(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2016-0135200
(43) 공개일자 2016년11월25일

(51) 국제특허분류(Int. Cl.)
H04L 29/06 (2006.01) *H04L 12/807* (2013.01)
H04L 29/08 (2006.01) *H04L 29/14* (2006.01)
(52) CPC특허분류
H04L 65/4084 (2013.01)
H04L 47/27 (2013.01)
(21) 출원번호 10-2016-7025368
(22) 출원일자(국제) 2015년03월16일
심사청구일자 없음
(85) 번역문제출일자 2016년09월12일
(86) 국제출원번호 PCT/US2015/020791
(87) 국제공개번호 WO 2015/142748
국제공개일자 2015년09월24일
(30) 우선권주장
61/954,936 2014년03월18일 미국(US)
14/289,181 2014년05월28일 미국(US)

(71) 출원인
퀄컴 인코포레이티드
미국 92121-1714 캘리포니아주 샌 디에고 모어하우스 드라이브 5775
(72) 발명자
마오, 이니안
미국 92121-1714 캘리포니아주 샌 디에고 모어하우스 드라이브 5775 퀄컴 인코포레이티드 (내)
우루피나르, 파티흐
미국 92121-1714 캘리포니아주 샌 디에고 모어하우스 드라이브 5775 퀄컴 인코포레이티드 (내)
(뒷면에 계속)
(74) 대리인
특허법인 남앤드남

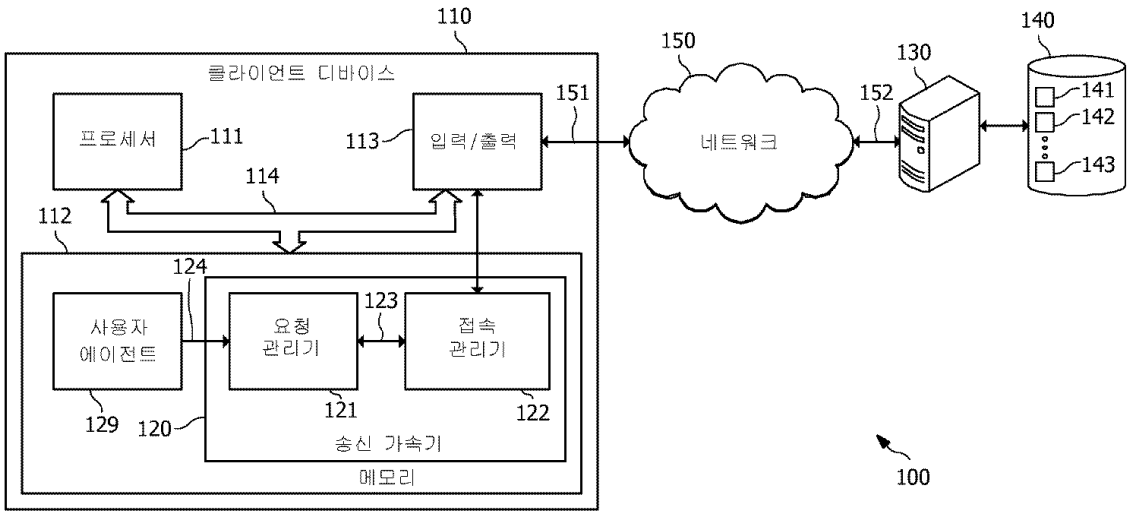
전체 청구항 수 : 총 23 항

(54) 발명의 명칭 **확장된 송신 제어 기능을 구현하는 송신 가속기**

(57) 요 약

클라이언트 디바이스의 UA(user agent)로의 콘텐츠의 전달을 가속화하기 위한 TA(transport accelerator) 시스템들 및 방법들이 본 개시의 실시예들에 따라 제공된다. 실시예들은 CM(connection manager) 및 RM(request manager)을 구현하는 TA 아키텍처를 포함한다. 실시예들의 CM은 콘텐츠 서버로부터 콘텐츠의 청크들을 요청하고, 콘텐츠의 청크들을 요청한 것에 응답하여 데이터를 수신하고 – 수신된 데이터는 콘텐츠의 요청된 청크로부터의 누락 데이터임 – , 누락 데이터에 대한 수신 확인응답(ACK)을 제공한다. 콘텐츠의 청크들 중 요청된 청크로부터의 누락 데이터인 수신된 데이터는 통신 프로토콜 스택을 통해 하나 이상의 콘텐츠 오브젝트들로의 조립을 위한 애플리케이션으로 전달될 수 있다.

대 표 도



(52) CPC특허분류
H04L 65/604 (2013.01)
H04L 65/80 (2013.01)
H04L 67/02 (2013.01)
H04L 67/32 (2013.01)
H04L 69/163 (2013.01)
H04L 69/321 (2013.01)
H04L 69/40 (2013.01)
(72) 발명자
루비, 마이클, 조지
미국 92121-1714 캘리포니아주 샌 디에고 모어하우스 드라이브 5775 퀄컴 인코포레이티드 (내)
마인더, 로렌즈, 크리스토프
미국 92121-1714 캘리포니아주 샌 디에고 모어하우스 드라이브 5775 퀄컴 인코포레이티드 (내)

명 세 서

청구범위

청구항 1

클라이언트 디바이스의 TA(transport accelerator)에 의해, 상기 클라이언트 디바이스의 UA(user agent)로의 콘텐츠의 전달을 가속화하기 위한 방법으로서,

콘텐츠를 제공하도록 동작 가능한 콘텐츠 서버와 상기 UA 사이의 통신 경로에 배치된 상기 TA를 사용하여 상기 UA에 대한 미디어 송신 동작을 개시하는 단계 - 상기 TA는 상기 콘텐츠의 어떤 데이터가 상기 콘텐츠 서버로부터 요청되는지를 제어하도록 동작 가능한 RM(request manager) 및 상기 콘텐츠의 데이터가 상기 콘텐츠 서버로부터 언제 요청되는지를 제어하도록 동작 가능한 CM(connection manager)을 포함하고, 상기 RM은 상기 콘텐츠의 수신된 데이터를 상기 UE로 전달하기 위해 상기 CM에 의해 사용되는 통신 프로토콜 스택과 상기 UA 사이의 통신 경로에 배치됨 - ,

상기 UA에 의해 제공되는 프래그먼트(fragment) 요청들 각각을 상기 콘텐츠의 청크들(chunks)을 요청하기 위한 복수의 청크 요청들로, 상기 RM에 의해, 세분화하는 단계,

상기 콘텐츠의 청크들을 요청하기 위한 상기 복수의 청크 요청들 중 청크 요청들을, 상기 RM에 의해 상기 CM에, 제공하는 단계,

상기 CM과 상기 콘텐츠 서버 사이에 설정된 하나 이상의 접속들을 통해 상기 콘텐츠 서버로부터 상기 콘텐츠의 청크들을, 상기 CM에 의해, 요청하는 단계,

상기 CM에 의해 상기 콘텐츠 서버로부터 요청된 콘텐츠의 하나 이상의 청크들의 수신된 데이터를, 상기 CM에 의해 상기 RM으로, 전달하는 단계 - 상기 수신된 데이터는 상기 콘텐츠의 하나 이상의 청크들 중 하나의 청크로부터의 누락 데이터이고, 콘텐츠 스트림으로의 조립을 위해 상기 CM에 의해 상기 통신 프로토콜 스택을 통해 상기 RM으로 전달됨 - ,

상기 콘텐츠의 하나 이상의 청크들의 데이터를 상기 UA에 의해 요청된 콘텐츠의 프래그먼트로 조립하는 단계,

상기 청크로부터 상기 누락 데이터를 검출할 때, 상기 누락 데이터를 완성하는 콘텐츠의 데이터가 상기 CM으로부터 상기 RM에 의해 수신될 때까지, 상기 콘텐츠의 프래그먼트를 상기 UA에 제공하는 것을, 상기 RM에 의해, 보류하는 단계, 및

콘텐츠 스트림의 부분으로서 상기 UA에 대한 상기 콘텐츠의 프래그먼트를, 상기 RM에 의해 상기 UA로, 전달하는 단계 - 상기 RM에 의해 상기 UA로 전달되는 상기 콘텐츠의 프래그먼트는 상기 누락 데이터를 완성하는 콘텐츠의 데이터를 포함함하고, 상기 RM은 상기 누락 데이터에 관련하여 TA 동작으로부터 상기 UA를 분리하도록 동작함 - 를 포함하는,

클라이언트 디바이스의 TA(transport accelerator)에 의해, 상기 클라이언트 디바이스의 UA(user agent)로의 콘텐츠의 전달을 가속화하기 위한 방법.

청구항 2

제 1 항에 있어서,

상기 콘텐츠의 하나 이상의 청크들 중 요청된 청크로부터의 누락 데이터인 상기 수신된 데이터를, 상기 통신 프로토콜 스택을 통해 상기 RM으로 전달하는 단계는,

상기 데이터가 누락된 경우 더미 데이터를 삽입하는 단계를 포함하는,

클라이언트 디바이스의 TA(transport accelerator)에 의해, 상기 클라이언트 디바이스의 UA(user agent)로의 콘텐츠의 전달을 가속화하기 위한 방법.

청구항 3

제 1 항에 있어서,

상기 수신된 데이터가 상기 하나 이상의 청크들 중 요청된 청크로부터의 누락 데이터인 것을 상기 RM으로 시그널링하는 단계를 더 포함하는,

클라이언트 디바이스의 TA(transport accelerator)에 의해, 상기 클라이언트 디바이스의 UA(user agent)로의 콘텐츠의 전달을 가속화하기 위한 방법.

청구항 4

제 1 항에 있어서,

마치 상기 데이터가 누락되지 않은 것처럼, 적어도 상기 누락 데이터에 대한 수신 확인응답(ACK)을, 상기 CM에 의해 상기 콘텐츠 서버에, 제공하는 단계를 더 포함하는,

클라이언트 디바이스의 TA(transport accelerator)에 의해, 상기 클라이언트 디바이스의 UA(user agent)로의 콘텐츠의 전달을 가속화하기 위한 방법.

청구항 5

제 4 항에 있어서,

상기 적어도 누락 데이터에 대한 ACK를 제공하는 단계는,

적어도 상기 누락 데이터에 대한 ACK를, 상기 CM에 의해 상기 콘텐츠 서버에, 선택적으로 제공하는 단계를 포함하고,

상기 CM은 클라이언트 기반 네트워크 혼잡 제어를 구현하기 위해 상기 누락 데이터에 대해 ACK를 상기 콘텐츠 서버에 제공할지 또는 제공하지 않을지를 선택적으로 결정하도록 적응되는,

클라이언트 디바이스의 TA(transport accelerator)에 의해, 상기 클라이언트 디바이스의 UA(user agent)로의 콘텐츠의 전달을 가속화하기 위한 방법.

청구항 6

제 1 항에 있어서,

상기 누락 데이터를 완성하는 상기 콘텐츠의 데이터를 획득하기 위한 하나 이상의 요청들을, 상기 RM에 의해 상기 CM에, 제공하는 단계를 더 포함하는,

클라이언트 디바이스의 TA(transport accelerator)에 의해, 상기 클라이언트 디바이스의 UA(user agent)로의 콘텐츠의 전달을 가속화하기 위한 방법.

청구항 7

제 6 항에 있어서,

상기 방법은, 상기 누락 데이터를 완성하는 상기 콘텐츠의 데이터를 획득하기 위해 요청할 에러 정정 인코딩 데이터의 양을, 상기 RM에 의해, 결정하는 단계를 더 포함하고,

상기 에러 정정 인코딩 데이터의 소스는 상기 수신된 데이터를 제공하는데 사용되는 상기 콘텐츠 서버 이외의 소스를 포함하는,

클라이언트 디바이스의 TA(transport accelerator)에 의해, 상기 클라이언트 디바이스의 UA(user agent)로의 콘텐츠의 전달을 가속화하기 위한 방법.

청구항 8

제 1 항에 있어서,

상기 콘텐츠 서버는 상기 TA에 의해 제공된 콘텐츠의 가속화된 전달을 지원하기 위한 동작을 위해 미리 설계되지 않은 콘텐츠 서버를 포함하는,

클라이언트 디바이스의 TA(transport accelerator)에 의해, 상기 클라이언트 디바이스의 UA(user agent)로의 콘텐츠의 전달을 가속화하기 위한 방법.

청구항 9

제 8 항에 있어서,

상기 콘텐츠 서버는 표준화된 TCP(Transport Control Protocol) 동작에 따라 콘텐츠에 대한 요청들에 응답하여 데이터를 제공하도록 동작 가능하고,

상기 수신된 데이터는 TCP 데이터로서 수신되고,

상기 통신 프로토콜 스택은 TCP 스택을 포함하는,

클라이언트 디바이스의 TA(transport accelerator)에 의해, 상기 클라이언트 디바이스의 UA(user agent)로의 콘텐츠의 전달을 가속화하기 위한 방법.

청구항 10

클라이언트 디바이스의 TA(transport accelerator)에 의해, 상기 클라이언트 디바이스의 UA(user agent)로의 콘텐츠의 전달을 가속화하도록 구성된 장치로서,

콘텐츠를 제공하도록 동작 가능한 콘텐츠 서버와 상기 UA 사이의 통신 경로에 배치된 상기 TA를 사용하여 상기 UA에 대한 미디어 송신 동작을 개시하기 위한 수단 - 상기 TA는 상기 콘텐츠의 어떤 데이터가 상기 콘텐츠 서버로부터 요청되는지를 제어하도록 동작 가능한 RM(request manager) 및 상기 콘텐츠의 데이터가 상기 콘텐츠 서버로부터 언제 요청되는지를 제어하도록 동작 가능한 CM(connection manager)을 포함하고, 상기 RM은 상기 콘텐츠의 수신된 데이터를 상기 UE로 전달하기 위해 상기 CM에 의해 사용되는 통신 프로토콜 스택과 상기 UA 사이의 통신 경로에 배치됨 - ,

상기 UA에 의해 제공되는 프래그먼트 요청들 각각을 상기 콘텐츠의 청크들을 요청하기 위한 복수의 청크 요청들로, 상기 RM에 의해, 세분화하기 위한 수단,

상기 콘텐츠의 청크들을 요청하기 위한 상기 복수의 청크 요청들 중 청크 요청들을, 상기 RM에 의해 상기 CM에, 제공하기 위한 수단,

상기 CM과 상기 콘텐츠 서버 사이에 설정된 하나 이상의 접속들을 통해 상기 콘텐츠 서버로부터 상기 콘텐츠의 청크들을, 상기 CM에 의해, 요청하기 위한 수단,

상기 CM에 의해 상기 콘텐츠 서버로부터 요청된 콘텐츠의 하나 이상의 청크들의 수신된 데이터를, 상기 CM에 의해 상기 RM으로, 전달하기 위한 수단 - 상기 수신된 데이터는 상기 콘텐츠의 하나 이상의 청크들 중 하나의 청크로부터의 누락 데이터이고, 콘텐츠 스트림으로의 조립을 위해 상기 CM에 의해 상기 통신 프로토콜 스택을 통해 상기 RM으로 전달됨 - ,

상기 콘텐츠의 하나 이상의 청크들의 데이터를 상기 UA에 의해 요청된 콘텐츠의 프래그먼트로 조립하기 위한 수단,

상기 청크로부터 상기 누락 데이터를 검출할 때, 상기 누락 데이터를 완성하는 콘텐츠의 데이터가 상기 CM으로부터 상기 RM에 의해 수신될 때까지, 상기 콘텐츠의 프래그먼트를 상기 UA에 제공하는 것을, 상기 RM에 의해, 보류하기 위한 수단, 및

콘텐츠 스트림의 부분으로서 상기 UA에 대한 상기 콘텐츠의 프래그먼트를, 상기 RM에 의해 상기 UA로, 전달하기 위한 수단 - 상기 RM에 의해 상기 UA로 전달되는 상기 콘텐츠의 프래그먼트는 상기 누락 데이터를 완성하는 콘텐츠의 데이터를 포함함하고, 상기 RM은 상기 누락 데이터에 관련하여 TA 동작으로부터 상기 UA를 분리하도록 동작함 - 을 포함하는,

클라이언트 디바이스의 TA(transport accelerator)에 의해, 상기 클라이언트 디바이스의 UA(user agent)로의 콘텐츠의 전달을 가속화하도록 구성된 장치.

청구항 11

제 10 항에 있어서,

상기 콘텐츠의 하나 이상의 청크들 중 요청된 청크로부터의 누락 데이터인 상기 수신된 데이터를, 상기 통신 프로토콜 스택을 통해 상기 RM으로 전달하기 위한 수단은,

상기 데이터가 누락된 경우 더미 데이터를 삽입하기 위한 수단을 포함하는,

클라이언트 디바이스의 TA(transport accelerator)에 의해, 상기 클라이언트 디바이스의 UA(user agent)로의 콘텐츠의 전달을 가속화하도록 구성된 장치.

청구항 12

제 10 항에 있어서,

상기 수신된 데이터가 상기 하나 이상의 청크들 중 요청된 청크로부터의 누락 데이터인 것을 상기 RM으로 시그널링하기 위한 수단을 더 포함하는,

클라이언트 디바이스의 TA(transport accelerator)에 의해, 상기 클라이언트 디바이스의 UA(user agent)로의 콘텐츠의 전달을 가속화하도록 구성된 장치.

청구항 13

제 10 항에 있어서,

마치 상기 데이터가 누락되지 않은 것처럼, 적어도 상기 누락 데이터에 대한 수신 확인응답(ACK)을, 상기 CM에 의해 상기 콘텐츠 서버에, 제공하기 위한 수단을 더 포함하는,

클라이언트 디바이스의 TA(transport accelerator)에 의해, 상기 클라이언트 디바이스의 UA(user agent)로의 콘텐츠의 전달을 가속화하도록 구성된 장치.

청구항 14

제 13 항에 있어서,

상기 적어도 누락 데이터에 대한 ACK를 제공하기 위한 수단은,

적어도 상기 누락 데이터에 대한 ACK를, 상기 CM에 의해 상기 콘텐츠 서버에, 선택적으로 제공하기 위한 수단을 포함하고,

상기 CM은 클라이언트 기반 네트워크 혼잡 제어를 구현하기 위해 상기 누락 데이터에 대해 ACK를 상기 콘텐츠 서버에 제공할지 또는 제공하지 않을지를 선택적으로 결정하도록 적응되는,

클라이언트 디바이스의 TA(transport accelerator)에 의해, 상기 클라이언트 디바이스의 UA(user agent)로의 콘텐츠의 전달을 가속화하도록 구성된 장치.

청구항 15

제 10 항에 있어서,

상기 누락 데이터를 완성하는 상기 콘텐츠의 데이터를 획득하기 위한 하나 이상의 요청들을, 상기 RM에 의해 상기 CM에, 제공하기 위한 수단을 더 포함하는,

클라이언트 디바이스의 TA(transport accelerator)에 의해, 상기 클라이언트 디바이스의 UA(user agent)로의 콘텐츠의 전달을 가속화하도록 구성된 장치.

청구항 16

제 15 항에 있어서,

상기 장치는, 상기 누락 데이터를 완성하는 상기 콘텐츠의 데이터를 획득하기 위해 요청할 에러 정정 인코딩 데이터의 양을, 상기 RM에 의해, 결정하기 위한 수단을 더 포함하고,

상기 에러 정정 인코딩 데이터의 소스는 상기 수신된 데이터를 제공하는데 사용되는 상기 콘텐츠 서버 이외의 소스를 포함하는,

클라이언트 디바이스의 TA(transport accelerator)에 의해, 상기 클라이언트 디바이스의 UA(user agent)로의 콘텐츠의 전달을 가속화하도록 구성된 장치.

