론 상의 카메라). 일부 실시예에서, 트리거는 지속 시간을 지정하는 스케줄을 기반으로 하고, 클라이언트는 지속 시간이 경과했음을 검출한 결과로 트리거를 검출한다.
- [0082] 일부 실시예에서, 프로세서는 클라이언트가 미디어 소스에 의해 발생된 제1 세트의 미디어 프레임을 인코딩하여 제1 세트의 인코딩 미디어 프레임을 생성하는데 제1 품질 설정을 사용하는 단계를 포함한다. 전송 버퍼에 저장된 미디어 데이터는 제1 세트의 인코딩 미디어 프레임을 포함한다. HTTP 요청 메시지의 바디를 전송하는 단계는 클라이언트가 제1 코덱 구성 정보를 전송하는 단계를 더 포함한다. 제1 코덱 구성 정보는 제1 품질 설정을 나타내는 정보를 포함한다. 프로세서는 클라이언트가 전송 버퍼에 저장된 데이터의 양을 모니터링하는 단계; 및 클라이언트가 전송 버퍼에 저장된 데이터의 양이 한계치를 초과함을 결정하는 단계를 더 포함한다. 프로세서는 전송 버퍼에 저장된 데이터의 양이 한계치를 초과함을 결정한 결과로, 클라이언트가 미디어 소스에 의해 발생된 제2 세트의 미디어 프레임을 인코딩하여 제2 세트의 인코딩 미디어 프레임을 생성하는데 제2 품질 설정을 사용하는 단계를 더 포함한다. 프로세서는 클라이언트가 라이브 미디어 피드에 대응하는 추가 미디어 데이터를 전송 버퍼에 저장하는 단계를 더 포함한다. 추가 미디어 데이터는 제2 세트의 인코딩 미디어 프레임을 포함한다. HTTP 요청 메시지의 바디를 전송하는 단계는 클라이언트가 제2 코덱 구성 정보를 전송하는 단계를 더 포함한다. 제2 코덱 구성 정보는 제2 품질 설정을 나타내는 정보를 포함한다.
- [0083] 일부 실시예에서, 프로세서는 클라이언트가 미디어 소스에 의해 발생된 제1 세트의 미디어 프레임을 인코딩하여 제1 세트의 인코딩 미디어 프레임을 생성하는 단계를 더 포함한다. 전송 버퍼에 저장된 미디어 데이터는 제1 세트의 인코딩 미디어 프레임을 포함한다. 프로세서는 클라이언트가 전송 버퍼에 저장된 데이터의 양을 모니터링하는 단계; 및 클라이언트가 전송 버퍼에 저장된 데이터의 양이 한계치를 초과함을 결정하는 단계를 더 포함한다. 프로세서는 전송 버퍼에 저장된 데이터의 양이 한계치를 초과함을 결정한 결과로, 클라이언트가 제1 세트의 미디어 프레임의 적어도 한 서브세트를 전송 버퍼로부터 드로핑하여 드로핑된 프레임이 서버로 전송되지 않게 하는 단계를 더 포함한다.
- [0084] 일부 실시예에서, 전송 버퍼는 미디어 프레임의 상기 서브세트를 포함하고 미디어 프레임의 서브세트에 대한 메

타 데이터를 더 포함하는 미디어 프래그먼트를 저장하고, 미디어 프레임의 상기 서브셋을 전송 버퍼로부터 드로핑하는 단계는 전송 버퍼로부터 미디어 프래그먼트를 드로핑하여 미디어 프래그먼트가 서버에 전송되지 않게 하는 단계를 포함한다. 일부 실시예에서, 프로세스는 클라이언트가 미디어 소스에 의해 발생된 비압축 비디오 프레임을 획득하는 단계; 클라이언트가 비압축 비디오 프레임을 인코딩하여 인코딩 비디오 프레임을 생성하는 단계; 클라이언트가: i) 인코딩된 비디오 프레임 및 ii) 인코딩된 비디오 프레임에 관련된 메타-데이터를 포함하는 비디오 프래그먼트를 발생시키는 단계; 및 클라이언트가 프래그먼트를 전송 버퍼에 저장하는 단계를 더 포함한다.

[0085] 일부 실시예에서, 비디오 프래그먼트는: i) ISO-BMFF 비디오 프래그먼트 및 ii) 공통 미디어 애플리케이션 포맷(CMAF) 비디오 프래그먼트 중 하나이다. 일부 실시예에서, 라이브 피드는 라이브 배급을 위한 수신 엔터티(서버)에 의해 더 처리될 수 있다.

[0086] 일부 실시예에 따라, HTTP 클라이언트는 (예를 들어, 이동 디바이스 또는 사용자 장비(UE)에서) 미디어 소스에 의해 생성된 라이브 미디어 (오디오 및/또는 비디오) 피드를 서버에 업스트리밍하는 프로세스(600)를 실행하도록 적응된다. 일부 실시예에 따라, HTTP 클라이언트는 수신기; 전송기; 데이터 저장 시스템; 및 프로세서를 포함하는 데이터 프로세싱 장치를 포함하고, 여기서 데이터 프로세싱 장치는 데이터 저장 시스템, 전송기, 및 수신기에 연결되고, 데이터 프로세싱 장치는 프로세스(600)를 실행하도록 구성된다.

[0087] 도 7은 프로세스(700)를 설명한다. 프로세스(700)는 미디어 소스에 의해 생성된 라이브 미디어 (오디오 및/또는 비디오) 피드를 클라이언트로부터 수신하기 위해 HTTP 버에 (예를 들면, HTTP 취합 서버(106)) 의해 실행될 수 있다. HTTP 서버(106)는 클라이언트와의 운송 계층 연결을 (예를 들면, TCP 연결) 통해, 바디를 포함하는 HTTP 요청 메시지에 대한 하이퍼텍스트 전달 프로토콜(HTTP) 헤더를 수신한다(단계 702). HTTP 헤더는 HTTP 요청 메시지의 바디의 크기를 나타내지 않는다. HTTP 서버(106)는, HTTP 헤더를 수신한 이후에, HTTP 요청 메시지의 바디를 수신한다(단계 704). HTTP 요청 메시지의 바디를 수신하는 단계는: HTTP 클라이언트로부터 제1 데이터 청크를 수신하는 단계(단계 706); HTTP 클라이언트로부터 제2 데이터 청크를 수신하는 단계(단계 708); 및 제1 데이터 청크를 수신한 이후이고 제2 데이터 청크를 수신하기 이전에, 스트리밍 비디오를 배급하기 위한 배급 시스템에 제1 데이터 청크를 전달하는 단계(단계 710)를 포함한다.

[0088] 일부 실시예에 따라, HTTP 서버는 미디어 소스에 의해 생성된 라이브 미디어 (오디오 및/또는 비디오) 피드를 클라이언트로부터 수신하는 프로세스(700)를 실행하도록 적응된다. 일부 실시예에 따라, HTTP 서버는 수신기; 전송기; 데이터 저장 시스템; 및 프로세서를 포함하는 데이터 프로세싱 장치를 포함하고, 여기서 데이터 프로세싱 장치는 데이터 저장 시스템, 전송기, 및 수신기에 연결되고, 데이터 프로세싱 장치는 프로세스(700)를 실행하도록 구성된다.

[0089] 도 8은 일부 실시예에 따라 HTTP 클라이언트(800)의 기능적 모듈을 도시하는 도면이다. 도 8에 도시된 바와 같이, HTTP 클라이언트(800)는 운송 모듈(802), 전송 모듈(804), 및 데이터 저장 모듈(806)을 포함한다. HTTP 클라이언트(800)는 미디어 소스에 의해 생성된 라이브 미디어 (오디오 및/또는 비디오) 피드를 서버에 업스트리밍하는데 사용될 수 있다. 전송 모듈(802)은 서버와의 운송 계층 연결을 (예를 들면, TCP 연결) 설정하도록 구성된다. 전송 모듈(804)은 서버에 운송 계층 연결을 통해 헤드 및 바디를 포함하는 하이퍼텍스트 전달 프로토콜(HTTP) 요청 메시지를 전송하도록 구성된다. HTTP 요청 메시지의 헤더는 HTTP 요청 메시지의 바디의 크기를 나타내지 않는다. 데이터 저장 모듈(806)은 미디어 데이터가 발생될 때 라이브 미디어 피드에 대응하는 미디어 데이터를 전송 버퍼에 저장하도록 구성된다. 품질 설정은 미디어 데이터를 발생하는데 사용된다. HTTP 요청 메시지의 바디를 서버에 운송 계층 연결을 통해 전송하는 단계는: 1) 클라이언트가 전송 버퍼에 현재 저장된 미디어 데이터의 적어도 일부를 운송 계층 연결을 통해 서버에 전송하는 단계; 2) 클라이언트가 미디어 데이터의 상기 적어도 일부를 전송 버퍼로부터 제거하는 단계; 3) 전송 버퍼 레벨 및/또는 시간에 걸친 전송 버퍼 레벨에서의 변화를 (예를 들면, 버퍼 레벨 경사도) 기반으로 라이브 미디어 피드에 대응하는 미디어 데이터를 발생하는데 사용되는 품질 설정을 클라이언트가 수정하여야 하는가 여부를 클라이언트가 결정하는 단계; 및 4) 전송 중지 트리거가 검출될 때까지 클라이언트가 단계(1), (2), 및 (3)을 반복하는 단계를 포함한다.

