

(19) 日本国特許庁(JP)

## (12) 特 許 公 報(B2)

(11) 特許番号

特許第6105717号  
(P6105717)

(45) 発行日 平成29年3月29日(2017.3.29)

(24) 登録日 平成29年3月10日(2017.3.10)

(51) Int.Cl.

H04N 21/845 (2011.01)  
H04N 21/2187 (2011.01)

F 1

H04N 21/845  
H04N 21/2187

請求項の数 14 (全 100 頁)

(21) 出願番号 特願2015-509146 (P2015-509146)  
 (86) (22) 出願日 平成25年4月25日 (2013.4.25)  
 (65) 公表番号 特表2015-519813 (P2015-519813A)  
 (43) 公表日 平成27年7月9日 (2015.7.9)  
 (86) 國際出願番号 PCT/US2013/038247  
 (87) 國際公開番号 WO2013/163448  
 (87) 國際公開日 平成25年10月31日 (2013.10.31)  
 審査請求日 平成28年4月11日 (2016.4.11)  
 (31) 優先権主張番号 13/456,474  
 (32) 優先日 平成24年4月26日 (2012.4.26)  
 (33) 優先権主張国 米国(US)

早期審査対象出願

(73) 特許権者 507364838  
 クアルコム、インコーポレイテッド  
 アメリカ合衆国 カリフォルニア 921  
 21 サン デイエゴ モアハウス ドラ  
 イブ 5775  
 (74) 代理人 100108453  
 弁理士 村山 靖彦  
 (74) 代理人 100163522  
 弁理士 黒田 晋平  
 (72) 発明者 マイケル・ジー・ルビー  
 アメリカ合衆国・カリフォルニア・921  
 21-1714・サン・ディエゴ・モアハ  
 ウス・ドライブ・5775

最終頁に続く

(54) 【発明の名称】低レイテンシストリーミングを処理するための改善されたブロック要求ストリーミングシステム

## (57) 【特許請求の範囲】

## 【請求項 1】

メディアサーバを用いて、提供されるべきコンテンツのデータを構造化するための方法であって、

提供されるべき前記コンテンツを取得するステップと、

前記コンテンツを表し、各メディアセグメントへと符号化されたメディアプレゼンテーションの1つまたは複数のフレームを含む符号化プロトコルに従って符号化される、複数のメディアセグメントを生成するステップであって、ランダムアクセスポイントが各メディアセグメント中で利用可能である、生成するステップと、

前記符号化プロトコルに従って符号化された複数のメディアフラグメントを生成するステップであって、前記メディアセグメントは前記複数のメディアフラグメントを含み、前記複数のメディアフラグメントの少なくともいくつかがランダムアクセスポイントを含み、少なくともいくつかがランダムアクセスポイントを含まず、ランダムアクセスポイントが、前記ランダムアクセスポイントの前のフラグメントと独立に前記ランダムアクセスポイントに続くフラグメントをデコードが復号できるセグメント内の位置を含む、ステップと、

前記メディアフラグメントに対するセグメントインデックスを生成するステップと、を含み、

前記セグメントインデックスが、前記メディアセグメント内のメディアフラグメントごとのプレゼンテーション時間範囲と、各メディアフラグメントが占める前記メディアセグ

10

20

メント中の対応するバイト範囲と、ランダムアクセスポイントが各メディアフラグメント内に存在するかどうかを示す、ランダムアクセスポイント存在インジケータを含み、

前記メディアセグメントは、低レイテンシストリーミングのための前記複数のメディアフラグメントを連結することによって生成されたライブプロファイル適合メディアセグメントであり、

前記方法はさらに、前記複数のライブプロファイル適合メディアセグメントを有する前記メディア表現の第1の表現と、低レイテンシストリームの前記複数のメディアフラグメントを有する前記メディア表現の第2の表現とに関する情報を記憶する単一のmedia presentation description(MPD)ファイルを生成するステップを含む、

方法。

10

#### 【請求項2】

前記メディアセグメントをキャッシュの中で生成するステップをさらに含み、前記メディアセグメントが前記キャッシュの中で生成された後で、前記メディアセグメントを生成するために使用された前記複数のメディアフラグメントが前記キャッシュからページされる、請求項1に記載の方法。

#### 【請求項3】

前記MPDファイルが、前記第2の表現内でのランダムアクセスポイントの発生の頻度を示すための属性を含む、請求項1に記載の方法。

#### 【請求項4】

前記頻度が期間である、請求項3に記載の方法。

20

#### 【請求項5】

前記頻度がメディアフラグメントの数である、請求項3に記載の方法。

#### 【請求項6】

前記複数のメディアフラグメントのうち前記ランダムアクセスポイントに対する位置を決定するステップであって、前記ランダムアクセスポイントは前記複数のメディアフラグメントのうち可変の非固定位置に配置されるステップをさらに含む、請求項1に記載の方法。

#### 【請求項7】

メディアサーバであって、

提供されるべきコンテンツを取得し、

30

前記コンテンツを表し、各メディアセグメントへと符号化されたメディアプレゼンテーションの1つまたは複数のフレームを含む符号化プロトコルに従って符号化される、複数のメディアセグメントを生成し、

前記符号化プロトコルに従って符号化された複数のメディアフラグメントを生成し、

前記メディアフラグメントに対するセグメントインデックスを生成する、

ように構成されたプロセッサ

を備え、

ランダムアクセスポイントが各メディアセグメント中で利用可能であり、

前記メディアセグメントは前記複数のメディアフラグメントを含み、前記複数のメディアフラグメントの少なくともいくつかがランダムアクセスポイントを含み、少なくともいくつかがランダムアクセスポイントを含まず、ランダムアクセスポイントが、前記ランダムアクセスポイントの前のフラグメントと独立に前記ランダムアクセスポイントに続くフラグメントをデコーダが復号できるセグメント内の位置を含み、

40

前記セグメントインデックスが、前記メディアセグメント内のメディアフラグメントごとのプレゼンテーション時間範囲と、各メディアフラグメントが占める前記メディアセグメント中の対応するバイト範囲と、ランダムアクセスポイントが各メディアフラグメント内に存在するかどうかを示す、ランダムアクセスポイント存在インジケータを含み、

前記メディアセグメントは、低レイテンシストリーミングのための前記複数のメディアフラグメントを連結することによって生成されたライブプロファイル適合メディアセグメントであり、

50

前記プロセッサはさらに、前記複数のライブプロファイル適合メディアセグメントを有する前記メディア表現の第1の表現と、低レイテンシストリームの前記複数のメディアフラグメントを有する前記メディア表現の第2の表現とに関する情報を記憶する单一のmedia presentation description(MPD)ファイルを生成するように構成される、  
メディアサーバ。

【請求項 8】

前記プロセッサは、前記メディアセグメントをキャッシュの中で生成するようにさらに構成され、前記メディアセグメントが前記キャッシュの中で生成された後で、前記メディアセグメントを生成するために使用された前記複数のメディアフラグメントが前記キャッシュからバージされる、請求項7に記載のメディアサーバ。 10

【請求項 9】

前記MPDファイルが、前記第2の表現内でのランダムアクセスポイントの発生の頻度を示すための属性を含む、請求項7に記載のメディアサーバ。

【請求項 10】

前記頻度が期間である、請求項9に記載のメディアサーバ。

【請求項 11】

前記頻度がメディアフラグメントの数である、請求項10に記載のメディアサーバ。

【請求項 12】

前記プロセッサは前記複数のメディアフラグメントのうち前記ランダムアクセスポイントに対する位置を決定するようにさらに構成され、前記ランダムアクセスポイントは前記複数のメディアフラグメントのうち可変の非固定位置に配置されるステップをさらに含む、請求項7に記載のメディアサーバ。 20

【請求項 13】

実行されたとき、1つまたは複数のコンピューティングデバイスに、

提供されるべきコンテンツを取得させ、

前記コンテンツを表し、各メディアセグメントへと符号化されたメディアプレゼンテーションの1つまたは複数のフレームを含む符号化プロトコルに従って符号化される、複数のメディアセグメントを生成させ、

前記符号化プロトコルに従って符号化された複数のメディアフラグメントを生成させ、

前記メディアフラグメントに対するセグメントインデックスを生成させ、 30

るコンピュータ実行可能命令を格納した1つまたは複数の非一時的コンピュータ可読記憶媒体であって、

ランダムアクセスポイントが各メディアセグメント中で利用可能であり、

前記メディアセグメントは前記複数のメディアフラグメントを含み、前記複数のメディアフラグメントの少なくともいくつかがランダムアクセスポイントを含み、少なくともいくつかがランダムアクセスポイントを含まず、ランダムアクセスポイントが、前記ランダムアクセスポイントの前のフラグメントと独立に前記ランダムアクセスポイントに続くフラグメントをデコーダが復号できるセグメント内の位置を含み、

前記セグメントインデックスが、前記メディアセグメント内のメディアフラグメントごとのプレゼンテーション時間範囲と、各メディアフラグメントが占める前記メディアセグメント中の対応するバイト範囲と、ランダムアクセスポイントが各メディアフラグメント内に存在するかどうかを示す、ランダムアクセスポイント存在インジケータを含み、 40

前記メディアセグメントは、低レイテンシストリーミングのための前記複数のメディアフラグメントを連結することによって生成されたライブプロファイル適合メディアセグメントであり、

前記命令はさらに、前記デバイスに、前記複数のライブプロファイル適合メディアセグメントを有する前記メディア表現の第1の表現と、低レイテンシストリームの前記複数のメディアフラグメントを有する前記メディア表現の第2の表現とに関する情報を記憶する单一のmedia presentation description(MPD)ファイルを生成させる、

非一時的コンピュータ可読記憶媒体。 50

**【請求項 14】**

前記コンピュータ実行可能命令は、実行されたとき、前記1つまたは複数のコンピューティングデバイスに、前記複数のメディアフラグメントのうち前記ランダムアクセスポイントに対する位置を決定させ、前記ランダムアクセスポイントは前記複数のメディアフラグメントのうち可変の非固定位置に配置されるステップをさらに含む、請求項13に記載の非一時的コンピュータ可読記憶媒体。

**【発明の詳細な説明】****【技術分野】****【0001】**

本発明は、改善されたメディアストリーミングシステムおよび方法に関し、より具体的には、ストリーミングされたメディアのプレゼンテーションを最適化するためにネットワークおよびバッファの状態に適応し、ストリーミングされたメディアデータの効率的かつ同時の配信、またはタイムリーに分配される配信を可能にする、システムおよび方法に関する。 10

**【背景技術】****【0002】**

高品質のオーディオおよびビデオが、インターネット、セルラーおよびワイヤレスネットワーク、電力線ネットワーク、ならびに他のタイプのネットワークのようなパケットベースのネットワークを通じて配信されることがより一般的になるにつれて、ストリーミングメディア配信はますます重要になり得る。配信されるストリーミングメディアが提示され得る際の品質は、元のコンテンツの解像度(または他の属性)、元のコンテンツの符号化品質、メディアを復号し提示するための受信側デバイスの能力、受信機において受信される信号の適時性および品質などを含む、多数の要因に依存し得る。知覚される良好なストリーミングメディア体験を作り出すには、受信機において受信される信号のトランスポートおよび適時性が特に重要であり得る。良好なトランスポートは、送信者が送信するものに対する、受信機において受信されるストリームの忠実性を提供することができ、一方適時性は、コンテンツに対する最初の要求の後で受信機がどれだけ迅速にそのコンテンツの再生を開始できるかを表し得る。 20

**【0003】**

メディア配信システムは、メディアソース、メディア送信先、およびソースと送信先とを分離するチャネル(時間および/または空間における)を有するシステムとして特徴付けられ得る。通常、ソースは、電気的に管理可能な形式のメディアへのアクセス権がある送信機と、メディア(またはその類似物)の受信を電気的に制御しメディアをメディア利用者(たとえば、受信機、記憶デバイスまたは要素、別のチャネルなどと何らかの方式で結合された表示デバイスを有するユーザ)に提供する能力のある受信機とを含む。 30

**【0004】**

多くの変形が可能であるが、一般的な例では、メディア配信システムは、電気的な形式のメディアコンテンツへのアクセス権を有する1つまたは複数のサーバを有し、1つまたは複数のクライアントシステムまたはデバイスは、メディアに対する要求をサーバへ行い、サーバは、サーバの一部として送信機を使用してメディアを搬送し、受信されたメディアが何らかの方式でクライアントによって利用され得るように、クライアントにおける受信機に送信する。単純な例では、所与の要求および応答に対して1つのサーバおよび1つのクライアントがあるが、そうである必要はない。 40

**【0005】**

従来、メディア配信システムは、「ダウンロード」モデルまたは「ストリーミング」モデルのいずれかとして特徴付けられ得る。「ダウンロード」モデルは、メディアデータの配信と、ユーザまたは受信者デバイスに対するメディアの再生との間の、タイミングの独立性により特徴付けられ得る。

**【0006】**

ある例として、メディアは、メディアが必要とされるときまたは使用されるであろうと 50

きよりも十分に前にダウンロードされ、メディアが使用されるときには、必要とされるだけの量がすでに受信者において利用可能である。ダウンロードの文脈における配信は、HTTP、FTP、またはFile Delivery over Unidirectional Transport(FLUTE)のようなファイルトランSPORTプロトコルを使用して実行されることが多く、配信レートは、TCP/IPのような、背後にあるフローおよび/または混雑制御プロトコルにより決まり得る。フローまたは混雑制御プロトコルの動作は、ユーザまたは宛先デバイスに対するメディアの再生とは独立であってよく、メディアの再生はダウンロードと同時に、またはそれ以外の時に行われ得る。

#### 【0007】

「ストリーミング」モードは、メディアデータの配信のタイミングと、ユーザまたは受信者デバイスに対するメディアの再生のタイミングとの、密接な結合により特徴付けられ得る。この文脈における配信は、制御のためのReal Time Streaming Protocol(RTSP)およびメディアデータのためのReal Time Transport Protocol(RTP)のような、ストリーミングプロトコルを使用して実行されることが多い。配信レートは、ストリーミングサーバによって決定されることがあり、データの再生レートと一致することが多い。

#### 【先行技術文献】

#### 【非特許文献】

#### 【0008】

【非特許文献1】L. Rizzo、「Effective Erasure Codes for Reliable Computer Communication Protocols」、Computer Communication Review、27(2):24-36 (1997年4月)

【非特許文献2】Bloemer他、「An XOR-Based Erasure-Resilient Coding Scheme」、Technical Report TR-95-48、International Computer Science Institute、カリフォルニア州バークレー(1995)

#### 【発明の概要】

#### 【発明が解決しようとする課題】

#### 【0009】

「ダウンロード」モデルのいくつかの欠点は、配信と再生とのタイミングの独立性により、メディアデータのいずれもが、再生のために必要とされるときに利用可能ではないことがあります(たとえば、メディアのデータレートよりも利用可能な帯域幅が少ないことが原因で)、再生の一時的な停止を引き起こし('ストール')、悪いユーザ体験をもたらすこと、またはメディアデータが再生のはるかに前にダウンロードされることが必要とされることがあります(たとえば、メディアのデータレートよりも利用可能な帯域幅が大きいことが原因で)、乏しいことがある受信側デバイスの記憶リソースを消費すること、および、コンテンツが最終的に再生されなければ、または別様に使用されなければ無駄になり得る配信のために貴重なネットワークリソースを消費することであり得る。

#### 【0010】

「ダウンロード」モデルの利点には、そのようなダウンロードを実行するために必要とされる技術、たとえばHTTPが非常に成熟しており、広範な用途にわたって広く展開されており適用可能であるということであり得る。ダウンロードサーバ、およびそのようなファイルダウンロードの大きな拡張性に対する解決法(たとえば、HTTPウェブサーバおよびコンテンツ配信ネットワーク)は容易に利用可能であることがあります、この技術に基づくサービスの展開を単純および低コストにする。

#### 【0011】

「ストリーミング」モデルのいくつかの欠点は、メディアデータの配信のレートが一般に、サーバからクライアントへの接続において利用可能な帯域幅に適合しないこと、および、帯域幅と遅延保証を提供する特別なストリーミングサーバまたはより複雑なネットワークアーキテクチャが必要とされることであり得る。利用可能な帯域幅に応じた配信データレートの変動に対応するストリーミングシステムが存在するが(たとえば、Adobe Flash Adaptive Streaming)、これらは一般に、すべての利用可能な帯域幅の利用において

10

20

30

40

50

、TCPのようなダウンロードトランスポートフロー制御プロトコルほど効率的ではない。

#### 【0012】

最近、「ストリーミング」モデルと「ダウンロード」モデルの組合せに基づく新たなメディア配信システムが、開発され展開されている。そのようなモデルの例は、本明細書では「ブロック要求ストリーミング」モデルと呼ばれ、メディアクライアントは、HTTPのようなダウンロードプロトコルを使用して、サービングインフラストラクチャからメディアデータのブロックを要求する。そのようなシステムにおける課題は、ストリームの再生を開始する能力、たとえば、パーソナルコンピュータを使用して、受信されたオーディオおよびビデオストリームを復号しレンダリングし、コンピュータスクリーン上にビデオを表示しつつ内蔵スピーカーを通じてオーディオを再生すること、または別の例として、セットトップボックスを使用して、受信されたオーディオおよびビデオストリームを復号しレンダリングし、テレビジョンディスプレイベイス上にビデオを表示しつつステレオシステムを通じてオーディオを再生することであり得る。10

#### 【0013】

他の課題、たとえば、復号のレイテンシを最小限にし、利用可能なCPUリソースの使用を減らすために、ソースのストリーミングレートについていくのに十分高速にソースブロックを復号できるようにすることも問題である。別の課題は、システムのコンポーネントが故障しても受信機に配信されるストリームの品質に悪影響を与えることのない、ロバストで拡張性のあるストリーミング配信の方法を提供することである。プレゼンテーションが分配されるにつれて、プレゼンテーションについての高速に変換する情報に起因して、他の問題が発生し得る。したがって、改善された処理および装置を有することが望ましい20。

#### 【課題を解決するための手段】

#### 【0014】

ブロック要求ストリーミングシステムは、そのようなシステムのユーザ体験および帯域幅の効率性の改善をもたらし、通常は、従来のファイルサーバによって提供される形式(HTTP、FTPなど)のデータを生成する取込システムを使用し、取込システムは、コンテンツを取り込んで、キャッシュを含むこともまたは含まないこともある、ファイルサーバによって提供されるべきファイルまたはデータ要素としてコンテンツを準備する。30

#### 【0015】

ある実施形態によれば、ブロック要求ストリーミングシステムのメディアサーバは、メディアプレゼンテーションコンテンツの低レイテンシストリーミングを可能にする。ライブプロファイルストリーミングのための比較的大きなセグメントファイルは、低レイテンシストリーミングのために、比較的小さなメディアフラグメントから統合され得る。メディアセグメントおよびメディアフラグメントは、同じ符号化プロトコルに従って符号化される。

#### 【0016】

以下の発明を実施するための形態は、添付の図面とともに、本発明の性質および利点のさらなる理解をもたらす。

#### 【図面の簡単な説明】40

#### 【0017】

【図1】本発明の実施形態によるブロック要求ストリーミングシステムの要素を示す図である。

【図2】ブロックサービングインフラストラクチャ(「BSI」)に結合されて、コンテンツ取込システムによって処理されるデータを受信する、クライアントシステムの要素をさらに詳細に示す、図1のブロック要求ストリーミングシステムを示す図である。

#### 【図3】取込システムのハードウェア/ソフトウェアの実装形態を示す図である。

#### 【図4】クライアントシステムのハードウェア/ソフトウェアの実装形態を示す図である50。

#### 【図5】セグメントおよびmedia presentation description(「MPD」)ファイル、なら

びに、MPDファイル内のセグメントの内訳、タイミング、および他の構造を含む、図1に示されるコンテンツ記憶装置のあり得る構造を示す図である。

【図6】図1および図5に示されるコンテンツ記憶装置に記憶され得る、通常のソースセグメントの詳細を示す図である。

【図7(a)】ファイル内の単純なおよび階層的インデクシングを示す図である。

【図7(b)】ファイル内の単純なおよび階層的インデクシングを示す図である。

【図8(a)】メディアストリームの複数のバージョンにわたって揃えられた探索点を伴う可変のロックサイジングを示す図である。

【図8(b)】メディアストリームの複数のバージョンにわたって揃えられていない探索点を伴う可変のロックサイジングを示す図である。 10

【図9(a)】メタデータテーブルを示す図である。

【図9(b)】サーバからクライアントへのロックおよびメタデータテーブルの送信を示す図である。

【図10】RAPの境界とは独立のロックを示す図である。

【図11】複数のセグメントにまたがる連続的なタイミングおよび不連続的なタイミングを示す図である。

【図12】スケーラブルなロックのある態様を示す図である。

【図13】ロック要求ストリーミングシステム内のいくつかの変数の経時的な展開のグラフィカルな表現を示す図である。

【図14】ロック要求ストリーミングシステム内のいくつかの変数の経時的な展開の別のグラフィカルな表現を示す図である。 20

【図15】閾値の関数として状態のセルグリッドを示す図である。

【図16】要求ごとに单一のロックおよび複数のロックを要求できる、受信機において実行され得るプロセスのフローチャートである。

【図17】柔軟性のあるパイプラインプロセスのフローチャートである。

【図18】ある時間における、要求の候補セット、要求の優先順位、およびどの接続で要求が出され得るかの例を示す図である。

【図19】ある時間から別の時間までの間に変化した、要求の候補セット、要求の優先順位、およびどの接続で要求が出され得るかの例を示す図である。 30

【図20】ファイル識別子に基づく一貫性のあるキャッシングプロキシサーバの選択のフローチャートである。

【図21】適切な式言語に対するシンタックス定義を示す図である。

【図22】適切なハッシュ関数の例を示す図である。

【図23】ファイル識別子構築ルールの例を示す図である。

【図24】TCP接続の帯域幅の変動を示す図である。

【図25】ソースデータおよび修復データに対する複数のHTTP要求を示す図である。

【図26】FECを伴う、およびFECを伴わない、例示的なチャネルザッピング時間を示す図である。

【図27】図1に示される取込システムの一部として、ソースセグメントおよび制御パラメータから修復セグメントを生成する、修復セグメント生成器の詳細を示す図である。 40

【図28】ソースロックと修復ロックとの関係を示す図である。

【図29】クライアントにおける異なる時間でのライブサービスのための手順を示す図である。

【図30】低レイテンシストリーミングのためのメディアフラグメントとメディアフラグメントとの関係を示す図である。

【発明を実施するための形態】

【0018】

図面において、同様の項目は同様の数字により参照され、同様のまたは同一の項目の複数のインスタンスを示すために、サブインデックスが括弧の中で与えられる。別段示されない限り、最後のサブインデックス(たとえば、「N」または「M」)は、いずれの特定の値 50

にも限定することは意図されず、1つの項目のインスタンスの数は、同じ数が示されサブインデックスが再使用される場合であっても、別の項目のインスタンスの数とは異なり得る。

#### 【0019】

本明細書で説明されるように、ストリーミングシステムの目的は、メディアをその記憶位置(またはメディアが生成されている位置)から、メディアが利用される位置、すなわち、メディアがユーザに提示され、または人または電子的な利用者によって別様に「使い果たされる」位置へと移すことである。理想的には、ストリーミングシステムは、受信側において中断されない再生(またはより一般的には、中断されない「消費」)を実現することができ、ユーザが1つまたは複数のストリームを要求してからすぐに、1つのストリームまたはストリームの集合体の再生を開始することができる。効率性の理由で、ユーザがあるストリームから別のストリームに切り替えるとき、または、ユーザがストリーム、たとえば「サブタイトル」ストリームのプレゼンテーションに従うときのように、ストリームがもはや必要とされることをユーザが示すと、各ストリームが停止されることも望ましい。ビデオなどのメディアコンポーネントの提示が続けられるが、このメディアコンポーネントを提示するために異なるストリームが選択される場合、限られた帯域幅を新たなストリームで専有し、古いストリームを停止することが好ましいことが多い。10

#### 【0020】

本明細書で説明される実施形態によるブロック要求ストリーミングシステムは、多くの利点をもたらす。一部の適用形態は、本明細書で説明される特徴のすべてよりも少ないものによって、適切に満足のいく体験を提供し得るので、実行可能なシステムが本明細書で説明される特徴のすべてを含む必要はないことを理解されたい。20

#### 【0021】

##### HTTPストリーミング

HTTPストリーミングは、特定のタイプのストリーミングである。HTTPストリーミングでは、ソースは、標準的なウェブサーバおよびコンテンツ配信ネットワーク(CDN)であってよく、標準的なHTTPを使用してよい。本技法は、ストリーミングのセグメント化および複数のストリームの使用を伴ってよく、すべてが標準化されたHTTP要求という文脈の範囲内にある。ビデオなどのメディアは、複数のビットレートで符号化されて、異なるバージョンまたは表現を形成し得る。「バージョン」および「表現」という用語は、本文書では同義的に使用される。各バージョンまたは表現は、場合によっては各々が数秒というオーダーの、より小さな断片へと分解され、セグメントを形成し得る。各セグメントは次いで、別個のファイルとしてウェブサーバまたはCDNに記憶され得る。30

#### 【0022】

クライアント側では、クライアントによってシームレスに一緒につなぎ合わされた個々のセグメントに対して、HTTPを使用して要求が行われ得る。クライアントは、利用可能な帯域幅に基づいて、異なるデータレートへと切り替えることができる。クライアントはまた、各々が異なるメディアコンポーネントを提示する、複数の表現を要求することができ、これらの表現の中のメディアを、一緒におよび同時に提示することができる。切り替えを引き起こすものには、たとえば、バッファ占有率およびネットワーク測定結果があり得る。安定状態で動作している場合、クライアントは、サーバに対する要求を調整して、目標のバッファ占有率を維持することができる。40

#### 【0023】

HTTPストリーミングの利点には、ビットレートの適応、高速な始動および探索、ならびに最小限の不必要的配信があり得る。これらの利点は、配信を再生の時間のすぐ前となるように制御すること、利用可能な帯域幅を最大限に利用すること(可変ビットレートのメディアを通じて)、ならびに、ストリーミングのセグメント化およびインテリジェントなクライアント側手順を最適化することによるものである。

#### 【0024】

クライアントがファイルの集合体(たとえば、本明細書では3gpセグメントと呼ばれる、50

3GPPによって規定されるフォーマットの)を使用してストリーミングサービスをユーザに提供できるように、media presentation descriptionが、HTTPストリーミングクライアントに提供され得る。media presentation description、および場合によってはこのmedia presentation descriptionの更新は、含まれるメディアを同期した方式でクライアントが提示でき、探索、ピットレートの切り替え、および異なる表現の中のメディアコンポーネントの結合プレゼンテーションなどの進んだ機能を提供できるように、メディアコンポーネントを各々含む、セグメントの構造化された集合体であるメディアプレゼンテーションを記述する。クライアントは、サービスのプロビジョニングのために、様々な方法でmedia presentation description情報を使用することができる。具体的には、データがクライアントの能力およびストリーミングサービス内のユーザに対して有用であるように、media presentation descriptionから、HTTPストリーミングクライアントは、集合体の中のどのセグメントがアクセスされ得るかを判定することができる。10

#### 【0025】

いくつかの実施形態では、media presentation descriptionは不变であり得るが、セグメントは動的に作成され得る。media presentation descriptionは、サービスに対するアクセスおよびダウンロードの時間を最小限にするために、可能な限り小型であり得る。他の専用のサーバ接続性、たとえば、クライアントとサーバとの間の定期的または頻繁なタイミングの同期は、最小限にされ得る。

#### 【0026】

メディアプレゼンテーションは、異なるタイプのアクセスネットワークへのアクセス、異なる現在のネットワーク状態、ディスプレイサイズ、アクセスピットレート、およびコーデックのサポートなどの、異なる能力を伴う端末によるアクセスを許容するように構築され得る。クライアントは次いで、適切な情報を抽出して、ストリーミングサービスをユーザに提供することができる。20

#### 【0027】

media presentation descriptionはまた、要件に従って、デプロイメントの柔軟性および小型化を可能にし得る。

#### 【0028】

最も単純な場合には、各Alternative Representationは、単一の3GPファイル、すなわち、3GPP TS26.244で定義されるものに準拠するファイルに、または、ISO/IEC 14496-1または派生した仕様(3GPP Technical Specification 26.244に記載される3GPファイルフォーマットなど)において定義されるようなISOベースのメディアファイルフォーマットに準拠する任意の他のファイルに、記憶され得る。この文書の残りの部分では、3GPファイルに言及するとき、ISO/IEC 14496-12および派生した仕様は、すべての説明された特徴を、ISO/IEC 14496-12または任意の派生した仕様において定義されるようなより一般的なISOベースのメディアファイルフォーマットへと対応付けることができることを理解されたい。クライアントは次いで、ファイルの最初の部分を要求して、メディアのメタデータ(通常は、「moov」ボックスとも呼ばれるMovieヘッダボックスに記憶される)を、ムービーフラグメントの時間およびバイトのオフセットとともに知ることができる。クライアントは次いで、HTTP partial get要求を出して、要求されたムービーフラグメントを取得することができる。3040

#### 【0029】

いくつかの実施形態では、各表現をいくつかのセグメントへと分割することが望ましいことがある。セグメントフォーマットが3GPファイルフォーマットに基づく場合、セグメントは、「時間的分割」と呼ばれる、ムービーフラグメントの重複しない時間スライスを含む。これらのセグメントの各々は、複数のムービーフラグメントを含んでよく、各々は、それ自体が有効な3GPファイルであり得る。別の実施形態では、表現は、メタデータ(通常はMovieヘッダ「moov」ボックス)を含む初期セグメントとメディアセグメントのセットとに分割され、各メディアセグメントがメディアデータを含み、初期セグメントおよび任意のメディアセグメントの連結物が有効な3GPファイルを形成するとともに、初期セグメ50

ントおよび1つの表現のすべてのメディアセグメントの連結物が有効な3GPファイルを形成する。全体の表現が、各セグメントを再生し、次いで各表現の開始時間に従って、ファイル内のローカルのタイムスタンプをグローバルなプレゼンテーション時間と対応付けることによって形成され得る。

#### 【0030】

この説明の全体で、「セグメント」への言及は、完全にもしくは部分的に構築される、または記憶媒体から読み取られる、または、たとえばHTTP要求を含むファイルダウンロードプロトコル要求の結果として別様に取得される任意のデータオブジェクトを含むものと理解されるべきであることに、留意されたい。たとえば、HTTPの場合、データオブジェクトは、ディスク上、またはHTTPサーバに接続される、もしくはその一部を形成する他の記憶媒体上に存在する実際のファイルに記憶されてよく、または、データオブジェクトは、HTTP要求に応答して実行される、CGIスクリプト、または他の動的に実行されるプログラムによって構築され得る。「ファイル」および「セグメント」という用語は、別段規定されない限り、本文書では同義的に使用される。HTTPの場合、セグメントは、HTTP要求応答のエンティティボディであると見なされ得る。10

#### 【0031】

「プレゼンテーション」および「コンテンツアイテム」という用語は、本文書では同義的に使用される。多くの例において、プレゼンテーションは、定められた「再生」時間を有する、オーディオ、ビデオ、または他のメディアプレゼンテーションであるが、他の変形も可能である。20

#### 【0032】

「ロック」および「フラグメント」という用語は、別段規定されない限り、本文書では同義的に使用され、インデックスが付けられたデータの最小の集合体を全般に指す。利用可能なインデクシングに基づいて、クライアントは、異なるHTTP要求でフラグメントの異なる部分を要求することができ、または、1つのHTTP要求で1つまたは複数の連続的なフラグメントまたはフラグメントの部分を要求することができる。ISOベースのメディアファイルフォーマットベースのセグメントまたは3GPファイルフォーマットベースのセグメントが使用される場合、フラグメントは通常、ムービーフラグメントヘッダ(「moof」)ボックスとメディアデータ(「mdat」)ボックスの組合せとして定義されるムービーフラグメントを指す。30

#### 【0033】

本明細書では、データを搬送するネットワークは、本明細書の説明を簡単にするために、パケットベースであると仮定され、本開示を読んだ後、当業者は、本明細書で説明された本発明の実施形態を、連続的なビットストリームネットワークなどの他のタイプの送信ネットワークに適用できることが認識される。

