



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2015-0002684
(43) 공개일자 2015년01월07일

- (51) 국제특허분류(Int. C1.)
H04N 21/2389 (2011.01) *H04L 29/08* (2006.01)
H04N 21/845 (2011.01) *H04L 29/06* (2006.01)
- (21) 출원번호 10-2014-7029670
- (22) 출원일자(국제) 2013년04월04일
심사청구일자 없음
- (85) 번역문제출일자 2014년10월22일
- (86) 국제출원번호 PCT/EP2013/057120
- (87) 국제공개번호 WO 2013/160080
국제공개일자 2013년10월31일
- (30) 우선권주장
12305464.5 2012년04월23일
유럽특허청(EPO)(EP)
- (71) 출원인
톰슨 라이센싱
프랑스 92130 이씨레물리노 잔 다르크 뤼 1-5
- (72) 발명자
뻬꼬니, 파비오
프랑스 에프-75014 빠리 뤼 들파 게떼 6
- (74) 대리인
양영준, 전경석, 백만기

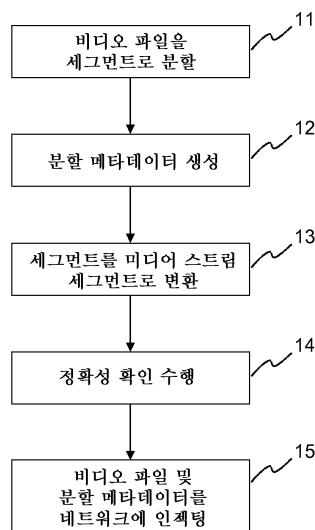
전체 청구항 수 : 총 10 항

(54) 발명의 명칭 피어 지원 비디오 분배

(57) 요 약

광대역 네트워크를 통한 비디오 콘텐츠의 피어 지원 분배를 위한 해결책이 개시된다. 비디오 파일은 세그먼트로 분할되고(11) 분할 메타데이터가 생성된다(12). 세그먼트는 카운터/타임스탬프 일치성을 갖는 미디어 스트림 세그먼트로 변환된다(13). 비디오 파일 및 분할 메타데이터는 피어 지원 비디오 전달 시스템에 인젝팅된다(15). 그 후 분할 메타데이터는 클라이언트에서 비디오 파일의 세그먼트를 생성하기(23) 위해 요구되는 비디오 파일의 오프셋의 다운로딩(22)을 개시하는 데 사용된다.

대 표 도 - 도3



특허청구의 범위

청구항 1

피어 지원 비디오 전달 시스템을 통해 프레임들을 포함하는 비디오 파일을 분배하기 위한 방법으로서,

상기 비디오 파일의 임의의 프레임은 유일한 시간적 세그먼트에만 존재하도록 상기 비디오 파일을 시간적 세그먼트들로 분할하는 단계(11) - 상기 각각의 시간적 세그먼트는 자립적(self-contained) 물리 오브젝트임 -;

적어도 상기 비디오 파일에 대한 상기 시간적 세그먼트들의 오프셋들, 연속성 카운터 값들, 및 세그먼트 시작 시간들을 포함하는 분할 메타데이터를 생성하는(12) 단계;

미디어 스트림 세그먼트들 내의 연속성 카운터들이 적절하게 증가되고 및/또는 상기 미디어 스트림 세그먼트들 내의 타임스탬프들은 유효하고 동기되도록, 상기 연속성 카운터들 및/또는 상기 타임스탬프들을 이용하는 미디어 스트림 포맷에 따른 상기 시간적 세그먼트를 미디어 스트림 세그먼트로 변환하는 단계(13); 및

상기 비디오 파일 및 상기 분할 메타데이터를 상기 피어 지원 비디오 전달 시스템에 인젝팅(injecting)하는 단계(15)

를 포함하는 비디오 파일 분배 방법.

청구항 2

제1항에 있어서,

상기 비디오 파일은 고정 기간(fixed-duration) 시간적 세그먼트들로 분할되는(11), 비디오 파일 분배 방법.

청구항 3

피어 지원 비디오 전달 시스템으로부터 프레임들을 포함하는 비디오 파일을 수신하기 위한 방법으로서,

적어도 상기 비디오 파일에 대한 상기 비디오 파일의 시간적 세그먼트들의 오프셋들, 연속성 카운터 값들, 및 세그먼트 시작 시간들을 포함하는 분할 메타데이터를 다운로딩하는 단계(21);

다운로드된 비디오 오프셋들을 결정하고 상기 결정된 비디오 오프셋들의 다운로딩을 개시하기 위해 상기 분할 메타데이터를 분석하는 단계(22);

상기 다운로드된 비디오 오프셋들로부터 비디오 세그먼트들을 생성하는 단계(23); 및

미디어 스트림 세그먼트들 내의 연속성 카운터들이 적절하게 증가되고 및/또는 상기 미디어 스트림 세그먼트들 내의 타임스탬프들은 유효하고 동기되도록, 상기 비디오 세그먼트들을 상기 연속성 카운터들 및/또는 상기 타임스탬프들을 이용하는 미디어 스트림 포맷에 따른 미디어 스트림 세그먼트들로 변환하는(24) 단계

를 포함하는 비디오 파일 수신 방법.

청구항 4

제1항 내지 제3항 중 어느 한 항에 있어서,

상기 비디오 파일은 MP4 파일 또는 MKV 파일인, 방법.

청구항 5

제1항 내지 제4항 중 어느 한 항에 있어서,

상기 시간적 세그먼트들은 MPEG 전송 스트림 세그먼트들로 변환되는(13, 24), 방법.