[0090] 도 9는 일부 실시예에 따라 HTTP 서버(900)의 기능적 모듈을 도시하는 도면이다. 도 9에 도시된 바와 같이, HTTP 서버(900)는 제1 수신 모듈(902), 제2 수신 모듈(904), 및 전달 모듈(906)을 포함한다. HTTP 서버(900)는 미디어 소스에 의해 생성된 라이브 미디어 (오디오 및/또는 비디오) 피드를 서버에 수신하는데 사용될 수 있다. 제1 수신 모듈(902)은 클라이언트와의 운송 계층 연결을 (예를 들면, TCP 연결) 통해 바디를 포함하는 HTTP 요청 메시지에 대한 하이퍼텍스트 전달 프로토콜(HTTP) 헤더를 수신하도록 구성된다. HTTP 헤더는 HTTP

요청 메시지의 바디의 크기를 나타내지 않는다. 제2 수신 모듈(904)은 제1 수신 모듈에 의해 HTTP 헤더를 수신한 이후에, HTTP 요청 메시지의 바디를 수신하도록 구성된다. HTTP 요청 메시지의 바디를 수신하는 단계는: HTTP 클라이언트로부터 제1 데이터 청크를 수신하는 단계; HTTP 클라이언트로부터 제2 데이터 청크를 수신하는 단계; 및 제1 데이터 청크를 수신한 이후이고 제2 데이터 청크를 수신하기 이전에, 스트리밍 비디오를 배급하기 위한 배급 시스템에 제1 데이터 청크를 전달 모듈(906)에 의해 전달하는 단계를 포함한다.

[0091] 도 10은 일부 실시예에 따른 HTTP 클라이언트(102)의 블록도이다. 도 10에 도시된 바와 같이, HTTP 클라이언트(102)는: 하나 이상의 프로세서(P)(1055)를 (예를 들면, 애플리케이션 지정 집적 회로(ASIC), 필드-프로그램가능 게이트 어레이(FPGA) 등과 같이, 범용 마이크로프로세서 및/또는 하나 이상의 다른 프로세서) 포함할 수 있는 데이터 프로세싱 장치(DPA)(1002); HTTP 클라이언트(102)가 AN 노드에 (예를 들면, 기지국) 데이터를 전송하고 그로부터 수신할 수 있게 하기 위해 안테나(1022)에 연결된 전송기(1005)와 수신기(1004); 및 하나 이상의 비휘발성 저장 디바이스 및/또는 하나 이상의 휘발성 저장 디바이스를 (예를 들면, 랜덤 액세스 메모리(RAM)) 포함할 수 있는 로컬 저장 유닛(1008)을 ("데이터 저장 시스템"이라 칭하여지는) 포함할 수 있다. HTTP 클라이언트(102)가 범용 마이크로프로세서를 포함하는 실시예에서는 컴퓨터 프로그램 제품(computer program product, CPP)(1041)이 제공될 수 있다. CPP(1041)는 컴퓨터 판독가능 명령(computer readable instructions, CRI)(1044)을 포함하는 컴퓨터 프로그램(CP)(1043)을 저장한 컴퓨터 판독가능 매체(computer readable medium, CRM)(1042)를 포함한다. CRM(1042)은 예를 들면, 제한되지는 않지만, 자기 매체 (예를 들어, 하드 디스크), 광학 매체, 메모리 디바이스 (예를 들어, 랜덤 액세스 메모리) 등과 같은, 비일시적 컴퓨터 판독가능 매체가 될 수 있다. 일부 실시예에서, 컴퓨터 프로그램(1043)의 CRI(1044)는 데이터 프로세싱 장치(1002)에 의해 실행될 때, CRI로 HTTP 클라이언트(102)가 상기에 설명된 단계들을 (예를 들면, 흐름도를 참조로 상기에 설명된 단계들) 실행하게 하도록 구성된다. 다른 실시예에서, HTTP 클라이언트(102)는 코드 필요없이 여기서 설명된 단계들을 실행하도록 구성될 수 있다. 즉, 예를 들어, 데이터 프로세싱 장치(1002)가 하나 이상의 ASIC만으로 구성될 수 있다. 따라서, 여기서 설명된 실시예의 특징은 하드웨어 및/또는 소프트웨어로 구현될 수 있다.

[0092] 도 11은 일부 실시예에 따른 HTTP 서버(106)의 블록도이다. 도 11에 도시된 바와 같이, HTTP 서버(106)는: 하나 이상의 프로세서(P)(1155)를 (예를 들면, 애플리케이션 지정 집적 회로(ASIC), 필드-프로그램가능 게이트 어레이(FPGA) 등과 같이, 범용 마이크로프로세서 및/또는 하나 이상의 다른 프로세서) 포함할 수 있는 데이터 프로세싱 장치(DPA)(1102); 네트워크 인터페이스(1148)가 연결되는 네트워크(110)에 (예를 들면, 인터넷 프로토콜(Internet Protocol, IP) 네트워크) 연결된 다른 노드에 HTTP 서버(106)가 데이터를 전송하고 그로부터 데이터를 수신할 수 있게 하기 위해 전송기(Tx)(1145) 및 수신기(Rx)(1147)를 포함하는 네트워크 인터페이스(1148); UE와의 무선 통신을 위해 안테나 시스템(1104)에 연결된 회로(1103) (예를 들면, 무선 송수신 회로); 및 하나 이상의 비휘발성 저장 디바이스 및/또는 하나 이상의 휘발성 저장 디바이스를 (예를 들면, 랜덤 액세스 메모리(RAM)) 포함할 수 있는 로컬 저장 유닛(1108)을 ("데이터 저장 시스템"이라 칭하여지는) 포함할 수 있다. HTTP 서버(106)가 범용 마이크로프로세서를 포함하는 실시예에서는 컴퓨터 프로그램 제품(CPP)(1141)이 제공될 수 있다. CPP(1141)는 컴퓨터 판독가능 명령(CRI)(1144)을 포함하는 컴퓨터 프로그램(CP)(1143)을 저장한 컴퓨터 판독가능 매체(CRM)(1142)를 포함한다. CRM(1142)은 예를 들면, 제한되지는 않지만, 자기 매체 (예를 들어, 하드 디스크), 광학 매체, 메모리 디바이스 (예를 들어, 랜덤 액세스 메모리) 등과 같은, 비일시적 컴퓨터 판독가능 매체가 될 수 있다. 일부 실시예에서, 컴퓨터 프로그램(1143)의 CRI(1144)는 데이터 프로세싱 장치(1102)에 의해 실행될 때, CRI로 HTTP 서버(106)가 상기에 설명된 단계들을 (예를 들면, 흐름도를 참조로 상기에 설명된 단계들) 실행하게 하도록 구성된다. 다른 실시예에서, HTTP 서버(106)는 코드 필요없이 여기서 설명된 단계들을 실행하도록 구성될 수 있다. 즉, 예를 들어, 데이터 프로세싱 장치(1102)가 하나 이상의 ASIC만으로 구성될 수 있다. 따라서, 여기서 설명된 실시예의 특징은 하드웨어 및/또는 소프트웨어로 구현될 수 있다.

[0093] 사용 사례

[0094] 스마트폰 및 드론과 같은 다른 디바이스의 진화는 업링크 방향을 강조하고 있고, 여기서 이러한 디바이스에는 방송 품질의 비디오 스트림을 생성할 수 있는 첨단 카메라가 장착된다. 여기서 설명되는 실시예는 이러한 경향을 이용하도록 유일하게 적용된다. 이후에는 예시적인 사용 사례가 설명된다.

[0095] 한가지 사용 사례는 예를 들어, 항공 뷰로부터, 라이브 이벤트를 캡처하기 위해 드론을 사용하는 것이고, 여기서 드론은 이벤트 위로 비행하여 네트워크를 (예를 들면, 브로드밴드 네트워크) 통해 브로드캐스터에 (또는 다른 제3 집단) 직접 콘텐츠를 송신할 수 있다. 이러한 사용 사례에서, 드론에는 고품질 카메라(302)만이 장착될 수 있고, 특정한 타겟을 따라가도록 가시선(line-of-sight) 또는 팔로미(follow-me) 모드에서 수동으로 동작된다. 드론은 모뎀을 (예를 들면, 5G 모뎀) 갖고 HTTP 클라이언트(310)를 운영하는 임의의 운영 체계를 실행할

수 있다. 즉, 드론은 클라이언트(102)의 한 예가 될 수 있다. 또는, 다른 실시예에서, 드론의 카메라(302)는 링크를 통해 (블루투스 링크와 같은) 모듈(304 내지 310)을 포함하는 디바이스에 연결될 수 있고, 이 디바이스는 드론에 근접해 위치한다. 상기에 설명된 바와 같이, 이러한 디바이스는 즉각적인 프로세싱을 위해 원격 서버(106)에 라이브 비디오를 스트리밍 할 수 있다. 이 사용 사례는 스포츠 이벤트에서 스키 및 랠리 경주와 같은 이벤트를 캡처하는데 사용될 수 있고, 올림픽과 같은 스포츠 이벤트에서 사용될 수 있다 (예를 들면, 2018년 도 올림픽에서, 5G 네트워크가 다양한 올림픽 이벤트를 방송하고 캡처하는데 도움이 될 수 있다).

[0096] 또 다른 사용 사례는 카메라(302)를 갖춘 스마트폰에서 뉴스를 방송하는 것이다. 즉, 스마트폰이 클라이언트(102)의 한 예가 될 수 있다. 스마트폰 기술은 개선되고 있고 첨단 스마트폰의 비디오 품질은 이미 전문 TV 프로덕션에서 (예를 들면, 뉴스 속보) 사용되기에 충분하다. 실시예는 스마트폰이 카메라를 호스팅할 수 있는 브로드밴드 액세스로 임의의 위치에서 리포터들이 즉각적으로 뉴스를 커버할 수 있게 한다.

[0097] 또 다른 사용 사례에서, 카메라(302)는 헬멧 카메라 (예를 들어, 운동선수가 착용한 카메라) 또는 바디 카메라 (예를 들어, 경찰관이 착용한 신체 카메라)이다. 카메라는 모듈(304 내지 310)을 포함하는 디바이스에 라이브 영상을 스트리밍할 수 있고, 이 디바이스는 상기에 설명된 바와 같이 즉각적인 프로세싱을 위해 원격 서버(106)에 라이브 비디오를 스트리밍할 수 있다. 카메라는 예를 들어, 캡처 대기시간을 줄이기 위해 또는 비디오 품질을 증가시키기 위해 또는 레코딩을 시작/중단하기 위해, 원격으로 동작될 수 있다. 실시예는 이러한 카메라 피드의 라이브 업링크 스트리밍을 지원한다.

