



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2016년04월27일
 (11) 등록번호 10-1616152
 (24) 등록일자 2016년04월21일

- (51) 국제특허분류(Int. Cl.)
 H04N 21/643 (2011.01) H04L 29/06 (2006.01)
- (21) 출원번호 10-2011-7021108
- (22) 출원일자(국제) 2010년03월09일
 심사청구일자 2015년02월10일
- (85) 번역문제출일자 2011년09월09일
- (65) 공개번호 10-2011-0139214
- (43) 공개일자 2011년12월28일
- (86) 국제출원번호 PCT/US2010/026710
- (87) 국제공개번호 WO 2010/107627
 국제공개일자 2010년09월23일
- (30) 우선권주장
 12/405,220 2009년03월16일 미국(US)
- (56) 선행기술조사문헌
 JP2008527601 A
 KR1020060083117 A
 KR1020130061170 A
 US20060224760 A1

- (73) 특허권자
 마이크로소프트 테크놀로지 라이선싱, 엘엘씨
 미국 워싱턴주 (우편번호 : 98052) 레드몬드 원
 마이크로소프트 웨이
- (72) 발명자
 보차로브 존 에이
 미국 워싱턴주 98052-6399 레드몬드 원 마이크로
 소프트 웨이 인터내셔널 페이턴츠 마이크로소프트
 코포레이션
 장 지취양 (샘)
 미국 워싱턴주 98052-6399 레드몬드 원 마이크로
 소프트 웨이 인터내셔널 페이턴츠 마이크로소프트
 코포레이션
 (뒷면에 계속)
- (74) 대리인
 제일특허법인

전체 청구항 수 : 총 18 항

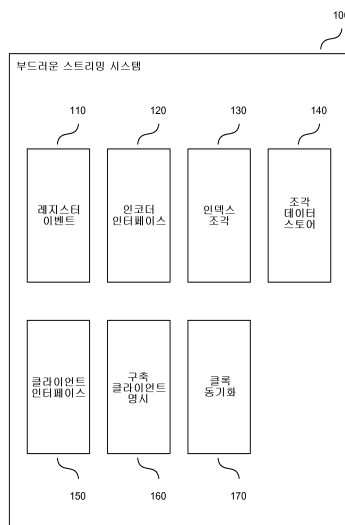
심사관 : 고상호

(54) 발명의 명칭 **컴퓨터 구현 방법 및 컴퓨터 시스템**

(57) 요약

부드러운 스트리밍 시스템은, 서버가 미디어 조각에 증분 제어 정보를 포함시키는 클라이언트와 서버 사이의 상태비보존형 프로토콜을 제공한다. 서버는 기존 인터넷 캐시 인프라스트럭처에 의해 캐시가능한 미디어 조각 요청에 대한 균일한 미디어 조각 응답을 제공한다. 부드러운 스트리밍 시스템은 하나 이상의 인코더로부터 조각으로 미디어 데이터를 수신하고, 각 조각의 인덱스를 생성하며, 조각을 저장한다. 서버는 서버 상에서 이용가능한 인코딩을 설명하는 메타데이터 정보와 조각의 인코딩을 포함하는 조각을 클라이언트에게 제공한다. 서버는, 클라이언트가 데이터를 너무 빠르게 또는 너무 느리게 요청하는지 여부를 클라이언트가 결정할 수 있도록 하는 정보를 각 조각 내에서 제공할 수도 있어서, 클라이언트는 서버가 인코더 데이터를 수신하고 있는 레이트에 맞는 케이션스로 그 요청 레이트를 적응시킬 수 있다.

대표도



(72) 발명자

두가라주 크리쉬나 프라카쉬

미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 인터내셔널 페이턴츠 마이크로소프트 코포레이션

시리바라 수드히르

미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 인터내셔널 페이턴츠 마이크로소프트 코포레이션

리우 린

미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 인터내셔널 페이턴츠 마이크로소프트 코포레이션

로이 아니르반

미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 인터내셔널 페이턴츠 마이크로소프트 코포레이션

가오 지민

미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 인터내셔널 페이턴츠 마이크로소프트 코포레이션

프리랜더 잭 이

미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 인터내셔널 페이턴츠 마이크로소프트 코포레이션

놀튼 크리스토퍼 지

미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 인터내셔널 페이턴츠 마이크로소프트 코포레이션

수드 비살

미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 인터내셔널 페이턴츠 마이크로소프트 코포레이션

명세서

청구범위

청구항 1

서버로부터 클라이언트로 스트리밍 미디어를 제공하기 위한 컴퓨터 구현 방법으로서,

상기 클라이언트로부터 매니페스트(manifest) 요청을 수신하는 단계와,

상기 서버가 수신한 인코더 정보 및 미디어 조각(media fragment)에 기초하여 상기 매니페스트 요청을 만족시키도록 클라이언트 매니페스트를 구축하는 단계 - 상기 클라이언트 매니페스트는 미디어 이벤트와 연관된 하나 이상의 미디어 조각(media segment)을 특정하고, 각각의 미디어 조각은 정의된 포맷을 사용하여 조각을 요청하는 어떠한 클라이언트라도 동일한 응답을 수신하도록 각각의 조각마다 상기 정의된 요청 포맷을 제공함으로써 네트워크 캐시 서버에 의해 개별적으로 캐시 가능하도록 구성되며, 동일한 스트리밍 미디어의 상이한 인코딩을 위한 조각 요청은 미디어 프리젠테이션 내의 동일한 시점(the same point)에 매핑됨 - 와,

상기 매니페스트 요청에 응답하여 구축된 상기 클라이언트 매니페스트를 상기 클라이언트에 제공하는 단계와,

클라이언트로부터 특정 미디어 조각을 식별하는 조각 요청을 수신하는 단계와,

요청된 미디어 조각을 설명하는 매니페스트 정보를 불러옴으로써 증분(incremental) 매니페스트를 구축하는 단계 - 상기 증분 매니페스트는 상기 클라이언트 매니페스트를 구축한 뒤에 상기 서버에 의해 수신된 하나 이상의 미디어 조각에 대한 정보를 더 포함함 - 와,

상기 요청된 미디어 조각과 구축된 상기 증분 매니페스트를 포함하는 상기 클라이언트 조각 요청에 대한 응답을 전송하는 단계

를 포함하되,

상기 단계들은 적어도 하나의 프로세서에 의해 수행되고, 상기 서버는 복수의 클라이언트 각각에 대한 상태 정보를 저장하지 않고 상기 복수의 클라이언트에게 미디어 조각을 제공하는

컴퓨터 구현 방법.

청구항 2

제 1 항에 있어서,

상기 매니페스트 요청은 상기 서버로부터 이용가능한 이벤트 및 상기 이벤트에 관련된 기존의 미디어 조각에 대한 정보의 하나 이상의 인코딩을 결정하고,

이용가능한 인코딩은, 클라이언트가 재생 중 어느 시점에서도 선택할 수 있는 상기 이벤트와 관련된 미디어의 복수의 비트 레이트(bit rate)를 포함하는

컴퓨터 구현 방법.

청구항 3

제 1 항에 있어서,

복수의 인코더가 존재하고,

클라이언트 매니페스트를 구축하는 단계는, 각 인코더로부터의 서버 매니페스트를 파싱하는 단계와 수신된 인코더 매니페스트들을 상기 클라이언트 매니페스트에 병합하는 단계를 포함하는

컴퓨터 구현 방법.

청구항 4

제 1 항에 있어서,

상기 클라이언트 매니페스트는, 상기 클라이언트로 하여금 상기 클라이언트 매니페스트에 기초하여 하나 이상의 구체적인 미디어 조각을 요청함으로써 재생을 시작하고 탐색에 응답할 수 있게 하는 방식으로, 요청 시점까지 복수의 인코더로부터 상기 서버에 의해 수신된 미디어 조각을 설명하는 동적 부분과 이용가능한 인코딩을 설명하는 복수의 인코더 매니페스트의 각각의 통합(union)인 정적(static) 부분을 포함하는

컴퓨터 구현 방법.

청구항 5

제 1 항에 있어서,

상기 클라이언트 매니페스트를 제공하는 단계는 캐시가능한(cacheable) HTTP 응답을 전송하는 단계를 포함하고, 상기 캐시가능한 HTTP 응답은 상기 클라이언트 매니페스트의 서버 제공(serve-provided) 수명에 기초한 캐시 수명을 포함하는

컴퓨터 구현 방법.

청구항 6

제 1 항에 있어서,

상기 클라이언트는 상기 클라이언트 매니페스트에 기초하여 복수의 인코딩 중 하나로 조각 요청을 시작할 수 있고,

상기 클라이언트는 초기에 제 1 저 비트 레이트 인코딩(first low bit rate encoding)을 선택하고 이후에 하나 이상의 검출된 조건에 기초하여 더 높은 비트 레이트로 조각을 요청하는

컴퓨터 구현 방법.

청구항 7

제 1 항에 있어서,

상기 조각 요청은, 상기 조각과 관련되는 시간 및 상기 조각과 관련되는 인코딩을 식별하는 구별되는 URL(Uniform Resource Locator)을 포함함으로써 특정 조각을 식별하는

컴퓨터 구현 방법.

청구항 8

제 1 항에 있어서,

상기 클라이언트 매니페스트 및 상기 증분 매니페스트는 상기 클라이언트로 하여금 완전한 미디어 이벤트에 관한 정보를 포괄하는 로컬 매니페스트를 구축할 수 있게 하고 상기 미디어 이벤트 내에서 어떠한 위치라도 건너 뛰고 재생할 수 있게 하는

컴퓨터 구현 방법.

청구항 9

캐시가능한 스트리밍 미디어 프리젠테이션을 전달하는 컴퓨터 시스템으로서,

소프트웨어 명령어를 실행하도록 구성된 프로세서 및 메모리와,

라이브 미디어 이벤트에 관한 정보를 수신하도록 구성된 레지스터 이벤트 구성요소 - 상기 라이브 미디어 이벤트에 대해 상기 시스템은 인코딩된 미디어 데이터를 수신하게 됨 - 와,

상기 시스템과 상기 인코딩된 미디어 데이터를 미디어 조각으로서 제공하는 하나 이상의 인코더 사이에 인터페이스를 제공하도록 구성된 인코더 인터페이스 구성요소와,

인코더로부터 수신된 미디어 조각의 인덱스 표를 생성하고 유지하도록 구성된 인덱스 조각 구성요소 - 각각의 미디어 조각은 정의된 포맷을 사용하여 조각을 요청하는 어떠한 클라이언트라도 동일한 응답을 수신하도록 각각의 조각마다 상기 정의된 요청 포맷을 제공함으로써 네트워크 캐시 서버에 의해 개별적으로 캐시 가능하도록 구성되고, 동일한 스트리밍 미디어의 상이한 인코딩을 위한 조각 요청은 미디어 프레젠테이션 내에서 동일한 시점에 매핑됨 - 와,

수신된 클라이언트 요청에 기초하여 클라이언트에게 제공하기 위해 수신된 미디어 조각 및 생성된 상기 조각의 인덱스 표를 저장하도록 구성된 조각 데이터 저장소와,

미디어 조각에 대한 클라이언트 요청을 수신하고 클라이언트에게 매니페스트 데이터 및 미디어 조각을 제공하도록 구성된 클라이언트 인터페이스 구성요소와,

상기 클라이언트 요청을 만족시키기 위해, 상기 시스템으로부터 이용가능한 인코딩 각각 및 요청 시점까지 상기 시스템에 의해 저장된 조각에 대한 정보를 포함하는 매니페스트를 구축하도록 구성된 구축 클라이언트 매니페스트 구성요소(build client manifest component) - 상기 구축 클라이언트 매니페스트 구성요소는 각각의 요청된 미디어 조각과 함께 포함시키기 위해 증분 매니페스트를 제공하도록 더 구성되고, 상기 증분 매니페스트는 현재 미디어 조각에 관한 정보 및 클라이언트 매니페스트를 구축한 후에 상기 시스템에 의해 수신된 하나 이상의 미디어 조각에 대한 정보를 클라이언트에게 제공함 - 와,

상기 시스템, 클라이언트 및 인코더의 클록을 동기화하도록 구성된 클록 동기화 구성요소를 포함하는 컴퓨터 시스템.