청구항 6

피어 지원 비디오 전달 시스템을 통해 프레임들을 포함하는 비디오 파일을 분배하기 위한 장치(30)로서,

상기 비디오 파일의 임의의 프레임은 유일한 시간적 세그먼트에만 존재하도록 상기 비디오 파일을 시간적 세그

먼트들로 분할하고(11), 상기 각각의 시간적 세그먼트는 차별적 물리 오브젝트이며, 적어도 상기 비디오 파일에 대한 상기 시간적 세그먼트들의 오프셋들, 연속성 카운터 값들, 및 세그먼트 시작 시간들을 포함하는 분할 메타데이터를 생성하기(12) 위한 분할기(31);

미디어 스트림 세그먼트들 내의 연속성 카운터들이 적절하게 증가되고 및/또는 상기 미디어 스트림 세그먼트들 내의 타임스탬프들은 유효하고 동기되도록, 상기 시간적 세그먼트들을 상기 연속성 카운터들 및/또는 상기 타임스탬프들을 이용하는 미디어 스트림 포맷에 따른 미디어 스트림 세그먼트들로 변환하기(13) 위한 변환기(32);

상기 비디오 파일 및 상기 분할 메타데이터를 상기 피어 지원 비디오 전달 시스템에 인젝팅하기(15) 위한 인젝터(34)

를 포함하는 비디오 파일 분배 장치(30).

청구항 7

제6항에 있어서,

상기 분할기(31)는 상기 비디오 파일을 고정 기간 시간적 세그먼트들로 분할하는(11), 비디오 파일 분배 장치(30).

청구항 8

피어 지원 비디오 전달 시스템으로부터 프레임들을 포함하는 비디오 파일을 수신하기 위한 장치(40)로서,

적어도 상기 비디오 파일에 대한 상기 비디오 파일의 시간적 세그먼트들의 오프셋들, 연속성 카운터 값들, 및 세그먼트 시작 시간들을 포함하는 분할 메타데이터를 다운로딩하기(21) 위한 네트워크 커넥터(41);

다운로딩된 비디오 오프셋들을 결정하고 상기 결정된 비디오 오프셋들의 다운로딩을 개시하기 위하여 상기 분할 메타데이터를 분석하기(22) 위한 분석기(42);

상기 다운로딩된 비디오 오프셋들로부터 비디오 세그먼트들을 생성하기(23) 위한 세그먼트 생성기(43); 및

미디어 스트림 세그먼트들 내의 연속성 카운터들이 적절하게 증가되고 및/또는 상기 미디어 스트림 세그먼트들 내의 타임스탬프들은 유효하고 동기되도록, 상기 비디오 세그먼트들을 상기 연속성 카운터들 및/또는 상기 타임스탬프들을 이용하는 미디어 스트림 포맷에 따른 미디어 스트림 세그먼트들로 변환하기(24) 위한 변환기(44)

를 포함하는 비디오 파일 수신 장치(40).

청구항 9

제6항 내지 제8항 중 어느 한 항에 있어서,

상기 비디오 파일은 MP4 파일 또는 MKV 파일인, 장치(30, 40).

청구항 10

제6항 내지 제9항 중 어느 한 항에 있어서,

상기 변환기(32, 44)는 상기 시간적 세그먼트들을 MPEG 전송 스트림 세그먼트들로 변환하는(13, 24), 장치(30, 40).

명세서

기술분야

[0001] 본 발명은 광대역 네트워크를 통한 비디오 콘텐츠의 피어 지원 분배를 위한 해결책에 관한 것이고, 더 구체적으로, 신속한 빨리 감기 또는 되감기 동작뿐만 아니라, 재생하기까지의 짧은 베파링 시간을 가능하게 하는 비디오 콘텐츠의 피어 지원 분배를 위한 해결책에 관한 것이다.

배경기술

[0002] 비디오는 오늘날 인터넷상에서 소비되는 콘텐츠의 큰 부분을 차지한다. 비디오 서비스를 제공하는 복수의 웹사

이트들은 무수히 많은 상이한 디바이스, 예를 들어, PC, TV, 스마트폰, 및 태블릿으로 비디오 클립 또는 영화들을 끊임없이 전달한다. 광가입자망(fiber-to-the-home) 및 다른 고속 광대역 기술들이 점점 보편화되어 비디오 콘텐츠의 소비는 증가할 것으로 기대된다.

[0003] 비디오 전달의 방대한 광대역 요구를 지원하기 위하여, 서비스 제공자는 CDNs(Content Delivery Networks), 사용자들이 콘텐츠를 다운로딩하는 지역적으로 분배된 서버들의 세트를 활용한다. 콘텐츠 분배의 효율을 더 개선하기 위하여, 사용자가 CDN 및 다른 사용자들의 셋톱박스 또는 홈 게이트웨이 양자 모두로부터 콘텐츠를 다운로딩하는 피어 지원 아키텍쳐가 제안되었다. 이러한 경우에, 셋톱박스 또는 게이트웨이는 자신의 로컬 하드 디스크 또는 플래시 메모리에 다수의 비디오 클립들을 저장하고, 그러한 클립의 단편들(fragments)을 다른 사용들에게 업로딩하기 위하여 사용자의 홈 연결의 이용 가능한 업로딩 광대역을 사용한다.