#### 【0034】

本明細書では、FEC符号は、本明細書の説明を簡単にするために、データの長くて変わりやすい配信時間に対する保護を行うと仮定され、本開示を読んだ後、当業者は、本発明の実施形態を、データのビット反転破損などの他のタイプのデータ送信の問題に適用できることが認識される。たとえば、FECがない場合、要求されたフラグメントの最後の部分がはるかに遅く到達すると、または、フラグメントの前の部分よりも到達時間の変動が大きいと、コンテンツザッピング時間が大きく、および変わりやすくなり得るが、FECおよび並列要求を使用すると、フラグメントに対して要求されたデータの過半のみ、データが復元され得る前に到達する必要があるので、コンテンツザッピング時間が短くなりコンテンツザッピング時間の変わりやすさが小さくなる。この説明では、符号化されるべきデータ(たとえば、ソースデータ)は、任意の長さ(最小で単一ビット)であり得る等しい長さの「シンボル」へと分解されていると仮定され得るが、シンボルは、データの異なる部分に対しては異なる長さであってよく、たとえば、異なるシンボルサイズがデータの異なるロックに対して使用され得る。40

#### 【0035】

50

この説明では、本明細書の説明を簡単にするために、FECが一度にデータの「ブロック」または「フラグメント」に適用されることが仮定され、すなわち、「ブロック」は、FECの符号化および復号のための「ソースブロック」である。クライアントデバイスは、本明細書で説明されたセグメントインデクシングを使用して、セグメントのソースブロック構造を決定するのを助けることができる。当業者は、他のタイプのソースブロック構造に本発明の実施形態を適用することができ、たとえば、ソースブロックは、フラグメントの一部分であってよく、または、1つまたは複数のフラグメントまたはフラグメントの部分を包含し得る。

#### 【0036】

ブロック要求ストリーミングとともに使用することが考えられるFEC符号は通常、システムティックなFEC符号であり、すなわち、ソースブロックのソースシンボルは、ソースブロックの符号化の一部として含まれ得るので、ソースシンボルが送信される。当業者が認識するように、本明細書で説明される実施形態は、システムティックではないFEC符号に等しく良好に適用される。システムティックなFECエンコーダは、ソースシンボルのソースブロックから、ある数の修復シンボルを生成し、ソースシンボルと修復シンボルの少なくともいくつかの組合せが、ソースブロックを表すチャネルを通じて送信される符号化シンボルである。いくつかのFEC符号が、「情報追加符号」または「噴水符号」のような、必要なだけの修復シンボルを効率的に生成するのに有用であることがあり、これらの符号の例には、「連鎖反応符号」および「多段階連鎖反応符号」がある。リードソロモン符号のような他のFEC符号は、実際には、各ソースブロックに対して限られた数の修復シンボルしか生成できない。

#### 【0037】

これらの例の多くにおいて、クライアントがメディアサーバまたは複数のメディアサーバに結合されること、および、クライアントがメディアサーバまたは複数のメディアサーバから1つまたは複数のチャネルを通じてストリーミングメディアを要求することが仮定される。しかしながら、より複雑な構成も可能である。

#### 【0038】

##### 利点の例

ブロック要求ストリーミングでは、メディアクライアントは、これらのブロック要求のタイミングと、ユーザに対するメディア再生のタイミングとの関係を維持する。このモデルは、上で説明された「ダウンロード」モデルの利点を保持しながら、通常発生する、データ配信に対するメディア再生の関係の解消に起因する欠点のいくつかを避けることができる。ブロック要求ストリーミングモデルは、TCPのようなトランスポートプロトコルにおいて利用可能な、レートおよび混雑制御機構を利用して、最大の利用可能な帯域幅がメディアデータのために使用されることを確実にする。加えて、ブロックへのメディアプレゼンテーションの分割は、符号化メディアデータの各ブロックが、複数の利用可能な符号化のセットから選択されることを可能にする。

#### 【0039】

この選択は、利用可能な帯域幅が時間とともに変化している場合でも利用可能な帯域幅に対してメディアデータレートが一致すること、クライアントの能力または構成に対してメディアの解像度もしくは復号の複雑さが適合していること、または、言語などのユーザ選好に対して適合していることを含む、任意の数の基準に基づき得る。この選択はまた、アクセシビリティコンポーネント、クローズドキャプション、サブタイトル、手話ビデオなどのような、補助コンポーネントのダウンロードおよび提示を含み得る。ブロック要求ストリーミングモデルを使用する既存のシステムの例には、Move Networks(商標)、Microsoft Smooth Streaming、およびApple iPhone(商標) Streaming Protocolがある。

#### 【0040】

一般に、メディアデータの各ブロックは、個別のファイルとしてサーバに記憶されてよく、次いでHTTPのようなプロトコルが、ユニットとしてファイルを要求するために、サーバ上で実行されるHTTPサーバソフトウェアとともに使用される。通常、クライアントはメ

10

20

30

40

50

タデータファイルを与えられ、メタデータファイルはたとえば、Extensible Markup Language(XML)フォーマットまたはプレイリストテキストフォーマットまたはバイナリフォーマットであってよく、利用可能な符号化物(たとえば、必要な帯域幅、解像度、符号化パラメータ、メディアタイプ、言語)のような、この文書では通常「表現」と呼ばれるメディアプレゼンテーションの特徴と、符号化物がブロックへと分割された方式とを記述する。たとえば、メタデータは、各ブロックのUniform Resource Locator(URL)を含み得る。文書化されたリソースにアクセスするために使用されるべきプロトコルがHTTPであることを示すために、URL自体が、文字列「http://」が追加されたものなどの方式を提供することができる。別の例は、使用されるべきプロトコルがFTPであることを示すための「ftp://」である。

10

#### 【0041】

他のシステムでは、たとえば、メディアブロックは、要求されているメディアプレゼンテーションの部分を示す、クライアントからの要求に応答して、サーバによって「その場で」時間内に構築され得る。たとえば、「http://」方式のHTTPの場合、このURLの要求の実行は、そのエンティティボディに何らかの特定のデータを含む、要求応答を提供する。この要求応答をどのように生成するかについてのネットワークにおける実装は、そのような要求にサービスするサーバの実装形態に応じて大きく異なり得る。

#### 【0042】

通常、各ブロックは、独立に復号可能であり得る。たとえば、ビデオメディアの場合、各ブロックは「探索点」で開始し得る。いくつかのコーディング方式では、探索点は、「ランダムアクセスポイント」または「RAP」と呼ばれるが、すべてのRAPが探索点として指定されなくてもよい。同様に、他のコーディング方式では、探索点は、H.264ビデオ符号化の場合、「Independent Data Refresh」フレームまたは「IDR」で開始するが、すべてのIDRが探索点として指定されなくてもよい。探索点は、デコーダが以前のフレームまたはデータまたはサンプルについてのデータを何ら必要とすることなく復号を開始できる、ビデオ(または他の)メディア中の位置であり、復号されているフレームまたはサンプルがスタンドアロン方式で符号化されず、たとえば、現在のフレームと以前のフレームの差分として符号化された場合には、デコーダが以前のフレームまたはデータまたはサンプルについてのデータを必要とすることがある。

20

#### 【0043】

そのようなシステムにおける課題は、ストリームの再生を開始する能力、たとえば、パソコン用コンピュータを使用して、受信されたオーディオおよびビデオストリームを復号しレンダリングし、コンピュータスクリーン上にビデオを表示しつつ内蔵スピーカーを通じてオーディオを再生すること、または別の例として、セットトップボックスを使用して、受信されたオーディオおよびビデオストリームを復号しレンダリングし、テレビジョンディスプレイデバイス上にビデオを表示しつつステレオシステムを通じてオーディオを再生することであり得る。主な課題は、ユーザがストリームとして配信された新たなコンテンツを見ることを決めて、その決定を表す動作をとるとき、たとえば、ユーザがブラウザウィンドウ内のリンクをクリックする、またはリモートコントロールデバイスの再生ボタンを押すときから、コンテンツがユーザのスクリーン上に表示され始めるときまでの遅延、を最小限にすることであってよく、この遅延は以後「コンテンツザッピング時間」と呼ばれる。これらの課題の各々が、本明細書で説明される改善されたシステムの要素によって対処され得る。

30

#### 【0044】

コンテンツザッピングの例は、ユーザが第1のストリームを介して配信された第1のコンテンツを見ており、次いでユーザが、第2のストリームを介して配信された第2のコンテンツを見ると決め、第2のコンテンツを見ることを始めるための動作を開始するときである。第2のストリームは、第1のストリームと同じセットのサーバまたは異なるセットのサーバから送信され得る。コンテンツザッピングの別の例は、ユーザがあるウェブサイトを訪問しており、ブラウザウィンドウ内のリンクをクリックすることによって、第1のストリ

40

50

ームを介して配信された第1のコンテンツを見るのを始めると決めるときである。同様の方式で、ユーザは、初めからではなく、ストリーム内の何らかの時間からコンテンツの再生を始めると決めることができる。ユーザは、時間位置を探索することをクライアントデバイスに対して示し、ユーザは、選択された時間が瞬時にレンダリングされることを期待し得る。コンテンツザッピング時間を最小限にすることは、多種多様な利用可能なコンテンツを検索して試すときに、高品質な高速のコンテンツサーフィン体験をユーザに対して可能にするために、ビデオ視聴にとって重要である。

#### 【0045】

最近では、送信の間のストリーミングメディアの保護のために前方誤り訂正(FEC)符号を使用するのを検討することが、一般的となっている。3GPP、3GPP2、およびDVBのようなグループによって標準化されたようなインターネットおよびワイヤレスネットワークをその例に含むパケットネットワークを通じて送信される場合、ソースストリームは、生成されまたは利用可能にされるときにパケットへと置かれるので、パケットは、生成され受信者に対して利用可能にされる順序でソースまたはコンテンツストリームを搬送するために使用され得る。10

#### 【0046】

FEC符号のこれらのタイプのシナリオへの一般的な適用では、エンコーダは、修復パケットの作成においてFEC符号を使用することができ、修復パケットは次いで、ソースストリームを含む元のソースパケットに加えて送信される。修復パケットは、ソースパケットの喪失が発生すると、喪失ソースパケットに含まれるデータを復元するために受信された修復パケットが使用され得るという、性質を有する。修復パケットは、完全に失われた喪失ソースパケットのコンテンツを復元するために使用され得るが、完全に受信された修復パケット、またはさらには、部分的に受信された修復パケットのいずれかである、修復パケットが、部分的なパケット喪失の発生から回復するために使用され得る。したがって、完全にまたは部分的に失われたソースパケットを復元するために、完全にまたは部分的に受信された修復パケットが使用され得る。20

#### 【0047】

さらに他の例では、他のタイプの破損が送信データに発生することがあり、たとえば、ビットの値が反転があるので、修復パケットが、そのような破損を訂正し、ソースパケットの可能な限り正確な回復を実現するために使用され得る。他の例では、ソースストリームは、必ずしも個別のパケットで送信されず、代わりに、たとえば連続的なビットストリームとして送信され得る。30

#### 【0048】

ソースストリームの保護を行うために使用され得る、多くのFEC符号の例がある。リードソロモン符号は、通信システムにおける誤りおよび抹消の訂正のためのよく知られている符号である。たとえば、パケットデータネットワークを通じた抹消訂正のために、リードソロモン符号のよく知られている効率的な実装形態は、L. Rizzo、「Effective Erasure Codes for Reliable Computer Communication Protocols」、Computer Communication Review、27(2):24-36(1997年4月)(以後「Rizzo」)、および、Bloemer他、「An XOR-Based Erasure-Resilient Coding Scheme」、Technical Report TR-95-48、International Computer Science Institute、カリフォルニア州バークレー(1995)(以後「XOR-Reed-Solomon」)または他で説明されるように、コーチー行列またはファンデルモンド行列を使用する。40

#### 【0049】

FEC符号の他の例には、LDPC符号、Luby<sup>1</sup>で説明されるような連鎖反応符号、および、Shokrollahi<sup>1</sup>におけるような多段階連鎖反応符号がある。

#### 【0050】

リードソロモン符号の変形のためのFEC復号プロセスの例は、RizzoおよびXOR-Reed-Solomonで説明されている。それらの例では、復号は、十分なソースおよび修復データパケットが受信された後で適用され得る。復号プロセスは、計算集約的であることがあり、利用50

可能なCPUリソースによっては、復号プロセスは、ブロックの中でメディアがまたがる時間の長さに対して、完了するのにかなりの時間がかかることがある。受信機は、メディアストリームの受信の開始とメディアの再生との間で必要とされる遅延を計算するとき、復号のために必要とされるこの時間の長さを考慮し得る。復号によるこの遅延は、特定のメディアストリームに対するユーザの要求から、再生の開始までの遅延として、ユーザにより知覚される。したがって、この遅延を最小限にすることが望ましい。

#### 【0051】

多くの適用形態において、パケットはさらに、FECプロセスが適用されるシンボルへと分割され得る。パケットは、1つまたは複数のシンボルを含んでよい(または1つのシンボル未満を含んでよいが、通常、シンボルは、パケットのグループの間での誤り条件が大きく相關していることが知られていない限り、パケットのグループにまたがって分割されない)。シンボルは任意のサイズを有し得るが、シンボルのサイズは最大でも、パケットのサイズに等しいことが多い。ソースシンボルは、送信されるべきデータを復号するシンボルである。修復シンボルは、ソースシンボルに加えて、ソースシンボルから直接または間接的に生成されるシンボルである(すなわち、送信されるべきデータは、ソースシンボルのすべてが利用可能であり修復シンボルのいずれもが利用可能ではない場合、完全に復元され得る)。

10

#### 【0052】

いくつかのFEC符号は、その符号化動作が、ブロック中にあるシンボルに依存し、そのブロック中にはないシンボルに対して独立であり得るという点で、ブロックベースであり得る。ブロックベースの符号化では、FECエンコーダは、そのブロック中のソースシンボルからあるブロックのための修復シンボルを生成し、次いで次のブロックに移ることができ、符号化されている現在のブロックに対するもの以外のソースシンボルを参照する必要がない。送信において、ソースシンボルを含むソースブロックは、符号化シンボル(これらは、いくつかのソースシンボル、いくつかの修復シンボル、またはこれら両方であり得る)を含む符号化ブロックによって表され得る。修復シンボルの存在により、ソースシンボルのすべてが各々の符号化ブロックにおいて必要とはされなくなる。

20

#### 【0053】

いくつかのFEC符号、特にリードソロモン符号では、ソースブロック当たりの符号化シンボルの数が増えるにつれて、符号化および復号の時間が非実用的に長くなり得る。したがって、実際には、特に、リードソロモン符号化または復号プロセスがカスタムハードウェアによって実行される典型的な場合、たとえば、パケット喪失に対してストリームを保護するためにDVB-H規格の一部として含まれるリードソロモン符号を使用するMPE-FECプロセスが、ソースブロック当たり全体で255個のリードソロモン符号化シンボルに制限される携帯電話内の特別なハードウェアにおいて実装される場合、ソースブロック当たりの生成され得る符号化シンボルの総数に対して、実用的な上限(いくつかの適用形態では、255がおよその実用的な限界である)が存在することが多い。シンボルは別個のパケットペイロードへと置かれることが必要とされることが多いので、このことが、符号化されるソースブロックの最大の長さに対して実用的な上限を課す。たとえば、パケットペイロードが1024バイト以下に制限され、各パケットが1つの符号化シンボルを搬送する場合、符号化ソースブロックは、最大でも255キロバイトであってよく、これは当然、ソースブロック自体のサイズの上限もある。

30

#### 【0054】

ソースのストリーミングレートについていくのに十分高速にソースブロックを復号できることなどの、FEC復号によってもたらされる復号のレイテンシを最小限にして、FEC復号の間の任意の時点で受信デバイスにおいて利用可能なCPUの小さな断片のみを使用するための他の課題が、本明細書で説明される要素により対処される。

40

#### 【0055】

必要とされているのは、システムのコンポーネントが故障しても受信機に配信されるストリームの品質に悪影響を与えることのない、ロバストで拡張性のあるストリーム配信の

50

方法を提供することである。

#### 【0056】

ブロック要求ストリーミングシステムは、プレゼンテーションの構造またはメタデータに対する変更、たとえば、利用可能なメディア符号化物の数に対する変更、または、ビットレート、解像度、アスペクト比、オーディオもしくはビデオコーデック、またはコーデックパラメータなどのメディア符号化物のパラメータに対する変更、または、コンテンツファイルと関連付けられるURLなどの他のメタデータの変更をサポートする必要がある。そのような変更は、より大きなプレゼンテーションの異なるセグメントの告知、たとえば構成の変更、機器の故障もしくは機器の故障からの回復、または他の理由でサービングインフラストラクチャが変わった結果として必要となるURLまたは他のパラメータの修正などの、異なるソースからのコンテンツと一緒に編集することを含む、数々の理由で必要とされ得る。10

#### 【0057】

プレゼンテーションが継続的に更新されるプレイリストファイルによって制御され得る、方法が存在する。このファイルは継続的に更新されるので、上で説明された変更の少なくとも一部は、これらの更新の中で行われ得る。従来の方法の欠点は、「ポーリング」とも呼ばれる、クライアントデバイスがプレイリストファイルを継続的に取り出すことを行わなければならず、サービングインフラストラクチャに負荷がかかること、および、このファイルが更新の期間よりも長くキャッシュされることが不可能であり、サービングインフラストラクチャのタスクがはるかに難しくなることである。上で説明された種類の更新がメタデータファイルに対するクライアントによる継続的なポーリングを必要とせずに行われるように、上記のことが本明細書の要素によって対処される。20

#### 【0058】

特にライブサービスにおける、プロードキャスト配信から通常知られている別の問題は、ユーザが番組に参加した時点よりも前にプロードキャストされたコンテンツをユーザが見ることができないということである。通常、ローカルでの個人による録画は、不必要にローカルの記憶容量を消費し、または、クライアントが番組に合わされていなかったので不可能であり、または、コンテンツ保護規則によって禁止されている。ネットワーク録画およびタイムシフト視聴は好ましいが、サーバへのユーザの個々の接続と、ライブサービスとは別個の配信プロトコルおよびインフラストラクチャとを必要とし、インフラストラクチャの重複および大きなサーバのコストをもたらす。このことも、本明細書で説明される要素によって対処される。30

#### 【0059】

##### システムの概要

本発明の一実施形態が図1を参照して説明され、図1は、本発明を具現化するブロック要求ストリーミングシステムの簡略化された図を示す。

#### 【0060】

図1では、取込システム103を含むブロックサービングインフラストラクチャ(「BSI」)101を含むブロックストリーミングシステム100が示されており、取込システム103は、コンテンツ102を取り込み、取込システム103とHTTPストリーミングサーバ104の両方へアクセス可能なコンテンツ記憶装置110へとそのコンテンツを記憶することによって、HTTPストリーミングサーバ104によるサービスのためにそのコンテンツを準備してパッケージングするためのものである。示されるように、システム100は、HTTPキャッシュ106も含み得る。動作において、HTTPストリーミングクライアントのようなクライアント108は、要求112をHTTPストリーミングサーバ104へ送信し、HTTPストリーミングサーバ104またはHTTPキャッシュ106から応答114を受信する。各々の場合において、図1に示される要素は、少なくとも一部は、プロセッサまたは他の電子回路で実行されるプログラムコードを含むソフトウェアで実装されてよい。40

#### 【0061】

コンテンツは、ムービー、オーディオ、2D平面ビデオ、3Dビデオ、他のタイプのビデオ50

、画像、時限のテキスト、時限のメタデータなどを含み得る。いくつかのコンテンツは、補助情報(局の識別情報、広告、株価、Flash(商標)シーケンスなど)を、再生されている他のメディアとともに提示するためのデータなどの、時限の方式で提示または利用されるべきデータを伴い得る。他のメディアを組み合わせる、かつ/または単なるオーディオおよびビデオを超える、他のハイブリッドプレゼンテーションも使用され得る。

#### 【0062】

図2で示されるように、メディアブロックは、たとえば、HTTPサーバ、コンテンツ配信ネットワークデバイス、HTTPプロキシ、FTPプロキシもしくはサーバ、または何らかの他のメディアサーバもしくはシステムであり得る、ブロックサービングインフラストラクチャ101(1)内に記憶され得る。ブロックサービングインフラストラクチャ101(1)は、たとえば、インターネットのようなインターネットプロトコル('IP')ネットワークであってよい、ネットワーク122に接続される。ブロック要求ストリーミングシステムクライアントは、6個の機能的コンポーネント、すなわち、上で説明されたメタデータを与えられ、メタデータによって示される複数の利用可能なブロックの中から要求されるべきブロックまたは部分ブロックを選択する機能を実行する、ブロック選択器123、ブロック選択器123から要求命令を受信し、規定されたブロック、ブロックの部分、または複数のブロックに対する要求を、ネットワーク122を通じてブロックサービングインフラストラクチャ101(1)へ送信し、返答としてそのブロックを含むデータを受信するのに必要な動作を実行する、ブロック要求器124、さらには、ブロックバッファ125、バッファモニタ126、メディアデコーダ127、およびメディアの利用を容易にする1つまたは複数のメディアトランスデューサ128を、有するものとして示されている。10

#### 【0063】

ブロック要求器124によって受信されるブロックデータは、一時的な記憶のために、メディアデータを記憶するブロックバッファ125へと渡される。あるいは、受信されたブロックデータは、図1に示されるように、ブロックバッファ125へと直接記憶されてもよい。メディアデコーダ127は、ブロックバッファ125によってメディアデータを与えられ、メディアトランスデューサ128へ適切な入力を与えるために必要な変換を、このデータに対して実行し、メディアトランスデューサ128は、メディアをユーザの利用または他の利用に適切な形態にする。メディアトランスデューサの例には、携帯電話、コンピュータシステムまたはテレビにおいて見出されるような、視覚ディスプレイデバイスが含まれ、スピーカーまたはヘッドフォンのような音声表現デバイスも含まれ得る。20

#### 【0064】

メディアデコーダの例は、H.264ビデオコーディング規格において説明されるフォーマットのデータを、各フレームまたはサンプルのための関連するプレゼンテーションタイムスタンプを伴うYUVフォーマットピクセルマップのような、ビデオフレームのアナログまたはデジタルのプレゼンテーションへと変換する、関数であろう。

#### 【0065】

バッファモニタ126は、ブロックバッファ125のコンテンツに関する情報を受信し、この情報および場合によっては他の情報に基づいて、本明細書で説明されるような、要求すべきブロックの選択を決定するのに使用されるブロック選択器123へ、入力を与える。40

#### 【0066】

本明細書で使用される用語では、各ブロックは、受信機が通常の速度でそのブロックに含まれるメディアを再生するのにかかるであろう時間の量を表す、「再生時間」または「継続時間」を有する。いくつかの場合には、ブロック内のメディアの再生は、前のブロックからのすでに受信されているデータに依存し得る。まれに、ブロック中のメディアの一部の再生は、後続のブロックに依存することがあり、この場合、ブロックの再生時間は、後続のブロックを参照することなくブロック内で再生され得るメディアに関して定義され、後続のブロックの再生時間は、後続のブロックを受信した後にのみ再生できるこのブロック内のメディアの再生時間の分だけ増える。後続ブロックに依存するブロック中のメディアを含むのはまれであるので、本開示の残りの部分では、1つのブロック中のメディア50

は後続のブロックに依存しないことを仮定するが、この変形が以下で説明される実施形態に容易に追加され得ることを当業者が認識するであろうことに留意されたい。

#### 【0067】

受信機は、「一時停止」、「早送り」、「巻き戻し」などのような制御を有し得る。これは、異なるレートでの再生によるブロックの消費をもたらし得るが、受信機が、シーケンス中の最後のブロックを除く合計の再生時間以下の合計時間でブロックの各々の連続的なシーケンスを取得し復号できる場合、受信機は、ストールすることなくメディアをユーザに提示することができる。本明細書のいくつかの説明では、メディアストリーム中の特定の位置は、メディア中の特定の「時間」と呼ばれ、これは、メディア再生の開始から、ビデオストリーム中の特定の位置に到達する時間までに経過した時間に対応する。メディアストリーム中の時間または位置は、従来からの概念である。たとえば、ビデオストリームが毎秒24個のフレームを含む場合、最初のフレームは、 $t=0.0$ 秒という位置または時間 10 を有すると言われることがあり、241番目のフレームは、 $t=10.0$ 秒という位置または時間を有すると言われることがある。当然、フレームベースのビデオストリームでは、位置または時間は連続的である必要はなく、それは、241番目のフレームの最初のビットから242番目のフレームの最初のビットの直前までのストリーム中のビットの各々が、すべて同じ時間の値を有し得るからである。

#### 【0068】

上記の用語を採用すると、ブロック要求ストリーミングシステム(BRSS)は、1つまたは複数のコンテンツサーバ(たとえば、HTTPサーバ、FTPサーバなど)へ要求を行う、1つまたは複数のクライアントを含む。取込システムは、1つまたは複数の取込プロセッサを含み、取込プロセッサは、コンテンツを(リアルタイムでまたはそれ以外で)受信し、BRSSによる使用のためにコンテンツを処理し、コンテンツサーバへアクセス可能な記憶装置に、場合によっては取込プロセッサによって生成されたメタデータとともに、コンテンツを記憶する。

#### 【0069】

BRSSは、コンテンツサーバと協調するコンテンツキャッシュも含み得る。コンテンツサーバおよびコンテンツキャッシュは、URLを含むHTTP要求の形態でファイルまたはセグメントに対する要求を受信する、従来のHTTPサーバおよびHTTPキャッシュであってよく、URLによって示されるファイルまたはセグメントのすべてよりも少数のものを要求するために、バイト範囲も含み得る。クライアントは、HTTPサーバの要求を行い、そうした要求への応答を処理する、従来のHTTPクライアントを含んでよく、ここでHTTPクライアントは、要求を作成し、その要求をHTTPクライアントへ渡し、HTTPクライアントから応答を取得し、クライアントデバイスによる再生のために表示プレーヤーへとその応答を与えるためにその応答を処理する(たとえば、記憶する、変換するなど)、新規のクライアントシステムによって動作させられる。通常、クライアントシステムは、どのメディアが必要になるかを事前には知らず(必要性はユーザの入力に応じたものであり得るので、ユーザの入力が変化するので、など)、そのため、受信されるとすぐに、または受信の後まもなく、メディアが「消費される」という点で、これは「ストリーミング」システムと呼ばれる。その結果、応答の遅延および帯域幅の制約は、プレゼンテーションに遅延を引き起こすことがあり、たとえば、ユーザがプレゼンテーションを見ている箇所にストリームが追いつくと、プレゼンテーションに一時停止が生じる。

#### 【0070】

良好な品質であると知覚される表示を提供するために、クライアント側、取込側、またはこれら両方のいずれかにおいて、いくつかの細部がBRSSに実装され得る。一部の場合には、実装される細部は、ネットワークにおけるクライアントとサーバとのインターフェースを考慮して、およびそれに対応するようになされる。いくつかの実施形態では、クライアントシステムと取込システムの両方が改善を認識し、他の実施形態では、片側のみが改善を認識する。そのような場合、片側が改善を認識していなくても、システム全体が改善から利益を得て、他の場合には、双方が改善を認識する場合にのみ利益が生じるが、片側

10

20

30

40

50

が改善を認識しなくても、システムは破綻することなく動作し続ける。

#### 【0071】

図3に示されるように、様々な実施形態に従って、取込システムは、ハードウェアコンポーネントとソフトウェアコンポーネントの組合せとして実装され得る。取込システムは、本明細書で論じられた方法の任意の1つまたは複数をシステムに実行させるように実行され得る、命令のセットを含み得る。システムは、コンピュータの形態である固有の機械として実現され得る。システムは、サーバコンピュータ、パーソナルコンピュータ(PC)、または、そのシステムによってとられるべき動作を規定する命令のセット(順次的な命令または他の命令)を実行できる、任意のシステムであってよい。さらに、単一のシステムのみが示されているが、「システム」という用語は、本明細書で論じられた方法の任意の1つまたは複数を実行するための命令のセット(または複数のセット)を、個々にまたは一緒に実行する、システムの任意の集合も含むと、解釈されるべきである。10

#### 【0072】

取込システムは、取込プロセッサ302(たとえば、中央演算処理装置(CPU))、実行中のプログラムコードを記憶できるメモリ304、およびディスク記憶装置306を含んでよく、これらのすべてが、バス300を介して互いに通信する。システムはさらに、ビデオディスプレイユニット308(たとえば液晶ディスプレイ(LCD)または陰極線管(CRT))を含み得る。システムはまた、英数字入力デバイス310(たとえばキーボード)、および、コンテンツのソースを受信しかつコンテンツ記憶装置に配信するための、ネットワークインターフェースデバイス312を含み得る。20

#### 【0073】

ディスク記憶ユニット306は、本明細書で説明される方法または機能の任意の1つまたは複数を具現化する、命令(たとえばソフトウェア)の1つまたは複数のセットが記憶され得る、機械可読媒体を含み得る。命令はまた、システムによるその実行の間、完全にまたは少なくとも部分的に、メモリ304内、および/または取込プロセッサ302内に存在してよく、メモリ304および取込プロセッサ302も、機械可読媒体を構成する。

#### 【0074】

図4に示されるように、様々な実施形態に従って、クライアントシステムは、ハードウェアコンポーネントとソフトウェアコンポーネントの組合せとして実装され得る。クライアントシステムは、本明細書で論じられた方法の任意の1つまたは複数をシステムに実行させるように実行され得る、命令のセットを含み得る。システムは、コンピュータの形態である固有の機械として実現され得る。システムは、サーバコンピュータ、パーソナルコンピュータ(PC)、または、そのシステムによってとられるべき動作を規定する命令のセット(順次的な命令または他の命令)を実行できる、任意のシステムであってよい。さらに、単一のシステムのみが示されているが、「システム」という用語は、本明細書で論じられた方法の任意の1つまたは複数を実行するための命令のセット(または複数のセット)を、個々にまたは一緒に実行する、システムの任意の集合も含むと、解釈されるべきである。30

#### 【0075】

クライアントシステムは、クライアントプロセッサ402(たとえば、中央演算処理装置(CPU))、実行中のプログラムコードを記憶できるメモリ404、およびディスク記憶装置406を含んでよく、これらのすべてが、バス400を介して互いに通信する。システムはさらに、ビデオディスプレイユニット408(たとえば液晶ディスプレイ(LCD)または陰極線管(CRT))を含み得る。システムはまた、英数字入力デバイス410(たとえばキーボード)、および、要求を送信し応答を受信するための、ネットワークインターフェースデバイス412を含み得る。40

#### 【0076】

ディスク記憶ユニット406は、本明細書で説明される方法または機能の任意の1つまたは複数を具現化する、命令(たとえばソフトウェア)の1つまたは複数のセットが記憶され得る、機械可読媒体を含み得る。命令はまた、システムによるその実行の間、完全にまたは少なくとも部分的に、メモリ404内、および/またはクライアントプロセッサ402内に存在50

してよく、メモリ404およびクライアントプロセッサ402も、機械可読媒体を構成する。

#### 【0077】

##### 3GPPファイルフォーマットの使用法

3GPPファイルフォーマット、または、MP4ファイルフォーマットまたは3GPP2ファイルフォーマットのようなISOベースのメディアファイルフォーマットに基づく任意の他のファイルが、以下の特徴を伴うHTTPストリーミングのためのコンテナフォーマットとして使用され得る。要求されたようなファイルまたはメディアセグメントの適切な断片をクライアントがダウンロードできるように、セグメントインデックスが、時間オフセットおよびバイト範囲をシグナリングするために各セグメント中に含まれ得る。メディアプレゼンテーション全体のグローバルなプレゼンテーションのタイミング、および、各々の3GPファイルまたはメディアセグメント内のローカルなタイミングは、正確に揃えられ得る。1つの3GPファイルまたはメディアセグメント内のトラックは、正確に揃えられ得る。複数の表現にまたがるトラックも、グローバルな時間軸に表現の各々を割り当てるこによって揃えられ得るので、表現の切り替えはシームレスであり得るとともに、異なる表現の中のメディアコンポーネントの結合プレゼンテーションが同期され得る。10

#### 【0078】

ファイルフォーマットは、以下の特性を伴う適応ストリーミングのためのプロファイルを含み得る。すべてのムービーデータはムービーフラグメントに含まれてよく、「moov」ボックスはサンプル情報を何ら含まなくてよい。オーディオおよびビデオサンプルデータは、TS26.244で規定されるようなプログレッシブダウンロードプロファイルに対する要件と同様の要件を伴って、インターリープされ得る。「moov」ボックスは、ファイルの始めに置かれてよく、それに続いて、セグメントインデックスとも呼ばれ、含んでいるセグメント中の各フラグメントまたはフラグメントの少なくとも1つのサブセットに対する時間およびバイト範囲のオフセット情報を含む、フラグメントオフセットデータが置かれてよい。20