청구항 17

제 10 항에 있어서,

상기 콘텐츠 서버는 상기 TA에 의해 제공된 콘텐츠의 가속화된 전달을 지원하기 위한 동작을 위해 미리 설계되지 않은 콘텐츠 서버를 포함하는,

클라이언트 디바이스의 TA(transport accelerator)에 의해, 상기 클라이언트 디바이스의 UA(user agent)로의 콘텐츠의 전달을 가속화하도록 구성된 장치.

청구항 18

제 17 항에 있어서,

상기 콘텐츠 서버는 표준화된 TCP(Transport Control Protocol) 동작에 따라 콘텐츠에 대한 요청들에 응답하여 데이터를 제공하도록 동작 가능하고,

수신된 데이터는 TCP 데이터로서 수신되고,

상기 통신 프로토콜 스택은 TCP 스택을 포함하는,

클라이언트 디바이스의 TA(transport accelerator)에 의해, 상기 클라이언트 디바이스의 UA(user agent)로의 콘텐츠의 전달을 가속화하도록 구성된 장치.

청구항 19

클라이언트 디바이스의 TA(transport accelerator)에 의해, 상기 클라이언트 디바이스의 UA(user agent)로의 콘텐츠의 전달을 가속화하기 위한 컴퓨터 프로그램 물건으로서,

상기 컴퓨터 프로그램 물건은 프로그램 코드가 기록된 비일시적인 컴퓨터-판독 가능 매체를 포함하고, 상기 프로그램 코드는,

콘텐츠를 제공하도록 동작 가능한 콘텐츠 서버와 상기 UA 사이의 통신 경로에 배치된 상기 TA를 사용하여 상기 UA에 대한 미디어 송신 동작을 개시하기 위한 프로그램 코드 - 상기 TA는 상기 콘텐츠의 어떤 데이터가 상기 콘텐츠 서버로부터 요청되는지를 제어하도록 동작 가능한 RM(request manager) 및 상기 콘텐츠의 데이터가 상기 콘텐츠 서버로부터 언제 요청되는지를 제어하도록 동작 가능한 CM(connection manager)을 포함하고, 상기 RM은 상기 콘텐츠의 수신된 데이터를 상기 UE로 전달하기 위해 상기 CM에 의해 사용되는 통신 프로토콜 스택과 상기 UA 사이의 통신 경로에 배치됨 - ,

상기 UA에 의해 제공되는 프래그먼트 요청들 각각을 상기 콘텐츠의 청크들을 요청하기 위한 복수의 청크 요청들로, 상기 RM에 의해, 세분화하기 위한 프로그램 코드,

상기 콘텐츠의 청크들을 요청하기 위한 상기 복수의 청크 요청들 중 청크 요청들을, 상기 RM에 의해 상기 CM에, 제공하기 위한 프로그램 코드,

상기 CM과 상기 콘텐츠 서버 사이에 설정된 하나 이상의 접속들을 통해 상기 콘텐츠 서버로부터 상기 콘텐츠의 청크들을, 상기 CM에 의해, 요청하기 위한 프로그램 코드,

상기 CM에 의해 상기 콘텐츠 서버로부터 요청된 콘텐츠의 하나 이상의 청크들의 수신된 데이터를, 상기 CM에 의해 상기 RM으로, 전달하기 위한 프로그램 코드 - 상기 수신된 데이터는 상기 콘텐츠의 하나 이상의 청크들 중 하나의 청크로부터의 누락 데이터이고, 콘텐츠 스트림으로의 조립을 위해 상기 CM에 의해 상기 통신 프로토콜 스택을 통해 상기 RM으로 전달됨 - ,

상기 콘텐츠의 하나 이상의 청크들의 데이터를 상기 UA에 의해 요청된 콘텐츠의 프래그먼트로 조립하기 위한 프로그램 코드,

상기 청크로부터 상기 누락 데이터를 검출할 때, 상기 누락 데이터를 완성하는 콘텐츠의 데이터가 상기 CM으로부터 상기 RM에 의해 수신될 때까지, 상기 콘텐츠의 프래그먼트를 상기 UA에 제공하는 것을, 상기 RM에 의해, 보류하기 위한 프로그램 코드, 및

콘텐츠 스트림의 부분으로서 상기 UA에 대한 상기 콘텐츠의 프래그먼트를, 상기 RM에 의해 상기 UA로, 전달하기

위한 프로그램 코드 - 상기 RM에 의해 상기 UA로 전달되는 상기 콘텐츠의 프래그먼트는 상기 누락 데이터를 완성하는 콘텐츠의 데이터를 포함함하고, 상기 RM은 상기 누락 데이터에 관련하여 TA 동작으로부터 상기 UA를 분리하도록 동작함 - 를 포함하는,

클라이언트 디바이스의 TA(transport accelerator)에 의해, 상기 클라이언트 디바이스의 UA(user agent)로의 콘텐츠의 전달을 가속화하기 위한 컴퓨터 프로그램 물건.

청구항 20

제 19 항에 있어서,

상기 수신된 데이터가 상기 하나 이상의 청크들 중 요청된 청크로부터의 누락 데이터인 것을 상기 RM으로 시그널링하기 위한 프로그램 코드를 더 포함하는,

클라이언트 디바이스의 TA(transport accelerator)에 의해, 상기 클라이언트 디바이스의 UA(user agent)로의 콘텐츠의 전달을 가속화하기 위한 컴퓨터 프로그램 물건.

청구항 21

제 19 항에 있어서,

마치 상기 데이터가 누락되지 않은 것처럼, 적어도 상기 누락 데이터에 대한 수신 확인응답(ACK)을, 상기 CM에 의해 상기 콘텐츠 서버에, 제공하기 위한 프로그램 코드를 더 포함하는,

클라이언트 디바이스의 TA(transport accelerator)에 의해, 상기 클라이언트 디바이스의 UA(user agent)로의 콘텐츠의 전달을 가속화하기 위한 컴퓨터 프로그램 물건.

청구항 22

제 21 항에 있어서,

상기 적어도 누락 데이터에 대한 ACK를 제공하기 위한 프로그램 코드는,

적어도 상기 누락 데이터에 대한 ACK를, 상기 CM에 의해 상기 콘텐츠 서버에, 선택적으로 제공하기 위한 프로그램 코드를 포함하고,

상기 CM은 클라이언트 기반 네트워크 혼잡 제어를 구현하기 위해 상기 누락 데이터에 대해 ACK를 상기 콘텐츠 서버에 제공할지 또는 제공하지 않을지를 선택적으로 결정하도록 적응되는,

클라이언트 디바이스의 TA(transport accelerator)에 의해, 상기 클라이언트 디바이스의 UA(user agent)로의 콘텐츠의 전달을 가속화하기 위한 컴퓨터 프로그램 물건.

청구항 23

클라이언트 디바이스의 TA(transport accelerator)에 의해, 상기 클라이언트 디바이스의 UA(user agent)로의 콘텐츠의 전달을 가속화하도록 구성된 장치로서,

적어도 하나의 프로세서, 및

상기 적어도 하나의 프로세서에 커플링된 메모리를 포함하고,

상기 적어도 하나의 프로세서는,

콘텐츠를 제공하도록 동작 가능한 콘텐츠 서버와 상기 UA 사이의 통신 경로에 배치된 상기 TA를 사용하여 상기 UA에 대한 미디어 송신 동작을 개시하고 - 상기 TA는 상기 콘텐츠의 어떤 데이터가 상기 콘텐츠 서버로부터 요청되는지를 제어하도록 동작 가능한 RM(request manager) 및 상기 콘텐츠의 데이터가 상기 콘텐츠 서버로부터 언제 요청되는지를 제어하도록 동작 가능한 CM(connection manager)을 포함하고, 상기 RM은 상기 콘텐츠의 수신된 데이터를 상기 UE로 전달하기 위해 상기 CM에 의해 사용되는 통신 프로토콜 스택과 상기 UA 사이의 통신 경로에 배치됨 - ,

상기 UA에 의해 제공되는 프래그먼트 요청들 각각을 상기 콘텐츠의 청크들을 요청하기 위한 복수의 청크 요청들로, 상기 RM에 의해, 세분화하고,

상기 콘텐츠의 청크들을 요청하기 위한 상기 복수의 청크 요청들 중 청크 요청들을, 상기 RM에 의해 상기 CM에, 제공하고,

상기 CM과 상기 콘텐츠 서버 사이에 설정된 하나 이상의 접속들을 통해 상기 콘텐츠 서버로부터 상기 콘텐츠의 청크들을, 상기 CM에 의해, 요청하고,

상기 CM에 의해 상기 콘텐츠 서버로부터 요청된 콘텐츠의 하나 이상의 청크들의 수신된 데이터를, 상기 CM에 의해 상기 RM으로, 전달하고 - 상기 수신된 데이터는 상기 콘텐츠의 하나 이상의 청크들 중 하나의 청크로부터의 누락 데이터이고, 콘텐츠 스트림으로의 조립을 위해 상기 CM에 의해 상기 통신 프로토콜 스택을 통해 상기 RM으로 전달됨 - ,

상기 콘텐츠의 하나 이상의 청크들의 데이터를 상기 UA에 의해 요청된 콘텐츠의 프래그먼트로 조립하고,

상기 청크로부터 상기 누락 데이터를 검출할 때, 상기 누락 데이터를 완성하는 콘텐츠의 데이터가 상기 CM으로부터 상기 RM에 의해 수신될 때까지, 상기 콘텐츠의 프래그먼트를 상기 UA에 제공하는 것을, 상기 RM에 의해, 보류하고, 그리고

콘텐츠 스트림의 부분으로서 상기 UA에 대한 상기 콘텐츠의 프래그먼트를, 상기 RM에 의해 상기 UA로, 전달하도록 구성되고,

상기 RM에 의해 상기 UA로 전달되는 상기 콘텐츠의 프래그먼트는 상기 누락 데이터를 완성하는 콘텐츠의 데이터를 포함함하고, 상기 RM은 상기 누락 데이터에 관련하여 TA 동작으로부터 상기 UA를 분리하도록 동작하는,

클라이언트 디바이스의 TA(transport accelerator)에 의해, 상기 클라이언트 디바이스의 UA(user agent)로의 콘텐츠의 전달을 가속화하도록 구성된 장치.

발명의 설명

기 술 분 야

[0001] [0001] 본 출원은 2014년 3월 18일에 출원된 "TRANSPORT ACCELERATOR IMPLEMENTING EXTENDED TRANSMISSION CONTROL FUNCTIONALITY"란 명칭의 공동-계류중인 미국 가특허 출원 번호 제 61/954,936 호, 및 2014년 5월 28일에 출원된 "TRANSPORT ACCELERATOR IMPLEMENTING EXTENDED TRANSMISSION CONTROL FUNCTIONALITY"란 명칭의 미국 특허 출원 일련 번호 제 14/289,181 호에 대한 우선권을 주장하고, 이로써 상기 출원의 개시내용들은 인용에 의해 본원에 통합된다. 본 출원은 공동으로 양도된 2014년 5월 28일에 출원된 "TRANSPORT ACCELERATOR IMPLEMENTING EXTENDED TRANSMISSION CONTROL FUNCTIONALITY"란 명칭의 미국 특허 출원 일련 번호[도켓 번호] 제 14/289,016 호, 2014년 5월 28일에 출원된 "TRANSPORT ACCELERATOR IMPLEMENTING ENHANCED SIGNALING"란 명칭의 일련 번호 제 14/289,348 호, 2014년 5월 28일에 출원된 "TRANSPORT ACCELERATOR IMPLEMENTING REQUEST MANAGER AND CONNECTION MANAGER FUNCTIONALITY"란 명칭의 일련 번호 제 14/289,403 호, 2014년 5월 28일에 출원된 "TRANSPORT ACCELERATOR IMPLEMENTING SELECTIVE UTILIZATION OF REDUNDANT ENCODED CONTENT DATA FUNCTIONALITY"란 명칭의 일련 번호 제 14/289,458 호, 2014년 5월 28일에 출원된 "TRANSPORT ACCELERATOR IMPLEMENTING A MULTIPLE INTERFACE ARCHITECTURE"란 명칭의 일련 번호 제 14/289,476 호, 및 2014년 5월 28일에 출원된 "TRANSPORT ACCELERATOR IMPLEMENTING CLIENT SIDE TRANSMISSION FUNCTIONALITY"란 명칭의 일련 번호 제 14/289,499 호에 관련되고, 상기 출원들 각각은 여기서 동시에 출원되고, 상기 출원의 개시내용들은 전체적으로 인용에 의해 본원에 명백히 통합된다.

배 경 기 술

[0002] [0002] 점점 더 많은 콘텐츠가 이용가능한 통신 네트워크들을 통해 전달되고 있다. 흔히, 이러한 콘텐츠는, 예를 들어, 오디오 데이터, 비디오 데이터, 이미지 데이터 등을 포함하는, 다양한 타입들의 데이터를 포함한다. 비디오 콘텐츠, 특히 고해상도 비디오 콘텐츠는 흔히 상대적으로 큰 데이터 파일 또는 데이터의 다른 콜렉션을 포함한다. 따라서, 이러한 콘텐츠를 소비하고 있는 최종 사용자 디바이스 상의 사용자 에이전트(UA) 또는 다른 클라이언트 디바이스는 흔히 원하는 비디오 콘텐츠를 포함하는 콘텐츠의 프래그먼트들의 시퀀스를 요청 및 수신한다. 예를 들어, UA는, 데이터, 흔히 멀티미디어 데이터를 요청하고, 추가의 프로세싱 및 가능하게는 사용자 디바이스 상의 디스플레이를 위해 요청된 데이터를 수신하는 사용자 디바이스 상에서 실행하는 프로세스 또는 클라이언트 애플리케이션을 포함할 수 있다.

[0003] [0003] 많은 타입들의 애플리케이션들이 오늘날 전술한 콘텐츠 전달을 위해 HTTP에 의존한다. 많은 이러한 애플리케이션들에서, HTTP 전송의 성능은 애플리케이션에 대한 사용자의 경험에 중요하다. 예를 들어, 라이브 스트리밍은 비디오 스트리밍 클라이언트의 성능을 저해할 수 있는 몇몇 제한들을 갖는다. 두 제한들이 특히 두드러진다. 첫 번째, 미디어 세그먼트들은 시간이 지남에 따라 차례로 이용가능하게 된다. 이러한 제한은 클라이언트가 데이터의 큰 부분을 연속적으로 다운로딩하는 것을 못하게 하며, 이는 차례로 다운로드 레이트 추정의 정확성에 영향을 미친다. 대부분의 스트리밍 클라이언트들이 "요청-다운로드-추정" 루프로 동작하기 때문에, 다운로드 추정이 부정확한 경우 스트리밍은 일반적으로 잘 잘 동작하지 않는다. 두 번째, 라이브 이벤트 스트리밍을 뷰잉 시, 사용자는 일반적으로 실제 라이브 이벤트 타임라인으로부터 긴 지연을 겪기를 원치 않는다. 이러한 거동은 스트리밍 클라이언트가 큰 버퍼를 구축하는 것을 못하게 하며, 이는 차례로 더 많은 재버퍼링을 유발할 수 있다.

[0004] [0004] 스트리밍 클라이언트가 (예를 들면, 대부분의 DASH(Dynamic Adaptive Streaming over HTTP) 클라이언트들이 하는 바와 같이) TCP(Transmission Control Protocol)를 통해 동작하면, 앞서 말한 엄격한 라이브 이벤트 타임라인은, 누락 또는 재정렬된 패킷이 존재할 때 둔화되는 전형적인 TCP 거동과 모순된다. 빌트-인 TCP 혼잡 제어 메커니즘은 라이브 스트리밍 동안에 재버퍼링 효과를 악화시키고, 반면에 라이브 이벤트에 대한 시청자들은 기꺼이 재버퍼링을 스킵하고 최종 이벤트 타임라인으로 점프할 가능성이 더 많다.

[0005] [0005] 동일한 이슈들이 또한 HTTP-기반 파일 다운로드에서 존재하고, 여기서 다운로드의 완료에 대한 데드라인이 존재하고, 그렇지 않다면 페널티가 발생된다. 예를 들면, 사용자가 웹 페이지, 사진을 액세스하거나 웹-기반 애플리케이션을 사용하려고 시도하면, 큰 다운로드 레이턴시는 사용자가 웹 페이지 또는 웹-기반 애플리케이션을 외면하게 할 수 있다.

[0006] [0006] 온-디맨드 스트리밍은 또한 유사한 제약들로 고통받는다. 예를 들면, 온-디맨드 스트리밍에서, 클라이언트 디바이스들은 재생을 사용자에게 제공하기 위한 올바른 순서로 가능한 한 빠르게 온-디맨드 스트림을 수신하고자 한다. 온-디맨드 콘텐츠를 스트리밍하는 것의 성능은 누락 및 재정렬된 패킷들, 재버퍼링 등에 의해 영향을 받는다.

발명의 내용

[0007] [0007] 클라이언트 디바이스의 TA(transport accelerator)에 의해, 클라이언트 디바이스의 UA(user agent)로의 콘텐츠의 전달을 가속화하기 위한 방법이 본 개시의 실시예들에 따라 제공된다. 실시예들에 따른 방법은 TA의 CM(connection manager)에 의해, 콘텐츠 서버로부터 콘텐츠의 하나 이상의 청크들(chunks)을 요청하는 단계, CM에 의해, 콘텐츠의 하나 이상의 청크들을 요청한 것에 응답하여 전송된 데이터를 수신하는 단계 – 수신된 데이터는 콘텐츠의 하나 이상의 청크들 중 요청된 청크로부터의 누락 데이터임 – , 및 CM에 의해 콘텐츠 서버에, 적어도 누락 데이터에 대한 수신 확인응답(ACK)을 제공하는 단계를 포함한다. 실시예들의 방법은 콘텐츠의 하나 이상의 청크들 중 요청된 청크로부터의 누락 데이터인 수신된 데이터를, 통신 프로토콜 스택을 통해 하나 이상의 콘텐츠 오브젝트들로의 조립을 위한 애플리케이션으로 전달하는 단계를 더 포함한다.

[0008] [0008] 클라이언트 디바이스의 TA(transport accelerator)에 의해, 클라이언트 디바이스의 UA(user agent)로의 콘텐츠의 전달을 가속화하기 위한 장치가 본 개시의 실시예들에 따라 제공된다. 실시예들에 따른 장치는 TA의 CM(connection manager)에 의해, 콘텐츠 서버로부터 콘텐츠의 하나 이상의 청크들을 요청하기 위한 수단, CM에 의해, 콘텐츠의 하나 이상의 청크들을 요청한 것에 응답하여 전송된 데이터를 수신하기 위한 수단 – 수신된 데이터는 콘텐츠의 하나 이상의 청크들 중 요청된 청크로부터의 누락 데이터임 – , 및 CM에 의해 콘텐츠 서버에, 적어도 누락 데이터에 대한 수신 확인응답(ACK)을 제공하기 위한 수단을 포함한다. 실시예들에 따른 장치는 콘텐츠의 하나 이상의 청크들 중 요청된 청크로부터의 누락 데이터인 수신된 데이터를, 통신 프로토콜 스택을 통해 하나 이상의 콘텐츠 오브젝트들로의 조립을 위한 애플리케이션으로 전달하기 위한 수단을 더 포함한다.