[0098] 실시예는 또한 사용자 생성 콘텐츠(user generated content, UGC)의 트렌드를 지원하기 위해 많이 사용될 수 있고, 여기서 이동 전화나 바디 카메라를 가진 사용자가 소셜 미디어 플랫폼을 사용하여 웹 사이트에 라이브 스트림을 취합하도록 허용한다.

[0099] ISO 베이스 미디어 파일 포맷(Base Media File Format, BMFF)을 사용하는 프래그먼트화 MP4 파일의 사용

[0100] 여기서 설명되는 실시예에는 ISO 베이스 미디어 파일 포맷(BMFF)을 사용하는 프래그먼트화 MP4 파일(fragmented MP4 file, fMP4)이 적용가능하다. ISO BMFF는 ISO/EC 14996-12에서 정의된다. 이는 박스 구조 위에 설립된다. 각 박스는 박스 크기 정보(4 바이트)로 시작되어 박스 타입 정보(또한, 4 바이트)로 이어지고, 일반적으로 4-문자로 표현된다 (예를 들면, 파일 타입에 대해 ftyp). 박스는 복합 박스가 될 수 있고, 이는 그 박스가 다른 차일드(child) 박스를 포함함을 의미한다. 프래그먼트화 MP4 파일은 (또는 ISO-BMFF 파일) 일반적인 MP4 파일과 다른 구조이다. 파일은 파일 타입(ftyp)으로 시작되어, 정확하게 하나의 무비 박스 및 다른 박스로 이어진다. 미디어 데이터는 무비 프래그먼트 박스(moov) 및 연관된 미디어 데이터 박스(mdaf)의 조합인 프래그먼트를 통해 설명된다.

[0101] ftyp 박스는 파일의 호환성 정보 및 사용되는 코덱을 포함한다.

[0102] 무비 박스(moov)는 코덱 구성 정보를 포함한다. DRM 보호 콘텐츠에 대해서는 추가 DRM 정보가 포함된다. 일반적인 MP4 파일과 다르게, 모든 시간-대-샘플 테이블은 비어 있다. 시간-대-샘플 테이블은 샘플 지속시간, 샘플 조성, 및 디코딩 시간을 설명하고, 미디어 데이터 박스에서 샘플의 길이 및 정확한 바이트-오프셋을 설명한다.

[0103] 프래그먼트화 MP4 파일은 많은 미디어 데이터 박스(mdaf)를 포함한다. mdaf 박스는 각 샘플의 실제 미디어 데이터를 포함하고, 이는 박스로 연속하여 연결된다. 프래그먼트화 MP4 파일에서의 각 미디어 데이터 박스에 대해, 각 개별 미디어 데이터 박스의 타이밍 및 바이트 오프셋 정보를 포함하는 전용 무비 프래그먼트 박스가 있다.

[0104] 프래그먼트화 MP4 파일의 예는 이제 설명될 소정의 박스 계층구조를 포함할 수 있다. 파일은 최상위 레벨, 단일 ftyp, 및 단일 moov 박스를 포함한다. 이때, 무비 프래그먼트 박스(moov) 및 미디어 데이터 박스(mdaf)가 많이 있다. mdaf 박스 만큼의 moov 박스가 있다. 다른 박스도 (예를 들면, DRM 설명에 대해) 또한 포함될 수 있다.

[0105] 연관된 미디어 데이터 박스와의 무비 프래그먼트의 조합은 한 프래그먼트로 고려될 수 있다. 일부 실시예에서, 이 프래그먼트는 HTTP 클라이언트 및 서버가 작업하는 세분화 객체가 될 수 있다. ISO/IEC 23000-19 (공통 미디어 애플리케이션 포맷, 또는 CMAF)에서, 무비 프래그먼트 박스와 미디어 데이터 박스의 조합은 CMAF 청트라 칭하여진다.

[0106] 프래그먼트화 MP4 파일의 moov 박스는 또한 코덱 구성을 포함하므로 초기화 세그먼트 또는 초기화 부분이라 칭하여진다. 설명되는 예에서, 프래그먼트화 MP4 파일은 H.264 코덱을 사용하는 단일 비디오 트랙(trak 박스)을

포함할 수 있다 (avc1 박스 및 연관된 속성값 엔트리로 표현되는).

- [0107] 추가 박스는 프래그먼트에 (moof 및 mdat 박스의 조합) 포함될 수 있다. moof 박스는 복합 박스로, 여러개의 다른 차일드 박스를 포함한다. 여기서 가장 중요한 차일드 박스는 트랙 프래그먼트 헤더 박스(tfhd) 및 트랙 프래그먼트 런 박스(trun)이다. 프래그먼트가 다수의 트랙으로부터의 미디어를 포함할 때 (오디오 및 비디오와 같이), 다수의 트랙 프래그먼트(traf) 박스가 프래그먼트에 주어진다.
- [0108] 트랙 프래그먼트 런 박스(trun)의 예는 프래그먼트의 mdat 섹션이 단일 샘플만을 (또는 비디오 프레임) 포함함을 나타낼 수 있다. data\_offset 필드는 (trun 박스의) 샘플의 시작 바이트 오프셋을 나타내는 mdat 섹션에 대한 포인터이다.
- [0109] 프래그먼트에 단일 샘플만이 있으므로 (현재 설명되는 예에서), 트랙 프래그먼트 런 박스에는 샘플 길이 정보가 없다. 샘플 크기는 본 예에서 default\_sample\_size로 트랙 프래그먼트 헤더 박스(tfhd)에 포함된다.
- [0110] 트랙 프래그먼트 헤더 박스는 default\_sample\_duration (예를 들면, 100), the default\_sample\_size (예를 들면, 4318), 및 default\_sample\_flag (예를 들면, 0x100c0)을 포함할 수 있다. default\_sample\_flag는 여기서 이것이 싱크(sync) 샘플임, 즉 샘플이 디코딩을 시작하는데 사용될 수 있음을 설명한다.
- [0111] default\_sample\_duration은 (tfhd 박스의) 샘플 지속시간을 (시간스케일 눈금에서) 나타낸다. 초기화 세그먼트로부터의 트랙 시간스케일로 (미디어 헤더 박스 또는 mdhd에서), 샘플 지속시간을 초로 (예를 들면, 100/3000 sec) 계산할 수 있다. 비디오는 초당 30 프레임의 프레임비율로 녹화된다.
- [0112] 프래그먼트에 다수의 샘플이 있을 때, 각 샘플이 자체 크기를 가지므로, 적어도 각 샘플의 세그먼트 크기가 trun 박스에 포함된다.
- [0113] trun 박스의 또 다른 예가 이제 설명된다. 프래그먼트가 30 프레임을 포함할 수 있고 플래그가 프레임의 코딩 의존도를 나타낼 수 있다. 이 샘플은 세그먼트화된 미디어 파일로부터 가져온 것임을 주목하여야 하고, 이는 프래그먼트 정보가 styp(세그먼트 타입) 및 선택적으로 sidx 박스와 같은 추가 정보와 분리된 파일로 결합될 수 있음을 의미한다.
- [0114] 본 출원이 우선권을 주장하는 미국 조건 출원은 표준을 수정하는 제안을 포함한 부록을 포함한다. 이 제안 중 관련된 부분이 ("기여"라 칭하여지는) 이후 재현된다.
- [0115] 2 일반적인 사용 사례 설명 및 일반적인 고려 사항
- [0116] 2.1 일반적인 사용 사례 및 설계 고려 사항
- [0117] 작업 항목은 라이브 업링크 비디오 스트리밍에 대한 프레임워크를 표준화하는 것을 목표로 한다. 한가지 고려 사항은 다양한 3GPP 액세스 시스템의 수단과 기능, 특히 새로운 NR 액세스 기술의 새로운 기능과 특징을 사용하는 것이다. 다양한 3GPP 액세스 시스템은 (예를 들면, LTE 또는 NR) 다른 기능을 갖고, 이는 디바이스 실현 (또한 디바이스 카테고리) 및 시스템 배포에 의존한다.
- [0118] 라이브 업링크 비디오는 일반적으로 라이브 비디오 배급 서비스에 대한 기여 또는 취함으로 사용된다. '기여 (contribution)'란 용어는 때로 미디어의 라이브 취함을 위한 전문 TV 프로덕션에서 사용된다. 이러한 비디오 시스템의 주요 특성은 비디오 흐름이 일부 중간 노드를 통해 비디오 소스로부터 일반적으로 많은 수신기에 이르는 단일 방향성이라는 점이다. 취함 링크와 가능한 배급 시스템의 관계는 도 1에 도시된다. 취함 링크는 배급 시스템으로부터 분리될 수 있다. 또는, 취함 서버가 나중에 사용하기 위해 비디오 데이터를 저장하거나 취함 서버가 비디오 스트림을 모니터링 화면으로 전달할 수 있다. 예를 들면, 소스의 라이브 비디오는 나중에 사용 되도록 파일 시스템에 저장되거나 라이브 비디오 소스로부터의 비디오 내용이 먼저 화면화되고 프로그램 감독이 특정한 라이브 소스를 사용하기로 결정할 때만 사용될 수 있다.
- [0119] 배급 측에서는 적응적 비트비율 스트리밍이 (DASH와 같은) 오늘날 일반적으로 사용된다. 적응적 비트비율에 대해, 콘텐츠는 다수의 품질 표현으로 (비디오 프로파일이라 칭하여지는), 예를 들면 수백 kbps에서 수백 Mbps까지의 범위에서 제공될 필요가 있다. 용도에 따라, DRM 보호가 배급을 위해 추가된다. IP 기반의 시스템을 위한 배급 시스템은 일반적으로 콘텐츠 전달 네트워크(Content Delivery Network, CDN)이다. 다른 배급 시스템도 (IP 및 전통적인 방송 시스템) 특정하게 가능하다.
- [0120] 라이브 취함 측에서 (작업 항목의 범위인), 라이브 미디어 소스는 프레임을 캡처하고, 이어서 프레임을 취함 서버에 (즉, 미디어 싱크) 전송한다. 본 내용에서는 비디오 콘텐츠가 일반적으로 오디오 또는 다른 미디어 보다

훨씬 높은 데이터 비율을 요구하므로, 비디오 콘텐츠에 초점을 맞춘다. 그러나, 오디오 또는 다른 미디어도 또한 본 내용에 적용가능하다.

