



(19) 대한민국특허청(KR)

(12) 등록특허공보(B1)

(45) 공고일자 2014년02월26일

(11) 등록번호 10-1367886

(24) 등록일자 2014년02월20일

(51) 국제특허분류(Int. Cl.)

H04N 7/64 (2006.01) H04N 7/015 (2006.01)

(21) 출원번호 10-2010-7027519

(22) 출원일자(국제) 2009년05월07일

심사청구일자 2010년12월07일

(85) 번역문제출일자 2010년12월07일

(65) 공개번호 10-2011-0015615

(43) 공개일자 2011년02월16일

(86) 국제출원번호 PCT/US2009/043184

(87) 국제공개번호 WO 2009/137705

국제공개일자 2009년11월12일

(30) 우선권주장

61/051,325 2008년05월07일 미국(US)

(56) 선행기술조사문헌

US7853856 B2

US7843947 B2

DVB Application. Layer FEC Evaluations and DVB. A Document A115 TM May, 3783-2007.

US20060075321 A1*

*는 심사관에 의하여 인용된 문헌

(73) 특허권자

디지털 파운튼, 임크.

미국 92121-1714 캘리포니아주 샌 디에고 모어하우스 드라이브 5775

(72) 발명자

루비, 마이클, 지.

미국 92121-1714 캘리포니아 샌디에고 모어하우스 드라이브 5775

스톡함퍼, 토마스

미국 92121-1714 캘리포니아 샌디에고 모어하우스 드라이브 5775

쇼크롤라히, 모함매드, 아민

미국 92121-1714 캘리포니아 샌디에고 모어하우스 드라이브 5775

(74) 대리인

남상선, 특허법인 남앤드남

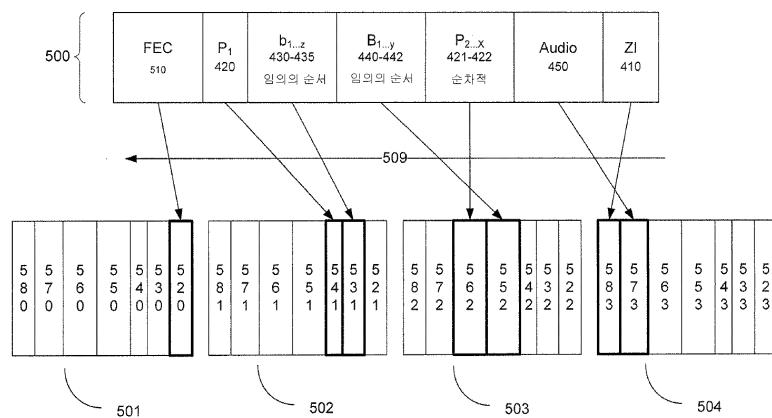
전체 청구항 수 : 총 20 항

심사관 : 김영태

(54) 발명의 명칭 브로드캐스트 채널 상에서의 고속 채널 재핑 및 고품질 스트리밍 보호

(57) 요 약

다수의 물리층 블록들 내의 소스 블록들의 송신의 시그널링은, 심볼들이, 심볼들이 생성되는 소스 블록들 및 소스 블록들에 대하여 우선순위화된 데이터의 송신 및 표시들에 대해 어떻게 관련되는지를 시그널링하여, 물리층 블록 내에서 인터리빙된 소스 블록들을 시그널링하기 위해 최소의 추가적인 오버헤드를 사용하여, 일부 경우들에서는 오버헤드 없이, 스트리밍 및 객체 전달 애플리케이션들 모두에 대해 이루어진다. 하나 이상의 채널들에 대한 스트리밍들의 정렬 및 송신은 전달되는 스트리밍들의 품질을 개선하면서, 채널 자원들 및 수신기 전력 자원들이 요구되는 양을 최소화시거나 개선시키기 위해 이루어진다.

대 표 도

특허청구의 범위

청구항 1

브로드캐스트 채널 상에서 데이터 스트림들을 전달하기 위한 전자 전달 시스템으로서,

상기 브로드캐스트 채널은 하나 이상의 소스들로부터 복수의 수신기들로 신호들을 전송하기 위한 채널이며,

각각의 수신기는 동일한 신호를 수신하려고 시도하며,

상기 브로드캐스트 채널 상에서 데이터 스트림들을 전달하기 위한 전자 전달 시스템은,

물리층 블록들의 물리층 패킷들 내에서 상기 데이터 스트림에 대한 데이터를 송신하는 송신기 시스템 – 상기 데이터는 우선순위 레벨에 따라 정렬된 상기 데이터 스트림의 블록들 및 서브-블록들이고, 소스 블록의 적어도 하나의 낮은 우선순위 서브-블록은 상기 소스 블록의 적어도 하나의 높은 우선순위 서브-블록의 이전에 송신됨 – 을 포함하고,

상기 송신되는 데이터가 상기 데이터 스트림과 어떻게 관련되는지에 대한 표시들은 적어도 부분적으로 상기 물리층 블록들로부터 얻을 수 있는,

전자 전달 시스템.

청구항 2

제1항에 있어서,

상기 송신되는 데이터가 상기 데이터 스트림과 어떻게 관련되는지에 대한 표시들은 적어도 부분적으로 상기 물리층 블록들의 헤더들 내의 정보로부터 얻을 수 있고,

상기 송신기 시스템은 상기 표시들을 포함하도록 상기 물리층 블록들의 헤더들을 구성하는,

전자 전달 시스템.

청구항 3

제1항에 있어서,

상기 송신되는 데이터가 상기 데이터 스트림과 어떻게 관련되는지에 대한 표시들은,

적어도 부분적으로 상기 물리층 패킷들의 헤더들 내의 정보로부터 얻을 수 있는,

전자 전달 시스템.

청구항 4

제1항에 있어서,

상기 송신되는 데이터는 데이터의 소스 블록들 내에서 심볼들로 편성되고(organized),

상기 표시들은 심볼이 소스 블록으로부터 어떻게 생성되는지에 대한 표시들 및 심볼과 소스 블록 간의 연관의 표시들을 포함하는,

전자 전달 시스템.

청구항 5

제4항에 있어서,

상기 표시들은 인코딩 심볼 식별자들이고,

상기 인코딩 심볼 식별자들은 물리층 블록들의 헤더들에서 적어도 부분적으로 전달되는,

전자 전달 시스템.

청구항 6

제4항에 있어서,

상기 표시들은 인코딩 심볼 식별자들이고,

상기 인코딩 심볼 식별자들은 제어 데이터 채널에서 전달되는,

전자 전달 시스템.

청구항 7

제4항에 있어서,

심볼들과 소스 블록들 간의 상기 연관은 물리층 블록들의 헤더들로부터 결정될 수 있는,

전자 전달 시스템.

청구항 8

제4항에 있어서,

상기 송신되는 데이터의 심볼들은 소스 블록들로부터 생성되는 FEC 복원(repair) 데이터를 포함하는,

전자 전달 시스템.

청구항 9

제4항에 있어서,

둘 이상의 논리적 데이터 스트림은 물리층 블록들의 단일 스트림 내에서 송신되는,

전자 전달 시스템.

청구항 10

제4항에 있어서,

상기 송신되는 데이터의 심볼들은 둘 이상의 물리층 블록들의 스트림에 걸쳐 송신되는,

전자 전달 시스템.

청구항 11

제4항에 있어서,

상기 송신되는 데이터의 심볼들이 상기 스트림 또는 객체 데이터와 어떻게 관련되는지에 대한 표시들은,

상기 송신되는 데이터의 심볼들을 전달하는 물리층 패킷들에서 적어도 부분적으로 전달되는,

전자 전달 시스템.

청구항 12

제4항에 있어서,

상기 소스 블록에 대해 전송되는 데이터는 상이한 우선순위들의 상이한 서브-블록들로 편성되는,

전자 전달 시스템.

청구항 13

제12항에 있어서,

상기 소스 블록의 서브-블록 구조의 표시들은 물리층 블록들의 헤더들로부터 결정되는,

전자 전달 시스템.

청구항 14

제12항에 있어서,

상기 소스 블록의 서브-블록 구조의 표시들은 물리층 블록들에서 전달되는 물리층 패킷들의 헤더들로부터 결정될 수 있는,

전자 전달 시스템.

청구항 15

제12항에 있어서,

상기 송신되는 데이터의 심볼들은 상이한 서브-블록들 및 서브-블록들의 조합들로부터 생성되는 FEC 복원 데이터를 포함하는,

전자 전달 시스템.

청구항 16

제12항에 있어서,

우선순위들의 상기 서브-블록들은 상기 서브-블록들의 송신 순서를 결정하는데 사용되는,

전자 전달 시스템.

청구항 17

제12항에 있어서,

우선순위들의 상기 서브-블록들은 상기 서브-블록들을 상기 물리층 블록들로 매핑시키는데 사용되는,

전자 전달 시스템.

청구항 18

제17항에 있어서,

상기 물리층 블록들로 매핑되는 상기 우선 순위들의 상기 서브-블록들은 상이한 물리층 블록들 사이에서 분할되는,

전자 전달 시스템.

청구항 19

브로드캐스트 채널 상에서 데이터 스트림들을 전달하기 위한 전자 전달 시스템에서 송신기로부터 수신기로 데이터를 전송하기 위한 방법으로서,

상기 브로드캐스트 채널은 하나 이상의 소스들로부터 복수의 수신기들로 신호들을 전송하기 위한 채널이며,

각각의 수신기는 동일한 신호를 수신하려고 시도하며,

상기 브로드캐스트 채널 상에서 데이터 스트림들을 전달하기 위한 전자 전달 시스템에서 송신기로부터 수신기로 데이터를 전송하기 위한 방법은,

상기 송신기로부터 물리층 패킷 블록들의 물리층 패킷들 내에서 상기 데이터 스트림에 대한 데이터를 송신하는 단계 – 상기 데이터는 우선순위 레벨에 따라 정렬된 상기 데이터 스트림의 블록들 및 서브-블록들이고, 소스 블록의 적어도 하나의 낮은 우선순위 서브-블록은 상기 소스 블록의 적어도 하나의 높은 우선순위 서브-블록의 이전에 송신됨 – 를 포함하며,

상기 송신되는 데이터가 상기 데이터 스트림과 어떻게 관련되는지에 대한 표시들은 적어도 부분적으로 상기 블록들의 물리층으로부터 얻을 수 있는,

데이터 전송 방법.

청구항 20

제19항의 방법을 실행하기 위한 컴퓨터-실행가능한 코드를 포함하는,
컴퓨터-판독가능한 매체.

명세서

기술분야

[0001]

본 발명은 일반적으로는 스트리밍 및 객체 전달에 관한 것이고, 구체적으로는 전달되는 스트림의 품질을 보호하기 위한 FEC를 사용하는 덜 신뢰성 있는 채널들 상에서의 스트리밍 및 객체 전달에 관한 것이다.

배경기술

[0002]

본원은 출원번호가 61/051,325이고 발명의 명칭이 "Fast Channel Zapping and High Quality Streaming Protection over a Broadcast Channel"이고 출원일이 2008년 5월 7일인 미국 가출원의 이익을 청구한다.

[0003]

스트리밍 데이터, 통상적으로는 오디오 및/또는 비디오 데이터 뿐만 아니라 텔레메트릭 데이터와 같은 다른 데이터 타입들을, 채널을 통해 송신하는 것을 고려하는 것은 일반적인 구현이 되었다. 한 가지 주요 관심사는, 전달되는 데이터의 품질이 높다는 것, 예를 들어, 오리지널 스트림 데이터의 전부 또는 대부분이 수신기 또는 수신기들의 세트로 전달됨을 보장하는 것이다. 예를 들어, 스트리밍 데이터가 전달되고 있는 채널은 완전히 신뢰 가능하지 않을 수 있는데, 예를 들어, 상기 데이터의 일부분은 전송 시 유실 또는 손상된다. 따라서, 높은 품질의 전달을 달성하기 위해 전달 품질 저하를 극복하기 위해 다른 측정들이 취해질 필요가 있는 경우가 종종 있으며, 여기서, 이러한 측정들은 예를 들어, 패킷 손상으로부터 보호하기 위해 물리층에서 또는 링크에서의, 패킷 손상으로부터 보호하기 위해 전송 또는 애플리케이션 층에서의 오리지널 데이터 스트림에 대한 FEC의 애플리케이션을 포함할 수 있다. 다른 측정들은, 유실된 또는 손상된 데이터를 재전송하기 위한 재전송 전략, 예를 들어 링크층 재전송 프로토콜 또는 애플리케이션 층 재전송 프로토콜의 사용을 포함한다.

[0004]

이러한 시스템의 설계시의 또 다른 주요 관심사는 예를 들어, 비디오 스트림의 시청을 시작하기 위해 최종 사용자가 먼저 요청하는 시간으로부터 비디오 스트림의 시청을 시작하는데 걸리는 시간량, 또는 현재 비디오 스트림의 시청을 중지하고 사용자 요청에 의해 트리거링되는 새로운 비디오 스트림의 시청을 시작하는데 걸리는 시간량이다. 이러한 시간량은 종종 채널 재핑(zapping) 시간이라 지칭된다. 통상적으로, 채널 재핑 시간이 작아질수록 최종 사용자 경험이 더 나아지며, 결과적으로 전체 서비스가 더 가치있게 된다. 예를 들어, 채널 재핑 시간이 가능한 작게, 예를 들어, 1초 미만이 되는 것이 요구조건이 된다.

[0005]

스트림들이 백채널 없이 매우 신뢰성 있는 채널들을 통해 전달될 때, 또는 상기 스트림들이 덜 신뢰성 있는 채널들을 통해 전달될 때 그리고 유실된 데이터의 재전송을 요청하기 위해 사용될 수 있는 백채널이 존재할 때 이러한 채널 재핑 시간들 및 고품질의 스트림 전달을 달성하는 것이 종종 가능하지만, 상기 스트림들이 덜 신뢰성 있는 채널들을 통해 전달될 때 그리고 백채널이 신뢰성을 향상시키기 위해 사용되지 않을 때 이러한 채널 재핑 시간들을 달성하는 것은 종종 도전과제가 되며, 대신 FEC의 사용이 적절할 수 있다.

[0006]

최근, 전송동안 스트리밍 매체의 보호를 위해 FEC들의 사용을 고려하는 것이 일반적인 구현이 되었다. 패킷 네트워크 – 상기 패킷 네트워크들의 예들은 인터넷 및 3GPP, 3GPP2 및 DVB와 같은 그룹들에 의해 표준화된 무선 네트워크들을 포함함 – 를 통해 송신될 때, 소스 스트림은 그것이 생성되거나 사용가능하게 될 때 패킷들 내에 위치되며, 따라서 상기 패킷들은 생성되거나 사용가능하게 된 순서로 상기 소스 스트림을 수신기들로 전달하는데 사용된다. 이러한 타입들의 시나리오에 대한 FEC 코드들의 통상적인 애플리케이션에 있어서, 상기 FEC 코드는 소스 스트림을 포함하는 오리지널 소스 패킷들에 추가적인 복원 패킷들을 추가하기 위해 사용되며, 이를 복원 패킷들은 소스 패킷 유실이 발생할 때 수신된 복원 패킷들이 상기 유실된 소스 패킷들 내에 포함된 데이터를 복원하는데 사용될 수 있다는 특징을 가진다. 다른 예들에서, 부분적인 패킷 유실이 발생할 수 있는데, 즉, 수신기들이 패킷의 다른 부분들을 수신하는 동안 상기 패킷의 일부분을 유실할 수 있으며, 따라서 이들 예들에서, 전체적으로 또는 부분적으로 수신된 복원 패킷들이 전체적으로 또는 부분적으로 유실된 패킷들을 복원하는데 사용될 수 있다. 다른 예들에서, 다른 타입들의 손상이 상기 송신된 데이터에 대해 발생할 수 있는데, 예를 들어, 비트들의 값들이 플립(flip)될 수 있으며, 따라서, 복원 패킷들은 이러한 손상을 보정하기 위해 사용될 수 있으며, 소스 패킷들의 가능한 정확한 복원을 제공할 수 있다. 다른 예들에서, 소스 스트림은 이산 패킷들

로 반드시 송신되는 것이 아니라, 대신 예컨대 연속적인 비트 스트림으로서 송신될 수 있다.

[0007] 소스 스트림의 보호를 제공하기 위해 사용될 수 있는 FEC 코드들의 많은 예들이 존재한다. 리드-솔로몬(Reed-Solomon) 코드들은 통신 시스템들에서 에러 및 삭제 보정을 위해 잘 알려져 있는 코드들이다. 예를 들어, 패킷 데이터 네트워크들에 대한 삭제 보정에 대해, 리드-솔로몬 코드들의 잘 알려져 있는 효율적인 구현예는 L.Rizzo 저, "Effective Erasure Codes for Reliable Computer Communication Protocols", Computer Communication Review, 27(2):24-36 (April 1997) (이하, "Rizzo") 및 J. Bloemer, M.Kalfane, R. Karp, M.Karpinski, M.Luby, D.Zuckerman 저, "An XOR-Based Erasure-Resilient Coding Scheme", Technical Report TR-95-48, International Computer Science Institute, Berkeley, California, (1995) (이하, "XOR-Reed-Solomon")에 기재된 바와 같은 Cauchy 또는 Vandermonde 행렬들을 사용하는 것이다. FEC 코드들의 다른 예들은, 예컨대 미국 특허 제 6,307,487호 (이하, "Luby I") 및 미국 공개 공보 제 2003/0058958호 (이하, "Shokrollahi I")에 각각 기재된 LDPC 코드들, 체인 반응 코드들 및 멀티-스테이지 반응 코드들을 포함하며, 이들은 모든 용도로 여기에 통합된다.