청구항 10

제 9 항에 있어서,

상기 레지스터 이벤트 구성요소는 인코딩된 미디어 데이터를 상기 시스템에 공급할 복수의 인코더 각각에 대한 식별자를 수신하도록 더 구성되는

컴퓨터 시스템.

청구항 11

제 9 항에 있어서,

상기 인코더 인터페이스 구성요소는 HTTP POST 요청을 통해 푸시된 미디어 데이터를 상기 인코더로부터 수신하는

컴퓨터 시스템.

청구항 12

제 9 항에 있어서,

상기 인코더 인터페이스 구성요소는 미디어 정보 및 미디어 메타데이터를 위한 박스를 포함하는 미디어 컨테이너 내의 인코딩된 미디어 데이터를 수신하는

컴퓨터 시스템.

청구항 13

제 9 항에 있어서,

상기 인덱스 조각 구성요소는 각각의 미디어 조각과 연관된 시간 스탬프에 기초하여 어느 인코더로부터 어떠한 미디어 조각이 수신되었는지를 추적하도록 더 구성되는

컴퓨터 시스템.

청구항 14

제 9 항에 있어서,

상기 클라이언트 인터페이스 구성요소는, 상기 시스템에 접속하는 클라이언트로부터 초기 매니페스트 요청을 수신하고, 상기 인덱스 표에 기초하여 상기 시스템으로부터 이용가능한 인코딩과 현재 시간까지 상기 시스템에 의해 저장된 조각에 대한 정보를 포함하는 클라이언트 매니페스트를 생성하도록 상기 구축 클라이언트 매니페스트 구성요소를 불러오고, 상기 클라이언트 매니페스트를 상기 클라이언트에 제공하도록 더 구성되는

컴퓨터 시스템.

청구항 15

제 9 항에 있어서,

상기 구축 클라이언트 매니페스트 구성요소는, 상기 클라이언트가 각각의 미디어 조각과 함께 상기 클라이언트를 위한 타이밍 힌트로서 예상할 수 있는 후속 조각의 개수를 상기 클라이언트에게 제공하도록 더 구성된

컴퓨터 시스템.

청구항 16

하나 이상의 인코더로부터 인코딩된 미디어 조각을 수신하도록 컴퓨터 시스템을 제어하는 명령어를 포함하는 컴퓨터 판독가능 저장 장치로서,

상기 명령어는 실행되는 경우에 프로세서로 하여금

미디어 이벤트를 설명하는 이벤트 등록(event registration)을 수신하는 단계 - 상기 미디어 이벤트에 관해 상기 시스템은 미디어 데이터를 하나 이상의 인코더로부터 수신하게 됨 - 와,

상기 인코더로부터 이용가능한 상기 미디어 데이터를 설명하는 매니페스트를 각각의 등록된 인코더로부터 요청하는 단계와,

각각의 인코더로부터 인코더 매니페스트를 수신하는 단계 - 상기 인코더 매니페스트는 미디어 이벤트와 연관된 하나 이상의 미디어 조각을 특정함 -

를 수행하게 하고,

복수의 미디어 조각 각각에 대해,

인코더로부터 미디어 조각을 수신하는 단계 - 각각의 미디어 조각은 정의된 포맷을 사용하여 조각을 요청하는 어떠한 클라이언트라도 동일한 응답을 수신하도록 각각의 조각마다 상기 정의된 요청 포맷을 제공함으로써 네트워크 캐시 서버에 의해 개별적으로 캐시 가능하도록 구성되고, 동일한 스트리밍 미디어의 상이한 인코딩을 위한 조각 요청은 미디어 프레젠테이션 내에서 동일한 시점에 매핑됨 - 와,

상기 수신된 미디어 조각을 인덱싱하고, 인덱스 정보를 상기 이용가능한 미디어 조각을 카탈로그하는 인덱스 표에 추가하는 단계와,

상기 조각 및 인덱스 정보를 상기 조각 및 인덱스 정보가 클라이언트 요청을 만족시키도록 이후에 검색될 수 있는 데이터 저장소에 저장함으로써 상기 수신된 미디어 조각을 보관(archiving)하는 단계와,

상기 수신된 조각에 관한 정보를 상기 매니페스트에 추가함으로써 상기 미디어 조각이 일부를 형성하는 미디어 이벤트에 관한 정보를 포함하는 서버 매니페스트를 구축하는 단계

를 수행하게 하는

컴퓨터 판독가능 저장 장치.

청구항 17

제16항에 있어서,

상기 명령어는 상기 프로세서로 하여금, 각각의 수신된 인코더 매니페스트에 관하여, 상기 인코더의 매니페스트를 함께 통합하고 통합된 상기 매니페스트를 이후에 상기 시스템이 제공할 수 있는 미디어 인코딩을 식별하기 위해 클라이언트가 검색할 수 있도록 저장하는 단계를 더 수행하게 하는

컴퓨터 판독가능 저장 장치.

청구항 18

제16항에 있어서,

상기 수신된 미디어 조각은 시간 스탬프 및 상기 미디어 조각을 인코딩한 상기 인코더의 식별자를 포함하고,

상기 시스템은 상기 미디어 조각과 연관된 상기 시간 스탬프를 사용하여 서로 다른 인코더에 의해 병렬로 생성된 미디어 조각을 상관시키는

컴퓨터 판독가능 저장 장치.

발명의 설명

배경 기술

[0001] 스트리밍 미디어는, (서버를 이용하는) 스트리밍 제공자에 의해 멀티미디어가 전달되고 있는 동안 (클라이언트를 이용하는) 최종 사용자에게 의해 계속적으로 수신되고 최종 사용자에게 보통으로 제시되는 멀티미디어이다. RTSP(Real-time Streaming Protocol), RTP(Real-time Transport Protocol), 및 RTCP(Real-time Transport Control Protocol)를 포함하여 스트리밍 미디어를 위한 몇몇의 프로토콜이 존재하며, 이들은 주로 함께 사용된다. IETF(Internet Engineering Task Force)에 의해 개발되고 1998년에 RFC(Request For Comments) 2326으로서 생성된 RTSP(Real Time Streaming Protocol)는 스트리밍 미디어 시스템에서 사용하기 위한 프로토콜이고, 이는 클라이언트가 "재생" 및 "정지"와 같은 VCR 유사 명령을 발함으로써 스트리밍 미디어 서버를 원격으로 제어할 수 있도록 하며, 서버 상에서 파일에 대한 시간-기반 액세스를 허용한다.

[0002] 스트리밍 데이터 자체의 전송은 RTSP 프로토콜의 일부가 아니다. 대부분의 RTSP 서버는 실제 오디오/비디오 데이터에 대한 전송 프로토콜로서 표준-기반 RTP를 사용하여, 다소간 메타데이터 채널로서 동작한다. RTP는 인터넷을 통해 오디오 및 비디오를 전달하기 위한 표준화된 패킷 포맷을 정의한다. RTP는 IETF의 오디오-비디오 전송 작업 그룹(Audio-Video Transport Working Group)에 의해 개발되었으며 1996년에 RFC 1889로 처음 발간되고 2003년에 RFC 3550에 의해 대체되었다. 이 프로토콜은 신택스와 동작에서 HTTP(Hypertext Transport Protocol)와 유사하지만, RTSP는 새로운 요청을 부가한다. HTTP가 상태 비보존형(stateless)인 반면, RTSP는 상태보존형(stateful) 프로토콜이다. 필요한 경우에 세션을 추적하기 위해 세션 ID가 사용된다. 서버가 클라

이언트로 메시지를 전송하는 경우에 일부 예외는 있지만, RTSP 메시지는 클라이언트로부터 서버로 전송된다.

[0003] RTP는 보통 RTCP와 함께 사용된다. RTP가 미디어 스트림(예를 들어, 오디오 및 비디오) 또는 아웃오브밴드(out-of-band) 시그널링(DTMF(dual-tone multi-frequency))을 반송하는 반면, RTCP는 송신 통계 및 QoS(quality of service) 정보를 모니터링하는데 사용된다. RTP는, 소스에서 목적지로 데이터를 반송하는 한 가지 유형의 메시지만을 허용한다. 많은 경우에, 세션에서 다른 메시지를 사용할 필요가 있다. 이들 메시지는 데이터의 흐름과 품질을 제어하고, 수신자가 소스 또는 소스들에 피드백을 보낼 수 있게 한다. RTCP는 이 목적으로 설계된 프로토콜이다. RTCP는 센터 리포트(sender report), 리시버 리포트(receiver report), 소스 설명 메시지(source description message), BYE 메시지(bye message) 및 애플리케이션 특정 메시지(application-specific message)의 다섯가지 유형의 메시지를 갖는다. RTCP는 RTP 흐름을 위한 아웃오브밴드 제어 정보를 제공한다. RTCP는 멀티미디어 데이터의 전달과 패킷징에서 RTP와 함께 동작하지만, 그 자체로 데이터를 전송하지 않는다. 이는 스트리밍 멀티미디어 세션에서 참가자에게 제어 패킷을 전송하기 위해 주기적으로 이용된다. RTCP의 기능 중 하나는 RTP에 의해 제공되고 있는 서비스의 품질에 대한 피드백을 제공하는 것이다. RTCP는, 미디어 접속에 대한 통계와 전송된 바이트, 전송된 패킷, 손실된 패킷, 지터, 피드백 및 라운드 트립 딜레이와 같은 정보를 수집한다. 애플리케이션은 이 정보를, 아마도 흐름을 제한하거나 다른 코덱이나 비트 레이트를 사용함으로써, 서비스 품질을 높이는데 사용할 수 있다.

[0004] 기존 미디어 스트리밍 구조에서의 한 가지 문제점은 서버와 클라이언트 사이의 밀접한 결합(tight coupling)이다. 클라이언트와 서버 사이의 상태보존형 접속은, 서버가 각 클라이언트의 현재 상태를 추적하기 때문에, 추가적인 서버 오버헤드를 생성한다. 이는 서버의 확장성(scalability)을 제한하기도 한다. 또한, 클라이언트는, 먼저 서버와 통신하고 서버가 적응 및 응답하기를 대기하지 않고는, 증가하는 패킷 손실, 감소하는 대역폭, 다른 콘텐츠 또는 기존 콘텐츠 수정(예를 들어, 속도 증가 또는 되감기)에 대한 사용자 요청 등과 같은 변화하는 조건에 빠르게 반응하지 못한다. 종종 클라이언트가 (예를 들어, RTCP를 통해서) 더 낮은 사용 가능한 대역폭을 보고하는 때에, 서버는 충분히 빠르게 적응하지 않아서 사용 가능한 대역폭을 초과하는 패킷은 수신되지 않고 새로운 더 낮은 비트 레이트 패킷은 적시에 서버로부터 전송되지 않으므로 클라이언트로 접속한 사용자가 지각할 미디어의 깨짐(break)을 유발한다. 이 문제를 피하기 위해, 클라이언트는 종종 데이터를 버퍼링하지만, 버퍼링은 레이턴시를 유발하고, 이는 라이브 이벤트에 대해서는 수용할 수 없는 것일 수 있다.