[0004] 예를 들어, Wang 등의 논문 "Is playing-as-downloading feasible in an eMule P2P file sharing system?", J. Zhejiang Univ.-Sci.C (Comput. & Electron.), 제11호 (2010년), 페이지 465 내지 475에서, 재생 중 다운로딩(playing-as-downloading) 능력을 갖는 미디어 스트리밍 지원을 포함하는 P2P(peer-to-peer) 스웜(swarm) 기술들을 위한 해결책을 논의한다. P2P 본래의 다운로딩 속도를 유지하면서, P2P 파일 공유 시스템이 VOD(video-on-demand) 기능을 지원할 수 있게 하는 WDCD(wish driven chunk distribution)로 명칭되는 새로운 알고리즘이 소개된다. 다음 재생 빈도(next-play-frequency)로 명칭되는 새로운 파라미터는 다운로딩 및 스트리밍 요청들 간의 응답 벨런스를 공략(strike)하기 위하여 콘텐츠 청크(chunk)에 추가된다.

발명의 내용

해결하려는 과제

[0005] 본 발명은 피어 지원 비디오 분배 시스템, 더 구체적으로 HLS(HTTP Live Streaming)와 같은 프로토콜을 지원하기 위하여 비디오 세그먼트의 연동적인(on-the-fly) 생성 문제에 초점을 맞춘다. HLS 표준은 다중 소스로부터의 다운로딩 및 적용적 스트리밍 레이트와 같은 바람직한 특징들을 지원하는 비디오 전달을 위한 프로토콜을 설명한다(<http://tools.ietf.org/html/draft-pantos-http-live-streaming-07> 참조). 그러나, HLS의 한 가지 제약은 그것이 MPEG-TS(MPEG Transport Stream) 컨테이너의 사용을 요구한다는 것이다. 따라서, 비디오가 기본적으로 MP4 또는 MKV(Matroska)와 같은 다른 포맷으로 저장되어 있다면, 그것은 HLS를 통하여 전달될 수 있기 전에 MPEG-TS로 변환되어야 한다. 또한, 세그먼트가 HLS 플레이어에 의해 함께 스티칭(stitched back)될 때 MPEG-TS 연속성 카운터 및 타임스탬프가 일치되도록 MPEG-TS 스트림은 청크, 또는 미디어 세그먼트로 분할되어야 한다. 본 단계는 분할(segmentation)로서 공지되어 있다. 다음 텍스트에서 "세그먼트" 및 "청크" 용어들은 구별 없이 사용될 것이다.

[0006] HLS 호환 분할을 제작하기 위하여 변환 및 분할을 수행하기 위한 현재 다수의 도구들이 존재한다. 그러나, 그 것들은 모두 전체 소스 파일을 입력으로서 취한다. 이것이 피어 지원 시스템의 문제인 이유를 보이기 위해 다음과을 고려하자. 광범위한 사용자 디바이스(PC, 스마트폰 등)를 지원하기 위하여, 시스템은 주어진 비디오를 HTTP를 통한 단일 MP4 파일로서, 또는 HLS를 통한 다중 MPEG-TS 세그먼트로서 전달할 수 있다. 그러나, 저장 공간을 절약하기 위하여 시스템은 각각의 비디오를 MP4 파일로만 저장한다. 사용자가 HLS를 통해 비디오를 요청할 때마다, 로컬 피어-투-피어 클라이언트는 MP4 콘텐츠를 다운로딩하고, 그것을 MPEG-TS로 변환하며, 결과를 HLS 청크로 분할한다. 그러나, 기존 HLS 분할기 도구는 전체 파일에서만 동작하기 때문에, 피어-투-피어 클라이언트는, HLS 청크가 생성되고 재생을 시작할 수 있기 전에, 전체 MP4 비디오 다운로딩을 끝내야 한다. 이것은 명백하게 수용가능하지 않다.

과제의 해결 수단

[0007] 네트워크를 통한 비디오 콘텐츠의 피어 지원 분배를 위한 개선된 해결책을 제안하는 것이 본 발명의 목적이다.

[0008] 본 발명에 따라, 피어 지원 비디오 전달 시스템을 통해 프레임들을 포함하는 비디오 파일을 분배하기 위한 방법은,

[0009] 비디오 파일의 임의의 프레임은 유일한 시간적 세그먼트에만 존재하도록 비디오 파일을 시간적 세그먼트로 분할하는 단계 - 상기 각각의 시간적 세그먼트는 자립적(self-contained) 물리 오브젝트임 -,

[0010] 적어도 비디오 파일에 대한 시간적 세그먼트의 오프셋, 연속성 카운터 값, 및 세그먼트 시작 시간을 포함하는 분할 메타데이터를 생성하는 단계,