- [0121] 캡처된 비디오 프레임은 취합 링크를 통해 전송되기 이전에 비트비율을 감소시키도록 압축될 수 있다. 일부 실시예에서의 기여 링크는 3GPP 액세스 시스템으로, 업링크 방향에서 사용된다.
- [0122] 디바이스 측에서의 (예를 들면, 개인 방화벽) 또한 3GPP 시스템을 통한 취합 링크 경로에 NAT 및 방화벽이 존재하기 때문에 (비록 이 기능은 표준화되지 않았지만 여전히 존재하는), 통신의 미디어 소스는 (예를 들면, 카메라) 취합 서버 방향으로 (예를 들면, 미디어 싱크) 통신 세션을 활성화시켜야 한다. 그 이유는 방화벽의 "오픈 포트"가 때로 보안상의 이유로 또는 (회사) 보안 규칙으로 인해 금지되기 때문이다. 또한, 경로에 네트워크 어드레스 변환기(Network Address Translator, NAT)가 있을 때, 타겟 디바이스에 전용 NAT 전달 규칙이 있어야 한다.
- [0123] 클라이언트 디바이스는 때로 동적 IP 어드레스로 구성된다. 방화벽 및 NAT 전달 규칙은 일반적으로 디바이스 IP 어드레스에 의존하므로, 매 IP 어드레스가 변할 때 조정될 필요가 있다.
- [0124] 클라이언트 디바이스와 달리, 취합 서버는 들어오는 연결을 수신하도록 적절하게 구성되고 (예를 들면, 도달할 수 있도록) 보호된다. 취합 서버가 NAT 뒤에 있을 때, 대응하는 NAT 전달 규칙이 구성된다.
- [0125] 그러므로, 3GPP UE 측에서 서버를 운영하는 것이 기술적으로 가능하지만, 이 제안은 UE 측에서 분리된 서버에 초점을 둔다.
- [0126] 2.2 비디오 압축 고려 사항
- [0127] 전문 방송 작업에서는 직렬 디지털 인터페이스(Serial Digital Interface, SDI)가 일반적으로 외부 이벤트로부터 방송 스튜디오에 라이브 비디오를 전송하기 위해 또한 라이브 비디오 처리를 위해 사용된다. SMPTE는 RTP를 통해 SDI 신호를 전달하기 위해 2022 RTP 페이로드 포맷에서 (IETF RFC 4175) 정의된다 (RTP 세션은 다양한 수단을 통해 설정될 수 있음을 주목하여야 한다). SDI가 압축되지 않은 비디오 프레임 또는 약간 압축된 비디오 프레임을 전달하므로 (JPEG2000을 통해), 고해상도 비디오에 대한 일반적인 SDI 비트비율은 기가비트 범위이다. SDI의 주요 이점은 매 프레임마다 스위치되는 능력과 (GoP 타입의 의존성 구조가 없다) 낮은 대기시간이다 (매 프레임이 즉각적으로 송신된다).
- [0128] 3GPP는 비디오 비트비율을 줄이기 위해 짧은 대기시간의 VoLTE의 대화식 비디오에도 (Video(ViLTE)) 압축된 비디오를 (H.264 또는 HEVC를 사용하여) 사용한다. 짧은 대기시간의 비디오 압축에 대해서는 결과의 비디오 비율보다 대기시간이 우선화되고, B-프레임과 같은 여러 압축 톨이 사용되지 않는다. 임의의 상세한 규범적 작업을 시작하기 이전에, 3GPP 라이브 업링크 비디오 해결법에 대한 타겟 비트비율/비트비율 범위 및 대기시간 제한을 논의하고 동의할 것을 추천한다.
- [0129] 2.3 적응적 비트비율 및 QoS
- [0130] 3GPP 시스템은 LTE 또는 NR과 같은 다른 세트의 액세스 시스템을 지원한다. 액세스 시스템은 지원되는 링크 비트비율 또는 QoS와 같은 다른 기능을 갖는다. 그 기능은 배포 실현에 따른 특정 레벨에, 예를 들면 할당된 캐리어 대역폭에 의존한다 (예를 들면, LTE 캐리어는 20MHz, 10MHz와 같이 다른 대역폭을 지원할 수 있다). 액세스 시스템은 캐리어 집합을 지원할 수 있고, 이는 다수의 무선 캐리어의 업링크 용량을 단일 업링크로 결합하는 것을 의미한다.
- [0131] 3GPP UE는 디바이스 실현 및 디바이스 공급자에 따라 다른 기능을 가질 수 있다.
- [0132] 시스템 특성은 디바이스 이동 패턴에 의존할 수 있다. 예를 들면, 채널 특성은 디바이스가 정지되어 있는가 이동 중인가 여부에 의존할 수 있다. 디바이스가 이동 중일 때, 디바이스는 한 기지국에서 다른 기지국으로 또는 한 액세스 네트워크에서 또 다른 액세스 네트워크로 핸드오버(handover)를 실행할 수 있다. 이는 일시적으로 스트림을 중단시킬 수 있고, 예를 들어 취합 서버 이전에 추가 버퍼링을 (또한 대기시간) 도입함으로써 완화될 수 있다.
- [0133] 링크 취합의 특정한 최소 비트비율의 이용가능성을 증가시키기 위해, 서비스 품질(Quality of Service, QoS) 및 다른 기술이 사용될 수 있다.
- [0134] 타겟 해결법은 네트워크의 서비스 품질(QoS) 메커니즘을 활용할 수 있도록 추천된다. 그러나, 3GPP 액세스 시스템, 이동성, 및 배포의 차이로 인하여, 업링크 스트리밍 해결법은 다른 비트비율로 작동할 수 있고 적응적 비

트비율 변경까지 지원하도록 추천된다.

- [0135] 2.4 보안 고려 사항
- [0136] 라이브 취합 해결법은 오용되는 것에 대해 적절하게 보호될 필요가 있다. 적어도 다음과 같은 오용 고려사항이 있다.
- [0137] 라이브 취합 해결법은 다수의 독립적인 미디어 소스와 함께 순차적으로 또는 동시에 사용하는 것이 가능하게 된다. 해결법은 각 미디어 소스에 대해 분리된 계정의 사용을 고려하거나 지원하여야 한다. 여기서, 계정은 취합 및 후처리 기능의 체인이고 (ABR 트랜스코더와 같이), 또한 배급 시스템일 수도 있다 (즉, 라이브 피드가 수신기에 제공되는 방법). 인증된 클라이언트만이 이러한 계정을 액세스 및 제어할 수 있고, 각 계정이 유일한 취합 포인트 설명을 갖도록 보장하여야 한다.
- [0138] 또한, 취합 서버가 들어오는 데이터에 대해 특정한 취합 포인트에서 (예를 들면, 포트 또는 URL) 청취하고 있을 때, 인증된 소스만이 미디어 데이터를 후처리 및 배급 체인으로 취합할 수 있음을 보장하여야 한다. 그렇지 않으면, 대체 비디오를 삽입하거나 쓸모없는 데이터만을 삽입하여, 라이브 서비스를 가로 채거나 스팸 처리할 수 있다. 그래서, 사용자 평면은 취합 서버가 인증된 콘텐츠 데이터와 인증되지 않은 데이터 사이를 구별할 수 있도록 보호되어야 한다.
- [0139] 2.5 기능 교환
- [0140] 도 1에 도시된 바와 같이, 취합 서버는 (미디어 싱크) ABR 트랜스코더와 같은 이어지는 기능에 데이터를 전달한다. 임의의 트랜스코딩 또는 임의의 비디오 렌더링을 위해, 시스템은 들어오는 스트림을 먼저 디코딩할 필요가 있다. 넓은범위의 다른 코덱 및 코덱 프로파일이 이용가능하다.
- [0141] 취합 서버가 수신된 스트림을 후처리할 수 있음을 보장하기 위해, 기능 교환 및 잠재적으로 코덱 협상이 필요하다. 가장 간단한 형태는 취합 서버가 기능을 (자막 지원 또는 지원되는 코덱과 같은) 발표하거나 노출하여, 클라이언트가 (미디어 소스) 라이브 취합에 적절한 서브세트를 선택할 수 있게 하는 것이다. 설정을 노출하거나 협상하는 다른 수단도 있다.
- [0142] 일반적인 취합 서버 기능은 지원되는 코덱 및 코덱 프로파일이다. 자막 스트림의 지원 또는 광고 기회 표시기 삽입과 같은 추가 기능이 있다.
- [0143] 기능 노출/협상 프레임워크는 확장될 수 있고 공급자 지정 기능을 허용하여야 한다.
- [0144] 3 실현 대안
- [0145] 실현하는데 고려될 수 있는 한 세트의 이용가능한 기술적 해결법이 있다. 다음에는 이용가능한 실현 대안 중 적어도 일부가 간략히 소개된다. 다른 실현 대안이 이용가능함을 주목하여야 한다.
- [0146] 3.1 RTP 기반의 구조
- [0147] 3.1.1 일반
- [0148] RTP는 비디오 전송을 위해 다양한 환경에서 일반적으로 사용되는 프로토콜이다. H.264 (RFC 6184) 또는 HEVC (RFC 7798) 비디오와 같이, 이용가능한 다양한 RPT 패이로드 포맷이 있다. 또한, RTP (RFC 2250) 내부에 MPEG2-운송 스트림 멀티플렉싱 데이터(MPEG2-Transport Stream multiplexed data)를 전달하거나 직렬 디지털 인터페이스(Serial Digital Interface)(SDI, SMPTE 2022-6) (RFC 4175)의 압축되지 않은 비디오도 전달하는데 이용가능한 포맷도 있다. SDI는 전문 TV 프로덕션 시스템에서 널리 사용됨을 주목한다.
- [0149] RTP는 UDP 운송을 사용하고, UDP 운송을 설정하고 구성하기 위해 전용 프로토콜이 필요하다. SRTP로, RTP에 대해 보안 프레임워크가 이용될 수 있다. 대안적으로 또는 보완으로, DTLS가 사용될 수 있다.
- [0150] 3.1.2 IMS 기본 기반의 라이브 취합 세션 설정
- [0151] 3GPP 대화 서비스는 IMS를 사용하여 구축되고, IMS는 업링크 라이브 비디오를 제공하기 위해 필요한 RTP 사용자 평면을 설정하는데 매우 적절하다 (3GPP TS 26.114). 라이브 업링크의 경우에서, 통신 채널은 단일 방향 비디오에 대해서만 사용되고, 이 사용 사례에 대한 IMS 시스템의 가능한 개선이 고려되어야 한다.
- [0152] 3GPP는 다양한 RTP 비디오 패이로드 포맷을, 특정하게 H.264 및 HEVC 비디오를 이미 지원하고 있다. 필요한 경우, 다른 패이로드 포맷도 추가될 수 있다.