[0005] 또한, 인터넷은 오디오, 비디오, 문서 등을 포함하는 다운로드 가능한 미디어 콘텐츠 아이템의 다양한 유형을 포함한다. 이들 콘텐츠 아이템은 보통, 수백 메가바이트의 비디오와 같이 매우 크다. 사용자는 주로 웹 브라우저를 통해 HTTP를 이용하여 인터넷 상에서 문서를 불러온다. 인터넷은 HTTP를 위한 데이터를 캐시하는데 효과적인 라우터와 프록시의 거대한 인프라스트럭처를 구축하였다. 서버들은 클라이언트들에게 캐시된 데이터를 제공함으로써 원 소스로부터 콘텐츠를 재요청하는 것보다 적은 리소스를 사용하고 지연을 감소시킬 수 있다. 예를 들어, 뉴욕의 사용자는 일본의 호스트로부터 서비스되는 콘텐츠 아이템을 다운로드하고 캘리포니아의 라우터를 통해 콘텐츠 아이템을 수신할 수 있다. 뉴저지의 사용자가 동일한 파일을 요청하는 경우, 캘리포니아의 라우터는 일본의 호스트로부터 데이터를 다시 요청하지 않고 콘텐츠 아이템을 제공할 수 있을 수 있다. 이는 과부하가 걸릴 수 있는 통신선 상의 네트워크 트래픽을 감소시키고, 뉴저지의 사용자가 더 낮은 레이턴시로 콘텐츠 아이템을 수신할 수 있도록 한다.

[0006] 불행히도, 라이브 미디어는 주로 기존의 프로토콜을 이용해서는 캐시될 수 없고, 각 클라이언트는 동일한 서버 또는 동일한 서버의 집합으로부터 미디어를 요청한다. 또한, 스트리밍 미디어가 캐시될 수 있는 경우, 보통 기존의 용이하게 입수할 수 있는 HTTP-기반 인터넷 캐싱 인프라스트럭처가 아니라 특별한 캐시 하드웨어에 의해 이루어진다. 캐싱의 부재는 서버가 동시에 다룰 수 있는 시청자 및 요청의 개수를 제한하고, 라이브 이벤트의 참석을 제한한다. 인터넷을 통해 2008 올림픽의 개막식과 같은 라이브 이벤트를 시청한 사용자의 기록적인 수가 보여주듯이, 세계는 극히 최근의 라이브 정보를 소비하는데 인터넷을 점점 더 많이 사용하고 있다. 현재 기술의 한계는 이 유형의 미디어 콘텐츠를 사용하기 위한 수단으로서 인터넷의 채용을 늦추고 있다.

발명의 내용

과제의 해결 수단

[0007] 서버가 미디어 조각에 증분 제어 정보를 포함시키는 클라이언트와 서버 사이의 상태비보존형 프로토콜을 제공하고, 통상의 제어 채널의 사용을 배제시키는, 부드러운 스트리밍 시스템이 여기서 설명된다. 또한, 서버는 미디

어 조각 요청에 대한 균일한 미디어 조각 응답을 제공하고, 그에 의해 기존 인터넷 캐시 인프라스트럭처가 스트리밍 미디어 데이터를 캐시할 수 있도록 한다. 부드러운 스트리밍 시스템은 하나 이상의 인코더로부터 조각으로 미디어 데이터를 수신하고, 각 조각의 인덱스를 생성하며, 조각을 저장한다. 이벤트가 진행함에 따라, 서버는 이벤트의 종료까지 클라이언트에 의해 요청된 조각을 제공한다. 각 조각은 클라이언트에 의해 재생되기 위한 조각의 미디어 콘텐츠 외에도 서버 상에서 사용가능한 인코딩을 설명하는 메타데이터 정보와 조각의 인코딩을 포함한다. 서버는 다수의 인코딩으로 조각을 제공할 수 있어서, 클라이언트는, 예를 들어, 네트워크 조건에 기초하여 다른 비트 레이트 또는 재생 스피드의 조각으로 빠르게 전환할 수 있다. 서버는, 클라이언트가 데이터를 너무 빠르게 또는 너무 느리게 요청하는지 여부를 클라이언트가 결정할 수 있도록 하는 정보를 각 조각 내에서 제공할 수도 있어서, 클라이언트는 서버가 인코더 데이터를 수신하고 있는 레이트에 맞는 케이던스로 그 요청 레이트를 적응시킬 수 있다. 그러므로, 부드러운 스트리밍 시스템은, 클라이언트 상태를 추적함이 없이, 그리고 클라이언트에 로컬인 캐시 서버로부터 클라이언트가 더 낮은 레이턴시로 미디어를 수신할 가능성을 높이면서, 더 확장가능한 스트리밍 미디어 서버를 제공한다.

[0008] 이 개요는 아래에서 상세한 설명에서 더 자세히 설명할 개념 중 선택된 것을 단순한 형태로 소개하기 위해 제공된다. 이 개요는 청구된 주제의 핵심 특징 또는 필수 특징을 식별하기 위한 것이 아니고, 청구된 주제의 범위를 제한하는데 사용되려는 것도 아니다.

도면의 간단한 설명

- [0009] 도 1은 일 실시형태에서 부드러운 스트리밍 시스템의 구성요소를 도시하는 블록도이다.
- 도 2는 일 실시형태에서 마이크로소프트 윈도우즈 및 마이크로소프트 인터넷 정보 서버(Internet Information Server)(IIS)를 사용하는 부드러운 스트리밍 시스템의 운영 환경을 도시하는 블록도이다.
- 도 3은 일 실시형태에서 인코더로부터 미디어 데이터를 수신하는 시스템의 처리를 도시하는 흐름도이다.
- 도 4는 일 실시형태에서 스트리밍 미디어를 위한 클라이언트 접속을 다루기 위한 시스템의 처리를 도시하는 흐름도이다.
- 도 5는 일 실시형태에서 인코더로부터 원 서버와 클라이언트로의 미디어 조각의 흐름을 도시하는 데이터 흐름도이다.

발명을 실시하기 위한 구체적인 내용

[0010] 서버가 미디어 조각(fragment)(즉, 청크)에 증분(incremental) 정보를 포함하고, 클라이언트와 서버 사이의 상태비보존형 프로토콜을 제공함으로써 통상의 제어 채널의 사용이 배제되는 부드러운 스트리밍 시스템(smooth streaming system)이 여기서 설명된다. 또한, 서버는 미디어 조각 요청에 대한 균일한 미디어 조각 응답을 제공하고(즉, 동일한 조각을 요청하는 클라이언트는 동일한 응답을 얻는다), 그에 의해 기존 인터넷 캐시 인프라스트럭처가 스트리밍 미디어 데이터를 캐시할 수 있도록 한다. 각각의 조각은, 조각이 인터넷 캐시 서버와 클라이언트의 브라우저 캐시 모두에 의해 식별되고 캐시될 수 있도록 하는 구별되는 URL(Uniform Resource Locator)을 갖는다. 캐싱은 서버 상의 부하를 감소시키고 더 많은 클라이언트가 동시에 동일한 콘텐츠를 볼 수 있도록 한다. 부드러운 스트리밍 시스템은 하나 이상의 인코더로부터 미디어 데이터를 조각으로 수신하고, 각 조각의 인덱스를 생성하고, 조각을 저장한다. 이벤트가 진행함에 따라, 서버는 이벤트 종료까지 클라이언트에 의해 요청된 조각을 제공한다. 각 조각은, 클라이언트에 의한 재생(playback)을 위한 조각의 미디어 콘텐츠 외에도, 서버 상에서 사용가능한 인코딩과 조각의 인코딩을 설명하는 메타데이터 정보를 포함한다. 클라이언트가, 예를 들어, 네트워크 상태에 기초하여 다른 비트레이트 또는 재생 스피드의 조각으로 빠르게 전환할 수 있도록 서버는 다수의 인코딩으로 조각을 제공할 수 있다. 서버는 클라이언트가, 클라이언트가 데이터를 너무 빨리 또는 너무 느리게 요청하고 있는지 여부를 결정할 수 있도록 하는 정보를 각 조각 내에서 제공할 수도 있어, 클라이언트가 그 요청 레이트를 서버가 인코더 데이터를 수신하고 있는 레이트에 맞는 케이던스(cadence)에 적응할 수 있다. 그러므로, 부드러운 스트리밍 시스템은 클라이언트 상태를 추적하지 않고도, 클라이언트가 클라이언트에 대해 로컬인 캐시 서버로부터 더 낮은 레이턴시로 미디어를 수신할 가능성을 높이면서 더 확장가능한 스트리밍 미디어 서버를 제공한다.

[0011] 일부 실시형태에서, 부드러운 스트리밍 시스템은 서버와 클라이언트 사이의 구체적인 데이터 송신 형식을 사용

한다. 클라이언트는 서버로부터 미디어의 일부를 포함하는 미디어의 조각을 요청한다. 예를 들어, 10분짜리 파일에 대해, 클라이언트는 2초 조각을 요청할 수 있다. 서버가 클라이언트로 데이터를 밀어내는(push) 통상의 스트리밍과 달리, 이 경우에는 클라이언트가 서버로부터 미디어 조각을 끌어당긴다(pull). 라이브 스트림의 경우, 서버는 클라이언트 요청에 응답하기 위해 즉석에서(on the fly) 미디어를 생성하고 조각을 생성하고 있을 수 있다. 그러므로, 서버가 얼마나 빠르게 조각을 생성하는지 그리고 클라이언트가 얼마나 빠르게 조각을 요청하는지에 있어서, 클라이언트는 서버에 대해 단지 몇 조각만 뒤쳐질 수 있다.

[0012] 각 조각은 메타데이터와 미디어 콘텐츠를 포함한다. 메타데이터는, 미디어 콘텐츠가 더 큰 미디어 요소의 어디에 해당하는지(예를 들어, 이 조각은 10분짜리 비디오 클립에서 1:10 오프셋을 나타낸다), 미디어 콘텐츠가 인코딩된 비트 레이트, 미디어 콘텐츠를 인코딩하는데 사용된 코덱 등과 같이 미디어 콘텐츠에 대한 유용한 정보를 설명할 수 있다. 클라이언트는 조각을 더 큰 미디어 요소의 스토리보드에 위치시키고 미디어 콘텐츠를 적절하게 디코딩하여 재생하는데 이 정보를 사용한다.

[0013] 도 1은 일 실시형태에서 부드러운 스트리밍 시스템의 구성요소를 도시하는 블록도이다. 부드러운 스트리밍 시스템(100)은 레지스터 이벤트 구성요소(110), 인코더 인터페이스 구성요소(120), 인덱스 조각 구성요소(130), 조각 데이터 저장(140), 클라이언트 인터페이스 구성요소(150), 구축 클라이언트 매니페스트 구성요소(160), 클록 동기화 구성요소(170)를 포함한다. 이들 구성요소 각각은 여기서 더 자세히 설명된다.

[0014] 레지스터 이벤트 구성요소(110)는 시스템이 인코딩된 미디어 데이터를 수신할 라이브 또는 다른 미디어 이벤트에 대한 정보를 수신한다. 정보는 서버에 인코딩된 미디어 데이터를 공급할 인코더 각각에 대한 네트워크 주소 정보 또는 기타 식별자를 포함할 수 있다. 정보는 인코더가 인코딩된 미디어 데이터를 공급하고 클라이언트가 미디어 데이터를 액세스할 수 있는 URL도 포함한다.

[0015] 인코더 인터페이스 구성요소(120)는 시스템과 인코딩된 미디어 데이터를 제공하는 하나 이상의 인코더 사이의 인터페이스를 제공한다. 인코더는 공통 네트워크 프로토콜을 이용하여 시스템에 데이터를 밀어낼 수 있다. 예를 들어, 인코더는 시스템에 인코딩된 미디어 데이터를 제공하기 위해 HTTP POST 요청을 이용할 수 있다. 인코더는 인코딩된 미디어 데이터의 소스인 인코더를 특정하는 구별되는 URL을 각각 이용할 수 있으며, 서버가 이를 미디어 이벤트가 레지스터된 때에 레지스터 이벤트 구성요소(110)에 의해 수신된 정보에 매칭시킬 수 있다.