#### 【0079】

既存のプログレッシブダウンロードプロファイルに従うファイルを、Media Presentation Descriptionが参照することも可能であり得る。この場合、クライアントは、Media Presentation Descriptionを使用して、複数の利用可能なバージョンの中から適切な代替のバージョンを単に選択することができる。クライアントはまた、プログレッシブダウンロードプロファイルに適合するファイルとともにHTTP partial get要求を使用して、各代替的なバージョンのサブセットを要求し、これによって、より効率性の低い形式の適応ストリーミングを実施し得る。この場合、プログレッシブダウンロードプロファイル中のメディアを含む異なる表現は、共通のグローバルな時間軸を依然として堅持して、複数の表現の間のシームレスな切り替えを可能にし得る。30

#### 【0080】

##### 進化した方法の概要

以下のセクションでは、改善されたブロック要求ストリーミングシステムのための方法が説明される。これらの改善の一部は、適用形態における必要性に応じて、これらの改善の他のものとともに、またはそれを伴わずに使用され得ることを理解されたい。一般的な動作では、受信機は、データの特定のブロックまたはブロックの部分に対する、サーバまたは他の送信機の要求を行う。セグメントとも呼ばれるファイルは、複数のブロックを含んでよく、メディアプレゼンテーションの1つのプレゼンテーションと関連付けられる。40

#### 【0081】

好ましくは、再生時間または復号時間から、セグメント内の対応するブロックまたはフラグメントのバイトオフセットへの対応付けを提供する、「セグメントインデクシング」または「セグメントマップ」とも呼ばれるインデクシング情報が生成される。このセグメントインデクシングは、セグメント内、通常はセグメントの始め(セグメントマップの少なくとも一部は始めにある)に含まれてよく、小さいことが多い。セグメントインデックスはまた、別個のインデックスセグメントまたはファイルにおいて提供され得る。特に、50

セグメントインデックスがセグメントに含まれる場合、受信機は、このセグメントマップの一部またはすべてを迅速にダウンロードして、続いてこれを使用して、時間オフセットと、ファイル内でのそれらの時間オフセットと関連付けられるフラグメントの対応するバイト位置との対応付けを決定することができる。

#### 【0082】

受信機は、バイトオフセットを使用して、関心のある時間オフセットと関連付けられない他のフラグメントと関連付けられるデータのすべてをダウンロードする必要なく、特定の時間オフセットと関連付けられるフラグメントからのデータを要求することができる。このようにして、セグメントマップまたはセグメントインデクシングは、関心のある現在の時間オフセットに関連するセグメントの部分に直接アクセスするための受信機の能力を大きく向上させることができ、コンテンツキャッシング時間の改善、ネットワークの状態が変化するに従ってある表現から別の表現へ迅速に変更する能力、および、受信機において再生されていないメディアをネットワークリソースがダウンロードするという無駄の削減を含む、利益を伴う。10

#### 【0083】

ある表現(本明細書では「切替元」表現と呼ばれる)から別の表現(本明細書では「切替先」表現と呼ばれる)への切り替えが考えられる場合、切替先表現の再生がランダムアクセスポイントからシームレスに開始できるように切替元表現の中のメディアがあるプレゼンテーション時間までダウンロードされるという意味で、シームレスな切り替えが可能にされることを確実にするために、セグメントインデックスはまた、切替先表現の中のランダムアクセスポイントの開始時間を特定して、切替元表現の中の要求されるべきデータの量を特定するために使用され得る。20

#### 【0084】

それらのブロックは、要求する受信機が受信機のユーザに対する出力を生成するために必要とするビデオメディアまたは他のメディアのセグメントを表す。コンテンツを送信するサーバから受信機がコンテンツを受信するときなどは、メディアの受信機はクライアントデバイスであり得る。例には、セットトップボックス、コンピュータ、ゲームコンソール、特別な設備を備えたテレビジョン、ハンドヘルドデバイス、特別な設備を備えた携帯電話、または他のクライアント受信機がある。30

#### 【0085】

多くの進化したバッファ管理方法が本明細書で説明される。たとえば、バッファ管理方法は、クライアントが、連続的に再生されるように時間内に受信され得る最高のメディア品質のブロックを要求することを可能にする。可変のブロックサイズという特徴は、圧縮効率を改善する。要求の頻度を制限しながら、要求するデバイスにブロックを送信するために複数の接続をもつことで、改善された送信性能が実現する。データの部分的に受信されたブロックは、メディアプレゼンテーションを継続するために使用され得る。最初に接続をブロックの特定のセットに専用とする必要なく、接続は複数のブロックに対して再使用され得る。複数のクライアントによる複数の可能なサーバの中からのサーバの選択の一貫性が改善され、これは、近くのサーバにおける重複するコンテンツの頻度を下げ、サーバがファイル全体を含む確率を上げる。クライアントは、メディアブロックを含むファイルのURLに組み込まれる、メタデータ(利用可能なメディア符号化物など)に基づいてメディアブロックを要求することができる。システムは、コンテンツの再生がメディア再生において後続の一時停止を引き起こすことなく開始できるまでに必要とされるバッファリング時間の量の、計算および最小化を実現することができる。利用可能な帯域幅のより大きな部分が、必要であれば、最も近い再生時間を伴うブロックに割り振られ得るように、利用可能な帯域幅は、複数のメディアブロックの間で共有され、各ブロックの再生時間が近づくにつれて調整され得る。40

#### 【0086】

HTTPストリーミングは、メタデータを利用し得る。プレゼンテーションレベルのメタデータには、たとえば、ストリーム継続時間、利用可能な符号化物(ビットレート、コード

50

ック、空間解像度、フレームレート、言語、メディアタイプ)、各符号化物のためのストリームメタデータに対するポインタ、および、コンテンツ保護(デジタル著作権管理(DRM)情報)がある。ストリームメタデータは、セグメントファイルのURLであり得る。

#### 【0087】

セグメントメタデータは、セグメント内の要求に対するバイト範囲対時間情報、およびランダムアクセスポイント(RAP)または他の探索点の識別情報を含んでよく、この情報の一部またはすべてが、セグメントインデクシングまたはセグメントマップの一部であり得る。

#### 【0088】

ストリームは、同じコンテンツの複数の符号化物を含み得る。各符号化物は次いで、セグメントへと分解されてよく、各セグメントは記憶単位またはファイルに対応する。HTTPの場合、セグメントは通常、URLによって参照され得るリソースであり、そのようなURLの要求は、要求応答メッセージのエンティティボディとしてのセグメントの返信をもたらす。セグメントは、複数のピクチャのグループ(GoP)を含み得る。各GoPはさらに、複数のフラグメントを含んでよく、ここで、セグメントインデクシングは、各フラグメントに対する時間/バイトオフセット情報を提供し、すなわち、インデクシングの単位はフラグメントである。

10

#### 【0089】

フラグメントまたはフラグメントの部分は、スループットを向上させるために、並列のTCP接続を通じて要求され得る。これは、ボトルネックリンク上の接続を共有するとき、または、混雑が原因で接続が失われるときに生じる問題を軽減することができるので、全体的な速度および配信の信頼性が向上し、これは、コンテンツザッピング時間の速度および信頼性をかなり向上させることができる。帯域幅は、過剰な要求によるレイテンシと引き換えであり得るが、スタベーションの危険を高め得る、はるかに未来のものに対する要求を行うことを避けるように、注意が払われなければならない。

20

#### 【0090】

同じサーバ上でのセグメントに対する複数の要求は、反復的なTCP始動遅延を避けるためにパイプライン化され得る(現在の要求が完了する前に次の要求を行う)。連続的なフラグメントに対する要求は、1つの要求へと統合され得る。

#### 【0091】

30

一部のCDNは、大きなファイルを好み、範囲要求を最初に見たときに、元のサーバからのファイル全体のバックグラウンドフェッチを引き起こし得る。しかしながら、大半のCDNは、データが利用可能になるとキャッシュからの範囲要求にサービスする。したがって、クライアント要求の何らかの部分を、セグメントファイル全体のものにすることが有利であり得る。これらの要求は、必要であれば後で取り消され得る。

#### 【0092】

有効な切替点は、ターゲットストリーム中の探索点、たとえば、具体的にはRAPであり得る。固定されたGoP構造、または複数のストリームにわたるRAPの整列(メディアの開始に基づく、またはGoPに基づく)のような、様々な実装形態が可能である。

#### 【0093】

40

一実施形態では、セグメントおよびGoPは、異なるレートのストリームにまたがって揃えられ得る。この実施形態では、GoPは、可変のサイズであってよく、複数のフラグメントを含んでよいが、フラグメントは異なるレートのストリームの間で揃えられない。

#### 【0094】

いくつかの実施形態では、ファイル冗長性が有利に利用され得る。これらの実施形態では、データの冗長なバージョンを生成するために、抹消符号が各フラグメントに付加される。好ましくは、ソースのフォーマッティングは、FECの使用が原因で変更されず、FEC修復データを含む追加の修復セグメントが、たとえば、元の表現の従属の表現として、取込システムにおける追加のステップとして生成され利用可能にされる。そのフラグメントに対するソースデータのみを使用してフラグメントを再構築することが可能であるクライア

50

ントは、サーバからのセグメント内のフラグメントに対するソースデータのみを要求することができる。サーバが利用不可能である場合、またはサーバに対する接続が遅い場合、これらはソースデータに対する要求の前または後のいずれかに判定され得るが、追加の修復データが、修復セグメントからのフラグメントのために要求されてよく、このことは、フラグメントを復元するのに十分なデータを信頼性をもって配信するための時間を減らし、場合によってはFEC復号を使用して、受信されたソースと修復データの組合せを使用し、フラグメントのソースデータを復元する。さらに、フラグメントが切迫した状態になると、すなわち、その再生時間が差し迫ると、フラグメントの復元を可能にするために追加の修復データが要求されてよく、このことは、リンク上でのそのフラグメントに対するデータの占有率を上げるが、リンク上の他の接続を切断して帯域幅を解放するよりも効率的である。これはまた、並列接続の使用によるスタベーションの危険を軽減し得る。

10

#### 【0095】

フラグメントフォーマットは、リアルタイムトранSPORT制御プロトコルRTCPを通じて実現されたオーディオ/ビデオの同期を伴う、リアルタイムトランSPORTプロトコル(RTP)パケットの記憶されたストリームであり得る。

#### 【0096】

セグメントフォーマットはまた、MPEG-2 TSの内部タイミングを通じて実現されたオーディオ/ビデオの同期を伴う、MPEG-2 TSパケットの記憶されたストリームであり得る。

#### 【0097】

ストリーミングをより効率的にするためのシグナリングの使用および/またはブロックの作成

20

多数の特徴が、改善された性能を提供するために、ブロック要求ストリーミングシステムにおいて使用されることもまたは使用されないこともある。性能は、ストールすることなくプレゼンテーションを再生するための能力、帯域幅の制約の中でメディアデータを取得すること、および/または、クライアント、サーバ、および/または取込システムにおいて、限られたプロセッサリソース内でメディアデータを取得することと関連し得る。これらの特徴の一部が、ここで説明される。

#### 【0098】

##### セグメント内のインデクシング

ムービーフラグメントに対するpartial GET要求を編成するために、クライアントは、ファイルまたはセグメント内のフラグメントに含まれるすべてのメディアコンポーネントの復号時間またはプレゼンテーション時間におけるバイトオフセットおよび開始時間、ならびにまた、どのフラグメントがランダムアクセスポイントを開始するまたは含むか(および代替的な表現の切替点として適切であるか)を知らされてよく、この情報は、セグメントインデクシングまたはセグメントマップと呼ばれることが多い。復号時間またはプレゼンテーション時間における開始時間は、直接的に表現されてよく、または、基準時間に対するデルタとして表現されてよい。

30

#### 【0099】

この時間およびバイトオフセットのインデクシング情報は、ムービーフラグメントごとの少なくとも8バイトのデータを必要とし得る。ある例として、500msのムービーフラグメントを伴う、単一のファイルに含まれる2時間のムービーに対して、これは、全体で約112キロバイトのデータである。プレゼンテーションを開始するときにこのデータのすべてをダウンロードすることは、大きな追加の始動遅延をもたらし得る。しかしながら、時間およびバイトのオフセットデータは階層的に符号化され得るので、クライアントは、クライアントが開始することを望むプレゼンテーション中の地点に関連する、時間およびオフセットのデータの小さな塊を迅速に見つけることができる。情報はまた、セグメントインデックスのいくつかの改良版が、メディアデータとインターリープされて配置され得るように、セグメント内に分布し得る。

40

#### 【0100】

表現が時間的に複数のセグメントへとセグメント化される場合、各セグメントに対する

50

完全な時間およびオフセットのデータはすでに十分小さいことがあるので、この階層的コーディングの使用は必要ではないことがあることに留意されたい。たとえば、セグメントが上の例で2時間ではなく1分である場合、時間-バイトオフセットのインデクシング情報は1キロバイト程度のデータであり、これは通常、単一のTCP/IPパケットに収まり得る。

#### 【0101】

異なる選択肢は、フラグメントの時間およびバイトのオフセットデータを3GPPファイルに追加することが可能である。

#### 【0102】

まず、Movie Fragment Random Access Box(「MFRA」)が、この目的で使用され得る。MFRAは、ムービーフラグメントを使用してファイル中のランダムアクセスポイントを読者が発見するのを支援し得る、表を提供する。この機能を支援するように、MFRAはついでに、ランダムアクセスポイントを含むMFRAボックスのバイトオフセットを含む。MFRAは、ファイルの終わりに、またはその近くに配置され得るが、そうであるとは限らない。Movie Fragment Random Access Offset Boxに対するファイルの終わりからスキャンして、その中のサイズ情報を使用することによって、Movie Fragment Random Access Boxの始まりを位置特定することが可能になり得る。しかしながら、HTTPストリーミングの終わりにMFRAを配置することは、通常、所望のデータにアクセスするための少なくとも3~4個のHTTP要求、すなわちファイルの終わりからMFRAを要求するための少なくとも1つのHTTP要求と、MFRAを取得するための1つのHTTP要求と、ファイル中の所望のフラグメントを取得するための最後の1つのHTTP要求とを必要とする。したがって、mfraが単一の要求の中の最初のメディアデータと一緒にダウンロードされ得るので、始めに配置することが望ましいことがある。また、HTTPストリーミングのためにMFRAを使用するのは非効率であることがあり、それは、「MFRA」中の情報は時間およびmoof\_offsetを除いて必要とされず、長さの代わりにオフセットを規定することはより多くのビットを必要とし得るからである。

10

20

#### 【0103】

第2に、Item Location Box(「ILOC」)が使用され得る。「ILOC」は、このファイルまたは他のファイルの中のメタデータリソースのディレクトリを、メタデータリソースを含むファイル、そのファイル内でのメタデータリソースのオフセット、およびメタデータリソースの長さを特定することによって提供する。たとえば、システムは、すべての外部的に参照されるメタデータリソースを1つのファイルへと統合し、それに従って、ファイルオフセットおよびファイル参照を再調整することができる。しかしながら、「ILOC」は、メタデータの位置を与えることが意図されているので、ILOCが実際のメタデータと共に存することは難しいことがある。

30

#### 【0104】

最後の、そして場合によっては最も適切なものは、正確なフラグメントの時間または継続時間とバイトオフセットとを効率的に提供する目的に特に専用である、Time Index Box(「TIDX」)と呼ばれる新たなボックスの仕様である。これは、次のセクションでより詳細に説明される。同じ機能を有する代替的なボックスは、Segment Index Box(「SIDX」)であり得る。本明細書では、別段示されない限り、これら2つは相互に交換可能であってよく、それは、両方のボックスが、正確なフラグメントの時間または継続時間とバイトオフセットとを効率的に提供するための能力を提供するからである。TIDXとSIDXとの違いは以下で与えられる。両方のボックスがセグメントインデックスを実装するので、TIDXボックスとSIDXボックスとをどのように相互に交換するかは明らかであろう。

40

#### 【0105】

##### セグメントインデクシング

セグメントは、特定された開始時間および特定された数のバイトを有する。複数のフラグメントは、単一のセグメントへと連結されてよく、クライアントは、要求されたフラグメントまたはフラグメントのサブセットに対応するセグメント内の特定のバイト範囲を特定する要求を出すことができる。たとえば、HTTPが要求プロトコルとして使用される場合

50

、HTTP Rangeヘッダがこの目的で使用され得る。この手法は、クライアントが異なるフラグメントのセグメント内の位置を規定するセグメントの「セグメントインデックス」へのアクセス権を有することを必要とする。この「セグメントインデックス」は、メタデータの一部として提供され得る。この手法は、各々のブロックが別個のファイルに保存される手法と比較して、はるかに少数のファイルしか作成され管理される必要がないという結果をもたらす。非常に大量の(たとえば、1時間のプレゼンテーションに対して数千に及び得る)ファイルの作成、転送、および記憶の管理は、複雑でエラーを発生させやすいことがあるので、ファイル数の低減が利点となる。

#### 【0106】

クライアントがセグメントのより小さい部分の所望の開始時間しか知らない場合、クライアントは、ファイル全体を要求し、次いでファイル全体を読み取って、適切な再生開始位置を決定することがある。帯域幅の使用率を改善するために、セグメントは、メタデータとしてインデックスファイルを含むことがあり、インデックスファイルは、個々のブロックのバイト範囲をブロックが対応する時間範囲と対応付け、これは、セグメントインデクシングまたはセグメントマップと呼ばれる。このメタデータは、XMLデータとしてフォーマットされてよく、または、たとえば3GPPファイルフォーマットのアトムおよびボックス構造に従った、バイナリであってよい。インデクシングは単純であってよく、このとき各ブロックの時間およびバイトの範囲は、ファイルの開始に対して絶対的であり、または、インデクシングは階層的であってよく、このとき一部のブロックは親ブロックへとグループ化され(かつ親ブロックは祖父母ブロックへとグループ化される、など)、所与のブロックに対する時間およびバイトの範囲は、ブロックの親ブロックの時間および/またはバイトの範囲に対して表現される。10

#### 【0107】

##### 例示的なインデクシングマップ構造

一実施形態では、メディアストリームの1つの表現に対する元のソースデータは、「メディアセグメント」と本明細書で呼ばれる1つまたは複数のメディアファイルに含まれてよく、各メディアセグメントは、メディアの連続的な時間セグメント、たとえば、5分のメディア再生を再生するために使用されるメディアデータを含む。

#### 【0108】

図6は、メディアセグメントの例示的な全体的な構造を示す。各セグメント内で、始めにおいて、または、ソースセグメント全体に分散して、時間/バイトオフセットセグメントマップを含むインデクシング情報も存在し得る。一実施形態の時間/バイトオフセットセグメントマップは、時間/バイトオフセットのペア( $T(0), B(0)$ ), ( $T(1), B(1)$ ), ..., ( $T(i), B(i)$ ), ..., ( $T(n), B(n)$ )のリストであってよく、 $T(i-1)$ は、すべてのメディアセグメントの中で最初のメディアの開始時間に対する、メディアの*i*番目のフラグメントの再生のためのセグメント内の開始時間を表し、 $T(i)$ は、*i*番目のフラグメントの終了時間(および、したがって、次のフラグメントの開始時間)を表し、バイトオフセット $B(i-1)$ は、このソースセグメント内のデータの始めの対応するバイトインデックスであり、ここでメディアの*i*番目のフラグメントはソースセグメントの始めに対して開始し、 $B(i)$ は、*i*番目のフラグメントの対応する終了バイトインデックス(および、したがって、次のフラグメントの最初のバイトのインデックス)である。セグメントが複数のメディアコンポーネントを含む場合、 $T(i)$ および $B(i)$ は、絶対的な方式でセグメント中の各コンポーネントに対して与えられてよく、または、基準メディアコンポーネントをサービスする別のメディアコンポーネントに対して表現されてよい。30

#### 【0109】

この実施形態では、ソースセグメント中のフラグメントの数は*n*であり、*n*はセグメントごとに変化し得る。

#### 【0110】

別の実施形態では、各フラグメントに対するセグメントインデックスにおける時間オフセットは、最初のセグメントの絶対的な開始時間および各フラグメントの継続時間によっ4050

て決定され得る。この場合、セグメントインデックスは、最初のフラグメントの開始時間と、セグメントに含まれるすべてのフラグメントの継続時間とを記録することができる。セグメントインデックスはまた、フラグメントのサブセットのみを記録し得る。その場合、セグメントインデックスは、含んでいるセグメントの終わりにおいて、または次のサブセグメントの始めにおいて終了する、1つまたは複数の連続的なフラグメントとして定義されるサブセグメントの継続時間を記録する。

#### 【0111】

各フラグメントに対して、フラグメントが探索点、すなわち、その点より後のメディアがその点よりも前のいずれのメディアにも依存しない点で開始するかどうか、またはそれを含むかどうかを示す値も存在し得るので、そのフラグメントから先のメディアは、前のフラグメントとは独立に再生され得る。探索点は、一般に、再生がすべての前のメディアとは独立に開始できる、メディア中の点である。図6はまた、ソースセグメントに対する可能なセグメントインデクシングの単純な例を示す。その例では、時間オフセット値はミリ秒の単位であり、したがって、このソースセグメントの最初のフラグメントは、メディアの始めから20秒で開始し、最初のフラグメントは485ミリ秒という再生時間を有する。最初のフラグメントの始めのバイトオフセットは0であり、最初のフラグメントの終わり/2番目のフラグメントの始めのバイトオフセットは50,245であり、したがって、最初のフラグメントのサイズは50,245バイトである。フラグメントまたはサブセグメントがランダムアクセスポイントで開始しないが、ランダムアクセスポイントがフラグメントまたはサブセグメントに含まれる場合、開始時間と実際のRAP時間との復号時間差またはプレゼンテーション時間差が与えられ得る。これにより、このメディアセグメントへの切り替えの場合、表示からの切り替えが提示されなければならなくなるまでに、クライアントが時間を正確に知れることができることになる。

#### 【0112】

単純なまたは階層的なインデクシングに加えて、またはその代わりに、デイジーチェーンインデクシングおよび/またはハイブリッドインデクシングが使用され得る。

#### 【0113】

異なるトラックに対するサンプル期間は同じではないことがあるので(たとえば、ビデオサンプルは33ms表示され得るが、オーディオサンプルは80ms続き得る)、ムービーフラグメント中の異なるトラックは、厳密に同じ時間に開始し終了しないことがあり、すなわち、オーディオはビデオのわずかに前またはわずかに後に開始することがあり、それを埋め合わせるように、逆のことが先行するフラグメントについて成り立つ。曖昧さを避けるために、時間およびバイトオフセットデータにおいて規定されるタイムスタンプは、特定のトラックに対して規定されてよく、これは、各表現に対して同じトラックであり得る。通常、これはビデオトラックである。これにより、クライアントが表現を切り替えているときに、クライアントが次のビデオフレームを正確に特定することが可能になる。

#### 【0114】

トラックの時間軸とプレゼンテーション時間との厳密な関係を維持し、上記の問題にもかかわらずオーディオ/ビデオの同期の円滑な再生および維持を確実にするように、プレゼンテーションの間に注意が払われ得る。

#### 【0115】

図7は、単純なインデックス700および階層的インデックス702のようないくつかの例を示す。

#### 【0116】

セグメントマップを含むボックスの2つの具体的な例が以下で与えられ、1つはtime index box('TIDX')と呼ばれ、1つは('SIDX')と呼ばれる。この定義は、ISOベースのメディアファイルフォーマットによるボックス構造に従う。同様のシンタックスを定義するための、ならびに同じセマンティクスおよび機能を有するそのようなボックスの他の設計は、読者には明らかであろう。

#### 【0117】

10

20

30

40

50

## Time Index Box

定義

【0118】

ボックスタイプ:「tidx」

【0119】

コンテナ:ファイル

【0120】

必須性:なし

【0121】

量:0または1のいずれかの数

10

【0122】

Time Index Boxは、ファイルのいくつかの領域をプレゼンテーションのいくつかの時間間隔と関連付ける、時間およびバイトオフセットインデックスのセットを提供し得る。Time Index Boxは、参照されるデータのタイプを示す、targettypeフィールドを含み得る。たとえば、targettype「moof」を伴うTime Index Boxは、時間とバイトの両方のオフセットに関して、ファイルに含まれるメディアフラグメントに対するインデックスを提供する。Time Index Boxというtargettypeを伴うTime Index Boxは、階層的時間インデックスを構築するために使用されてよく、ファイルのユーザが、インデックスの要求される部分へと迅速に進むことを可能にする。

【0123】

20

セグメントインデックスは、たとえば、次のシンタックスを含み得る。

【0124】

```
aligned(8) class TimeIndexBox extends FullBox('frai'){
    unsigned int(32) targettype;
```

【0125】

30

```
    unsigned int(32) time_reference_track_ID;
    unsigned int(32) number_of_elements;
    unsigned int(64) first_element_offset;
    unsigned int(64) first_element_time;
```

```
    for(i=1;i<=number_of_elements;i++)
    {
```

```
        bit(1) random_access_flag;
```

```
        unsigned int(31) length;
```

```
        unsigned int(32) deltaT;
```

```
}
```

```
}
```

【0126】

セマンティクス

【0127】

targettype:このTime Index Boxによって参照されるボックスデータのタイプである。これは、Movie Fragment Header(「moof」)またはTime Index Box(「tidx」)のいずれかであり得る。

40

【0128】

time-reference\_track\_id:このインデックスにおける時間オフセットが規定される対象のトラックを示す。

【0129】

number\_of\_elements:このTime Index Boxによってインデックスが付けられる要素の数。

【0130】

first\_element\_offset:最初のインデックスが付けられた要素のファイルの始めからの

50

バイトオフセット。

【0131】

first\_element\_time:time\_reference\_track\_idによって特定されたトラックのMedia Headerボックス中で規定される時間軸を使用した、最初のインデックスが付けられた要素の開始時間。

【0132】

random\_access\_flag:要素の開始時間がランダムアクセスポイントである場合は1である。それ以外の場合は0である。

【0133】

length:インデックスが付けられた要素のバイト単位での長さ。

10

【0134】

deltaT:この要素の開始時間と次の要素の開始時間との間の、time\_reference\_track\_idによって特定されるトラックのMedia Headerボックス中で規定される時間軸での差分。

【0135】

Segment Index Box

【0136】

Segment Index Box(「sidx」)は、ムービーフラグメントのコンパクトなインデックスと、セグメント中の他のSegment Index Boxとを提供する。Segment Index Box中には2つのループ構造がある。最初のループは、サブセグメントの最初のサンプル、すなわち、2番目のループによって参照される最初のムービーフラグメント中のサンプルを記録する。2番目のループは、サブセグメントのインデックスを提供する。「sidx」ボックスのコンテナは、直接、ファイルまたはセグメントである。

20

【0137】

シンタックス

【0138】

```
aligned(8) class SegmentIndexBox extends FullBox('sidx',version,0){
```

【0139】

```
    unsigned int(32) reference_track_ID;
```

【0140】

```
    unsigned int(16) track_count;
```

30

【0141】

```
    unsigned int(16) reference_count;
```

【0142】

```
    for(i=1; i<= track_count; i++)
```

【0143】

```
{
```

【0144】

```
    unsigned int(32) track_ID;
```

【0145】

```
    if(version==0)
```

40

【0146】

```
{
```

【0147】

```
    unsigned int(32) decoding_time;
```

【0148】

```
}else
```

【0149】

```
{
```

【0150】

```
    unsigned int(64) decoding_time;
```

50

```

【0 1 5 1】
}
【0 1 5 2】
}
【0 1 5 3】
for(i=1;i<=reference_count;i++)
【0 1 5 4】
{
【0 1 5 5】
bit (1)      reference_type;          10
【0 1 5 6】
unsigned int(31)      reference_offset;
【0 1 5 7】
unsigned int(32)      subsegment_duration;
【0 1 5 8】
bit(1)      contains_RAP;
【0 1 5 9】
unsigned int(31)      RAP_delta_time;
【0 1 6 0】
}
【0 1 6 1】
}
【0 1 6 2】
セマンティクス
【0 1 6 3】
reference_track_IDは、参照トラックのtrack_IDを与える。
【0 1 6 4】
track_count:次のループにおいてインデックスが付けられるトラックの数(1以上)。
【0 1 6 5】
reference_count:2番目のループにおいてインデックスが付けられる要素の数(1以上).      30
【0 1 6 6】
track_ID:トラックフラグメントがこのインデックスによって特定される最初のムービー
一フラグメントに含まれる、トラックのIDであり、このループ中のちょうど1つのtrack_I
Dがreference_track_IDに等しい。
【0 1 6 7】
decoding_time:(トラックのMedia Header Boxのtimescaleフィールドにおいて記録さ
れるような)トラックの時間軸で表現される、2番目のループ中の最初のアイテムによ
って参照されるムービーフラグメント中のtrack_IDによって特定されるトラック中の最初のサ
ンプルの復号時間。
【0 1 6 8】
reference_type:0に設定される場合、参照がムービーフラグメント('moof')ボックス
に対するものであることを示し、1に設定される場合、参照がセグメントインデックス('
sidx')ボックスに対するものであることを示す。          40
【0 1 6 9】
reference_offset:含んでいるSegment Index Boxの後の最初のバイトから、参照され
るボックスの最初のバイトまでの、バイト単位の距離。
【0 1 7 0】
subsegment_duration:参照がSegment Index Boxに対するものである場合、このフィ
ールドは、そのボックスの2番目のループ中のsubsegment_durationフィールドの合計を担
持し、参照がムービーフラグメントに対するものである場合、このフィールドは、示され      50

```

るムービーフラグメント、および、ループ中の次のエントリーによって記録される最初のムービーフラグメントまたはサブセグメントの終わりのいずれか早い方までの後続のムービーフラグメントにおける、参照トラック中のサンプルのサンプル継続時間の合計を担持し、(トラックのMedia Header Boxのtimescaleフィールドにおいて記録されるように)継続時間がトラックの時間軸で表現される。

#### 【0171】

`contains_RAP`: 参照がムービーフラグメントに対するものであるとき、`reference_track_ID`に等しい`track_ID`を伴うトラックのそのムービーフラグメント内のトラックフラグメントが少なくとも1つのランダムアクセスポイントを含む場合は、このビットは1であってよく、それ以外の場合はこのビットは0に設定される。参考がセグメントインデックスに対するものであるとき、そのセグメントインデックス中の参照のいずれかがこのビットを1へと設定させる場合のみ、このビットは1に設定され、それ以外の場合は0である。10

#### 【0172】

`RAP_data_time`: `contains_RAP`が1である場合、ランダムアクセスポイント(RAP)のプレゼンテーション(複合)時間を与え、`contains_RAP`が0である場合、値0で保留される。時間は、このエントリーによって記録されるサブセグメントの最初のサンプルの復号時間と、`reference_track_ID`に等しい`track_ID`を伴うトラック中のランダムアクセスポイントのプレゼンテーション(複合)時間との差分として表される。

#### 【0173】

TIDXとSIDXの違い20

TIDXおよびSIDXは、インデクシングに関して同じ機能を提供する。SIDXの最初のループは、最初のムービーフラグメントのグローバルなタイミングを追加で提供するが、グローバルなタイミングは、ムービーフラグメント自体にも、絶対的に、または基準トラックに対して相対的に含まれ得る。

#### 【0174】

SIDXの2番目のループは、TIDXの機能を実装する。具体的には、SIDXは、`reference_type`によって参照される各インデックスに対する参照の対象の混合物を有することを可能にするが、TIDXは、TIDXのみまたはMOOFのみを参照するだけである。TIDXの`number_of_elements`は、SIDXの`reference_count`に対応し、TIDXの`time-reference_track_id`はSIDXの`reference_track_ID`に対応し、TIDXの`first_element_offset`は2番目のループの最初のエントリーの`reference_offset`に対応し、TIDXの`first_element_time`は最初のループの`reference_track`の`decoding_time`に対応し、TIDXの`random_access_flag`は、SIDXでは必ずしもRAPがフラグメントの始めに配置されなくてもよいので`RAP_delta_time`を必要とするという追加の自由度を伴って、SIDXの`contains_RAP`に対応し、TIDXの`length`はSIDXの`reference_offset`に対応し、かつ最後に、TIDXの`deltaT`はSIDXの`subsegment_duration`に対応する。したがって、2つのボックスの機能は等価である。30