- [0011] 미디어 스트림 세그먼트 내의 연속성 카운터가 적절하게 증가되고 및/또는 미디어 스트림 세그먼트 내의 타임스탬프는 유효하고 동기되도록, 시간적 세그먼트를 연속성 카운터 및/또는 타임스탬프를 이용하는 미디어 스트림 포맷에 따른 미디어 스트림 세그먼트로 변환하는 단계, 및
- [0012] 비디오 파일 및 분할 메타데이터를 피어 지원 비디오 전달 시스템에 인젝팅(injecting)하는 단계를 포함한다.
- [0013] 그에 따라, 피어 지원 비디오 전달 시스템을 통해 프레임을 포함하는 비디오 파일을 분배하기 위한 장치는,
- [0014] 비디오 파일의 임의의 프레임은 유일한 시간적 세그먼트에만 존재하도록 비디오 파일을 시간적 세그먼트로 분할하고, 각각의 시간적 세그먼트는 자립적 물리 오브젝트이며, 적어도 비디오 파일에 대한 시간적 세그먼트의 오프셋, 연속성 카운터 값, 및 세그먼트 시작 시간을 포함하는 분할 메타데이터를 생성하기 위한 분할기,
- [0015] 미디어 스트림 세그먼트 내의 연속성 카운터가 적절하게 증가되고 및/또는 미디어 스트림 세그먼트 내의 타임스탬프는 유효하고 동기되도록, 시간적 세그먼트를 연속성 카운터 및/또는 타임스탬프를 이용하는 미디어 스트림 포맷에 따른 미디어 스트림 세그먼트로 변환하기 위한 변환기, 및
- [0016] 비디오 파일 및 분할 메타데이터를 피어 지원 비디오 전달 시스템에 인젝팅하기 위한 인젝터를 포함한다.
- [0017] 본 발명의 일 측면에 따라, 비디오 파일, 예를 들어, MP4 파일 또는 MKV 파일은 먼저 세그먼트로 분할된다. 바람직하게, 이러한 세그먼트는 고정 기간(fixed-duration) 세그먼트이다. 분할 메타데이터는 분할 동안에 생성된다. 결과 세그먼트는 카운터/타임스탬프 일치성을 갖는 미디어 스트림 세그먼트, 예를 들어, MPEG-TS 세그먼트로 변환된다. 선택적으로, 정확성(sanity) 확인이 미디어 스트림 세그먼트에 수행된다. 그 뒤에, 비디오 파일 및 분할 메타데이터는 피어 지원 비디오 전달 시스템에 인젝팅된다.
- [0018] 피어 지원 비디오 전달 시스템으로부터 비디오 파일을 수신하기 위한 방법은,
- [0019] 적어도 비디오 파일에 대한 비디오 파일의 시간적 세그먼트의 오프셋, 연속성 카운터 값, 및 세그먼트 시작 시간을 포함하는 분할 메타데이터를 다운로딩하는 단계,
- [0020] 다운로딩될 비디오 오프셋을 결정하고 결정된 비디오 오프셋의 다운로딩을 개시하기 위하여 분할 메타데이터를 분석하는 단계,
- [0021] 다운로딩된 비디오 오프셋으로부터 비디오 세그먼트를 생성하는 단계, 및
- [0022] 미디어 스트림 세그먼트 내의 연속성 카운터가 적절하게 증가되고 및/또는 미디어 스트림 세그먼트 내의 타임스탬프는 유효하고 동기되도록, 비디오 세그먼트를 연속성 카운터 및/또는 타임스탬프를 이용하는 미디어 스트림 포맷에 따른 미디어 스트림 세그먼트로 변환하는 단계를 포함한다.
- [0023] 그에 따라, 피어 지원 비디오 전달 시스템으로부터 비디오 파일을 수신하기 위한 장치는,
- [0024] 적어도 비디오 파일에 대한 비디오 파일의 시간적 세그먼트의 오프셋, 연속성 카운터 값, 및 세그먼트 시작 시간을 포함하는 분할 메타데이터를 다운로딩하기 위한 네트워크 커넥터,
- [0025] 다운로딩될 비디오 오프셋을 결정하고 결정된 비디오 오프셋의 다운로딩을 개시하기 위하여 분할 메타데이터를 분석하기 위한 분석기,
- [0026] 다운로딩된 비디오 오프셋으로부터 비디오 세그먼트를 생성하기 위한 세그먼트 생성기, 및
- [0027] 미디어 스트림 세그먼트 내의 연속성 카운터가 적절하게 증가되고 및/또는 미디어 스트림 세그먼트 내의 타임스탬프는 유효하고 동기되도록, 연속성 카운터 및/또는 타임스탬프를 이용하는 미디어 스트림 포맷에 따른 비디오 세그먼트를 미디어 스트림 세그먼트로 변환하기 위한 변환기를 포함한다.
- [0028] 미디어 스트림 세그먼트를 재생하기 위하여, 제1 단계에서 분할 메타데이터 파일이 다운로딩된다. 그 후 분할 메타데이터가 어느 비디오 오프셋이, 예를 들어, MP4 또는 MKV 오프셋이 다운로딩 되어야 하는지를 결정하기 위하여 분석된다. 그 후에, 결정된 비디오 오프셋의 다운로딩이 개시된다. 그 후, 비디오 세그먼트는 다운로딩된 비디오 오프셋으로부터 생성되며, 이는 카운터/타임스탬프 일치성을 갖는 미디어 스트림 세그먼트, 예를 들어, MPEG-TS 세그먼트로 최종 변환된다.
- [0029] 본 발명에 따른 해결책은 피어가 전체 소스 비디오 파일을 다운로딩하지 않고 미디어 스트림 세그먼트를 제작하게 할 수 있다. 이것은 재생하기까지의 짧은 베파링 시간을 가능하게 하는데, 즉, 신속한 빨리 감기 또는 되감기 동작들뿐만 아니라 전체 파일이 다운로딩되기 전에 재생이 시작될 수 있다. 또한, 파일이 하나의 포맷으로

만 저장되기 때문에, 본 해결책은 저장 요구에 관하여 매우 효율적이다. 게다가, 그것은 존재하는 표준들과 호환된다.

[0030] 더 바람직한 이해를 위해 본 발명은 이제 도면들을 참조하여 다음 설명에서 더 상세하게 설명될 것이다. 본 발명은 이러한 예시적인 실시예에 제한되지 않으며, 또한 구체화된 특징들은 첨부 청구항들에서 정의된 바와 같은 본 발명의 범위를 벗어나지 않고 편의상 결합 및/또는 수정될 수 있는 것으로 이해된다.