[0016] 인코더 인터페이스 구성요소(120)는 MP4 또는 기타 미디어 컨테이너(예를 들어, MKV)와 같은 수신된 인코딩된 미디어 데이터의 구체적 포맷을 특정할 수 있다. MP4 컨테이너 포맷은 하나의 파일에서 다수 유형의 데이터가 관련될 수 있도록 한다. MP4 컨테이너를 이루는 개별 데이터는 박스라고 불리고, 각 박스는 통상 그 박스에 저장된 데이터의 유형을 식별하는 레이블을 갖는다. 인코더는 인코딩된 미디어 데이터 그 자체 뿐만 아니라, 인코딩된 미디어 데이터를 인코딩하는데 사용된 인코딩 유형과 같은 메타데이터 정보를 박스 내에 위치시킬 수 있다.

[0017] 인덱스 조각 구성요소(130)는 다양한 인코더로부터 수신된 조각의 인덱스 표를 생성하고 유지한다. 시스템(100)은 잠재적인 많은 인코더로부터 이벤트 동안 계속하여(on-going basis) 미디어 조각을 수신하고 있기 때문에, 시스템(100)은 어떤 미디어 조각이 수신되었는지와 어떤 인코더로부터 (또는 어떤 포맷으로) 수신되었는지를 추적하기 위해 인덱스 표를 사용한다. 각 인코더는 미디어 조각을 식별하기 위해 공통의 방법을 사용할 수 있어서(예를 들어, 동기화된 클록을 사용한 시간 스탬프), 인덱스 조각 구성요소(130)는 라이브 이벤트에서 동일한 기간을 나타내는 상이한 인코더들로부터의 조각들을 상관(correlate)시킬 수 있다. 이러한 방식으로, 시스템(100)은 미디어 조각이 손실되는 때를 검출할 수 있고 사용가능한 미디어 조각에 관한 매니페스트 정보(manifest information)를 클라이언트에 제공할 수 있다.

[0018] 조각 데이터 스토어(140)는, 수신된 클라이언트 요청에 기초하여 클라이언트에게 제공하기 위해 생성된 조각의 인덱스 표와 수신된 미디어 조각을 저장한다. 조각 데이터 스토어는 데이터베이스, 디스크 드라이브 또는 데이터 저장소의 다른 형태(예를 들어, SAN(Storage Area Network) 또는 클라우드 기반 저장 서비스)를 포함할 수 있다.

[0019] 클라이언트 인터페이스 구성요소(150)는 미디어 조각에 대한 클라이언트 요청을 수신하고 클라이언트에게 매니페스트 데이터 및 미디어 조각을 제공한다. 클라이언트가 최초에 시스템(100)에 접속하면, 클라이언트는 클라이언트 매니페스트 요청을 전송할 수 있다. 클라이언트 인터페이스 구성요소(150)는 구축 클라이언트 매니페스트 구성요소(160)가 시스템(100)으로부터 사용가능한 인코딩 및 인덱스 표에 기초하여 현재까지 시스템(100)에 의해 저장된 조각에 관한 정보를 포함하는 매니페스트를 생성하도록 한다. 클라이언트는 계속되는 라이브 조각

의 요청을 시작하거나 프리젠테이션의 이전 부분으로 시간적으로 되돌아가는데 이 정보를 사용할 수 있다. 이는, 예를 들어, 클라이언트가 이미 진행 중인 라이브 이벤트에 가입하고 이벤트의 이전 부분을 따라잡고자 하는 경우에 사용될 수 있다.

[0020] 구축 클라이언트 매니페스트 구성요소(160)는 시스템(100)으로부터 사용가능한 인코딩 및 현재까지 시스템(100)에 의해 저장된 조각 조각에 관한 정보를 포함하는 클라이언트 요청을 만족하는 매니페스트(manifest)를 구축한다. 구축 클라이언트 매니페스트 구성요소(160)는 잠재적으로 후속하는 조각뿐만 아니라 현재 미디어 조각에 관하여 정보를 클라이언트에 제공하는 매니페스트를 각 미디어 조각과 함께 포함하도록 제공할 수도 있다. 최초에 수신된 매니페스트와 각 미디어 조각과 함께 제공되는 후속하는 매니페스트를 결합함으로써, 클라이언트는 시작부터 현재까지의 미디어 이벤트에 대한 완전한 정보를 포함하는 최신 매니페스트를 구축할 수 있다. 미디어 이벤트가 완료되면, 클라이언트는, 클라이언트가 요구에 따라 미디어 이벤트를 보는데 사용할 수 있는 미디어 이벤트의 완전한 스토리보드를 갖는다.

[0021] 일부 실시형태에서, 클라이언트 인터페이스 구성요소(150)는, 클라이언트가 미디어 조각이 사용가능하게 된 후에 소정 양의 시간 후에 요청을 하도록 고무하는 방식으로 클라이언트 요청에 응답한다. 예를 들어, 시스템(100)이 인코더로부터 하나 이상의 후속 조각을 수신하는 때까지 시스템(100)은 특정 미디어 조각으로 응답하지 않을 수 있다. 이는 시스템(100)이 현재 조각 응답에 후속 조각에 대한 매니페스트 정보를 포함할 수 있도록 한다. 시스템(100)은 각 미디어 조각과 함께 클라이언트가 기대할 수 있는 후속 조각의 수를 클라이언트에게 제공할 수도 있다. 이는 클라이언트에게 타이밍 힌트가 된다. 클라이언트가 제공된 개수보다 적은 후속 조각에 관한 정보와 함께 미디어 조각을 수신한다면, 클라이언트는 클라이언트가 서버로부터 데이터를 너무 빠르게 요청하고 있다고 간주할 수 있다. 반면, 클라이언트가 제공된 개수보다 많은 후속 조각에 대한 정보와 함께 미디어 조각을 수신한다면, 클라이언트는 클라이언트가 서버로부터 데이터를 너무 천천히 요청하고 있다고 간주할 수 있다. 그러므로, 여하한 특정 조각 요청에 응답하여, 구축 매니페스트 구성요소(160)는 시스템(100)이 그 시점까지 수신한 만큼 많은 후속 조각에 관한 매니페스트 정보를 제공한다.

[0022] 클록 동기화 구성요소(170)는 시스템(100), 클라이언트 및 인코더의 클록을 동기화한다. 절대 시간은 시스템(100)과 관련이 없지만, 다수의 인코더에 걸쳐 특정 조각을 식별할 수 있는 것과 클라이언트에게 조각을 요청하는 레이트(즉, 케이던스)를 제공하는 것은 시스템(100)과 관련이 있다. 예를 들어, 클라이언트가 데이터를 너무 빠르게 요청하는 경우, 서버는 아직 데이터를 갖지 않고 에러 응답(예를 들어, HTTP 404 not found 에러 응답)으로 응답하여, 불필요하게 대역폭을 소비하는 많은 불요(spurious) 요청을 생성할 것이다. 반면, 클라이언트가 데이터를 너무 천천히 요청하는 경우, 클라이언트는 재생을 위한 적시에 데이터를 가지지 않아 사용자에게 재생되는 미디어에서 감지가능한 중단을 생성할 수 있다. 또한, 인코더는 완전히 상이할 수 있는 인코딩으로 미디어 조각을 생성하고, 동일한 시간 기간을 나타내는 상이한 인코딩의 2개의 조각을 상관하는 유의미한 방법을 제공하지 않고 조각들이 미디어 이벤트의 전체 타임라인의 어디에 맞지도 알려주지 않는다. 클록 동기화 구성요소(170)는, 서버, 인코더 및 클라이언트가 특정 시간에 유사한 클록 값을 가질 수 있도록 함으로써 이 정보를 제공한다. 인코더는 각 미디어 조각을 인코더가 조각을 생성한 시간으로 표시할 수도 있다. 이러한 방식으로, 클라이언트가 특정 조각을 요청하면, 클라이언트는 클라이언트가 선택한 인코딩에 무관하게 동일한 기간을 나타내는 조각을 얻게 된다.

[0023] 부드러운 스트리밍 시스템이 구현되는 연산 장치는 중앙 처리 유닛, 메모리, 입력 장치(예를 들어, 키보드 및 포인팅 장치), 출력 장치(예를 들어, 디스플레이 장치), 및 저장 장치(예를 들어 디스크 드라이브 또는 기타 비휘발성 저장 매체)를 포함할 수 있다. 메모리 및 저장 장치는 시스템을 구현 또는 인에이블하는 컴퓨터-실행가능한(computer-executable) 명령(예를 들어, 소프트웨어)으로 인코딩될 수 있는 컴퓨터 관독가능한 저장 매체이다. 또한, 데이터 구조 및 메시지 구조가 저장되거나 통신 링크 상의 신호와 같은 데이터 송신 매체를 통해 송신될 수 있다. 인터넷, LAN(local area network), WAN(wide area network), 점대점 다이얼 업 전송, 휴대전화 네트워크 등 다양한 통신 링크가 사용될 수 있다.

[0024] 시스템의 실시형태는 퍼스널 컴퓨터, 서버 컴퓨터, 핸드헬드 또는 랩탑 장치, 멀티프로세서 시스템, 마이크로프로세서 기반 시스템, 프로그램가능 가전제품(consumer electronics), 디지털 카메라, 네트워크 PC, 미니컴퓨터, 메인프레임 컴퓨터, 위의 시스템 또는 장치 중 여하한 것을 포함하는 분산 컴퓨팅 환경 등을 포함하는 다양한 운영 환경에서 구현될 수 있다. 컴퓨터 시스템은 휴대전화, PDA(personal digital assistant), 스마트폰, 퍼스널 컴퓨터, 프로그램 가능 가전 제품, 디지털 카메라 등일 수 있다.

[0025] 시스템은 프로그램 모듈과 같이 하나 이상의 컴퓨터 또는 다른 장치에 의해 실행되는 컴퓨터 실행가능 명령의

일반적인 맥락에서 설명될 수 있다. 일반적으로, 프로그램 모듈은 특정 태스크를 수행하거나 특정 추상 (abstract) 데이터 유형을 구현하는 루틴, 프로그램, 객체, 컴포넌트, 데이터 구조 등을 포함한다. 통상, 프로그램 모듈의 기능은 다양한 실시형태에서 희망에 따라 결합되거나 분산될 수 있다.

[0026] 상술한 바와 같이, 구축 클라이언트 매니페스트 구성요소는 클라이언트 매니페스트를 생성한다. 다음은 전형적인 클라이언트 매니페스트의 예이다.

```
<?xml version="1.0" encoding="utf-8"?>
<!--Created with Expression Encoder version 2.1.1205.0-->
<SmoothStreamingMedia MajorVersion="1" MinorVersion="0"
  Duration="6537916781" LookAheadFragmentCount="3"
  IsLive="TRUE">
  <StreamIndex Type="video" Subtype="wvc1" Chunks="327"
    Url="QualityLevels({bitrate})/Fragments(video={start
time})">
    <QualityLevel Bitrate="1450000" FourCC="wvc1" width="848"
      Height="480" CodecPrivateData="..." />
    <QualityLevel Bitrate="1050000" FourCC="wvc1" width="592"
      Height="336" CodecPrivateData="..." />
    <c n="0" t="12345678" d="20000000" />
    <c n="1" t="32345678" d="20000000" />
    <c n="2" t="52345678" d="20000000" />
    <c n="3" t="72345678" d="20000000" />
  </StreamIndex>
</SmoothStreamingMedia>
```

[0027]

[0028] 클라이언트 매니페스트는 서버가 그 동안 보관(archive)한 모든 조각에 대한 정보뿐만 아니라 디코딩 정보를 열거한다. 총 미디어 조각 개수 및 길이(duration)는 클라이언트가 요청을 한 때까지 서버가 보관한 미디어 조각에만 대한 것이다 (이는 클라이언트가 빠르게 탐색 바를 구축할 수 있도록 한다). 각각의 미디어 조각에 대해, "t"는 절대 타임스탬프를 의미한다. 클라이언트는 이 값을 조각 URL을 구성하는데 사용한다 (예를 들어, "Fragments(video={start time})"). LookAheadFragmentCount는 "TrackFragmentReferenceBox"가 후술하는 것과 같이 참조할 후속 조각의 목표 개수를 나타낸다. "IsLive"는 라이브 브로드캐스트가 여전히 계속되고 있는지 여부를 나타낸다.