#### 【0175】

可変のブロックサイジングおよびサブGoPブロック

ビデオメディアにとって、ビデオ符号化構造と要求のブロック構造との関係は重要であり得る。たとえば、ランダムアクセスポイント(「RAP」)のような探索点で各ブロックが開始し、かつ各ブロックは、等しい期間のビデオ時間を表す場合、ビデオメディア中の少なくともいくつかの探索点の配置は固定され、かつ探索点はビデオ符号化物内で一定の間隔で発生する。ビデオ符号化の当業者によく知られているように、探索点がビデオフレーム間の関係に従って配置されると、特に、前のフレームとの共通点がほとんどないフレームに配置されると、圧縮効率が改善され得る。ブロックが等しい量の時間を表すというこの要件は、したがって、圧縮が準最適となり得るように、ビデオ符号化に対する制約を課す。

#### 【0176】

探索点が固定された位置にあることを要求するのではなく、ビデオプレゼンテーション内の探索点の配置がビデオ符号化システムによって選ばれることを可能にするのが望まし4050

い。ビデオ符号化システムが探索点を選ぶのを可能にすることで、ビデオ圧縮が改善されるので、所与の利用可能な帯域幅を使用してより高品質のビデオメディアを提供することができ、ユーザ体験の改善をもたらす。現在のブロック要求ストリーミングシステムは、すべてのブロックが同じ継続時間(ビデオ時間に関して)であること、および各ブロックが探索点で開始しなければならないことを要求し得るので、これは既存のシステムの欠点である。

#### 【0177】

上記のものに対する利点を提供する新規のブロック要求ストリーミングシステムが、ここで説明される。一実施形態では、ビデオコンポーネントの第1のバージョンのビデオ符号化プロセスは、圧縮効率を最適化するために、探索点の配置を選ぶように構成され得るが、探索点と探索点の間の長さに対する最大値があるという要件を伴う。この後者の要件は、符号化プロセスによる探索点の選択を制約するので、圧縮効率を下げる。しかしながら、この圧縮効率の低下は、探索点と探索点の間の長さに対する最大値が小さすぎなければ(たとえば、約1秒より長い)、通常の固定された位置が探索点に対して要求される場合に引き起こされるものよりも小さい。さらに、探索点と探索点の間の長さに対する最大値が数秒である場合、探索点の配置が完全に自由である場合と比較した圧縮効率の低下は、一般に非常に小さい。

#### 【0178】

この実施形態を含む多くの実施形態では、一部のRAPが探索点ではない、すなわち、探索点として選ばれない、2つの連続する探索点の間にあるRAPであるフレームが存在し得る、ということがあり得るが、それは、たとえば、RAPが周囲の探索点に時間的に近すぎるから、または、RAPの前または後の探索点のRAPとの間のメディアデータの量が小さすぎるからである。

#### 【0179】

メディアプレゼンテーションのすべての他のバージョン内の探索点の配置は、最初の(たとえば、最高のメディアデータレートの)バージョンの探索点と同じになるように制約され得る。これは、エンコーダに探索点の自由な選択を許容する場合と比較して、これらの他のバージョンに対する圧縮効率を下げる。

#### 【0180】

探索点の使用は通常、フレームが独立に復号可能であることを必要とし、これは一般に、そのフレームの圧縮効率を低くする。独立に復号可能であることを要求されないフレームは、他のフレーム中のデータに関して符号化されてよく、これは一般に、符号化されるべきフレームと基準フレームとの共通性の量に応じた量だけ、そのフレームの圧縮効率を上げる。探索点の配置の効率的な選択は、前のフレームとの共通性が低いフレームを、探索点フレームとして優先的に選ぶので、独立に復号可能な方法でフレームを符号化することによって引き起こされる圧縮効率の低下を最小限にする。

#### 【0181】

しかしながら、あるフレームと可能性のある基準フレームとの共通性のレベルは、コンテンツの異なる表現と高く相關付けられ、それは、元のコンテンツが同一であるからである。結果として、第1の変形における探索点と同じ位置にあるように、他の変形における探索点を制約することは、圧縮効率に大きな差を生まない。

#### 【0182】

探索点の構造は好ましくは、ブロック構造を決定するために使用される。好ましくは、各探索点はブロックの始まりを決定し、2つの連続する探索点の間のデータを包含する1つまたは複数のブロックが存在し得る。良好な圧縮を伴う符号化のために、探索点と探索点の間の長さは固定されないので、すべてのブロックが同じ再生期間を有することは要求されない。いくつかの実施形態では、ブロックは、コンテンツのバージョンの間で揃えられ、すなわち、コンテンツの1つのバージョンにおいてフレームの特定のグループにまたがるブロックがある場合、コンテンツの別のバージョンにおいてフレームの同じグループにまたがるブロックがある。コンテンツの所与のバージョンに対するブロックは重複せず、

10

20

30

40

50

コンテンツの各々のフレームは各バージョンのちょうど1つのブロックの中に含まれる。

#### 【0183】

探索点と探索点の間の可変の長さ、およびしたがって可変の長さのGoPの効率的な使用を可能にする可能化機構は、セグメントに含まれ得る、または他の手段によってクライアントに提供され得る、セグメントインデクシングまたはセグメントマップであり、すなわち、これは、プレゼンテーションの各ブロックの開始時間および継続時間を含む、提供され得るこのプレゼンテーション中のこのセグメントと関連付けられるメタデータである。クライアントは、プレゼンテーションがセグメント内にある特定の点で開始することをユーザが要求した場合、プレゼンテーションが開始すべきブロックを決定するときにこのセグメントインデクシングデータを使用することができる。そのようなメタデータが提供されない場合、プレゼンテーションは、コンテンツの始めにおいてのみ開始することができ、または、ランダムに、もしくは、所望の点に近い近似的な点において(たとえば、要求された開始点(時間的な)を平均のブロック期間で割って開始ブロックのインデックスを与えることによって、開始ブロックを選ぶことによって)開始することができる。10

#### 【0184】

一実施形態では、各ブロックは、別個のファイルとして提供され得る。別の実施形態では、複数の連続的なブロックが単一のファイルへと統合されてセグメントを形成することができる。この第2の実施形態では、各ブロックの開始時間および継続時間と、ブロックが開始するファイル内のバイトオフセットとを含む、各バージョンのメタデータが提供され得る。このメタデータは、初期プロトコル要求に応答して提供されてよく、すなわち、セグメントまたはファイルとは別々に入手可能であってよく、または、たとえばファイルの始まりにおいて、同じファイルまたはセグメント内にブロック自体として含まれ得る。当業者には明らかなように、このメタデータは、メタデータをクライアントにトランスポートするのに必要とされるネットワークリソースを減らすために、gzipまたはデルタ符号化などの圧縮された形式またはバイナリ形式で符号化されてよい。20

#### 【0185】

図6は、ブロックが可変サイズであるセグメントインデクシングの例を示し、ブロックの範囲は、部分的なGoP、すなわち、1つのRAPと次のRAPとの間の部分的な量のメディアデータである。この例では、探索点は、RAPインジケータによって示され、1というRAPインジケータ値は、ブロックがRAPまたは探索点で開始すること、またはそれを含むことを示し、0というRAPインジケータは、ブロックがRAPまたは探索点を含まないことを示す。この例では、最初の3個のブロック、すなわち、バイト0~157,033は第1のGoPを含み、これは、プレゼンテーション継続時間が1.623秒であり、プレゼンテーション時間はコンテンツの中で20秒から21.623秒まで続く。この例では、最初の3個のブロックのうちの最初のブロックは、0.485秒というプレゼンテーション時間を含み、セグメント中のメディアデータの最初の50,245バイトを含む。この例では、ブロック4、5、および6は第2のGoPを含み、ブロック7および8は第3のGoPを含み、ブロック9、10、および11は第4のGoPを含む。探索点として指定されないメディアデータ中に他のRAPが存在があるので、セグメントマップ中でRAPとしてシグナリングされないように留意されたい。30

#### 【0186】

図6を再び参照すると、クライアントまたは受信機がメディアプレゼンテーションの中で約22秒の時間オフセットで開始するコンテンツにアクセスすることを望む場合、クライアントは、後でより詳細に説明されるMPDのような、他の情報をまず使用して、関連するメディアデータがこのセグメント内にあるとまず判定することができる。クライアントは、セグメントの第1の部分をダウンロードして、セグメントインデクシングを取得することができ、セグメントインデクシングはこの場合、たとえばHTTPバイト範囲要求を使用すると、わずか数バイトである。セグメントインデクシングを使用して、クライアントは、クライアントがダウンロードすべき最初のブロックが、最大でも22秒の時間オフセットを伴いRAPで開始する、すなわち探索点である最初のブロックであると判定することができる。この例では、ブロック5が22秒よりも小さい時間オフセットを有し、すなわち、ブロ40  
50

ック5の時間オフセットは21.965秒であるが、セグメントインデクシングは、ブロック5がRAPで開始しないことを示すので、代わりに、セグメントインデクシングに基づいて、クライアントは、ブロック4をダウンロードすることを選択する。それは、ブロック4の開始時間が最大でも22秒であり、すなわち、ブロック4の時間オフセットが21.623秒でありRAPで開始するからである。したがって、セグメントインデクシングに基づいて、クライアントは、バイトオフセット157,034において開始するHTTP範囲要求を行う。

#### 【0187】

セグメントインデクシングが利用可能ではなければ、クライアントはこのデータをダウンロードする前にすべての前の157,034バイトのデータをダウンロードしなければならなかつた可能性があり、はるかに長い始動時間またはチャネルザッピング時間、および有用ではないデータの無駄なダウンロードにつながる。あるいは、セグメントインデクシングが利用可能ではなければ、クライアントは、所望のデータがセグメント内で開始する箇所を概算することができるが、概算は不正確であることがある、クライアントは適切な時間を逃して、そして手戻りが必要になることがある、これはやはり始動遅延を増やす。

10

#### 【0188】

一般に、各ブロックは、前のブロックと一緒にメディアプレーヤーによって再生され得る、メディアデータの部分を包含する。したがって、セグメント内に含まれるかまたは他の手段を通じてクライアントに提供されるかのいずれかである、ブロッキング構造および、セグメントインデクシングブロッキング構造のクライアントへのシグナリングは、高速なチャネルザッピングと、ネットワークの変動および途絶に直面したときのシームレスな再生とを行うためのクライアントの能力を大きく改善することができる。セグメントインデクシングによって可能にされるような、可変の長さのブロックおよびGoPの部分のみを包含するブロックのサポートは、ストリーミング体験を大きく改善することができる。たとえば、図6と、プレゼンテーションの中の約22秒においてクライアントが再生を開始することを望む上の例とを再び参照すると、クライアントは、1つまたは複数の要求を通じて、ブロック4内のデータを要求し、次いでこれを、再生の開始が可能になるとすぐにメディアプレーヤーに与えることができる。したがって、この例では、再生は、ブロック4の42,011バイトがクライアントにおいて受信されるとすぐに開始し、したがって、高速なチャネルザッピング時間を可能にする。代わりに、クライアントが、再生が開始しようとする前にGoP全体を要求することが必要であったとすると、これは144,211バイトのデータなので、チャネルザッピング時間はより長くなつたであろう。

20

#### 【0189】

他の実施形態では、RAPまたは探索点はまた、ブロックの中央部に存在することがあり、そのRAPまたは探索点がブロックまたはフラグメント内のどこにあるかを示す、セグメントインデクシング中のデータがあり得る。他の実施形態では、時間オフセットは、ブロック内の最初のフレームのプレゼンテーション時間の代わりに、ブロック内の最初のフレームの復号時間であり得る。

30

#### 【0190】

図8(a)および図8(b)は、複数のバージョンまたは表現にわたって揃えられた探索点構造の可変ブロックサイジングの例を示し、図8(a)は、複数のバージョンのメディアストリームにわたって揃えられた探索点を伴う可変ブロックサイジングを示し、一方図8(b)は、複数のバージョンのメディアストリームにわたって揃えられない探索点を伴う可変ブロックサイジングを示す。

40

#### 【0191】

時間が秒単位で上部に示されており、2つの表現に対する2つのセグメントのブロックおよび探索点が、この時間軸に対するそれらのタイミングに関して、左から右へと示されるので、示される各ブロックの長さは、再生時間に比例し、ブロック中のバイトの数には比例しない。この例では、2つの表現の両方のセグメントに対するセグメントインデクシングは、探索点に対して同じ時間オフセットを有するが、場合によっては、探索点のと探索点の間に異なる数のブロックまたはフラグメントを有し、各ブロック中のメディアデータ

50

の異なる量が原因で、ブロックに対する異なるバイトオフセットを有する。この例では、クライアントが約23秒のプレゼンテーション時間で表現1から表現2に切り替えることを望む場合、クライアントは、表現1に対するセグメント中のブロック1.2まで要求し、ブロック2.2で開始する表現2に対するセグメントの要求を開始することができるので、表現1の中の探索点1.2と同時期のプレゼンテーションで切り替えが発生し、これは、表現2の中の探索点2.2と同時である。

#### 【0192】

前述のことから明らかなように、説明されるブロック要求ストリームシステムは、コンテンツ内の特定の位置に探索点を配置するようにビデオ符号化を制約せず、このことは、既存のシステムの問題の1つを軽減する。

10

#### 【0193】

上で説明された実施形態では、同じコンテンツプレゼンテーションの様々な表現に対する探索点が揃えられるように編成される。しかしながら、多くの場合、この整列の要件を緩和することが好ましい。たとえば、探索点が揃えられた表現を生成する能力を有さない符号化ツールが、表現を生成するために使用されていることがある。別の例として、コンテンツプレゼンテーションは、異なる表現の間の探索点の整列を伴わずに、異なる表現へと独立に符号化され得る。別の例として、表現は、レートがより低いと、かつ切り替えられなければならないことが頻繁であると、または、早送りまたは巻き戻しまたは高速探索のようなトリックモードをサポートするための探索点をより頻繁に含むと、より多くの探索点を含み得る。したがって、コンテンツプレゼンテーションに対する様々な表現にわたって揃えられていない探索点の効率的およびシームレスな処理をブロック要求ストリーミングシステムが行うことを可能にする方法を提供することが望ましい。

20

#### 【0194】

この実施形態では、複数の表現にわたる探索点の配置は揃わなくてよい。ブロックは、新たなブロックが各探索点で開始するように構築されるので、表現の異なるバージョンのブロック間の整列はなくてよい。異なる表現の間のそのような揃えられない探索点構造の例が、図8(b)に示される。時間が秒単位で上部に示されており、2つの表現に対する2つのセグメントのブロックおよび探索点が、この時間軸に対するそれらのタイミングに関して、左から右へと示されるので、示される各ブロックの長さは、再生時間に比例し、ブロック中のバイトの数には比例しない。この例では、2つの表現の両方のセグメントに対するセグメントインデクシングは、場合によっては、探索点に対して異なる時間オフセットを有し、また、場合によっては、探索点と探索点の間に異なる数のブロックまたはフラグメントを有し、各ブロック中のメディアデータの異なる量が原因で、ブロックに対する異なるバイトオフセットを有する。この例では、約25秒のプレゼンテーション時間においてクライアントが表現1から表現2に切り替えることを望む場合、クライアントは、表現1に対するセグメント中のブロック1.3まで要求し、ブロック2.3で開始する表現2に対するセグメントの要求を開始することができるので、表現1の中のブロック1.3の再生の最中にある、表現2の中の探索点2.3と同時期のプレゼンテーションにおいて切り替えが発生し、したがって、ブロック1.2に対するメディアの一部は再生されない(しかし、再生されないブロック1.3のフレームに対するメディアデータは、再生されるブロック1.3の他のフレームを復号するための受信機バッファへとロードされなければならないことがある)。

30

#### 【0195】

この実施形態では、以前に選択されたバージョンとは異なる表現からのブロックを選択することを要求されたときは常に、その最初のフレームが、最後に選択されたブロックの最後のフレームの後続のフレームよりも遅くない、最遅のブロックが選ばれるように、ブロック選択器123の動作が修正され得る。

40

#### 【0196】

この最後に説明された実施形態は、最初のバージョン以外のバージョン内へと探索点の配置を制約するという要件を取り除くことができるので、これらのバージョンに対する圧縮効率が上がり、所与の利用可能な帯域幅に対するより高品質なプレゼンテーションと、

50

改善されたユーザ体験とをもたらす。さらなる考慮事項は、コンテンツの複数の符号化物(バージョン)にわたる探索点の整列という機能を実行するビデオ符号化ツールが、広く利用可能ではないことがあるということであり、したがって、この最後に説明された実施形態の利点は、現在利用可能なビデオ符号化ツールが使用され得るというものである。別の利点は、異なるバージョンのコンテンツの符号化が、異なるバージョンに対する符号化プロセス間の調整を何ら必要とすることなく、並列に進行し得るということである。別の利点は、追加のバージョンのコンテンツが、特定の探索点の位置のリストを符号化ツールに提供する必要なく、より後の時点で符号化されプレゼンテーションに追加され得るということである。

#### 【0197】

10

一般に、ピクチャがピクチャのグループ(GoP)として符号化される場合、シーケンス中の最初のピクチャは探索点であり得るが、常にそうである必要はない。

#### 【0198】

##### 最適なブロック区分

ブロック要求ストリーミングシステムにおける関心事である1つの問題は、符号化されたメディア、たとえばビデオメディアの構造と、ブロック要求に使用されるブロック構造との相互作用である。ビデオ符号化の当業者によく知られているように、各ビデオフレームの符号化された表現に対して必要とされるビットの数が、場合によってはかなり、フレームごとに変動することが多い。結果として、受信されたデータの量と、そのデータによって符号化されるメディアの継続時間との関係は、単純ではないことがある。さらに、ブロック要求ストリーミングシステム内のブロックへのメディアデータの分割は、複雑さの次元をさらに大きくする。具体的には、いくつかのシステムでは、ブロックのメディアデータは、ブロック全体が受信されるまで再生されないことがあり、たとえば、ブロック内のメディアデータの構成、または抹消符号を使用するブロック内のメディアサンプル間の依存関係が、この性質をもたらし得る。ブロックサイズとブロック継続時間とのこれらの複雑な相互作用、および、再生を開始する前にブロック全体を受信する潜在的な必要性の結果として、再生が開始する前にメディアデータがバッファリングされる、保守的な手法をクライアントシステムが採用することが一般的である。そのようなバッファリングは、長いチャネルザッピング時間、したがって悪いユーザ体験をもたらす。

20

#### 【0199】

30

Pakzadは、データストリームの背後にある構造に基づいて連続的なブロックへとデータストリームをどのように区分するかを決定するための、新しく効率的な方法である「ブロック区分方法」について説明しており、さらに、ストリーミングシステムの状況において、これらの方針のいくつかの利点について説明している。Pakzadのブロック区分方法をブロック要求ストリーミングシステムに適用するための、本発明のさらなる実施形態がここで説明される。この方法は、メディアデータの任意の所与の要素(たとえば、ビデオフレームまたはオーディオサンプル)の再生時間が、任意の隣接するメディアデータ要素の再生時間とは与えられた閾値未満の分だけ異なるように、概略的なプレゼンテーション時間の順序で提示されるようにメディアデータを並べるステップを含み得る。そのように順序付けられたメディアデータは、Pakzadの言葉遣いにおけるデータストリームであると見なされてよく、このデータストリームに適用されるPakzadの方法のいずれも、データストリームによってブロック境界を特定する。隣接するブロック境界の任意のペアの間のデータは、本開示の言葉遣いでは「ブロック」とあると見なされ、本開示の方法は、ブロック要求ストリーミングシステム内のメディアデータのプレゼンテーションを提供するために適用される。本開示を読んだ当業者に明らかのように、Pakzadにおいて開示される方法のいくつかの利点が次いで、ブロック要求ストリーミングシステムの状況で実現され得る。

40

#### 【0200】

Pakzadにおいて説明されるように、部分的なGoPまたはGoPよりも多くの部分を包含するブロックを含む、セグメントのブロック構造の決定は、高速なチャネルザッピング時間を可能にするためのクライアントの能力に影響を与える。Pakzadでは、目標の始動時間が

50

与えられた場合に、クライアントが任意の探索点における表現のダウンロードを開始し、目標の始動時間が経過した後に再生を開始すると、時間的な各点において、クライアントがダウンロードしたデータの量が、少なくとも目標のダウンロードレートとダウンロードの開始から経過した時間との積である限り、再生がシームレスに継続することを確実にする、ロック構造と目標のダウンロードレートとを提供する。クライアントが目標の始動時間および目標のダウンロードレートへのアクセス権を有することが有利であり、それは、このことが、時間的に最早の点でいつ表現の再生を開始するかを決定するための手段をクライアントに与え、上で説明された条件をダウンロードが満たす限り表現の再生をクライアントが続けることを可能にするからである。したがって、後で説明される方法は、目標の始動時間および目標のダウンロードレートをMedia Presentation Descriptionに含めるための手段を提供するので、上で説明された目的のために使用され得る。

10

#### 【0201】

##### メディアプレゼンテーションデータのモデル

図5は、セグメントおよびmedia presentation description(「MPD」)ファイル、ならびに、MPDファイル内のセグメントの内訳、タイミング、および他の構造を含む、図1に示されるコンテンツ記憶装置のあり得る構造を示す。MPD構造またはファイルの可能な実装形態の詳細が、ここで説明される。多くの例において、MPDはファイルとして説明されるが、非ファイル構造も使用され得る。

#### 【0202】

図に示されるように、コンテンツ記憶装置110は、複数のソースセグメント510、MPD 500、および修復セグメント512を保持する。MPDは、期間記録501を含んでよく、期間記録501が今度は、初期化セグメント504およびメディアセグメント505への参照のようなセグメント情報503を含む、表現記録502を含んでよい。

20

#### 【0203】

図9(a)は例示的なメタデータテーブル900を示し、一方、図9(b)はHTTPストリーミングクライアント902がHTTPストリーミングサーバ906への接続を通じてどのようにメタデータテーブル900およびメディアブロック904を取得するかの例を示す。

#### 【0204】

本明細書で説明される方法では、クライアントが利用可能なメディアプレゼンテーションの表現に関する情報を含む、「Media Presentation Description」が提供される。表現は、クライアントが異なる代替物の中から1つを選択するという意味で代替物であることがある、または、表現は、クライアントが、各々が場合によっては代替物のセットからのものである、表現のいくつかを選択し、それらと一緒に提示するという意味で相補的であることがある。表現は、有利にはグループへと割り当てられてよく、クライアントは、1つのグループ中の表現について、それらの表現が各々互いに代替物であることを理解するようにプログラムまたは構成され、一方、異なるグループからの表現は、2つ以上の表現が一緒に提示されるようにされる。言い換えると、グループ中の2つ以上の表現がある場合、クライアントはそのグループから1つの表現を選び、次のグループから1つの表現を選ぶなどして、1つの表現を形成する。

30

#### 【0205】

表現を記述する情報は、有利には、表現、ビデオフレームレート、ビデオ解像度、およびデータレートを復号するために必要とされるコーデックのプロファイルおよびレベルを含む、適用されたメディアコーデックの詳細情報を含み得る。Media Presentation Descriptionを受信するクライアントは、この情報を使用して、表現が復号または提示に適しているかどうかを事前に判定することができる。区別する情報が表現のバイナリデータのみに含まれる場合、すべての表現からのバイナリデータを要求し、適格性についての情報を発見するために関連する情報を解析し抽出することが必要であるので、上記のことは利点となる。これらの複数の要求およびデータの解析および抽出には、ある時間がかかることがあります、これは、長い始動時間を、したがって悪いユーザ体験をもたらす。

40

#### 【0206】

50

加えて、Media Presentation Descriptionは、時刻に基づいてクライアント要求を制約する情報を含み得る。たとえば、ライブサービスに対しては、クライアントは、「現在のブロードキャスト時間」に近いプレゼンテーションの要求する部分へと制約され得る。ライブブロードキャストでは、現在のブロードキャスト時間よりも前に、与えられた閾値を超えてブロードキャストされた、コンテンツに対するサービングインフラストラクチャからのデータを除去することが望ましいことがあるので、上記のことは利点となる。これは、サービングインフラストラクチャ内での記憶リソースの再使用のために望ましいことがある。これはまた、提供されるサービスのタイプに応じて望ましいことがあり、たとえば、いくつかの場合には、プレゼンテーションは、受信するクライアントデバイスの何らかの加入モデルが原因で、ライブのみで利用可能にされることがあり、一方、他のメディアプレゼンテーションは、ライブおよびオンデマンドで利用可能にされてよく、他のプレゼンテーションは、第1のクラスのクライアントデバイスに対してはライブのみで、第2のクラスのクライアントデバイスに対してはオンデマンドのみで、第3のクラスのクライアントデバイスに対してはライブまたはオンデマンドのいずれかの組合せで利用可能にされ得る。メディアプレゼンテーションデータのモデル(以下の)で説明される方法は、クライアントが、サービングインフラストラクチャにおいて利用可能ではないことがあるデータについて、ユーザに対する要求を行いかつオファーを調整することを避けられるように、上記のような方針をクライアントが知らされることを可能にする。ある代替形態として、たとえば、クライアントは、このデータが利用可能ではないという通知をユーザに提示することができる。

10

20

#### 【0207】

本発明のさらなる実施形態では、メディアセグメントは、ISO/IEC 14496-12に記載されるISOベースのメディアファイルフォーマットに、または、派生した仕様(3GPP Technical Specification 26.244に記載される3GPファイルフォーマットなど)に準拠し得る。(上記の)Usage of 3GPP File Formatセクションは、ブロック要求ストリーミングシステム内のこのファイルフォーマットのデータ構造の効率的な使用を可能にする、ISOベースのメディアファイルフォーマットに対する新規の改善を記載する。この参照文献に記載されるように、メディアプレゼンテーションの時間セグメントとファイル内のバイト範囲との間の高速および効率的な対応付けを可能にする情報が、ファイル内で与えられ得る。メディアデータ自体が、ISO/IEC14496-12において定義されるムービーフラグメントの構築に従って構造化され得る。時間およびバイトのオフセットを提供するこの情報は、階層的に、または情報の单一のブロックとして構造化され得る。この情報は、ファイルの始めに与えられ得る。Usage of 3GPP File Formatセクションに記載されるような効率的な符号化を使用してこの情報を準備することは、ブロック要求ストリーミングシステムによって使用されるファイルダウンロードプロトコルがHTTPである場合に、たとえばHTTP partial GET要求を使用して、クライアントが上記の情報を迅速に取り出すことを可能にして、これは、始動、探索、またはストリーム切替の時間を短くするので、改善されたユーザ体験をもたらす。

30

#### 【0208】

メディアプレゼンテーション中の表現は、通常は代替物である複数の表現にわたるシームレスな切り替えを確実にして、2つ以上の表現の同期したプレゼンテーションを確実にするように、グローバルな時間軸で同期される。したがって、適応HTTPストリーミングメディアプレゼンテーション内の表現に含まれるメディアのサンプルのタイミングは、複数のセグメントにわたる連続的なグローバルな時間軸に関連し得る。

40

#### 【0209】

複数のタイプのメディア、たとえば、オーディオおよびビデオを含む、符号化されたメディアのブロックは、異なるタイプのメディアに対して、異なるプレゼンテーション終了時間を有し得る。ブロック要求ストリーミングシステムでは、そのようなメディアブロックは、各メディアタイプが継続的に再生されるような方式で連続的に再生され得るので、1つのブロックからの1つのタイプのメディアサンプルは、別のタイプの先行するブロック

50

のメディアサンプルの前に再生されてよく、これは本明細書では「連続ブロック接合」と呼ばれる。代替形態として、そのようなメディアブロックは、1つのブロックの任意のタイプの最早のサンプルが、先行するブロックの任意のタイプの最遅のサンプルより後に再生されるような方式で再生されてよく、これは、本明細書では「不連続ブロック接合」と呼ばれる。連続ブロック接合は、両方のブロックが、順番に符号化された、同じコンテンツアイテムおよび同じ表現からのメディアを含む場合、または他の場合に、適切であり得る。通常、1つの表現の中で、連続ブロック接合は、2つのブロックを接合するときに適用され得る。ブロック境界においてメディアトラックを揃える必要なく、既存の符号化が適用されることが可能でありかつセグメント化が行われることが可能であるので、上記のことは有利である。これは、図10に示されており、ビデオストリーム1000は、RAP 1204のようなRAPを伴う、ブロック1202および他のブロックを含む。

10

## 【0210】

## Media Presentation Description

メディアプレゼンテーションは、HTTPストリーミングサーバ上のファイルの構造化された集合体として見られ得る。HTTPストリーミングクライアントは、ストリーミングサービスをユーザに提示するのに十分な情報をダウンロードすることができる。代替的な表現は、3GPPファイルフォーマットに準拠する、または、少なくとも、3GPファイルから、または3GPファイルへ簡単に変換され得るデータ構造の良好に定義されたセットに準拠する、1つもしくは複数の3GPファイルまたは3GPファイルの一部からなり得る。

20

## 【0211】

メディアプレゼンテーションは、`media presentation description`によって記述され得る。`Media Presentation Description (MPD)`は、適切なファイル要求、たとえばHTTP GET要求を構築して適切な時間にデータにアクセスし、ストリーミングサービスをユーザに提供するためにクライアントが使用できる、メタデータを含み得る。`media presentation description`は、適切な3GPPファイルおよびファイルの断片を選択するために、HTTPストリーミングクライアントに対して十分な情報を提供することができる。アクセス可能となるようにクライアントにシグナリングされる単位は、セグメントと呼ばれる。

30

## 【0212】

`media presentation description`はとりわけ、次のような要素および属性を含み得る。

## 【0213】

## MediaPresentationDescriptionの要素

ストリーミングサービスをエンドユーザーに提供するためにHTTPストリーミングクライアントによって使用されるメタデータをカプセル化する要素。`MediaPresentationDescription`の要素は、次の属性および要素の1つまたは複数を含み得る。

40

## 【0214】

`Version`: 拡張性を確実にするためのプロトコルのバージョン数。

## 【0215】

`PresentationIdentifier`: プrezentationが他のプレゼンテーションの中から一意に識別され得るようにする情報。私的なフィールドまたは名称も含み得る。

## 【0216】

`UpdateFrequency`: `media presentation description`の更新頻度。すなわち、実際の`media presentation description`をクライアントがどの程度頻繁にリロードし得るかである。存在しない場合、メディアプレゼンテーションは不变であり得る。メディアプレゼンテーションを更新することは、メディアプレゼンテーションがキャッシュされ得ないことを意味し得る。

## 【0217】

`MediaPresentationDescriptionURI`: `media presentation description`に日付を付けるためのURI。

## 【0218】

50

**Stream:** ストリームまたはメディアプレゼンテーションのタイプ、すなわち、ビデオ、オーディオ、またはテキストを記述する。ビデオストリームタイプは、オーディオを含んでよくかつテキストを含んでよい。

#### 【 0 2 1 9 】

**Service:** 追加の属性を伴うサービスタイプを記述する。サービスタイプは、ライブおよびオンデマンドであり得る。これは、何らかの現在の時間を超える探索およびアクセスが許可されないことをクライアントに知らせるために使用され得る。

#### 【 0 2 2 0 】

**MaximumClientPreBufferTime:** クライアントがメディアストリームを事前にバッファリングできる最大の時間の量。このタイミングは、クライアントがこの最大の事前バッファリング時間を超えてダウンロードすることを制約される場合、ストリーミングをプログレッシブダウンロードに対して差別化し得る。事前バッファリングに関する制約が適用され得ないことを示す値は、存在しなくてよい。10

#### 【 0 2 2 1 】

**SafetyGuardIntervalLiveService:** サーバにおけるライブサービスの最大のターンアラウンドタイムについての情報。現在の時間において情報のいずれがすでにアクセス可能かを示すものをクライアントに提供する。この情報は、クライアントおよびサーバがUTC時間で動作することが予測され、厳密な時間同期が行われていない場合、必要であり得る。

#### 【 0 2 2 2 】

**TimeShiftBufferDepth:** クライアントが現在の時間に対してライブサービスをどの程度まで前に戻せるかについての情報。この深度の延長によって、タイムシフト視聴およびキャッシュアップサービスが、サービスのプロビジョニングの具体的な変更を伴わずに可能にされ得る。20