[0008] 리드-솔로몬 코드들의 변형들에 대한 FEC 디코딩 프로세스의 예들은 "Rizzo" 및 "XOR-Reed-Solomon"에 기재된다. 이들 예들에서, 디코딩은 충분한 소스 및 복원 데이터 패킷들이 수신된 경우 적용된다. 디코딩 프로세스는 계산상 힘들고(computationally intensive) 사용가능한 CPU 자원들에 의존적이므로, 이는 블록 내에서 매체에 의해 걸쳐져 있는 시간 길이에 비해 완료하는데 상당한 시간이 소모될 수 있다.

[0009] 많은 애플리케이션들에서, 패킷들은 FEC 프로세스가 적용되는 심볼들로 추가적으로 세분된다. 심볼들은 임의의 사이즈를 가질 수 있지만, 종종 심볼의 사이즈는 기껏해야 상기 패킷의 사이즈와 동일하다. 이하, 인코딩 블록을 포함하는 심볼들을 "소스 심볼들"이라 지칭하고, FEC 프로세싱동안 생성된 심볼들을 "인코딩 심볼들"이라 지칭한다. 일부 FEC 코드들, 특히 리드-솔로몬 코드들에 대해, 인코딩 및 디코딩 시간은 소스 블록당 인코딩 심볼들의 수가 커짐에 따라 구현불가능하게 커진다. 따라서, 실제로, 소스 블록당 생성될 수 있는 인코딩 심볼들의 전체 수에 대한 상한 경계, 예를 들어, 255가 종종 존재한다. 심볼들이 상이한 패킷 페이로드들로 종종 위치되므로, 이는 때때로 소스 블록의 인코딩 시 최대 길이에 대한 구현가능한 상한 경계를 두는데, 예를 들어, 패킷 페이로드가 최대 1024 바이트인 경우 인코딩된 소스 블록은 최대 255KB(킬로바이트)가 될 수 있으며, 또한 이는 물론 각각의 심볼이 개별 패킷으로 전송되는 경우 소스 블록 자체의 사이즈에 대한 상한 경계이다.

[0010] 긴 시간 기간에 걸쳐 확산되어 송신되는 스트림 내의 데이터의 블록들에 대해 FEC 인코딩 및 디코딩을 적용하는 것이 종종 바람직한데, 왜냐하면 더 큰 시간 구간 동안 송신되는 데이터의 블록들에 대해 FEC 코딩을 적용하는 것은 일반적으로 더 작은 시간 구간 동안 송신되는 데이터 블록들에 대한 FEC 코딩과 동일한 대역폭 오버헤드에 대한 더 양호한 보호를 상기 스트림에 제공하기 때문이다. 이는 많은 채널들이 시간 상관된 유실 및/또는 손상 특성들에 관련하기 때문이다. 예를 들어, 데이터는 버스트로 송신될 수 있거나, 또는 채널 특성들이 그들이 다른 짧은 시간 구간들 동안 존재하는 것보다 더 열악한 경우 짧은 시간 기간들이 존재한다.

[0011] 더 긴 시간 기간 동안 확산되어 송신되는 데이터 블록들에 적용되는 FEC 인코딩을 사용하는 것의 도전과제는 이것이 채널 재.FlatStyle 시간에 악영향을 줄 수 있다는 점이다. 예를 들어, 수신기에서, 인코딩된 데이터의 블록은 전체 블록에 대한 충분한 데이터가 수신된 이후에만 완전히 복원되고 재생(play out)될 수 있다. 따라서, FEC 인코딩된 데이터의 블록이 더 긴 시간 구간 동안 송신되는 경우, 채널 재.FlatStyle 시간은 수용불가능하게 커질 수 있다.

[0012] 더 긴 시간 구간 동안 FEC 인코딩된 데이터의 블록을 송신하는 동시에 짧은 채널 재.FlatStyle 시간의 목적을 달성하기 위한 한가지 방법은 FEC 인코딩된 데이터 중 가장 중요한 데이터가 마지막에 송신되고 가장 덜 중요한 데이터가 먼저 송신되는 방식으로 상기 데이터를 정렬(order)하는 것이다. 예를 들어, 출원번호가 11/423,391이고 발명의 명칭이 "Forward Error Correcting (FEC) Coding and Streaming"(이하, "FEC Streaming")이고 모든 목적을 위해 여기에 포함되는 미국 특허출원은 소스 블록에 대한 소스 데이터를 수신하기 전에 FEC 복원 데이터를 송신하고, 이에 의해 수신기로 하여금 상기 소스 블록에 대한 소스 데이터의 일부분을 수신하게 하고, 상기 수신기가 소스 블록의 중간에서 상기 스트림을 조인(join)시킨다 할지라도 재생을 위해 미디어 플레이어로 그것의 송신을 시작하게 하여, 결과적으로 채널 재.FlatStyle 시간을 최소화하기 위한 방법들을 기술한다.

[0013] 또 다른 관심사는 송신될 실제 데이터를 식별하는데 사용되는 헤더 데이터에 의해 사용되는 채널 자원들의 양을 최소화하는 것이다. 일반적으로, 헤더 데이터는 데이터를 전달하기 위해 사용가능한 용량에 부정적으로 영향을 미치는 오버헤드이다. 예를 들어, 헤더 데이터 중 4바이트가 실제 데이터 중 각 100바이트들을 식별하는데 사용되는 경우, 헤더 오버헤드는 유효(significant) 4%이다. 특히 스트리밍 및 객체 전달 애플리케이션들 모두에 대해, 그러나 일반적으로는 임의의 데이터 전달 애플리케이션에 대해, 가능한 많은 헤더 오버헤드를 최소화

하는 것이 바람직하다.

[0014] 짧은 채널 재평 시간들이 요구될 때 신뢰성을 향상시키기 위해 백채널들이 사용되지 않는 경우, 덜 신뢰성 있는 채널들을 통해 고품질 스트림이 전달되도록 하는 방법들, 프로세스들 및 장치가 바람직하다. 주어진 신뢰성 레벨, 예를 들어, 헤더 오버헤드들 및 FEC 오버헤드들을 달성하기 위한 물리적 자원들의 최소화 역시 가장 중요하다.

발명의 내용

[0015] 실시예들은 고품질의 전달을 제공하고 짧은 채널 재평 시간들을 허락하기 위해 FEC 코드들을 사용하는 채널을 통해 스트리밍 데이터를 송신 및 수신하기 위한 신규한 방법들 및 프로세스들을 제공한다. 스트리밍 및 객체 전달 모두를 위해 이러한 시스템에서 요구되는 헤더 오버헤드들을 최소화하는 신규한 시그널링 방법들이 설명된다. 스트림들을 송신 및 보호하는 신규한 배열(arrangement)들 역시 설명된다.

[0016] 후속하는 상세한 설명은 첨부 도면들과 함께 본 발명의 속성 및 이점들의 더 양호한 이해를 제공할 것이다.

도면의 간단한 설명

[0017] 도 1은 본 발명의 일 실시예에 따른 통신 시스템의 블록 다이어그램이다.

도 2는 알려져 있는 시스템에 대한 수신기 레이턴시의 컴포넌트들을 예시하는 도면이다.

도 3은 FEC 복원 심볼들이 이들이 생성되는 대응하는 소스 심볼들 이전에 송신될 때 수신기 레이턴시의 컴포넌트들을 예시하는 도면이다.

도 4는 실시예가 데이터를 서브-블록들로 우선순위화하여 상기 서브 블록들을 우선순위화된 송신 순서로 매핑하는 방법을 예시하는 블록 다이어그램이다.

도 5는 실시예가 필수(integral) 서브-블록들의 각각의 물리층 블록으로의 매핑에 기초하여 서브-블록들을 물리층 블록들로 매핑할 수 있는 방법을 예시하는 블록 다이어그램이다.

도 6은 실시예가 서브-블록들을 물리층 블록으로 매핑할 수 있는 방법 – 여기서, 동일한 양의 서브-블록 데이터가 각각의 물리층 블록으로 매핑되며, 여기서 서브-블록들은 물리층 블록들에 걸쳐 때때로 분할됨 – 을 예시하는 블록 다이어그램이다.

발명을 실시하기 위한 구체적인 내용

[0018] 여기서 설명되는 실시예들은 스트리밍 및 객체 전달 애플리케이션들 모두에 대해, 다수의 물리층 블록들 내에서 소스 블록들의 송신을 시그널링하기 위한 신규한 방법들을 제공한다. 이들 시그널링 방법들은 심볼들이, 심볼들이 생성되는 소스 블록들 및 소스 블록들에 대하여 우선순위화된 데이터의 송신 및 표시들에 대해 어떻게 관련되는지를 시그널링하여, 물리층 블록 내에서 인터리빙된 소스 블록들을 시그널링하기 위해 최소의 추가적인 오버헤드를 사용하는, 일부 경우들에서는 오버헤드를 사용하지 않는 방법들을 포함한다. 전달되는 스트림들의 품질을 개선하는 하나 이상의 채널들 상에서 스트림들을 정렬하고 송신하면서, 동시에 필요한 채널 자원들의 양 및 요구되는 수신기 전력 자원들을 최소화 또는 개선하기 위한 추가적인 방법들이 설명된다.

[0019] 이하에서, 데이터를 전달하는 네트워크는, 당업자가 여기서 설명되는 프로세스들 및 방법들이 예컨대 연속적 비트-스트림 네트워크들과 같은 다른 타입들의 전송 네트워크들에 대해 어떻게 적용되는지를 용이하게 알 수 있다는 인지와 더불어, 여기서의 설명을 간략화하기 위해 패킷-기반인 것으로 가정된다. 이하에서, FEC 코드들은, 당업자가 여기서 설명되는 프로세스들 및 방법들이 예컨대 비트-플립들과 같은 다른 타입들의 데이터 전송 손상에 대해 어떻게 적용되는지를 용이하게 알 수 있다는 인지와 더불어, 여기서의 설명을 간략화하기 위해 패킷들 내에서 유실된 패킷들 및 유실된 부분 데이터에 대한 보호를 제공하는 것으로 가정된다.

[0020] 도 1은 체인 반응 코딩을 사용하는 통신 시스템(100)의 블록 다이어그램이다. 통신 시스템(100)에서, 입력 파일(101) 또는 입력 스트림(105)은 입력 심볼 생성기(110)에 제공된다. 입력 심볼 생성기(110)는 입력 파일 또는 스트림으로부터 하나 이상의 입력 심볼들(IS(0), IS(1), IS(2), ...)의 시퀀스를 생성하며, 각각의 입력 심볼은 (괄호에 삽입된 정수로서 도 1에 표기된) 위치 및 값을 가진다. 입력 심볼들에 대해 가능한 값들, 즉, 그 알파벳은 통상적으로 2^M 개의 심볼들의 알파벳이며, 따라서, 각각의 입력 심볼은 입력 파일의 M개 비트에 대해 코딩한다. M의 값은 일반적으로 통신 시스템(100)의 사용에 의해 결정되지만, 범용 시스템은 M이 사용 경우마다

달라질 수 있도록 입력 심볼 생성기(110)에 대한 심볼 사이즈 입력을 포함할 수 있다. 입력 심볼 생성기(110)의 출력은 인코더(115)에 제공된다.

[0021] 키 생성기(120)는 인코더(115)에 의해 생성될 각각의 출력 심볼에 대한 키를 생성한다. 각각의 키는 Luby I 또는 Shokrollahi I에 기재된 방법들, 또는 키들이 이 키 생성기에 의해 생성되었든 또는 다른 키 생성기에 의해 생성되었든 간에, 스트림 내의 동일한 입력 파일 또는 데이터 블록에 대해 생성된 키들의 더 큰 부분(fractio n)이 고유함을 보증하는 임의의 비교가능한 방법 중 하나에 따라 생성된다. 예를 들어, 키 생성기(120)는 각각의 키를 생성하기 위해 카운터(125), 고유한 스트림 식별자(130)의 출력, 및/또는 난수 생성기(135)의 조합을 사용할 수 있다. 키 생성기(120)의 출력은 인코더(115)에 제공된다. 다른 예들에서, 예를 들어, 일부 스트리밍 애플리케이션들에서, 키들의 세트들은 고정되고 스트림의 각각의 데이터 블록에 대해 다시 사용될 수 있다.

[0022] 키 생성기(120)에 의해 생성되는 각각의 키(I)로부터, 인코더(115)는 입력 심볼 생성기에 의해 제공되는 입력 심볼들 중 값 B(I)를 가지는 출력 심볼을 생성한다. 각각의 출력 심볼의 값은, 그것의 키 및 여기서 출력 심볼의 "연관된 입력 심볼들" 또는 단순히 그것의 "연관들"이라 지칭되는 입력 심볼들 중 하나 이상의 일부 기능에 기초하여 생성된다. 통상적으로, M은 입력 심볼들 및 출력 심볼들에 대해 동일한데, 즉, 이를 모두는 동일한 비트들의 수에 대해 코딩되지만, 항상 그렇지는 않다.

[0023] 일부 실시예들에서, 입력 심볼들의 수 K는 연관들을 선택하기 위해 인코더에 의해 사용된다. K가 미리 알려져 있지 않은 경우, 예를 들어, 입력이 스트림이며 K가 스트림 내의 각각의 블록 사이에서 달라지는 경우, K는 단지 추정치일 수 있다. 값 K는 또한 입력 심볼들에 대한 저장 공간을 할당하기 위해 인코더(115)에 의해 사용될 수 있다.

[0024] 인코더(115)는 송신 모듈(140)로 출력 심볼들을 제공한다. 송신 모듈(140)에는 또한 키 생성기(120)로부터 이러한 출력 심볼 각각의 키가 제공된다. 송신 모듈(140)은 출력 심볼들을 전송하며, 사용되는 키잉 방법에 따라, 송신 모듈(140)은 또한 전송되는 출력 심볼들의 키들에 관한 일부 데이터를, 채널(145)을 통해, 수신 모듈(150)로 전송할 수 있다. 채널(145)은 삭제 채널인 것으로 가정되지만, 이는 통신 시스템(100)의 적절한 동작을 위한 요건은 아니다. 모듈들(140, 145 및 150)은, 송신 모듈(140)이 출력 심볼들 및 그들의 키들에 대한 임의의 요구되는 데이터를 채널(145)로 전송하도록 적응되고 수신 모듈(150)이 채널(145)로부터 심볼들 및 그들의 키들에 대한 잠재적인 일부 데이터를 수신하도록 적응되는 한, 임의의 적절한 하드웨어 컴포넌트들, 소프트웨어 컴포넌트들, 물리적 매체, 또는 이들의 임의의 조합일 수 있다. K의 값은, 상기 연관들을 결정하는데 사용되는 경우, 채널(145)을 통해 송신될 수 있거나 또는 인코더(115) 및 디코더(155)의 합의에 의한 시간 이전에 세팅될 수 있다.

[0025] 채널(145)은 텔레비전 송신기로부터 텔레비전 수상기까지의 브로드캐스트 링크 또는 인터넷을 통한 경로와 같은 실시간 채널, 또는 한 포인트에서 다른 포인트까지의 전화 접속일 수 있거나, 채널(145)은 CD-ROM, 디스크 드라이브, 웹 사이트 등과 같은 저장 채널일 수 있다. 심지어 채널(145)은 한 개인이 전화선을 통해 개인용 컴퓨터로부터 인터넷 서비스 공급자(ISP)로 입력 파일을 전송할 때 형성되는 채널과 같은 저장 채널 또는 실시간 채널의 조합일 수도 있으며, 상기 입력 파일은 웹 서버 상에 저장되고 후속적으로 인터넷을 통해 수신자에게 전송된다.

[0026] 여기서, 채널(145)은 패킷 네트워크를 포함하며, 통신 시스템(100)은 임의의 둘 이상의 패킷들이 채널(145)을 통해 전송 중에(in transit) 보존될 수 있다는 점을 가정하지 못할 수도 있다. 따라서, 출력 심볼들의 키는 전술된 키잉 방식들 중 하나 이상을 사용하여 결정되며, 반드시 출력 심볼들이 수신 모듈(150)을 빠져나가는 순서로 결정되는 것은 아니다.

[0027] 수신 모듈(150)은 상기 출력 심볼들을 디코더(155)에 제공하며, 임의의 데이터 수신 모듈(150)은 키 재생성기(160)에 제공되는 이들 출력 심볼들에 관한 키들을 수신한다. 키 재생성기(160)는 상기 수신된 출력 심볼들에 대한 키들을 재생성하고 이들 키들을 디코더(155)에 제공한다. 디코더(155)는 입력 심볼들(다시, IS(0), IS(1), IS(2), ...)을 복원하기 위해, 대응하는 출력 심볼들과 함께 키 재생성기(160)에 의해 제공되는 키들을 사용한다. 디코더(155)는 복원된 입력 심볼들을 입력 파일 리어셈블러(165)에 제공하며, 상기 입력 파일 리어 셱블러(165)는 입력 파일(101)의 카페(170) 또는 입력 스트림(105)의 카페(175)를 생성한다.

[0028] 미디어 스트리밍 애플리케이션들에서 사용될 때, 소스 미디어 스트림을 형성하는 소스 패킷들은 때때로 소스 블록들이라 지칭되는 그룹들 내에서 수집된다. 예를 들어, 소스 블록은 고정된 시간 길이에 걸쳐 있는 소스 패킷들의 그룹일 수 있으며, 예를 들어, 리드-솔로몬 삭제 코드는, 소스 블록의 오리지널 소스 패킷들과 함께 수신

기들에 송신되는 복원 패킷들을 생성하기 위해 이들 소스 블록들에 독립적으로 적용될 수 있다.