- [0153] IMS는 유니캐스트 운송 세션을 설정하는데, 또한 코덱 협상 및 선택에 SIP/SDP를 사용한다. IMS는 허가 및 인증을 위한 프레임워크를 제공한다. SRTP 및/또는 DTLS는 사용자 평면 취합이 오용되지 않도록 보호하는데 사용될 수 있다.
- [0154] 세션 설정은 IMS와 함께 할 수 있다. 미디어 소스는 IMS 클라이언트이다. 미디어 싱크는 MRF이다 (여기서). 또 다른 예는 자동-응답 IMS 클라이언트가 미디어 싱크로 사용되는 것이 될 수 있다.
- [0155] 3GPP TS 26.114는 비디오 비율 제어를 정의한다. SCReAM와 같은 다른 RTP 비디오 비율 제어 구조가 존재한다. 한가지 대안은 IET에서 표준화 중인 SCReAM (Self-Clocked Rate Adaptation for Multimedia, 멀티미디어에 대한 자체 클럭 비율 적응화)이다. SCReAM은 네트워크 혼잡도 제어를 다루고, 또한 미디어 소스에 대해 추천되는 비트비율을 제공한다. SCReAM의 구현도 이용가능하다.
- [0156] 3.1.3 RTSP 기반의 라이브 취합 세션 설정
- [0157] 3GPP 패키지 스위칭 스트리밍(Packet Switched Streaming, PSS) 서비스는 다운링크 스트리밍 세션 설정에 RTSP를 사용한다. RTSP에서 라이브 업링크 비디오 스트리밍 해결법을 구축하는 것이 자연스럽게 보여지고, 여기서 RTSP 서버는 (미디어 싱크) 인프라구조에 위치한다.
- [0158] UE 측에 RTSP 서버를 배치하는 것은 기술적으로 가능하더라도 비현실적이다. 특히, 소비자 디바이스는 방화벽을 사용해 차폐된다. 일부 MNO는 네트워크 어드레스 변환을 사용하고 IP 어드레스를 동적으로 할당한다.
- [0159] RTSP 클라이언트는 미디어 소스로 동작하고 RTSP 서버는 미디어 싱크로 동작하여야 한다. RTSP 클라이언트는 RTSP 서버 쪽으로 RTP 업링크 스트리밍 세션을 설정하게 된다. 기존 RTSP 셋업 과정은 취합 링크에 대한 UDP 운송을 설정하는데 사용될 수 있다. 이때, 업링크 라이브 스트리밍 세션은 "RTSP 녹화" 방법을 사용해 시작되고 잠재적으로 RTSP 세트 매개변수를 사용하여 수정된다.
- [0160] RTSP 설명 방법은 선택된 코덱 및 코덱 구성을 알리는데 사용된다. RTSP 취합 서버에 의해 지원되는 코덱을 문의하는데는 별도의 과정이 사용되어야 한다.
- [0161] RTSP 클라이언트 및 UDP 사용자 평면 데이터를 인증하고 허락하기 위한 보안 과정이 더 연구되고 논의될 필요가 있다. SRTP 또는 DTLS는 사용자 평면 취합이 오용되는 것을 보호하는데 사용될 수 있다.
- [0162] RTP 스트림에 대해 다양한 비디오 비율 제어 구조가 존재하고, 예를 들어 저하된 커버리지로 인해 처리량이 낮아지는 경우에 대한 지연 요구사항을 만족시키기 위해 구현되어야 한다. SCReAM는, 섹션 3.1.2에서 소개된 바와 같이, 한가지 실현 대안이다.
- [0163] 3.1.4 WebRTC 기반의 라이브 취합 세션 설정
- [0164] WebRTC는 서비스와 같이 통신을 위한 브라우저에서 오늘날 널리 지원된다. WebRTC는 양방향 통신 서비스를 위해 설계되지만, 단일 방향 스트리밍 서비스에 대해서도 성공적으로 테스트되고 사용되었다. WebRTC 게이트웨이가 취합 서버로 사용될 수 있다.
- [0165] WebRTC는 통신 운송으로 SRTP/UDP를 사용한다. SRTP 및 DTLS의 조합을 사용한 보안이 구축된다.
- [0166] 3.2 TCP에서의 RTMP
- [0167] 업링크 스트리밍에 대해 가장 일반적인 스트리밍 프로토콜은 Adobe의 실시간 메시징 프로토콜(Real Time Messaging Protocol, RTMP)이다. RTMP는 잘 정의된 포트에서의 (즉, 포트 1935) 확실한 업링크 스트리밍을 위해 TCP를 사용한다. 인프라구조 측에서의 서버 구성성분과 함께 TCP 및 HTTP 기반의 업링크 스트리밍 포맷의 이점은 방화벽 및 NAT 문제를 방지하는 것이다. TCP의 사용에는 낮은 대기시간을 보장하는 TCP 구성이 필요하고, 이는 TCP 송신 버퍼의 적절한 설정 뿐만 아니라 낮은 대기시간을 보장하는 혼잡도 제어 알고리즘의 사용을 포함하고, 상세한 내용은 tbd에 있다.
- [0168] RTMP 스트림은 RTMP 프로토콜 핸들러 구조(rtmp://)에 의해 식별될 수 있으므로, rtmp://ex.com/live.swf 형태의 URL이 애플리케이션에 의해 해석될 수 있다. 별도의 공지된 포트는 (포트 1935) RTMP 구조에서 정의되므로, 포트가 요구되지 않는 것으로 제공된다. 물론, RTMP URL은 다른 포트도 허용한다.
- [0169] RTMP는 자체 메시지 포맷 및 멀티플렉싱 메커니즘을 정의한다. RTMP를 지원하기 위해, 송신자 및 수신자는 모두 필요한 범위의 RTMP 메시지 타입 및 메시지 처리 과정을 지원하여야 한다.

- [0170] RTMP는 메시지 기반의 프로토콜이다. 각 메시지는 길이, 때로 타임스탬프(timestamp), 및 일부 타입의 정보를 포함한다. 메시지는 멀티플렉싱되고 인터리브(interleave)되기 위해 더 작은 RTMP 청크로 세분될 수 있다. RTMP는 "청크 스트림(chunk stream)"을 정의하고, 이는 멀티플렉싱될 수 있다. RTMP 청크와 HTTP 청크 사이에 분명한 차이점이 있음을 주목한다.
- [0171] RTMP는 모든 비디오 코덱, 오디오 코덱, 및 자막 해결법을 지원하지는 않는다. 예를 들어, HEVC가 현재 지원되지 않는 것으로 보인다.
- [0172] 위키피디아(Wikipedia)에 따라, (RTMPT를 사용하여) HTTP를 통해 RTMP를 터널링(tunneling)하는 것이 가능하다. 그러나, RTMP 사양서에는 이 기능에 대한 설명이 없다. 전통적으로, RTMP는 주로 서버로부터 클라이언트로의 다운로드 스트리밍에 사용되었다.
- [0173] 3.3 MPEG2-TS 또는 프래그먼트화된 MP4와의 HTTP
- [0174] HTTP는 또한 라이브 업링크 비디오 스트리밍에 사용될 수 있다. HTTP의 이점은 HTTP 또는 소스 인증과 같은 모든 HTTP 지정 보안 기능이 재사용될 수 있다는 점이다. 여기서, MPEG2-TS [ISO/IEC 13818-1] 또는 프래그먼트화 MP4 [ISO/IEC 14996-12]가 업링크 스트리밍에 적절한 포맷이다. 또한, 인프라구조는 포트 80 또는 443에서의 HTTP 또는 HTTP 트래픽이 임의의 중간 방화벽을 통과하게 허용하도록 구성된다.
- [0175] 두가지 경우 모두, 이동 3GPP 디바이스 상의 HTTP 클라이언트가 HTTP 요청을 사용하여 HTTP 취합 서버에 HTTP 연결을 오픈한다. 라이브 업링크 비디오는 이때 요청의 HTTP 바디에 제공된다. HTTP 클라이언트는 비디오 콘텐츠를 직접 HTTP 바디로 파이핑하는데 HTTP 1.0 원칙을 사용할 수 있고, 또는 HTTP 1.1 청크 전달 인코딩을 사용할 수 있다. HTTP 청크 전달 인코딩은 클라이언트가 이어지는 HTTP 트랙잭션을 위해 (지속적인 TCP 연결) 설정된 TCP 연결을 재사용하도록 허용한다. TCP를 통한 RTMP의 경우와 같이, TCP가 정확하게 구성되는 것을 보장하는게 중요하다.
- [0176] 도 5는 HTTP에 대한 라이브 취합 포맷을 갖는 프래그먼트화 MP4를 사용하는 호출 흐름을 설명한다. 미디어 소스는 여기서 HTTP 클라이언트이다. 미디어 싱크는 여기서 HTTP 서버이고, 여기서 디-지터 버퍼로 도시된 후처리 체인에 수신 스트림 청크를 전달한다.
- [0177] 클라이언트는 라이브 업링크 스트림이 시작되기 이전에, 먼저 HTTP 취합 서버의 기능을 문의하고 체크한다. HTTP는 기능을 문의하는데 사용될 수 있다.
- [0178] 이어서, HTTP 클라이언트는 HTTP 요청의 바디 내에서의 HTTP 청크 전달을 사용해 라이브 스트림을 업로딩한다. HTTP 청크로 운반되는 프래그먼트는 ISO/IEF 14996-12에 따라 포맷된다. 무비 박스('moov')는 코덱, 코덱 구성, 및 잠재적으로 다른 정보를 포함한다. 클라이언트가 라이브 취합을 종료할 때, 서버는 HTTP 응답을 (이 경우에는 201 Created) 제공한다.
- [0179] MPEG2\_TS [ISO/IEC 13818-1]이 취합 포맷으로 사용될 때, 프래그먼트는 MPEG2-TS에 따라 포맷된다.
- [0180] 이 해결법에서는 비율 제어가 구현되어야 하고, 이는 바람직하게 TX 버퍼를 모니터링하고 그에 따라 미디어 소스를 조정한다.
- [0181] 4 해결법 요구 사항
- [0182] 다음의 해결법 특징은 (독점 리스트가 아니고) 라이브 업링크 스트리밍 프레임워크에서의 표준 작업을 위해 제안된다:
- [0183] NAT 및 방화벽의 존재가 고려되어야 한다. 취합 서버는 인프라구조 측에 배치되므로, 미디어 소스가 항상 3GPP UE 측에서의 통신 개시기에 위치한다.
- [0184] 해결법은 라이브 취합에만 초점을 맞춘다. 미디어 배급 또는 미디어 저장 실현은 취합과 독립적이어야 한다.
- [0185] 해결법은 인증 및 허가를 지원하여야 한다. 사용자 계정 또는 채널에 따라 라이브 취합 트래픽을 분리하는 것이 가능해야 한다.
- [0186] 인증된 트래픽 소스만이 비디오 트래픽을 취합 서버로 취합시킬 수 있어야 한다.
- [0187] 해결법은 3GPP QoS 프레임워크를 활용할 수 있어야 한다.
- [0188] 해결법은 이용가능한 액세스 네트워크에 (NR 또는 LTE와 같은) 따라, 다른 비트비율을 지원하여야 한다.