#### 【 0 2 2 3 】

**LocalCachingPermitted:** このフラグは、ダウンロードされたデータが再生された後でHTTPクライアントがそのデータをローカルにキャッシュできるかどうかを示す。

#### 【 0 2 2 4 】

**LivePresentationInterval:** StartTimeおよびEndTimeを規定することによって表現が利用可能であり得る時間間隔を含む。StartTimeは、サービスの開始時間を示し、EndTimeはサービスの終了時間を示す。EndTimeが規定されない場合、終了時間は現時点では未知であり、UpdateFrequencyが、サービスの実際の終了時間の前にクライアントが終了時間に対するアクセス権を得ることを確実にし得る。30

#### 【 0 2 2 5 】

**OnDemandAvailabilityInterval:** プrezentation間隔は、ネットワーク上でのサービスの利用可能性を示す。複数のプレゼンテーション間隔が提供され得る。HTTPクライアントは、任意の規定された時間枠の外側でサービスにアクセスすることが不可能であり得る。OnDemand Intervalのプロビジョニングによって、追加のタイムシフト視聴が規定され得る。この属性はまた、ライブサービスのために存在し得る。ライブサービスのために存在する場合、サーバは、すべての与えられた利用可能な間隔の間に、オンデマンドサービスとしてサービスにアクセスすることができる。したがって、LivePresentationIntervalは、いずれのOnDemandAvailabilityIntervalとも重複し得ない。40

#### 【 0 2 2 6 】

**MPDFileInfoDynamic:** メディアプレゼンテーション中のファイルのデフォルトの動的な構造を記述する。さらなる詳細が以下で与えられる。MPDレベルに対するデフォルトの仕様は、いくつかのまたはすべての代替的な表現に対して同じ規則が使用される場合には、不必要的繰返しをなくし得る。

#### 【 0 2 2 7 】

**MPDCodecDescription:** メディアプレゼンテーション中の主要なデフォルトのコーデックを記述する。さらなる詳細が以下で与えられる。MPDレベルに対するデフォルトの仕様は、いくつかのまたはすべての表現に対して同じコーデックが使用される場合には、不必要

な繰返しをなくし得る。

**【0228】**

MPDMoveBoxHeaderSizeDoesNotChange:MoveBox Headerがメディアプレゼンテーション全体の中の個々のファイルの間でサイズが変化するかどうかを示すためのフラグ。このフラグは、ダウンロードを最適化するために使用されてよく、特定のセグメントフォーマット、特にセグメントがmoovヘッダを含むフォーマットの場合にのみ存在し得る。

**【0229】**

FileURIPattern:メディアプレゼンテーション内のファイルに対する要求メッセージを生成するためにクライアントによって使用されるパターン。異なる属性は、メディアプレゼンテーション内のファイルの各々の固有のURIの生成を可能にする。基本URIはHTTP URIであり得る。  
10

**【0230】**

Alternative Representation:表現のリストを記述する。

Alternative Representationの要素

**【0231】**

1つの表現に対するすべてのメタデータをカプセル化するXML要素。Alternative Representationの要素は、次の属性および要素を含み得る。

**【0232】**

RepresentationID:メディアプレゼンテーション内のこの特定のAlternative Representationに対する固有のID。  
20

**【0233】**

FileInfoStatic:1つの代替的なプレゼンテーションのすべてのファイルの開始時間およびURIの明示的なリストを提供する。ファイルのリストの不变のプロビジョニングは、メディアプレゼンテーションの正確なタイミングの記述という利点をもたらし得るが、代替的な表現が多くのファイルを含む場合には特に、さほど小型ではないことがある。また、ファイル名は任意の名前を有し得る。

**【0234】**

FileInfoDynamic:1つの代替的なプレゼンテーションの開始時間およびURIのリストを構築するための暗黙的な方法を提供する。ファイルのリストの動的なプロビジョニングは、より小型の表現という利点をもたらし得る。開始時間の順序のみが与えられる場合、タイミングの利点はここでも変わらず保たれるが、ファイル名は、FilePatternURIに基づいて動的に構築されることになる。各セグメントの期間のみが与えられる場合、表現は小型であり、ライブサービス内での使用に適し得るが、ファイルの生成はグローバルなタイミングによって支配され得る。  
30

**【0235】**

APMoveBoxHeaderSizeDoesNotChange:MoveBox HeaderがAlternative Descriptionの中の個々のファイルの間でサイズが変化するかどうかを示すフラグ。このフラグは、ダウンロードを最適化するために使用されてよく、特定のセグメントフォーマット、特にセグメントがmoovヘッダを含むフォーマットの場合にのみ存在し得る。

**【0236】**

APCodecDescription:代替的なプレゼンテーションの中のファイルの主要なコーデックを記述する。

**【0237】**

Media Descriptionの要素

MediaDescription:この表現に含まれるメディアに対するすべてのメタデータをカプセル化し得る要素。具体的には、この代替的なプレゼンテーションの中のトラックについての情報とともに、可能であれば、トラックの推奨されるグルーピングについての情報を含み得る。MediaDescriptionの属性は、次の属性を含む。

**【0238】**

TrackDescription:この表現に含まれるメディアに対するすべてのメタデータをカプセ  
50

ル化するXML属性。TrackDescriptionの属性は、次の属性を含む。

【0239】

TrackID:代替的な表現の中のトラックの固有のID。これは、トラックがグルーピング記述の一部である場合に使用され得る。

【0240】

Bitrate:トラックのビットレート。

【0241】

TrackCodecDescription:このトラックにおいて使用されるコーデックに対するすべての属性を含むXML属性。TrackCodecDescriptionの属性は、次の属性を含む。

【0242】

MediaName:メディアタイプを定義する属性。メディアタイプは、「オーディオ」、「ビデオ」、「テキスト」、「アプリケーション」、および「メッセージ」を含む。

【0243】

Codec:プロファイルおよびレベルを含むコーデックタイプ。

【0244】

LanguageTag:適用可能な場合の言語タグ。

【0245】

MaxWidth,MaxHeight:ビデオに対する、含まれるビデオのピクセル単位の高さおよび幅。

【0246】

SamplingRate:オーディオに対するサンプリングレート。

【0247】

GroupDescription:異なるパラメータに基づく適切なグルーピングのために推奨をクライアントに提供する属性。

【0248】

GroupType:クライアントがそれに基づいてどのようにトラックをグルーピングするかを決めることができるタイプ。

【0249】

media presentation descriptionの中の情報は、有利には、ファイル/セグメントまたはその一部に対する要求を適切な時間に実行するために、HTTPストリーミングクライアントによって使用され、たとえば、アクセス帯域幅、表示能力、コーデック能力などとともに、言語などのユーザの選好に関して、その能力と一致する適切な表現からセグメントを選択する。さらに、Media Presentation descriptionは、グローバルなタイムラインに対して時間的に揃えられ対応付けられる表現を記述するので、クライアントはまた、表現を切り替えるための適切な動作を開始するための進行中のメディアプレゼンテーションの間にMPD中の情報を使用して、表現を一緒に提示し、またはメディアプレゼンテーション内を探索することができる。

【0250】

シグナリングセグメント開始時間

表現は、時間的に複数のセグメントへと分割され得る。あるセグメントの最後のフラグメントと次のセグメントの次のフラグメントとの間に、トラック間タイミング問題が存在する。加えて、不变の継続時間のセグメントが使用される場合、別のタイミング問題が存在する。

【0251】

各セグメントに対して同じ継続時間を使用することは、MPDが小型および不变であるという利点を有し得る。しかしながら、各セグメントは依然として、ランダムアクセスポイントで開始し得る。したがって、ビデオ符号化は、これらの特定の点においてランダムアクセスポイントを与えるように制約されることがあり、または、実際のセグメント継続時間は、正確にはMPDにおいて規定されるようなものではないことがある。ストリーミングシステムがビデオ符号化プロセスに対して不必要な制約を課さないことが望ましいことが

10

20

30

40

50

るので、第2の選択肢が好まれ得る。

#### 【0252】

具体的には、ファイル継続時間がd秒としてMPDにおいて規定される場合、n番目のファイルは、時間(n-1)dにおける、またはその直後のランダムアクセスポイントで開始し得る。

#### 【0253】

この手法では、各ファイルは、グローバルなプレゼンテーション時間を単位とするセグメントの正確な開始時間についての情報を含み得る。これをシグナリングするための3つの可能な方法は、以下のことを含む。

#### 【0254】

(1) 第1に、各セグメントの開始時間を、MPDにおいて規定される厳密なタイミングに制限する。しかし、そうすると、メディアエンコーダはIDRフレームの配置に関して何ら柔軟性をもつことができず、ファイルストリーミングのために特別な符号化を必要とし得る。

10

#### 【0255】

(2) 第2に、各セグメントに対するMPDに厳密な開始時間を追加する。オンデマンドの場合には、MPDの小型性が低下し得る。ライブの場合には、これはMPDの定期的な更新を必要とすることがあり、スケーラビリティが低下し得る。

#### 【0256】

(3) 第3に、表現の告知された開始時間またはMPD中のセグメントの告知された開始時間に対するグローバル時間または厳密な開始時間を、セグメントがこの情報を含むという意味で、セグメントに追加する。これは、適応ストリーミングに専用の新たなボックスに追加され得る。このボックスはまた、「TIDX」または「SIDX」ボックスによって提供されるような情報を含み得る。この第3の手法の結果は、セグメントの1つの始めに近い特定の位置を探すときに、クライアントが、MPDに基づいて、要求される探索点を含むセグメントの後続のセグメントを選び得る、というものである。この場合の単純な応答は、取り出されたセグメントの開始よりも前(すなわち、探索点の後の次のランダムアクセスポイント)に探索点を動かすことであり得る。通常、ランダムアクセスポイントは、少なくとも数秒ごとに設けられる(かつ、ランダムアクセスポイントをよりまれにすることによる符号化の利益がほとんどないことが多い)ので、最悪の場合でも、探索点は規定よりも数秒後に動かされ得る。代替的に、クライアントは、セグメントに対するヘッダ情報を取り出す際に、要求された探索点が以前のセグメントの中に実際にあると判定し、代わりにそのセグメントを要求することができる。これは、探索動作を実行するために必要とされる時間を時々増加させ得る。

20

30

#### 【0257】

##### アクセス可能なセグメントのリスト

メディアプレゼンテーションは、元のメディアコンテンツに対する何らかの異なるバージョンの符号化を各々提供する、表現のセットを含む。有利には、表現自体が、他のパラメータと比較した場合の、表現の差分パラメータについての情報を含む。表現はまた、明示的にまたは暗黙的に、アクセス可能なセグメントのリストを含む。

40

#### 【0258】

セグメントは、メタデータのみを含むタイムレスセグメントおよび主にメディアデータを含むメディアセグメントにおいて区別され得る。Media Presentation Description(「MPD」)は有利には、暗黙的にまたは明示的に、セグメントの各々に対する異なる属性を識別し割り当てる。各セグメントに有利に割り当てられた属性は、セグメントがアクセス可能である期間、それらを通じてセグメントがアクセス可能なリソースおよびプロトコルを含む。加えて、メディアセグメントは、有利には、メディアプレゼンテーション中のセグメントの開始時間、メディアプレゼンテーション中のセグメントの継続時間のような属性を割り当てられる。

#### 【0259】

50

`OnDemandAvailabilityInterval`のような`media presentation description`中の属性において有利に示されるように、メディアプレゼンテーションが「オンデマンド」タイプである場合、`media presentation description`は通常、セグメント全体を記述し、セグメントがアクセス可能である場合およびセグメントがアクセス可能ではない場合を示すものを提供する。セグメントの開始時間は、有利には、同じメディアプレゼンテーションの再生を異なる時間に開始する2つのクライアントが同じ`media presentation description`とともに同じメディアセグメントを使用できるように、メディアプレゼンテーションの開始に対して表現される。これは、有利なことに、セグメントをキャッシュする能力を改善させる。

## 【0260】

10

属性`Service`などの`media presentation description`中の属性によって有利に示されるように、メディアプレゼンテーションが「ライブ」タイプである場合、実際の時刻を超えるメディアプレゼンテーションを含むセグメントは一般に、セグメントがMPDにおいて完全に記述されていても、生成されず、または少なくともアクセス可能ではない。しかしながら、メディアプレゼンテーションサービスが「ライブ」タイプであることを示すものによって、クライアントは、MPDに含まれる情報およびMPDのダウンロード時間に基づく、実時間のクライアント内部時間NOWに対するタイミング属性とともに、アクセス可能なセグメントのリストを生成することができる。実時間NOWにおいてMPDのインスタンスに対して動作している参照クライアントがリソースにアクセスできるように、サーバがリソースをアクセス可能にするという意味で有利に、サーバは動作する。

## 【0261】

20

具体的には、参照クライアントは、MPDに含まれる情報およびMPDのダウンロード時間に基づく、実時間のクライアント内部時間NOWに対するタイミング属性とともに、アクセス可能なセグメントのリストを生成する。時間が進むと、クライアントは同じMPDを使用して、メディアプレゼンテーションを連続的に再生するために使用され得る新たなアクセス可能なセグメントリストを作成する。したがって、サーバは、これらのセグメントが実際にアクセス可能になる前に、MPD中のセグメントを告知することができる。これは、MPDの頻繁な更新およびダウンロードを減らすので、有利である。

## 【0262】

30

各々が開始時間`tS`を伴うセグメントのリストが、`FileInfoStatic`のような要素の中のプレイリストによって明示的に、または、`FileInfoDynamic`のような要素を使用することによって暗黙的に記述されると仮定する。`FileInfoDynamic`を使用するセグメントリストを生成するための有利な方法が、以下で説明される。この構築規則に基づいて、クライアントは、本明細書では`FileURI(r,i)`と呼ばれる、各表現`r`に対するURIのリスト、および、インデックス`i`を伴う各セグメントの開始時間`tS(r,i)`に対するアクセス権を有する。

## 【0263】

セグメントのアクセス可能な時間枠を作成するためにMPD中の情報を使用することは、次の規則を使用して実行され得る。

## 【0264】

40

`Service`のような属性によって有利に示されるように、「オンデマンド」タイプのサービスでは、`OnDemandAvailabilityInterval`のようなMPD要素によって有利に表されるように、クライアントにおける現在の実時間NOWが利用可能な任意の範囲内にある場合、このオンデマンドプレゼンテーションのすべての記述されるセグメントがアクセス可能である。クライアントにおける現在の実時間NOWが利用可能な任意の範囲の外にある場合、このオンデマンドプレゼンテーションの記述されるセグメントのいずれもアクセス可能ではない。

## 【0265】

`Service`のような属性によって有利に示されるように、「ライブ」タイプのサービスでは、開始時間`tS(r,i)`は、有利には、実時間で利用可能な時間を表す。利用可能開始時間は、イベントのライブサービス時間と、キャプチャ、符号化、および公開のためのサーバ

50

における何らかのターンアラウンドタイムとの組合せとして導出され得る。たとえば、このプロセスのための時間は、たとえばMPD中のSafetyGuardIntervalLiveServiceとして規定される安全ガードインターバルtGをたとえば使用して、MPD中で規定され得る。これは、UTC時間と、HTTPストリーミングサーバ上でのデータの利用可能性との差分を最小にする。別の実施形態では、MPDは、イベントのライブ時間とターンアラウンドタイムとの間の差分として、ターンアラウンドタイムを提供することなくMPD中のセグメントの利用可能時間を明示的に規定する。次の説明では、任意のグローバルな時間が利用可能時間として規定されると仮定される。ライブメディアプロードキャスティングの当業者は、この説明を読んだ後、media presentation description中の適切な情報からこの情報を導出することができる。

10

**【0266】**

LivePresentationIntervalのようなMPD要素によって有利に表されるように、クライアントにおける現在の実時間NOWがプレゼンテーション間隔の任意の範囲の外にある場合、このライブプレゼンテーションの記述されるセグメントのいずれもアクセス可能ではない。クライアントにおける現在の実時間NOWがライブプレゼンテーション間隔の中にある場合、このライブプレゼンテーションの記述されるセグメントの少なくともいくつかのセグメントがアクセス可能であり得る。

**【0267】**

アクセス可能なセグメントの制約は、次の値によって支配される。

**【0268】**

20

(クライアントに対して利用可能なものとしての)実時間NOW。

**【0269】**

たとえばmedia presentation description中のTimeShiftBufferDepthとして規定される許容されるタイムシフトバッファ深度tTSB。

**【0270】**

相対的なイベント時間 $t_1$ におけるクライアントは、(NOW-tTSB)からNOWの間隔の中で、または、継続時間dを伴うセグメントの終了時間も含まれて(NOW-tTSB-d)からNOWという間隔をもたらすような間隔の中で、開始時間tS(r,i)を伴うセグメントを要求することのみを許容され得る。

**【0271】**

30

**MPDの更新**

いくつかの実施形態では、サーバは、たとえばサーバの位置が変わるとき、またはメディアプレゼンテーションが異なるサーバからの何らかの広告を含むとき、またはメディアプレゼンテーションの継続時間が未知であるとき、または、サーバが次のセグメントのロケータを不明瞭にすることを望むとき、ファイルまたはセグメントのロケータと、セグメントの開始時間とを事前に知らない。

**【0272】**

そのような実施形態では、サーバは、すでにアクセス可能である、またはMPDのこのインスタンスが公開された直後にアクセス可能になるセグメントのみを記述し得る。さらに、いくつかの実施形態では、クライアントは有利には、ユーザがメディアコンテンツの生成から可能な限り早く、含まれるメディアプログラムを体験するように、MPD中で記述されるメディアに近いメディアを消費する。MPD中の記述されるメディアセグメントの終わりに到達するとクライアントが予測するとすぐに、クライアントは有利には、MPDの新たなインスタンスを要求して、サーバが新たなメディアセグメントを記述する新たなMPDを公開したという予測のもとで連続的な再生を続ける。サーバは有利には、クライアントが連続的な更新のための手順を利用できるように、MPDの新たなインスタンスを生成し、MPDを更新する。サーバは、セグメントの生成とともにMPD更新手順を適合させることができ、共通のクライアントとして動作する参照クライアントの手順が動作し得る。

40

**【0273】**

MPDの新たなインスタンスが前方の短い時間しか記述しない場合、クライアントは、MPD

50

の新たなインスタンスを頻繁に要求する必要がある。これは、スケーラビリティの問題と、不必要に頻繁な要求による不必要的アップリンクおよびダウンリンクのトラフィックとをもたらし得る。

#### 【0274】

したがって、一方では、セグメントを必ずしもまだアクセス可能にすることなく、可能な限り遠い未来のセグメントを記述すること、他方では、新たなサーバの位置を表現し、広告などの新たなコンテンツの挿入を可能にし、またはコーデックパラメータの変更を提供するために、MPDの予測されない更新を可能にすることに、意味がある。

#### 【0275】

さらに、いくつかの実施形態では、メディアセグメントの継続時間は、たとえば数秒の範囲のように、短いことがある。メディアセグメントの継続時間は、有利には、ライブサービス、またはセグメントの記憶または配信を扱う他の様における端末間遅延を補償するために、または他の理由で、配信またはキャッシングの特性に対して最適化され得る適切なセグメントサイズへと調整されることについて柔軟である。セグメントがメディアプレゼンテーション継続時間と比較して小さい場合は特に、大量のメディアセグメントリソースおよび開始時間が、*media presentation description*において記述される必要がある。結果として、*media presentation description*のサイズは大きいことがあり、これは、*media presentation description*のダウンロード時間に悪影響を与えるので、メディアプレゼンテーションの開始遅延、およびアクセスリンク上での帯域幅の使用率にも影響を与える。したがって、プレイリストを使用したメディアセグメントのリストの記述を許容するだけではなく、テンプレートまたはURL構築規則を使用することによる記述も許容することが有利である。テンプレートおよびURL構築規則は、この説明では同義的に使用される。10

#### 【0276】

加えて、テンプレートは、有利には、ライブの場合に現在の時間を超えるセグメントロケータを記述するために使用され得る。そのような場合、MPDの更新は、それ自体はロケータとしては不要であるとともに、セグメントリストはテンプレートによって記述される。しかしながら、それでも、表現またはセグメントの記述の変更を要求する、予測されないイベントが起こり得る。複数の異なるリソースからのコンテンツが一緒に接合される場合、たとえば、広告が挿入された場合、適応HTTPストリーミング*media presentation description*の変更が必要とされ得る。異なるソースからのコンテンツは、種々の方式で異なり得る。別の理由は、ライブプレゼンテーションの間に、ある元のライブサーバから別のサーバへのフェイルオーバーを実現するために、コンテンツファイルに使用されるURLを変更する必要があり得る、ということである。20

#### 【0277】

いくつかの実施形態では、MPDが更新される場合、MPDに対する更新が、次のような意味で更新されたMPDが前のMPDに適合するように実行されることが有利である。上記の意味とは、参照クライアントおよび、したがって任意の実装されるクライアントが、前のMPDの有効時間までの任意の時間に更新されたMPDから、MPDの前のインスタンスから生成したであろうものと同一の機能をもつ、アクセス可能なセグメントのリストを生成する、という意味である。この要件は、(a)更新時間の前は古いMPDに適合しているので、クライアントが古いMPDとの同期を伴わずに新たなMPDを使用して直ちに開始できること、および(b)MPDへの実際の変更が起きる時間と更新時間が同期される必要はないことを確実にする。言い換えると、MPDへの更新は事前に告知されてよく、サーバは、新たな情報が利用可能になると、MPDの異なるバージョンを維持する必要なく、MPDの古いインスタンスを置き換えることができる。30

#### 【0278】

表現のセットまたはすべての表現に対するMPDの更新にまたがるメディアのタイミングについて、2つの可能性が存在し得る。(a)既存のグローバルな時間軸がMPDの更新にまたがって続く(本明細書では「連続MPD更新」と呼ばれる)か、または(b)現在の時間軸が終了40

し、変更の後のセグメントで新たな時間軸が開始する(本明細書では「不連続MPD更新」と呼ばれる)かのいずれかである。

#### 【0279】

これらの選択肢の間の差は、メディアフラグメントの、および、したがってセグメントのトラックが一般に、トラック間の異なるサンプル粒度が原因で、同時に開始しかつ終了しないことを考えると明白であり得る。通常のプレゼンテーションの間、フラグメントの1つのトラックのサンプルは、前のフラグメントの別のトラックのいくつかのサンプルの前にレンダリングされ得る。すなわち、単一のトラック内には重複はないことがあるが、フラグメントの間にはある種の重複が存在する。

#### 【0280】

(a)と(b)の差は、そのような重複がMPDの更新にまたがって可能であり得るかどうかである。MPDの更新が、完全に別々のコンテンツの接合によるものである場合、新たなコンテンツが前のコンテンツと接合されるべき新たな符号化物を必要とするので、そのような重複は一般に実現するのが難しい。したがって、いくつかのセグメントに対する時間軸を再開することによって、メディアプレゼンテーションの不連続な更新のための能力を提供し、場合によっては更新の後の表現の新たなセットも定義することが有利である。また、コンテンツが独立に符号化されセグメント化された場合、コンテンツの前の断片のグローバルな時間軸内に収まるようにタイムスタンプを調整することも避けられる。

#### 【0281】

説明されたメディアセグメントのリストに新たなメディアセグメントを追加するだけである場合などの、更新がより小さな理由によるものである場合、または、URLの位置が変更される場合、重複および連続的な更新が許可され得る。

#### 【0282】

不連続MPD更新の場合、前の表現の最後のセグメントの時間軸は、セグメント中の任意のサンプルの最遅のプレゼンテーション終了時間で終了する。次の表現の時間軸(または、より正確には、新たな期間とも呼ばれる、メディアプレゼンテーションの新たな部分の最初のメディアセグメントの最初のプレゼンテーション時間)は通常、および有利に、シームレスおよび連続的な再生が確実にされるように、最後の期間のプレゼンテーションの終わりと同じこの瞬間に開始する。

#### 【0283】

2つの場合が図11に示されている。

#### 【0284】

MPDの更新をセグメント境界へと制約することが好まれかつ有利である。そのような変更または更新をセグメント境界へと制約する根拠は、次の通りである。第1に、各表現に対するバイナリメタデータ、通常はMovie Headerへの変更は、少なくともセグメント境界において起こり得る。第2に、Media Presentation Descriptionは、セグメントへのポインタ(URL)を含み得る。ある意味で、MPDは、メディアプレゼンテーションと関連付けられるすべてのセグメントファイルと一緒にグルーピングする、「包括的」データ構造である。この包含関係を維持するために、各セグメントは単一のMPDによって参照されてよく、MPDが更新されるとき、MPDは有利にはセグメント境界においてのみ更新される。

#### 【0285】

セグメント境界は一般に揃えられる必要はないが、異なるソースから接合されたコンテンツの場合、および不連続MPD更新の場合は、一般に、セグメント境界を揃えること(具体的には、各表現の最後のセグメントが同じビデオフレームにおいて終了してよく、そのフレームのプレゼンテーション時間よりも後のプレゼンテーション開始時間を伴うオーディオサンプルを含まなくてよいこと)が理にかなう。そして、不連続更新は、期間と呼ばれる共通の瞬間ににおいて表現の新たなセットを開始することができる。表現のこの新たなセットが有効になる開始時間は、たとえば、期間開始時間によって与えられる。各表現の相対的な開始時間は0にリセットされ、期間の開始時間は、グローバルなメディアプレゼンテーションの時間軸におけるこの新たな期間に表現のセットを配置する。

10

20

30

40

50

**【 0 2 8 6 】**

連続MPD更新では、セグメント境界は揃えられる必要はない。各々の代替的な表現の各セグメントは、単一のMedia Presentation Descriptionによって支配され得るので、動作しているMPDにおいて追加のメディアセグメントが記述されないという予測によって一般に引き起こされる、Media Presentation Descriptionの新たなインスタンスに対する更新要求は、消費されることが予測される表現のセットを含む表現の消費されるセットに応じて、異なる時間に起こり得る。

**【 0 2 8 7 】**

より一般的な場合のMPDの要素および属性の更新をサポートするために、表現または表現のセットだけではなく任意の要素が、有効時間と関連付けられ得る。よって、MPDのいくつかの要素が更新される必要がある場合、たとえば、表現の数が変化した場合、またはURL構築規則が変化した場合、これらの要素は各々、つながっていない有効時間を伴う要素の複数のコピーを提供することによって、規定された時間において個々に更新され得る。

10

**【 0 2 8 8 】**

有効時間と関連付けられる説明された要素が、メディアプレゼンテーションのグローバルな時間軸の期間において有効であるように、有効性は有利には、グローバルなメディア時間と関連付けられる。

**【 0 2 8 9 】**

上で論じられたように、一実施形態では、有効時間は、表現の完全なセットにのみ追加される。そして、各々の完全なセットが1つの期間を形成する。そして、有効時間が期間の開始時間を形成する。言い換えると、有効要素を使用する特定の場合には、表現の完全なセットは、表現のセットに対するグローバルな有効時間によって示される、時間的なある期間に有効であり得る。表現のセットの有効時間は、期間と呼ばれる。新たな期間の始めにおいて、前のセットの表現の有効性は期限切れになり、表現の新たなセットが有効になる。有効期間は好ましくはつながっていないことに再び留意されたい。

20

**【 0 2 9 0 】**

上で述べられたように、Media Presentation Descriptionに対する変更は、セグメント境界において発生するので、各表現に対して、要素に対する変更は実際に次のセグメント境界において発生する。クライアントは次いで、メディアのプレゼンテーション時間内の時間の各瞬間にに対するセグメントのリストを含む、有効なMPDを形成することができる。

30

**【 0 2 9 1 】**

不連続ブロック接合は、ブロックが異なる表現からの、または異なるコンテンツからの、たとえば、コンテンツのセグメントおよび広告からのメディアデータを含む場合、または他の場合において、適切であり得る。プレゼンテーションメタデータに対する変更がブロック境界のみで発生することが、ブロック要求ストリーミングシステムにおいて要求され得る。これは、ブロック内のメディアデコーダパラメータを更新することはブロック間のみでそれらを更新することよりも複雑であり得るので、実装上の理由で有利であり得る。この場合、規定された有効間隔の開始よりも早くない最初のブロック境界から、規定された有効間隔の終了よりも早くない最初のブロック境界まで、要素が有効であると見なされるように、上で説明された有効間隔が概算として解釈され得ることが、有利に規定され得る。

40

**【 0 2 9 2 】**

上記の例示的な実施形態は、メディアプレゼンテーションに対する変更と題された、後で提示されるセクションにおいて説明されるブロック要求ストリーミングシステムに対する新規の改善について説明する。

**【 0 2 9 3 】**

セグメント継続時間のシグナリング

不連続更新は、期間と呼ばれるつながっていない一連の間隔へと、プレゼンテーション

50

を実質的に分割する。各期間は、メディアサンプルタイミングに対する固有の時間軸を有する。ある期間内の表現のメディアタイミングは、有利には、各期間に対する、またはある期間中の各表現に対する、セグメント継続時間の別個の小型のリストを規定することによって、示され得る。

#### 【0294】

たとえば期間開始時間と呼ばれる、MPD内の要素と関連付けられる属性は、メディアプロレゼンテーション時間内のいくつかの要素の有効時間を規定することができる。この属性は、MPDの任意の要素に追加され得る(有効性を割り当てられ得る属性が要素に変更され得る)。

#### 【0295】

不連続MPD更新では、すべての表現のセグメントが不連続点において終了し得る。これは一般に、不連続点の前の最後のセグメントが前のセグメントとは異なる継続時間を持つことを、少なくとも示唆する。セグメント継続時間のシグナリングは、すべてのセグメントが同じ継続時間を持つことを示すこと、または、各セグメントに対して別個の継続時間を示すことを伴い得る。セグメント継続時間のリストのための小型の表現を有することが望ましいことがあり、これは、セグメント継続時間の多くが同じ継続時間を持つ場合に効率的である。

#### 【0296】

1つの表現または表現のセットの中の各セグメントの継続時間は、有利には、不連続更新の開始、すなわち、MPDにおいて記述される最後のメディアセグメントまでの期間の開始からの、単一の間隔に対するすべてのセグメント継続時間を規定する、単一の文字列によって実行され得る。一実施形態では、この要素のフォーマットは、セグメント継続時間のエントリーのリストを含む生成物に準拠するテキスト文字列であり、ここで、各エントリーは、継続時間属性durと属性の任意選択の乗数multとを含み、第1のエントリーの継続時間<dur>の第1のエントリーセグメントの<mult>、第2のエントリーの継続時間<dur>の第2のエントリーセグメントの<mult>などを含むことを示す。

#### 【0297】

各継続時間エントリーは、1つまたは複数のセグメントの継続時間を規定する。<dur>値の後に文字「\*」および数が続く場合、この数は、この継続時間を伴う連続するセグメントの数を、秒単位で規定する。乗算記号「\*」がない場合、セグメントの数は1である。後に続く数を伴わずに「\*」が存在する場合、すべての後続のセグメントが規定された継続時間を有し、リスト中にさらなるエントリーはなくてよい。たとえば、文字列「30\*」は、30秒という継続時間をすべてのセグメントが有することを意味する。文字列「30\*12 1 0.5」は、継続時間30秒の12個のセグメントに、継続時間10.5秒の1つのセグメントが続くことを示す。

#### 【0298】

セグメント継続時間が各々の代替的な表現に対して別々に規定される場合、各間隔中のセグメント継続時間の合計は、各表現に対して同一であり得る。ビデオトラックの場合、間隔は、各々の代替的な表現において同じフレームで終了し得る。

#### 【0299】

当業者は、本開示を読めば、セグメント継続時間を小型に表現するための同様のまたは等価な方法を見出し得る。

#### 【0300】

別の実施形態では、セグメントの継続時間は、単一の継続時間属性<duration>によって、最後の1つを除いて表現中のすべてのセグメントに対して一定であるものとしてシグナリングされる。次の不連続更新の開始点または新たな期間の開始が提供される限り、これは、最後のセグメントの継続時間が次の期間の開始にまで達することを示唆し、不連続更新の前の最後のセグメントの継続時間は、より短くてよい。

#### 【0301】

表現メタデータに対する変更および更新

10

20

30

40

50

Movie Header「moov」の変更のようなバイナリコーディングされた表現メタデータの変更を示すことは、異なる方法で達成され得る。すなわち、(a)MPDの中で参照される別個のファイル中に、すべての表現に対する1つのmoovボックスがあり得る、(b)各Alternative Representationの中で参照される別個のファイル中に、各々の代替的な表現に対する1つのmoovボックスがあり得る、(c)各セグメントがmoovボックスを含み得るので、自己完結型である、(d)MPDとともに1つの3GPファイルの中にすべての表現に対するmoovボックスがあり得る、である。