[0009] [0009] 클라이언트 디바이스의 TA(transport accelerator)에 의해, 클라이언트 디바이스의 UA(user agent)로의 콘텐츠의 전달을 가속화하기 위한 컴퓨터 프로그램 물건이 본 개시의 실시예들에 따라 제공된다. 실시예들에 따른 컴퓨터 프로그램 물건은 프로그램 코드가 기록된 비일시적인 컴퓨터-판독 가능 매체를 포함한다. 실시예들의 프로그램 코드는, TA의 CM(connection manager)에 의해, 콘텐츠 서버로부터 콘텐츠의 하나 이상의 청크들을 요청하기 위한 코드, CM에 의해, 콘텐츠의 하나 이상의 청크들을 요청한 것에 응답하여 전송된 데이터를 수신하기 위한 코드 – 수신된 데이터는 콘텐츠의 하나 이상의 청크들 중 요청된 청크로부터의 누락 데이터임 – , 및 CM에 의해 콘텐츠 서버에, 적어도 누락 데이터에 대한 수신 확인응답(ACK)을 제공하기 위한 코드를 포함한

다. 실시예들의 프로그램 코드는 콘텐츠의 하나 이상의 청크들 중 요청된 청크로부터의 누락 데이터인 수신된 데이터를, 통신 프로토콜 스택을 통해 하나 이상의 콘텐츠 오브젝트들로의 조립을 위한 애플리케이션으로 전달하기 위한 코드를 더 포함한다.

[0010] [0010] 클라이언트 디바이스의 TA(transport accelerator)에 의해, 클라이언트 디바이스의 UA(user agent)로의 콘텐츠의 전달을 가속화하도록 구성된 장치가 본 개시의 실시예들에 따라 제공된다. 실시예들의 장치는 적어도 하나의 프로세서 및 적어도 하나의 프로세서에 커플링된 메모리를 포함한다. 적어도 하나의 프로세서는 실시예들에 따라, TA의 CM(connection manager)에 의해 콘텐츠 서버로부터, 콘텐츠의 하나 이상의 청크들을 요청하고, CM에 의해, 콘텐츠의 하나 이상의 청크들을 요청한 것에 응답하여 전송된 데이터를 수신하고 - 수신된 데이터는 콘텐츠의 하나 이상의 청크들 중 요청된 청크로부터의 누락 데이터임 - , 그리고 CM에 의해 콘텐츠 서버에, 적어도 누락 데이터에 대한 수신 확인응답(ACK)을 제공하도록 구성된다. 적어도 하나의 프로세서는 실시예들에 따라, 콘텐츠의 하나 이상의 청크들 중 요청된 청크로부터의 누락 데이터인 수신된 데이터를, 통신 프로토콜 스택을 통해 하나 이상의 콘텐츠 오브젝트들로의 조립을 위한 애플리케이션으로 전달하도록 추가로 구성된다.

[0011] [0011] 본 개시의 추가의 실시예들은 클라이언트 디바이스의 TA(transport accelerator)에 의해, 클라이언트 디바이스의 UA(user agent)로의 콘텐츠의 전달을 가속화하기 위한 방법을 제공한다. 실시예들의 방법은 콘텐츠를 제공하도록 동작 가능한 콘텐츠 서버와 UA 사이의 통신 경로에 배치된 TA를 사용하여 UA에 대한 미디어 송신 동작을 개시하는 단계를 포함하고, TA는 콘텐츠의 어떤 데이터가 콘텐츠 서버로부터 요청되는지를 제어하도록 동작 가능한 RM(request manager) 및 콘텐츠의 데이터가 콘텐츠 서버로부터 언제 요청되는지를 제어하도록 동작 가능한 CM(connection manager)을 포함하고, RM은 콘텐츠의 수신된 데이터를 UE로 전달하기 위해 CM에 의해 사용되는 통신 프로토콜 스택과 UA 사이의 통신 경로에 배치된다. 상기 방법은 CM에 의해 콘텐츠 서버로부터 요청된 콘텐츠의 하나 이상의 청크들의 수신된 데이터를, CM에 의해 RM으로, 전달하는 단계 - 수신된 데이터는 콘텐츠의 하나 이상의 청크들 중 하나의 청크로부터의 누락 데이터이고, 콘텐츠 스트림으로의 조립을 위해 CM에 의해 통신 프로토콜 스택을 통해 RM으로 전달되고, RM은 누락 데이터에 관련하여 TA 동작으로부터 UA를 분리하도록 동작함 - , 콘텐츠의 하나 이상의 청크들의 데이터를 UA에 의해 요청된 콘텐츠의 프래그먼트로 조립하는 단계, 및 콘텐츠 스트림의 부분으로서 UA에 대한 콘텐츠의 프래그먼트를, RM에 의해 UA로, 전달하는 단계를 더 포함하고, RM에 의해 UA로 전달되는 콘텐츠의 프래그먼트는 누락 데이터를 완성하는 콘텐츠 데이터를 포함한다.

[0012] [0012] 본 개시의 실시예들은 클라이언트 디바이스의 TA(transport accelerator)에 의해, 클라이언트 디바이스의 UA(user agent)로의 콘텐츠의 전달을 가속화하도록 구성된 장치를 제공한다. 실시예들의 장치는 콘텐츠를 제공하도록 동작 가능한 콘텐츠 서버와 UA 사이의 통신 경로에 배치된 TA를 사용하여 UA에 대한 미디어 송신 동작을 개시하기 위한 수단을 포함하고, TA는 콘텐츠의 어떤 데이터가 콘텐츠 서버로부터 요청되는지를 제어하도록 동작 가능한 RM(request manager) 및 콘텐츠의 데이터가 콘텐츠 서버로부터 언제 요청되는지를 제어하도록 동작 가능한 CM(connection manager)을 포함하고, RM은 콘텐츠의 수신된 데이터를 UE로 전달하기 위해 CM에 의해 사용되는 통신 프로토콜 스택과 UA 사이의 통신 경로에 배치된다. 상기 장치는 CM에 의해 콘텐츠 서버로부터 요청된 콘텐츠의 하나 이상의 청크들의 수신된 데이터를, CM에 의해 RM으로, 전달하기 위한 수단 - 수신된 데이터는 콘텐츠의 하나 이상의 청크들 중 하나의 청크로부터의 누락 데이터이고, 콘텐츠 스트림으로의 조립을 위해 CM에 의해 통신 프로토콜 스택을 통해 RM으로 전달되고, RM은 누락 데이터에 관련하여 TA 동작으로부터 UA를 분리하도록 동작함 - , 콘텐츠의 하나 이상의 청크들의 데이터를 UA에 의해 요청된 콘텐츠의 프래그먼트로 조립하기 위한 수단, 및 콘텐츠 스트림의 부분으로서 UA에 대한 콘텐츠의 프래그먼트를, RM에 의해 UA로, 전달하기 위한 수단을 더 포함하고, RM에 의해 UA로 전달되는 콘텐츠의 프래그먼트는 누락 데이터를 완성하는 콘텐츠 데이터를 포함한다.

[0013] [0013] 본 개시의 실시예들은 클라이언트 디바이스의 TA(transport accelerator)에 의해, 클라이언트 디바이스의 UA(user agent)로의 콘텐츠의 전달을 가속화하기 위한 컴퓨터 프로그램 물건을 제공한다. 컴퓨터 프로그램 물건은 프로그램 코드가 기록된 비일시적인 컴퓨터-판독 가능 매체를 포함한다. 실시예들의 프로그램 코드는 콘텐츠를 제공하도록 동작 가능한 콘텐츠 서버와 UA 사이의 통신 경로에 배치된 TA를 사용하여 UA에 대한 미디어 송신 동작을 개시하기 위한 프로그램 코드를 포함하고, TA는 콘텐츠의 어떤 데이터가 콘텐츠 서버로부터 요청되는지를 제어하도록 동작 가능한 RM(request manager) 및 콘텐츠의 데이터가 콘텐츠 서버로부터 언제 요청되는지를 제어하도록 동작 가능한 CM(connection manager)을 포함하고, RM은 콘텐츠의 수신된 데이터를 UE로 전달하기 위해 CM에 의해 사용되는 통신 프로토콜 스택과 UA 사이의 통신 경로에 배치된다. 프로그램 코드는 CM에 의해 콘텐츠 서버로부터 요청된 콘텐츠의 하나 이상의 청크들의 수신된 데이터를, CM에 의해 RM으로, 전달하기

위한 프로그램 코드 - 수신된 데이터는 콘텐츠의 하나 이상의 청크들 중 하나의 청크로부터의 누락 데이터이고, 콘텐츠 스트림으로의 조립을 위해 CM에 의해 통신 프로토콜 스택을 통해 RM으로 전달되고, RM은 누락 데이터에 관련하여 TA 동작으로부터 UA를 분리하도록 동작함 - , 콘텐츠의 하나 이상의 청크들의 데이터를 UA에 의해 요청된 콘텐츠의 프래그먼트로 조립하기 위한 프로그램 코드, 및 콘텐츠 스트림의 부분으로서 UA에 대한 콘텐츠의 프래그먼트를, RM에 의해 UA로, 전달하기 위한 프로그램 코드를 더 포함하고, RM에 의해 UA로 전달되는 콘텐츠의 프래그먼트는 누락 데이터를 완성하는 콘텐츠 데이터를 포함한다.

[0014] [0014] 클라이언트 디바이스의 TA(transport accelerator)에 의해, 클라이언트 디바이스의 UA(user agent)로의 콘텐츠의 전달을 가속화하도록 구성된 장치가 본 개시의 실시예들에 따라 제공된다. 상기 장치는 적어도 하나의 프로세서 및 적어도 하나의 프로세서에 커플링된 메모리를 포함한다. 실시예들의 적어도 하나의 프로세서는 콘텐츠를 제공하도록 동작 가능한 콘텐츠 서버와 UA 사이의 통신 경로에 배치된 TA를 사용하여 UA에 대한 미디어 송신 동작을 개시하도록 구성되고, TA는 콘텐츠의 어떤 데이터가 콘텐츠 서버로부터 요청되는지를 제어하도록 동작 가능한 RM(request manager) 및 콘텐츠의 데이터가 콘텐츠 서버로부터 언제 요청되는지를 제어하도록 동작 가능한 CM(connection manager)을 포함하고, RM은 콘텐츠의 수신된 데이터를 UE로 전달하기 위해 CM에 의해 사용되는 통신 프로토콜 스택과 UA 사이의 통신 경로에 배치된다. 적어도 하나의 프로세서는 CM에 의해 콘텐츠 서버로부터 요청된 콘텐츠의 하나 이상의 청크들의 수신된 데이터를, CM에 의해 RM으로, 전달하고 - 수신된 데이터는 콘텐츠의 하나 이상의 청크들 중 하나의 청크로부터의 누락 데이터이고, 콘텐츠 스트림으로의 조립을 위해 CM에 의해 통신 프로토콜 스택을 통해 RM으로 전달되고, RM은 누락 데이터에 관련하여 TA 동작으로부터 UA를 분리하도록 동작함 - , 콘텐츠의 하나 이상의 청크들의 데이터를 UA에 의해 요청된 콘텐츠의 프래그먼트로 조립하고, 그리고 콘텐츠 스트림의 부분으로서 UA에 대한 콘텐츠의 프래그먼트를, RM에 의해 UA로, 전달하도록 추가로 구성되고, RM에 의해 UA로 전달되는 콘텐츠의 프래그먼트는 누락 데이터를 완성하는 콘텐츠 데이터를 포함한다.

도면의 간단한 설명

[0015] [0015] 도 1은 본 개시의 실시예들에 따른 확장된 송신 프로토콜 동작에 적응된 시스템을 도시한다.

[0016] [0016] 도 2는 본 개시의 실시예들에 따라 수용될 수 있는 누락 데이터를 갖는 콘텐츠 스트림의 표현을 도시한다.

[0017] [0017] 도 3은 본 개시의 실시예들의 송신 가속기의 동작을 예시하는 사다리도를 도시한다.

[0018] [0018] 도 4는 본 개시의 실시예들의 송신 가속기의 요청 관리기의 동작을 예시한 흐름도를 도시한다.

[0019] [0019] 도 5는 본 개시의 실시예들의 송신 가속기의 접속 관리기의 동작을 예시한 흐름도를 도시한다.

발명을 실시하기 위한 구체적인 내용

[0016] [0020] 용어 “예시적인”은, 오직 “예, 예증 또는 예시로서 기능하는” 것을 의미하도록 본 명세서에서 사용된다. 본 명세서에서 “예시적인” 것으로 설명되는 임의의 양상은 반드시 다른 양상들에 비해 선호되거나 유리한 것으로 해석될 필요는 없다.

[0017] [0021] 이 설명에서, 용어 "애플리케이션"은 또한, 실행가능한 콘텐츠, 예컨대: 객체 코드, 스크립트, 바이트 코드, 마크업 언어 파일들, 및 패치들을 갖는 파일들을 포함할 수 있다. 게다가, 본 명세서에 언급되는 "애플리케이션"은 또한, 특성상 실행가능하지 않은 파일들, 예컨대, 오픈될 필요가 있는 문서들 또는 액세스될 필요가 있는 다른 데이터 파일들을 포함할 수 있다.

[0018] [0022] 본 명세서에 사용되듯이, 용어 "콘텐츠"는 비디오, 오디오, 비디오와 오디오의 결합들, 또는 하나 이상의 품질 레벨들에 있는 다른 데이터, 비트 레이트, 해상도 또는 다른 팩터에 의해 결정되는 품질 레벨을 갖는 데이터를 포함할 수 있다. 콘텐츠는 또한 실행기능한 콘텐츠, 예컨대: 객체 코드, 스크립트들, 바이트 코드, 마크업 언어 파일들, 및 패치들을 포함할 수 있다. 게다가, "콘텐츠"는 또한, 특성상 실행가능하지 않은 파일들, 예컨대, 오픈될 필요가 있는 문서들 또는 액세스될 필요가 있는 다른 데이터 파일들을 포함할 수 있다.

[0019] [0023] 본 설명에서 사용되는 바와 같이, 용어 "프래그먼트"는 사용자 디바이스에 의해 요청되고 그리고/또는 수신될 수 있는 콘텐츠의 하나 이상의 부분들을 지칭한다.

[0020] [0024] 본 설명에서 사용되는 바와 같이, 용어 "스트리밍 콘텐츠"는 시간 기간 동안 콘텐츠의 전달 또는 콘텐츠의 실시간 전달을 가능하게 하는 하나 이상의 표준들에 따라 서버 디바이스로부터 전송되고 사용자 디바이스에

서 수신될 수 있는 콘텐츠를 지칭한다. 스트리밍 콘텐츠 표준들의 예들은, 디인터리빙된(또는 다수의) 채널들을 지원하는 표준들 및 디인터리빙된 (또는 다수의) 채널들을 지원하지 않는 표준들을 포함한다.

[0021] [0025] 본 명세서에 사용되는 바와 같이, 용어 "컴포넌트", "데이터베이스", "모듈", "시스템" 등은 컴퓨터 관련 엔티티, 하드웨어, 펌웨어, 하드웨어 및 소프트웨어의 결합이나 실행중인 소프트웨어 중 어느 하나를 지칭하도록 의도된다. 예를 들어, 컴포넌트는, 프로세서 상에서 구동하는 프로세스, 프로세서, 객체, 실행가능한 것, 실행의 스레드, 프로그램 및/또는 컴퓨터일 수 있지만, 이에 한정되지는 않는다. 예시로서, 컴퓨팅 디바이스 상에서 구동하는 애플리케이션 및 컴퓨팅 디바이스 둘 모두는 컴포넌트일 수 있다. 하나 이상의 컴포넌트들이 프로세스 및/또는 실행의 스레드 내에 상주할 수 있고, 컴포넌트는 하나의 컴퓨터 상에 로컬화될 수 있고 그리고/또는 둘 이상의 컴퓨터들 사이에서 분포될 수 있다. 게다가, 이러한 컴포넌트들은 다양한 데이터 구조들이 저장되어 있는 다양한 컴퓨터 판독 가능 매체로부터 실행될 수 있다. 컴포넌트들은 예컨대 하나 이상의 데이터 패킷들(예를 들면, 로컬 시스템에서, 분산 시스템에서 및/또는 신호에 의한 다른 시스템들과의 네트워크(예를 들어, 인터넷)를 통해 다른 컴포넌트와 상호 작용하는 하나의 컴포넌트로부터의 데이터)을 갖는 신호에 따라 로컬 및/또는 원격 프로세스들을 통해 통신할 수 있다.

[0022] [0026] 본 명세서에 사용되듯이, 용어 "사용자 설비", "사용자 디바이스" 및 "클라이언트 디바이스"는 웹 서버로부터의 콘텐츠를 요청하여 이를 수신하고 정보를 웹 서버에 송신할 수 있는 디바이스들을 포함한다. 이러한 디바이스들은 정적 디바이스들 또는 모바일 디바이스들일 수 있다. 용어 "사용자 장비", "사용자 디바이스" 및 "클라이언트 디바이스"는 상호 교환가능하게 사용될 수 있다.

[0023] [0027] 본 명세서에 사용되듯이, 용어 "사용자"는 사용자 디바이스 상에서 또는 클라이언트 디바이스 상에서 콘텐츠를 수신하고 정보를 웹사이트에 송신하는 개인을 지칭한다.

[0024] [0028] 도 1은 통신 네트워크를 통해, 예컨대, 오디오 데이터, 비디오 데이터, 이미지 데이터, 파일 데이터 등을 포함할 수 있는 콘텐츠의 전달을 제공하기 위해 본 명세서의 개념들에 따라 적응되는 시스템(100)을 도시한다. 따라서, 클라이언트 디바이스(110)는 네트워크(150)를 통해 서버(130)와 통신 상태에 있는 것으로 도시되며, 그로 인해, 서버(130)는 본 개시의 개념들에 따라, 데이터베이스(140)에 저장된 다양한 콘텐츠를 클라이언트 디바이스(110)로 전달할 수 있다. 단일의 클라이언트 디바이스 및 단일의 서버 및 데이터베이스만이 도 1에 제시되었지만, 시스템(100)은 복수의, 임의의 또는 모든 이러한 디바이스들을 포함할 수 있다는 것이 인지되어야 한다. 예를 들어, 서버(130)는, 콘텐츠 전달을 위한 고레벨의 요구를 서빙하기 위해, 서버 팜의 서버를 포함할 수 있고, 여기서, 복수의 서버들이 중앙에 그리고/또는 분포된 구성으로 배치될 수 있다. 대안적으로, 서버(130)는 예컨대, 콘텐츠의 일부 또는 전부가 디바이스 상에 또한 콜로케이트되고 서버(130)를 통해 (예를 들어, 네트워크(150)를 통하는 대신 I/O 엘리먼트(113)를 통해 직접 송신 가속기(120)에 연결되는) 송신 가속기(120)로 제공되는 데이터베이스(140)(캐시)에 상주할 경우, 송신 가속기(120)로서 동일한 디바이스 상에 콜로케이트될 수 있다. 마찬가지로, 사용자들은 복수의 클라이언트 디바이스들을 소유할 수 있고 그리고/또는 복수의 사용자들 각각은, 하나 이상의 클라이언트 디바이스들을 소유할 수 있고, 이들 중 임의의 것 또는 전부는 본 명세서의 개념들에 따른 콘텐츠 전달을 위해 적응된다.