[0029] 송신기에서, 소스 스트림은 소스 패킷들이 도착할 때 소스 블록들로 연속적으로 분할될 수 있고, 이후 복원 패킷들이 각각의 소스 블록에 대해 생성되어 송신된다. 특히 라이브 또는 인터랙티브 스트리밍 애플리케이션들에 대한 FEC 코드들의 사용에 의해 추가된 전체 종단-대-종단 지연을 최소화시키는 것이 바람직하고, 따라서, FEC 솔루션의 전체 설계가 소스 패킷들이 송신되기 이전에 송신기에서 가능한 적게 지연되고, 소스 블록에 대한 모든 소스 및 복원 패킷들이 가능한 적은 전체 지연을 가지고 송신되는 경우가 바람직하다. 또한, FEC 인코딩된 스트림의 레이트가 가능한 스무스(smooth)하고, 즉, FEC 인코딩된 스트림 레이트에서 가능한 적은 변경가능성이 존재하거나 또는 적어도 오리지널 소스 스트림 내에 이미 존재하는 임의의 변경가능성에 대한 어떠한 증폭도 존재하지 않는 경우가 바람직한데, 왜냐하면 이것이 FEC 인코딩된 스트림 대역폭 사용을 더 많이 예측가능하게 하며, 다른 가능한 경쟁 스트림들 및 네트워크에 대한 영향을 최소화하기 때문이다. 또한, 패킷들이 소스 블록에 대해 송신되는 기간 동안 상기 소스 블록에 대해 패킷들로 송신되는 데이터가 가능한 균일하게 확산되는 경우가 바람직한데, 왜냐하면 이것이 버스트 유실들에 대한 최상의 보호를 제공하기 때문이다.

[0030] 수신기에서, 패킷들이 유실되거나 또는 (예를 들어, CRC 검사들을 사용하여 검출 및 폐기될 수 있는) 에러들을 가지고 수신되는 경우, 충분한 복원 패킷들이 수신되었다고 가정하면, 상기 복원 패킷들은 유실된 소스 패킷들 중 하나 이상을 복원하는데 사용될 수 있다.

[0031] 일부 애플리케이션들에서, 패킷들은 FEC 프로세스가 적용되는 심볼들로 추가적으로 세분된다. 일부 FEC 코드들, 특히, 리드-솔로몬 코드들에 대해, 인코딩 및 디코딩 시간은, 소스 블록당 심볼들의 수가 커짐에 따라 구현불가능하게 커지며, 소스 블록당 생성될 수 있는 인코딩 심볼들에 대한 전체 수에 대한 상한 경계가 종종 존재한다. 심볼들이 일반적으로 애플리케이션 층에서 사용될 때 상이한 패킷 페이로드들로 위치되므로, 이는 소스 블록의 인코딩 시 최대 길이에 대한 실제 상한 경계를 위치시키고, 또한 이는 물론 소스 블록 자체의 사이즈에 대한 상한 경계이다.

[0032] 많은 애플리케이션들에 대해, 긴 시간 기간동안 보호가 제공될 예정이거나 미디어 스트리밍 레이트가 높을 때, 패킷당 하나의 심볼을 전달함으로써 지원될 수 있는 것 보다 더 큰 소스 블록 사이즈에 대한 보호를 제공하는 것이 유리할 수 있다. 이러한 경우들에서, 더 짧은 소스 블록들을 사용하고, 다음으로 상이한 소스 블록들로부터 소스 패킷들을 인터리빙하는 것은 개별 소스 블록으로부터의 소스 패킷들이 더 긴 시간 기간동안 확산되는 솔루션을 제공한다. 또 다른 관련 방법은 패킷들에 맞지 않는 더 긴 심볼들로부터 더 큰 심볼들로부터 더 큰 소스 블록을 형성하고, 상기 심볼들을 연속적인 패킷들에 들어갈 수 있는 서브 심볼들로 분할하는 것이다. 이러한 방법을 사용하여, 심볼에 대한 잠재적으로 상이한 서브-심볼 유실 또는 손상 패턴들의 댓가로 더 큰 소스 블록들이 지원될 수 있다. 그러나, 채널이 버스트 형태로(burstly) 또는 강하게 상관되는 손상을 나타내는 많은 경우들에서, 심볼을 포함하는 서브-심볼들의 유실 또는 손상은 매우 상관되며, 따라서, 종종 이 방법을 사용할 때 제공되는 FEC 보호의 품질저하가 거의 없다.

용어

FEC 코드들

[0035] 이 설명에서, 인코딩된 데이터(소스 데이터)가, (단일 비트까지 다운되는) 임의의 길이일 수 있는, 동일한 길이의 "심볼들"로 분할되었다고 가정한다. 심볼들은 패킷들로 데이터 네트워크를 통해 전달될 수 있으며, 심볼들의 전체 수는 각각의 패킷에 포함되거나 명시적으로 전달된다. 일부 경우들에서, 소스 패킷이 다수의 심볼 길이가 아닌 것이 가능하며, 이 경우, 패킷 내의 마지막 심볼은 절단(truncate)될 수 있다. 이 경우, FEC 코딩의 목적으로, 이 마지막 심볼은 고정된 비트의 패턴, 예를 들어, 제로-값의 비트들로서 패딩되는 것으로 목시적으로 가정되어, 그 결과 이들 비트들이 패킷 내에서 전달되지 않는다 할지라도 수신기는 여전히 이 마지막으로 절단된 비트를 풀(full) 심볼로 채울 수 있다. 다른 실시예들에서, 고정된 비트들의 패턴은 패킷으로 위치될 수 있으며, 이에 의해, 상기 심볼들을 패킷과 동일한 길이로 효과적으로 패딩할 수 있다. 심볼의 사이즈는 종종 비트로 측정될 수 있으며, 여기서 심볼은 M개 비트의 사이즈를 가지며 상기 심볼은 2^M개의 심볼들의 알파벳으로부터 선택된다. 넌-바이너리 디지트(non-binary digit)들 역시 참작될 수 있지만, 바이너리 비트들이 더 일반적으로 사용되므로 선호된다.

[0036] 여기서 스트리밍에 대해 고려되는 FEC 코드들은 통상적으로 계층적 FEC 코드들인데, 즉, 소스 블록의 소스 심볼들은 소스 블록의 인코딩의 일부분으로서 포함되며, 따라서 소스 심볼들이 전송된다. 이후, 계층적 FEC 코드가 소스 심볼들의 소스 블록으로부터 소정 개수의 복원 심볼들을 생성하며, 그 다음으로, 상기 소스 및 복원 심볼

들의 조합은 소스 블록으로 송신되는 인코딩 심볼이 된다. FEC 코드들 중 몇몇은 필요한 만큼 많은 복원 심볼들을 효율적으로 생성하는 성능을 가진다. 이러한 코드들은 "정보 추가 코드(information additive code)들" 및 "파운틴 코드(fountain code)들"이라 지칭되며, 이들 코드들의 예들은 "체인 반응 코드들" 및 "멀티-스테이지 체인 반응 코드들"을 포함한다.

[0037] 리드-솔로몬 코드들과 같은 다른 FEC 코드들은 실제로 제한된 수의 소스 심볼들로부터 제한된 수의 복원 심볼들만을 생성할 수 있다. 이들 타입들의 코드들에 대해, 소스 블록은 여전히 상대적으로 클 수 있으며, 여기서 소스 블록은 소스 블록의 다수의 소스 심볼들이 기껏해야 실제 개수의 소스 심볼들에 대한 상한 경계가 되도록, 그리고 소스 블록으로부터 생성되는 원하는 수의 복원 심볼들이 기껏해야 실제 개수의 복원 심볼들에 대한 상한 경계가 되도록 충분히 큰 사이즈의 심볼들로 분할된다. 이들 심볼들이 물리적 패킷들의 전송을 위한 적절한 사이즈보다 더 큰 경우, 상기 심볼들은 이러한 패킷들로 개별적으로 전달될 수 있는 서브-심볼들로 추가적으로 분할될 수 있다. 후속 설명들을 간략화하기 위해, 심볼들은 통상적으로 나눌 수 없는(indivisible) 유닛들로서 설명되는 반면, 많은 경우들에서 심볼들은 다수의 서브-심볼로 구성될 수 있으며, 심볼들이 상기 설명들에서 서브-심볼들로 분할될 수 있다는 점이 이해되며, 결과적인 방법들 및 프로세스들은 심볼들을 사용하는 설명들과 상당히 유사할 것이다.

[0038] 패킷들 내의 심볼들을 전달하기 위한 많은 다른 방법들이 존재하며, 아래 설명들은 간략함을 위해 이 예를 사용하지만, 그것이 제한적이거나 포괄적이도록 의도되지는 않는다. 아래 설명의 컨텍스트에 있어서, 용어 "패킷"은 문자 그대로 단일 유닛의 데이터로서 송신되는 것을 의미하도록 제한되도록 의도되지는 않는다. 대신, 그것은 단일 유닛의 데이터로서 송신되거나 송신되지 않을 수 있는 부분적인 심볼들 또는 심볼들의 논리적 그룹(logical grouping)을 정의하는 더 넓은 개념(notion)을 포함하도록 의도된다.

[0039] 또한, 심볼들, 예컨대, 송신시 그들의 값을 변경시키거나 다른 방식들로 손상되는 심볼들의 유실이 아닌 데이터의 손상 형태들이 존재하며, 아래에 설명되는 방법들은 상기 심볼들에 동일하게 적용된다. 따라서, 아래 설명들은 종종 심볼들의 유실을 설명할 것이지만, 상기 방법들은 다른 타입들의 손상 및 FEC 삭제 코드들이 아닌 다른 타입들의 FEC 코드들, 예컨대, FEC 에러-정정 코드들 및 FEC 체크-섬 코드들 및 FEC 인증 코드들에 동일하게 적용된다.

스트리밍

[0041] 소스 스트리밍의 FEC 보호를 제공할 목적으로, 소스 스트리밍은 하나 이상의 논리적 스트리밍들의 조합일 수 있으며, 상기 논리적 스트리밍들의 조합의 예들은 오디오 RTP 스트리밍 및 비디오 RTP 스트리밍의 조합, MIKEY 스트리밍 및 RTP 스트리밍의 조합, 둘 이상의 비디오 스트리밍들의 조합, 제어 RTCP 트래픽 및 RTP 스트리밍의 조합이다. 소스 스트리밍이, 예를 들어, 소스 비트 스트리밍, 소스 심볼 스트리밍, 또는 소스 패킷 스트리밍인 포맷으로 송신기에 도달할 때, 상기 송신기는 상기 스트리밍을 소스 블록들로 버퍼링하고 소스 블록들로부터 복원 스트리밍을 생성할 수 있다. 송신기는 예를 들어, 패킷 네트워크를 통해 송신될 패킷들로 소스 스트리밍 및 복원 스트리밍을 스케줄링 및 송신한다. FEC 인코딩된 스트리밍은 결합된 소스 및 복원 스트리밍이다. 수신기는 예를 들어, 유실률 또는 비트-플립률로 인해 손상되었을 수 있는 FEC 인코딩된 스트리밍을 수신한다. 수신기는 소스 스트리밍의 오리지널 소스 블록들 중 일부 또는 전부를 재구성하려 시도하여, 수신기에서 이를 오리지널 소스 스트리밍의 재구성된 부분들이 예컨대 미디어 플레이어에 대해 사용가능하게 한다.

[0042] 스트리밍 애플리케이션에 대해, 소스 스트리밍을 보호하기 위해 FEC 코드들을 사용하는 방법의 설계에 대한 입력들인 몇몇 키 파라미터들 및 통상적으로 최적화를 위해 중요한 몇몇 키 메트릭들이 존재한다.

[0043] 상기 설계에서의 2개의 키 입력 파라미터들은 보호 기간 및 보호량이다. 소스 블록에 대한 송신기 보호 기간은 상기 소스 블록으로부터 생성되는 심볼들이 송신되는 동안의 시간 드레이션이다. 소스 블록의 보호량은 소스 블록 내의 소스 심볼들의 수의 부분(fraction) 또는 퍼센트(percentage)로서 표현되는, 소스 블록에 대해 송신되는 FEC 복원 심볼들의 수이다. 예를 들어, 보호 기간이 2초이고 보호량이 20%이고 소스 블록 내에 10,000개의 소스 심볼들이 존재한다면, 상기 소스 블록에 대한 상기 10,000개의 소스 심볼들 및 2,000개의 복원 심볼들은 2초의 시간 윈도우를 통해 송신된다. 소스 블록당 보호량 및 보호 기간 모두는 소스 블록마다 달라질 수 있다. 예를 들어, 소스 블록이 바람직하게는 소스 스트리밍 내의 특정 소스 패킷들 간에 걸쳐져 있지 않는 경우, 예를 들어, 제 1 패킷이 MPEG2 비디오 스트리밍 내의 GOP(Group of Pictures)의 마지막 패킷이고, 제 2 연속적인 패킷이 다음 GPO의 제 1 패킷인 경우, 소스 블록은 제 1 패킷 이후에 그리고 제 2 패킷 이전에 심지어 이것이 발생한다 할지라도 보호 기간의 종단 이전에 종료될 수 있다. 이는 FEC 보호 블록이 비디오 코딩 블록에 맞게 조정되는(aligned) 것을 허용하는데, 이는 비디오 버퍼링 및 FEC 버퍼링으로 인한 수신기 레이턴시가 최소화될

수 있다는 이점을 포함하여 많은 이점들을 가질 수 있다. 다른 애플리케이션들에서, 다양한 이유들로, 각각의 연속적인 소스 블록에 대해 동일한 보호 기간 및/또는 소스 블록 사이즈를 항상 유지하는 것이 유리할 수 있다. 아래 설명의 많은 부분들에서, 간략함을 위해, 보호 기간 및 보호량 모두가 각각의 후속적인 소스 블록에 대해 동일한 것으로 가정된다. 당업자에게, 이것이 제한적이 아니라는 것이 명백해야 하는데, 왜냐하면 당업자라면 본원의 판독시에 여기서 설명되는 프로세스들 및 방법들이 보호량 또는 보호 기간 또는 이들 모두가 소스 블록마다 달라질 때 그리고 소스 블록 사이즈들이 소스 블록마다 달라질 때 적용되는 방법을 용이하게 결정할 수 있기 때문이다.

[0044] 후속 논의들 중 일부를 간략화하기 위해, 오리지널 스트림의 소스 심볼들이 고정 레이트로 FEC 인코딩을 수행하기 위한 송신기에 도달하며, 수신기가 소스 심볼들을 상기 수신기에서 사용가능하게 하면, 소스 심볼을 송신하는 제 1 소스 블록에 있어서 어떠한 심볼 유실도 없고, 각각의 후속하는 소스 블록에 있어서, 인코딩 심볼 유실은 기껏해야 성공적인 FEC 디코딩을 허용할 정도라고 가정하는 경우, 후속적인 소스 심볼들은 동일한 고정 레이트로 수신기에 의해 사용가능하게 된다는 점이 종종 가정된다. 이러한 간략화 가정은 후속적으로 설명되는 프로세스들 및 방법들의 동작 또는 설계에 내재되지 않으며, 이들 프로세스들을 이러한 가정으로 어떠한 식으로든 제한하도록 의도되는 것이 아니라, 단지 상기 프로세스들 및 방법들의 특징들에 대한 설명들 중 일부를 간략화하기 위한 도구로서 도입된다. 예를 들어, 가변 레이트의 스트림들에 대해, 대응하는 조건은 소스 심볼들이 이들이 송신기에 도달하는 레이트와 동일하거나 이에 가까운 레이트로 수신기에 의해 사용가능하게 된다는 점이다.

[0045] 최소화를 위해 중요한 일부 키 메트릭들은 송신기에 의해 도입되는 레이턴시인 송신기 레이턴시를 포함한다. 송신기 레이턴시의 최소화는 비디오 스트리밍과 같은 일부 애플리케이션들 또는 화상 회의와 같은 인터랙티브 애플리케이션들에 대한 바람직한 목표이다. 송신기 레이턴시의 최소화를 보조하는 전체 설계의 일 양상은 송신기가 소스 심볼들이 송신기에 도달하는 것과 동일한 순서로 상기 소스 심볼들을 송신하는 것이다. 송신기 레이턴시를 최소화하는 다른 설계 양상들이 추후 설명된다.

[0046] 또다른 중요한 메트릭은 채널 재평 시간이다. 이것은 수신기가 스트림을 조인시키거나 요청하고 상기 스트림으로부터 인코딩 심볼들의 수신을 시작할 때와 수신기가 먼저 상기 스트림으로부터의 소스 심볼들을 사용가능하게 할 때까지 사이의 시간이다. 일반적으로, 채널 재평 시간을 최소화시키는 것이 바람직한데, 왜냐하면, 이것은 심볼들이 수신기에 의해 디코딩되어 이를 따라 전달되기 이전에 심볼들을 저장하기 위한 메모리 요건들을 수신기에서 최소화하고, 또한 이것은 스트림이 조인될 때와 상기 스트림이 예컨대 비디오 스트림의 플레이백에 대해 먼저 사용가능해지기 시작할 때 사이의 시간량을 최소화하기 때문이다.

[0047] 많은 알려져 있는 시스템들에 대해, 채널 재평 시간을 최소화하는 한가지 중요한 양상은 송신기가 소스 심볼들의 오리지널 송신 순서를 유지하는 것이다. 후속 섹션에서, FEC 코드들을 적용하고, 채널 재평 시간들을 최소화하는 방식들로 각각의 소스 블록에 대한 인코딩된 데이터를 송신하기 위해, 블록 내의 소스 심볼들을 정렬(ordering) 및 인코딩하는 신규한 방식들이 설명된다.