도면의 간단한 설명

[0031] 도 1은 피어 지원 비디오 전달 시스템을 도시한다.

도 2는 피어 지원 비디오 전달에 대한 본 발명에 따른 방법을 도시적으로 예시한다.

도 3은 MP4 파일의 분할 및 분할 메타데이터 파일의 생성을 예시한다.

도 4는 HLS 세그먼트의 재생성을 도시한다.

도 5는 피어 지원 비디오 전달 시스템에서 인젝팅을 위한 비디오 파일을 준비하는 디바이스를 도시적으로 예시한다.

도 6은 피어 지원 비디오 전달 시스템으로부터 비디오 파일을 수신하기 위한 디바이스를 도시적으로 예시한다.

발명을 실시하기 위한 구체적인 내용

[0032] 다음 설명에서, MP4 대 HLS 변환 연동의 경우가 고려될 것이다. 그러나, 본 발명은 또한 다른 입력 포맷, 예를 들어, MKV 대 HLS 변환에 대하여 동작한다. 또한, 본 발명이 기반한 매커니즘은 HLS에 특정하지 않고, DASH(Dynamic Adaptive Streaming over HTTP)와 같은 유사한 타겟 프로토콜들에 적용될 수 있다.

[0033] 도 1은 네트워크(1)를 통한 비디오 전달에 대한 피어 지원 시스템의 상이한 컴포넌트를 도시한다. 피어-투-피어 클라이언트는 네트워크(1)에 연결된 각각의 홈 게이트웨이(2)에서 실행하는 것으로 가정된다. 비디오는 PC(3) 및 태블릿(4)과 같은 상이한 사용자 디바이스로부터 시청될 수 있다. 사용자가 비디오를 시청하기 시작할 때, 게이트웨이(2) 상의 피어-투-피어 클라이언트는 다른 피어-투-피어 클라이언트 및/또는 CDN(5)으로부터 비디오 데이터의 조각들을 다운로딩하고, 스트리밍과 유사하게 그것을 사용자 디바이스로 보낸다. 물론, 피어-투-피어 클라이언트는 홈 게이트웨이(2) 대신 사용자 디바이스에서 또한 실행할 수 있다.

[0034] 도 2는 피어 지원 비디오 전달을 위한 본 발명에 따른 방법을 도시적으로 예시한다. 본 방법은 2개의 단계를 필수적으로 포함한다. 제1 단계(10)에서, MP4파일은 분할되고 분할 메타데이터 파일이 생성된다. 제2 단계(20)에서, HLS 세그먼트가 재생성된다.

[0035] 제1 단계(10)는 MP4 파일이 먼저 시스템에, 즉, 피어-투-피어 저장소뿐만 아니라 CDN에 인젝팅될 때마다 실행되고, 다운로딩이 이용가능해진다. 이 단계는 파일을 분석하고, 피어-투-피어 클라이언트가 HLS 청크 연동을 생성해야 할 때마다 실행되는, 제2 단계(20) 동안 사용될 일부 메타데이터를 생성한다.

[0036] 도 3은 MP4 파일의 분할 및 분할 메타데이터 파일의 생성을 더 상세하게 설명한다. 이 단계는 다수의 단계들을 포함한다. 먼저 MP4 파일이 MP4 세그먼트, 바람직하게는 고정 기간 세그먼트로 분할되고(11), 분할 메타데이터가 생성된다(12). 이러한 세그먼트는 카운터/타임스탬프 일치성을 갖는 MPEG-TS 세그먼트로 변환된다(13). 그 후, MPEG-TS 정확성 확인이 수행되고(14), 비디오 파일 및 분할 메타데이터가 CDN 및 P2P 네트워크에 인젝팅된다(15).

[0037] 분할 단계(11)는 전체 MP4 소스 파일을 저장하는 컴퓨터에서 실행된다. 컴퓨터는 모든 타겟 세그먼트를 생성하기 위하여 분할기 도구를 호출한다. 예를 들어, 파일이 k개의 청크로 분할되어야 하는 경우에, 도구는 각각의 세그먼트 당 한번씩, k번 호출된다. 각각의 호출에서, 도구는 세그먼트 시작 시간 및 종료 시간을 파라미터로서 수신한다. 예를 들어, 세그먼트가 5초 길이이고, 제3 세그먼트가 생성되어야 하는 경우에, 도구는 "시작 시간 t=10" 및 "종료 시간 t=15"으로 호출된다.

[0038] 여기서 핵심적인 문제는 생성된 세그먼트가 오버래핑하는 비디오/오디오 프레임을 포함해서는 안 된다는 것인데, 즉, 프레임은 하나를 초과하는 세그먼트에 존재해서는 안 된다. 마찬가지로, 모든 프레임들은 적어도 하나의 세그먼트에 존재해야 한다. 이러한 2개의 조건들은, 결과 세그먼트를 스티칭하는 것이 연속적이면서 오류 없는 재생을 초래할 것을 보장한다.