[0025] [0029] 클라이언트 디바이스(110)는 네트워크(150)를 통해 콘텐츠의 전달을 수신하도록 동작가능한 디바이스들의 다양한 구성들을 포함할 수 있다. 예를 들어, 클라이언트 디바이스(110)는 유선 디바이스, 무선 디바이스, 개인 컴퓨팅 디바이스, 태블릿 또는 패드 컴퓨팅 디바이스, 포터블 셀룰러 전화, WiFi 가능 디바이스, 블루투스 가능 디바이스, 텔레비전, 디스플레이를 가진 한 쌍의 안경, 한 쌍의 증강 현실 안경 또는 임의의 이용가능한 방법 또는 인프라구조를 사용하여 서버(130)와 통신할 수 있는 네트워크(150)에 연결되는 임의의 다른 통신, 컴퓨팅 또는 인터페이스 디바이스를 포함할 수 있다. 클라이언트 디바이스(110)는 "클라이언트 디바이스"로 지칭되는데, 그 이유는 클라이언트 디바이스가 서버(130)의 클라이언트로 기능하는 디바이스로서 기능하거나 이에 연결될 수 있기 때문이다.

[0026] [0030] 예시된 실시예의 클라이언트 디바이스(110)는, 프로세서(111), 메모리(112) 및 입력/출력(I/O) 엘리먼트(113)를 포함하는 것으로 본 명세서에 도시된, 복수의 기능 블록들을 포함한다. 간략화를 위해 도 1의 묘사에 도시되지는 않았지만, 클라이언트 디바이스(110)는 추가의 기능 블록들, 예컨대, 사용자 인터페이스, 라디오 주파수(RF) 모듈, 카메라, 센서 어레이, 디스플레이, 비디오 플레이어, 브라우저 등을 포함할 수 있으며, 이들 중 일부 또는 전부는 본 명세서의 개념들에 따른 동작을 위해 사용될 수 있다. 전술한 기능 블록들은 하나 이상의 버스(예컨대, 버스(114))를 통해 동작가능하게 연결될 수 있다. 버스(114)는 연결된 엘리먼트들, 모듈들 및 컴포넌트들이 통신하고 인터폴레이트하도록 허용하기 위한 논리적 그리고 물리적 연결들을 포함할 수 있다.

[0027] [0031] 메모리(112)는 임의의 타입의 휘발성 또는 비휘발성 메모리이며, 일 실시예에서 플래시 메모리를 포함할 수 있다. 메모리(112)는 클라이언트 디바이스(110)에 영구적으로 설치될 수 있거나 탈착식 메모리 엘리먼트, 예컨대 탈착식 메모리 카드일 수 있다. 단일 엘리먼트로 도시되었지만, 메모리(112)는 다수의 개별 메모리들 및/또는 메모리 타입들을 포함할 수 있다.

[0028] [0032] 메모리(112)는, 예컨대 애플리케이션들, 운영체제, 파일들, 전자 문서들, 콘텐츠 등을 형성할 수 있는 다양한 컴퓨터 판독가능 코드 세그먼트들을 저장 또는 다른 방식으로 포함할 수 있다. 예를 들어, 예시된 실시예의 메모리(112)는, 프로세서(예를 들면, 프로세서(111))에 의해 실행될 때 본 명세서에 설명된 대로 동작가능한 논리 회로들을 제공하는, 송신 가속기(TA)(120) 및 UA(129)를 한정하는 컴퓨터 판독가능 코드 세그먼트들을 포함한다. 메모리(112)에 의해 저장되는 코드 세그먼트는 전술한 TA(120) 및 UA(129) 외에도 애플리케이션들을 제공할 수 있다. 예를 들어, 메모리(112)는 본 명세서의 실시예들에 따라 서버(130)로부터의 콘텐츠를 액세싱하는데 유용한 브라우저와 같은 애플리케이션들을 저장할 수 있다. 이러한 브라우저는 웹 브라우저, 예컨대, 웹 콘텐츠를 액세스하고 뷰잉하기 위해 그리고 서버(130)가 웹 서버라면, 연결들(151 및 152)을 통해 네트워크(150)를 거쳐 서버(130)와 HTTP(hypertext transfer protocol)를 통해 통신하기 위한 HTTP 웹 브라우저일 수 있다. 예로서, HTTP 요청은 클라이언트 디바이스(110)의 브라우저로부터, 연결들(151 및 152)을 통해 네트워크(150)를 거쳐 서버(130)로 전송될 수 있다. HTTP 응답은 서버(130)로부터, 연결들(151 및 152)을 통해 네트워크(150)를 거쳐 클라이언트 디바이스(110)의 브라우저로 전송될 수 있다.

[0029] [0033] UA(129)는 서버, 예컨대 서버(130)로부터 콘텐츠를 요청 및/또는 수신하도록 동작가능하다. UA(129)는, 데이터, 예컨대, 멀티미디어 데이터를 요청하고 추가의 프로세싱을 위해 그리고 가능하게는 클라이언트 디바이스(110)의 디스플레이 상에 디스플레이하기 위해 요청된 데이터를 수신하는, 예를 들어, 클라이언트 애플리케이션 또는 프로세스, 예컨대, 브라우저, DASH 클라이언트, HTTP 라이브 스트리밍(HLS) 클라이언트 등을 포함할 수 있다. 예를 들어, 클라이언트 디바이스(110)는, 인터넷 브라우저에서 구동하도록 구성되는 백 미디어, 예컨대, 독립형 미디어 플레이백 애플리케이션 또는 브라우저 기반 미디어 플레이어를 플레이하기 위한 UA(129)를 포함하는 코드를 실행할 수 있다. 실시예들에 따른 동작에서, UA(129)는 어느 콘텐츠 파일의 프래그먼트들의 시퀀스들 또는 프래그먼트들을 스트리밍 콘텐츠 세션 동안 다양한 시점에서 전달을 위해 요청할지를 결정한다. 예를 들어, UA(129)의 DASH 클라이언트 구성은, 각 시점에서, 예컨대, 최근 다운로드된 조건들에 기초하여, 콘텐츠의 어느 표현(예를 들어, 고해상도 표현, 중간 해상도 표현, 저해상도 표현 등)으로부터 어느 프래그먼트를 요청할지를 결정하도록 동작할 수 있다. 마찬가지로, UA(129)의 웹 브라우저 구성은 웹 페이지들 또는 이들의 일부 등에 대해 요청하도록 동작할 수 있다. 전형적으로, UA는 HTTP 요청들을 사용하여 이러한 프래그먼트들을 요청한다.

[0030] [0034] TA(120)는 본원의 개념들에 따라, (예를 들어, 비디오 스트리밍, 파일 다운로드, 웹 기반 애플리케이션들, 일반 웹 페이지 등을 제공하는데 사용될 수 있는 전술한 콘텐츠 프래그먼트들인) 원하는 콘텐츠의 프래그먼트들의 시퀀스들 또는 프래그먼트들의 향상된 전달을 제공하도록 적응된다. 실시예들의 TA(120)는, 표준 인터페이스, 예컨대, HTTP 1.1 인터페이스 구현 표준화된 TCP 전송 프로토콜들을 오직 지원하는 범용 또는 레거시 UA(즉 TA와 상호작용하도록 사전 설계되지 않은 UA)가 프래그먼트 요청들을 수행하는 것을 허용하도록 적응되어 이러한 요청들을 실행하는 TA를 사용하는 것으로부터 역시 유익하게 한다. 부가적으로 또는 대안적으로, 실시예들의 TA(120)는 향상된 인터페이스의 기능을 이용하도록 설계된 UA들에 추가의 이익들을 제공하는 것을 용이하게 하기 위해 향상된 인터페이스를 제공한다. 실시예들의 TA(120)는 기존 콘텐츠 전달 프로토콜들에 따라, 예컨대, 표준화된 TCP 전송 프로토콜들을 구현하는 HTTP 인터페이스를 통해 TCP를 사용하여 프래그먼트 요청을 실행하도록 적응되며, 그로 인해, 일반 또는 레거시 콘텐츠 서버(즉, TA와 상호작용하도록 사전 설계되지 않은 콘텐츠 서버)가, UA 및 클라이언트 디바이스로 프래그먼트들의 향상된 전달을 제공하면서, 요청들을 서빙하게 한다.

[0031] [0035] 전술한 강화된 프래그먼트 전달 기능을 제공하는데 있어서, 본원의 실시예들의 TA(120)는 본 명세서에 설명되듯이 아키텍처 컴포넌트들 및 프로토콜들을 포함한다. 예를 들어, 도 1에 예시된 실시예의 TA(120)는 요청 관리자(RM)(121) 및 연결 관리자(CM)(122)를 포함하며, 이들은 이하에 추가로 설명되듯이, 다양한 향상된 프래그먼트 전달 기능을 제공하기 위해 협력한다.

[0032] [0036] 애플리케이션들, 운영 체제, 파일들, 전자 문서들, 콘텐츠 등을 형성하는 전술한 코드 세그먼트들에 부가하여, 메모리(112)는 클라이언트 디바이스(110)의 기능 블록들에 의해 사용되는 다양한 레지스터들, 버퍼들, 및 저장소 셀들을 포함하거나 그렇지 않으면 제공할 수 있다. 예를 들어, 메모리(112)는 예컨대, 클라이언트 디바이스(110)에 의한 플레이백 및 서버(130)로부터의 스트리밍을 위한 프래그먼트들의 데이터를 스풀링하기 위

해 선입선출(FIFO) 메모리를 제공할 수 있는 플래이-아웃 버퍼를 포함할 수 있다.

[0033] [0037] 실시예들의 프로세서(111)는 클라이언트 디바이스(110)의 기능 및 동작을 제어하기 위해 명령들을 실행할 수 있는 임의의 범용 또는 특수 목적 프로세서일 수 있다. 단일 엘리먼트로 도시되었지만, 프로세서(111)는 다수의 프로세서들, 또는 분포된 프로세싱 아키텍처를 포함할 수 있다.

[0034] [0038] I/O 엘리먼트(113)는 다양한 입력/출력 컴포넌트들을 포함하고 그리고/또는 이에 커플링될 수 있다. 예를 들어, I/O 엘리먼트(113)는, 디스플레이, 스피커, 마이크로폰, 키패드, 포인팅 디바이스, 터치 감응식 스크린, 사용자 인터페이스 제어 엘리먼트들 및 사용자가 클라이언트 디바이스(110)로부터의 출력들을 수신하고 입력 명령들을 제공하게 하는 임의의 다른 디바이스들 또는 시스템들을 포함하고 그리고/또는 이에 커플링될 수 있다. 임의의 또는 모든 이러한 컴포넌트들은 클라이언트 디바이스(110)의 사용자 인터페이스를 제공하도록 사용될 수 있다. 부가적으로 또는 대안적으로, I/O 엘리먼트(113)는 디스크 제어기, 네트워크 인터페이스 카드(NIC), 라디오 주파수(RF) 트랜시버, 및 클라이언트 디바이스(110)의 입력 및/또는 출력 기능을 용이하게 하는 임의의 다른 디바이스들 또는 시스템들을 포함하고 그리고/또는 이에 커플링될 수 있다.

[0035] [0039] 스트리밍 콘텐츠에 액세스하고 플레이하기 위한 동작에서, 클라이언트 디바이스(110)는 링크들(151 및 152)을 이용하여 네트워크(150)를 통해 서버(130)와 통신하여, 렌더링될 때 콘텐츠의 플레이백을 제공하는 (예를 들어, 전술한 프래그먼트들로서) 콘텐츠 데이터를 획득한다. 따라서, UA(129)는 클라이언트 디바이스(110)에 콘텐츠 플레이백 환경을 구축하기 위해 프로세서(111)에 의해 실행된 콘텐츠 플레이어 애플리케이션을 포함할 수 있다. 특정 콘텐츠 파일의 플레이백을 개시할 때, UA(129)는 서버(130)의 콘텐츠 전달 플랫폼과 통신하여 콘텐츠 식별자(예를 들어, 하나 이상의 리스트들, 매니페스트들, 구성 파일들, 또는 원하는 콘텐츠의 미디어 세그먼트들 또는 프래그먼트들, 그리고 이들의 타이밍 경계들을 식별하는 다른 식별자들)를 획득할 수 있다. 미디어 세그먼트들 및 이들의 타이밍에 관한 정보는 UA(129)의 스트리밍 콘텐츠 로직에 의해 사용되어 콘텐츠의 플레이백에 대한 요청 프래그먼트들을 제어한다.

[0036] [0040] 서버(130)는 원하는 콘텐츠를 클라이언트 디바이스들에 서빙하도록 동작가능한 하나 이상의 시스템을 포함한다. 예를 들어, 서버(130)는 네트워크(150)를 통해 다양한 클라이언트 디바이스들에 콘텐츠를 스트리밍하도록 동작 가능한 표준 HTTP 웹 서버를 포함할 수 있다. 서버(130)는 콘텐츠를 사용자 디바이스(110)에 전달할 수 있는 임의의 시스템 또는 방법을 포함하는 콘텐츠 전달 플랫폼을 포함할 수 있다. 콘텐츠는 서버(130)와 통신하고 있는 하나 이상의 데이터베이스들, 예컨대, 예시된 실시예의 데이터베이스(140)에 저장될 수 있다. 데이터베이스(140)는 서버(130)에 저장될 수 있거나, 서버(130)에 통신가능하게 커플링되는 하나 이상의 서버들에 저장될 수 있다. 데이터베이스(140)의 콘텐츠는 다양한 형태들의 데이터, 예컨대, 비디오, 오디오, 스트리밍 텍스트, 및 서버(130)에 의해 시간 기간 동안 클라이언트 디바이스(110)에 전달될 수 있는 임의의 다른 콘텐츠, 예컨대 라이브 웹캐스트 콘텐츠 및 저장된 미디어 콘텐츠를 포함할 수 있다.

[0037] [0041] 데이터베이스(140)는 복수의 상이한 소스 또는 콘텐츠 파일들 및/또는 임의의 특정 콘텐츠의 복수의 상이한 표현들(예를 들어, 고해상도 표현, 중간 해상도 표현, 저해상도 표현 등)을 포함할 수 있다. 예를 들어, 콘텐츠 파일(141)은 고해상도 표현, 결국 전달시, 특정 멀티미디어 컴필레이션의 높은 비트 레이트 표현을 포함할 수 있는 반면, 콘텐츠 파일(142)은 저해상도 표현, 결국 전달시 동일한 특정 멀티미디어 컴필레이션의 낮은 비트 레이트 표현을 포함할 수 있다. 추가적으로 또는 부가적으로, 임의의 특정 콘텐츠의 상이한 표현들은, 예컨대, 콘텐츠 파일(143)에 의해 제공될 수 있는 순방향 에러 정정(FEC) 표현(예를 들어, 콘텐츠 데이터의 중복(redundant) 인코딩을 포함하는 표현)을 포함할 수 있다. URL(Uniform Resource Locator), URI(Uniform Resource Identifier) 및/또는 URN(Uniform Resource Name)은 본 명세서의 실시예들에 따른 이러한 콘텐츠 파일들 모두와 관련되며, 결국, 이러한 URL들, URI들 및/또는 URN들이 요청된 데이터를 식별 및 액세스하기 위해, 아마도 다른 정보, 예컨대 바이트 범위들과 함께 사용될 수 있다.

[0038] [0042] 네트워크(150)는, 무선 네트워크, 유선 네트워크, 광역 네트워크(WAN), 로컬 영역 네트워크(LAN), 또는 본원에 설명된 콘텐츠의 전달에 적절한 임의의 다른 네트워크일 수 있다. 일 실시예에서, 네트워크(150)는 인터넷의 적어도 일부들을 포함할 수 있다. 클라이언트 디바이스(110)는, 예컨대, 네트워크 연결(151)로 표시된 양방향 연결을 통해 네트워크(150)에 연결될 수 있다. 대안적으로, 클라이언트 디바이스(110)는, 예컨대 MBMS(Multimedia Broadcast Multimedia System) 가능 네트워크(예를 들어, 연결들(151, 152) 및 네트워크(150)는 MBMS 네트워크를 포함하고, 서버(130)는 BM-SC(Broadcast Multicast Service Center) 서버를 포함할 수 있음)에 의해 제공되는 양방향 연결을 통해 연결될 수 있다. 연결은 유선 연결이거나 무선 연결일 수 있다. 일 실시예에서, 연결(151)은 무선 연결, 예컨대, 셀룰러 4G 연결, WiFi(wireless fidelity) 연결, 블루투스 연

결, 또는 다른 무선 연결일 수 있다. 서버(130)는, 예컨대, 네트워크 연결(152)로 표현된 양방향 연결을 통해 네트워크(150)에 연결될 수 있다. 서버(130)는 단방향 연결(예를 들어, 3GPP TS.26.346 또는 ATSC 3.0 네트워크에 설명된 프로토콜들 및 서비스들을 이용하는 MBMS 네트워크)을 통해 네트워크(150)에 연결될 수 있다. 연결은 유선 연결이거나 무선 연결일 수 있다.

[0039] [0043] 도 1에 예시된 실시예의 클라이언트 디바이스(110)는 본원의 개념들에 따라 원하는 콘텐츠의 프래그먼트들의 시퀀스들 또는 프래그먼트들의 향상된 전달을 제공하도록 동작가능한 TA(120)를 포함한다. 앞서 논의된 바와 같이, 예시된 실시예의 TA(120)는, 다양한 향상된 프래그먼트 전달 기능을 제공하기 위해 협조하는 RM(121) 및 CM(122)을 포함한다. 실시예들의 UA(129)와 RM(121) 사이의 인터페이스(124) 및 RM(121)과 CM(122) 사이의 인터페이스(123)는 HTTP 유형 연결을 제공한다. 예를 들어, 전술한 인터페이스들은 본원의 실시예들에 따른 개선된 프래그먼트 전달의 특정 기능적 양상들을 지원하기 위해 (예를 들어, HTTP의 시그널링 기술들과 유사한 시그널링 기술들을 사용하여 제공되는) 추가의 시그널링을 비롯하여 표준 HTTP 프로토콜들을 물론 사용할 수 있다.

[0040] [0044] 실시예들에 따른 동작에서, RM(121)은 UA(129)로부터 프래그먼트들에 대한 요청을 수신하고, 요청된 프래그먼트들을 신뢰할 수 있게 수신 및 복구하기 위해 CM(122)으로부터 어떠한 데이터를 요청할지를 결정한다. 본원의 실시예들에 따라, RM(121)은 일반 또는 레거시 UA(즉, RM과 상호작용하도록 미리 설계되지 않은 UA)로부터 프래그먼트 요청들을 수신하고 이에 응답하도록 적응되고, 이로써 그러한 레거시 UA들과의 호환 가능성을 제공한다. 따라서, RM(121)은 TA(120)의 확장된 송신 프로토콜 동작으로부터 UA(129)를 분리하도록 동작할 수 있다. 그러나, 뒤따르는 논의로부터 더 완전히 이해될 바와 같이, UA(129)는 확장된 송신 프로토콜 동작을 위해 적응될 수 있고, 이로써 RM(121) 및 UA(129)는, 가령, RM(121)과 UA(129) 사이의 시그널링의 사용을 통해 확장된 송신 프로토콜 동작의 하나 이상의 특징을 구현하도록 협력하여, 그러한 특징들을 구현한다.