#### 【0302】

(a)および(b)の場合、単一の「moov」は有利には、「moov」ボックスの有効な時間がつながっていない限りより多くの「moov」ボックスがMPDにおいて参照され得るという意味で、上記の有効性という概念と組み合わされ得る。たとえば、期間の境界の定義によって、古い期間における「moov」の有効性は、新たな期間の開始によって期限切れになり得る。

10

#### 【0303】

選択肢(a)の場合、単一のmoovボックスへの参照が、有効要素を割り当てられ得る。Multiple Presentationヘッダが許可され得るが、一度に1つのみが有効であり得る。別の実施形態では、ある期間中の表現のセット全体の有効時間、または上で定義されたような期間全体の有効時間が、通常はmoovヘッダとして提供される、この表現メタデータに対する有効時間として使用され得る。

#### 【0304】

選択肢(b)の場合、各表現のmoovボックスへの参照が、有効要素を割り当てられ得る。Multiple Representationヘッダが許可され得るが、一度に1つのみが有効であり得る。別の実施形態では、表現全体の有効時間、または上で定義されたような期間全体の有効時間が、通常はmoovヘッダとして提供される、この表現メタデータに対する有効時間として使用され得る。

20

#### 【0305】

選択肢(c)の場合、MPD中のシグナリングは追加されなくてよいが、moovボックスが今後来るセグメントのいずれかのために変化するかどうかを示すために、メディアストリーム中に追加のシグナリングが追加され得る。これは、「セグメントメタデータ内での更新のシグナリング」において以下でさらに説明される。

30

#### 【0306】

##### セグメントメタデータ内での更新のシグナリング

可能性のある更新についての知識を得るためにmedia presentation descriptionを頻繁に更新するのを避けるために、メディアセグメントとともに任意のそのような更新をシグナリングすることが有利である。media presentation descriptionのような更新されたメタデータが利用可能であり、アクセス可能なセグメントリストの作成を続けることを成功させるためにある時間内にアクセスされなければならないことを示し得る、追加の1つまたは複数の要素がメディアセグメント自体の中で提供され得る。加えて、そのような要素は、更新されたメタデータファイルについて、URLのようなファイル識別子、または、ファイル識別子を構築するために使用され得る情報を提供することができる。更新されたメタデータファイルは、有効間隔も付随する追加のメタデータとともに、有効間隔を示すように修正されたプレゼンテーションに対する元のメタデータファイルにおいて提供されるメタデータと等しい、メタデータを含み得る。そのような指示は、メディアプレゼンテーションに対するすべての利用可能な表現のメディアセグメントにおいて提供され得る。メディアブロック内でそのような指示を検出すると、ブロック要求ストリーミングシステムにアクセスするクライアントは、ファイルダウンロードプロトコルまたは他の手段を使用して、更新されたメタデータファイルを取り出すことができる。こうして、クライアントは、media presentation descriptionの変更についての情報と、変更が起きる、または起きた時間についての情報とを与えられる。有利には、各クライアントは、あり得る更新または変更に対して多くの回数ファイルを「ポーリング」し受信するのではなく、そ

40

50

のような変化が起きたときに一度だけ、更新されたmedia presentation descriptionを要求する。

#### 【0307】

変更の例は、表現の追加または除去、ビットレート、解像度、アスペクト比、含まれるトラックまたはコーデックのパラメータの変更などの1つまたは複数の表現への変更、および、URL構築規則に対する変更、たとえば広告のための異なる発信元サーバを含む。一部の変更は、表現と関連付けられるMovie Header(「moov」)アトムのような初期化セグメントにのみ影響を与えるが、他の変更は、Media Presentation Description(MPD)に影響を与える。

#### 【0308】

オンデマンドコンテンツの場合、これらの変更およびタイミングは事前に知られていてよく、Media Presentation Descriptionにおいてシグナリングされ得る。

#### 【0309】

ライブコンテンツでは、変更は、変更が起こる点まで知られないことがある。1つの解決法は、特定のURLにおいて利用可能なMedia Presentation Descriptionが動的に更新されるのを可能にすること、および変更を検出するためにこのMPDを定期的に要求するようクライアントに求めることである。この解決法は、スケーラビリティ(元のサーバの負荷およびキャッシュの効率性)の観点で欠点を有する。多数の視聴者がいる状況では、キャッシュは、以前のバージョンがキャッシュからなくなった後、かつ、新たなバージョンが受信されこれらのすべてが発信元サーバに転送され得る前に、MPDに対する多くの要求を受信することができる。発信元サーバは、MPDの各々の更新されたバージョンに対するキャッシュから、常に要求を処理する必要があり得る。また、更新は、簡単には、メディアプレゼンテーションの変更と時間的に揃えられないことがある。

10

#### 【0310】

HTTPストリーミングの利点の1つは、スケーラビリティのために、標準的なウェブインフラストラクチャおよびサービスを利用する能力であるので、好ましい解決法は、「不变の」(すなわち、キャッシュ可能な)ファイルのみを伴ってよく、ファイルが変化したかどうかを確認するためにクライアントがファイルを「ポーリング」することに依存しない。

#### 【0311】

media presentation description、および、適応HTTPストリーミングメディアプレゼンテーションにおける「moov」アトムのようなバイナリ表現メタデータを含む、メタデータの更新を解決するための解決法が論じられ提案される。

20

#### 【0312】

ライブコンテンツの場合、MPDが構築されるとき、MPDまたは「moov」が変化し得る点は知られていないことがある。更新を確認するためのMPDの頻繁な「ポーリング」は、帯域幅およびスケーラビリティの理由で一般に避けられるべきであるので、MPDに対する更新は、セグメントファイル自体の中で「帯域内」で示されてよく、すなわち、各メディアセグメントは、更新を示すための選択肢を有し得る。上記のセグメントフォーマット(a)から(c)に応じて、異なる更新がシグナリングされ得る。

#### 【0313】

一般に、セグメント内の信号において、以下の指示、すなわち、この表現内の次のセグメント、または現在のセグメントの開始時間よりも大きな開始時間を有する任意の次のセグメントを要求する前にMPDが更新され得るという、インジケータが有利に提供され得る。更新は事前に告知されてよく、更新が次のセグメントよりも遅い任意のセグメントにおいてのみ起こる必要があることを示す。このMPD更新はまた、メディアセグメントのロケータが変更された場合、Movie Headerのようなバイナリ表現メタデータを更新するために使用され得る。別の信号は、このセグメントが完了すると、時間を前に進めるこれ以上のセグメントが要求されるべきではないことを示し得る。

30

#### 【0314】

セグメントがセグメントフォーマット(c)に従ってフォーマットされる場合、すなわち

40

50

、各メディアセグメントがムービーヘッダのような自己初期化メタデータを含み得る場合、後続のセグメントが更新されたMovie Header(moov)を含むことを示す、さらなる別の信号が追加され得る。これは、有利には、ムービーヘッダがセグメント中に含まれることを可能にするが、Movie Headerは、前のセグメントがMovie Header更新を示す場合、または、表現を切り替えるときの探索またはランダムアクセスの場合にのみ、クライアントによって要求される必要がある。他の場合には、クライアントは、ダウンロードからムービーヘッダを除外するセグメントにバイト範囲要求を出すことができるので、有利なことに帯域幅を節約する。

## 【0315】

さらに別の実施形態では、MPD更新の指示がシグナリングされる場合、信号はまた、更新されたMedia Presentation Descriptionに対するURLのようなロケータを含み得る。更新されたMPDは、不連続更新の場合の新たな期間および古い期間のような有効属性を使用して、更新の前と後の両方で、表現を記述することができる。これは、有利には、以下でさらに説明されるようなタイムシフト視聴を可能にするように使用され得るが、有利には、MPDの含む変更が効力をもつ前に、MPD更新が任意の時間にシグナリングされることも可能にする。クライアントは、直ちに新たなMPDをダウンロードし、それを進行中のプレゼンテーションに適用することができる。

## 【0316】

特定の実現形態では、media presentation descriptionに対する任意の変更、moovヘッダ、またはプレゼンテーションの終了のシグナリングは、ISOベースのメディアファイルフォーマットのボックス構造を使用してセグメントフォーマットの規則に従ってフォーマットされる、ストリーミング情報ボックスに含まれ得る。このボックスは、異なる更新のいずれに対しても固有の信号を提供し得る。

## 【0317】

Streaming Information Box

## 【0318】

定義

## 【0319】

ボックスタイプ:「snif」

コンテナ:なし

必須性:なし

量:0または1

## 【0320】

Streaming Information Boxは、ファイルがその一部であるストリーミングプレゼンテーションについての情報を含む。

## 【0321】

シンタックス

## 【0322】

```
aligned(8) class StreamingInformationBox
extends FullBox('snif'){
    unsigned int(32) streaming_information_flags;
```

## 【0323】

///以下は任意選択のフィールドである

```
    string mpd_location
}
```

## 【0324】

セマンティクス

## 【0325】

streaming\_information\_flagsは、次のうちの0個以上の論理和を含む。

## 【0326】

10

20

30

40

50

0x00000001 Movie Header 更新が後に続く  
【0327】  
0x00000002 Presentation Description の更新  
【0328】  
0x00000004 プレゼンテーションの終了  
【0329】

mpd\_locationは、Presentation Description updateフラグが設定される場合のみ存在し、新たなMedia Presentation Descriptionに対するUniform Resource Locatorを提供する。

【0330】

ライブサービスのためのMPD更新に対する例示的な使用事例

サービス提供者が、本明細書で説明される改善されたブロック要求ストリーミングを使用して、生のフットボールイベントを提供することを望んでいると仮定する。場合によっては、数百万のユーザが、イベントのプレゼンテーションにアクセスすることを望み得る。ライブイベントは、中断が宣言されるとき、または他の動作の一時休止のときに休憩により散発的に中断され、この間に、広告が追加され得る。通常、休憩の正確なタイミングの事前の通知は、まったくまたはほとんどない。

【0331】

サービス提供者は、ライブイベントの間の機器のいずれかの故障に備えて、シームレスな切り替えを可能にするために、冗長なインフラストラクチャ(たとえば、エンコーダおよびサーバ)を提供する必要があり得る。

【0332】

ユーザのアンナが、自分のモバイルデバイスでバスの中からサービスにアクセスし、サービスが直ちに利用可能になると仮定する。彼女の隣に座っているのは別のユーザのポールであり、彼は自分のラップトップコンピュータでイベントを見ている。ゴールが記録され、2人の両方が同時にこのイベントを祝う。その試合の最初のゴールはさらに興奮するものであったことをポールはアンナに話し、アンナは、時間的に30分前のイベントを見られるように、サービスを使用する。そのゴールを見た後、彼女はライブイベントに戻る。

【0333】

その使用事例に対処するために、サービス提供者は、MPDを更新し、更新されたMPDが利用可能であることをクライアントにシグナリングし、クライアントがほぼリアルタイムでデータを提示できるようにクライアントがストリーミングサービスにアクセスすることを許容することが可能でなければならない。

【0334】

MPDの更新は、本明細書の他の箇所で説明されるように、セグメントの配信と同期しない方式で実行可能である。サーバは、MPDがある時間の間更新されないことを、受信機に対して保証することができる。サーバは、現在のMPDを利用し得る。しかしながら、MPDが何らかの最小更新期間の前に更新される場合、明示的なシグナリングは必要とされない。

【0335】

クライアントは異なるMPD更新インスタンスについて動作していることがあり、したがってクライアントはドリフトを有し得るので、完全に同期した再生はほとんど達成されない。MPD更新を使用して、サーバは変更を伝えることができ、クライアントは、プレゼンテーションの間でも、変更を警告され得る。セグメントごとの帯域内シグナリングが、MPDの更新を示すために使用され得るので、更新は、セグメント境界に制限され得るが、それは多くの用途では許容可能であるはずである。

【0336】

MPDの実時間での公開時間と、MPD更新が必要であることをシグナリングするためにセグメントの始めに追加される任意選択のMPD更新ボックスとを提供する、MPD要素が追加され得る。更新は、MPDのように、階層的に行われ得る。

【0337】

10

20

30

40

50

MPD「公開時間」は、MPDの固有の識別子と、MPDがいつ出されたかとを提供する。それはまた、更新手順のためのアンカーを提供する。

## 【0338】

MPD更新ボックスは、「styp」ボックスの後のMPDにおいて見い出され、かつBox Type='mupe'によって定義されてよく、コンテナを必要とせず、必須ではなく、かつ0または1という量を有する。MPD更新ボックスは、セグメントがその一部であるメディアプレゼンテーションについての情報を含む。

## 【0339】

例示的なシンタックスは次の通りである。

## 【0340】

10

```
aligned(8) class MPDUpdateBox
```

## 【0341】

```
extends FullBox('mupe') {
```

## 【0342】

```
unsigned int(3) mpd_information_flags;
```

## 【0343】

```
unsigned int(1) new_location_flag;
```

## 【0344】

```
unsigned int(28) latest_mpd_update_time;
```

## 【0345】

20

///以下は任意選択のフィールドである

## 【0346】

```
string mpd_location
```

## 【0347】

```
}
```

## 【0348】

クラスMPDUpdateBoxの様々なオブジェクトのセマンティクスは次の通りであり得る。

## 【0349】

mpd\_information\_flags:次のうちの0個以上の論理和

0x00 Media Presentation Descriptionの今の更新

30

## 【0350】

0x01 Media Presentation Descriptionの前方の更新

0x02 プrezentationの終了

## 【0351】

0x03-0x07 予備

## 【0352】

new\_location\_flag:1に設定される場合、新たなMedia Presentation Descriptionがmpd\_locationで規定される新たな位置において利用可能である。

## 【0353】

latest\_mpd\_update\_time:最遅のMPDのMPD発行時間に対する、MPD更新が必要になるまでの時間(ms単位)を規定する。クライアントは、今との間の任意の時間にMPDを更新することを選び得る。

40

## 【0354】

mpd\_locationは、new\_location\_flagが設定される場合にのみ存在し、存在する場合、新たなMedia Presentation Descriptionに対するUniform Resource Locatorを提供する。

## 【0355】

更新によって使用される帯域幅が問題である場合、サーバは、これらの一部のみが更新されるように、何らかのデバイス能力のためのMPDに提供することができる。

## 【0356】

50

## タイムシフト視聴およびネットワークPVR

タイムシフト視聴がサポートされる場合、セッションの継続時間の間に、2つ以上のMPDまたはMovie Headerが有効であるということが起こり得る。この場合、必要なときにMPDを更新することによって、しかし有効性機構または期間の概念を追加することによって、有効なMPDが時間枠全体で存在することができる。これは、任意のMPDおよびMovieヘッダが、タイムシフト視聴のための有効な時間枠内にある任意の期間に対して告知されることを、サーバが確実にし得るということを意味する。利用可能なMPDおよび現在のプレゼンテーションに対するメタデータが有効であることを確実にするのは、クライアント次第である。小規模のMPD更新のみを使用した、ライブセッションのネットワークPVRセッションへの移行も、サポートされ得る。

10

### 【0357】

#### 特別なメディアセグメント

ISO/IEC 14496-12のファイルフォーマットがブロック要求ストリーミングシステム内で使用されるときの問題は、上で説明されたように、複数のファイル中のプレゼンテーションの単一のバージョンに対するメディアデータを、連続的な時間セグメントへと配列された状態で記憶するのが有利であり得ることである。さらに、各ファイルがランダムアクセスポイントで開始するように配列するのが有利であり得る。さらに、ビデオ符号化プロセスの間に探索点の位置を選び、プレゼンテーションを、符号化プロセスの間に行われた探索点の選択に基づいて、探索点で各自開始する複数のファイルへとセグメント化することが有利であることがあり、各ランダムアクセスポイントは、ファイルの始めに配置されることもまたはされないこともあるが、各ファイルはランダムアクセスポイントで開始する。上で説明された特性を伴う一実施形態では、プレゼンテーションメタデータ、またはMedia Presentation Descriptionは、各ファイルの正確な継続時間を含んでよく、継続時間はたとえば、ファイルのビデオメディアの開始時間と次のファイルのビデオメディアの開始時間の差を意味するものと理解される。プレゼンテーションメタデータ中のこの情報に基づいて、クライアントは、メディアプレゼンテーションのためのグローバルな時間軸と、各ファイル内のメディアのためのローカルな時間軸との対応付けを構築することが可能である。

20

### 【0358】

別の実施形態では、プレゼンテーションメタデータのサイズは、有利には、代わりに、各々のファイルまたはセグメントが同じ継続時間を有することを規定することによって、低減され得る。しかしながら、この場合、およびメディアファイルが上の方法に従って構築される場合、ランダムアクセスポイントがファイルの開始からちょうど規定された継続時間の位置にある点に存在しないがあるので、各ファイルの継続時間は、media presentation descriptionで規定される継続時間とは厳密には等しくないことがある。

30

### 【0359】

上で言及された相違にもかかわらず、ブロック要求ストリーミングシステムの正常な動作を実現するための、本発明のさらなる実施形態がここで説明される。この方法では、ファイル内のメディアのローカルな時間軸(ファイル中のメディアサンプルの復号および合成のタイムスタンプがそれに対してISO/IEC 14496-12に従って規定される、タイムスタンプ0から開始する時間軸が意図される)の、グローバルなプレゼンテーション時間軸へのマッピングを規定する要素が、各ファイル内で提供され得る。このマッピング情報は、ローカルなファイル時間軸でのタイムスタンプ0に対応する、グローバルなプレゼンテーション時間における単一のタイムスタンプを含み得る。マッピング情報は、代替的に、ローカルなファイル時間軸におけるタイムスタンプ0に対応するグローバルなプレゼンテーション時間と、ファイルの開始に対応するグローバルなプレゼンテーション時間との間の差を、プレゼンテーションメタデータで提供された情報に従って規定する、オフセット値を含み得る。

40

### 【0360】

そのようなボックスの例は、たとえば、トラックフラグメントメディア調整(「tfma」)

50

ボックスとともに、トラックフラグメント復号時間(「`tfdt`」)ボックスまたはトラックフラグメント調整ボックス(「`tfad`」)であり得る。

#### 【0361】

セグメントリスト生成を含む例示的なクライアント

例示的なクライアントがここで説明される。これは、MPDの適切な生成および更新を確実にするための、サーバのための参照クライアントとして使用され得る。

#### 【0362】

HTTPストリーミングクライアントは、MPDにおいて提供される情報によって導かれる。クライアントは、時間T、すなわち、MPDの受信に成功することが可能であった時間においてクライアントが受信したMPDへのアクセス権を有すると仮定される。受信が成功したと判定することは、クライアントが更新されたMPDを取得すること、または、前の受信の成功からMPDが更新されていないことをクライアントが確認することを含み得る。

10

#### 【0363】

例示的なクライアントの挙動が紹介される。連続的なストリーミングサービスをユーザに提供するために、クライアントはまずMPDを解析し、場合によってはプレイリストを使用して、またはURL構築規則を使用して、以下で説明されるようなアカウントセグメントリスト生成手順を考慮して、現在のシステム時間におけるクライアントローカル時間に対する各表現のためのアクセス可能なセグメントのリストを作成する。次いで、クライアントは、表現属性中の情報および他の情報、たとえば、利用可能な帯域幅およびクライアント能力に基づいて、1つまたは複数の表現を選択する。グルーピングに応じて、表現は、単独で、または他の表現と一緒に提示され得る。

20

#### 【0364】

各表現に対して、クライアントは、表現に対する「`moov`」ヘッダのようなバイナリメタデータ、および、もしあれば、選択された表現のメディアセグメントを取得する。クライアントは、場合によってはセグメントリストを使用して、セグメントまたはセグメントのバイト範囲を要求することによって、メディアコンテンツにアクセスする。クライアントは最初に、プレゼンテーションを開始する前にメディアをバッファリングすることができ、プレゼンテーションが開始すると、クライアントは、MPD更新手順を考慮してセグメントまたはセグメントの一部を継続的に要求することによって、メディアコンテンツの消費を続ける。

30

#### 【0365】

クライアントは、更新されたMPD情報および/または環境からの更新された情報、たとえば、利用可能な帯域幅の変化を考慮して、表現を切り替えることができる。ランダムアクセスポイントを含むメディアセグメントに対する任意の要求によって、クライアントは異なる表現に切り替えることができる。前に進むと、すなわち、現在のシステム時間(プレゼンテーションに対する時間を表すために「NOW時間」と呼ばれる)が進むと、クライアントはアクセス可能なセグメントを消費する。NOW時間の各々の進行によって、クライアントは、場合によっては、本明細書で規定される手順に従って、各プレゼンテーションに対するアクセス可能なセグメントのリストを拡張する。

40

#### 【0366】

メディアプレゼンテーションの終わりにまだ到達しておらず、かつ、現在の再生時間が、任意の消費しているまたは消費されるべきプレゼンテーションに対する、MPDにおいて記述されるメディアのメディアを使い果たすとクライアントが予測する閾値内に入った場合、クライアントは、新たなフェッチ時間受信時間Tによって、MPDの更新を要求することができる。受信されると、クライアントは次いで、アクセス可能なセグメントリストの生成において、場合によっては更新されるMPDおよび新たな時間Tを考慮する。図29は、クライアントでの異なる時間におけるライブサービスのための手順を示す。

#### 【0367】

アクセス可能なセグメントリストの生成

HTTPストリーミングクライアントがMPDへのアクセス権を有し、実時間NOWに対してアク

50

セス可能なセグメントリストを生成することを望み得ると仮定する。クライアントはある精度でグローバルな時間基準に対して同期されるが、有利には、HTTPストリーミングサーバに対する直接の同期は必要とされない。

#### 【 0 3 6 8 】

各表現に対するアクセス可能なセグメントリストは、好ましくは、セグメント開始時間およびセグメントロケータのペアのリストとして定義され、セグメント開始時間は、一般性を失うことなく、表現の開始に対するものとして定義され得る。この概念が適用される場合、表現の開始は、期間の開始と揃えられ得る。それ以外の場合は、表現の開始はメディアプレゼンテーションの始めにあり得る。

#### 【 0 3 6 9 】

クライアントは、たとえば、本明細書でさらに定義されるような、URL構築規則およびタイミングを使用する。説明されたセグメントのリストが取得されると、このリストはさらにアクセス可能なセグメントへと制約され、これは、完全なメディアプレゼンテーションのセグメントのサブセットであり得る。構築は、クライアントのNOW時間における、時計の現在値によって支配される。一般に、セグメントは、利用可能時間のセットの中の、任意の時間NOWに対してのみ利用可能である。この枠の外にある時間NOWに対しては、セグメントは利用可能ではない。加えて、ライブサービスでは、何らかの時間checktimeが、どの程度未来までメディアが記述されるかについての情報を提供することを前提とする。checktimeは、MPDにより記録されるメディア時間軸上で定義される。クライアントの再生時間がchecktimeに達すると、クライアントは有利には新たなMPDを要求する。

10

#### 【 0 3 7 0 】

次いで、セグメントリストはさらに、利用可能なメディアセグメントが、メディアセグメントの開始時間と表現の開始時間の合計が、NOW引くtimeShiftBufferDepth引く最後の記述されたセグメントの継続時間と、checktimeまたはNOWのいずれか小さい方の値との間隔に入るようなメディアセグメントのみであるように、MPD属性TimeShiftBufferDepthとともにchecktimeによって制約される。

#### 【 0 3 7 1 】

##### スケーラブルブロック

場合によっては、受信機において現在受信されている1つまたは複数のブロックが、プレゼンテーションの一時停止を伴わずに再生されるように時間内に完全に受信される見込みがなくなるほど、利用可能な帯域幅が少なくなる。受信機は、そのような状況を事前に検出し得る。たとえば、受信機は、6単位時間ごとに5単位のメディアを符号化するブロックを受信しており、4単位のメディアのバッファを有すると判定し得るので、受信機は、約24単位時間後に、プレゼンテーションをストールまたは一時休止させなければならないと予測し得る。十分な通知によって、受信機は、たとえば、ブロックの現在のストリームを中止することによってそのような状況に反応し、コンテンツの異なる表現からの1つまたは複数のブロック、たとえば、単位再生時間当たりにより少量の帯域幅を使用するブロックを要求することを開始し得る。たとえば、受信機が、同じサイズのブロックに対して少なくとも20%長いビデオ時間をブロックが符号化した表現へと切り替えると、受信機は、帯域幅の状況が改善されるまで、ストールする必要性をなくすことが可能であり得る。

30

#### 【 0 3 7 2 】

しかしながら、受信機に、中止された表現からすでに受信されていたデータを完全に廃棄させるのは、無駄であり得る。本明細書で説明されるブロックストリーミングシステムの実施形態では、各ブロック内のデータは、ブロック内のデータのいくつかのプレフィックスが、受信されているブロックの残りを伴わずにプレゼンテーションを続けるために使用され得るように、符号化され構成され得る。たとえば、スケーラブルビデオ符号化のよく知られている技法が使用され得る。そのようなビデオ符号化方法の例には、H.264 Scalable Video Coding(SVC)またはH.264 Advanced Video Coding(AVC)の時間的スケーラビリティがある。有利には、この方法は、1つまたは複数のブロックの受信が、たとえば利用可能な帯域幅の変化が原因で中止され得る場合であっても、プレゼンテーションが

40

50

、受信されているブロックの一部に基づいて継続することを可能にする。別の利点は、単一のデータファイルがコンテンツの複数の異なる表現に対するソースとして使用され得る、ということである。これは、たとえば、要求された表現に対応するブロックのサブセットを選択する、HTTP partial GET要求を利用することによって可能である。

#### 【0373】

本明細書で詳述される1つの改善は、改善されたセグメント、スケーラブルなセグメントマップである。スケーラブルなセグメントマップは、セグメント中の異なるレイヤの位置を含むので、クライアントはそれに従ってセグメントの一部にアクセスし、レイヤを抽出することができる。別の実施形態では、セグメント中のメディアデータは、セグメントの始めから徐々にデータをダウンロードするときに、セグメントの品質が上がるよう、順序付けられる。別の実施形態では、フラグメント要求がスケーラブルな手法に対応するように行われ得るように、品質の漸進的な上昇が、セグメントに含まれる各ブロックまたはフラグメントに対して適用される。

#### 【0374】

図12は、スケーラブルなブロックのある態様を示す図である。その図では、送信機1200は、メタデータ1202、スケーラブルレイヤ1(1204)、スケーラブルレイヤ2(1206)、およびスケーラブルレイヤ3(1208)を出力し、後ろに書かれたものの方が遅れている。受信機1210は次いで、メタデータ1202、スケーラブルレイヤ1(1204)、およびスケーラブルレイヤ2(1206)を使用して、メディアプレゼンテーション1212を提示することができる。

#### 【0375】

##### 独立したスケーラビリティレイヤ

上で説明されたように、受信機がメディアデータの特定の表現の要求されたブロックを、再生のために時間内に受信することが不可能である場合に、ストールしなければならないことは、悪いユーザ体験をもたらすので、ブロック要求ストリーミングシステムにとって望ましくない。プレゼンテーションの任意の所与の部分が時間内に受信されない見込みがほとんどなくなるように、利用可能な帯域幅よりもはるかに少なくなるように、選ばれる表現のデータレートを制約することによって、ストールは回避され、減らされ、または軽減され得るが、この戦略は、利用可能な帯域幅によって原理的にサポートされ得るものよりもメディア品質が必然的にはるかに低くなるという欠点を有する。可能なものよりも低い品質のプレゼンテーションは、悪いユーザ体験として解釈され得る。したがって、ブロック要求ストリーミングシステムの設計者は、クライアント手順、クライアントのプログラミング、またはハードウェアの構成の設計において、ユーザが悪いメディア品質を被り得る、利用可能な帯域幅よりもはるかに低いデータレートを有するコンテンツバージョンを要求すること、または利用可能な帯域幅が変化するにつれてプレゼンテーションの間にユーザが高確率の一時停止を被り得る、利用可能な帯域幅に近いデータレートを有するコンテンツバージョンを要求することのいずれかの、選択に直面する。

#### 【0376】

そのような状況を処理するために、本明細書で説明されるブロックストリーミングシステムは、受信機がレイヤ化された要求を行うことができ送信機がレイヤ化された要求に応答できるように、複数のスケーラビリティレイヤを独立に処理するように構成され得る。

#### 【0377】

そのような実施形態では、各ブロックに対する符号化メディアデータは、本明細書では「レイヤ」と呼ばれる複数のつながっていない断片に区分され得るので、レイヤの組合せは、ブロックに対するメディアデータの全体を含み、レイヤのいくつかのサブセットを受信したクライアントは、コンテンツの表現の復号および提示を実行することができる。この手法では、ストリーム中のデータの順序は、連続的な範囲が品質的に向上するようなものであり、メタデータはこのことを反映する。

#### 【0378】

上記の性質を伴うレイヤを生成するために使用され得る技法の例は、たとえばITU-T Standards H.264/SVCにおいて記載されるような、Scalable Video Codingの技法である

10

20

30

40

50

。上記の性質を伴うレイヤを生成するために使用され得る技法の別の例は、たとえばITU-T Standards H.264/SVCにおいて提供されるような、時間的スケーラビリティレイヤの技法である。

#### 【0379】

これらの実施形態では、メタデータは、MPDにおいて、またはセグメント 자체において提供されてよく、このことは、任意の所与のブロックの個々のレイヤおよび/またはレイヤの組合せおよび/または複数のブロックの所与のレイヤおよび/または複数のブロックのレイヤの組合せに対する要求の構築を可能にする。たとえば、ブロックを含むレイヤが単一のファイル内に記憶されてよく、個々のレイヤに対応するファイル内のバイト範囲を規定するメタデータが提供されてよい。

10

#### 【0380】

バイト範囲を規定することが可能なファイルダウンロードプロトコル、たとえばHTTP1.1が、個々のレイヤまたは複数のレイヤを要求するために使用され得る。さらに、本開示を検討した当業者には明らかなように、可変サイズのブロックの構築、要求、およびダウンロード、ならびにブロックの可変の組合せに関して上で説明された技法は、この文脈においても適用され得る。

#### 【0381】

##### 組合せ

上で説明されたような、レイヤへと区分されたメディアデータの使用による既存の技法と比較して、ユーザ体験の改善および/またはサービングインフラストラクチャの能力の要件の低減を達成するために、ブロック要求ストリーミングクライアントによって有利に利用され得る、いくつかの実施形態がここで説明される。

20

#### 【0382】

第1の実施形態では、ブロック要求ストリーミングシステムの既知の技法は、いくつかの場合には異なるバージョンのコンテンツが異なる組合せのレイヤによって置き換えられるという修正を伴って、適用され得る。すなわち、既存のシステムがコンテンツの2つの別個の表現を提供し得る場合、ここで説明される改善されたシステムは2つのレイヤを提供することができ、既存のシステムにおけるコンテンツの1つの表現が、改善されたシステムにおける第1のレイヤに対して、ピットレート、品質および場合によっては他の尺度に関して同様である場合、既存のシステムにおけるコンテンツの第2の表現は、改善されたシステムにおける2つのレイヤの組合せに対して、ピットレート、品質および場合によっては他の尺度に関して同様である。結果として、改善されたシステム内で必要とされる記憶容量は、既存のシステムにおいて必要とされるものと比較して低減される。さらに、既存のシステムのクライアントは、1つの表現または他の表現のブロックに対する要求を出すことができるが、改善されたシステムのクライアントは、ブロックの第1のレイヤまたは両方のレイヤのいずれかに対する要求を出すことができる。結果として、2つのシステムにおけるユーザ体験は同様である。さらに、異なる品質に対しても、より高い確率でキャッシュされる共通のセグメントが使用されるので、改善されたキャッシングが実現される。

30

#### 【0383】

第2の実施形態では、ここで説明されるレイヤの方法を利用する改善されたブロック要求ストリーミングシステム中のクライアントは、メディア符号化のいくつかのレイヤの各々に対して別個のデータバッファを保持し得る。クライアントデバイス内のデータ管理の当業者には明らかなように、これらの「別個の」バッファは、別個のバッファに対する物理的または論理的に別個のメモリ領域の割振りによって、または、バッファリングされたデータが单一のまたは複数のメモリ領域に記憶され、異なるレイヤからのデータの分離が別個のレイヤからのデータの記憶位置に対する参照を含むデータ構造の使用を通じて論理的に達成される、他の技法によって実装され得るので、以下では、「別個のバッファ」という用語は、別個のレイヤのデータが別々に識別され得る任意の方法を含むものと理解されるべきである。クライアントは、各バッファの占有率に基づいて、各ブロックの個々の