[0048] 이제 설명되는 바와 같이, 많은 알려져 있는 시스템들에 대해, 채널 재평 시간은 통상적으로 다수의 컴포넌트들을 포함한다. 순차적인 소스 블록들로 분할되는 스트림에 대한 이들 컴포넌트들의 예가 도 2에 도시된다. 도 2는 보호 기간당 단일 소스 블록이 존재하는 전형적인(classical) IPTV 배치에서 사용될 수 있는 설계를 도시하며, 상기 보호 기간동안 각각의 소스 블록에 대한 복원 심볼들이 상기 소스 블록에 대한 소스 심볼들 바로 다음에 송신되며, 상기 예는 수신기가 소스 블록의 시작시에 스트림을 조인하는 경우를 도시한다. 이 예에서의 채널 재평 시간의 두가지 컴포넌트들은 보호 기간 및 디코딩 레이턴시이다. 수신기 보호 기간은 수신기가 소스 블록으로부터 수신된 인코딩 심볼들을 버퍼링하는 시간이다. 송신기 보호 기간 및 수신기 보호 기간은, 송신기와 수신기 사이의 채널이 각각의 비트, 바이트, 심볼 또는 패킷을 송신기로부터 수신기로 이동시키기 위해 소모하는 시간량의 측면에서 어떠한 변경도 가지지 않는 경우, 동일하다는 점에 유의한다. 따라서, 실제로, 송신기 보호 기간은 전달시의 네트워크 타이밍 변경들로 인해 동일한 소스 블록에 대해 수신기 보호 기간과는 다를 수 있다. 설명을 간략화하기 위해, 이하에서, 송신기 보호 기간 및 수신기 보호 기간은 각각의 소스 블록에 대해 동일하다고 가정하며, 용어 "보호 기간"이 송신기 보호 기간 및 수신기 보호 기간에 대해 동일한 뜻으로 사용될 것인데, 다시 말해, 네트워크 전달 시간은 모든 데이터에 대해 동일한 것으로 가정되며, 당업자가 네트워크 전달 변동(fluctuation)들로 인해 송신기 및 수신기 보호 기간들에서의 차이를 고려하기 위해 여기서 설명되는 방법들 및 장치에 대해 필요한 변경을 수행할 수 있다는 점에 유의한다.

[0049] 수신기 레이턴시의 보호 기간 컴포넌트는 이들 공지된 시스템들에서는 불가피한데, 왜냐하면 제 1 소스 블록에

있어서, 어떠한 소스 심볼들의 유실도 존재하지 않는다 할지라도, 후속하는 소스 블록들 내의 인코딩 심볼들의 유실이 존재할 때 모든 후속적인 소스 심볼들의 스무스한 소스 심볼 전달을 보장하기 위해 적어도 보호 기간 동안 소스 심볼들을 사용가능하게 하기 위한 자연이 이루어져야 하기 때문이다. 보호 기간 동안, 소스 블록의 FEC 디코딩의 일부 또는 대부분 또는 전부는 인코딩 심볼의 수신과 동시에 발생할 수 있다. 보호 기간의 종단에서, 소스 블록의 제 1 소스 심볼이 수신기로부터 사용가능해지기 이전에 발생하는 추가적인 FEC 디코딩이 존재할 수 있으며, 이 시간 기간은 도 2에서는 디코딩 레이턴시라 명기(label)된다. 또한, 제 1 소스 심볼이 사용가능하게 되는 이후라 할지라도, 소스 블록의 제 2 및 후속 소스 심볼들이 사용가능해지기 전에 발생하는 추가적인 FEC 디코딩이 존재할 수 있다. 간략함을 위해, 이러한 추가적인 FEC 디코딩은 도 2에 도시되지 않으며, 이 예에서, 충분히 빠른 레이트로 처음 심볼 이후의 모든 소스 심볼들을 디코딩하기 위한 충분한 사용가능한 CPU 자원들이 존재한다고 가정된다.

[0050] 이들 공지된 시스템들에 있어서, 수신기가 소스 블록의 중간에서 스트림을 조인시키면, 소스 패킷들의 오리지널 송신 순서가 송신기에 의해 유지되는 한, 채널 재평 시간은 제 1 부분 소스 블록으로부터의 소스 심볼들의 어떤 손실도 없는 경우 보호 기간 더하기 디코딩 레이턴시만큼 작아질 수 있다. 따라서, 이들 공지된 시스템들에 대해, 송신기가 소스 심볼들의 오리지널 순서를 유지하는 것이 바람직하다.

[0051] 스트리밍 방법의 또다른 목적은 FEC 종단-대-종단 레이턴시를 최소화하는 것인데, 상기 FEC 종단-대-종단 레이턴시는 소스 패킷이 FEC 인코딩 이전에 송신기에서 스트리밍하기 위해 대기할 때와 그것이 FEC 디코딩이 적용된 이후 수신기에서 플레이백을 위해 사용가능해질 때 사이에서 FEC의 사용에 의해 도입되는 최악의 경우의 전체 레이턴시이다.

[0052] 스트리밍 방법의 또다른 목적은 FEC가 사용될 때 송신 레이트의 변동들을 최소화하기 위한 것이다. 이 목적에 대한 한가지 이유는 패킷 네트워크들 내에서, 변동하는 송신 레이트를 가지는 스트림들이, 상기 스트림의 송신 레이트에서의 피크치들이 제한된 용량을 가지는 네트워크 내의 포인트들에서 다른 트래픽의 피크치들과 일치할 때의 혼잡 또는 버퍼 오버플로우들로 인해 더 패킷 유실되기 쉽다는 점이다. 최소한, FEC 인코딩된 스트림의 레이트의 변동들은 오리지널 소스 스트림의 레이트들의 변동들보다는 더 열악하지 않아야 하며, 바람직하게는 오리지널 소스 스트림에 대해 더 많은 FEC 보호가 적용될수록 FEC 인코딩된 스트림의 레이트의 변동들은 더 작아진다. 특수한 경우로서, 오리지널 스트림이 일정한 레이트로 송신되는 경우, FEC 인코딩된 스트림 역시 가능한 거의 일정한 레이트로 송신되어야 한다.

[0053] 스트리밍 방법의 또다른 목적은 수신기에서 가능한 간단한 로직을 사용할 수 있게 되는 것이다. 이것은 많은 상황들에서 중요한데, 왜냐하면 수신기가 제한된 계산상의 메모리 또는 다른 자원 성능들을 가지는 디바이스 내에 탑재될 수 있기 때문이다. 또한, 일부 경우들에서 송신시 심볼들의 상당한 유실 및 손상이 존재할 수 있으며, 따라서, 수신기는 조건들이 개선될 때 스트림 수신이 계속되는 것을 이해할만한 컨텍스트가 거의 없거나 전혀 존재하지 않는 엄청난(catastrophic) 유실 또는 손상 시나리오들로부터 복원해야 할 수도 있다. 따라서, 수신기 로직이 더 간단하고 더 로버스트(robust) 할수록 수신기가 스트림의 수신으로부터 소스 스트림의 심볼들을 더 빠르고 더 신뢰성 있게 복원하고 사용가능하게 하기 시작할 수 있을 것이다.

[0054] 하나의 소스 블록에 대해 송신될 FEC 인코딩된 데이터가 다른 소스 블록들에 대해 송신된 데이터와 인터리빙된 더 긴 시간 기간 동안 확산되어 송신되는 경우, 상기 소스 블록에 대한 FEC 인코딩된 데이터의 송신은 채널의 유실 또는 손상에 대한 가능한 최상의 보호를 보장하기 위해 시간상 가능한 균일하게 송신되어야 한다.

[0055] 소스 블록에 대한 데이터의 송신은 수신기가 시간적인 면에서 결정된 우선순위 순서로 소스 블록의 소스 데이터를 복원할 수 있게 되어야 한다.

[0056] 스트림에 대해 송신되는 데이터는 헤더 오버헤드들을 최소화하기 위해 스트림과 연관된 가능한 적은 헤더 정보를 가지고 송신되어야 한다. 바람직하게는 어떠한 헤더 정보도 상기 스트림과 함께 송신되지 않으며, 헤더 정보의 일부 또는 전부가 시스템에 내장된 다른 정보로부터 유도되거나 상기 다른 정보로부터 이미 사용가능하게 되며, 그리고/또는 상기 헤더 정보의 일부 또는 전부는 다른 정보, 예를 들어 수신기에서의 정보의 도달 타이밍으로부터 추론될 수 있다.

[0057] 다음 섹션에서, 이들 목적들 중 일부 또는 전부를 만족시키는 방법들, 프로세스들 및 장치를 설명한다.

개선된 송신 및 수신 방법들 및 프로세스들

[0059] 일부 경우들에서, 블록으로서 전달될 데이터는 우선순위화될 수 있다. 다른 경우들에서, 블록으로서 전달될 데이터는 반드시 우선순위화되지는 않는다. 임의의 경우에 있어서, 데이터의 오리지널 스트림은 소스 블록들로

분할되며, FEC 복원 데이터는 각각의 상기 소스 블록에 대해 생성되며, 그후 오리지널 소스 블록 데이터 및 상기 소스 블록으로부터 생성된 FEC 복원 데이터를 포함하는, 각각의 상기 소스 블록에 대해 인코딩되는 데이터는 상기 소스 블록의 오리지널 플레이-아웃보다 더 많은 시간동안 확산된다(따라서, 후속 소스 블록들에 대한 인코딩된 데이터는 서로 인터리빙된다). 이를 경우들에서, 적용되는 FEC 코드들은 삭제 코드들일 수 있으며, 상기 삭제 코드들은, 비록 다른 타입들의 FEC 코드들, 예를 들어, 에러-정정 코드들일 수 있는 FEC 코드들, 또는 데이터 무결성을 인증하는데 사용될 수 있는 FEC 코드들 역시 참작될 수 있음에도 불구하고, 원하는 보호량까지 데이터의 유실에 대해 스트림의 데이터를 보호한다. 이를 경우들에서, 보호 기간이라 지칭되는, 스트림의 각각의 블록에 대해 인코딩된 데이터가 송신되는 시간이 더 길수록, 그리고 상기 인코딩된 데이터가 상기 보호기간 동안 더 균일하게 확산될수록, 애플리케이션층 FEC 코드에 의해 제공되는 패킷 유실에 대한 보호 레벨이 더 양호해진다.

[0060] 본 발명의 일 실시예에 있어서, 인코딩된 데이터의 송신은, 여기서 물리층 패킷이라 지칭되는 균일한 사이즈 부분들, 예를 들어 각각 120 바이트의 물리적 채널 내에서 송신된다. 물리층 패킷들은 각각의 물리층 패킷을 순상으로부터 보호하기 위해 이들에 적용되는 물리층 FEC 코드를 가질 수 있다. 일부 경우들에서, 다수의 물리층 패킷들은 슬롯 당 동일한 수의 물리층 패킷들의 슬롯들, 예를 들어, 512개의 물리층 패킷들의 슬롯들로 분할된다. 물리층에서의 프로토콜들은 때때로 각각의 시간 슬롯 내의 물리층 패킷들을 구별하고 고유하게 식별하기 위해 사용될 수 있다. 이를 경우들에서, FEC 심볼들은 물리층 패킷들로 직접 매핑될 수 있고, 또한 어느 심볼들이 어느 물리층 패킷에서 전달되는지에 대한 식별은 물리층 패킷의 신원의 결정 방법에 의해 거의 또는 전적으로 결정될 수 있으며, 심볼 데이터와 함께 각각의 물리층 패킷 내의 심볼 식별 데이터를 전달할 필요성을 완화시키거나 또는 상기 필요성을 완전히 제거한다. 일부 경우들에서, 심볼 식별 데이터의 일부량, 또는 스트림 또는 소스 블록의 어느 부분으로부터 심볼이 생성되었는지에 대한 일부 정보는, 바람직하게는 상기 심볼과 함께 물리층 패킷에서 전달된다. 예를 들어, 121바이트의 물리층 패킷에 대해, 1바이트의 이러한 심볼 식별 데이터가 존재할 수 있고 나머지 120바이트의 심볼 사이즈가 존재할 수 있는 반면, 상기 심볼이 오리지널 소스 스트림들로부터 생성되는 방법을 완전히 결정하기 위해, 상기 심볼과 함께 물리층 패킷에서 전달되는 심볼 식별 데이터 및 예컨대, 프레임 내의 물리층 패킷의 위치에 의해, 그리고/또는 물리층 패킷을 포함하는 프레임의 식별자에 의해, 그리고/또는 물리층 패킷 및/또는 상기 물리층 패킷을 포함하는 프레임의 수신 타이밍에 의해 상기 물리층 패킷이 고유하게 식별되는 방법의 조합으로부터 유래할 수 있다. 예를 들어, 심볼이 유래하는 소스 블록의 일부분을 식별하는 1바이트 식별자가 존재하며, 여기서, 예컨대, 소스 블록의 상이한 부분들은 소스 블록의 일부분이 그 일부인 데이터의 우선순위에 의해 명기되거나 그리고/또는 다수의 스트림들 중 소스 심볼이 유래하는 스트림에 의해 명기된다.

[0061] 복원 패킷들이 예를 들어 "FEC 스트리밍"에서 설명된 바와 같이 소스 패킷들의 이전에 송신되는 경우 상기 프로세스에 대해 특정 개선들이 이루어질 수 있다. 이러한 방식은 송신기에 추가적인 지연을 주입하는 비용을 가지는데, 왜냐하면, 소스 패킷들이 일반적으로 버퍼에 저장되어 복원 패킷들 이후에 송신될 것이기 때문이다. 또 다른 예로서, 복원 데이터는 소스 블록의 일부 또는 전부로부터 생성될 수 있다. 예를 들어, 복원 데이터의 일부분들은 전체 소스 블록으로부터 생성될 수 있으며, 다른 부분들은 소스 블록의 하나 이상의 우선순위층들로부터 생성될 수 있다. 둘 이상의 물리층 패킷에 걸쳐 있을 수 있는 애플리케이션층 패킷 또는 물리층 패킷내의 심볼과 함께 전달되는 식별 심볼 데이터가 존재하는 경우, 복원 심볼에 대한 이러한 식별 심볼 데이터의 일부분은 소스 블록의 어느 부분들로부터 그것이 생성되는지를 식별할 수 있다.

시그널링 방법들

[0063] 일부 실시예들에서, 각각의 심볼에 대해, 상기 심볼과 연관되는 헤더 데이터, 예컨대 1바이트의 헤더 데이터가 상기 심볼에 대한 정보, 예컨대, 둘 이상의 스트림이 존재하는 경우 스트리밍 식별자, 소스 블록이 둘 이상의 물리층 블록에 걸쳐 송신될 예정인 경우 세그먼트 식별자, 소스 블록이 다수의 서브-블록들을 포함하는 경우 서브-블록 식별자, 소스 블록 내의 심볼들의 심볼 순서에 따른 소스 블록 내의 심볼의 위치 등을 시그널링하는데 사용될 수 있다. 일부 실시예들에서, 이러한 헤더 데이터의 일부 또는 전부는 물리층 패킷들 내의 각각의 심볼과 함께 송신될 수 있다. 다른 실시예들에서, 각각의 심볼에 대한 헤더 데이터는 다른 정보로부터 유도되는 대부분 또는 전체이며, 물리층 패킷들 내의 각각의 심볼과 함께 헤더 데이터가 거의 송신되지 않거나 어떤 헤더 데이터도 송신되지 않는다.

소스 블록 내의 심볼들

[0065] 바람직하게는, 소스 블록의 심볼들의 순서는 명시적으로 또는 목시적으로 결정되며, 송신기에서 수신기에서와

동일한 순서일 수 있다. 상기 순서에 대한 특정 다른 바람직한 특징들은 때때로 스트리밍 또는 객체 전달 애플리케이션들에 대해 이점이 있을 수 있다. 바람직한 특징은, 예를 들어, 소스 블록에 대한 심볼들의 순서가 모든 복원 심볼들에 선행하는 순서로 먼저 모든 소스 심볼들을 가지는 것일 수 있다. 또다른 예는 상기 심볼들은 소스 블록의 서브-블록 구조에 의해 결정되는 순서에 있다는 점인데, 예를 들어, 소스 블록의 제 1 서브-블록과 연관된 모든 심볼들은 상기 순서에서 첫번째, 소스 블록의 제 2 서브-블록과 연관된 모든 심볼들은 상기 순서에서 두번째인 등의 식이다. 이전에 설명된 바와 같이, 심볼들은 다수의 서브-심볼들을 포함할 수 있다.

[0066] 소스 블록 내의 ESI들

ESI(인코딩된 심볼 식별자)는 일부 경우들에서 소스 블록 내의 소스 심볼들의 수와 같은 다른 정보와 관련하여, 상기 심볼이 소스 블록으로부터 생성되는 방법을 결정하는 임의의 식별자이다. ESI는 심볼들을 생성하기 위해 송신기에서 또는 심볼들을 식별 및/또는 복원하기 위해 수신기에서 명시적으로 사용될 수 있거나, 또는 상기 ESI가 목시적으로 사용될 수 있다. 바람직하게는, 각각의 소스 블록에 대한 심볼들은 송신기 및 수신기가 심볼 순서 내에서 상기 심볼의 위치로부터 주어진 심볼에 대한 ESI를 결정할 수 있는 방식으로 순서화된다. 예를 들어, 상기 심볼이 소스 블록에 대한 심볼 순서 내에서 j번째 심볼인 경우, 이는 상기 심볼에 대한 ESI가 j인 경우일 수 있으며, 여기서 j는 양의 정수이다.

바람직하게는, 그러나 배타적이지 않게, 심볼들의 ESI들 및 심볼 순서 간의 매핑은 송신기 및 수신기 모두에 의해 용이하게 계산될 수 있다. 예를 들어, 심볼들의 순서화된 세트에 대한 연속적인 ESI들은 0, 1, 2, 3, ..., j, j+1 등일 수 있는데, 즉, 상기 ESI들은 제로에서 시작하는 연속적인 정수들이며 따라서 상기 심볼 위치는 이 경우 ESI와 동일하다. 또다른 예에서, 심볼들의 순서화된 세트에 대한 연속적인 ESI들은 5, 10, 15, 20, ..., 5*j, 5*(j+1) 등일 수 있다. 송신기 및 수신기 모두가 심볼 순서 내에서 심볼 위치가 주어진 경우 주어진 심볼에 대한 ESI를 결정하는 것을 허용하는 순서화된 심볼들의 세트로의 ESI들의 매핑을 결정하는 많은 다른 방법들이 존재한다. 바람직하게는, 송신기 및 수신기 모두에 의해 계산하기 쉬운 ESI 시퀀스는 소스 블록과 연관된 심볼들의 심볼 순서를 표현하는데 사용될 수 있다.