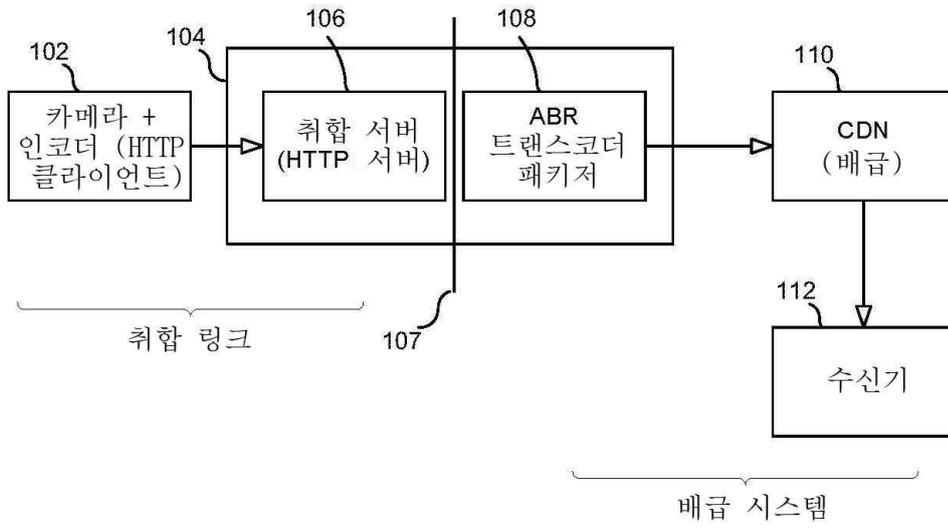
- [0189] 해결법은 적어도 H.264 및 HEVC 비디오 코덱을 지원하여야 한다.
- [0190] 해결법은 예를 들어, 이동성과 같이 변화하는 링크 조건에 따라 비트비율을 적응시킬 수 있어야 한다.
- [0191] 해결법은 품질/결과 비트비율 보다 대기시간을 (낮은 대기시간의 라이브 취합이 요구될 때), 또는 대기시간 보다 품질/결과 비트비율을 (더 나은 압축 효율 또는 재전송을 위한 시간이 있을 때) 우선화할 수 있어야 한다.
- [0192] 해결법은 기능 검색 또는 기능 협상을 지원하여야 한다.
- [0193] 5 제안
- [0194] 상기에 주어진 정보를 고려하도록 제안된다. 해결법 설계 요구 사항 및 고려 사항을 수집하기 위해 영구적인 문서 또는 기술적 보고서를 작성하도록 제안된다 (예를 들면, 섹션 2, 3, 및 4에 주어진 바와 같이).
- [0195] 또한, 라이브 업링크 비디오 스트리밍 프레임워크에 대한 기술적 사양서에 HTTP 기반의 해결법을 (섹션 3.3에서 설명된 바와 같이) 포함하도록 제안된다.
- [0196] 여기서는 본 내용의 다양한 실시예가 설명되었지만, 이들은 단지 예로 제시된 것이고 제한되지는 않음을 이해하여야 한다. 따라서, 본 발명의 폭 및 범위는 상기에 설명된 예시적인 실시예 중 임의의 것에 의해 제한되지 않아야 한다. 또한, 모든 가능한 변형에서 상술된 요소의 임의의 조합은 다른 방법으로 여기서 기술되지 않는 한 또는 다른 방법으로 문맥상에 명확히 모순되지 않는 한, 본 내용에 포함된다.
- [0197] 부가적으로, 도면에 도시되고 상기에 설명된 프로세스가 순차적인 단계로 도시되지만, 이는 단지 설명을 위한 것이다. 따라서, 일부 단계가 추가될 수 있고, 일부 단계가 생략될 수 있고, 단계의 순서가 재배열될 수 있고, 또한 일부 단계가 병렬로 실행될 수 있는 것으로 고려된다.

**부호의 설명**

- [0198] 102 : HTTP 클라이언트
- 104 : 미디어 프로세싱 유닛
- 106 : HTTP 취합 서버
- 108 : ABR 트랜스코더-패키저
- 110 : 콘텐츠 배급 네트워크(content distribution network, CDN)
- 112 : 단말 사용자 수신기
- 302 : 카메라
- 308 : 전송 버퍼
- 309 : 버퍼 모니터