오늘날, 존재하는 도구들은 이러한 타입의 독창적인(out-of-the-box) 분할을 지원하지 않는다. 예를 들어, 대중적인 도구 MP4Box는 "시작 시간" 및 "종료 시간" 파라미터에 의해 특정되는 개별 청크들을 생성할 수 있다. 그러나, 결과 청크들은 오버래핑하는 비디오/오디오 프레임들을 포함할 것이다. 그럼에도 불구하고, 그것의 소스 코드를 수정하지 않고 MP4Box가 사용되도록 하는 이 문제에 대한 차선책이 존재한다. MP4Box의 최신 개발 버전은 스플릿 청크(-split-chunk) S:E 옵션을 지원한다. 이 옵션은 시작 시간 S 및 종료 시간 E에 의해 특정된 청크를 생성한다. 그러나, 본 도구는 특정 프레임에 그것을 정렬시키기 위하여 시작 시간을 조정할 수 있다. 예를 들어, "스플릿 청크 5:10"으로 도구를 호출하는 것은 t=4.96에서 시작하고 t=10에서 종료하는 청크를 초래할 수 있다. 이전 청크가 t=5에서 끝난다면, 그 후 일부 프레임은 양쪽의 청크들에서 존재할 수 있고, 청크가 함께 스티칭될 때 문제를 야기할 수 있다.

이러한 문제를 위한 차선책은 2개의 패스에서 MP4Box를 호출하는 것을 포함한다. 제1 패스 동안, 그것은 (5초의 기간 동안) 다음 읍션들을 이용하여 호출된다.

세그먼트 2 스플릿 청크 5:10

세그먼트 3 스플릿 청크 10:15

세그먼트 4 스플릿 청크 15:20

각각의 호출에서 MP4Box는, 시작 시간이 특정 값, 예를 들어, $t=15$ 대신에 $t=14.96$ 으로 조정된 것을 출력할 수 있다. 이러한 수정된 시작 시간들을 임시 파일에 저장된다.

제2 페스 동안, MP4Box는 다시 모든 청크들을 생성하기 위해 호출되지만, 다음 청크의 시작 시간에 따라 청크의 총길이 시간을 대체한다. 살기 실시간에서 이것을 다음 유통들을 초래한다.

세그먼트 1 스플릿 청크 0:5

세그먼트 2 스플릿 청크 5:10

세그먼트 3 스플렁 청크 10:14 96

세그먼트 4에 대한 옵션이 시작 시간을 $t=15$ 에 특정했다는 것에 주목하자. 이것은 실제 제1 페스에서 실시했던 경관 같이 MP4Box가 청크의 시작 시간을 14.96으로 조정한 것인가 때문에 의도되다.

제2 페스 동안, 시스템은 MP4Box에 의해 액세스된 MP4 소스 파일 오프셋들을 모니터하고 그것들은 분할 메타데이터 파일에 저장한다. 최종적으로 조정된 시작 시간은 또한 분할 메타데이터 파일에 추가된다.

다음은 JSON(JavaScript Object Notation)의 코딩을 사용하는 파일 메타데이터 파일의 예시이다.

3

"index": 1

"start": 0

"end": 5.00

```
[0065]     "mp4offsets": [
[0066]         [0,303103],
[0067]         [8441856,8444766]
[0068]     ],
[0069]     "continuity": "..."
[0070] },
[0071] {
[0072]     "index": 2,
[0073]     "start": 5.00,
[0074]     "end": 10.00,
[0075]     "mp4offsets": [
[0076]         [0,40959],
[0077]         [290816,876543],
[0078]         [925696,929791],
[0079]         [995328,1011711],
[0080]         [8441856,8444766]
[0081]     ],
[0082]     "continuity": "..."
[0083] },
[0084] {
[0085]     "index": 3,
[0086]     "start": 10.00,
[0087]     "end": 14.96,
[0088]     "mp4offsets": [
[0089]         [0,40959],
[0090]         [847872,1658879],
[0091]         [1687552,1691647],
[0092]         [1789952,1810431],
[0093]         [8441856,8444766]
[0094]     ],
[0095]     "continuity": "..."
[0096] },
[0097] {
[0098]     "index": 4,
[0099]     "begin": 14.96,
[0100]     "end": 19.88,
```

```

[0101]     "mp4offsets": [
[0102]         [0, 40959],
[0103]         [1626112, 2400255],
[0104]         [2437120, 2441215],
[0105]         [2531328, 2547711],
[0106]         [8441856, 8444766]
[0107]     ],
[0108]     "continuity": "...",
[0109] },
[0110] ]

```

[0111] MPEG-TS 연속성 카운터 값은 본 실시예 파일에서 생략되었다는 것에 주의하자.

[0112] 본 발명에 활용된 이해들 중 하나는 특정 세그먼트를 생성하는 경우에 분할기 도구가 전체 소스 MP4 파일을 액세스하지 않을 것이라는 사실이다. 예를 들어, 제3 세그먼트를 생성하는 경우에 본 도구는 청크를 생성하기 위해 추출되어야하는 비디오/오디오 프레임에 대응하는 오프셋의 범위 외에 MP4 메타데이터가 위치된 파일의 시작 또는 종료에도 액세스할 것이다.

[0113] 따라서, 본 발명에 따라 분할기 도구는 도구가 관독하는 모든 소스 MP4 파일 오프셋들을 결정하기 위해 각각의 호출이 있을 때마다 모니터될 것이다. 이러한 오프셋들은 분할 메타데이터 파일에 저장되고 재생성 단계(20)에서 사용될 것이다. 본 도구에 의해 관독된 소스 MP4 파일 오프셋들을 모니터하기 위한 다수의 방법들이 있다. 하나의 가능성은 모든 파일 시스템 호출을 추적하기 위한 에스트레이스(strace) 유닉스 명령을 이용하여 도구를 호출하는 것이다.