[0069] 물리층 블록 내의 물리층 패킷들

물리층 패킷들이 물리층 블록들 내에서 송신되는 경우, 물리층 블록 내의 물리층 패킷들의 순서화는 종종 전체 아키텍처의 특징들에 의해 결정될 수 있다. 또한, 하나의 물리층 블록의 또다른 물리층 블록과의 차별화는, 예를 들어, 타이밍 정보 및 물리층 시그널링에 기초하여, 송신기 및 수신기에 의해 결정될 수 있다. 순서화된 심볼들은 선형 합동(linear congruential) 매핑을 포함하는 다양한 상이한 방법들을 사용하여, 또는 연속적인 심볼들이 물리층 블록의 송신 내에서 시간 다양화된(time diversified) 방식으로 송신될 물리층 패킷들에 매핑되는 것을 보장하는 매핑을 사용하여 물리층 패킷들로 매핑될 수 있는데, 예를 들어, 각각의 연속적인 심볼이 물리층 블록의 송신의 상이한 시간 사분면(quadrant)에서 송신되는 물리층 패킷으로 매핑되거나, 또는 연속적인 심볼들이 주파수들의 매우 다양한(divergent) 세트들을 가지고 송신되는 물리층 패킷들로 매핑된다. 물리층 블록에서 송신될 심볼들의 순서화된 세트는 제 1 세그먼트 식별자와 연관된 심볼들로 구성될 수 있으며, 상기 제 1 세그먼트 식별자는 제 2 세그먼트 식별자와 연관된 심볼들에 선행하고, 상기 제 2 세그먼트 식별자는 제 3 세그먼트 식별자와 연관된 심볼들에 선행하는 등의 식이며, 여기서 세그먼트 식별자들의 전체 수는 하나 이상일 수 있다. 각각의 세그먼트 식별자와 연관된 심볼들 중에서, 상기 심볼들은 연속적으로 증가하는 ESI에 의해 순서화될 수 있다. 바람직한 특징은 물리층 내의 물리층 패킷들 및 순서화된 심볼들 사이의 매핑이 (명시적으로 또는 목시적으로) 공지되어 있으며 송신기들 및 수신기들에 의해 결정하기에 용이하다는 점이다.

이전에 설명되는 바와 같이, 심볼들은 다수의 서브-심볼들로 구성될 수 있으며, 각각의 물리층 패킷은 하나 이상의 서브-심볼들을 전달할 수 있지만 심볼을 전달할만큼 충분히 크지 않을 수 있다. 이들 경우들에서, 물리층 패킷들로의 심볼들의 매핑을 위한 방법들 및 프로세스들의 이전 설명은 이러한 것을 추가적으로 고려하도록 용이하게 수정될 수 있다. 예를 들어, ESI는 심볼들 뿐만 아니라 심볼 내의 특정 서브 심볼을 식별하도록 수정될 수 있는데, 예를 들면, ESI는 심볼 및 서브-심볼 식별자 모두이다. 또다른 예로서, 상기 매핑은 심볼의 서브-심볼들이 항상 연속적으로 송신되도록 될 수 있으며, 순서화된 심볼들로부터 물리층 패킷으로의 매핑은 상기 심볼의 제 1 서브-심볼을 전달하는 물리층 패킷을 식별한다.

일부 경우들에서, 시그널링 데이터 중 많은 양, 예를 들면, 물리층 블록들 내의 물리층 패킷들의 위치로부터 심볼 순서화에서의 심볼들의 위치 또는 심볼들의 ESI들을 유도하는 성능, 물리층 블록 식별자, 및 물리층 블록 해더 정보 내에서 전달되는 다른 정보가 물리층 블록들에서 사용가능할 수 있다.

[0073]

본 발명의 일부 실시예들에서, 하나의 심볼, 즉 소스 또는 복원 심볼이 헤더를 식별하는 헤더의 최소량과 함께 각각의 물리층 패킷에서 전달된다. 소스 블록에 대한 순서화된 심볼들의 세트가 송신기와 수신기 모두에 공지된 프로세스를 사용하여 물리층 블록 내에서 물리층 패킷들에 순차적으로 매핑된다. 예를 들어, 순서화된 512개의 심볼들의 세트는 512개의 물리층 패킷들에 순차적으로 매핑될 수 있다. 심볼들의 순서화는 송신기에서 결정될 수 있고, 대역 밖에서 명시적으로, 또는 바람직하게는 각각의 블록에 대한 심볼들의 순서를 결정하는 미리-결정된 프로세스들을 통해 송신기와 수신기 사이에서 묵시적으로 수신기에 전달될 수 있다. 둘 이상의 소스 블록으로부터의 심볼들이 동일한 물리층 블록 내에서 물리층 패킷들로 매핑될 예정인 경우, 소스 블록들이 순서화된다면, 소스 블록들의 순서화와 함께 각각의 소스 블록에 대한 심볼들의 순서화는 물리층 블록 내의 물리층 패킷에 매핑될 모든 심볼들의 순서를 결정하는데 사용될 수 있다. 다른 실시예들에서, 다수의 심볼들이 각각의 물리층 패킷에서 전달된다. 또 다른 실시예들에서, 예를 들어 심볼들이 서브-심볼들로 분할되고, 각각의 서브-심볼이 물리층 패킷 내에서 전달될 때, 심볼은 둘 이상의 물리층 패킷에 걸쳐 있을 수 있다. 당업자가 인지할 바와 같이, 여기서 설명되는 프로세스들 및 방법들 역시 다른 실시예들에 적용될 수 있다.

[0074]

일부 실시예들에서, 물리층 블록은 상이한 층의 블록, 예를 들어, 논리 블록 또는 데이터, 애플리케이션 정의된 데이터 블록, 또는 전송 블록, 또는 미디어층 블록일 수 있다. 또한, 물리층 패킷들은 전송 패킷들, 또는 논리 패킷들, 또는 애플리케이션 패킷들, 또는 미디어층 패킷일 수 있다. 당업자가 이해할 바와 같이, 이를 실시예들의 많은 본질적으로 동일한 변형들이 존재한다.

[0075]

세그먼트들

[0076]

소스 블록과 연관된 소스들 및 복원 심볼들은 둘 이상의 물리층 블록 내에서 송신될 수 있다. 소스 또는 복원 심볼의 세그먼트 식별자는 소스 블록에 대한 임의의 심볼들을, 바람직하게는 역순서로 전달하는 제 1 물리층 블록에 대해, 상기 심볼이 어느 물리층에서 전달되는지를 표시하는데 사용될 수 있다. 예를 들어, 소스 블록에 대한 임의의 심볼들을 전달하는 마지막 물리층 블록에서 전달되는 소스 블록과 연관된 모든 심볼들은 세그먼트 식별자 0을 가질 수 있는 반면, 각각의 이전 물리층 블록 내의 소스 블록과 연관된 모든 심볼들에 대한 세그먼트 식별자는 상기 소스 블록에 대한 임의의 심볼들을 전달하는 후속적인 물리층 블록 내의 세그먼트 식별자보다 1 더 큰 세그먼트 식별자를 가질 수 있다. 모든 연속적인 물리층 블록들이 특정 소스 블록에 대한 심볼들을 전달하는 물리층 블록들 중 상기 소스 블록에 대한 심볼들을 전달할 수 있지는 않은데, 예를 들면, 제 1 물리층 블록은 소스 블록에 대한 심볼들을 전달할 수 있고, 다음 제 2 물리층 블록은 상기 소스 블록에 대한 어떠한 심볼들도 전달하지 않을 수 있는 반면, 그 다음 제 3 물리층 블록은 상기 소스 블록에 대한 심볼들을 전달할 수 있다는 점에 유의한다. 다른 경우들에서, 소스 블록의 세그먼트 구조는 물리층 패킷 순서 내의 물리층 패킷 위치 또는 하나의 소스 블록에 대한 하나의 세그먼트의 종단 및 또 다른 소스 블록에 대한 새로운 세그먼트의 시작을 표시하는 세그먼트 경계 표시자인 물리층 블록을 표시함으로써 시그널링될 수 있다. 예를 들어, 2000개의 물리층 패킷들을 가지는 물리층 블록에 대해, 제 1의 500개의 물리층 패킷들은 제 1 소스 블록으로부터의 세그먼트에 대응하며, 그 다음 600개의 물리층 패킷들은 제 2 소스 블록으로부터의 세그먼트에 대응하며, 나머지 900개의 물리층 패킷들은 제 3 소스 블록으로부터의 세그먼트에 대응하며, 세그먼트 경계 표시자들 500, 600은 제 1 소스 블록의 세그먼트가 제 1의 500개의 물리층 패킷들에 대응하고, 제 2 소스 블록의 세그먼트는 그 다음 600개의 물리층 패킷들에 대응하고, 제 3 소스 블록의 세그먼트는 나머지 900개의 물리층 패킷들에 대응함을 표시하는데 사용될 수 있다. 대안적으로, 세그먼트 경계 식별자들은 심볼들의 유닛들로 표시될 수 있으며, 물리층 블록 내의 심볼들의 순서화에 대해 결정될 수 있다.

[0077]

일부 바람직한 실시예들에서, 각각의 물리층 블록 내에서, 기껏해야 각각의 세그먼트 식별자들과 연관된 하나의 소스 블록이 존재하며, 따라서, 세그먼트 식별자는 상기 심볼들을 상이한 소스 블록들과 고유하게 구분하기 위해 사용될 수 있으며, 그 결과, 이 경우들에서, 세그먼트 식별자 역시 물리층 블록 내에서 전달되는 심볼들을 구분하기 위해 소스 블록 식별자로서 사용된다. 다른 실시예들에서, 심볼들에 대한 소스 블록 식별자들은 다른 수단들에 의해, 예를 들어, 각각의 심볼과 연관된 헤더 데이터 내에 소스 블록 식별자를 포함시킴으로써, 또는 각각의 물리층 블록과 연관된 헤더 데이터 내에 소스 블록 식별자를 포함시킴으로써 전달된다. 소스 블록 식별자가 물리층 블록들의 헤더들에서 반드시 전달되는 것이 아니라, 대신 다수의 물리층 블록들에 대한 헤더 정보를 포함하는 별개의 물리층 블록 내의 다른 장소들, 예를 들어, 제어 데이터 스트림에서 전달되거나, 또는 또 다른 네트워크를 통해 송신될 수 있는 다른 변형들이 존재한다. 당업자는 많은 다른 유사한 변형들을 인지할 것이다.

[0078]

서브-블록들

[0079]

인코딩된 또는 인코딩되지 않은 소스 블록은 둘 이상의 서브-블록을 포함할 수 있는데, 예를 들어, 서브-블록들은 심볼들의 논리적으로 연관된 부분들에 대응하는 소스 블록과 연관된 상이한 소스 및 복원 심볼들에 대응한다. 예를 들어, 제 1 서브-블록을 포함하는 소스 및/또는 복원 심볼들의 제 1 세트는 소스 블록과 연관된 비디오의 일부분의 고해상도 비디오를 렌더링하는데 사용될 수 있는 소스 블록의 일부분에 대응할 수 있는 반면, 제 2 서브-블록을 포함하는 소스 및/또는 복원 심볼들의 제 2 세트는 상기 제 1 서브-블록의 일부 또는 전부와 연관되어 사용될 때 소스 블록과 연관된 비디오의 일부분의 고해상도 비디오를 렌더링할 수 있다. 또 다른 예로서, 서브-블록 식별자는 소스 블록과 연관된 복원 심볼들 중 일부 또는 전부를 식별할 수 있으며, 서브-블록 식별자는 소스 블록과 연관된 소스 심볼들 중 일부 또는 전부를 식별할 수 있다. 일부 경우들에서, 서브-블록 식별자는 각각의 서브-블록을 번호로 명시적으로 라벨링함으로써 시그널링될 수 있다. 예를 들어, 소스 블록의 제 1 서브-블록은 서브-블록 식별자 0을 가질 수 있는 반면, 소스 블록의 제 2 서브-블록은 서브-블록 식별자 1을 가질 수 있다. 다른 경우들에서, 서브-블록 구조는 예를 들어, ESI 또는 심볼 순서 내의 하나의 서브-블록의 종단과 새로운 서브-블록의 시작을 표시하는 서브-블록 경계 표시자인 심볼 순서 내의 ESI 또는 심볼 위치를 표시함으로써 시그널링될 수 있다. 예를 들어, 900개의 소스 심볼들 및 100개의 복원 심볼들을 가지는 소스 블록에 대해, 상기 심볼들의 ESI들은 제로에서 시작하는 연속적인 정수들이며, 제 1 서브-블록은 소스 심볼들을 포함하고 제 2 서브-블록은 복원 심볼들을 포함하고, 상기 서브-블록 경계 표시자(900)는 0 내지 899까지의 ESI들을 가지는 심볼들에 대응하고 제 2 서브-블록은 ESI 900을 가지는 심볼로 시작하는 것을 표시하기 위해 사용될 수 있다. 소스 또는 복원 심볼의 서브-블록 표시자는 상기 심볼이 어느 서브-블록의 일부분인지를 표시한다.

[0080]

각각의 심볼을 가지고 송신되는 헤더 데이터 방법

[0081]

일 실시예에서, 물리층 패킷 내의 심볼과 함께 송신될 각각의 심볼과 연관된 헤더 데이터는 세그먼트 식별자, 서브-블록 식별자 및 ESI를 포함한다. 예를 들어, 가능한 세그먼트 식별자들의 수가 8이고 가능한 서브-블록 식별자들의 수가 8이고, ESI들의 수가 1024인 경우, 헤더 데이터의 16비트 또는 등가적으로 2바이트가 각각의 심볼에 대해 충분하다. 물리층 블록 내의 각각의 물리층 패킷 내에서, 물리층 패킷의 컨텐츠들은 상기 심볼과 연관된 헤더 데이터와 함께 심볼을 포함하며, 상기 헤더 데이터는 세그먼트 식별자 및 서브-블록 식별자를 포함한다.

[0082]

이 실시예에서, 수신기는 물리층 블록 내에서 수신된 물리층 패킷들을 다음과 같이 프로세싱할 수 있다. 물리층 블록 내의 물리층 패킷들의 수신시, 수신기는 그것이 판독할 수 있는 각각의 물리층 패킷 내의 심볼과 연관된 헤더 데이터로부터 결정할 수 있다. 헤더 데이터에 대해, 수신기는 물리층 패킷 내에 포함되는 심볼에 대한 세그먼트 식별자, 서브-블록 식별자 및 ESI를 결정할 수 있다. 세그먼트 식별자로부터, 수신기는 상기 심볼이 가능한 소스 블록들 중 어느 소스 블록과 연관되는지를 결정할 수 있다. 서브-블록 식별자로부터, 상기 수신기는 상기 심볼이 소스 블록의 가능한 서브 블록들 중 어느 서브-블록과 연관되는지를 결정할 수 있다. 상기 ESI로부터, 수신기는 상기 심볼 대 소스 블록 및 상기 심볼 대 상기 소스 블록의 서브-블록의 관계를 결정할 수 있으며, 상기 ESI는 소스 블록 내의 심볼들의 심볼 위치를 결정하기 위해 사용될 수 있으며, 그리고/또는 수신된 복원 심볼들 및 다른 소스 심볼들로부터 유실된 소스 심볼들로부터 복원하기 위해 FEC 디코딩시 사용될 수 있다.

[0083]

그 후, 수신기는, 이 정보에 기초하여, 특정 동작들에 대해 결정할 수 있다. 예를 들어, 상기 수신기는 상이한 목적들로 심볼들과 연관된 서브-블록 데이터를 사용할 수 있다. 예를 들어, 상기 서브-블록 데이터는 소스 블록의 일부 또는 전부를 복원하기 위해 FEC를 디코딩하는 방법을 결정하기 위해 부분적으로 사용될 수 있다. 상기 서브-블록 데이터는 또한 수신기 내의 상위 레벨 기능을 지원하기 위해, 상위계층 애플리케이션, 예를 들어, 수신기 내의 멀티미디어 플레이어 프로세스에 대해 데이터의 어느 부분들이 전달되어야 하는지를 결정하기 위해, 예를 들면, 복원된 소스 블록의 어느 부분들이 멀티미디어의 재생을 위해 멀티미디어 플레이어에 그 전체가 전달되어야 하는지를 결정하기 위해 사용될 수 있다. 예를 들어, 수신기가 제 1 물리층 블록을 수신할 때, 제 1 세그먼트 식별자와 연관된 심볼들의 일부분은 상기 제 1 세그먼트 식별자와 연관된 소스 블록과 연관된 저품질 비디오 부분의 재생을 위해 멀티미디어 플레이어로 전달될 수 있는 제 1 서브-블록과 연관될 수 있다. 상기 수신기는 또한 후속적인 물리층 블록들에서 수신되는 동일한 소스 블록들에 대한 심볼들과 추출된 그리고/또는 복원된 심볼들을 결합시키고, FEC 디코딩하고 그리고 또는 미디어 플레이어로 전달하도록, 이들을 아마 심볼들의 서브-블록들의 유닛들로 또는 심볼들의 서브 블록들의 세트들로 결합시키기 위해, 제 1 세그먼트 식별자가 아닌 세그먼트 식별자들을 가지는 소스 블록들과 연관된 상기 추출된 그리고/또는 복원된 심볼들을 저장하도록 결정할 수도 있다.