40

50

レイヤに対する要求を出し、たとえば、レイヤは、1つのレイヤからのデータに対する要求が、優先順位のより低いレイヤに対する任意のバッファの占有率がそのより低いレイヤに対する閾値より低い場合に出され得ないように、優先順序で並べられ得る。この方法では、より優先順位の高いレイヤを受信することも要求される帯域幅を利用可能な帯域幅が下回る場合、より低いレイヤのみが要求されるように、優先権が、より優先順位の低いレイヤからデータを受信することに対して与えられる。さらに、たとえば、より低いレイヤがより高い閾値を有するように、異なるレイヤと関連付けられる閾値は異なり得る。より高いレイヤに対するデータがブロックの再生時間の前に受信され得ないように、利用可能な帯域幅が変化する場合、より低いレイヤに対するデータは、必然的にすでに受信されているであろうから、プレゼンテーションはより低いレイヤ単独で継続し得る。バッファ占有率の閾値は、データのバイト、バッファに含まれるデータの再生継続時間、ブロックの数、または任意の他の適切な尺度に関して定義され得る。

#### 【0384】

第3の実施形態では、第1および第2の実施形態の方法は、レイヤのサブセットを各々含む複数のメディアプレゼンテーションが提供されるように(第1の実施形態のように)、かつ、第2の実施形態が表現内のレイヤのサブセットに適用されるように、組み合わされ得る。

#### 【0385】

第4の実施形態では、第1の、第2の、および/または第3の実施形態の方法は、たとえば、独立した表現の少なくとも1つが、第1の、第2の、および/または第3の実施形態の技法が適用される複数のレイヤを含むように、コンテンツの複数の独立した表現が提供される実施形態と組み合わされ得る。

#### 【0386】

##### 進化したバッファマネージャ

バッファモニタ126(図2参照)と組み合わせて、進化したバッファマネージャが、クライアント側バッファを最適化するために使用され得る。ブロック要求ストリーミングシステムは、メディア再生が迅速に開始でき円滑に継続でき、一方、最大限のメディア品質をユーザまたは宛先デバイスに同時に提供できることを、確実にするのを望み得る。これは、最高のメディア品質を有するがプレゼンテーションの一時休止を強いることなく再生されるように迅速に開始されその後で時間内に受信されることも可能であるブロックを、クライアントが要求することを必要とし得る。

#### 【0387】

進化したバッファマネージャを使用する実施形態では、マネージャは、メディアデータのどのブロックを要求するか、および、それらの要求をいつ行うかを決定する。たとえば、進化したバッファマネージャは、提示されるべきコンテンツに対するメタデータのセットを与えられてよく、このメタデータは、コンテンツに対して利用可能な表現、および各表現に対するメタデータのリストを含む。表現に対するメタデータは、表現のデータレートおよび他のパラメータ、たとえば、ビデオ、オーディオ、または他のコーデックおよびコーデックパラメータ、ビデオ解像度、復号の複雑さ、オーディオの言語、ならびに、クライアントにおける表現の選択に影響を与える任意の他のパラメータについての情報を含み得る。

#### 【0388】

表現に対するメタデータはまた、表現がセグメント化された先のブロックの識別子を含んでよく、これらの識別子は、ブロックを要求するためにクライアントに対して必要とされる情報を提供する。たとえば、要求プロトコルがHTTPである場合、識別子は、場合によっては、URLによって識別されるファイル内のバイト範囲またはタイムスパンを特定する追加の情報を伴う、HTTL URLであってよく、このバイト範囲またはタイムスパンは、URLによって識別されるファイル内の特定のブロックを特定する。

#### 【0389】

ある特定の実装形態では、進化したバッファマネージャは、受信機が新たなブロックに

10

20

30

40

50

に対する要求をいつ行うかを決定し、それ自体が要求の送信を処理し得る。新規の態様では、進化したバッファマネージャは、多くの帯域幅を使いすぎることと、ストリーミング再生の間にメディアを使い果たすこととのバランスをとる、バランスング比率の値に従って、新たなブロックに対する要求を行う。

#### 【0390】

ブロックバッファ125からのバッファモニタ126によって受信される情報は、各イベントの指示、メディアデータがいつ受信されるか、どれだけの量が受信されたか、メディアデータの再生がいつ開始され停止されたか、およびメディア再生の速度を含み得る。この情報に基づいて、バッファモニタ126は、現在のバッファサイズを表す変数 $B_{current}$ を計算し得る。これらの例では、 $B_{current}$ は、クライアントまたは他のデバイスの1つまたは複数のバッファに含まれるメディアの量を表し、追加のブロックまたは部分的なブロックが受信されていなければ1つまたは複数のバッファに記憶されるブロックまたは部分的なブロックによって表されるメディアのすべての再生にかかるであろう時間を $B_{current}$ が表すように、 $B_{current}$ は時間の単位で評価され得る。したがって、 $B_{current}$ は、まだ再生されていないものではなく、クライアントにおいて利用可能なメディアデータの、通常の再生速度における「再生継続時間」を表す。10

#### 【0391】

時間が経過するにつれて、 $B_{current}$ の値は、メディアが再生されるにつれて減少し、ブロックに対する新たなデータが受信されるたびに増加し得る。この説明では、そのブロックのデータ全体がブロック要求器124において利用可能なときにブロックが受信されると仮定されるが、代わりに他の尺度が、たとえば部分的なブロックの受信を考慮するために使用され得ることに留意されたい。実際には、ブロックの受信は、ある期間にわたって起こり得る。20

#### 【0392】

図13は、メディアが再生されブロックが受信されるにつれて、 $B_{current}$ の値の経時的な変動を示す。図13に示されるように、 $B_{current}$ の値は、 $t_0$ 未満の時間では0であり、データが受信されていないことを示す。 $t_0$ において、最初のブロックが受信され、 $B_{current}$ の値は受信されたブロックの再生継続時間と等しくなるように増加する。この時点で、再生はまだ開始されていないので、 $B_{current}$ の値は $t_1$ まで不变のままであり、 $t_1$ において、2番目のブロックが到達し、 $B_{current}$ がこの2番目のブロックのサイズの分だけ増える。この時点で、再生が開始し、 $B_{current}$ の値は $t_2$ まで線形に減少し始め、 $t_2$ において3番目のブロックが到達する。30

#### 【0393】

$B_{current}$ の経過は、この「のこぎり波」の方式で続き、ブロックが受信されるたびにステップ状に増加し( $t_2$ 、 $t_3$ 、 $t_4$ 、 $t_5$ および $t_6$ において)、それらの時間の間で、データが再生されるにつれて滑らかに減少する。この例では、再生は、コンテンツに対する通常の再生レートで進行するので、ブロック受信の間の曲線の傾きはちょうど-1であり、1秒のメディアデータが、経過するそれぞれの1秒の実時間の間に再生されることを意味する。フレームベースのメディアが、毎秒所与の数のフレーム、たとえば毎秒24フレーム再生されると、-1という傾きは、データの各々の個々のフレームの再生を示す小さなステップ関数、たとえば、各フレームが再生されるときに1秒の-1/24のステップによって近似される。40

#### 【0394】

図14は、 $B_{current}$ の経時的な展開の別の例を示す。その例では、最初のブロックは $t_0$ に到達し、再生は直ちに開始する。 $B_{current}$ の値が0に達する時間 $t_3$ まで、ブロックの到達および再生が続く。これが起きると、再生のためのさらなるメディアデータが利用可能ではなく、メディアプレゼンテーションの一時停止が強いられる。時間 $t_4$ において、4番目のブロックが受信され、再生が再開し得る。したがって、この例は、4番目のブロックの受信が所望されるよりも遅く、再生の一時停止と、したがって悪いユーザ体験とをもたらした事例を示す。したがって、進化したバッファマネージャおよび他の特徴の目標は、このイベントの確率を下げつつ、高いメディア品質を同時に維持することである。50

## 【0395】

バッファモニタ126はまた、別の尺度 $B_{ratio}(t)$ を計算することができ、これは、期間の長さに対する所与の期間において受信されたメディアの比である。より具体的には、 $B_{ratio}(t)$ は $T_{received}/(T_{now}-t)$ に等しく、ここで $T_{received}$ は、現在の時間よりも早い何らかの時間 $t$ から現在の時間 $T_{now}$ までの期間に受信されたメディアの量(その再生時間によって評価される)である。

## 【0396】

$B_{ratio}(t)$ は、 $B_{current}$ の変化率を評価するために使用され得る。 $B_{ratio}(t)=0$ は、時間 $t$ からデータが受信されていない場合であり、メディアが再生していると仮定すると、 $B_{current}$ は、その時間から $(T_{now}-t)$ だけ減少するであろう。 $B_{ratio}(t)=1$ は、時間 $(T_{now}-t)$ の間、メディアが再生されているのと同じ量だけメディアが受信される場合であり、 $B_{current}$ は、時間 $t$ において時間 $T_{now}$ における値と同じ値を有するであろう。 $B_{ratio}(t)>1$ は、時間 $(T_{now}-t)$ の間、再生するのに必要なデータよりも多くのデータが受信された例であり、 $B_{current}$ は、時間 $t$ から時間 $T_{now}$ まで増加する。

10

## 【0397】

バッファモニタ126はさらに、値Stateを計算し、これは個別の値の数をとり得る。バッファモニタ126はさらに、関数NewState( $B_{current}, B_{ratio}$ )を備え、これは、 $t < T_{now}$ に対して $B_{current}$ の現在の値および $B_{ratio}$ の値が与えられると、新たなState値を出力として提供する。 $B_{current}$ および $B_{ratio}$ がこの関数にStateの現在の値とは異なる値を返させる場合は常に、新たな値がStateに割り当てられ、この新たなState値がブロック選択器123に示される。

20

## 【0398】

関数NewStateは、ペア( $B_{current}, B_{ratio}(T_{now}-T_x)$ )のすべてのあり得る値の空間への参照によって評価されてよく、ここで $T_x$ は固定された(設定された)値であってよく、または、たとえば $B_{current}$ の値から $T_x$ の値にマッピングする構成テーブルによって $B_{current}$ から導出されてよく、または、Stateの以前の値に依存してよい。バッファモニタ126は、この空間の1つまたは複数の区分を与えられ、各区分はつながっていない領域のセットを含み、各領域はState値によって注記される。関数NewStateの評価は次いで、区分を特定しペア( $B_{current}, B_{ratio}(T_{now}-T_x)$ )が入る領域を決定する動作を含む。そして戻り値は、その領域と関連付けられる注記である。単純な場合には、1つの区分のみが与えられる。より複雑な場合には、区分は、NewState関数の以前の評価の時点におけるペア( $B_{current}, B_{ratio}(T_{now}-T_x)$ )に、または他の要因に依存し得る。

30

## 【0399】

特定の実施形態では、上で説明された区分は、多数の $B_{current}$ の閾値および多数の $B_{ratio}$ の閾値を含む構成テーブルに基づき得る。具体的には、 $B_{current}$ の閾値を、 $B_{thresh}(0)=0$ 、 $B_{thresh}(1)、…、B_{thresh}(n_1)、B_{thresh}(n_1+1)=$  とし、ここで $n_1$ は、 $B_{current}$ の0ではない閾値の数である。 $B_{ratio}$ の閾値を、 $B_{r-thresh}(0)=0、B_{r-thresh}(1)、…、B_{r-thresh}(n_2)、B_{r-thresh}(n_2+1)=$  とし、ここで $n_2$ は、 $B_{ratio}$ の閾値の数である。これらの閾値は、セルの $(n_1+1)$ 対 $(n_2+1)$ のグリッドを含む区分を定義し、ここで、 $j$ 番目の行の $i$ 番目のセルは、 $B_{thresh}(i-1) \leq B_{current} < B_{thresh}(i)$ 、かつ $B_{r-thresh}(j-1) \leq B_{ratio} < B_{r-thresh}(j)$ である領域に対応する。上で説明されたグリッドの各セルは、たとえば、メモリに記憶された特定の値と関連付けられることによって、state値によって注記され、そして関数NewStateは、値 $B_{current}$ および $B_{ratio}(T_{now}-T_x)$ によって示されるセルと関連付けられるstate値を返す。

40

## 【0400】

さらなる実施形態では、ヒステリシス値が、各閾値と関連付けられ得る。この改善された方法では、関数NewStateの評価は、次のように、時間的に修正される閾値のセットを使用して構築される時間的な区分に基づき得る。NewStateの最後の評価で選ばれたセルに対応する $B_{current}$ の範囲よりも小さい各 $B_{current}$ の閾値に対して、閾値は、その閾値と関連付けられるヒステリシス値を減算することによって減らされる。NewStateの最後の評価で

50

選ばれたセルに対応する $B_{current}$ の範囲よりも大きい各 $B_{current}$ の閾値に対して、閾値は、その閾値と関連付けられるヒステリシス値を換算することによって増やされる。NewStateの最後の評価で選ばれたセルに対応する $B_{ratio}$ の範囲よりも小さい各 $B_{ratio}$ の閾値に対して、閾値は、その閾値と関連付けられるヒステリシス値を減算することによって減らされる。NewStateの最後の評価で選ばれたセルに対応する $B_{ratio}$ の範囲よりも大きい各 $B_{ratio}$ の閾値に対して、閾値は、その閾値と関連付けられるヒステリシス値を加算することによって増やされる。修正された閾値は、NewStateの値を評価するために使用され、次いで閾値は元の値に戻される。

#### 【0401】

空間の区分を定義する他の方法は、本開示を読んだ当業者には明らかであろう。たとえば、空間全体の中の半空間を定義し、多数のそのような半空間の交差部分として各々のつながっていないセットを定義するために、区分は、 $B_{ratio}$ と $B_{current}$ の線形結合に基づく不等式、たとえば、実数値 0、1、および 2に対して  $1 \cdot B_{ratio} + 2 \cdot B_{current} < 0$  の形式の線形の不等式の閾値の使用によって定義され得る。10

#### 【0402】

上の説明は、基本的なプロセスについて説明するものである。本開示を読んだリアルタイムプログラミングの当業者には明らかなように、効率的な実装が可能である。たとえば、新たな情報がバッファモニタ126に与えられるたびに、たとえばロックに対するさらなるデータが受信されない場合、NewStateが新たな値に移行する未来の時間を計算することが可能である。次いで、タイマーがこの時間に設定され、さらなる入力がないと、このタイマーの満了により、新たなState値がロック選択器123に送信される。結果として、計算は、連続的にではなく、新たな情報がバッファモニタ126に提供されたとき、または、タイマーが満了したときにのみ実行される必要がある。20

#### 【0403】

Stateの適切な値は、「Low」、「Stable」、および「Full」である。閾値の適切なセットおよび得られるセルグリッドの例が、図15に示される。

#### 【0404】

図15において、 $B_{current}$ の閾値は、「+/-値」として下に示されるヒステリシス値とともに、ミリ秒単位で水平軸上に示されている。 $B_{ratio}$ の閾値は、「+/-値」として下に示されるヒステリシス値とともに、パーセント単位(すなわち、1000を乗算された)で垂直軸上に示されている。State値は、「Low」、「Stable」、および「Full」に対してそれぞれ「L」、「S」、および「F」として、グリッドセルへと注記される。30

#### 【0405】

ロック選択器123は、新たなロックを要求する機会があるときは常に、ロック要求器124から通知を受信する。上で説明されたように、ロック選択器123は、たとえば各ロックのメディアデータレートについての情報を含む、利用可能な複数のロックおよびそれらのロックのためのメタデータに関する情報を与えられる。

#### 【0406】

ロックのメディアデータレートについての情報は、特定のロックの実際のメディアデータレート(すなわち、バイト単位のロックサイズを秒単位の再生時間で割ったもの)、ロックが属する表現の平均メディアデータレート、または、ロックが属する表現を、一時休止を伴わずに持続的に再生するために必要とされる利用可能な帯域幅の尺度、または上記の組合せを含み得る。40

#### 【0407】

ロック選択器123は、バッファモニタ126によって最後に示されたState値に基づいてロックを選択する。このState値が「Stable」である場合、ロック選択器123は、以前の選択されたロックと同じ表現からロックを選択する。選択されるロックは、それに対するメディアデータがこれまで要求されていない表現の中の、ある期間に対するメディアデータを含む(再生の順序で)最初のロックである。

#### 【0408】

50

20

30

40

50

State値が「Low」である場合、ブロック選択器123は、以前に選択されたブロックよりもメディアデータレートが低い表現からブロックを選択する。多数の要因が、この場合の表現の正確な選択に影響を与える。たとえば、ブロック選択器123は、入来データの統合レートを示すものを与えられてよく、その値よりも低いメディアデータレートを伴う表現を選ぶことができる。

#### 【0409】

State値が「Full」である場合、ブロック選択器123は、以前に選択されたブロックよりもメディアデータレートが高い表現からブロックを選択する。多数の要因が、この場合の表現の正確な選択に影響を与える。たとえば、ブロック選択器123は、入来データの統合レートを示すものを与えられてよく、その値よりも高くないメディアデータレートを伴う表現を選ぶことができる。10

#### 【0410】

多数の追加の要因がさらに、ブロック選択器123の動作に影響を与える。具体的には、バッファモニタ126が「Full」状態を示し続けたとしても、選択されたブロックのメディアデータレートが上がる頻度が制限され得る。さらに、ブロック選択器123が「Full」状態の指示を受信するが、メディアデータレートのより高いブロックが利用可能ではない(たとえば、最後に選択されたブロックがすでに、利用可能な最高のメディアデータレートであったため)ことがあり得る。この場合、ブロック選択器123は、ブロックバッファ125においてバッファリングされたメディアデータの総量が上から抑えられるように、選ばれた時間だけ、次のブロックの選択を遅らせることができる。20

#### 【0411】

選択プロセスの間に考慮される追加の要因が、ブロックのセットに影響を与える。たとえば、利用可能なブロックは、ブロック選択器123に与えられた特定の範囲内に符号化の分解能が入る表現からのブロックに限定され得る。

#### 【0412】

ブロック選択器123はまた、メディア復号のための計算リソースの利用可能性のような、システムの他の様子を監視する他のコンポーネントから入力を受け取ることができる。そのようなリソースが乏しくなると、ブロック選択器123は、復号がメタデータ内の計算の複雑さに関してより低いと示されるブロック(たとえば、分解能またはフレームレートがより低い表現は一般に、復号の複雑さがより低い)を選ぶことができる。30

#### 【0413】

上で説明された実施形態は、バッファモニタ126内のNewState関数の評価における値 $B_{ratio}$ の使用が、 $B_{current}$ のみを考慮する方法と比較して、プレゼンテーションの開始のときの品質のより高速な向上を可能にするという、かなりの利益をもたらす。 $B_{ratio}$ を考慮しなければ、メディアデータレートがより高い、したがって品質がより高いブロックをシステムが選択できるようになる前に、大量のバッファリングされたデータが蓄積され得る。しかしながら、 $B_{ratio}$ 値が大きいと、これは、利用可能な帯域幅が、以前に受信されたブロックのメディアデータレートよりもはるかに大きいこと、および、バッファリングされたデータが比較的少なくてても(すなわち、 $B_{current}$ に対する小さな値)、メディアデータレートがより高い、したがって品質がより高いブロックを要求することが安全なままであることを示す。同様に、 $B_{ratio}$ 値が低い( $<1$ 、たとえば)と、これは、利用可能な帯域幅が、以前に要求されたブロックのメディアデータレートを下回ったこと、および、したがって、 $B_{current}$ が高くて、たとえば、 $B_{current}=0$ となる点に達しメディアの再生がストールするのを避けるために、より低いメディアデータレート、したがってより低い品質へとシステムが切り替えることを示す。この改善された挙動は、ネットワーク条件、および、したがって配信速度が非常に高速かつ動的に変化し得る環境、たとえば、モバイルデバイスへのユーザのストリーミングにおいて、特に重要であり得る。40

#### 【0414】

別の利点は、構成データを使用して( $B_{current}, B_{ratio}$ )の値の空間の区分を規定することによりもたらされる。そのような構成データは、プレゼンテーションメタデータの一部50

として、または他の動的な手段によって、バッファモニタ126に提供され得る。実際の展開では、ユーザネットワーク接続の挙動は、ユーザの間で、および単一のユーザに対しても時間とともに大きく変動し得るので、すべてのユーザに対して良好に動作する区分を予測するのは困難であり得る。そのような構成情報をユーザへ動的に提供できることで、蓄積された経験に従って、良好な構成設定が時間とともに発展することが可能になる。

#### 【0415】

##### 可変の要求サイジング

各要求が単一のブロックに対するものであり、各ブロックが短いメディアセグメントを符号化する場合、高頻度の要求が必要とされ得る。メディアブロックが短い場合、ビデオ再生はブロックからブロックへと高速に移動し、このことは、表現を変更することによって、選択されたデータレートを調整または変更するためのより頻繁な機会を受信機に与え、再生がストールすることなく継続できる確率を高める。しかしながら、高頻度の要求の欠点は、サーバネットワークに対する利用可能な帯域幅がクライアントにおいて制約されるようのある種のネットワーク、たとえば、クライアントからネットワークへのデータリンクの容量が限られている、または、無線条件の変化により短期間または長期間限られるようになり得る、3Gおよび4GワイヤレスWANのようなワイヤレスWANネットワークでは、持続可能ではないことがあるということである。

10

#### 【0416】

高頻度の要求はまた、サービングインフラストラクチャに対する高い負荷を意味し、これは、容量の要件について関連するコストをもたらす。したがって、欠点のすべてを伴わずに、高頻度の要求の利点のいくつかを有することが望ましい。

20

#### 【0417】

ブロックストリーミングシステムのいくつかの実施形態では、高頻度の要求の柔軟性が、より低頻度の要求と組み合わされる。この実施形態では、ブロックは、上で説明されたように構築され、やはり上で説明されたように、複数のブロックを含むセグメントへと統合され得る。プレゼンテーションの始めにおいて、各要求が単一のブロックを参照する、または、複数の同時の要求がブロックの一部を要求するために行われる、上で説明されたプロセスが、高速なチャネルザッピング時間、およびしたがって、プレゼンテーションの始めにおける良好なユーザ体験を確実にするために適用される。続いて、以下で説明されるある条件が満たされると、クライアントは、単一の要求に複数のブロックを包含する要求を出すことができる。これは、ブロックがより大きなファイルまたはセグメントへと統合され、バイトまたは時間の範囲を使用して要求され得るので、可能になる。連続的なバイトまたは時間の範囲は、単一のより大きなバイトまたは時間の範囲へと統合されてよく、複数のブロックに対する単一の要求をもたらし、不連続なブロックも、1つの要求で要求され得る。

30

#### 【0418】

単一のブロック(または部分的なブロック)を要求するか、または複数の連続するブロックを要求するかを判断することによって行われ得る1つの基本的な構成は、要求されるブロックが再生される可能性が高いかどうかの判断に基づく構成を有する。たとえば、別の表現へとまもなく変更する必要がある可能性が高い場合、クライアントが単一のブロック、すなわち少量のメディアデータに対する要求を行うことがより好ましい。このことの1つの理由は、別の表現への切り替えが差し迫っている可能性があるときに複数のブロックに対する要求が行われると、要求の最後の数ブロックが再生される前に切り替えが行われ得るからである。したがって、これらの最後の数ブロックのダウンロードは、切替先の表現のメディアデータの配信を遅らせることがあり、これは、メディア再生のストールを引き起こし得る。

40

#### 【0419】

しかしながら、単一のブロックに対する要求は、高頻度の要求をもたらす。一方、別の表現へとまもなく変更する必要がある可能性が高い場合、これらのブロックのすべてが再生される可能性が高いので、複数のブロックに対する要求を行うことが好ましい可能性が

50

あり、これは低頻度の要求をもたらし、表現の差し迫った変更がないことが一般的である場合は特に、要求のオーバーヘッドをかなり下げ得る。

#### 【0420】

従来のブロック統合システムでは、各要求において要求される量は動的に調整されず、すなわち、通常は各要求がファイル全体に対するものであり、または、各要求がほぼ同じ量の表現のファイル(場合によっては時間で評価され、場合によってはバイトで評価される)に対するものである。したがって、すべての要求がより小さければ、要求のオーバーヘッドは大きく、一方すべての要求がより大きければ、このことはメディアストールのイベントの確率を上げ、かつ/または、ネットワーク条件が変化するにつれて高速に表現を変更する必要性を避けるためにより低品質の表現が選ばれる場合、より低品質なメディア再生をもたらす。10

#### 【0421】

満たされると、後続の要求に複数のブロックを参照させ得る条件の例は、バッファサイズ $B_{current}$ の閾値である。 $B_{current}$ が閾値を下回る場合、出された各要求は単一のブロックを参照する。 $B_{current}$ が閾値以上である場合、出された各要求は複数のブロックを参照する。複数のブロックを参照する要求が出される場合、各々の単一の要求において要求されるブロックの数は、いくつかの可能な方法の1つで決定され得る。たとえば、その数は定数、たとえば2であり得る。あるいは、単一の要求において要求されるブロックの数は、バッファ状態に、具体的には $B_{current}$ に依存し得る。たとえば、閾値の数が設定されてよく、単一の要求において要求されるブロックの数は、 $B_{current}$ 未満の複数の閾値の最大値から導出される。20

#### 【0422】

満たされると、要求に複数のブロックを参照させ得る条件の別の例は、上で説明された変数Stateの値である。たとえば、Stateが「Stable」または「Full」であるとき、要求は複数のブロックに対して出され得るが、Stateが「Low」であるとき、すべての要求が1つのブロックに対するものであり得る。

#### 【0423】

別の実施形態が図16に示される。この実施形態では、次の要求が出されるべきである場合(ステップ1300で判定される)、現在のStateの値および $B_{current}$ が、次の要求のサイズを決定するために使用される。現在のStateの値が「Low」である場合、または現在のStateの値が「Full」であり現在の表現が利用可能な最高のものではない場合(ステップ1310で判定され、答えは「Yes」)、次の要求は、短くなるように、たとえば、次のブロック(ステップ1320で決定され要求が行われるブロック)のためのものになるように選ばれる。このことの根拠は、これらが、まもなく表現の変化がある可能性が高い場合の条件である、ということである。現在のStateの値が「Stable」である場合、または、現在のStateの値が「Full」であり現在の表現が利用可能な最高のものである場合(ステップ1310で判定され、答えは「No」)、次の要求において要求される連続的なブロックの継続時間が、何らかの固定された  $<1$  に対して  $B_{current}$  の  $<1$  部分に比例するように選ばれ(ステップ1330で判定され、ステップ1340で要求が行われるブロック)、たとえば、 $=0.4$  に対して、 $B_{current}=5$  秒である場合、次の要求は約2秒のブロックに対するものであってよく、 $B_{current}=10$  秒である場合、次の要求は約4秒のブロックに対するものであってよい。このことの1つの根拠は、これらの条件では、 $B_{current}$  に比例する時間に対して、新たな表現への切り替えが行われる可能性が低い可能性がある、ということである。3040

#### 【0424】

#### 柔軟なパイプライン化

ブロックストリーミングシステムは、特定の背後にあるトランスポートプロトコル、たとえば、TCP/IPを有する、ファイル要求プロトコルを使用し得る。TCP/IPまたは他のトランスポートプロトコル接続の始めにおいて、利用可能な帯域幅の完全な利用を達成するにはかなりの時間がかかり得る。これは、新たな接続が開始されるたびに「接続始動ペナルティ」をもたらし得る。たとえば、TCP/IPの場合、接続開始ペナルティは、接続を確立す50

るための初期のTCPハンドシェイクにかかる時間と、混雑制御プロトコルが利用可能な帯域幅の完全な利用を達成するのにかかる時間の両方によって、発生する。

#### 【0425】

この場合、接続始動ペナルティが引き起こされる頻度を下げるために、単一の接続を使用して複数の要求を出すことが望ましいことがある。しかしながら、いくつかのファイルトランSPORTプロトコル、たとえばHTTPは、トランSPORT層接続全体を切断すること以外に、要求を取り消すための機構を提供しないので、新たな接続が古い接続の代わりに確立されるときに、接続始動ペナルティを引き起こす。出される要求は、利用可能な帯域幅が変化し、異なるメディアデータレートが代わりに要求されると判定されると取り消される必要があることがあり、すなわち、異なる表現へと切り替えるための判断がある。出された要求を取り消すための別の理由は、メディアプレゼンテーションが終了され(場合によっては、プレゼンテーション中の異なる点における同じコンテンツアイテムの、または場合によっては、新たなコンテンツアイテムの)新たなプレゼンテーションが開始されることをユーザが要求したかどうかであり得る。10

#### 【0426】

知られているように、接続開始ペナルティは、接続を開放したままにすること、および、後続の要求に対して同じ接続を再使用することによって、回避されることが可能であり、やはり知られているように、複数の要求が同じ接続において同時に出現する場合、接続は完全に利用される状態を保たれ得る(HTTPの状況では「パイプライン化」として知られる技法)。しかしながら、複数の要求を同時に出すこと、またはより一般的には、前の要求がある接続を通じて完了する前に複数の要求が出されるような方式で出すことの欠点は、接続がそれらの要求に対する応答を搬送することに費やされるので、それに対して要求が出されるべき変更が望ましいものになると、もはや所望されないすでに出された要求を取り消すことが必要になる場合、接続が切断され得るということであり得る。20

#### 【0427】

出された要求が取り消される必要のある確率は、以下のような意味で、要求の提出と要求されたブロックの再生時間との時間間隔の長さに一部依存し得る。その意味とは、この時間間隔が長い場合、出された要求が取り消される必要のある確率も高い(利用可能な帯域幅がその間隔の間に変化する可能性が高いので)というものである。

#### 【0428】

知られているように、一部のファイルダウンロードプロトコルは、単一の背後にあるトランSPORT層接続が有利なことに複数のダウンロード要求のために使用され得る、という性質を有する。たとえば、複数の要求に対する単一の接続の再使用は、最初のもの以外の、要求について上で説明された「接続始動ペナルティ」を避けるので、HTTPは上記の性質を有する。しかしながら、この手法の欠点は、接続が各々の出された要求において要求されるデータをトランSPORTすることに費やされるので、1つまたは複数の要求が取り消される必要がある場合、接続が切断され、代わりの接続が確立されるときに接続始動ペナルティを引き起こし得るか、または、クライアントがもはや必要とされないデータの受信を待機し、後続のデータの受信に遅延を引き起こし得るかのいずれかである、ということである。40

#### 【0429】

ここで、この欠点を引き起こすことなく接続の再使用の利点を保持し、また、接続が再使用され得る頻度をさらに向上させる、実施形態について説明する。

#### 【0430】

本明細書で説明されるブロックストリーミングシステムの実施形態は、始めの接続を特定の要求のセットに費やす必要なく、複数の要求のために接続を再使用するように構成される。基本的に、ある既存の接続ですでに出された要求がまだ完了していないが、完了に近い場合、新たな要求がその接続で出される。既存の要求が完了するまで待機しないことの1つの理由は、以前の要求が完了すると、接続速度が下がることがあり、すなわち、背後にあるTCPセッションがアイドル状態になることがあり、またはTCPのcwnd変数がかなり50

下がることがあり、したがって、その接続の上で出された新たな要求の初期ダウンロード速度がかなり下がるからである。追加の要求を出す前に完了の近くまで待機することの1つの理由は、前の要求が完了するかなり前に新たな要求が出されると、新たに出された要求がかなりの期間開始すらしないことがあり、新たに出された要求が開始する前のこの期間の間に、たとえば表現を切り替えるという判断が原因で、新たな要求を行うという判断がもはや有効ではないことがあり得るからである。したがって、この技法を実施するクライアントの実施形態は、接続のダウンロード能力を低速化することなく、可能な限り遅く、その接続で新たな要求を出す。

#### 【0431】

方法は、ある接続で出された最新の要求に応答して、この接続で受信されるバイトの数を監視するステップと、試験をこの数に適用するステップとを含む。これは、監視し試験するように受信機(または、可能な場合送信機)を構成させることによって行われ得る。

#### 【0432】

試験に通ると、さらなる要求がその接続で出され得る。適切な試験の1つの例は、受信されたバイトの数が、要求されたデータのサイズの固定された断片より大きいかどうかである。たとえば、この断片は80%であり得る。適切な試験の別の例は、図17において示されるように、次の計算に基づく。計算において、Rを接続のデータレートの推定値とし、Tをラウンドトリップタイム('RTT')の推定値とし、Xをたとえば0.5から2の間の値に設定される定数であり得る係数とし、RおよびTの推定値は定期的に更新される(ステップ1410で更新される)。Sを最後の要求において要求されたデータのサイズとし、Bを受信される要求されたデータのバイトの数とする(ステップ1420で計算される)。