[0029]

일부 실시형태에서, 클라이언트가 특정 미디어 조각을 요청하는 때에, 부드러운 스트리밍 시스템은 후속 미디어 조각에 대한 정보를 제공한다. 예를 들어, 서버는 몇 개의 추가 조각(예를 들어 2 조각)이 사용가능할 때까지 준비된 특정 조각을 유지할 수 있다. 그러면, 서버는 조각을 다음 몇 조각에 대한 매니페스트 정보와 함께 전송할 수 있다. 클라이언트는 이 정보를 어떤 것이 올지를 알고 적절하게 적응하는데 사용할 수 있다. 이는 클라이언트가 요청 레이트를 지능적으로 조정할 수 있게 한다. 예를 들어, 클라이언트가 조각을 요청하고 이후 조각에 대하여 어떤 정보도 가지지 않으면, 클라이언트는 데이터를 너무 빨리 요청하고 있다는 것을 알게 된다. 클라이언트가 조각을 요청하고 너무 많은 이후 조각에 대한 정보를 수신하면, 클라이언트는 정보를 너무 천천히 요청하는 것일 수 있다. 그러므로, 클라이언트는 어드밴스 메타데이터를 힌트로서 사용하여 적응할 수 있다.

[0030]

후속 미디어 조각에 대한 정보는 커스텀 박스(custom box)를 이용하여 MP4 컨테이너에 저장될 수 있다. 예를 들어, 서버는 상기한 'traf' 박스에 다음의 정의로 "Track FragmentReferenceBox"를 삽입할 수 있다.

```
Box Type: 'uuid', {d4807ef2-ca39-4695-8e54-26cb9e46a79f}
Container: 'traf'
Mandatory: Yes
Quantity: Exactly one
```

```
aligned(8) class TrackFragmentReferenceBox
extends Box('uuid', {d4807ef2-ca39-4695-8e54-26cb9e46a79f })
{
    unsigned int(8) version;
    bit(24) flags = 0;
    unsigned int (8) fragment_count;
    for(i=1; i * fragment_count; i++){
        if(version==1){
            unsigned int(64) fragment_absolute_time;
            unsigned int(64) fragment_duration;
        } else {
            unsigned int(32) fragment_absolute_time;
            unsigned int(32) fragment_duration;
        }
    }
}
```

[0031]

[0032]

Fragment_count는 이 박스가 참조하고 있는 동일한 트랙의 직후 조각의 개수를 특정한다. 조각은 MP4 스트림에서 나타나는 것과 동일한 순서로 열거된다. 이 개수는 1 이상이다. Fragment_absolute_time은 후속 조각의 첫 번째 샘플의 절대 타임스탬프를 나타내는 32비트 또는 64비트 정수를 특정한다. Fragment_duration은 후속 조각의 길이를 나타내는 32비트 또는 64비트 정수를 특정한다. ('fragment_count'에서와 같이) "TrackFragmentReferenceBox" 박스 내 후속 조각의 수는 서버에서 구성가능한 설정이다. 서버가 조각 요청을 수신한 때에, 서버가 구성된 값과 같이 "TrackFragmentReferenceBox"를 채우기에 충분한 후속 조각을 갖는다면, 서버는 디폴트 캐시 제어 설정으로 보통의 응답 핸들링 코드 경로를 따를 수 있다.

[0033]

대신 서버가 적어도 하나의, 그러나 "TrackFragmentReferenceBox"를 채우기에 충분하지 않은 후속 조각을 갖는 경우, 서버는 여전히 제한된 후속 조각의 정보로 즉시 조각 응답을 리턴할 수 있다. 서버는 작은 캐시 타임아웃 값을 (조각 길이에 따라) 설정할 수 있고 미래 요청에 대해 꼭 찬 "TrackFragmentReferenceBox"로 응답을 갱신할 것을 기대할 수 있다. 적은 양의 후속 조각 정보는 클라이언트에게, 클라이언트가 데이터를 너무 빠르게 요청하고 있다는 힌트이다. 서버가 이 트랙을 위한 후속 조각을 갖지 않는다면, "조각이 일시적으로 범위를 벗어남(fragment temporarily out of range)"을 나타내는 특정 에러 코드와 함께 요청을 실패시킬 수 있다. 에러 응답은 작은 시간 윈도우에 대해 캐시가능할 수 있다. 클라이언트는 이 에러를 검출하고 작은 지연 후에 동일한 요청을 재시도한다. 한 가지 예외는 라이브 세션이 중단되었고 서버가 마지막 조각을 서비스하려고 하는 경우인데, 이 경우 어떠한 후속 조각 정보도 존재하지 않을 것이고 서버는 최종 스트림 조각으로 요청에 응답한다.

[0034]

도 2는 일 실시형태에서 마이크로소프트 윈도우즈 및 마이크로소프트 인터넷 정보 서버(Internet Information Server)(IIS)를 사용하는 부드러운 스트리밍 시스템의 운영 환경을 도시하는 블록도이다. 환경은 보통 소스 클라이언트(210), 콘텐츠 전달 네트워크(240) 및 외부 네트워크(270)를 포함한다. 소스 클라이언트는 미디어 또는 라이브 이벤트의 소스이다. 소스 클라이언트는 미디어 소스(220) 및 하나 이상의 인코더(230)를 포함한다. 미디어 소스(220)는 각각 다수의 카메라 앵글을 제공하는 카메라, 오디오 포착하는 마이크로폰(microphone capture audio), 슬라이드 프리젠테이션, 텍스트(폐쇄 캡션 서비스로부터 등), 이미지 및 기타 유형의 미디어를 포함할 수 있다. 인코더(230)는 미디어 소스(220)로부터 데이터를 하나 이상의 인코딩 포맷으로 병렬적으로 인코딩한다. 예를 들어, 인코더(230)는 다양한 비트 레이트에서 인코딩된 미디어를 생성할 수 있다.

[0035]

부드러운 스트리밍 시스템이 동작하는 콘텐츠 전달 네트워크(240)는 하나 이상의 인제스트(ingest) 서버(250)와 하나 이상의 근원(origin) 서버(260)를 포함한다. 인제스트 서버(250)는 인코더(230)로부터 각각의 인코딩 포맷으로 인코딩된 미디어를 수신하고 인코딩된 미디어를 설명하는 매니페스트를 생성한다. 인제스트 서버(250)는 여기서 설명하는 미디어 조각을 생성 및 저장할 수 있고, 또는 조각을 요청되는 바에 따라 즉석으로 생성할 수도 있다. 인제스트 서버(250)는 푸시된 데이터를 HTTP POST를 통하거나 인코더(230)로부터 데이터를 요청함으로써 풀링(pull)을 통하는 등으로 인코더(230)로부터 수신할 수 있다. 인코더(230) 및 인제스트 서버(250)는 다양한 중복 구성에서 연결될 수 있다. 예를 들어, 각 인코더는 인코딩된 미디어 데이터를 실패가 발생할 때까

지 인제스트 서버(250) 각각에게 또는 단 하나의 인제스트 서버에게 보낼 수 있다. 근원 서버(260)는 미디어 조각에 대한 클라이언트 요청에 응답하는 서버이다. 근원 서버(260)는 다양한 중복 구성으로 구성될 수도 있다.

[0036] 일부 실시형태에서, 인제스트 서버(250)는 인코더 미디어 스트림을 인제스트하는데 전용인 하나 이상의 서버를 포함한다. 관리자 또는 콘텐츠 저자(author)는, 인제스트 서버(250)의 클라이언트가 특정 미디어 요소(예를 들어, 라이브 이벤트를) 찾을 수 있는 URL을 정의하는 발간 지점을 생성할 수 있다. 예를 들어, IIS를 이용하여, 관리자는 URL "http://ingserver/pubpoint.isml"을 발간할 수 있다. 발간 지점은 인코더(230)에 의해 새로운 미디어 데이터를 인제스트 서버(250)에 제공하는데 사용되고, 근원 서버(260)에 의해 인제스트 서버(250)로부터 미디어 데이터를 요청하는데 사용된다. 각 인코더는 인제스트 서버(250)에 접속하기 위해 구별되는 URL을 이용할 수 있어서 인제스트 서버(250)는 동일한 데이터의 상이한 인코딩을 검출할 수 있다. 예를 들어, 앞서의 예에서의 URL에 기초하여, 인코더는 URL "http://ingserver/pubpoint.isml/Streams(stream1)"을 이용하여 인제스트 서버에 미디어 데이터를 제공하기 위해 HTTP POST를 보낼 수 있다. 인제스트 서버(250)는 인제스트 서버(250)의 클라이언트(예를 들어, 근원 서버(260))에 의한 추후의 복구(retrieval)를 위해 수신된 데이터를 저장한다. POST는 MP4 컨테이너와 같은 다양한 유형의 미디어 포맷을 포함할 수 있다. MP4 컨테이너는 박스라고 불리는 다양한 유형의 정보를 포함하는데, 이는 통상 사용된 인코딩의 유형을 설명하는 "ftyp"와 시퀀스 데이터를 포함하는 "moov" 와 같은 4글자의 코드로 레이블된다. MP4를 이용하던지 기타 컨테이너 포맷을 이용하던지, 인코더는 미디어 요소를 설명하는 매니페스트를 포함하는 "ManifestBox"와 같이 스트림에 대한 박스 또는 정보를 추가할 수 있다.

[0037] 인제스트 서버(250)가 데이터 요청을 수신하면, 인제스트 서버(250)는 이전에 저장된 데이터를 제공한다. 인제스트 서버(250)는, 사용가능한 인코더 스트림을 식별하는 인코더 스트림 매니페스트 요청 및 특정 스트림으로부터의 데이터(스트림 데이터의 일부 포함) 요청을 포함하여 몇몇 유형의 요청을 지원할 수 있다. 요청의 유형은 요청의 URL에 의해 식별될 수 있다. 예를 들어, 인제스트 서버(250)가 URL "http://ingserver/pubpoint.isml/StreamManifest"을 수신하는 경우, 인제스트 서버(250)는 각 사용가능한 인코더에 대한 식별자를 포함하는 인코더 매니페스트를 리턴한다. 인제스트 서버(250)가 URL "http://ingserver/pubpoint.isml/Streams(stream1)"을 수신하는 때에, 인제스트 서버(250)는 식별자 "Encoder1"과 관련된 인코더에 대한 대응 미디어 스트림을 응답으로 전송한다. 응답은 상술한 캐시된 "ftyp," "ManifestBox," 및 "moov" 박스와 같은 MP4 데이터와 이를 따르는 FIFO 버퍼 내 미디어 조각을 포함할 수 있다. 인제스트 서버(250)는 "http://ingserver/pubpoint.isml/Streams(stream1)/StartTime(12345678)" 형태의 부분 데이터 요청을 (예를 들어, 장애극복(failover) 시나리오 동안) 수신할 수도 있는데, 이는 인제스트 서버(250)가 "ftyp," "ManifestBox," 및 "moov" 박스의 전송을 생략하고 특정 타임스탬프에 가장 가까운 미디어 조각으로부터 시작을 시도하도록 한다.