- [0084] 당업자가 인지할 바와 같이, 상기 실시예들의 변형들 및 결합들이 존재한다. 예를 들어, 심볼과 함께 송신되는 상기 심볼에 대한 헤더 데이터는 ESI가 아닌 서브-블록 식별자들 및 세그먼트 식별자들을 포함할 수 있다. 변형의 또 다른 예로서, 오직 ESI만이 심볼과 함께 헤더 데이터, 및 다른 데이터로부터 결정되는 것으로 사용되는 경우, 세그먼트 식별자 또는 서브-블록 식별자와 같은 다른 데이터로 송신될 수 있다.
- [0085] 변형의 또 다른 예로서, 각각의 심볼과 연관된 헤더 데이터는 서브-블록 데이터를 포함하지 않을 수도 있다. 이 경우, 서브-블록 식별자는 예컨대 유도된 ESI들에 의해 묵시적으로 결정될 수 있거나, 또는 서브-블록들은 소스 블록의 세그먼트들과 일치하거나, 또는 서브-블록킹이 사용되지 않는다.
- [0086] 변형의 또 다른 예로서, 각각의 심볼과 연관된 헤더 데이터는 세그먼트 식별자를 포함할 수도 있고 포함하지 않을 수도 있다. 이 경우, 세그먼트 식별자는 예를 들어, 각각의 물리층 블록 내에서 고정된 양의 물리층 패킷들을 할당함으로써 묵시적으로 결정될 수 있거나, 또는 서브 블록들은 세그먼트들과 일치하거나, 또는 세그먼트화가 사용되지 않는다.
- [0087] 변형의 또 다른 예로서, 각각의 심볼과 연관된 헤더 데이터는 또한 스트림 식별자를 포함할 수 있다. 이 경우, 스트림 식별자는 심볼이 소수의 스트림들, 예를 들어, 오디오 스트림 또는 비디오 스트림 중 어느 스트림과 연관되는지를 결정할 수 있다. 스트림 식별자는 다른 식별자들에 의해 범위가 정해질 수 있는데(scope), 예를 들어, 상기 스트림들, 예를 들어, 동일한 스트림들에 대한 오디오 및 비디오 스트림들이 논리적으로 연결되는 경우, 예를 들어, 서브-블록 식별자는 스트림 식별자들 중 일부 또는 모두를 범위로 할 수 있다는 점에 유의한다. 스트림 식별자 역시 다른 식별자들을 범위로 할 수 있는데, 예를 들어, 상기 스트림들이 예컨대 상이한 프로그램 세그먼트들에 대한 오디오/비디오 스트림들과 같이 논리적으로 독립적인 경우, 예를 들어, 스트림 식별자는 상기 서브-블록 식별자들의 일부 또는 전부를 범위로 할 수 있다는 점에 유의한다.
- [0088] 각각의 심볼을 가지고 송신되는 헤더 데이터가 없는 방법
- [0089] 또 다른 실시예에서, 물리층 패킷에서 전달되는 심볼과 연관된 어떠한 헤더 데이터도 존재하지 않는다. 대신, 일부 최소 데이터는 물리층 블록의 헤더 데이터 내에서 전달될 수 있다. 상기 최소 데이터는, 예를 들어, 세그먼트 테이블을 포함할 수 있는데, 여기서 세그먼트 테이블의 각각의 행은 상기 물리층 블록에서 전달되는 소스 블록에 대한 세그먼트의 모든 심볼들 중 소스 블록의 상기 세그먼트에 대해 심볼 순서에서 첫번째 심볼의 ESI 및 상기 물리층 블록에서 전달되는 소스 블록에 대한 세그먼트의 다수의 심볼들을 포함하는 세그먼트 식별자에 대응한다. 상기 세그먼트 내의 심볼들의 번호는, 예를 들어, 각각의 세그먼트 내의 심볼들의 수가 모든 물리층 블록들에 대해 항상 동일하므로 일부 실시예들에서 포함되지 않을 수 있다.
- [0090] 일부 실시예들에서, 세그먼트 테이블은, 동일한 세그먼트 식별자가 동일한 물리층 블록 내의 둘 이상의 소스 블록들에 대해 사용될 경우들에 있어서, 소스 블록 테이블 대신일 수 있다.
- [0091] 최소 데이터는 또한, 예컨대 서브-블록 테이블을 포함할 수 있는데, 이는 각각의 소스 블록에 대한 심볼들이 물리층 블록 내의 어느 서브-블록들에서 전달되는지를 표시한다. 이 서브-블록 테이블에 대한 많은 형태들이 존재하는데, 예를 들어, 서브-블록 정보는 세그먼트 테이블 내의 적절한 세그먼트 식별자 행의 각각의 행에 첨부(append)될 수 있거나, 또 다른 예로서, 서브-블록 정보는 개별 테이블에 저장될 수 있다. 일부 실시예들에서, 서브-블록 테이블은, 예컨대 서브-블로킹이 사용되지 않기 때문에 또는 서브-블로킹 시그널링이 더 상위의 애플리케이션 층에서 핸들링(handle)되지 않기 때문에 포함되지 않을 수 있다.
- [0092] 이 실시예에서, 수신기는 물리층 블록 내의 수신된 물리층 패킷들을 다음과 같이 프로세싱할 수 있다. 수신기는 물리층 블록 헤더 데이터로부터 서브-블록 테이블 및/또는 세그먼트 테이블을 판독 및 저장한다. 세그먼트 테이블로부터, 수신기는 심볼들의 수 및 물리층 블록과 함께 전달되는 심볼들이 존재하는 소스 블록의 각각의 세그먼트와 연관된 초기 ESI를 결정할 수 있다. 심볼을 전달하는 물리층 패킷의 위치의 물리층 식별로부터, 각각의 세그먼트와 연관된 초기 ESI 및 수를 포함하는 세그먼트 테이블로부터, 그리고 물리층 패킷들에 대한 물리층 블록에 포함되는 소스 블록의 모든 세그먼트들로부터, 수신기는 심볼의 ESI 및 심볼이 어느 소스 블록과 연관되는지를 결정할 수 있다. 서브-블록 테이블로부터, 유사한 방식으로, 수신기는 상기 심볼이 소스 블록의 어느 서브-블록과 연관되는지를 결정할 수 있다.
- [0093] ESI로부터, 수신기는 상기 심볼 대 상기 소스 블록의 관계 및 상기 심볼 대 상기 소스 블록의 서브-블록의 관계를 결정할 수 있으며, 여기서 상기 ESI는 소스 블록 내의 심볼들의 심볼 위치를 결정하는데 사용될 수 있고, 그리고/또는 수신된 복원 심볼들로부터 수신되지 않은 소스 심볼들 및 다른 소스 심볼들을 복원하기 위해 FEC 디코딩시에 사용될 수 있다.

- [0094] 그후 수신기는, 이 정보에 기초하여, 여기서 설명되는 "각각의 심볼을 가지고 송신되는 헤더 데이터" 방법에 대해 전술된 내용을 포함하여, 특정 동작들에 대해 결정할 수 있다.
- [0095] 당업자가 이해할 바와 같이, 상기 내용의 많은 변형들이 존재한다. 변형의 일 예로서, 각각의 심볼과 연관된 헤더 데이터는, 예를 들어, 이 목적을 위해 각각의 물리층 패킷의 1바이트 부분을 사용하는, 서브-블록 식별자를 포함할 수 있다. 이는, 서브-블록 구조가 전체 소스 블록에 걸쳐 있는 반면 소스 블록에 대한 데이터의 송신이 몇몇 물리층 블록들에 걸쳐 있을 수 있어서, 결과적으로 각각이 심볼을 가지고 송신되는 헤더 데이터 내의 서브-블록 식별자의 전달은 소스 블록의 전송 중에 채널을 조인시키는 수신기로 하여금 소스 블록의 서브-블로킹을 신속하게 이해하게 할 수 있는, 일부 경우들에서 바람직할 수 있다.
- [0096] 또다른 예로서, 서브-블로킹이 사용되지 않을 수 있다.
- [0097] 또다른 예로서, 각각의 물리층 패킷과 연관된 헤더 데이터는 예를 들어, 동일한 물리층 블록 내에서 개별 데이터로서 송신될 수 있거나 다른 수단에 의해 수신기로 전달될 수 있는데, 예를 들면, 상기 수신기에 대해 사용 가능한 제어 채널 내에서 송신되거나, 또는 또다른 예로서, 다수의 물리층 블록에 대한 헤더 정보를 포함하는 개별 물리층 블록에서 송신되거나, 또는 또다른 예로서 또다른 네트워크를 통해 송신될 수 있다.
- [0098] 또다른 예로서, 각각의 심볼과 연관된 헤더 데이터는 또한 스트림 식별자를 포함할 수 있다. 이 경우에서, 스트림 식별자는 심볼이 소수의 스트림들, 예를 들어, 오디오 스트림 또는 비디오 스트림 중 어느 스트림과 연관되는지를 결정할 수 있다. 스트림 식별자는 다른 식별자들에 의해 범위가 정해질 수 있는데, 예를 들어, 스트림들, 예를 들어, 동일한 프로그램 세그먼트에 대한 오디오 및 비디오 스트림들이 논리적으로 연결되는 경우, 예를 들어, 서브-블록 식별자가 스트림 식별자들 중 일부 또는 전부를 범위로 할 수 있다는 점에 유의한다. 스트림 식별자 역시 다른 식별자들을 범위로 할 수 있는데, 예를 들어, 상이한 프로그램 세그먼트들에 대한 오디오/비디오 스트림과 같은 상기 스트림들이 논리적으로 독립적인 경우, 예를 들어, 스트림 식별자는 서브-블록 식별자들 중 일부 또는 전부를 범위로 할 수 있다는 점에 유의한다. 스트림 식별자 역시, 세그먼트 식별자들 및 서브-블록 식별자에 대해 전술된 것과 유사한 포맷으로 물리층 블록에 대한 헤더 데이터 내에 포함될 수 있으며, 어느 경우든, 수신기로 스트림 구조를 전달하기 위해 각각의 심볼과 연관된 헤더 데이터 내에 스트림 식별자를 포함시키는 것이 필수적이지는 않을 수도 있다.
- [0099] 일예로서, 소스 블록당 세그먼트들의 수가 4이고, 서브-블록들의 수가 3이고, 물리층 블록 당 물리층 패킷들의 수가 512이고, 각각 사이즈 100바이트인 3개의 심볼들이 300바이트의 각각의 물리층 패킷에 포함된다고 가정하면, 그 결과 각각의 물리층 블록은 $3 \times 512 = 1536$ 개의 심볼들을 포함한다. 이후, 특정 제 1 물리층 블록에 대한 제 1 세그먼트 테이블 및 제 2 물리층 블록에 대한 제 2 세그먼트 테이블은 도 3에 도시된 바와 같을 수 있으며, 여기서 제 2 물리층 블록은 제 1 물리층 블록 다음에 연속적으로 송신된다. 이 예에서, 세그먼트 식별자는 세그먼트 테이블에서 명시적으로 전달되지 않을 수 있지만, 대신 테이블 내의 행 번호에 의해 포함될 수 있는데, 즉, 행 j 는 세그먼트 식별자 j 에 대응한다.
- [0100] 제 1 세그먼트 테이블에서, 식별자 0을 가지는 세그먼트에 대한 심볼들의 수는 450이며, 이는 150개의 물리층 패킷들에 의해 전달될 것이며, 상기 제 1의 450개의 심볼들은 순서화된 심볼들 대 물리층 패킷 매핑에 따라 매핑된다. 세그먼트 식별자 0을 가지는 심볼들에 대한 ESI들은 이 예에서는 0 내지 449의 범위를 가지는 연속적인 정수들이다. 식별자 1을 가지는 세그먼트에 대한 심볼들의 수는 300이며, 이는 제 1의 150개의 물리층 패킷들 다음의 100개의 물리층 패킷들에 의해 전달될 것이며, 상기 300개의 심볼들은 순서화된 심볼들 대 물리층 패킷 매핑에 따라 매핑된다. 세그먼트 식별자 1을 가지는 심볼들에 대한 ESI들은 이 예에서 420 내지 719의 범위를 가지는 연속적인 정수들이다.
- [0101] 제 2 세그먼트 테이블에서, 식별자 0을 가지는 세그먼트에 대한 심볼들의 수는 420이며, 이는 140개의 물리층 패킷들에 의해 전달될 것이며, 상기 제 1의 420개의 심볼들은 순서화된 심볼들 대 물리층 패킷 매핑에 따라 매핑된다. 제 1 세그먼트 테이블에서 세그먼트 식별자 j ($j=0,1,2$)를 가지는 소스 블록은 제 2 세그먼트 테이블에서 세그먼트 식별자 $j+1$ 을 가지는 소스 블록과 동일할 수 있다는 점에 유의한다. 따라서, 제 1 세그먼트 테이블에서 식별자 j 를 가지는 세그먼트에 대한 초기 ESI는 이러한 매핑하에서, 초기 ESI 및 제 2 세그먼트 테이블에서 식별자 $j+1$ 을 가지는 세그먼트의 심볼들의 수의 합이 된다.
- [0102] 데이터가 물리층 블록들의 헤더들에서 반드시 전달되는 것이 아니라, 대신, 다른 장소들, 예를 들어, 다수의 물리층 블록들에 대한 헤더 정보를 포함하는 개별 물리층 블록 내의 제어 데이터 스트림에서 전달될 수 있거나, 또는 다른 네트워크를 통해 송신되는, 다른 변형들이 존재한다. 당업자라면, 전술된 방법의 많은 다른 유사한

변형들을 인지할 것이다.

[0103] FEC 페이로드 ID로 그리고 FEC 페이로드 ID로부터의 매핑들

FEC 코드들이 표준으로 기술되는 예를 들어, IETF RFC 5052 (Internet Engineering Task Force Request for Comments 5052) 및 IETF RFC 5053 (Internet Engineering Task Force Request for Comments 5053)에서 기술되는 바와 같이, 많은 애플리케이션층에 대해, 통상적으로, FEC 페이로드 ID(식별자)가 심볼 또는 심볼들의 그룹 또는 애플리케이션층 패킷에서 송신되는 서브-심볼들의 그룹과 연관된다. 가장 간단한 경우에 대해, FEC 페이로드 ID가 심볼과 연관되는 경우, FEC 페이로드 ID는 심볼이 생성되었던 소스 블록 번호, 상기 심볼의 ESI 및, 일부 경우들에서는, 가장 작게 연관된 ESI를 가지는 복원 심볼의 초기 ESI를 포함한다(그리고 이러한 초기 ESI는, 상기 소스 심볼들이 제1 서브-블록의 일부분이며 상기 복원 심볼을 제 2 서브-블록의 일부분으로서 식별하는 서브-블록 식별자로서 보여질 수 있다).

전술된 방법들 및 프로세스들의 일부에 있어서, FEC 페이로드 ID는 각각의 심볼과 함께 송신되지 않으며, 대신 채널 용량을 최대화하기 위해 각각의 심볼과 함께 송신되는 헤더 데이터의 양을 최소화시키는 다른 수단들이 설명된다. 일부 경우들에 있어서, 송신기에서 FEC 페이로드 ID를 사용하는 것으로부터 수신기로 이 정보를 전달하기 위해 전술된 수단을 사용하는 것으로 송신 포맷을 변환하는 것이 유용하다. 또한, 일부 경우들에 있어서, 수신기에서, 상기 수신기로 이 정보를 전달하기 위해 전술된 수단을 사용하는 것으로부터 FEC 페이로드 ID를 사용하는 것으로 송신 포맷을 변환하는 것이 유용하다. 예를 들어, 심볼들을 식별하기 위해 FEC 페이로드 ID를 사용하는 이미 개발된 소프트웨어가 존재할 수 있으며, 전술된 수단을 사용하는 송신 포맷과 호환가능한 연관된 데이터 및 심볼들의 출력 스트림을 산출하기 위해 이 소프트웨어를 사용하여 생성된 연관된 헤더 데이터 및 심볼들의 출력 스트림을 취하는 것이 편리할 수 있다.

[0106] FEC 페이로드 ID 포맷으로 그리고 FEC 페이로드 ID 포맷으로부터의 매핑 방법들은 위에서 제공된 설명으로부터 용이하게 도출될 수 있다.

[0107] 채널 재핑을 최적화하는 배열들의 송신

채널을 통해 송신될 우선순위화된 스트림에 대해, 송신될 데이터는 상이한 물리층 블록들, 예를 들어, 프레임들 또는 수퍼-프레임들로 분할되며, 소스 블록에 대해 송신될 심볼 데이터는 다수의 이러한 물리층 블록들에 대해 우선순위화된 방식으로, 이들의 우선순위의 역순서로 인터리빙될 수 있다. 예를 들어, "FEC 스트리밍"으로 설명되는 바와 같이, 소스 블록에 대한 복원 데이터는 이를 설명들의 상황에서, 채널 재핑 시간을 감소시키기 위해 소스 블록에 대한 소스 데이터 이전에 송신될 수 있다. 소스 블록에 대해 주어진 우선순위 레벨인 데이터를 포함하는 데이터는 함께 서브-블록으로 그룹화될 수 있다. 예를 들어, 전술된 예에 계속하여, 복원 심볼들이 더 낮은 우선순위의 서브-블록들로 고려되고, 그리고 소스 심볼들이 두번째로 높은 우선순위의 서브-블록로 고려될 수 있으며, 따라서, 더 낮은 우선순위의 서브-블록이 더 높은 우선순위의 서브-블록 이전에 송신될 수 있다.