[0114] 변환 단계(13) 동안 모든 MP4 세그먼트는 MPEG-TS 컨테이너로 변환되고, 단일 MPEG-TS 파일로 함께 스티칭된다. 단계(13)는 스티칭된 MPEG-TS 파일이 오류 없이 재생될 수 있도록 해야한다. 이것은 연속성 카운터가 적절하게 증가되고, PCR 및 PTS/DTS 타임스탬프가 유효하고 동기하도록 보장하는 것을 포함한다.

[0115] 이 단계를 수행하기 위한 하나의 가능성은 MP4에서 MPEG-TS로의 변환을 지원하는 ffmpeg 도구 (<http://ffmpeg.org/> 참조)를 활용하는 것이다. 그러나, ffmpeg에 의해 생성된 세그먼트는 연속성 카운터, PCR(Program Clock Reference) 및 PTS(Presentation Time Stamp)/DTS(Decode Time Stamp)의 값이 부정확할 수 있기 때문에 단순히 연속적으로 스티칭될 수 없다. 해결책은 ffmpeg에 의해 제작된 MPEG-TS 세그먼트를 연결시키고, 연속성 카운터, PCR 및 PTS/DTS 값을 맞추는 새나타이저(sanitizer) 도구를 실행하는 것이다. 이것은 다음과 같이 성취될 수 있다.

[0116] 연속성 카운터에 있어서, 각각의 세그먼트 전환에서의 연속성 카운터 값은 이전 세그먼트에서 관측된 최종 카운터 값에 일치되도록 수정된다. 이를 최종 카운터 값은 분할 메타데이터 파일에 저장된다.

[0117] PCR/PTS/DTS 값에 있어서, 각각의 세그먼트가 개별적으로 상이한 시간 기준을 갖고 생성되기 때문에, 각각의 세그먼트 전환에서 PCR/PTS/DTS 값은 0으로 리셋될 것이다. 그러나, 간단한 조정으로 충분하다. PCR/PTS/DTS 값은, 분할 메타데이터 파일에서 찾을 수 있는 세그먼트의 시작 시간과 동일한 값 만큼 증가된다.

[0118] MPEG-TS 정확성 확인 단계(14)는 선택사항이다. 그것은 불연속성 또는 부정확한 타임스탬프 값으로 인한 재생 오류가 없다는 것을 확인하기 위하여 변환 단계(13)에서 제작된 MPEG-TS를 확인한다.

[0119] 인젝팅 단계(15)는 MP4 파일을 CDN 및 피어-투-피어 저장 공간에 인젝팅하는 것을 포함한다. 분할 메타데이터 파일은 피어들의 용이한 액세스를 위해 P2P 추적기에 저장될 수 있다. MPEG-TS 세그먼트는 CDN측에서의 연동 변환을 피하기 위하여 CDN에 또한 인젝팅될 수 있다.

[0120] 도 4는 HLS 세그먼트의 재생성 단계(20)를 더 상세하게 도시한다. 또한 본 단계(20)는 다수의 단계들을 포함한다. 제1 단계(21)에서, 분할 메타데이터 파일이 다운로딩된다. 제2 단계(22)에서, 어느 MP4 오프셋이 다운로딩되어야 하는지를 결정하기 위해 분할 메타데이터가 분석되고, 결정된 MP4 오프셋의 다운로딩이 개시된다. 그

후, MP4 세그먼트가 생성되고(23) 카운터/타임스탬프 일치성을 갖는 MPEG-TS 세그먼트로 변환된다(24).

[0121] 재생성 단계(20)는 비디오가 HLS를 이용하여 사용자 디바이스로 전달될 때마다 실행된다. 전형적으로, 디바이스는 세그먼트가 빨리 감기 또는 되감기 동작 이후에 순서를 벗어나도록 요청될 수도 있지만, 세그먼트를 순차적으로, 즉, 세그먼트 1, 세그먼트 2 등을 요청할 것이다.

[0122] 제1 다운로딩 단계(21) 동안, 피어-투-피어 클라이언트는 특정 비디오에 대한 분할 메타데이터 파일의 사본을 획득한다. 하나의 가능성은 파일이 이전에 추적기에 업로딩되었다면, 추적기로부터 그것을 다운로딩하는 것이다.

[0123] MP4 오프셋 다운로딩(22)에 있어서, 피어-투-피어 클라이언트는 각각의 세그먼트를 생성하기 위해 어느 MP4 오프셋들이 다운로딩되어야 하는지를 결정하기 위하여 메타데이터를 분석한다. 예를 들어, 상기 주어진 실시예에서, 제3 세그먼트에 대한 오프셋 범위는 {[0,40959], [847872,1658879], [1687552,1691647], [1789952,1810431], [8441856,8444766]}이다. 그 후 클라이언트는 이러한 오프셋을 다운로딩하고, 그것을 로컬 MP4 파일에 입력한다. 어느 피어 지원 시스템에서와 같이, 그러한 다운로딩은 다른 피어 및/또는 CDN으로부터의 전송을 포함할 수 있을 것이다.