[0038] 근원 서버(260)는 미디어 클라이언트로부터 미디어 스트림 요청을 수신하고 하나 이상의 인제스트 서버(250)로부터 요청된 미디어 스트림을 불러온다. 인제스트 서버(250)와 같이, 관리자 또는 콘텐츠 저자는 근원 서버 상에서 발간 지점을 등록하고 인제스트 서버(250) 및/또는 인코더 URL을 발간 지점과 관련시킨다. 근원 서버(260)는 먼저 (예를 들어, HTTP GET 요청을 이용하여) 사용 가능한 스트림을 설명하는 매니페스트를 인제스트 서버(250)로부터 요청할 수 있다. 그러면 근원 서버는 각각의 인코더 스트림에 대해 인제스트 서버에게 별도의 요청을 제출하고, 인제스트 서버는 인코더로부터 수신된 바와 같이 요청된 미디어 스트림으로 응답한다.

[0039] 근원 서버(260)는, 미디어 스트림에 의해 제공되고 있는 더 큰 미디어 요소의 일부를 나타내는 미디어 조각 및 미디어 스트림에 관한 매니페스트 정보를 별도로 수신할 수 있다. 근원 서버(260)는 근원 서버(260)가 각 인코더로부터의 데이터를 상관시킬 수 있도록 하는 각 인코더에 의해 제공되는 타임스탬프 또는 기타 식별자에 기초하여 각 스트림으로부터 수신된 각 조각의 인덱스를 구축한다. 근원 서버(260)는 수신된 데이터로부터, 미디어 클라이언트 요청에 응답하는, 그들 자신의 MP4 컨테이너나 기타 저장 포맷을 구축할 수 있다. 라이브 이벤트로부터 알려진 포맷의 파일을 구축함으로써, 근원 서버는 이벤트 후에 재빨리 미디어 파일의 통일된 다운로드를 제공하는 것이 가능할 수 있다.

[0040] 근원 서버(260)가 미디어 클라이언트 요청을 수신하는 때에, 근원 서버(260)는 서버가 구축한 인덱스를 인코더 매니페스트(들)로부터 수신된 정적(static) 스트림 정보에 첨부함으로써 클라이언트 매니페스트를 생성한다. 다수의 스트림이 있는 경우, 근원 서버(260)는 스트림 매니페스트를 종합(comprehensive) 클라이언트 매니페스트로 병합한다. 이는 클라이언트가, 근원 서버(260)로부터 추가적인 정보를 획득하지 않고도 클라이언트가 요청하는 인코딩 유형에 있어 선택적일 수 있도록 한다. 서버는 HTTP 응답과 같은 기존의 인터넷 인프라스트럭처

에 의해 캐시될 수 있는 표준 응답 유형을 이용하여 클라이언트에게 매니페스트를 제공한다. 매니페스트 데이터는 시간에 걸쳐 변할 수 있기 때문에, 서버는 매니페스트 응답에 대해 짧은 캐시 타임아웃 값(예를 들어, TTL(time to live))을 설정할 수 있다.

[0041] 외부 네트워크(270)는 에지 서버(280)와 기타 인터넷 (또는 다른 네트워크) 인프라스트럭처 및 클라이언트(290)를 포함한다. 클라이언트가 미디어 조각에 대한 요청을 하면, 클라이언트는 요청을 근원 서버(260)에 어드레싱한다. 네트워크 캐싱의 설계 때문에, 에지 서버(280) 중 하나가 데이터를 포함하면, 그 에지 서버는 요청을 전달하지 않고 클라이언트에 응답할 수 있다. 그러나, 데이터가 에지 서버에서 사용가능하지 않으면, 에지 서버는 요청을 근원 서버(260) 중 하나에게 전달한다. 유사하게, 근원 서버(260) 중 하나가 사용가능하지 않은 데이터의 요청을 수신하면, 근원 서버는 인제스트 서버(250) 중 하나로부터 데이터를 요청할 수 있다.

[0042] 도 3은 일 실시형태에서 인코더로부터 미디어 데이터를 수신하는 시스템의 처리를 도시하는 흐름도이다. 블록 (310)에서 시작하면, 시스템은 시스템이 하나 이상의 인코더로부터 미디어 데이터를 수신할 미디어 이벤트를 설명하는 이벤트 등록(registration)을 수신한다. 예를 들어, 이벤트 등록은 각 인코더의 ID(identification), 미디어 이벤트의 설명, 및 인코더가 인코딩된 미디어 데이터를 제공할 URL을 포함한다. 블록 320에서 계속하면, 시스템은 인입 스트림을 파싱하여, 시스템이 기대할 수 있는 모든 인코더 스트림을 설명하는 스트림 매니페스트와, 자신이 나타나는 스트림에 대해 사용가능한 미디어 데이터를 설명하는 서버 매니페스트를 획득한다. 시스템은 풀링 또는 푸시 모델을 사용하여 동작할 수 있다. 예를 들어, 시스템은 인코더의 구성 정보를 요청하는 HTTP GET 요청을 인코더에게 보낼 수 있고, 또는 시스템은 이 정보를 스트림의 일부로서 단순히 인코더로부터 수신할 수 있다.

[0043] "푸시"의 경우에(예를 들어, 인코더 POST)의 경우에, 두 가지 모두의 매니페스트는 커스텀 박스의 스트림의 시작에 포함되어서, 해야할 요청이 없고, 시스템은 매니페스트를 파싱할 수 있다. "풀링"의 경우(예를 들어, 서버 GET)에, 스트림 매니페스트는 적용불가능하고 (발간 지점 정의가 균등한 정보를 포함한다), 시스템은 커스텀 박스로서 그 정보를 포함한다. 스트림 매니페스트는, 다운스트림 서버 및 클라이언트에 여하한 데이터를 제시하기 전에 인코더로부터 서버가 획득하는 스트림의 세트를 특정하는데 사용된다. 스트림 매니페스트가 없으면, 서버가 인코더 스트림의 전부가 아니라 일부를 획득한 레이스 조건이 존재하고, 다운스트림 서버 또는 클라이언트는 불완전한 그림을 얻는다. 각각의 인입 인코더 스트림은 이 정보를 제공하는 스트림 매니페스트를 포함하기 때문에, 서버 관리자가 어떤 스트림을 기대할지를 특정하지 않는다는 점에서 시스템은 자율적("Self-managing")이다.

[0044] 블록 330에서 계속하면, 시스템은 각 인코더로부터 인코더 매니페스트를 수신한다. 시스템은 인코더의 매니페스트들을 함께 병합하고, 시스템이 제공할 수 있는 미디어 인코딩을 아는 것에 관심이 있는 클라이언트에 의해 추후에 불러오기 위해 병합된 매니페스트를 저장한다. 블록 340에서 계속하면, 시스템은 인코더로부터 미디어 조각을 수신한다. 미디어 조각은 타임스탬프, 미디어 조각을 인코딩한 인코더의 식별자, 및 미디어 조각에 대한 기타 정보를 포함할 수 있다. 시스템은 조각이 어떤 스트림 안에서 오는지 알고 스트림 식별자 외에도 어떤 인코더가 스트림을 생성했는지에 대한 식별 정보를 갖고 있으므로, 인코더 식별자는 통상 사용되지 않는다. 블록 350에서 계속하면, 시스템은 수신된 미디어 조각을 인덱싱하고, 시스템으로부터 사용가능한 미디어 조각을 카탈로그하는 시스템에 의해 유지되는 인덱스 표에 인덱스 정보를 추가한다. 시스템은 상이한 인코더들에 의해 병렬로 생성되는 미디어 조각을 상관시키기 위해 미디어 조각과 관련된 타임스탬프를 사용할 수 있다.

[0045] 블록 360에서 계속하면, 시스템은 조각 및 인덱스 정보를, 클라이언트 요청을 만족시키기 위해 조각 및 인덱스 정보를 이후에 불러올 수 있는 데이터 스토어에 저장함으로써 조각을 보관한다. 블록 370에서 계속하면, 시스템은, 수신된 조각에 대한 정보를 매니페스트에 추가함으로써, 미디어 조각이 그 일부인 미디어 이벤트에 관한 정보를 포함하는 서버 매니페스트를 구축한다. 시스템으로부터 사용가능한 그 때 존재하는 미디어 조각에 관한 클라이언트 정보를 주기 위해 클라이언트가 접속하는 때에, 서버가 이 매니페스트를 클라이언트에게 제공한다. 이벤트가 종료한 때, 서버 매니페스트는, 미디어 이벤트의 온디맨드(on-demand) 보기를 위해 클라이언트에 제공될 수 있는 미디어 이벤트의 완전한 설명을 포함한다. 결정 블록 380에서 계속하면, 시스템이 인코더로부터 더 많은 조각을 기대하면 (예를 들어, 라이브 이벤트가 여전히 진행 중), 시스템은 다음 인코더 조각을 수신하기 위해 블록 340으로 루프(loop)하고, 그렇지 않으면 시스템이 종료한다.

[0046] 도 4는 일 실시형태에서 스트리밍 미디어를 위해 클라이언트 접속을 다루기 위한 시스템의 처리를 도시하는 흐름도이다. 블록 410에서 시작하면, 시스템은 클라이언트로부터 매니페스트 요청을 수신한다. 라이브 이벤트에 대해, 많은 클라이언트가 동시에 접속할 수 있지만, 모두가 이벤트의 시작에서 접속하지는 않는다. 예를 들어,

미디어 조각이 데이터의 2초를 포함하고 클라이언트가 이벤트 시각 1분 후에 접속하면, 시스템으로부터 사용가능한 30 조각이 이미 있을 것이다. 클라이언트는 시스템으로부터 사용가능한 이벤트의 인코딩(시스템에 데이터를 제공하는 인코더에 의해 결정됨)을 결정하기 위한 초기 매니페스트 및 그 때 존재하는 조각에 대한 정보를 요청한다. 서버와 클라이언트 사이의 접속은 상태비보존형임을 유의하라. 서버는 보통 특정 클라이언트에 대해 리소스를 전용시키지 않는다. 차라리, 서버는 여하한 인입 요청을 청취하고 있으며, 각 요청은 특정한 조각 또는 기타 정보를 요청하고, 서버는 요청에 응답하고 여하한 클라이언트의 서버로의 요청의 상태 또는 이력을 구체적으로 추적하지 않고 다음 요청으로 나아간다.

[0047] 블록 420에서 계속하면, 시스템은 수신된 조각 및 시스템이 최초에 인코더 매니페스트를 요청한 때에 수신된 인코더 정보에 기초하여 클라이언트 요청을 만족하기 위해 매니페스트를 구축한다. 클라이언트 매니페스트는 사용가능한 인코딩을 설명하는 각 인코더 매니페스트의 통합인 정적 부분과 그때까지 인코더로부터 서버에 의해 수신한 미디어 조각을 설명하는 동적 부분을 포함한다. 블록 430에서 계속하면, 시스템은 클라이언트 요청에 응답하여 클라이언트에 구축된 클라이언트 매니페스트를 제공한다. 일부 실시형태에서, 요청은 표준 HTTP GET 요청이고 응답은 HTTP 응답(예를 들어, 200 OK)이다. 시스템은 응답 상에서 캐시 수명을 제공할 수 있어서, 합리적인 시간량 내에 후속 클라이언트 요청이 인터넷 캐싱 인프라스트럭처에 의해 서비스될 수 있다. 그러나, 매니페스트의 동적 부분은 재빨리 낡은 것이 되기 때문에, 너무 많이 남아버린 매니페스트 정보를 클라이언트에 남기게 되는 캐싱을 피하기에 충분하게 캐시 수명이 짧다. 매니페스트에 기초하여, 클라이언트는 클라이언트가 선택한 어떤 인코딩으로도 조각 요청을 시작할 수 있다. 예를 들어, 클라이언트는 최초에 낮은 비트 레이트 인코딩을 선택할 수 있고, 네트워크 대역폭이 클라이언트가 조각을 비트 레이트로 수신할 수 있는 능력을 제한할 때까지 후속 조각에 대해 더 높은 비트 레이트 인코딩을 선택할 수 있다.