[0041] [0045] 실시예들의 CM(122)에 대해 RM(121)에 의해 이루어지는 데이터 요청들(본원에서 "청크 요청들"로 지칭됨, 여기서 요청된 데이터는 "청크"를 포함함)의 크기는 UA(129)에 의해 요청되는 프래그먼트 및 RM(121)이 복구하는 프래그먼트의 크기보다 훨씬 더 작을 수 있다. 따라서, UA(129)로부터 각각의 프래그먼트 요청은 그 프래그먼트를 복구하기 위해 CM(122)에 대해 다수의 청크 요청들을 생성 및 제조하도록 RM(121)을 트리거링할 수 있다.

[0042] [0046] CM(122)로의 RM(121)에 의해 이루어진 청크 요청들 중 일부는 아직 도착하지 않은 이미 요청된 데이터에 대한 것일 수 있고, RM(121)이 생각한 것은 결코 도착하지 않을 수 있거나, 너무 늦게 도착할 수 있다. 부가적으로 또는 대안적으로, CM(122)으로의 RM(121)에 의해 이루어진 청크 요청들 중 일부는 오리지널 프래그먼트로부터 생성된 FEC 인코딩된 데이터에 대한 것일 수 있고, 이로써 RM(121)은 프래그먼트 또는 이의 일부 부분을 복구하기 위해 CM(122)으로부터 수신된 데이터를 FEC 디코딩할 수 있다. RM(121)은 복구된 프래그먼트들을 UA(129)로 전달한다. 따라서, 본원의 실시예들에 따른 RM들의 다양한 구성들이 존재할 수 있고, 가령, 이것은 FEC 데이터를 사용하지 않고 따라서 오리지널 소스 프래그먼트들로부터 데이터의 부분들만을 요청하는 기본 RM 구성(RM-기본) 및 오리지널 소스 프래그먼트들뿐만 아니라 소스 프래그먼트들로부터 생성된 매칭 FEC 프래그먼트들로부터 데이터의 부분들을 요청할 수 있는 FEC RM 구성(RM-FEC)을 포함할 수 있다.

[0043] [0047] 실시예들의 RM(121)은 타이밍 및/또는 대역폭 이용 가능성 제약들을 인지하지 못할 수 있고, 이로써 RM(121)과 CM(122) 사이의 비교적 간단한 인터페이스를 가능하게 하고, 따라서, RM(121)은 RM(121)에 의한 그러한 제약들의 고려 없이 청크 요청들을 하도록 동작할 수 있다. 대안적으로, RM(121)은, 가령, CM(122) 또는 클라이언트 디바이스(110) 내의 다른 모듈들에 의해 RM(121)에 공급될 수 있는 타이밍 및/또는 대역폭 이용 가능성 제약들의 인지를 위해 적응될 수 있고, 따라서 RM(121)은 그러한 제약들에 기초하여 청크 요청들을 하도록 동작할 수 있다.

[0044] [0048] 실시예들의 RM(121)은 복수의 상이한 CM 구성들과의 동작을 위해 적응된다. 또한, 일부 실시예들의 RM(121)은, 가령, 복수의 CM들로부터 동일한 프래그먼트 또는 프래그먼트들의 시퀀스의 데이터 청크들을 요청하기 위해 하나보다 더 많은 CM과 동시에 인터페이스할 수 있다. 각각의 그러한 CM은, 예를 들면, 상이한 네트워크 인터페이스를 지원할 수 있다(예를 들면, 제 1 CM은 온-디바이스 캐시에 대한 로컬 인터페이스를 가질 수 있고, 제 2 CM은 3G 네트워크 인터페이스에 대한 HTTP/TCP 접속들을 사용할 수 있고, 제 3 CM은 4G/LTE 네트워크 인터페이스에 대한 HTTP/TCP 접속들을 사용할 수 있고, 제 4 CM은 WiFi 네트워크 인터페이스에 대한 HTTP/TCP 제약들을 사용할 수 있고 기타 등등).

[0045] [0049] 실시예들에 따른 동작에서, CM(122)은 청크 요청들을 수신하기 위해 RM(121)과 인터페이스하고, 그러한

요청들을 네트워크(150)를 통해 전송한다. CM(122)은 청크 요청들에 대한 응답들을 수신하고, 응답들을 다시 RM(121)으로 전달하고, 여기서 UA(129)에 의해 요청된 프래그먼트들은 수신된 청크들로부터 분해된다. CM(122)의 기능은 RM(121)에 의해 이루어지는 청크 요청들의 데이터를 언제 요청할지를 결정하도록 동작한다. 본원의 실시예들에 따라, CM(122)은 일반 또는 레거시 서버(즉, TA와 상호작용하도록 미리 설계되지 않은 서버)로부터 청크들을 요청 및 수신하도록 적응된다. 예를 들면, CM(122)이 데이터를 요청한 서버(들)는 표준 HTTP 웹 서버들을 포함할 수 있다. 대안적으로, CM(122)이 데이터를 수신하는 서버(들)는 MBMS(Multimedia Broadcast Multimedia System) 서비스 전개에서 사용되는 BM-SC 서버들을 포함할 수 있다.

[0046] [0050] 위에서 논의된 RM(121)과 마찬가지로, 실시예들에 따른 CM들의 다양한 구성들이 존재할 수 있다. 예를 들면, 다중 접속 CM 구성(예를 들면, CM-mHTTP)이 제공될 수 있고, 이로써 CM은 다수의 TCP 접속들을 통해 HTTP를 사용하도록 적응된다. 다중 접속 CM 구성은, 가령, 네트워크 조건들, 데이터에 대한 요구, 혼잡 윈도우 등에 의존하여 접속들(예를 들면, TCP 접속들)의 수를 동적으로 변경하도록 동작할 수 있다. 다른 예로서, 확장된 송신 프로토콜 CM 구성(예를 들면, CM-xTCP)이 제공될 수 있고, 여기서 CM은 확장된 형태의 TCP 접속(본원에서 xTCP로 지칭됨)뿐만 아니라 HTTP를 사용한다. 그러한 확장된 송신 프로토콜은 본원의 개념들에 따라 TA(120)에 의한 프래그먼트들의 개선된 전달을 가능하게 하도록 적응된 동작을 제공할 수 있다. 예를 들면, xTCP의 실시예는, (패킷들이 분실될 때 TCP의 중복 확인응답 방식과 대조적으로) 전송된 패킷들이 분실될 때조차 확인응답들을 다시 서버에 제공한다. 그러한 xTCP 데이터 패킷 확인응답 방식은, 데이터 패킷들이 누락되었다고 결정한 것에 응답하여, 데이터 패킷들이 송신되는 레이트를 서버가 감소시키는 것을 회피하기 위해 TA(120)에 의해 사용될 수 있다. 또 다른 예로서, 독점(proprietary) 프로토콜 CM 구성(예를 들면, CM-rUDP)이 제공될 수 있고, 여기서 CM은 독점 UDP(User Datagram Protocol) 프로토콜을 사용하고, 서버로부터 응답 데이터를 전송하는 레이트는 일정한 미리 구성된 레이트일 수 있거나, 바람직하지 않게 네트워크를 혼잡하게 하지 않고서, 전송 레이트가 가능한 한 높은 것을 보장하기 위해 프로토콜 내의 레이트 관리가 존재할 수 있다. 그러한 독점 프로토콜 CM은 독점 프로토콜을 지원하는 독점 서버들과 협력하여 동작할 수 있다.

[0047] [0051] 예시된 실시예가 서버(130)로부터의 소스 파일로부터 데이터를 요청하는 CM(122)에 관련하여 논의되었지만, 소소 파일들은 서버들 상에서 이용 가능할 수 있거나, CM이 데이터를 액세스해야 하는 인터페이스의 타입에 의존하여 클라이언트 디바이스 상에 로컬적으로 저장될 수 있다는 것이 인지되어야 한다. 일부 실시예들에서, 매칭 소스 파일들로부터 FEC 인코딩을 사용하여 생성된 리페어 심볼들을 포함하는 FEC 파일들은 또한 서버들 상에서 이용 가능할 수 있다. 그러한 실시예들에서, 예를 들면, 각각의 소스 파일에 대해 하나의 FEC 파일이 존재할 수 있고, 여기서 각각의 FEC 파일은 데이터를 요청하는데 사용되는 CM의 특정 실시예와 독립적인 당분야에 알려진 FEC 인코딩 기술들을 사용하여 소스 파일로부터 생성된다.

[0048] [0052] 또한, 실시예들에 따라, 클라이언트 디바이스(110)는 본원에서 헬퍼 디바이스들로 지칭되는 하나 이상의 다른 디바이스들(예를 들면, 인근에 배치된 다양한 구성들의 디바이스들)에 (예를 들면, WiFi 또는 블루투스 인터페이스를 통해) 접속할 수 있고, 여기서 그러한 헬퍼 디바이스들은 3G 또는 LTE 접속을 통해, 잠재적으로 상이한 헬퍼 디바이스들에 대해 상이한 캐리어들을 통해 서버(130)와 같은 하나 이상의 서버들에 대한 접속을 가질 수 있다. 따라서, 클라이언트 디바이스(110)는 청크 요청들을 서버(130)와 같은 하나 이상의 서버들로 전송하기 위해 헬퍼 디바이스들의 접속을 사용할 수 있다. 이러한 경우에, 헬퍼 디바이스들 각각에 접속하고 청크 요청들을 이들로 전송하고 이들에 대한 응답들을 수신하기 위한 TA(120) 내의 CM이 존재할 수 있다. 그러한 실시예에서, 헬퍼 디바이스들은 동일한 프래그먼트에 대한 상이한 청크 요청을 동일하거나 상이한 서버들로 전송할 수 있다(예를 들면, 동일한 프래그먼트가 다수의 서버들 상의 헬퍼 디바이스들에 대해 이용 가능할 수 있고, 여기서 예를 들면, 상이한 서버들이 상이한 콘텐츠 전달 네트워크 제공자들 중 동일한 것에 의해 제공됨).

[0049] [0053] 실시예들에 따른 동작에서, RM(121)으로부터의 요청들에 응답하여 일반적으로 완전한 데이터를 제공하지만, 그럼에도 불구하고 CM(122)은 일부 상황들에서 부분적인 응답들을 RM(121)을 제공할 수 있다. 그러한 부분적인 응답은, 데이터의 일부가 RM에 의해 요청되지만 데이터의 전부가 요청되지는 않는 경우에, RM(121)에 제공될 수 있다(즉, CM(122)에 의해 제공된 하나 이상의 응답들 내에 데이터 갭들 또는 홀들이 존재할 수 있음). 실시예들의 CM(122)은 (예를 들면, 아마도 각각의 응답에서 한번에 몇몇의 바이트들만을 RM에 제공하는 것을 삼가기 위해 응답을 어그리게이팅하기에 충분한 지연을 포함하여) 가능한 한 바로 응답들을 RM(121)에 제공하도록 동작한다. 예를 들면, CM(122)은 인터페이스(123)의 TCP 스택을 통해 데이터를 RM(121)으로 전달할 수 있지만, 요청된 청크의 콘텐츠 중 일부 부분은 CM(122)에 의해 수신되지 않는다. 따라서, 상술된 부분적인 응답들은 응답들이 이용가능한 한 바로 실시예들에 따라 RM(121)에 제공될 수 있다.

[0050] [0054] 앞서 말한 부분적인 응답들을 가능하게 하는데 있어서, 실시예들의 CM(122)에 의해 RM(121)에 제공되는

응답들은 반환되는 데이터의 바이트 범위들의 표시를 포함할 수 있다. 예를 들면, CM에 의해 제공되는 프래그먼트의 바이트들(0-15999)에 대한 RM 요청에 대한 응답은 바이트들(0-3599), 바이트들(4500-7699) 및 바이트들(9000-14999)을 포함할 수 있고(즉, 바이트들(3600-4499, 7700-8999, 및 15000-15999)에 대한 데이터가 응답으로부터 누락됨), 따라서 실시예들의 CM(122)으로부터의 응답은 부분적인 응답을 표시하고 및/또는 반환되는 데이터의 바이트 범위를 표시하도록 증강(augment)되고, 여기서 예를 들면, 바이트 범위들은 HTTP 응답 헤더 또는 그의 등가물에서 전달된다. 반환되는 데이터의 바이트 범위를 표시하도록 동작 가능한 구현의 추가의 예로서, 실시예들의 CM(122)과 RM(121) 사이의 인터페이스는 개별적인 소프트웨어 API를 포함할 수 있고, 여기서 "양호한" 및/또는 "불량한" 바이트 범위들은 실제 데이터 스트림으로부터 별개로 시그널링된다. 부가적으로 또는 대안적으로, 시그널링은, CM으로부터 RM으로 반환되는 데이터가 수신된 실제 데이터, 더미 데이터 양자 및 어느 부분이 정확하게 수신되는지 및 어느 부분이 아닌지를 지정하기 위한 특수 마커들을 포함한다는 점에서, "스트림 내"에서 수행될 수 있다. 예를 들면, 특수 비트 패턴들은, 그러한 시그널링에 대한 부가적인 인터페이스를 사용하지 않고서, 데이터의 수신인(예를 들면, 이러한 경우에 RM)이 데이터 스트림을 파싱하고 정확히 수신된 데이터만을 추출하는 것을 가능하게 하기 위한 마커들로서 사용될 수 있다. 따라서, 실시예들의 RM(121)과 CM(122) 사이에 제공된 TA(120)의 인터페이스(123)는 앞서 말한 통신을 가능하게 하기 위해 개선된 인터페이스(예를 들면, HTTP 1.1 인터페이스 프로토콜들에 기초한 개선된 인터페이스)를 제공한다.

[0051] [0055] 실시예들의 CM(122)은 부가적으로 또는 대안적으로 본원의 개념들에 따라 유용한 다른 정보를 RM(121)에 제공할 수 있다. 예를 들면, CM(122)은, 가능하게는, 측정이 될 때를 표시하는 타임 스탬프와 결합하여, 다운로딩된 데이터의 종합 양의 표시를 RM(121)에 제공할 수 있다. RM(121)은, RM이 다수의 CM들과 인터페이싱할 때, 청크 요청들 중 어느 부분을 특정 CM으로 전송할지를 결정하는 것을 포함하여, 다양한 목적들로 그러한 정보를 사용할 수 있다. 다른 예로서, RM(121)은 이러한 정보를 어그리게이팅하고, UA가 미래에 어떠한 프래그먼트 요청들을 RM에 제공할지를 결정하기 위해 정보를 사용할 수 있도록 하기 위해 이를 UA(129)에 제공할 수 있다. 예를 들면, 다운로드 속도가 마지막 5 초에 걸쳐 평균 1 Mbps라는 것을 표시하는 정보를 RM(121)이 UA(129)에 다시 전달하는 경우에, UA는 다음의 프래그먼트 요청이 750 Kbps로 인코딩된 비디오 프래그먼트에 대한 것이어야 한다고 결정하기 위해 이러한 정보를 사용할 수 있고, 반면에 다운로드 속도가 마지막 5 초에 걸쳐 평균 2 Mbps라는 것을 정보가 표시하면, UA는 다음의 프래그먼트 요청이 1.5 Mbps로 인코딩된 비디오 프래그먼트에 대한 것이어야 한다고 결정할 수 있다. 따라서, 실시예들의 UA(129)와 RM(121) 사이에 제공된 TA(120)의 인터페이스(124)는 앞서 말한 통신을 가능하게 하기 위한 개선된 인터페이스(예를 들면, HTTP 1.1 인터페이스 프로토콜들에 기초한 개선된 인터페이스)를 제공한다.

[0052] [0056] 이전에 언급된 바와 같이, 실시예들의 TA(120)는 가령, HTTP 인터페이스를 통해 TCP를 사용하여 기존의 콘텐츠 전송 프로토콜들에 따라 프래그먼트 요청을 실행하도록 적응된다. 따라서, 실시예들의 CM(122)은 TCP에 따른 시그널링을 사용하여, 가령, 그의 TCP 수신기를 사용하여 청크 요청들을 서버(130)에 제공하도록 적응된다. 그러나, 기존의 TCP 구현들은, 누락 또는 재정렬된 패킷들이 스트림에서 검출될 때 둔화하도록 동작한다. 따라서, 본원의 실시예들이 CM(122) 및 따라서 TA(120)는 확장된 송신 프로토콜을 구현하도록 적응된다. 예를 들면, CM(122)은 본원에서 CM-xTCP로 지칭되는 확장된 송신 프로토콜 구성을 포함할 수 있고, 이로써 CM(122)은 본원에서 xTCP로 지칭되는 확장된 형태의 TCP 접속뿐만 아니라 HTTP를 사용한다. CM(122)은 본원에 설명된 xTCP 동작을 위해 적응된 TCP 수신기(즉, xTCP 수신기)를 포함할 수 있다. xTCP의 실시예는, 패킷들이 분실될 때조차 확인응답을 서버(130)에 다시 제공함으로써 TA(120)에 의한 UA(129)로의 프래그먼트들의 개선된 전달을 가능하게 한다. 즉, 실시예들의 xTCP는 누락된 패킷들의 재송신을 위한 중복 ACK들로서 그러한 거동을 우회하는 수정된 TCP를 제공한다. 본원의 개념들에 따른 xTCP의 구현들은 네트워크에서 과도한 혼잡을 발생시키지 않고서 빠른 다운로드를 달성하도록 적응된다. 동작 시에, xTCP 프로토콜은, 데이터가 요청의 프리픽스(prefix)가 아닐지라도 서버(130)로부터 수신된 모든 데이터를 즉시 제공할 수 있다(즉, 자신의 요청들에 응답하여 CM으로 반환되는 데이터 내에 홀들이 존재할 수 있음). 앞서 말한 xTCP 데이터 패킷 확인응답들은, 데이터 패킷들이 누락되었다고 결정한 것에 응답하여 데이터 패킷들이 송신되는 레이트를 서버가 감소시키는 것을 회피하도록 사용된다.

[0053] [0057] xTCP의 앞서 말한 동작을 이해하는데 있어서, 기존의 TCP에 따라 제공된 동작을 검토하는 것이 도움이 된다. 전형적인 TCP 수신기는 NextByteExpected 및 LastByteReceived을 포함하여, 스트리밍 콘텐츠에 관련하여 몇몇의 중요한 내부 파라미터들을 사용한다. 이러한 파라미터들은 도 2의 콘텐츠 스트림(200)에서 NextByteExpected(210) 및 LastByteReceived(220)로 도시된다. NextByteExpected 전의 모든 데이터가 완전히 수신되고, 따라서 애플리케이션에 의해 판독되고, TCP 계층에 의해 나중에 폐기될 수 있다. 그러나, NextByteExpected 및 LastByteReceived 사이에서, 누락 데이터(예를 들면, 누락 데이터 패킷들 또는 수신된 데

이터 내의 "홀")가 존재할 수 있거나 존재할 수 없다. 누락 데이터(230)로 표현된 바와 같이, 데이터 내에 홀들이 존재할 때, LastByteReceived의 값은 NextByteExpected의 값보다 더 클 것이고, 이러한 2 바이트 오프셋들 사이의 데이터는 TCP 계층에 의해 버퍼링되어야 하고, 애플리케이션에 의해 판독될 수 없다. 수신 윈도우(RWND)의 좌측 에지는 또한 NextByteExpected보다 더 앞설 수 없다.