[0109] 도 4는 실시예가 데이터를 서브-블록들로 우선순위화하고 상기 서브-블록들을 우선순위화된 송신 순서로 매핑할 수 있는 방법의 예를 예시한다. 도 4에서, 데이터 스트림(470)은 데이터의 다양한 블록들 및 서브-블록들로 표현된다. 예를 들어, 데이터 스트림(470)은 오디오 블록(450) 및 I-Frame(ZI)(410)과 같은 다양한 비디오 블록들 및 P₁-P_x(420-422), b₁-b_z(430-435), 및 B₁-B_y(440-442)과 같은 다양한 심볼 데이터의 서브-블록들로 도시된다. 도 4에서, P₁(420)은 각각 b₁-b_z(430-435), B₁-B_y (440-442), P₂-P_x(421-422), 오디오 블록(450) 및 I-프레임(ZI)(410)에 선행하는, 스트림 내의 가장 높은 우선순위의 서브-블록을 표현한다. 이를 우선순위 레벨들이 주어지면, 스트림의 블록들 및 서브-블록들은 송신 배열(480)에 의해 예시되는 바와 같이 정렬될 수 있다. 가장 낮은 우선순위 블록(ZI)(410)은 전송 시작시 수신기에 의해 전송될 수 있는 반면, 가장 높은 우선순위 데이터(P₁)(420)는 마지막으로 송신될 수 있다. 추가적으로, 다양한 서브-블록들 사이의 종속성들 역시 우선순위화된 송신 순서를 생성할 때 고려될 수 있다. 예를 들어, 일부 실시예들에 따라, 서브 블록들(b₁ 및 b₂)는 P₁에 종속적일 수 있다. 이를 실시예들에서, P₁이 전송되기 전에 이를 종속적인 서브-블록들을 전송하는 것이 유리할 수 있다. 따라서, P₁이 수신되자마자, P₁ 내의 모든 데이터 및 그것의 종속적인 서브-블록들 모두는 수신기에서 신속하게 사용가능하게 될 수 있다. 송신 배열이 결정되면, 상기 송신 배열은 상기 데이터를 상이한 물리층 블록들로 적절하게 분할하는데 사용될 수 있다.

[0110] 우선순위화된 서브-블록들을 물리층 블록으로 매핑시키기 위한 한 방법은 서브-블록들을 각각의 물리층 블록으

로 매핑하는 것이다. 도 5는 이 방법의 일 실시예의 예시를 도시한다. 도 5는 다양한 물리층 블록들(501-504)로 분할된 데이터의 세트(500)를 도시한다. 도 5의 블록들은 화살표(509)에 의해 표시되는 방향으로 전송되는 것으로 표현된다. 예를 들어, 물리층 블록(501)은 물리층 블록(504) 앞에서 송신되고(따라서, 물리적 블록(504) 이전에 전송되고), 물리층 블록(501) 내에서, 섹션(580)은 섹션(520) 앞에서 전송된다. 도 5에 예시된 바와 같이, 데이터(500) 중 일부는 각각의 물리층 블록(501-504)에 위치된다. 명료함의 목적으로, 데이터(500) 내의 데이터의 각각의 세그먼트는, 각각의 세그먼트가 각각의 물리층 블록의 대응하는 섹션에 위치된다 할지라도, 물리층 블록들(501-504) 중 오직 하나에 위치되는 것으로 도시된다. FEC 데이터(510)는 520-523에서 물리층 블록들로 위치되고; P_1 데이터(420)는 540-543에서 물리층 블록들로 위치되고; b_1-b_2 데이터(430-435)는 530-533에서 물리층 블록들로 위치되고; B_1-B_y 데이터(440-442)는 550-553에서 물리층 블록들로 위치되고; P_2-P_x 데이터(421-422)는 560-563에서 물리층 블록들로 위치되고; 오디오 데이터(450)는 570-573에서 물리층 블록들로 위치되고; 그리고 I-프레임(ZI)(410)은 물리층 블록들(580-583)로 위치된다. 도 5에 예시되는 방식으로의 서브 블록들의 물리층 블록들로의 매핑의 한가지 이점은 각각의 우선순위 그룹의 세그먼트들이 각각의 물리층 블록 내에 포함될 것이기 때문에 수신기에서의 재생 동작이 더 예측가능할 것이라는 점이다. 그러나, 각각의 물리층 블록 내의 다양한 세그먼트들은 통상적으로 상이한 사이즈들일 것인데, 왜냐하면, 다양한 우선순위 레벨들이 통상적으로 상이한 양의 데이터를 포함할 것이기 때문이다. 이는 데이터를 패킹해제(unpack)하기 위한 수신기에서의 더 복잡한 프로세싱으로 인해 수신기에서의 잠재적인 성능 이슈들을 초래할 수 있으며, 상이한 세그먼트 사이즈들로 인해 stat-muxing을 가지는 이슈들이 존재할 수 있다.

[0111]

또 다른 방법은 상이한 물리층 블록에 대해 가능한 균일하게 심볼 데이터를 확산시키는 것인데, 왜냐하면 이것이 일반적으로 채널 손상들에 대한 최상의 보호를 제공하기 때문이다. 도 6은 이 방법의 일 실시예의 예시를 도시한다. 도 6은 다양한 블록들(601-604)로 분할되는 데이터의 세트(600)를 도시한다. 도 6의 블록은 화살표(609)에 의해 표시되는 방향으로 전송되는 것으로 표현된다. 예를 들어, 물리층 블록(601)은 물리층 블록(604) 이전에 전송되며(따라서, 물리적 블록(604) 이전에 전송되며), 물리층 블록(601) 내에서, 섹션(640)은 섹션(610) 이전에 전송된다. 도 6에 도시된 바와 같이, 심볼 데이터(600) 내의 다양한 데이터 특징들은 블록들(605-608) 내에 함께 그룹화되었다. 이들 블록들(605-608)은 이후 동일한 양들의 물리층 블록들(601-604)로 매핑되었다. 명료성의 목적으로, 데이터(600)의 각각의 세그먼트는, 각각의 세그먼트가 각각의 물리층 블록의 대응하는 섹션으로 위치된다 할지라도, 물리층 블록들(601-604) 중 하나로만 위치되는 것으로 도시된다. 예를 들어, 블록(605)은 610-613으로 매핑되고, 블록(606)은 620-623으로 매핑되고, 블록(607)은 630-633으로 매핑되고, 블록(608)은 640-643으로 매핑된다. 도 6에 예시된 매핑의 결과로서, 일부 서브-블록들은 그룹들 사이에서 분할된다. 예를 들어, 데이터 세그먼트들(B_1-B_y)로부터의 데이터(440-442)는 블록들(606 및 607) 모두에 포함될 수 있다. 추가적으로, 주어진 물리적 블록은 특정 우선순위로부터의 어떠한 데이터도 포함하지 않을 수 있다. 예를 들어, 블록(601)은 610에서 어떠한 FEC 데이터(510)도 포함하지 않을 수 있지만, 블록(604)은 613에서 P_1 으로부터의 어떠한 데이터도 포함하지 않을 수 있다. 도 6에 예시된 방법의 한가지 이점은 물리층 블록들의 세그먼트가 동일한 사이즈이므로, 수신기가 세그먼트들을 패킹해제하기 위해 더 적은 프로세싱을 요구할 것이라는 점이다. 이는 개선된 수신기 성능을 초래할 수 있다. 추가적으로, 균일한 세그먼트 사이즈는 stat-muxing을 더 용이하게 한다. 그러나, 임의의 주어진 물리층 블록에 포함될 정확한 우선순위 레벨들에 대한 어떠한 보장들도 존재하지 않을 수 있으므로, 수신기에서의 재생 동작은 덜 예측가능할 것이다.

[0112]

데이터 매핑동안의 한가지 관심사는 소스 블록에 대한 높은 우선순위 데이터 중 충분한 양이 이러한 높은 우선순위 데이터가 수신되자마자 수신기로 하여금 재생을 시작하게 하기 위해 제 1 물리층 블록에서 전송된다는 점이다. 이를 달성하기 위한 한가지 방법은 높은 우선순위 데이터의 양이 기껏해야 소스 블록에 대해 송신될 데이터의 전체 양의 $1/N$ 부분이 되게 하는 방식으로 인코딩된 또는 인코딩되지 않은 소스 블록들 내의 데이터를 우선순위화 하는 것이며, 여기서 N 은 데이터가 소스 블록에 대해 전송될 동안의 물리층 블록들의 수이며, 상기 경우에서, 일부 소스 블록에 대한 높은 우선순위 데이터는 수신기가 제 1 물리층 블록을 수신한 이후 사용 가능해져야 한다. 일반적으로, 상기 요건이 데이터의 제 1 J개의 특징들이 수신기가 K개의 물리층 블록들을 수신한 이후 일부 제 1 소스 블록에 대해 사용 가능해질 필요가 있는 것인 경우, 이는 상기 제 1 J개의 특징들 내의 데이터의 부분이 기껏해야 K/N 인 경우 달성될 수 있다.

[0113]

바람직한 분할 전략의 예는 다음과 같은데, 이는 상기 방법이 적용되든 적용되지 않든 간에 사용될 수 있다. 소스 블록에 대해 송신되는 데이터가 N개의 물리층 블록들 내에서 송신될 것이라고 가정하되, 여기서 상기 송신되는 데이터는 소스 블록에 대한 소스 심볼들 및 만약 존재한다면 송신될 소스 블록으로부터 생성되는 FEC 복원

심볼들을 포함한다. 소스 블록에 대해 송신되는 데이터가 K개의 우선순위들로 분할된다고 가정하되, 여기서 우선순위 j를 가지고 송신되는 데이터의 비는 P_j이며, $j = 1, \dots, K$ 이다.

[0114] 전술된 바와 같이, 우선순위 j를 가지고 송신되는 데이터는 서브-블록 j라 지칭되는 서브-블록으로 그룹화될 수 있다. 이후, 마지막 물리층 블록에서 송신되는 상기 송신되는 데이터의 부분은 P_{-1} 의 최대값 및 $1/N$ 일 수 있는데, 즉, 가장 높은 우선순위의 서브-블록 1 내의 모든 데이터 및 가능하게는 나머지 데이터의 일부가 마지막 물리층 블록 N에서 송신된다. M_{-1} 이 최대값이라고 하고, $L_{-1} = 1 - M_{-1}$ 을 상기 데이터의 부분이 마지막 물리층 블록 N에서 송신된 이후 물리층 블록들 $N-1, \dots, 1$ 에서 송신될 데이터의 나머지 부분이라고 하자. 그러면, 물리층 블록 N-1에서 송신되는 상기 송신되는 데이터의 부분은 $P_{-1} + P_{-2} - M_{-1}$ 및 $1/N-1$ 의 최대값일 수 있는데, 즉, 가장 높은 우선순위의 서브-블록 및 두번째로 높은 우선순위의 서브-블록 모두가 마지막 2개의 물리층 블록들에서 송신되고, 그리고 가능하게는 나머지 데이터 중 일부 역시 송신된다. 이는 제 1의 2개의 우선순위들의 데이터가 2개의 물리층 블록들이 수신된 이후에 수신기에서 재생될 것임을 가정한다.

[0115] 이 방법은 어느 송신되는 데이터가 각각의 물리층 블록에서 송신될지를 결정하도록 확장될 수 있다. 이 방법은 또한 수신기가 송신되는 소스 블록 데이터를 재생하기 위한 수신기 요건들이 상이한 경우, 예를 들어, 우선순위 2의 송신되는 데이터가 2개의 물리층 블록 이후 대신 3개의 물리층 블록들을 수신한 이후 재생될 예정인 경우로 확장될 수도 있다. 상기 방법들은 또한 동일한 물리적 채널 상에서 많은 상이한 스트림들 또는 스트림들의 번들(bundle)들을 멀티플렉싱하기 위한 요건에 의해 수정될 수 있으며, 여기서, 각각의 물리층 블록에서 사용 가능한 공간의 양은 각각의 스트림 또는 번들화된 스트림들에 대한 각각의 우선순위를 가지는 송신되는 데이터 중 얼마만큼이 각각의 블록에서 송신될 것인지를 결정하기 위해 사용된다.

[0116] 전술된 우선순위들이 완전한 순서화를 기술하지는 않는데, 즉, 상기 우선순위들은 부분적인 순서일 수 있고, 어느 경우든, 우선순위화된 데이터를 위치시키는 순서에 대한 선택들이 존재하며, 실제로, 일부 실시예들에 있어서, 우선순위의 측면에서 비교불가한 우선순위화된 데이터가 송신 순서로 함께 믹싱될 수 있다는 점에 유의한다.

[0117] 전술된 바와 같이, 이를 제시된 송신 배열들 중 임의의 것의 달성은 여기서 설명되는 개선된 송신 및 수신 방법들 및 프로세스들, 예를 들어, 각각의 심볼과 함께 송신되는 헤더 데이터가 아닌 각각의 심볼과 함께 송신되는 헤더 데이터를 포함하는, ESI들을 사용하여 달성될 수 있다.

소스 블록의 부분적 FEC 코딩

[0119] FEC 복원 데이터는 전체 소스 블록으로부터 생성될 수 있으며, 소스 블록으로부터의 충분한 소스 심볼들 더하기 소스 블록으로부터 생성된 복원 심볼들이 수신되는 경우 소스 블록의 전체 또는 상당한 부분들을 복원하는 성능을 제공한다. FEC 복원 데이터는 소스 블록의 오직 일부분들로부터만 생성될 수 있는데, 예를 들어, FEC 복원 데이터의 한 세트는 소스 블록의 제 1 부분으로부터 생성될 수 있고, FEC 복원 데이터의 제 2 세트는 소스 블록의 제 2 부분으로부터 생성될 수 있다. 일 예로서, 소스 블록의 제 2 부분은 소스 블록의 제 1 부분 더하기 소스 블록의 일부 추가적인 부분들을 포함할 수 있다. 소스 블록에 대한 소스 심볼들은 낮은 우선순위의 소스 서브-블록 및 높은 우선순위의 소스 서브-블록으로 분할된다고 가정하자. 그러면, FEC 복원 심볼들의 제 1 서브-블록은 상기 높은 우선순위의 소스 서브-블록으로부터 생성될 수 있고, FEC 복원 심볼들의 제 2 서브-블록들은 상기 낮은 우선순위의 소스 서브-블록 및 높은 우선순위의 소스 서브-블록의 연접(concatenation)으로부터 생성될 수 있다. 이후 서브-블록들의 송신 순서는: FEC 복원 심볼들의 제 2 서브-블록, 낮은 우선순위의 소스 서브-블록, FEC 복원 심볼들의 제 1 서브-블록, 높은 우선순위의 소스 서브-블록이 될 수 있다. 이 경우, 수신기가 높은 우선순위의 서브-블록의 일부 또는 전부만을 수신하는 경우, 너무 많은 손상이 존재하지 않는다면, 상기 수신기는 이것을 즉시 재생하려고 시도할 수 있다. 수신기가 FEC 복원 심볼들의 제 1 서브-블록 및 높은 우선순위의 소스 서브-블록의 일부 또는 전부를 수신하는 경우, 상기 수신기는 너무 많은 손상이 존재하지 않는다면, FEC 복원 심볼들의 제 1 서브-블록을 사용하여 상기 높은 우선순위의 소스 서브-블록을 복원하려고 시도할 수 있다. 수신기가 낮은 우선순위의 소스 서브-블록의 일부 또는 전부, FEC 복원 심볼들의 제 1 서브-블록 및 높은 우선순위의 소스 서브-블록을 수신하는 경우, 상기 수신기는 FEC 복원 심볼들의 제 1 서브-블록을 사용하여 높은 우선순위의 소스 서브-블록의 손상된 부분들을 복원시키려 할 수 있고, 이후 상기 낮은 우선순위의 소스 서브-블록의 수신된 부분들 및 상기 높은 우선순위 소스 서브-블록의 복원된 부분들을 미디어 플레이어로 송신하려 할 수 있다. 수신기가 모든 4개의 서브-블록들 중 일부 또는 전부를 수신하는 경우, 상기 수신기는 소스 심볼들 모두를 복원하기 위해 상기 FEC 복원 심볼들 모두를 사용할 수 있다.

[0120] 상기 방법들이, 각각의 서브-블록에 대한 FEC 보호를 개별적으로 제공하는데 바람직할 수 있으며, 예를 들어,

FEC 복원 심볼들의 제 2 서브-블록이 보호되면, 단지 낮은 우선순위의 소스 서브-블록 대신 전체 소스 블록이 바람직할 수 있다는 점에 유의한다. 예를 들어, 2개의 소스 서브-블록들 각각이 100개의 소스 심볼들을 각각 포함하며, 2개의 FEC 복원 서브-블록들 각각이 50개의 복원 심볼들을 각각 포함한다고 가정하자. 전술된 방법들의 사용은, 높은 우선순위의 소스 서브 블록으로부터의 소스 심볼들 중 60개가 유실되고 낮은 우선순위 소스 서브-블록으로부터의 소스 심볼들 중 30개가 유실되는 경우라도, 전체 소스 블록의 복원을 가능하게 하는 반면, 2개의 소스 서브-블록들이 2개의 FEC 복원 서브-블록들에 의해 독립적으로 보호된다면 높은 우선순위의 서브-블록의 복원은 가능하지 않다(서브 블록의 60개의 소스 심볼들이 유실되면 상기 서브-블록을 보호하는 50개의 복원 심볼들만이 존재한다). 이러한 FEC 보호는 예를 들어, 리드-솔로몬 코드들을 사용하여 실현될 수 있으며, 실험들은 오버랩하는 서브-블록들을 보호하기 위해 전술된 방식으로 사용될 때 리드-솔로몬 코드들이 대부분의 이상적인 복원 특징들을 보인다는 점을 보여준다.