[0048] 블록 440에서 계속하면, 시스템은 클라이언트로부터 조각 요청을 수신한다. 클라이언트는 구체적인 URL을 이용하여 그 조각을 식별할 수 있다. URL은 인코딩뿐만 아니라 조각의 시간을 식별할 수 있다. 예를 들어, URL은 "http://server/event.isml/QualityLevels(1500000)/Fragments(video=20000000)"의 형태일 수 있는데, 여기서 QualityLevels 파라미터는 초당 비트로 측정된 비트 레이트이고, video는 요청되고 있는 트랙의 이름이고, "video=" 후에 오는 값은 100 나노초 단위(단위의 스케일은 프리젠테이션이 인코딩되는 방식에 의존한다)의 시간 위치이다. 블록 450에서 계속하면, 시스템은 조각 데이터 스토어로부터의 매니페스트 정보와 로컬 인덱스 표를 불러오으로써 요청된 조각을 설명하는 증분(incremental) 매니페스트를 구축한다. 시스템은 여기서 설명된 바와 같이 증분 매니페스트에 하나 이상의 후속 조각에 대한 매니페스트 정보를 포함할 수도 있다.

[0049] 블록 460에서 계속하면, 시스템은 요청된 미디어 조각과 구축된 증분 매니페스트를 포함하는 클라이언트 조각 요청에 대한 응답을 보낸다. 초기 매니페스트 및 각각의 증분 매니페스트에 기초하여, 클라이언트는 전체 미디어 이벤트에 관한 정보를 포괄하는 로컬 매니페스트를 구축할 수 있다. 매니페스트는 클라이언트가 미디어 이벤트 내의 여하한 위치를 빠르게 뛰어넘고(skip around) 재생할 수 있도록 한다. 블록 470에서 계속하면, 시스템은 다음 조각 요청을 기다린다. 결정 블록 480에서 계속하면, 새로운 조각 요청이 수신되면, 시스템은 블록 440으로 루프하여 조각 요청을 처리하고, 그렇지 않으면 시스템은 블록 470으로 루프하여 계속 기다린다. 블록 480 후에, 이들 단계는 종료한다.

[0050] 여기서 설명된 단계에서, 부드러운 스트리밍은 각 클라이언트의 상태를 알지 못하고, 클라이언트에 대해 상태를 추적하지 않음을 유의하라. 사실, 특정 클라이언트에 대해, 클라이언트가 시스템에 이야기하지도 않고 전체 미디어 이벤트를 재생하는 것이 가능하다. 클라이언트가 각각의 요청된 매니페스트 및 미디어 조각을 네트워크에 걸쳐 분산된 캐시 서버로부터 수신할 수 있기 때문에 이것이 가능하다. 클라이언트는, 클라이언트가 관찰한(client-observed) 네트워크 조건에 기초한 희망 비트 레이트, 또는 클라이언트가 표시하는(client-displayed) 제어(예를 들어, 빨리감기, 찾기, 되감기 등)와 사용자의 상호작용에 기초한 희망 위치와 같은 인자에 기초하여 원하는 데이터를 요청한다. 이는 서버가 다른 태스크에 리소스를 집중할 수 있도록 하고 확장성(scalability)을 극적으로 증가시킨다. 많은 사람이 참석하는 라이브 이벤트에 대해, 이는 더 많은 시청자가 이벤트를 볼 수 있다는 것을 의미한다.

[0051] 도 5는 일 실시형태에서 인코더로부터 근원 서버와 클라이언트와의 미디어 조각의 흐름을 도시하는 데이터 흐름도이다. 인코더(505)는 여기서 설명된 바와 같이, 직접 또는 인제스트 서버를 통해 근원 서버(510)로 미디어 데이터(520)를 연속적으로 제공한다. 미디어 데이터는 예를 들어 라이브 이벤트에 기초한 MP4 스트림의 조각을 포함할 수 있다. 근원 서버(510)는 로컬 데이터 스토어 등에 각 미디어 조각을 보관(525)한다. 근원 서버(510)는 클라이언트(515)로부터 매니페스트 요청(530)을 수신한다. 근원 서버(510)는 최근의 미디어 조각 정보에 기초하여 클라이언트 매니페스트를 생성한다(535). 근원 서버(510)는 클라이언트(515)에게 클라이언트 매니

페스트 응답(540)를 제공한다. 그러면 클라이언트(515)는 하나 이상의 미디어 조각 요청(545)을 전송하고, 근원 서버(510)는 요청된 미디어 조각과 잠재적으로는 후속 미디어 조각에 대한 정보로 응답한다(550). 미디어 이벤트가 일어나고 있고 인코더(505)가 새로운 미디어 데이터를 제공하고 있는 이상 도면 왼쪽의 데이터 흐름은 계속된다. 클라이언트(515)가 미디어 조각을 요청하고 있는 이상 도면 오른쪽의 데이터 흐름은 계속되는데, 이는 미디어 이벤트 동안 그리고 클라이언트가 미디어 이벤트의 온디맨드 보기를 요청하므로 이벤트 후에 일어날 수 있다.

[0052] 일부 실시형태에서, 부드러운 스트리밍 시스템은 라이브 미디어 스트림에 대해 DVR(digital video recorder) 같은 기능을 제공한다. 달리 말하면, 사용자는 서버에 대한 작업이나 상태 추적을 추가하지 않고 라이브 스트림을 중지하거나, 라이브 스트림 내에서 탐색하거나 할 수 있다. 라이브 스트림에서, 장면 손실(missed scene), 휴식을 위한 중지, 뒤늦게 이벤트에 참가하여 시작부터 시청하려는 경우 등과 같이, 사용자가 다양한 순서 및 다양한 시간에서 미디어 조각을 재생할 수 있도록 하는 시스템에 의해 인에이블되는 몇몇 시나리오가 있다. 여기서 설명된 어셈블된 매니페스트에 기초하여, 시스템은 사용자가 어떻게 라이브 스트림을 시청하는지에 대한 사용자 제어를 제공한다. 이들 제어는 오늘날 DVR을 통해 TV와 사용가능하다. 부드러운 스트리밍 시스템은 사용자 동작에 응답하고 매니페스트에서 다양한 위치를 탐색하고 적당한 미디어 조각을 요청함으로써 라이브 스트림의 비라이브 모드 재생을 관리하기 위한 클라이언트 제어를 포함한다. 또한, 클라이언트는 재생 동안에 라이브와 비라이브 보기 사이에서 전환할 수 있다.

[0053] 일부 실시형태에서, 부드러운 스트리밍 시스템은 클라이언트에게 웹 브라우저 플러그인을 제공함으로써 동작한다. 예를 들어, 시스템은 클라이언트에게 마이크로소프트 실버라이트(Microsoft Silverlight) 애플리케이션을 제공할 수 있다. 마이크로소프트 실버라이트는 XAP 파일이라고 불리는 컨테이너에 포함된 애플리케이션으로의 참조를 웹 페이지에서 수신한다. 마이크로소프트 실버라이트는 XAP 파일을 추출하고 애플리케이션을 촉발한다. 사용자의 컴퓨터 시스템이 악성 또는 예러 애플리케이션 코드로부터 보호되도록 마이크로소프트 실버라이트는 애플리케이션에 실행되는 샌드박스(sandboxed) 보안 환경을 제공한다. 마이크로소프트 실버라이트는, 사용자의 컴퓨터 시스템 및 하드웨어를 잠재적으로 위협한 애플리케이션 동작으로부터 차단하는 방식으로 미디어를 재생하기 위해 애플리케이션이 호출할 수 있는 API(application programming interfaces)를 제공한다. 그러므로, 마이크로소프트 실버라이트 및 기타 브라우저 플러그인은 부드러운 스트리밍 시스템이 동작할 것으로 기대할 수 있는 클라이언트 환경의 모든 기능성을 제공할 수 있다.

[0054] 일부 실시형태에서, 부드러운 스트리밍 시스템은 관련된 미디어 스트림을 동기화하기 위한 로직을 제공한다. 예를 들어, 라이브 시청각 이벤트는 하나 이상의 비디오 스트림(예를 들어, 카메라 앵글)과 하나 이상의 오디오 스트림(예를 들어, 언어)을 포함할 수 있다. 클라이언트가 오디오 및 비디오 미디어 조각을 별개로 다운로드할 때, 클록 동기화와 관련하여 더 설명하는 바와 같이 각각의 미디어 조각과 관련된 시간 정보를 정렬시킴으로써 클라이언트는 오디오 및 비디오 미디어 콘텐츠를 동기를 맞추어(in sync) 재생한다. 시스템은 슬라이드 프리젠테이션 내의 슬라이드, 이미지, 텍스트 등과 같은 다른 유형의 데이터도 동기화할 수 있다.

[0055] 일부 실시형태에서, 부드러운 스트리밍 시스템은 상이한 레이트에서 재생되는 스트림을 클라이언트에게 제공한다. 예를 들어, 서버는 2x, 5x, 0.5x 및 기타 재생 속도를 포함할 수 있다. 클라이언트는, 미디어가 빨리 감기(예를 들어, 2x) 또는 되감기(예를 들어, 0.5x)되고 있는 모습을 사용자에게 제공하기 위해 다른 레이트의 스트림으로 전환할 수 있다. 전환하기 위해, 클라이언트는 단순히 다른 미디어 조각을 예를 들어 다른 URL에서 요청한다. 클라이언트는 수신된 특정 미디어 조각을 계속하여 재생함으로써 미디어 조각을 현재 레이트에서 재생하는 것과 다른 레이트에서 미디어 조각을 재생하는 것 사이에서 부드럽게 전환할 수 있다. 이는 사용자의 요청과 미디어 재생 변화 사이의 작은 레이턴시로 최종 사용자에게 무결절(seamless) 경험을 제공한다. 클라이언트가, 예를 들어, 미디어를 2배 빠르게 재생하기 위해 2배의 데이터를 다운로드 하는 것이 아니라 가속된 레이트로 인코딩된 감소된 크기의 미디어 인코딩을 다운로드하기 때문에, 이는 네트워크 대역폭을 절약하기도 한다.

[0056] 일부 실시형태에서, 부드러운 스트리밍 시스템은 메타데이터에서 하이라이트 마커(highlight marker)를 제공한다. 하이라이트는, 운동 경기에서 선수가 골을 넣은 지점과 같이 미디어 내의 재미있는 부분을 포함할 수 있다. 클라이언트는, 하이라이트 마커와 관련된 미디어의 미디어 조각을 재생함으로써 이벤트가 끝난 후에 하이라이트 릴(reel)을 재생할 수 있다. 클라이언트가 라이브 이벤트를 수신하지 않았으면, 클라이언트는 미디어에 대한 매니페스트를 요청하고 그 후에 하이라이트에 대응하는 미디어 조각만을 요청할 수 있다. 사용자가 하이라이트 전후에 미디어를 더 보기 원하는 경우(예를 들어, 사용자 빨리감기 또는 되감기에 의해 표시됨), 클라이언트는 추가의 미디어 조각을 요청하여 미디어의 요청된 부분을 재생할 수 있다. 그러므로, 시스템은 클라이언트

언트에 대해 매니페스트에서 하이라이트 정보를 제공할 수 있다.