[0054] [0058] 기존의 TCP 구현들에서, 수신기는 NextByteExpected까지 데이터 수신인 확인응답(ACK)만을 제공할 것이다. NextByteExpected 및 NextByteExpected에 대응하는 데이터의 바이트를 포함하지 않는 LastByteReceived 사이에 데이터 누락된 비순차적인(out of order) 패킷들이 수신된 후에(즉, NextByteExpected에서 시작하여 누락 데이터의 홀이 여전히 존재함), 수신기는 NextByteExpected에 대응하는 중복 ACK들을 전송할 것이다. 전송기(예를 들면, 콘텐츠 서버)는 네트워크에서 혼잡의 사인들로서 그러한 중복 ACK들을 해석하고, 혼잡 윈도우를 감소시킴으로써 전송 레이트를 둔화시킴으로써 반응할 것이고, 여기서 혼잡 윈도우는 임의의 시간에 미처리(outstanding)된 것일 수 있는 바이트들의 수를 결정하는 인수이다. 예를 들면, 콘텐츠 서버는 3과 같은 미리 결정된 수의 중복 ACK들을 수신한 후에 혼잡 윈도우를 2 배 감소할 수 있고, 이로써 TCP의 전송 레이트를 늦춘다. 혼잡 윈도우는 단지 RTT(Round Trip Time)(즉, 데이터에 대한 요청에 대한 응답을 위해 요구된 시간)에 비례하여 증가된다. 예를 들면, 추가의 중복 ACK들이 검출되지 않는 경우, 콘텐츠 서버는 RTT 당 하나의 패킷만큼 혼잡 윈도우를 증가시킬 수 있다. 따라서, TCP의 전송 레이트는, 특히, 큰 RTT가 경험되는 경우에 종종 매우 느리게 증가한다.

[0055] [0059] 본원의 실시예들의 CM(122)에 의한 xTCP의 구현은, 패킷 분실들 또는 특정 시나리오들에서의 패킷 분실들이 서버(130)에 의한 전송 레이트의 둔화를 발생시키지 않는 그러한 방식으로 TCP의 동작을 변경한다. 실시예들에 따른 동작에서, xTCP는, 패킷들이 서버(130)로부터 클라이언트 디바이스(110)로의 송신 시에 분실될 때조차 확인응답들을 다시 서버(130)에 제공함으로써 TA(120)에 의한 UA(129)로의 프래그먼트들의 개선된 전달을 가능하게 한다. 또한, 실시예들은 본원에 설명된 xTCP 동작을 가능하게 하도록 적응된 요청 전송 기술들을 구현할 수 있다. 예를 들면, 청크 요청들이 적시에 전송되는 것을 보장하기 위해, TA(120)는 (예를 들면, 전형적인 TCP 수신기의 부분으로서 종종 구현될 수 있는) 네이글(Nagle)의 알고리즘을 디스에이블하도록 동작할 수 있고, 이것은 그렇지 않다면 요청들을 전송할 때 부가적인 지연(예를 들면, 대략 RTT)을 도입할 수 있다. 이것은 CM(122)이 TCP_NODELAY 소켓 옵션을 설정함으로써 실시예들에 따라 달성될 수 있다. 본원의 실시예들에 의해 구현되는 요청 전송 기술들의 추가의 적응은 청크에 대한 요청의 크기를 작게 (예를 들면, 하나의 MSS(maximum segment size) 미만, 그래서 그들이 하나의 패킷에 맞춤) 만들고, 가령, 청크 요청이 몇몇의 패킷들로 전송되는 것으로 인해 청크 요청의 부분이 분실될 위험성을 감소시키기 위해 하나의 전송 시스템 호에서 청크 요청들을 전송하는 것을 포함할 수 있다.

[0056] [0060] TCP와 같은 많은 콘텐츠 전송 프로토콜들에 따라, 업링크는 비교적 적은 데이터만을 전송하고, 따라서, 획득되지 않은 패킷들의 빠른 재송신을 획득하는 것이 어려울 수 있다. 따라서, 업링크 분실들을 완화하도록 동작 가능한 기술들은 실시예들에 따라 TA(120)에 의한 프래그먼트들의 개선된 전달을 가능하게 하도록 구현될 수 있다. 예를 들면, 각각의 청크 요청이 하나의 패킷으로 전송되고, TA(120)가 접속 시에 2 개의 요청들을 파이프라이닝하는 경우에, 2 개의 파이프라이닝된 청크 요청들 중 제 1 청크 요청이 분실되면, TA(120)에 단일의 중복 ACK가 제공된다. 그러한 중복 ACK의 수신인은 종종 재송신을 트리거링하기에 충분하지 않고, 따라서 제 1 청크 요청은 재송신 후에 타임아웃이 발생할 때까지 일반적으로 재전송되지 않을 것이고, 이것은 비교적 긴 시간일 수 있다. 제 1 청크 요청에 대한 응답은 마찬가지로 상당히 지연될 것이다. 따라서, 업링크 분실 완화를 위한 기술들은 본원의 실시예들의 TA(120)에 의해 구현된다.

[0057] [0061] 업링크 분실 완화 기술들을 구현하는데 있어서, 일부 콘텐츠 전송 프로토콜 구현들(예를 들면, TCP)이 상술된 업링크와 같이 더 적은 데이터를 송신하는 스트림들에 대한 알고리즘들을 갖는다는 것이 인지되어야 한다. 리눅스(Linux)에서, 예를 들면, "신(thin) 스트림" 최적화들은, 더 적은 비행중인(in-flight) 데이터가 존재하는 상황들에서 단일 중복 ACK(예를 들면, 3 개의 중복 ACK들 대신에) 후의 재송신을 제공한다. 본원의 xTCP 동작을 구현하는 실시예들은, xTCP 콘텐츠 송신 가속 시에 업링크 분실들의 효과를 완화하기 위해 청크들을 요청하는데 사용되는 제어 스트림에 관련하여, 가능하게는, (예를 들면, 아래에 설명되는) 다른 특징들 또는 기술들과 관련하여 그러한 특징을 가능하게 할 수 있다.

[0058] [0062] 실시예들은, 요청 세그먼트가 분실될 때 중복 ACK를 트리거링할 목적으로 각각의 청크 요청 후에 프로토콜 데이터 유닛들(예를 들면, TCP 세그먼트들)을 부가하도록 적응될 수 있다. 그러한 기술을 구현하는데 있어서, HTTP GET 요청들이 불변의 프리픽스(immutable prefix) "GET"를 갖고, 이로써 TA(120)는 먼저 추측에 근거하여(speculatively) 그 프리픽스를 전송하고, 이어서 나중 시간에, 가령, 실제로 전송 요청이 존재할 때 요청

의 리마인더를 전송하도록 적응될 수 있다는 것이 인지되어야 한다. 이어서, TA(120)는 다음의 요청에 대해 추측에 근거한 "GET"를 다시 전송할 수 있다. 요청의 관련 부분이 분실되는 경우에, DUP ACK는 서버(130)가 수신하는 "GET" 부분에 의해 트리거링되어, 분실된 청크 요청을 재송신하도록 TA(120)를 유도할 것이다. 마찬가지로, 추측에 근거한 프리픽스는 동일한 방식으로 재송신될 수 있다. 이러한 기술의 구현을 통해, 요청 당 더 많은 패킷들이 사용되는 것으로 인해 다소 더 많은 수의 재송신들의 비용을 지불하지만, 최악의 경우의 지연들(예를 들면, 타임아웃들로 인한 재송신들)이 회피될 수 있다.

[0059] [0063] 도 3의 사다리도(UA(129), RM(121), CM(122) 및 서버(130) 사이의 상호작용을 나타내는 시퀀스도(300)) 및 도 4의 흐름도(RM(121)에 의한 프로세싱을 나타내는 흐름(400)) 및 도 5의 흐름도(CM(122)에 의한 프로세싱을 나타내는 흐름(500))은 본원의 개념들에 따른 이러한 동작을 예시한다.

[0060] [0064] UA(129)는, 예를 들면, 스트리밍 콘텐츠 세션과 연관하여 콘텐츠의 프래그먼트에 대한 요청(예를 들면, 도 3의 프래그먼트 요청(301))을 발행할 수 있다. UA(129)에 의해 발행된 프래그먼트 요청은 본원의 개념에 따른 표준 통신 프로토콜들(예를 들면, TCP) 또는 확장된 통신 프로토콜(예를 들면, xTCP)을 사용할 수 있다. 예를 들면, 아래의 논의로부터 인지될 바와 같이, 실시예들의 RM(121)은 완전한 프래그먼트 응답들을 UA(129)에 제공하도록 동작하고, 이로써 송신 가속기의 이점들을 수용하기 위해 레거시 UA의 수정을 요구하지 않고서 TA(120)의 개선된 동작을 가능하게 한다.

[0061] [0065] 도 3-5에 예시된 예들이 간단함을 위해 단일 프래그먼트 및 단일 청크 요청/응답을 도시하지만, 다수의 그러한 요청들 및 응답들이, 병렬이든지 및/또는 직렬이든지, 수행될 수 있다는 것이 인지되어야 한다. 예를 들면, UA(129)는 임의의 특정 시점에서 RM(121)에 펜딩된 복수의 프래그먼트 요청들을 가질 수 있다. 또한, RM(121)은 UA(129)에 의해 요청된 하나 이상의 프래그먼트들에 대해 CM(122)에 펜딩된 복수의 청크 요청들을 가질 수 있고, 마찬가지로 CM(122)은 임의의 특정 시점에서 서버(130)에 펜딩된 복수의 청크 요청들을 가질 수 있다.

[0062] [0066] RM(121)은, 가령, 본원의 개념에 따른 표준 통신 프로토콜들(예를 들면, TCP) 또는 확장된 통신 프로토콜(예를 들면, xTCP)을 사용하여 직렬 및/또는 병렬로 다음의 청크 요청(예를 들면, 도 3 및 도 4의 블록(401)의 청크 요청(302))으로서 CM(122)으로 발행될 수 있는 복수의 청크 요청들로 프래그먼트 요청을 분해(break)할 수 있다. 청크 요청(302)과 같은 청크 요청들은 실시예들에 따라 파일 또는 파일의 바이트 범위에 대한 HTTP 요청들을 포함할 수 있다.

[0063] [0067] RM(121)으로부터 청크 요청을 수신한 것에 응답하여, CM(122)은 콘텐츠를 서버(130)로 스트리밍하기 위해 콘텐츠 데이터의 다음 청크에 대한 요청(예를 들면, 도 3의 청크 요청(303))을 발행할 수 있다(도 5의 블록(501)). CM(122)과 RM(121) 사이에 제공되는 시그널링은 청크 크기 시그널링과 같은 청크 요청 준비(readiness) 시그널링 이외에 또는 대안으로 정보를 포함할 수 있다. 실시예들의 CM(122)은 서버(130)로 발행된 청크 요청에 관련하여 확장된 통신 프로토콜(예를 들면, xTCP)을 사용한다. 그러한 확장된 통신 프로토콜의 동작이 서버(130)의 거동을 제어할 수 있지만, 그럼에도 불구하고 서버(130)는 본원의 실시예들에 따른 확장된 통신 프로토콜에 따른 동작을 위해 구체적으로 적응되지 않은 표준 레거시 서버를 포함할 수 있다는 것이 인지되어야 한다. 그러나, 이어지는 논의로부터 더 양호하게 이해될 바와 같이, 실시예들의 CM(122)의 동작은 확장된 통신 프로토콜에 따라 적응된다.

[0064] [0068] CM(122)은 요청된 콘텐츠 데이터를 포함하는 복수의 데이터 패킷들(예를 들면, 도 3의 패킷(들)(304)) 또는 이들의 일부 부분(예를 들면, 도 2의 콘텐츠 스트림(200)으로서)을 수신할 수 있다(도 5의 블록(502)). 콘텐츠 스트림이 임의의 홀들 또는 누락 데이터를 포함하지 않는다면(예를 들면, 도 5의 블록(503)에서 어떠한 늦은 또는 비순차적인 데이터도 검출되지 않는다면), CM(122)은 전형적인 통신 프로토콜(예를 들면, TCP) 동작에 따라 적절한 하나 이상의 ACK(예를 들면, 도 3의 ACK(들)(305))를 서버(130)로 전송하고(도 5의 블록(504)), RM(121)에 제공하기 위해 데이터 패킷들을 요청된 청크로 조립할 수 있다(도 5의 블록(505)).

[0065] [0069] 그러나, 확장된 통신 프로토콜(예를 들면, xTCP)에 따른 CM(122)의 동작 시에, CM은, 누락 데이터(예를 들면, 도 2의 누락 데이터(230))가 존재하는지 여부와 상관없이, ACK들을 수신된 마지막 바이트(예를 들면, 도 2의 LastByteReceived(220))까지 서버(130)로 전송할 수 있고, 이로써 서버(130)가 레거시 통신 프로토콜(예를 들면, 전형적인 TCP) 기술들에 따라 동작 가능한 경우에서조차 서버(130)의 동작이 혼잡 윈도우를 감소시키는 것을 방지한다. 따라서, 콘텐츠 스트림이 홀들 또는 누락 데이터를 포함하면(예를 들면, 늦은 또는 비순차적인 데이터가 도 5의 블록(503)에서 검출되면), CM(122)은 LastByteReceived(220)에 대해 적절한 하나 이상의 ACK들(예를 들면, 도 3의 ACK(들)(302) 및 도 5의 블록(506))을 서버(130)로 전송할 수 있다. 예를 들면,

CM(122)은, 선택적으로 아래에 설명되는 더미 데이터를 포함하는 수신된 데이터를 그의 xTCP 수신기로 전달하도록 동작하거나, 그렇지 않다면 누락된 데이터가 CM에 도착한 것처럼 수신 확인응답을 콘텐츠 서버로 전송하도록 xTCP 수신기를 트리거링하기 위해 CM에 의해 제어된다. 따라서, 콘텐츠 스트림에서 누락 데이터가 존재함에도 불구하고, 서버(130)가 갭 없이 연속해서 ACK들을 수신할 수 있기 때문에, 누락 패킷들은 전송기(예를 들면, 레거시 TCP 전송기)를 둔화시키지 않을 것이다.

[0066] [0070] 실시예들의 CM(122)은, 가령, 네트워크 혼잡을 회피하도록 적응된 동작을 제공하기 위해, 누락 데이터가 검출되는 경우, 하나 이상의 ACK들이 전송기에 제공되는 때를 결정하기 위한 로직을 사용할 수 있다. 예를 들면, 누락 데이터가 반복적으로 검출될 때(예를 들면, 지정된 시간 기간 내에 미리 결정된 수의 누락 패킷들, 미리 결정된 수의 연속적인 누락 데이터 패킷들 등), CM(122)은 네트워크 혼잡을 회피하기 위한 제어를 제공하기 위해 하나 이상의 중복 ACK들을 서버(130)로 전송하도록 동작할 수 있다. CM(122)과 서버(130) 사이에 TCP 접속이 사용되고, 누락 또는 재정렬된 패킷이 존재할 때 서버(130)가 데이터 패킷들의 송신을 둔화시키기 위한 전형적인 TCP 거동을 구현하는 경우에, CM(122)의 xTCP 수신기는 서버(130)에 의한 데이터 송신 레이트에서 감소(예를 들면, 혼잡 윈도우의 감소)를 인보크하기 위해 알려진 다수의 중복 ACK들을 전송하도록 동작할 수 있다. 부가적으로 또는 대안적으로, 실시예들의 CM(122)은 네트워크 혼잡을 회피하기 위한 제어를 제공하기 위해 하나 이상의 ACK들을 서버(130)로 전송하는 것을 보류하도록 동작할 수 있다(예를 들면, CM은 서버에 의한 데이터 송신 레이트에서 감소를 인보크하기 위해 알려진 다수의 ACK들을 보류하도록 동작할 수 있다).

[0067] [0071] 부가적으로 또는 대안적으로, CM(122)은 (예를 들면, 반드시 누락 데이터에 응답하지는 않지만 일반적으로 레이트 제어 메커니즘으로) 둔화시키기 위한 중복 ACK들을 전송할 수 있고, CM(122)이 네트워크 혼잡을 검출할 때 그리고 전송 레이트를 둔화시키는 경우에, 데이터 송신이 이로울 것이다. 예를 들면, CM(122)은, 누락 데이터가 존재하지 않을 때조차 중복 ACK들을 전송할 수 있고, 이것은, CM(122)이 잠재적으로 서버(130)로부터 데이터의 일부 작은 부분을 두 번 수신하는 것을 희생하고서 전송기 혼잡 윈도우를 감소시키고 따라서 전송 레이트를 둔화시키도록 서버(130)로 시그널링할 것이다. 다른 예로서, CM(122)은 CM(122)에 의해 수신되지 않는 데이터 중 적어도 일부의 누락을 시그널링하고, 한편 동시에 전송기 혼잡 윈도우를 감소시키고 따라서 전송 레이트를 둔화시키도록 서버(130)로 시그널링하고, 또한 이전의 송신에서 분실된 누락 데이터의 부분들을 서버(130)부터 수신하여, 반드시 모든 누락 데이터는 아니지만 누락 데이터 중 일부에 대한 중복 ACK들을 전송할 수 있다.

[0068] [0072] 앞서 말한 것으로부터 인지될 수 있듯이, CM(122)의 로직은 네트워크 혼잡 회피를 위한 클라이언트 측 제어를 제공할 수 있다. 서버(130)가 본원의 확장된 통신 프로토콜의 기능들을 구현하도록 적응되지 않은 레거시 콘텐츠 서버를 포함하는 경우에서조차, 그러한 클라이언트 측 제어가 제공될 수 있다.

[0069] [0073] 또한, 누락 데이터를 검출한 것에 응답하여, 본원의 실시예들의 CM(122)의 확장된 통신 프로토콜 수신기는 데이터를 수신하는 것을 가능하게 하고 및/또는 네트워크 혼잡을 회피하도록 수신 윈도우를 조절할 수 있다. 예를 들면, CM(122)의 확장된 통신 프로토콜 수신기가 누락 패킷들 및/또는 RTT 변형들을 발견할 때, CM(122)은 앞서 논의된 바와 같이 ACK들을 LastByteReceived까지 전송하도록 동작할 수 있을 뿐만 아니라, 네트워크에서 혼잡을 제어하기 위해 수신 윈도우 크기를 조절하도록 동작할 수 있다. 수신 윈도우 스케일링은, 누락 데이터 없이 데이터 패킷들을 연속하여 수신하는 것이 수신 윈도우를 (예를 들면, 미리 결정된 최대 수신 윈도우 크기까지) 증가시킬 방식에 따라 이루어질 수 있고, 반면에 더 많은 누락 패킷들을 검출하는 것은 CM(122)의 확장된 통신 프로토콜 수신기로 하여금 수신 윈도우를 (예를 들면, 미리 결정된 최소 수신 윈도우 크기 아래로) 스케일링 다운하게 할 것이다. 실시예들에 따라 구현된 수신 윈도우 스케일링은 부가적으로 또는 대안적으로, 가령, RTT가 비교적 긴 경우 다소 더 큰 수신 윈도우 스케일링 및 RTT가 비교적 짧은 경우 다소 더 작은 수신 윈도우 스케일링을 구현하기 위해 RTT를 고려할 수 있다. 따라서, 실시예들에 따른 수신 윈도우 스케일링 동작은, 네트워크에서 상당한 혼잡을 도입하지 않고서, 전송기 거동을 적절한 레벨의 적극성으로 제어하기 위해 수신 윈도우를 사용할 수 있다.