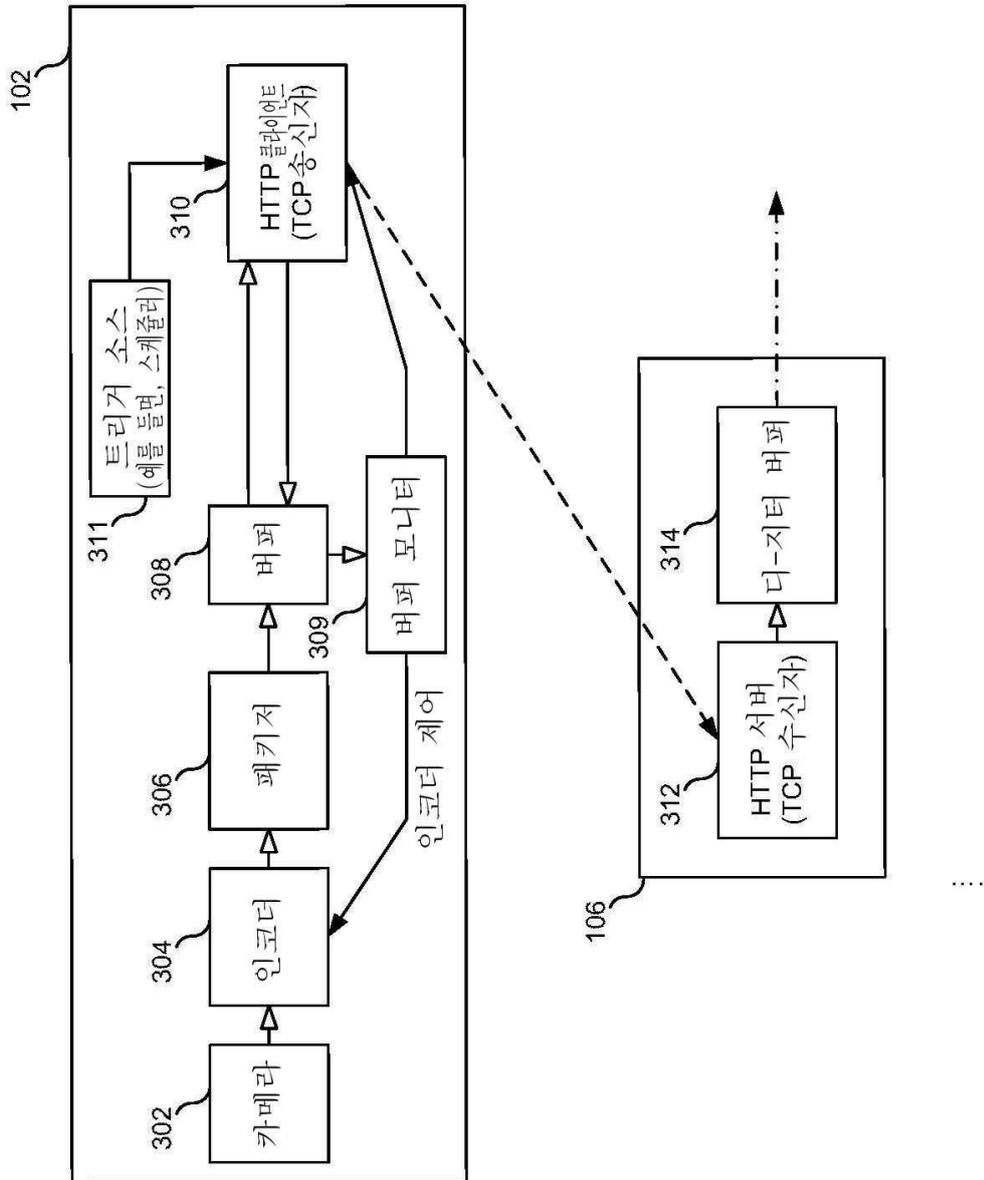
도면

도면1

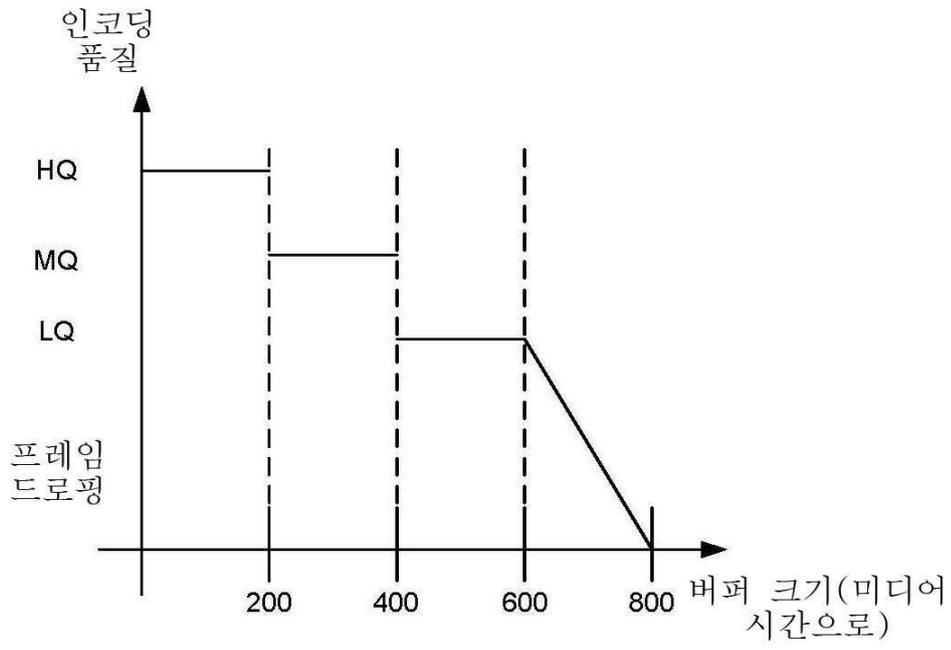




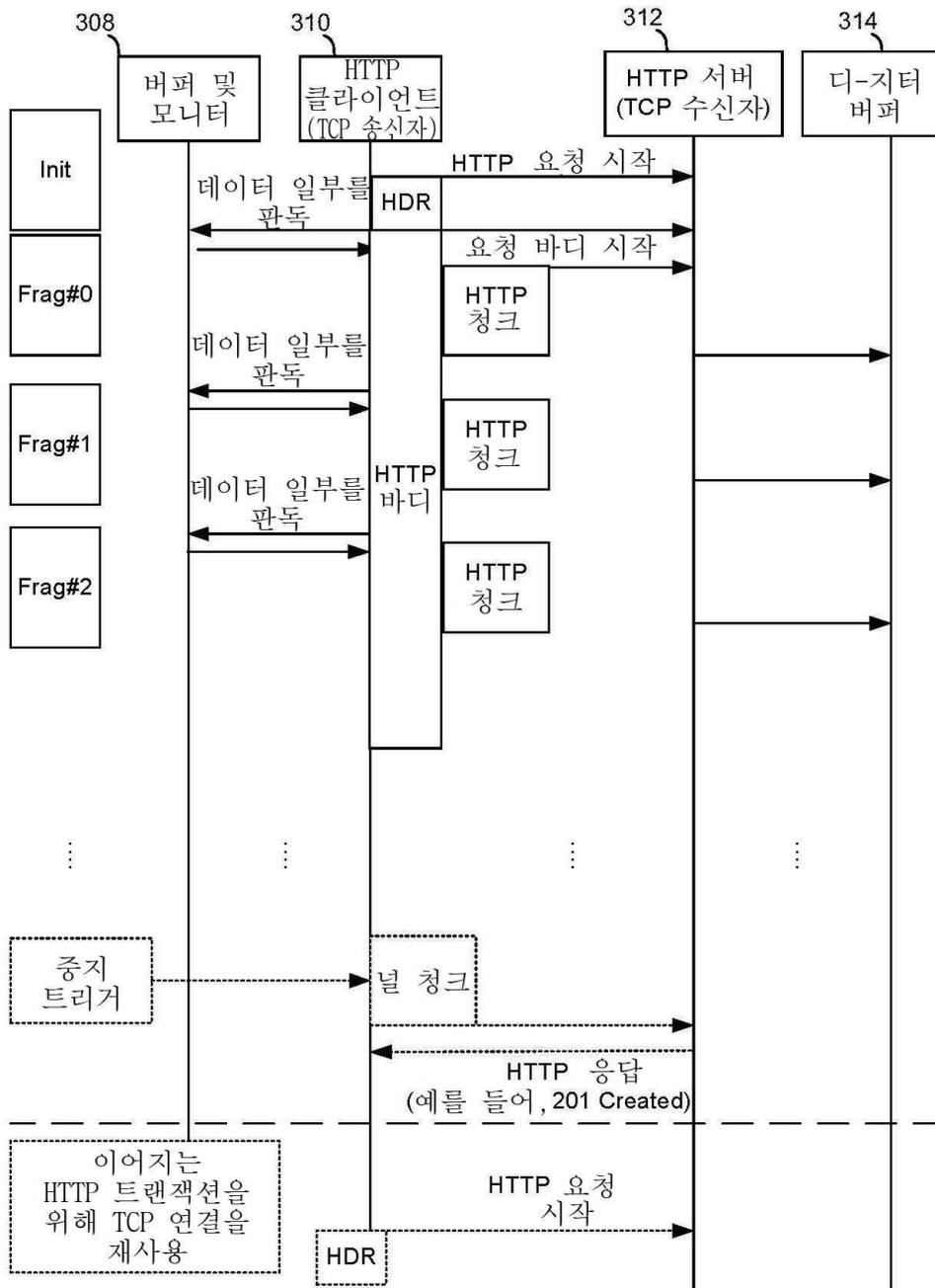
도면3



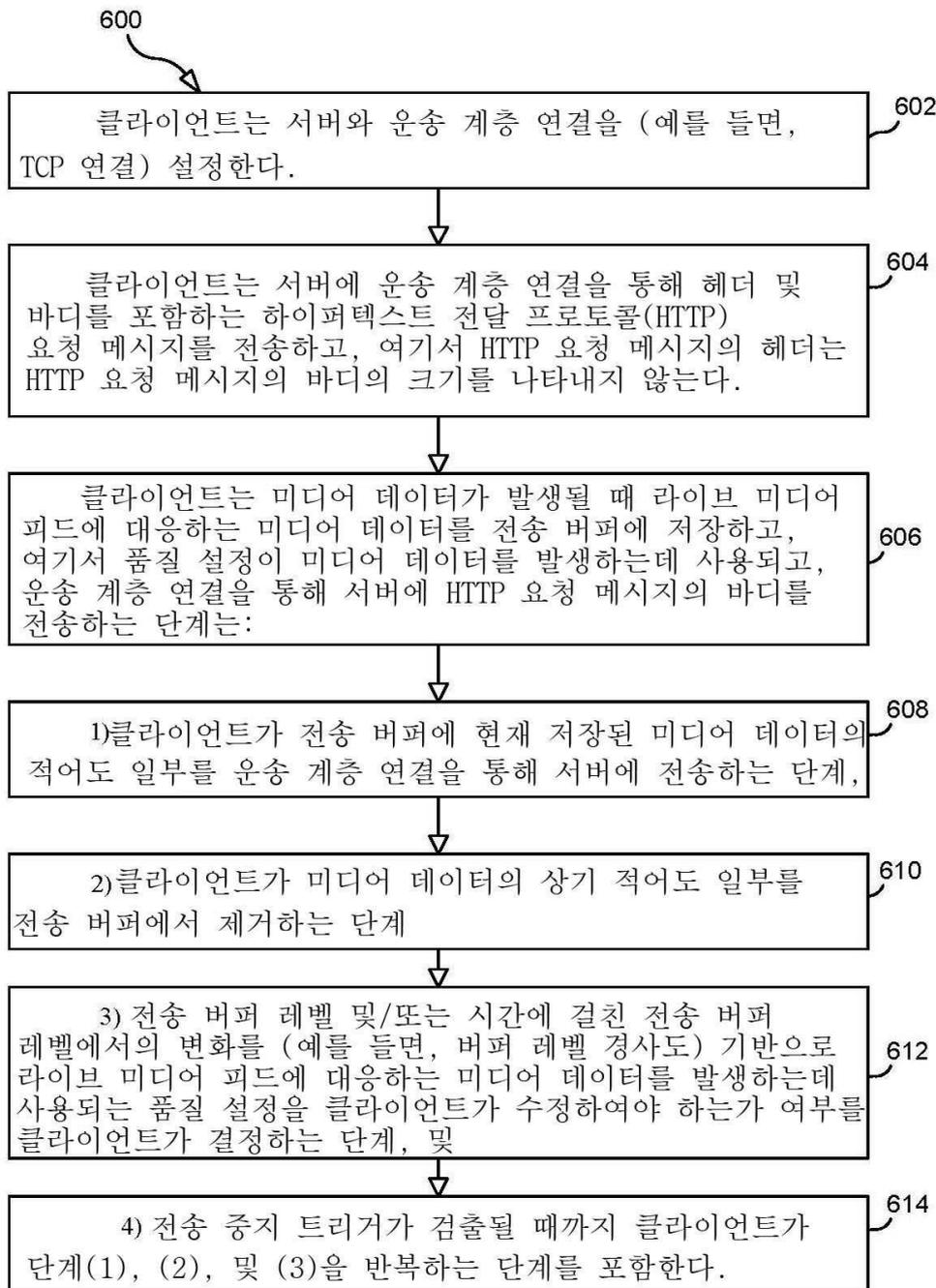
도면4



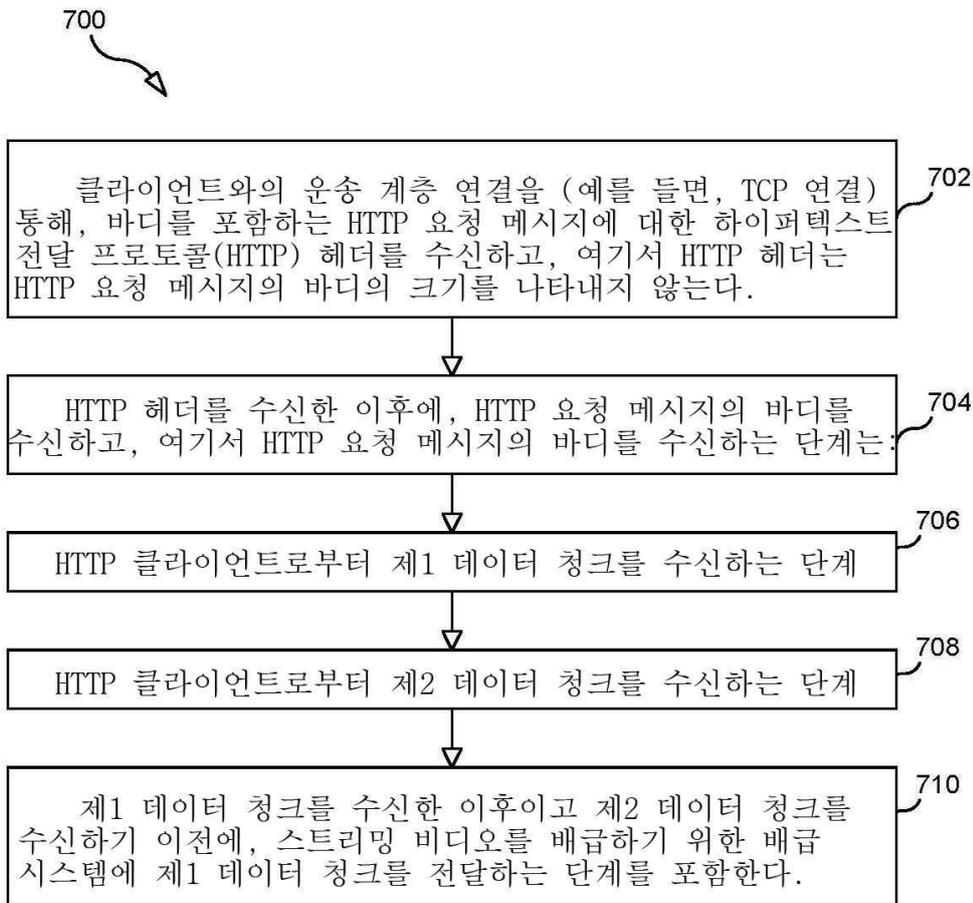
도면5



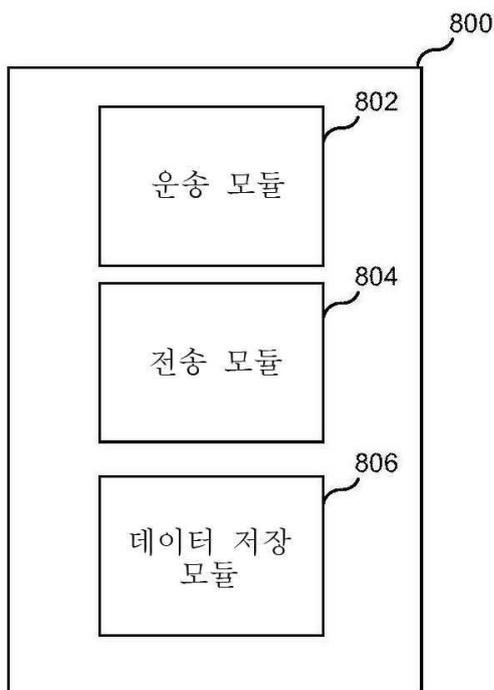
도면6



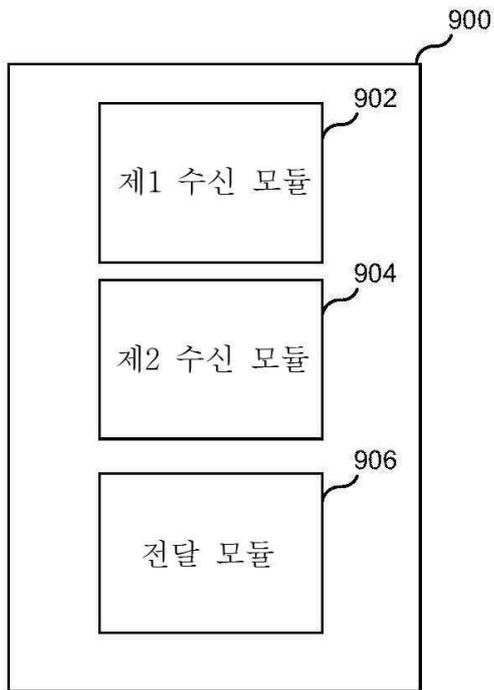
도면7