[0121]

이들 방법들 역시 너무 긴 시간 기간동안의 보호가 수신되는 전체 시간 기간들을 때때로 파괴되게 하는 경우의 보호를 위해 유용하다. 대신, 짧은 블록들에 대해 FEC 보호를 제공하고 이후 짧은 블록들을 포함하는 더 긴 블록들에 대해 FEC 보호를 제공하는 것 역시 바람직할 수 있다. 이러한 방식으로, 주변 시간 시간들에서 너무 많은 손실을 가지지 않고 중단(outage)되는 경우, 짧은 블록들에 대한 FEC 보호는 이들로 하여금 복원되게 할 수 있는 반면, 더 긴 블록들에 대한 추가적인 FEC 보호는 더 많은 유실이 더 긴 시간 구간 동안 존재하게 한다.

[0122]

다수의 물리층 블록 스트림들의 수신

[0123]

논리적으로 연결되는 스트림들이 물리층 블록들의 단일 스트림을 통해 송신되는 스트리밍 애플리케이션들에 대해, 전체 물리적 채널은 다수의 이러한 물리층 블록 스트림들로 구성될 수 있다. 예를 들어, 각각의 물리층 블록 스트림은 256 Kbps, 또는 1 Mbps일 수 있는 반면, 이 예에서 전체 가용 물리적 채널이 12.5 내지 50 Mbps가 되도록 50개의 이러한 스트림들이 존재할 수 있다.

[0124]

통상적으로, 수신기는, 전력 문제점들 및 메모리 문제점들을 포함하는 다양한 상이한 이유들로 인해, 한번에 물리층 블록들의 스트림들 중 하나를 수신할 수 있다. 그러나, 수신기가 물리층 블록들의 둘 이상의 스트림들을 수신하는 것에 대한 장점들이 존재할 수 있다. 예를 들어, 수신기가 모든 이러한 스트림들을 수신 중인 경우, 한 스트림에서 또 다른 스트림으로의 채널 재핑은 거의 동시에 발생할 수 있으며, 수신기가 이동하는 새로운 스트림은 가장 높은 품질 레벨에서 시작하여 재생될 수 있는데, 왜냐하면, 새로운 스트림들에 대한 모든 데이터가 수신기가 채널들을 상기 스트림으로 변경하기 이전의 시간 기간동안 도달하기 때문이다. 이는 스트림들이 긴 보호 기간을 가지는 FEC 보호를 사용하여 보호되는 경우라 할지라도, 또한 상기 스트림들이 매우 압축되는 이러한 방식으로 비디오 인코딩되는 경우, 예를 들어, 때때로, I-프레임들이라 지칭되고, 때때로 IDR-프레임들(Independent Data Refresh 프레임들)이라 지칭되는 비디오 스트림 내의 리프레시 프레임들이 그들의 큰 사이즈로 인해 빈번하지 않게 송신되는 경우라 할지라도 참이다. 이는 통상적으로 GOP(Group of Pictures: 화상 그룹)이 걸쳐 있는 시간이 매우 압축된 비디오 스트림에서보다 다소 클 수 있다는 점을 의미한다. 예를 들어, 비디오 스트림에 대한 GOP 뉴레이션이 10초일 수 있고, FEC 보호가 전체 10초의 GOP를 보호하기 위해 제공될 수 있다. 이 경우, 전술된 방법들 중 일부를 사용하지 않고, 스트림으로부터의 높은 우선순위의 데이터가 가능한 빨리 디스플레이되고, 이후 더 낮은 우선순위의 데이터 역시 스트림 재생이 진행됨에 따라 재생 품질을 향상시키도록 디스플레이되며, 수신기가 한번에 오직 하나의 채널을 수신하고 있었다면 채널 재핑 시간은 10초가 될 수 있는 반면, 수신기가 모든 채널들을 수신중인 경우, 채널 재핑 시간은 거의 동시적일 수 있다.

[0125]

수신기가 물리층 패킷들의 둘 이상의 스트림을 동시에 수신하는 솔루션을 고려할 때 가능한 일부 최적화들이 존재한다. 예를 들어, 수신기는, 예를 들어, 재생을 위해 현재 미디어 플레이어로 송신중인 스트림들만을 FEC 디코딩, 예를 들어, 에러-정정 디코딩 또는 삭제 보호 디코딩하는 것만을 필요로 한다. 다른 스트림들에 대한 데이터는 저장될 수 있고, 수신기가 채널들을 변경하는 경우 단지 FEC 디코딩될 수 있으며, 그후 FEC 디코딩은 거의 즉시 미디어 재생을 시작하기 위해 새로운 채널에 대해 이미 수신된 데이터에 대해 매우 빨리 발생할 수 있다.

[0126]

또다른 가능한 최적화로서, 수신기가 한번에 오직 하나의 스트림을 수신하는 경우, 수신기가 먼저 상기 스트림을 조인시킬 때 재생하기 위해 사용가능한 스트림의 이전 부분들을 가지는 경우 필요하지 않은 스트림에 포함되는 리던던트 데이터가 존재할 수 있다. 이러한 리던던트 데이터의 예들은 비디오 스트림에서 단독으로 매우 자주 포함되는 낮은 품질의 비디오 IDR 프레임들일 수 있으므로, 수신기는, 스트림이 저하된 품질이라 할지라도, 상기 스트림을 수신하여 일부 비디오를 거의 즉시 재생하기 시작할 수 있다. 수신기가 높은 품질의 IDR 프레임 및 이전에 송신된 모든 후속 프레임들을 포함하는, 스트림의 이전 부분들을 가지는 경우, 빈번한 낮은 품질의

IDR 프레임들을 포함할 필요성이 없게 될 것이다. 상기 낮은 품질의 IDR 프레임들은 사용가능한 대역폭 중 충분한 양을 사용할 수 있는데, 예를 들어, 각각의 낮은 품질의 IDR 프레임이 3 KB이고 이들이 매초 256 Kbps 스트림으로 송신되는 경우, 상기 낮은 품질의 IDR 프레임들은 상기 사용가능한 대역폭의 9% 이상을 사용한다. 상기 낮은 품질의 IDR 프레임들의 송신은 수신기가 변경시키려는 스트림에 대한 데이터를 채널이 상기 스트림으로 변경하기 이전에 수신하고 있는 경우 필수적이지 않다.

[0127] 물리층 블록들의 다수의 스트림들을 청취하는 것의 한가지 단점은 그것이 단일 스트림을 청취하는 것보다 수신기에서 더 많은 전력을 사용한다는 것이다. 추가적으로, 단일 스트림보다 다수의 스트림들로부터 수신되는 데이터를 저장하기 위해 더 많은 메모리 및 다른 자원들이 필요하다. 이를 단점들을 최소화시키기 위해 사용될 수 있는 일부 방법들이 존재한다. 한가지 이러한 방법은 상기 이점들을 달성하기 위해 수신기가 한번에 오직 소수의 스트림들만을 수신할 필요가 있는 방식으로 사용가능한 스트림들에 대해 전체적으로(globally) 로직 및/또는 데이터를 정렬하는 것이다.

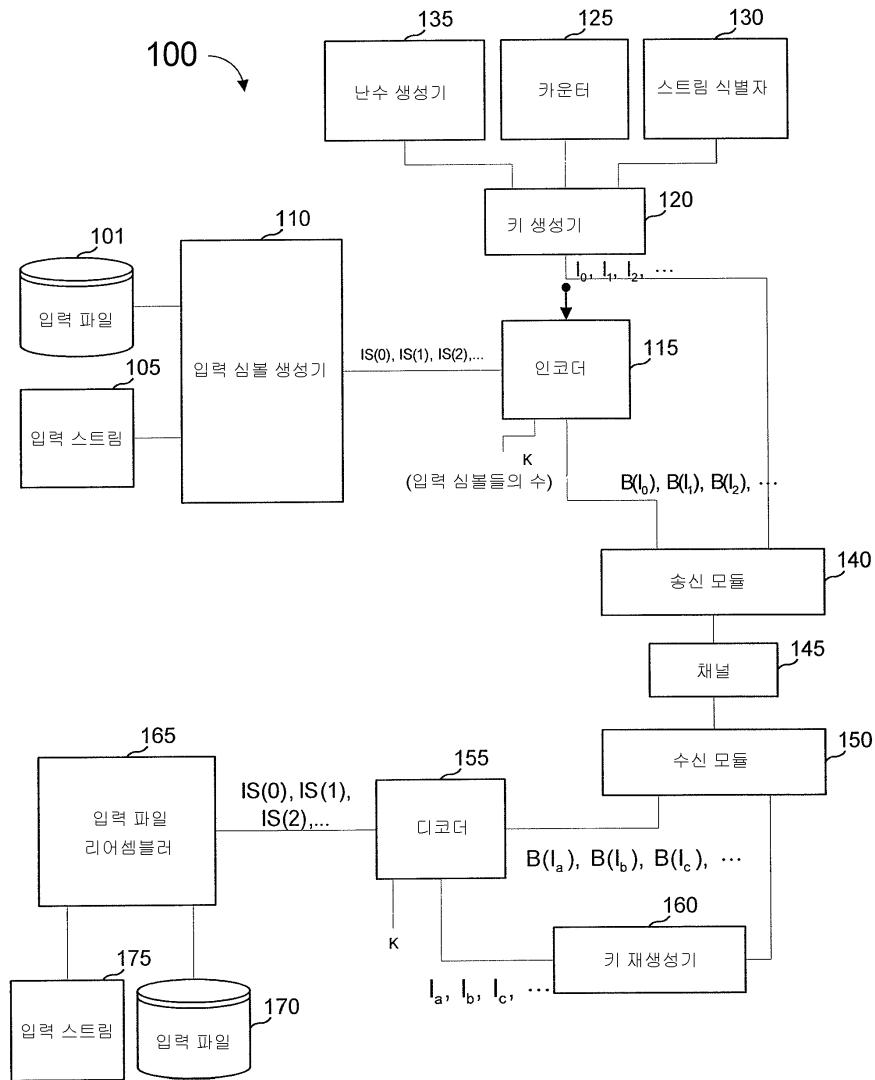
[0128] 예를 들어, 수신기가 채널들을 어느 스트림들로 변경시킬 가능성이 있는지를 예측할 수 있는 로직이 존재하는 경우, 상기 로직은 수신기가 채널로의 실제 변경 이전에 이들 가능성 있는 채널들을 수신하게 할 수 있다.

[0129] 또 다른 예로서, 물리층 블록 스트림들 내의 데이터는 IDR 스트림이라 지칭되는 모든 다른 비디오 스트림들에 대한 IDR 프레임들 모두를 전달하는 하나의 물리층 블록 스트림들이 존재하도록, 그후 각각의 다른 물리층 블록 스트림이 해당 비디오 스트림에 대한 IDR 프레임들을 제외하고 비디오 스트림들 중 하나에 대한 데이터 모두를 전달하도록 정렬될 수 있다. 이 예에서, 수신기는 IDR 스트림을 수신하는 동시에(항상, 또는 적절한 경우 간헐적으로) 미디어 플레이어에 의해 현재 재생중인 비디오 스트림에 대한 현재 물리층 블록 스트림을 수신할 수 있다. 따라서, 수신기는, 셀-네일 채널 가이드 모드에서 사용가능한 비디오 스트림들의 일부 또는 전부에 관한 정보를 디스플레이할 때 재생하거나, 또는 채널 변경이 수신기에서 이루어질 때 새로운 비디오 스트림의 디스플레이를 시작하기 위해 상기 수신기가 사용할 수 있는, 비디오 스트림들의 일부 또는 전부에 대한 IDR 프레임들을 사용가능하게 할 수 있다. IDR 스트림은 항상 수신될 수 있거나, 또는 간헐적으로 수신될 수 있는데, 예를 들어, 현재 재생중인 비디오 스트림에 대한 IDR-프레임들을 포함하는 IDR 스트림으로부터의 물리층 블록들만이 수신될 수 있다. 모든 경우들에서, FEC 보호는 원하는 경우 각각의 물리층 블록 스트림 상에 제공될 수 있다. 이들 방법들의 한가지 장점은 수신기가 임의의 시점에서 기껏해야 2개의 물리층 블록 스트림들만을 수신하면서, 현재 모든 물리층 블록 채널들의 수신의 장점을 모두 또는 대부분을 얻는다는 것이다.

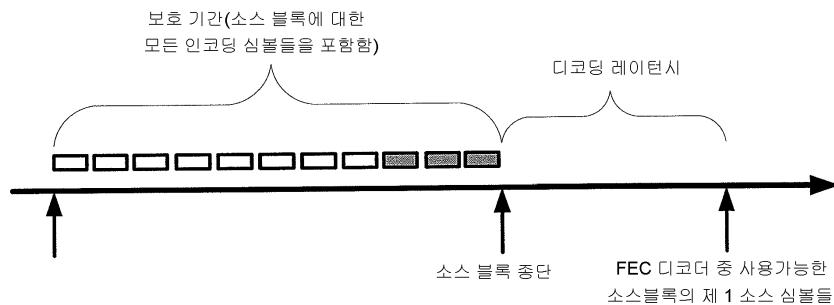
[0130] 본 발명이 예시적인 실시예들에 대해 설명되었지만, 당업자라면 다수의 수정들이 가능하다는 점을 인지할 것이다. 예를 들어, 여기서 설명되는 프로세스들은 하드웨어 컴포넌트들, 소프트웨어 컴포넌트들, 및/또는 이들의 임의의 조합을 사용하여 구현될 수 있다. 예를 들어, 여기서 설명되는 방법들은 상기 방법을 실행하도록 컴퓨터의 프로세서에 지시할 수 있는 컴퓨터-실행가능한 코드를 포함하는 CD-ROM, DVD 등과 같은 컴퓨터-관련 가능한 매체 상에 내장될 수 있다. 따라서, 본 발명이 예시적인 실시예들에 대해 설명되었지만, 본 발명이 후속하는 청구항들의 범위 내에서 모든 수정들 및 등가물들을 커버하도록 의도된다는 점이 이해될 것이다.

도면

도면1



도면2



도면3

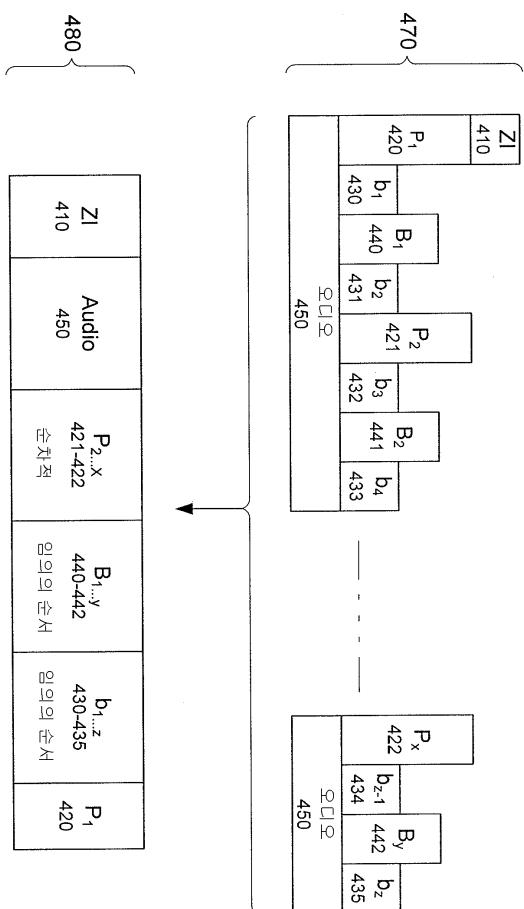
제 1 세그먼트 테이블

세그먼트 식별자	심볼들의 수	초기 ESI
0	450	0
1	300	420
2	360	630
3	426	930

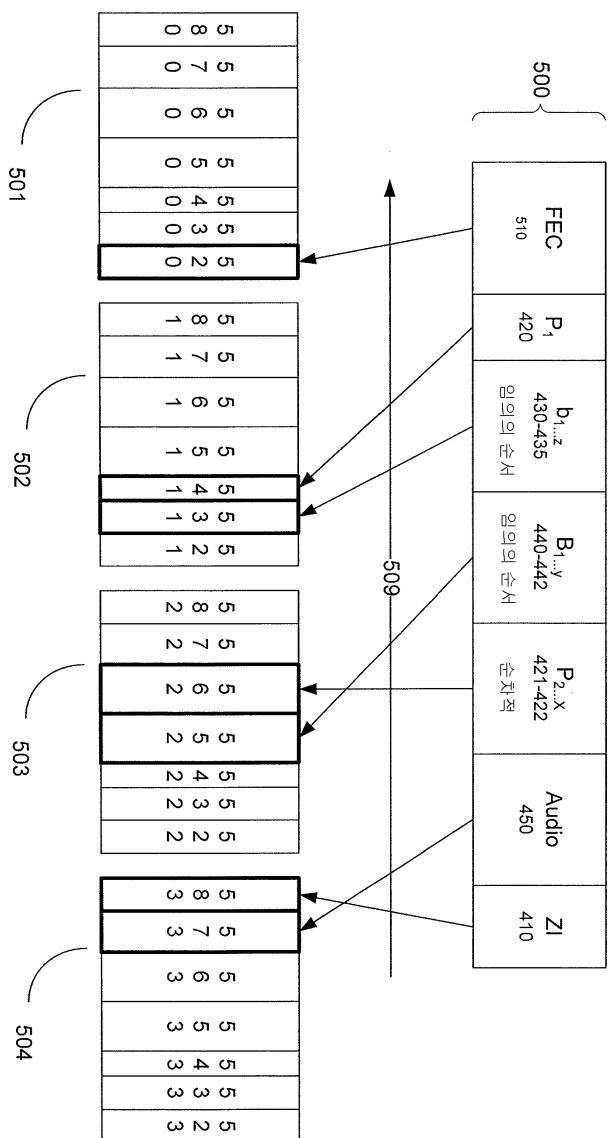
제 2 세그먼트 테이블

세그먼트 식별자	심볼들의 수	초기 ESI
0	420	0
1	360	270
2	330	600
3	426	960

도면4



도면5



도면6