[0070] [0074] 수신 윈도우 크기의 앞서 말한 조절은 수신 버퍼 크기의 변화 없이 실시예들에 따라 구현될 수 있다. 즉, 수신 윈도우 스케일링은 전송기(예를 들면, 서버(130))에 수신 윈도우 조절을 통지하고 따라서 전송기 거동을 제어하기 위한 기술로서 구현될 수 있다. 따라서, 실시예들의 CM(122)의 확장된 통신 프로토콜 수신기는 수신 버퍼 크기를 변경하지 않고서 조절된 수신 윈도우 크기를 서버(130)로 전송하도록 동작할 수 있다.

[0071] [0075] 예시된 실시예의 CM(122)은 LastByteReceived(220)까지 수신된 데이터를 RM(121)에 제공한다(도 3의 청크 데이터(306) 및 도 5의 블록(507)). RM(121)이 데이터 분석 기술들, 대역외 시그널링(예를 들면, CM(122)에

의해 설정된 외부 레지스터들 또는 플래그들) 등을 사용하여 CM(122)에 의해 제공되는 청크 내에서 누락 데이터를 검출하도록 동작할 수 있지만, 실시예들은 CM(122)과 RM(121) 사이의 통신에서 누락 데이터의 범위(예를 들면, 시작 바이트 오프셋 및 바이트들의 수)를 표시하는 데이터를 제공하도록 동작한다. 예를 들면, RM(121)과 CM(122) 사이의 통신 인터페이스가 표준 통신 프로토콜을 사용하는 경우에, RM(121)의 로직은, 가령, 청크 데이터 자체, 헤더 데이터, CM(122)에 의해 설정된 레지스터들 또는 플래그들 등을 분석하는 것을 통해 청크 요청에 응답하여 제공된 불완전한 데이터를 통해 청크들을 식별하도록 적응될 수 있다. RM(121)과 CM(122) 사이의 통신 인터페이스가 확장된 통신 프로토콜을 사용하는 경우에, 아래에 추가로 논의되는 바와 같이, 청크 요청에 응답하여 반환된 불완전한 데이터를 갖는 청크들에 관련하여 시그널링이 제공될 수 있다.

[0072] [0076] 누락 데이터(즉, 상술된 홀들)를 갖는 청크 데이터(306)를 RM(121)에 제공할 때, CM(122)은 완전한 것처럼 보이는 데이터를 RM(121)의 통신 프로토콜 스택에 제공하기 위해 홀들을 더미 데이터(예를 들면, 제로들)로 채우도록 동작할 수 있다. 예를 들면, CM(122)의 xTCP 수신기는, CM이 수신된 데이터(그 더미 데이터를 가짐)를 RM(121)으로 전달하기 전에 누락 데이터를 더미 데이터로 채우도록 동작할 수 있다. 누락 데이터의 범위에 관한 상술된 정보는 나중에 데이터를 완성하는 것뿐만 아니라 불완전한 동안에 데이터가 UA(129)에 제공되는 것을 방지하는 것 둘 모두에서 RM(121)에 의해 사용될 수 있다. 즉, RM(121)의 로직은, 가령, 누락 데이터의 범위에 관한 정보에 대한 참조를 통해 CM(122)에 의해 제공되는 데이터가 불완전하다는 것을 인식하고, 따라서 요청된 프래그먼트의 모든 데이터가 수신될 때까지 그 데이터를 UA(129)에 제공하는 것을 보류할 수 있다. 실시예들에 따른 동작에서, CM(122)은, 누락 데이터의 범위에 관한 정보를 통해, 더미 데이터를 갖는 수신된 데이터를 RM(121)으로 전달하기 위해 xTCP 수신기를 제어할 수 있다.

[0073] [0077] RM(121)은, UA(129)에 의해 요청된 프래그먼트를 복구하도록 청크 데이터를 프로세싱하기 위해 청크 요청에 응답하여 CM(122)에 의해 제공된 청크 데이터(예를 들면, 청크 데이터(306))를 수신한다(도 4의 블록(402)). 청크 데이터(306)로서 제공된 콘텐츠의 청크는, 예를 들면, HTTP 어드레싱 가능 파일 또는 파일의 바이트 범위를 포함할 수 있다. 예시된 실시예의 RM(121)은, CM(122)에 의해 제공된 청크 데이터가 누락 데이터인지를 결정하도록 동작한다(블록(403)). 청크 데이터가 홀들 또는 누락 데이터를 포함하지 않는다면, RM(121)은 특정 프래그먼트에 대한 청크들 모두가 수신되었는지를 결정하도록 진행하고(도 4의 블록(404)); 모든 청크들이 수신된 경우 프래그먼트를 조립하고 프래그먼트를 UA(129)에 제공하도록 진행하고(블록(405)), 프래그먼트의 청크들 모두가 수신되지 않은 경우 프래그먼트에 대한 다음의 청크 요청을 발행하도록 진행한다(블록(401)). 그러나, 청크 데이터가 홀들 또는 누락 데이터를 포함하면, RM(121)은 데이터 분석 기술들, 대역외 시그널링, CM(122)과 RM(121) 사이의 통신에서 누락 데이터의 범위를 나타내는 데이터(예를 들면, 시작 바이트 오프셋 및 바이트들의 수) 등을 사용하여 CM(122)에 의해 제공되는 청크 내에서 누락 데이터를 검출하도록 동작할 수 있다. 청크 데이터에서 누락 데이터를 검출할 때, 예시된 실시예의 RM(121)은 요청된 프래그먼트의 데이터 모두가 수신될 때까지 그 데이터를 UA(129)에 제공하는 것을 보류하고, 따라서 청크 데이터를 포함하는 프래그먼트를 UA(129)에 제공하기 전에 누락 데이터를 완성하거나 채우기 위한 동작을 구현하도록 진행한다(블록(406)).

[0074] [0078] 예를 들면, RM(121)은 늦은 데이터가 CM(122)에 의해 수신되고 후속으로 RM(121)에 제공되도록 일정 시간 기간을 대기하도록 동작할 수 있다. 부가적으로 또는 대안적으로, RM(121)은 누락 데이터에 대한 재송신을 위한 요청(예를 들면, 누락 데이터 요청(307))을 CM(122)에 제공하도록 동작할 수 있다. 실시예들에 따른 동작에서, 누락 데이터의 재송신을 위한 요청은 네트워크 상에서 이미 얻으려고 애쓰고 있는 청크 요청들 뒤에 이루어질 것이다(잠재적으로 파이프라이닝됨). 따라서, 본원에서 청크 요청에 의해 제공된 콘텐츠의 청크들의 크기는 얼마나 많은 데이터가 임의의 특정 시간에 미처리된 것일지 및 재송신을 위한 요청이 얼마나 빨리 서빙될 수 있는지를 제어하기 위해 (예를 들면, 16 킬로바이트들 내지 128 킬로바이트 범위 내에서) 비교적 작을 수 있다.

[0075] [0079] 실시예들은 앞서 말한 누락 데이터의 재송신을 위한 요청에 관련하여 우선순위화 기술을 구현할 수 있다. 예를 들면, RM(121)은, 요청 내에서 또는 그렇지 않다면 그와 연관하여(예를 들면, 대역외 시그널링), CM(122)으로 하여금, CM(122)에 의해 이미 프로세싱된 하나 이상의 청크 요청들(예를 들면, 초기 청크 요청들, 낮은 우선순위 또는 낮은 서비스 품질 스트림들과 연관된 청크 요청들 등과 같은 더 낮은 우선순위 요청들) 전에, 요청을 서버(130)로 송신하게 할지에 대한 시그널링을 CM(122)에 제공할 수 있다.

[0076] [0080] 실시예들에 따른 동작에서, RM(121)에 의해 제공된 누락 데이터 요청은 누락 데이터의 특정 바이트 범위에 대한 요청을 포함할 수 있다. 부가적으로 또는 대안적으로, 누락 데이터 요청은, 수신된 패킷들(예를 들면, 패킷(들)(304))을 제공하는데 사용되는 콘텐츠의 상이한 표현으로 제공될 수 있는 에러 정정 인코딩(예를 들면, FEC) 데이터에 대한 요청을 포함할 수 있다. 누락 데이터의 바이트 범위에 관한 정보를 대기하지 않고서(예를

들면, 누락 데이터 바이트 범위 정보를 요청하기 위한 부가적인 RTT를 회피함), 요청을 개시하는 것을 가능하게 하기 위해, 실시예들에 따라, 데이터의 원래 요청된, 비-에러 정정 인코딩 표현보다는 에러 정정 인코딩 데이터가 요청될 수 있다. 그러한 구현들은 라이브 스트리밍 상황들에서 특히 이로울 수 있고, 여기서 콘텐츠의 생성 및 클라이언트 디바이스에 의한 콘텐츠의 소비 사이에 매우 짧은 시간 지연(예를 들면, 1초)이 예상된다.

[0077] [0081] 에러 정정 인코딩 데이터가 본원의 실시예들에 따라 홀로 또는 비-에러 정정 인코딩 데이터와 결합하여 요청될 수 있다는 것이 인지되어야 한다. 에러 정정 인코딩 데이터가 비-에러 정정 인코딩 데이터와 결합하여 요청되는 경우에, 실시예들은 충분한 데이터(에러 정정 인코딩되든지, 비-에러 정정 인코딩되든지 또는 이들의 조합이든지 간에)가 데이터 복구를 위한 요청에 응답하여 수신되자마자 누락 데이터를 복구하도록 동작할 수 있다. 충분한 양의 데이터가 RM(121)에 의해 수신되자마다 누락 데이터의 복구를 가능하게 하기 위해, 이미 수신된 데이터와 결합하든지 및/또는 RM(121)에 의해 또한 요청될 수 있는 누락 데이터의 일부와 결합하든지, 누락 데이터를 복구하기 위해 적절한 양의 에러 정정 인코딩 데이터가 요청될 수 있다. 예를 들면, RM(121)은 수신된 데이터 및/또는 에러 정정 인코딩되지 않은 재-요청된 데이터에 기초하여 누락 데이터를 획득하기 위해 요청할 에러 정정 인코딩 데이터의 양을 결정하도록 동작할 수 있다.

[0078] [0082] RM(121)으로부터 누락 데이터 요청을 수신한 것에 응답하여, CM(122)은 적절한 콘텐츠에 대한 대응하는 요청(예를 들면, 누락 데이터 요청(308))을 서버(130)로 발행할 수 있다. 앞서 말한 청크 요청들과 마찬가지로, 실시예들의 CM(122)은 서버(130)에 발행된 누락 데이터 요청에 관련하여 xTCP, 표준 TCP 등과 같은 통신 프로토콜을 사용한다. CM(122)은 요청된 누락 데이터 또는 이들의 일부 부분을 포함하는 하나 이상의 데이터 패킷들(예를 들면, 패킷(들)(309))을 수신할 수 있다. CM(122)은, 앞서 논의된 바와 같이, 적절한 하나 이상의 ACK(예를 들면, ACK(들)(310))를 서버(130)로 전송하고, 데이터를 (예를 들면, 누락 데이터(311)로서) RM(121)에 제공할 수 있다.

[0079] [0083] 누락 데이터를 완성하거나 채우기 위한 동작들을 수행하면(블록(406)), 도 4에 예시된 실시예의 RM(121)은 데이터가 완전한지를 결정하도록 진행된다(블록(407)). 데이터가 완전하지 않다면(예를 들면, 데이터 내의 홀이 채워지지 않음), 예시된 실시예에 따른 프로세싱은 누락 데이터를 완성하거나 채우기 위한 추가의 동작(블록(406))을 수행하도록 복귀한다. 예를 들면, RM(121)은 추가의 누락 데이터 요청들을 전송할 수 있다. 그러나, 데이터가 완전하다고 결정되면, 예시된 실시예에 따른 프로세싱은 특정 프래그먼트에 대한 청크들 모두가 수신되었는지를 결정하도록 진행된다(블록(404)). 프래그먼트에 대한 청크들 모두가 수신된 경우(또는 FEC가 요청될 때, 청크 요청들로부터 충분한 데이터가 수신되었음), RM(121)은 프래그먼트를 조립하고(블록(405)), 프래그먼트를 (예를 들면, 프래그먼트(312)로서) UA(129)에 제공하도록 진행할 수 있다. 또한, 실시예들에 따른 동작에서, RM은, 충분한 데이터/청크들이 도착하자마자 프래그먼트를 (예를 들면, 블록(405)에서) 조립하기 시작할 수 있고, 시작으로부터 응답의 연속적인 부분이 구성될 수 있다. 실시예들의 RM은, 응답의 시작으로부터 연속적인 (즉, 데이터 내에 홀이 없음) 충분한 재구성된 데이터가 존재하자마자 부분적인 응답을 UA에 제공하도록 동작할 수 있다. 따라서, 앞서 말한 것으로부터 인지될 수 있듯이, RM(121)은 적시에 기반하여 다수의 응답들로부터 데이터를 어그리게이팅하고, UA에 의해 요청된 프래그먼트 또는 이의 연속적인 부분을 복구하기 위해 충분한 데이터가 도착하자마자 완전한 프래그먼트 응답들 또는 일부 실시예들에서 부분적은 프래그먼트 응답들을 UA(129)에 제공하도록 동작할 수 있다. 대응하여, 스트리밍 콘텐츠의 다른 프래그먼트에 대한 다음의 청크 요청이 적절할 때 발행될 수 있다(블록(401)).

[0080] [0084] 앞서 말한 예시적인 동작이 확장된 송신 프로토콜 CM 동작(예를 들면, xTCP 동작)을 제공하지만, 실시예들은 그러한 동작을 선택적으로 구현할 수 있다. 실시예들에 따른 동작에서, 특정 바이트들 또는 요청들은, 가령, 표준 TCP 동작을 사용하여 획득되도록, 확장된 송신 프로토콜을 사용하지 않고서 습득을 위해 지정될 수 있다. 예를 들면, 특정 바이트들(예를 들면, 확장된 송신 프로토콜 구현 전에 전통적으로 스트림을 설정하기 위한 스트림의 초기 프레임들, 신뢰할 수 있는 수신을 보장하기 위한 스트리밍 매니페스트 정보 등)은 TA(120)에 의한 확장된 송신 프로토콜의 구현 없이 송신을 위해 선택될 수 있다. CM(122)은, 예를 들면, 전통적인 TCP 동작과 xTCP 동작 사이에서 그의 거동을 교번하도록 동작하도록 구성될 수 있고, 가령, RM(121)에 의해 요청된 각각의 청크 내의 바이트들의 제 1 넘버(SB)에 대해, CM(122)에 의해 구현된 xTCP 수신기는 전통적인 TCP 모드에서 동작하고 이후에 xTCP 동작을 통해 공격적인 모드에서 동작한다. 공격적인 거동은 청크 요청의 완료 때까지 구현될 수 있다. 파라미터 값(SB)은 디폴트 값 0(예를 들면, 비 보존 모드)을 가질 수 있고, 각각의 청크 요청에 대해 RM(121)에 의해 공급된 값에 의해 무시될 수 있다. SB가 "-1"과 동일한 경우와 같은 특수 경우는, 전통적인 TCP로부터 완전히 신뢰할 수 있는 송신 거동이 예상된다는 것을 표시하도록 정의될 수 있다. 앞서 말한 것으로부터, 확장된 송신 프로토콜 동작의 그러한 선택적인 구현이 TA(120)를 사용하여 송신되는 복수의 콘텐츠

스트림들의 콘텐츠의 하나 이상의 스트림들에 관련한 것 및/또는 TA(120)를 사용하여 송신된 콘텐츠의 스트림의 하나 이상의 부분들에 관련한 것일 수 있다는 것이 인지되어야 한다.

[0081] [0085] 누락 데이터가 검출될 때, CM(122)의 확장된 송신 프로토콜 수신기 동작이 ACK들을 항상 발행하도록 적응된 경우에, 서버(130)는 실시예들에 따른 동작에서 분실 또는 지연된 패킷들로 인해 콘텐츠 스트림의 전송 레이트를 늦출 수 없다. 마찬가지로, 실시예들의 CM(122)은 ACK들을 전송하는데 있어서 둔화될 수 없고, 수신 윈도우를 계속해서 앞설 수 있다.

[0082] [0086] 그러나, 본원의 실시예들의 확장된 송신 프로토콜 구성들의 동작은 네트워크에서 과도한 혼잡을 발생시키지 않고서 스트리밍 콘텐츠의 고속 다운로드를 달성하는 것을 돕는다. 따라서, 확장된 송신 프로토콜 수신기 동작의 적극성은, 가령, 누락 패킷들의 수 및/또는 추정된 RTT에 기초하여 네트워크 조건들 등에 대해 튜닝될 수 있다. 그러한 확장된 송신 프로토콜 구성들을 구현하는 실시예들은 적응형 스트리밍 클라이언트 내의 애플리케이션 계층 레이트 요청 로직과 같은 레이트 적응 메커니즘을 통해 UA(129)를 적응시킬 수 있다. 부가적으로 또는 대안적으로, CM(122)은 본원에 실시예에 따른 레이트 적응 메커니즘을 통해 적응될 수 있다.

[0083] [0087] 송신 레이트 및/또는 네트워크 혼잡에 관련한 클라이언트 측 제어를 제공하도록 동작 가능한 레이트 적응 메커니즘을 제공하는데 있어서, 실시예들의 CM(122)은, 누락 데이터가 검출될 때 항상 ACK들을 발행하지는 않는 확장된 송신 프로토콜 수신기(예를 들면, xTCP 수신기) 구성을 구현하도록 적응될 수 있다. 예를 들면, CM(122)은 네트워크 조건들 또는 다른 인수들이 서버(130)에 의한 데이터의 전송이 (예를 들면, 네트워크 혼잡을 회피하거나 덜해주기 위해) 둔화되어야 한다는 것을 표시하는지를 결정하기 위한 로직을 구현할 수 있고, 이로써 데이터 전송 레이트 제어가 수신기에 의해 제공되는 실시예들을 제공한다. 그러한 실시예들에서, xTCP 수신기 동작은 콘텐츠 스트림 전송 레이트들을 제어하기 위한 적절한 시그널링을 제공하기 위해(예를 들면, 중복 ACK들을 선택적으로 제공하기 위해) TCP 송신기의 거동(예를 들면, 중복 ACK들을 수신하는 것에 대한 반응)의 지식을 레버리지(leverage)할 수 있다. 그러한 xTCP 수신기의 적극성은 누락 패킷들의 수, 추정된 RTT 등에 기초하여 튜닝될 수 있다.