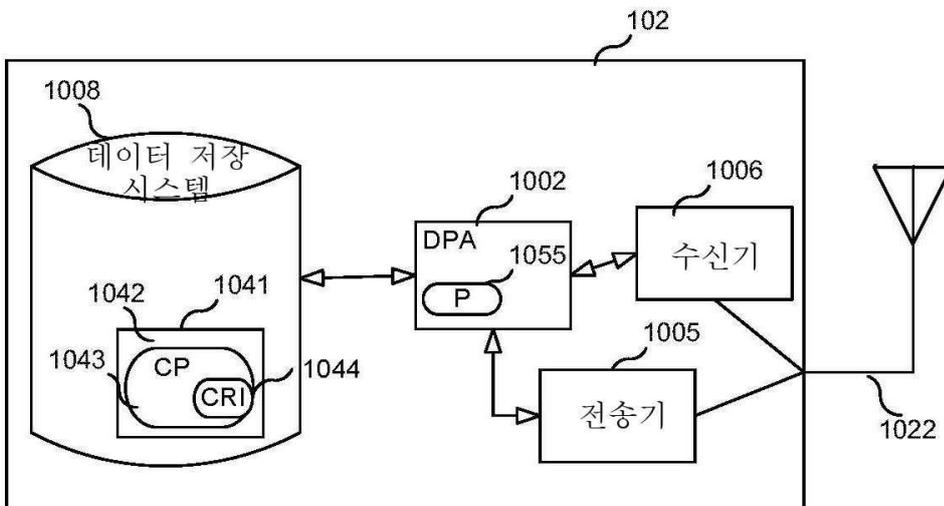
도면8



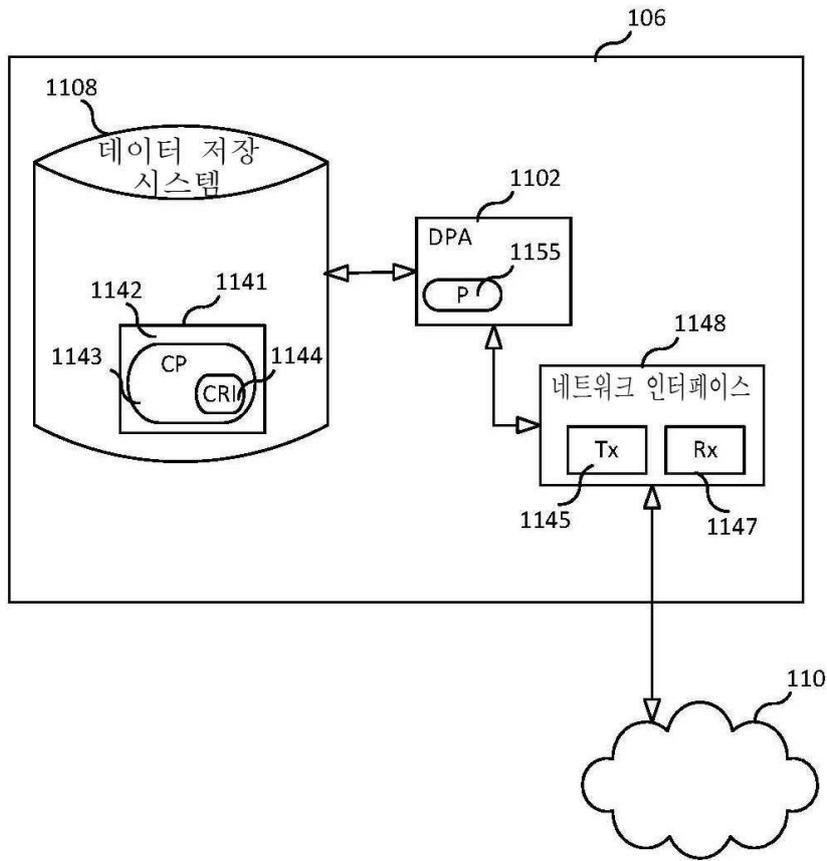
도면9



도면10



도면11



【심사관 직권보정사항】

【직권보정 1】

【보정항목】 청구범위

【보정세부항목】 청구항 11

【변경전】

제1항 내지 제4항 중 임의의 한 항에 있어서,

상기 라이브 피드는 라이브 배급을 위한 수신 엔터티(상기 서버(106))에 의해 더 처리될 수 있는 방법.

【변경후】

제1항 내지 제4항 중 임의의 한 항에 있어서,

상기 라이브 미디어 피드는 라이브 배급을 위한 수신 엔터티(상기 서버(106))에 의해 더 처리될 수 있는 방법.