[0057] 일부 실시형태에서, 부드러운 스트리밍 시스템은 인라인 광고를 지원한다. 라이브 이벤트에 대해, 이벤트 시작 시에는 광고 시간(commercial break)이 언제 일어날지 알지 못할 수 있다. 이벤트 코디네이터는 광고 시간이 되면 프로덕션 중에 버튼을 눌러, 시스템이 미디어 스트림 메타데이터에 광고 마커를 삽입하도록 할 수 있다. 클라이언트가 광고 마커를 수신하는 때에, 클라이언트는 이전에 식별된 광고와 관련된 미디어 조각을 요청하고 수신할 수 있다. 예를 들어, 시스템은 초기 매니페스트에서 잠재적 광고의 목록을 제공할 수 있다. 광고는 다른 미디어와 유사한 미디어 조각에서 제공될 수 있고, 라이브 이벤트를 제공하는 동일한 서버에서 저장되지 않을 수 있다. 광고 마커를 만나면, 클라이언트는 메인 스트림의 재생을 중지하고 광고를 불러와서 표시하며 그 후 메인 스트림의 재생을 다시 시작한다.

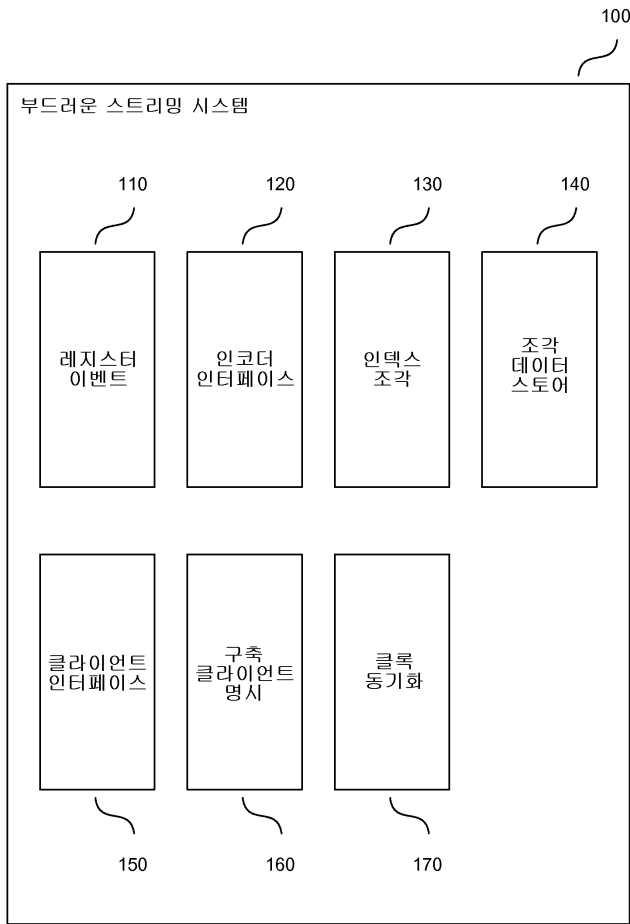
[0058] 일부 실시형태에서, 부드러운 스트리밍 시스템은 가입(subscription) 또는 기타 지불 모델에 기초하여 어떤 인코딩이 사용가능하지를 결정한다. 예를 들어, 콘텐츠 제공자는 라이브 이벤트의 고해상도(HD) 버전에 이벤트의 표준 해상도(SD) 버전보다 더 많이 과금할 수 있다. 이 경우에, 시스템은 지불 모델의 조건이 만족되었는지 여부(예를 들어, 사용자의 계정이 최신인지)에 기초하여 특정 비트 레이트로의 전환을 인에이블 또는 디스에이블 할 수 있다. 이 정보는 클라이언트에 제공되는 매니페스트에 포함될 수 있다. 콘텐츠 제공자는 낮은 비트 레이트 또는 하이라이트만 있는 미디어(highlight only) 등의 일부 인코딩을 무료로 제공하는 반면 다른 것에는 과금할 수 있다.

[0059] 일부 실시형태에서, 부드러운 스트리밍 시스템은 시스템의 다양한 요소에 대해 장애극복을 제공한다. 예를 들어, 시스템은 중복 인코더, 인제스트 서버, 근원 서버 등을 포함할 수 있다. 인코더 장애극복 동안, 서버는 "StartTime(nnnn)"을 인코더 URL에 추가할 수 있는데, 여기서 "nnnn"은 서버가 성공적으로 수신한 마지막 조각의 절대 타임스탬프이다. 장애극복 URL의 예는 "http://encoder:port/StartTime(12345678)"이 된다. MP4 박스를 이용하는 경우, 백업 인코더는 스트림을 시작할 때 "ftyp," "ManifestBox," 및 "moov" 박스를 재전송할 필요가 없다. 인코더 장애극복이 조각 손실을 유발하면, 그들 조각이 클라이언트에 의해 요청되는 경우 서버는 "404 - File Not Found"를 리턴할 것이다.

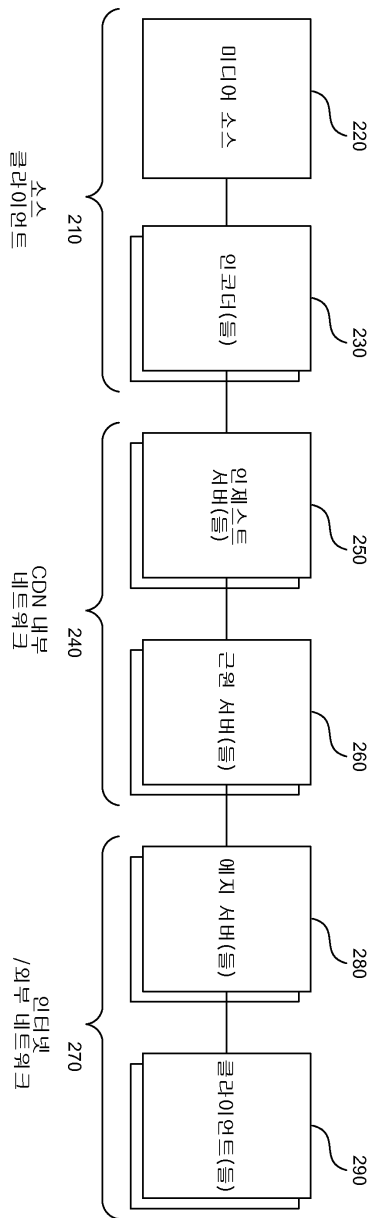
[0060] 상기로부터, 부드러운 스트리밍 시스템의 구체적인 실시형태들이 예시를 위해 여기에서 설명되었으나, 다양한 변형이 본 발명의 사상 및 범위를 벗어남이 없이 이루어질 수 있음을 인식할 것이다. 예를 들어, 시청각 데이터가 예에서 사용되었지만, 텍스트(예를 들어, 스트리밍 스탭 인용(stock quotes)), 슬라이드(예를 들어, 프리젠테이션), 등을 포함하는 다른 데이터 유형이 시스템과 함께 사용될 수 있다. 따라서, 본 발명은 첨부된 청구 범위에 의하지 않고는 제한되지 않는다.

도면

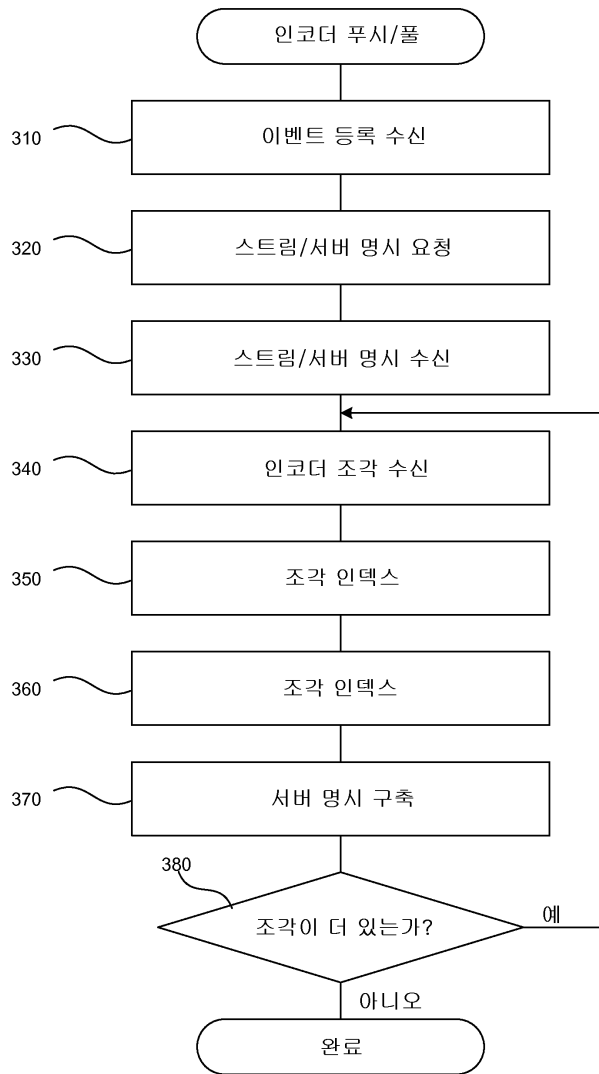
도면1



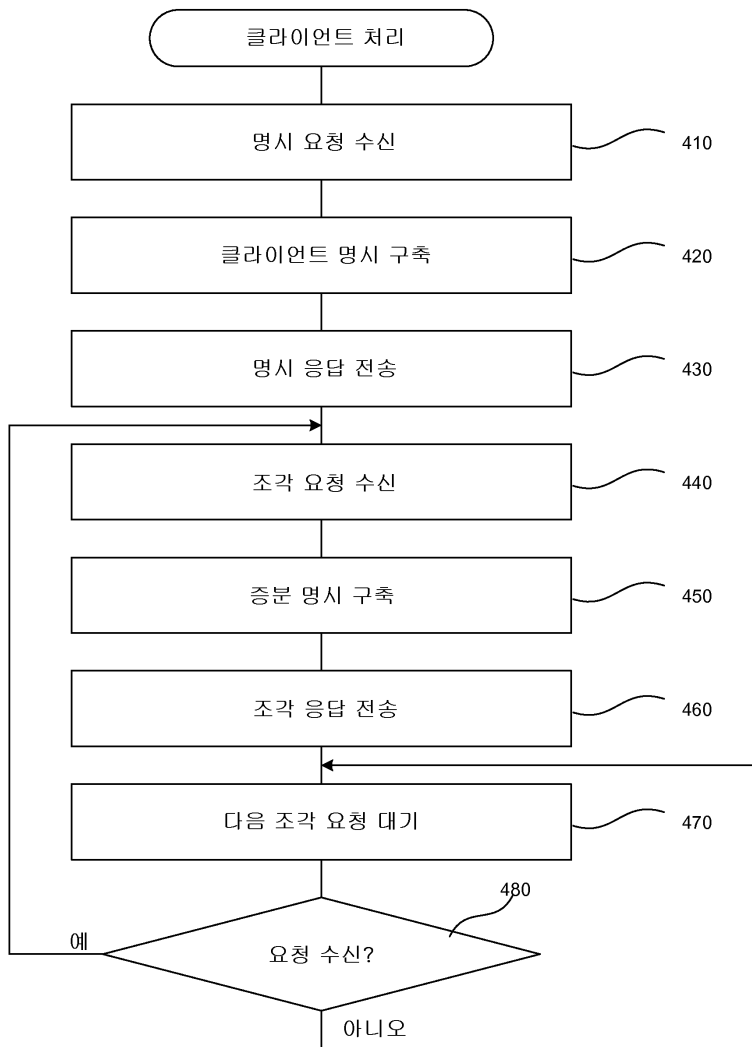
도면2



도면3



도면4



도면5