[0084] [0088] 부가적으로 또는 대안적으로, CM(122)의 확장된 송신 프로토콜 수신기가 데이터 전송 레이트 제어를 제공하도록 적응되는 경우에, 확장된 송신 프로토콜 수신기가 누락 패킷들 및/또는 RTT 변형들을 발견할 때, CM(122)은, 네트워크에서 혼잡을 제어하기 위해 (예를 들면, 더 적은 요청된 데이터가 네트워크에서 미처리되도록 수신 윈도우 크기를 감소시키기 위해) 수신 윈도우 크기를 조절하면서, ACK들을 수신된 마지막 바이트(예를 들면, LastByteReceived(220))까지 전송하도록 동작할 수 있다. 수신 윈도우 스케일링은, 가령, 수신 윈도우 크기를 전통적인 TCP, RTT 등에 의해 생성될 추정된 혼잡 윈도우 크기에 기초하기 위한 공식에 따라 이루어질 수 있다. 동작 시에, 수신 윈도우는, 누락 데이터가 없는 데이터 패킷들을 연속적으로 수신하는 것에 응답하여 증가될 수 있고, 반면에 수신 윈도우는 (예를 들면, 미리 결정된 윈도우의 시간 내에) 더 많은 분실 패킷들을 검출한 것에 응답하여 추가로 감소될 수 있다. 따라서, 수신 윈도우는, 네트워크에서 상당한 혼잡을 도입하지 않고서 올바른 레벨의 적극성으로 전송기 거동을 제어하는데 사용될 수 있다. 수신 버퍼 크기의 변화가 수신 윈도우 조절과 연관하여 구현될 수 있지만, 본원에 실시예들에 따라 구현된 수신 윈도우 조절은 수신 버퍼 크기의 변화를 사용하지 않지만, 대신에 전송기에 수신 윈도우 조절을 통지하도록 동작할 수 있다. 이를 위해, CM(122)의 확장된 송신 프로토콜 수신기는, 수신 버퍼 크기의 대응하는 변화를 구현하지 않고서, 원하는 제어를 구현하기 위해 조절된 수신 윈도우 크기를 서버(130)로 전송할 수 있다.

[0085] [0089] 앞서 설명된 TA(120)의 실시예들이 확장된 송신 프로토콜(예를 들면, 예시적인 실시예들의 xTCP)을 사용하여 원하는 콘텐츠의 프래그먼트들 또는 프래그먼트들의 시퀀스들의 개선된 전달을 제공하도록 적응되지만, 콘텐츠의 개선된 전달은 다른 구현들 및/또는 기술들을 사용하여 본원의 개념들에 따라 송신 가속기에 의해 제공될 수 있다. 예를 들면, TA(120)의 실시예들은 CM 구성(예를 들면, CM-yTCP)이 제공될 수 있는 비수정된 송신 프로토콜을 포함할 수 있고, 여기서 CM은 비수정된 의미론(semantics)을 갖는 TCP(본원에서 yTCP로 지칭됨)뿐만 아니라 HTTP를 사용한다. 실시예들의 yTCP 구성에서, TCP의 오버-디-와이어(over-the-wire) 프로토콜은 수정되지 않지만, CM은 비순차적인 수신된 데이터 스트림을 엿보고(peek) 이를 RM으로 전달하도록 적응되고, 차례로 RM은, UA가 비순차적인 데이터 전달을 처리하도록 장착되면, 수신된 세그먼트들을 비순차적으로 UA로 전달할 수 있다. 실시예들의 yTCP 동작에 따라 비순차적으로 전달되는 데이터를 포함하는 TCP 스택의 버퍼를 CM이 엿보고 이를 RM으로 전달하는 것을 통해, 세그먼트가 분실되면, 단지 그 세그먼트는 지연을 갖고 RM에 도착할 것이고, 한편 이것 다음에 비순차적인 세그먼트들이 RM에서 제시간에 도착하고, 이러한 비순차적인 세그먼트들은 그들이 RM에 도착할 때 RM으로부터 UA로 전달될 수 있다. yTCP를 구현하는 그러한 실시예들 자체가 정상 TCP 혼잡 제어에 영향을 받을 수 있지만, yTCP의 구현은 네트워크 혼잡 및 따라서 패킷 분실 가능성 자체를 과장할 가능성

이 없다. 따라서, CM-yTCP 구성은, 전체 수신 레이트를 증가시키지 않고서, 가끔의 패킷 분실들로 인해 (비순차적으로 수신된 데이터에 앞서 분실된 데이터의 도착을 대기하는 대신에, 비순차적으로 빠르게 수신된 데이터를 RM 및/또는 UA에 제공함으로써) 실속(stalling) 가능성을 감소시키는데 사용될 수 있다.

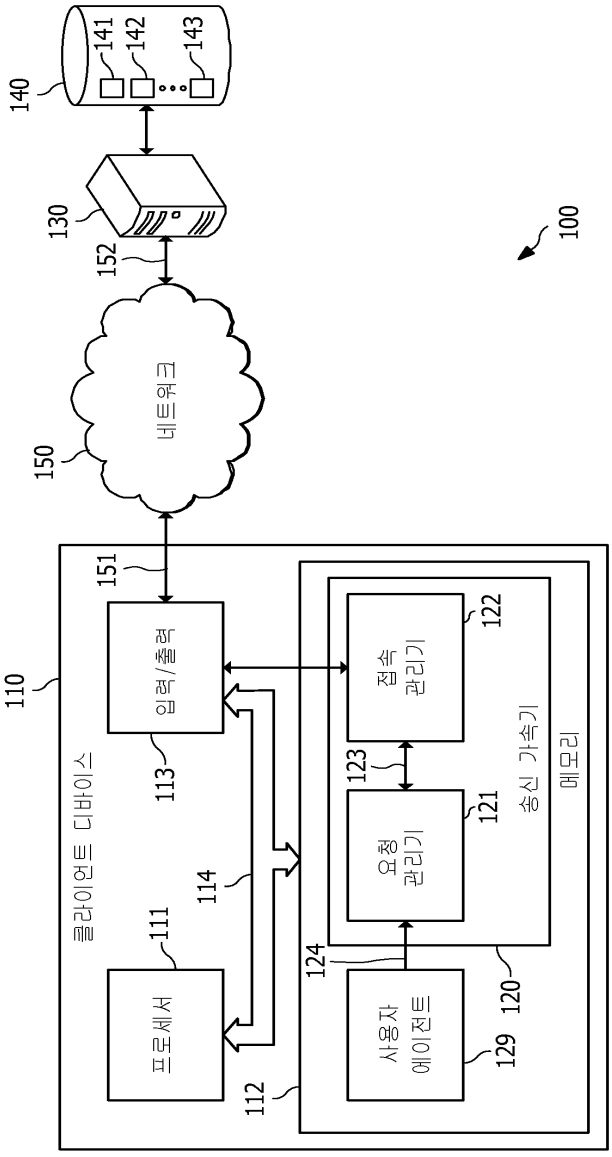
[0086] [0090] 실시예들의 yTCP의 구현에서, CM(122)의 yTCP 수신기는, 수신기가 비순차적으로 수신하고 따라서 정상적으로 UA(129)로 비순차적으로 전달되지 않는 세그먼트들을 엿보고 이를 RM(121) 및 UA(129)로 전달하도록 적응된다. 비수정된 TCP 동작에서, 패킷 분실이 발생하는 경우에, 분실된 세그먼트가 전송기에 의해 재송신되고 클라이언트에 의해 수신되지 않는다면 그리고 수신될 때까지, 클라이언트는 보통 분실된 세그먼트 이후에 임의의 데이터를 수신하지 않는다는 것이 인지되어야 한다. 따라서, 지연을 갖고 수신되는 데이터의 양이 꽤 상당할 수 있다. 예를 들면, 데이터가 2 Mbits/초의 레이트로 수신되고, RTT가 200ms이면, 대략 40KB의 비순차적인 데이터가 수신된 후에, 재송신된 세그먼트가 수신될 것이다. 이러한 데이터의 분실은 이러한 40KB의 데이터가 지연(예를 들면, 앞서 말한 예에서, 단일 지연된 세그먼트보다 약 20 배 데이터가 지연됨)을 갖고 전달되게 할 것이다. 또한, 실제로, RTT는 전형적으로 큐잉 지연으로 인해 몇 초 정도로 성장할 것이고, 따라서 단일 패킷 분실은 매우 긴 시간 동안 접속을 지연시킬 수 있다. CM(122)이 모든 수신된 세그먼트들을 포함하는 TCP를 수신 버퍼를 엿보고 세그먼트들을 RM(121)으로 전달하는 것을 통해 획득된 정보는 (세그먼트들이 순차적으로 수신되든지 또는 비순차적으로 수신되든지 간에) 패킷 분실들을 완화하기 위한 하나 이상의 기술들을 구현하는데 사용될 수 있다.

[0087] [0091] 아마도 하나 이상의 다른 기술들과 관련하여 yTCP 동작의 구현은 앞서 말한 문제들을 완화하는데 사용될 수 있다. 예를 들면, RM(121)은 각각의 프래그먼트 요청에 대한 일부 FEC 데이터를 요청하도록 동작할 수 있다. 동작 시에, RM(121)은 RM에 의해 처리될 수 있는 에러 가능성(예를 들면, 1%)을 가정하고, 더 많은 세그먼트들이 분실되지 않는다면, 높은 정정 가능성을 가능하게 하는 일정량의 FEC 데이터를 요청할 수 있다. 부가적으로 또는 대안적으로, RM(121)은 이력 정보로부터 에러 가능성을 추정할 수 있다(예를 들면, TCP 전달에서 홀들을 보는 yTCP 동작에 기초하여 에러 가능성을 결정함). 그러한 FEC 데이터는, 대역폭 활용도를 수용불가하게 증가시키지 않고 누락 데이터를 복구하는데 사용될 수 있다. 그러한 CM-yTCP 구성은 지연된 데이터의 양 및 요청 및 송신되는 FEC 데이터의 양 둘 모두를 감소시키는 것을 가능하게 할 수 있다.

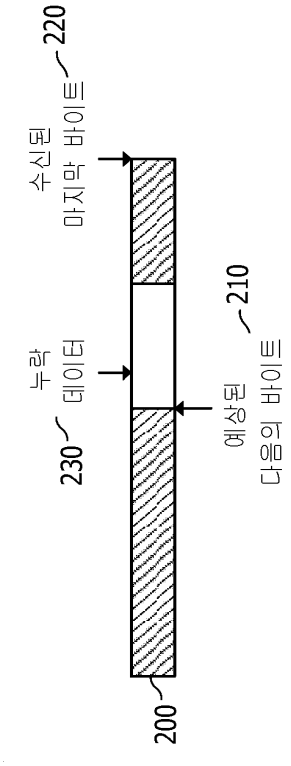
[0088] [0092] 선택된 양상들이 상세하게 예시 및 설명되었지만, 다양한 치환 및 변경들이 이하의 청구항들에 의해 정의된 대로 본 발명의 사상 및 범위로부터 벗어나지 않고 이 양상들에서 행해질 수 있음이 이해될 것이다.

도면

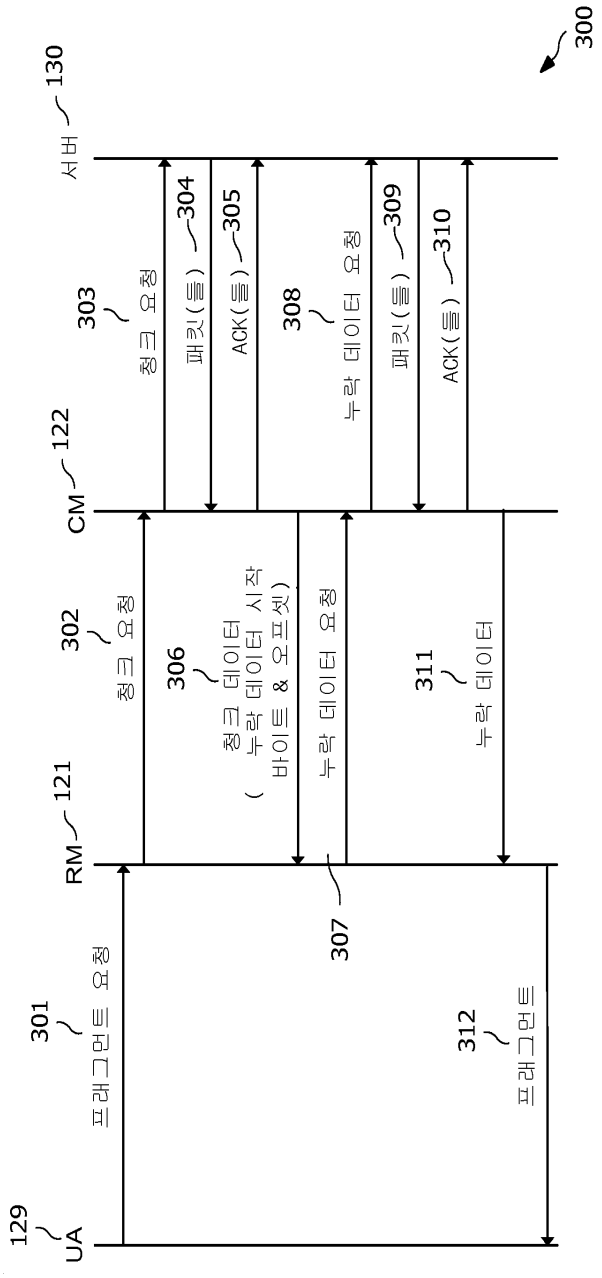
도면1


140
141
142
143
130
152
150
네트워크
100
110
클라이언트 디바이스
151
입력/출력
113
114
111
프로세서
112
124
사용자 에이전트
129
120
123
요청 관리기
121
접속 관리기
122
송신 가속기
메모리

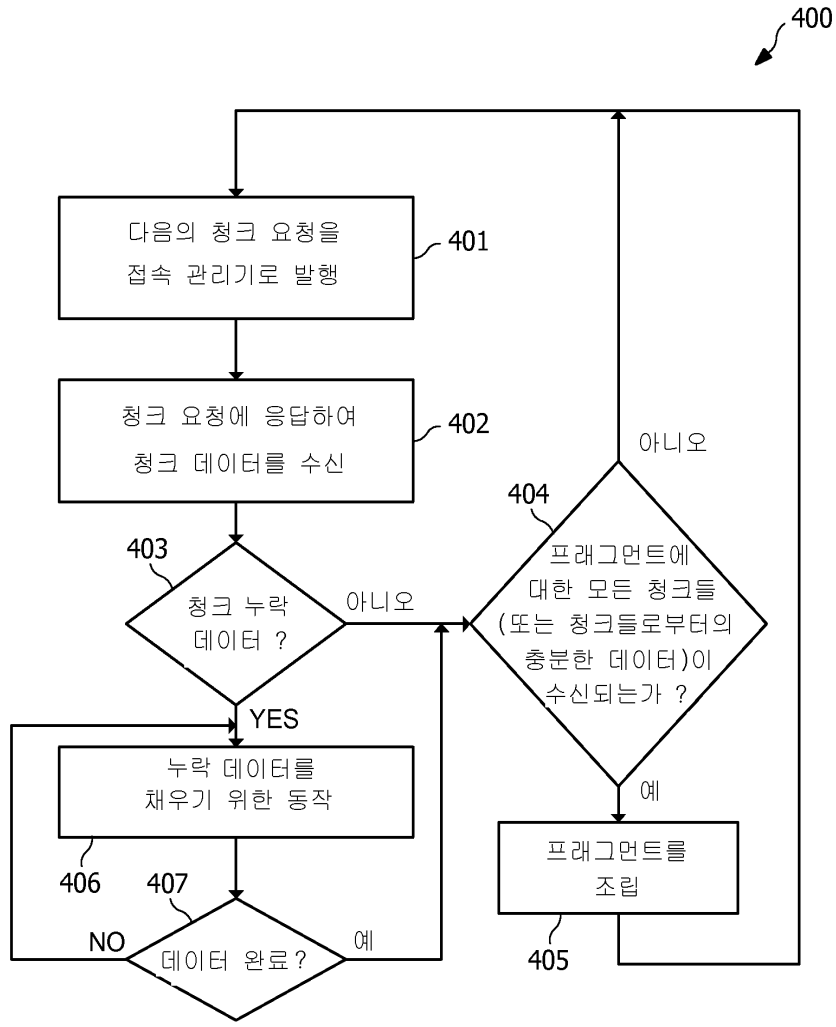
도면2



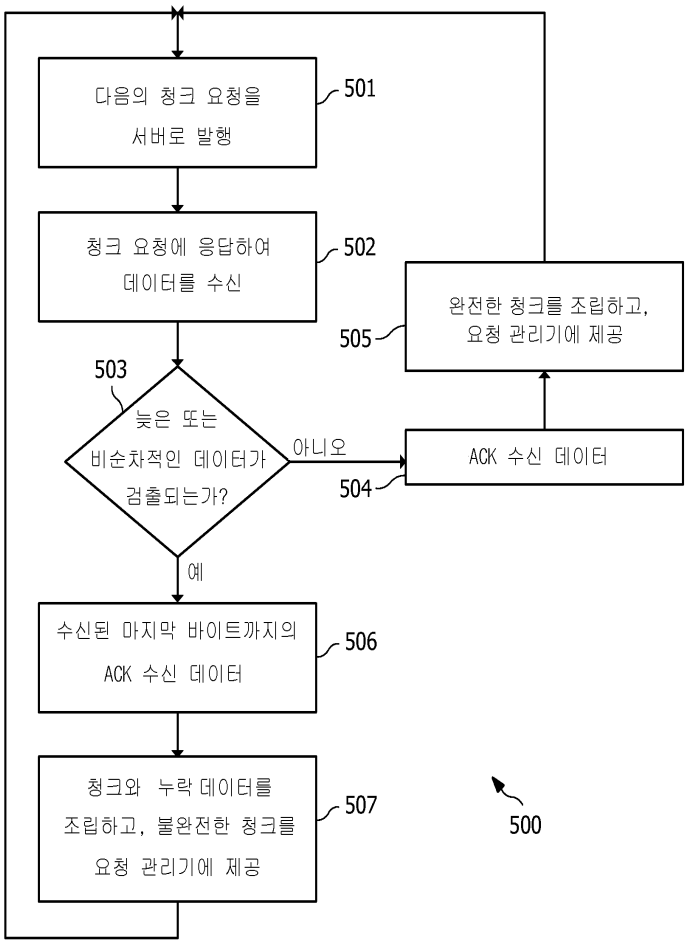
도면3


300
서버 130
CM 122
RM 121
UA 129
303
청크 요청
304
패킷(들)
305
ACK(들)
308
누락 데이터 요청
309
패킷(들)
310
ACK(들)
302
청크 요청
306
청크 데이터
(누락 데이터 시작
바이트 & 오프셋)
307
누락 데이터 요청
311
누락 데이터
301
프래그먼트 요청
312
프래그먼트

도면4


400
다음의 청크 요청을
접속 관리기로 발행
401
청크 요청에 응답하여
청크 데이터를 수신
402
403
청크 누락
데이터 ?
아니오
YES
404
프래그먼트에
대한 모든 청크들
(또는 청크들로부터의
충분한 데이터)이
수신되는가 ?
아니오
예
누락 데이터를
채우기 위한 동작
406
407
NO
데이터 완료?
예
프래그먼트를
조립
405

도면5


다음의 청크 요청을
서버로 발행
501
청크 요청에 응답하여
데이터를 수신
502
완전한 청크를 조립하고,
요청 관리기에 제공
505
503
늦은 또는
비순차적인 데이터가
검출되는가?
아니오
ACK 수신 데이터
504
예
수신된 마지막 바이트까지의
ACK 수신 데이터
506
청크와 누락 데이터를
조립하고, 불완전한 청크를
요청 관리기에 제공
507
500