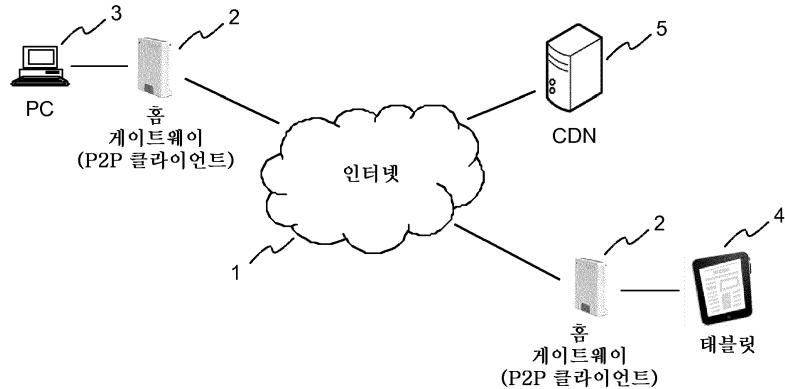
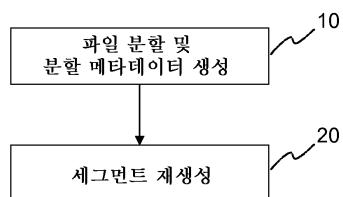
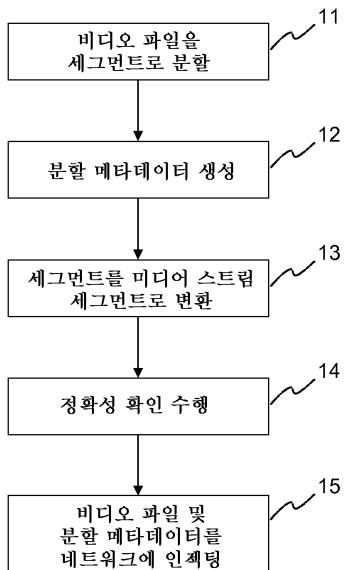
[0124] 일단 단계(22)가 종료되면, 로컬 MP4 파일은 특정 세그먼트를 생성하기 위해 분할기 도구에 의해 판독될 모든 오프셋들에서의 오리지널 MP4 데이터를 포함한다. 다른 오프셋들은 0, 가비지(garbage)를 포함할 수 있고, 또는 다른 세그먼트를 위해 그러한 오프셋들을 이전에 불러왔다면 오리지널 MP4 데이터를 포함할 수 있다.

[0125] 세그먼트 생성 단계(23) 동안, 분할기 도구는 특정된 세그먼트를 생성하기 위해 호출된다. 전술한 차선책이 활용된다면, 그 후 MP4Box 옵션들은 메타데이터 파일에 포함된 시작 시간을 이용하여 조정되어야 한다.

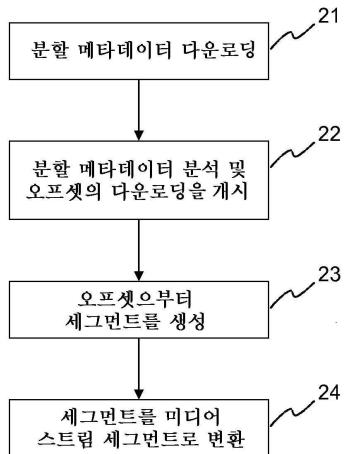
[0126] 최종적으로, 변환 단계(24) 동안, MP4 세그먼트는 MPEG-TS 세그먼트로 변환되고, 불연속성 및 부정확한 타임스탬프 값을 피하기 위하여 정확성이 확인된다. 본 단계는 변환 단계(13)와 유사하고, 유일한 차이는 이전 비디오 세그먼트가 로컬에 이용 가능하지 않을 수 있기 때문에 분할 메타데이터 파일에 포함된 정보를 이용하여 연속성 카운터 값이 조정될 수 있다는 것이다.

[0127] 피어 지원 비디오 전달 시스템을 통해 비디오 파일을 분배하기 위한 디바이스(30)는 도 5에서 도시적으로 예시된다. 디바이스(30)는 비디오 파일을 세그먼트로, 바람직하게는 고정 기간 세그먼트로 분할하고 분할 메타데이터를 생성하기 위한 분할기(31)를 포함한다. 변환기(32)는 이러한 세그먼트들을 카운터/타임스탬프 일치성을 갖는 MPEG-TS 세그먼트로 변환한다. 선택적인 정확성 확인기(33)는 MPEG-TS의 정확성 확인을 수행하기 위해 제공된다. 최종적으로, 인젝터(34)는 비디오 파일 및 분할 메타데이터를 CDN 및 P2P 네트워크에 인젝팅한다.

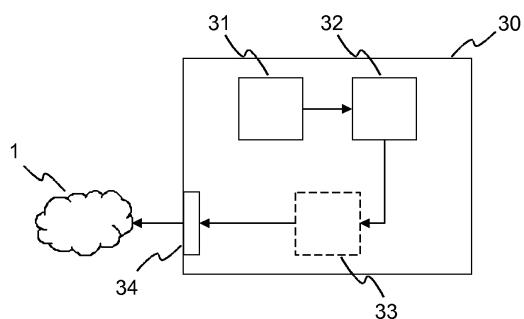
[0128] 피어 지원 비디오 전달 시스템으로부터 비디오 파일을 수신하기 위한 디바이스(40)는 도 6에서 도시적으로 예시된다. 디바이스(40)는 분할 메타데이터 파일을 다운로딩 하기 위한 네트워크 연결(41)을 포함한다. 분석기(42)는 어느 비디오 오프셋들이 다운로딩되어야 하는지를 결정하기 위하여 메타데이터를 분석하고, 결정된 비디오 오프셋의 다운로딩을 개시한다. 세그먼트 생성기(43)는 비디오 세그먼트를 생성하고, 이것은 그 후 변환기(44)에 의해 카운터/타임스탬프 일치성을 갖는 MPEG-TS 세그먼트로 변환된다.

도면**도면1****도면2****도면3**

도면4



도면5



도면6