#### 【0433】

適切な試験は、受信機(または可能な場合、送信機)に、不等式( $S-B < X \cdot R \cdot T$ (ステップ1430で試験される))を評価するためのルーチンを実行させることであり、「Yes」の場合、動作を行う。たとえば、その接続で出される準備ができている別の要求があるかどうかを確認するための試験が行われてよく(ステップ1440で試験される)、「Yes」の場合、その接続に対してその要求を出し(ステップ1450)、「No」の場合、プロセスはステップ1410に戻り、更新および試験を続ける。ステップ1430の試験の結果が「No」である場合、プロセスはステップ1410に戻り、更新および試験を続ける。

#### 【0434】

ステップ1430の不等式の試験(たとえば、適切にプログラムされた要素によって実行される)は、受信されるべき残りのデータの量が、Xと、1つのRTT内の現在の推定される受信レートで受信され得るデータの量とを乗じたものと等しいときに、各々の後続の要求が出されるようになる。ステップ1410のデータレートRを推定するための多数の方法が、当技術分野で知られている。たとえば、データレートは $Dt/t$ として推定されてよく、ここで $Dt$ は、先行する $t$ 秒の間に受信されたビットの数であり、 $t$ はたとえば、1秒または0.5秒または何らかの他の期間であり得る。別のある方法は、指標加重平均、または、入来するデータレートの1次無限インパルス応答(IIR)フィルタである。ステップ1410のRTT、Tを推定するための多数の方法が、当技術分野で知られている。

#### 【0435】

ステップ1430の試験は、以下でより詳細に説明されるように、あるインターフェース上のすべてのアクティブな接続の統合に対して適用され得る。

#### 【0436】

方法はさらに、候補要求のリストを構築するステップと、各候補要求を要求が行われ得る対象の適切なサーバのセットと関連付けるステップと、優先順序で候補要求のリストを並べるステップとを含む。候補要求のリスト中のいくつかのエントリーは、同じ優先順位を有し得る。各候補要求と関連付けられる適切なサーバのリスト中のサーバは、ホスト名によって特定される。各ホスト名は、よく知られているように、ドメインネームシステムから取得され得るインターネットプロトコルアドレスのセットに対応する。したがって、候補要求のリストに対する各々のあり得る要求は、インターネットプロトコルアドレスの

10

20

30

40

50

セット、具体的には、候補要求と関連付けられるサーバと関連付けられるホスト名と関連付けられるインターネットプロトコルアドレスのセットの結合物と関連付けられる。ステップ1430で説明される試験がある接続に対して満たされ、その接続で新たな要求がまだ出されていない場合は常に、その接続の宛先のインターネットプロトコルアドレスと関連付けられる候補要求のリスト上の最高の優先順位の要求が選ばれ、この要求はその接続で出される。その要求はまた、候補要求のリストから除去される。

#### 【0437】

候補要求は、候補要求のリストから除去(削除)されてよく、新たな要求が、候補リスト上の既存の要求よりも高い優先順位を伴う候補リストに追加されてよく、候補リスト上の既存の要求は、優先順位を変更されてよい。どの要求が候補要求のリスト上にあるかということの動的な性質、および、候補リスト上の要求の優先順位の動的な性質により、ステップ1430で説明されたタイプの試験がいつ満たされるかに応じて、どの要求が次に出され得るかが変化し得る。10

#### 【0438】

たとえば、ある時間tにおいて、ステップ1430で説明された試験に対する答えが「Yes」である場合、出される次の要求が要求Aであり、一方、ステップ1430で説明された試験に対する答えが何らかの時間t'>tまで「Yes」ではない場合、出される次の要求は代わりに要求Bである、ということがあり得る。それは、要求Aが時間tとt'の間に候補要求のリストから除去されたため、または、要求Bが時間tとt'の間に要求Aよりも高い優先順位を伴って候補要求のリストに追加されたため、または、要求Bが時間tにおいて候補リスト上にあったが要求Aよりも優先順位が低く、時間tとt'の間に要求Bの優先順位が要求Aの優先順位よりも高くされたためである。20

#### 【0439】

図18は、要求の候補リスト上の要求のリストの例を示す。この例では、3個の接続があり、A、B、C、D、E、およびFと標識された候補リスト上の6個の要求がある。候補リスト上の要求の各々は、示されるように接続のサブセットで出されることが可能であり、たとえば、要求Aは接続1で出されてよく、一方要求Fは接続2または接続3で出されることが可能である。各要求の優先順位も図18において標識されており、より小さな優先順位の値は、要求の優先順位がより高いことを示す。したがって、優先順位0を伴う要求AおよびBは優先順位が最高の要求であり、一方3という優先順位の値を伴う要求Fは候補リスト上の要求の中で最も優先順位が低い。30

#### 【0440】

この時間tの点において、接続1がステップ1430で説明される試験に通ると、要求Aまたは要求Bのいずれかが接続1で出される。代わりに、接続3がこの時間tにおいてステップ1430で説明された試験に通ると、要求Dは接続3で出され得る最高の優先順位を伴う要求なので、要求Dは接続3で出される。

#### 【0441】

すべての接続に対して、時間tからより後の時間t'までのステップ1430で説明された試験に対する答えが「No」であり、時間tとt'の間で要求Aの優先順位が0から5に変化したと仮定すると、要求Bは候補リストから除去され、優先順位0を伴う新たな要求Gが候補リストに追加される。そして、時間t'において、新たな候補リストは図19に示されるようであり得る。40

#### 【0442】

時間t'において、接続1がステップ1430で説明された試験に通ると、優先順位4を伴う要求Cはこの時点において接続1で出され得る候補リスト上の優先順位が最高の要求なので、要求Cは接続1で出される。

#### 【0443】

この同じ状況において、代わりに、要求Aが時間tにおいて接続1で出されたと仮定する(要求Aは、図18に示されるように、時間tにおいて接続1に対する2つの最高の優先順位の選択の1つであった)。すべての接続に対して、時間tからより後の時間t'までのステップ14350

0で説明された試験に対する答えが「No」であり、接続1は依然として、時間tよりも前に出された要求に対するデータを、少なくとも時間t'まで配信しているので、要求Aは少なくとも時間t'まで開始されなかつたであろう。時間t'において要求Cを出すことが、時間tにおいて要求Aを出すことよりも良い判断であり、それは、t'の後で、要求Aが開始されたのと同時に要求Cが開始するから、および、その時点までには、要求Cは要求Aよりも優先順位が高くなっているからである。

#### 【0444】

別の代替形態として、ステップ1430で説明されたタイプの試験がアクティブな接続の統合に適用される場合、そのインターネットプロトコルアドレスが候補要求のリスト上の第1の要求と関連付けられる、または、前記第1の要求と同じ優先順位を伴う別の要求と関連付けられる宛先を有する接続が選ばれ得る。10

#### 【0445】

多数の方法が、候補要求のリストの構築に対して可能である。たとえば、候補リストは、時間的な順序でプレゼンテーションの現在の表現のデータの次のn個の部分に対する要求を表すn個の要求を含んでよく、ここで、データの最も早い部分に対する要求は最高の優先順位を有し、データの最も遅い部分に対する要求は最低の優先順位を有する。いくつかの場合には、nは1であり得る。nの値は、バッファサイズB<sub>current</sub>、または変数State、または、クライアントのバッファ占有率の状態の別の尺度に依存し得る。たとえば、多数の閾値がB<sub>current</sub>および各閾値と関連付けられる値に対して設定されてよく、nの値は、B<sub>current</sub>未満の最高の閾値と関連付けられる値であると考えられる。20

#### 【0446】

上で説明された実施形態は、接続に対する要求の柔軟な割振りを確実にし、(接続の宛先IPアドレスが要求と関連付けられるホスト名のいずれかに割り振られるIPアドレスではなかったために)最高の優先順位の要求が既存の接続に適さない場合でも、その接続を再使用することに対する選好が与えられることを確実にする。B<sub>current</sub>またはStateまたはクライアントのバッファ占有率の状態の別の尺度に対するnの依存性は、時間的順序で再生されるべき次のデータの部分と関連付けられる要求の発行および完了を行う差し迫った必要がクライアントにあるときに、そのような「優先順位が狂った」要求が出されないと確実にする。

#### 【0447】

これらの方法は有利には、協調的なHTTPおよびFECと組み合わされ得る。30

#### 【0448】

##### 一貫性のあるサーバの選択

よく知られているように、ファイルダウンロードプロトコルを使用してダウンロードされるべきファイルは、一般に、ホスト名およびファイル名を含む識別子によって識別される。たとえば、これはHTTPプロトコルに対して当てはまり、この場合、識別子はUniform Resource Identifier(URI)である。ホスト名は、インターネットプロトコルアドレスによって識別される、複数のホストに対応し得る。たとえば、これは、複数の物理的な機械にまたがって、複数のクライアントからの要求の負荷を分散する一般的な方法である。具体的には、この手法は一般的に、コンテンツ配信ネットワーク(CDN)によって行われる。この場合、物理ホストのいずれかに対する接続で出された要求は、成功することが予想される。クライアントがそれを使用してホスト名と関連付けられるインターネットプロトコルアドレスの中から選択できる、多数の方法が知られている。たとえば、これらのアドレスは通常、ドメインネームシステムを介してクライアントに提供され、優先順序で提供される。クライアントは次いで、最高の優先順位の(第1の)インターネットプロトコルアドレスを選ぶことができる。しかしながら、一般に、この選択がどのように行われるかについて、クライアント間での協調は存在せず、その結果、異なるクライアントが異なるサーバから同じファイルを要求することがある。これにより、近くの複数のサーバのキャッシュに同じファイルが記憶されることがあり、これは、キャッシュインフラストラクチャの効率性を下げる。40

**【 0 4 4 9 】**

このことは、同じブロックを要求する2つのクライアントが同じサーバからこのブロックを要求する確率を有利に上げるシステムによって処理され得る。ここで説明される新規の方法は、要求されるべきファイルの識別子によって決定される方式で、かつ、インターネットプロトコルアドレスおよびファイル識別子の同じまたは同様の選択を提示された異なるクライアントが同じ選択を行うような方式で、利用可能なインターネットプロトコルアドレスの中から選択するステップを含む。

**【 0 4 5 0 】**

方法の第1の実施形態が、図20を参照して説明される。クライアントはまず、ステップ1710において示されるように、インターネットプロトコルアドレスのセット $IP_1$ 、 $IP_2$ 、…、 $IP_n$ を取得する。ステップ1720において判断されるように、それに対して要求が出されるべきファイルがある場合、クライアントは、ステップ1730～1770において決定されるように、そのファイルに対する要求をどのインターネットプロトコルアドレスで出すかを決定する。インターネットプロトコルアドレスのセットおよび要求されるべきファイルの識別子が与えられると、方法は、ファイル識別子によって決定される方式で、インターネットプロトコルアドレスを順序付けるステップを含む。たとえば、各インターネットプロトコルアドレスに対して、ステップ1730において示されるように、インターネットプロトコルアドレスおよびファイル識別子の連結物を含む、バイト文字列が構築される。ステップ1740において示されるように、ハッシュ関数がこのバイト文字列に適用され、得られるハッシュ値は、ステップ1750において示されるように、固定された順序、たとえば、数値が増加する順序に従って並べられ、インターネットプロトコルアドレスの順序を生じさせる。同じハッシュ関数がすべてのクライアントにより使用され得るので、すべてのクライアントによる所与の入力に対して同じ結果がハッシュ関数により生成されることが保証される。ハッシュ関数は、クライアントのセット中のすべてのクライアントへと統計的に構成されてよく、または、クライアントのセット中のすべてのクライアントが、クライアントがインターネットプロトコルアドレスのリストを取得したときにハッシュ関数の部分的または完全な記述を取得してよく、または、クライアントのセット中のすべてのクライアントが、クライアントがファイル識別子を取得したときにハッシュ関数の部分的または完全な記述を取得してよく、または、ハッシュ関数は他の手段によって求められてよい。この順序で最初にあるインターネットプロトコルアドレスが選ばれ、このアドレスが次いで、ステップを1760および1770において示されるように、接続を確立し、ファイルのすべてまたは一部に対する要求を出すために使用される。

**【 0 4 5 1 】**

上の方法は、新たな接続がファイルを要求するために確立されるときに適用され得る。これはまた、多数の確立された接続が利用可能であり、これらの1つが新たな要求を出すために選ばれ得る場合に適用され得る。

**【 0 4 5 2 】**

さらに、確立された接続が利用可能であり、要求が優先順位の等しい候補要求のセットの中から選ばれ得る場合、たとえば、上で説明されたハッシュ値の同じ方法によって候補要求の順序が生じ、この順序で最初に現れる候補要求が選ばれる。再び、接続および要求の各々の組合せに対するハッシュを計算し、固定された順序に従ってこれらのハッシュ値を順序付け、要求と接続の組合せのセットにより生じた順序で最初に現れる組合せを選ぶことによって、方法は、優先順位の等しい接続および要求のセットの中から、接続と候補要求の両方を選択するように、組み合わされ得る。

**【 0 4 5 3 】**

この方法は、次のような理由で利点がある。図1(BSI 101)または図2(BSI 101)に示されるような、ブロックサービングインフラストラクチャによって行われる通常の手法、特にCDNによって一般に行われる手法は、クライアント要求を受信する複数のキャッシングプロキシサーバを提供することになる。キャッシングプロキシサーバは、所与の要求において要求されるファイルを与えられないことがあり、この場合、そのようなサーバは通常

10

20

30

40

50

、要求を別のサーバに転送し、通常は要求されたファイルを含む応答をそのサーバから受信し、その応答をクライアントに転送する。キャッシングプロキシサーバはまた、ファイルに対する後続の要求に直ちに応答できるように、要求されたファイルを記憶(キャッシュ)することができる。上で説明された一般的な手法は、所与のキャッシングプロキシサーバに記憶されるファイルのセットが、そのキャッシングプロキシサーバが受信した要求のセットによって大きく左右されるという性質を有する。

#### 【0454】

上で説明された方法は、次の利点を有する。クライアントのセット中のすべてのクライアントがインターネットプロトコルアドレスの同じリストを与えられると、これらのクライアントは、同じファイルに対して出されるすべての要求に対して同じインターネットプロトコルを使用する。インターネットプロトコルアドレスの2つの異なるリストがあり、各クライアントがこれらの2つのリストの1つを与えられる場合、クライアントは、同じファイルに対して出されるすべての要求に対して、多くても2つの異なるインターネットプロトコルアドレスを使用する。一般に、クライアントに与えられるインターネットプロトコルアドレスのリストが同様である場合、クライアントは、同じファイルに対して出されるすべての要求に対する与えられたインターネットプロトコルアドレスの小さなセットを使用する。近隣のクライアントはインターネットプロトコルアドレスの同様のリストを与えられる傾向があるので、近隣のクライアントが、それらのクライアントが利用可能であるキャッシングプロキシサーバの小さな部分のみからのファイルに対する要求を出す可能性が高い。したがって、ファイルをキャッシングするキャッシングプロキシサーバは少数しか存在せず、これは有利なことに、ファイルをキャッシングするために使用されるキャッシングリソースの量を最小限にする。

#### 【0455】

好ましくは、ハッシュ関数は、様々な入力の非常に小さな部分が同じ出力にマッピングされ、様々な入力が基本的にランダムな出力にマッピングされ、インターネットプロトコルアドレスの所与のセットに対して、インターネットプロトコルアドレスの所与の1つがステップ1750で生成されたソートされたリストにおいて最初にあるファイルの比率は、リスト中のすべてのインターネットプロトコルアドレスに対してもほぼ同一であるという性質を有する。一方、所与の入力に対してハッシュ関数の出力がすべてのクライアントに対して同一であるという意味で、ハッシュ関数が決定論的であることは重要である。

#### 【0456】

上で説明された方法の別の利点は、次の通りである。クライアントのセット中のすべてのクライアントが、インターネットプロトコルアドレスの同じリストを与えられると仮定する。上で説明されたハッシュ関数の性質により、これらのクライアントからの異なるファイルに対する要求は、インターネットプロトコルアドレスのセットにわたって均一に分散される可能性が高く、これは転じて、要求がキャッシングプロキシサーバにわたって均一に分散されることを意味する。したがって、これらのファイルを記憶するために使用されるキャッシングリソースは、キャッシングプロキシサーバにわたって均一に分散され、ファイルに対する要求は、キャッシングプロキシサーバにわたって均一に分散される。したがって、方法は、キャッシングインフラストラクチャにわたる、記憶のバランスと負荷のバランスの両方を実現する。

#### 【0457】

上で説明された手法に対する多数の変形が当業者に知られており、多くの場合、これらの変形は、所与のプロキシに記憶されるファイルのセットが、キャッシングプロキシサーバが受信した要求のセットによって少なくとも一部決定されるという性質を保持する。所与のホスト名が複数の物理キャッシングプロキシサーバに解決する一般的な事例では、すべてのこれらのサーバが、頻繁に要求される任意の所与のファイルのコピーを最終的に記憶することが一般的である。そのような重複は望ましくないことがあり、それは、キャッシングプロキシサーバ上の記憶リソースが制限され、結果として、ファイルが時々キャッシングから除去(ページ)され得るからである。ここで説明される新規の方法は、この重複が

10

20

30

40

50

減らされるような方式で、所与のファイルに対する要求がキャッシングプロキシサーバに向けられることを確実にし、これによって、キャッシングからファイルを除去する必要性を減らし、任意の所与のファイルがプロキシキャッシング中に存在する(すなわち、プロキシキャッシングからページされていない)確率を高める。

#### 【0458】

ファイルがプロキシキャッシング中に存在する場合、クライアントに送信される応答はより高速になり、このことは、メディア再生の一時停止、したがって悪いユーザ体験をもたらし得る、要求されるファイルの到達の遅延の確率を減らす際に有利である。加えて、ファイルがプロキシキャッシング中に存在しない場合、要求は別のサーバに送信されることがあり、サービングインフラストラクチャと、サーバ間のネットワーク接続の両方に、追加の負荷を引き起こす。多くの場合、要求が送信されるサーバは離れた位置にあることがあり、このサーバからキャッシングプロキシサーバへのファイルの返信は、送信コストを招き得る。したがって、ここで説明される新規の方法は、これらの送信コストの低減をもたらす。

10

#### 【0459】

##### 確率的なファイル全体の要求

HTTPプロトコルがRange要求とともに使用される場合の特定の問題は、サービングインフラストラクチャにスケーラビリティをもたらすために一般に使用されるキャッシングサーバの挙動である。HTTPキャッシングサーバがHTTP Rangeヘッダをサポートするのは一般的であり得るが、異なるHTTPキャッシングサーバの厳密な挙動は、実装形態により変化する。多くのキャッシングサーバの実装形態は、ファイルがキャッシング中で利用可能な場合に、キャッシングからのRange要求にサービスする。HTTPキャッシングサーバの一般的な実装形態は常に、キャッシングサーバがファイルのコピーを有さない限り(キャッシングサーバまたは発信元サーバ)、Rangeヘッダを含む下流のHTTP要求を上流のノードに転送する。いくつかの実装形態では、Range要求に対する上流の応答はファイル全体であり、このファイル全体がキャッシングされ、下流のRange要求に対する応答がこのファイルから抽出され送信される。しかしながら、少なくとも1つの実装形態では、Range要求に対する上流の応答は単に、Range要求自体の中のデータバイトであり、これらのデータバイトはキャッシングされず、代わりに、下流のRange要求に対する応答として送信されるだけである。結果として、クライアントによるRangeヘッダの使用は、ファイル自身が決してキャッシングに運ばれず、ネットワークの望ましくないスケーラビリティの特性が失われるという結果をもたらし得る。

20

#### 【0460】

上では、キャッシングプロキシサーバの動作が説明され、複数のロックの統合であるファイルからのロックを要求する方法も説明された。たとえば、これは、HTTP Range要求ヘッダの使用によって達成され得る。そのような要求は、以下で「部分的な要求」と呼ばれる。ロックサービングインフラストラクチャ101がHTTP Rangeヘッダに対する完全なサポートを提供しない場合に利点を有する、さらなる実施形態がここで説明される。一般に、ロックサービングインフラストラクチャ内のサーバ、たとえばコンテンツ配信ネットワークは、部分的な要求をサポートするが、ローカルの記憶装置(キャッシング)に部分的な要求に対する応答を記憶できない。そのようなサーバは、ファイル全体がローカルの記憶装置に記憶されない限り、要求を別のサーバに転送することによって部分的な要求を満たすことができ、ファイル全体がローカルの記憶装置に記憶される場合は、要求を別のサーバに転送することなく応答が送信され得る。

30

#### 【0461】

ロックサービングインフラストラクチャがこの挙動を示す場合、上で説明されたロック統合の新規の改善を利用するロック要求ストリーミングシステムは性能が悪い可能性があり、それは、部分的な要求であるすべての要求が別のサーバに転送され、キャッシングプロキシサーバによって要求がサービスされず、キャッシングプロキシサーバを1位に置くという目標を無意味にするからである。上で説明されたようなロック要求ストリ

40

50

一ミングプロセスの間に、クライアントは、ファイルの始めにあるブロックを、何らかの時点で要求することができる。

#### 【0462】

ここで説明される新規の方法によれば、ある条件が満たされた場合は常に、そのような要求が、ファイル中の最初のブロックに対する要求からファイル全体に対する要求へと変換され得る。ファイル全体に対する要求がキャッシングプロキシサーバによって受信されると、プロキシサーバは通常、応答を記憶する。したがって、完全なファイルに対するものかまたは部分的な要求かにかかわらず、後続の要求がキャッシングプロキシサーバによって直接サービスされ得るように、これらの要求の使用によって、ファイルがローカルキャッシングプロキシサーバのキャッシュに運ばれる。上記の条件は、所与のファイルと関連付けられる要求のセット、たとえば、問題のコンテンツアイテムを見ているクライアントのセットによって生成される要求のセットの間で、条件がこれらの要求の少なくとも与えられた断片に対して満たされるというものであり得る。10

#### 【0463】

適切な条件の例は、ランダムに選ばれた数が与えられた閾値を上回るというものである。この閾値は、ファイル全体の要求への単一のブロック要求の変換が、平均して、要求の与えられた断片に対して、たとえば10回に1回(この場合、ランダムな数が間隔[0,1]から選ばれてよく、閾値は0.9であり得る)発生するように、設定され得る。適切な条件の別の例は、ブロックと関連付けられる何らかの情報およびクライアントと関連付けられる何らかの情報に対して計算されるハッシュ関数が、値の提供されたセットのうちの1つをとることである。この方法は、頻繁に要求されるファイルに対して、ファイルがローカルのプロキシサーバのキャッシングへと運ばれるという利点を有するが、ブロック要求ストリーミングシステムの動作は、各要求が単一のブロックに対するものである標準的な動作からは大きく変更されない。多くの事例では、単一のブロック要求からファイル全体の要求への要求の変更が起きる場合、クライアント手順は別様に、ファイル内の他のブロックを要求することを続ける。そうである場合、問題のブロックがファイル全体の要求の結果としていずれの場合にも受信されるので、そのような要求は抑制され得る。20

#### 【0464】

##### URL構築およびセグメントリスト生成および探索

セグメントリスト生成は、オンデマンドの場合のメディアプレゼンテーションの開始に対する、または実時間で表される、何らかの開始時間starttimeで開始する特定の表現に対して、特定のクライアントローカル時間NOWにおいてMPDからクライアントがセグメントリストをどのように生成し得るかという問題を扱う。セグメントリストは、メディアセグメントのリストとともに、ロケータ、たとえば、任意選択の初期の表現メタデータに対するURLを含み得る。各メディアセグメントは、starttime、継続時間、およびロケータを割り当てられていてよい。starttimeは通常、セグメント中の含まれるメディアのメディア時間の概略値を表すが、必ずしもサンプルの正確な時間を表さない。starttimeは、適切なときにダウンロード要求を出すために、HTTPストリーミングクライアントにより使用される。各々の開始時間を含むセグメントリストの生成は、様々な方法で行われ得る。URLはプレイリストとして提供されてよく、または、URL構築規則は有利には、セグメントリストの小型の表現のために使用されてよい。3040

#### 【0465】

URL構築に基づくセグメントリストは、たとえば、MPDが、FileDynamicInfoまたは等価な信号などの、特定の属性または要素によってセグメントリストをシグナリングする場合に、実行され得る。URL構築からセグメントリストを作成するための一般的な方法が、以下の「URL構築の概要」セクションで与えられる。プレイリストベースの構築は、たとえば、異なる信号によってシグナリングされ得る。セグメントリストを探索し、正確なメディア時間に到達することも、この状況において有利に実施される。

#### 【0466】

##### URL構築の概要

前に説明されたように、本発明の一実施形態では、クライアントデバイスがプレゼンテーションのブロックに対するファイル識別子を構築することを可能にするURL構築規則を含むメタデータが提供され得る。ここで、URL構築規則に対する変更、利用可能な符号化物の数に対する変更、ビットレート、アスペクト比、解像度、オーディオまたはビデオのコーデックもしくはコーデックのパラメータ、または他のパラメータのような、利用可能な符号化物と関連付けられるメタデータに対する変更を提供する、ブロック要求ストリーミングシステムに対するさらなる新規の改善について説明する。

#### 【0467】

この新規の改善において、プレゼンテーション全体の中の時間間隔を示す、メタデータファイルの各要素と関連付けられる追加のデータが提供され得る。この時間間隔の中では、要素は有効であると見なされてよく、その時間間隔以外では、要素は無視されてよい。さらに、一度だけ、または多くとも一度現れることが以前に許可された要素が複数回現れ得るように、メタデータのシンタックスが増強され得る。この場合、そのような要素に対して、規定された時間間隔がばらばらでなければならないということを規定する、追加の制約が提供され得る。任意の所与の時間の瞬間ににおいて、その時間間隔が所与の時間の瞬間を含む要素のみを考慮することは、元のメタデータシンタックスと矛盾しないメタデータファイルをもたらす。そのような時間間隔を、有効間隔と呼ぶ。したがって、この方法は、上で説明された種類の単一のメタデータファイルの変更の中でのシグナリングを実現する。有利には、そのような方法は、プレゼンテーション内の規定された点における、説明された種類の変更をサポートするメディアプレゼンテーションを提供するために使用され得る。10

#### 【0468】

##### URL構築器

本明細書で説明されるように、ブロック要求ストリーミングシステムの一般的な特徴は、利用可能なメディア符号化物を識別しそれらの符号化物からのブロックを要求するためにクライアントによって必要とされる情報を提供する、「メタデータ」をクライアントに提供する必要があるということである。たとえば、HTTPの場合、この情報は、メディアブロックを含むファイルのURLを含み得る。所与の符号化物に対するブロックのURLを一覧にする、プレイリストファイルが提供され得る。各符号化物に対して1つの、複数のプレイリストファイルが、異なる符号化物に対応するプレイリストを一覧にするプレイリストのマスタープレイリストとともに、提供される。このシステムの欠点は、メタデータが非常に大きくなることがあるので、クライアントがストリームを開始するとき、メタデータが要求されるのにある時間がかかるということである。このシステムのさらなる欠点は、メディアデータブロックに対応するファイルが、リアルタイム(ライブ)で撮影されているメディアストリーム、たとえば、生のスポーツイベントまたはニュース番組から「その場で」生成される、ライブコンテンツの場合に明白である。この場合、プレイリストファイルは、新たなブロックが利用可能になるたびに(たとえば、数秒ごとに)更新され得る。クライアントデバイスは、プレイリストファイルを繰り返しフェッチして、新たなブロックが利用可能かどうかを判定し、それらのURLを取得することができる。このことは、サービングインフラストラクチャに大きな負荷を課すことがあり、具体的には、一般には数秒のオーダーのブロックサイズに等しい更新間隔よりも長く、メタデータファイルがキャッシュされ得ないということを意味する。30

#### 【0469】

ブロック要求ストリーミングシステムの1つの重要な態様は、ブロックを要求するためにファイルダウンロードプロトコルとともに使用されるべき、ファイル識別子、たとえばURLをクライアントに知らせるために使用される方法である。たとえば、プレゼンテーションの各表現に対して、メディアデータのブロックを含むファイルのURLを一覧にするプレイリストが提供される方法。この方法の欠点は、プレイリストの少なくともいくつかにおいて、再生が開始し得る前にファイル自体がダウンロードされる必要があり、チャネルザッピング時間が長くなり、したがって悪いユーザ体験を引き起こすということである。40

10

20

30

40

50

いくつかのまたは多くの表現を伴う長いメディアプレゼンテーションでは、ファイルURLのリストは大きいことがあるので、プレイリストファイルは大きいことがあり、チャネルザッピング時間がさらに長くなる。

#### 【0470】

この方法の別の欠点は、ライブコンテンツの場合に発生する。この場合、URLの完全なリストが事前に利用可能にされず、プレイリストファイルは、新たなブロックが利用可能になるにつれて定期的に更新され、クライアントは、更新されたバージョンを受信するために、プレイリストファイルを定期的に要求する。このファイルは頻繁に更新されるので、キャッシングプロキシサーバ内に長く記憶され得ない。これは、このファイルに対する要求の非常に多くが、他のサーバに、最終的にはファイルを生成するサーバに転送されることを意味する。人気のあるメディアプレゼンテーションの場合には、このことは、このサーバおよびネットワークに対する大きな負荷をもたらすことがあり、これは転じて、遅い応答時間、およびしたがって、長いチャネルザッピング時間および悪いユーザ体験をもたらし得る。最悪の場合には、サーバが過負荷になり、これにより、一部のユーザがプレゼンテーションを見られなくなる。10

#### 【0471】

使用され得るファイル識別子の形式に制約を課すのを避けることが、ブロック要求ストリーミングシステムの設計において望ましい。これは、多数の考慮事項は、特定の形式の識別子の使用の動機となり得るからである。たとえば、ブロックサービングインフラストラクチャがコンテンツ配信ネットワークである場合、ネットワークにわたる記憶またはサービング負荷の分配を望むことに関連するファイル名または記憶の協定、または、システムの設計時には予測され得ない特定の形式のファイル識別子につながる他の要件が存在し得る。20

#### 【0472】

ここで、上で言及された欠点を軽減しつつ、適切なファイル識別情報の表現方法を選ぶための柔軟性を保持する、さらなる実施形態が説明される。この方法では、メタデータは、ファイル識別子構築規則を含む、メディアプレゼンテーションの各表現に対して提供され得る。ファイル識別子構築規則は、たとえば、テキスト文字列を含み得る。プレゼンテーションの所与のブロックに対するファイル識別子を決定するために、ファイル識別子構築規則の解釈の方法が提供されてよく、この方法は、入力パラメータの決定と、入力パラメータとともにファイル識別情報構築規則の評価とを含む。入力パラメータは、たとえば、識別されるべきファイルのインデックスを含んでよく、ここで、最初のファイルはインデックス0を有し、2番目のファイルはインデックス1を有し、3番目のファイルはインデックス2を有し、以下同様である。たとえば、各ファイルが同じ時間の長さ(またはほぼ同じ時間の長さ)にわたる場合、プレゼンテーション内の任意の所与の時間と関連付けられるファイルのインデックスは、簡単に決定され得る。あるいは、各ファイルがまたがるプレゼンテーション内の時間は、プレゼンテーションまたはバージョンメタデータ内で提供され得る。30

#### 【0473】

一実施形態では、ファイル識別子構築規則は、入力パラメータに対応する何らかの空間識別子を含み得るテキスト文字列を含み得る。ファイル識別子構築規則の評価の方法は、テキスト文字列内の空間識別子の位置を決定するステップと、対応する入力パラメータの値の文字列表現によって、各々のそのような空間識別子を置き換えるステップとを含む。40

#### 【0474】

別の実施形態では、ファイル識別子構築規則は、式言語に準拠するテキスト文字列を含み得る。式言語は、その言語における式が準拠し得るシンタックスの定義と、シンタックスに準拠する文字列を評価するための規則のセットとを含む。

#### 【0475】

ここで、図21以下を参照して、具体的な例が説明される。拡張バッカスナウア記法で定義される、適切な式言語に対するシンタックスの定義の例は、図21に示される通りである50

。図21の<expression>プロダクションに準拠する文字列を評価するための規則の例は、<expression>プロダクションに準拠する文字列を、次のように、<literal>プロダクションに準拠する文字列へと再帰的に変換することを含む。

#### 【0476】

<literal>プロダクションに準拠する<expression>は変更されない。

#### 【0477】

<variable>プロダクションに準拠する<expression>は、<variable>プロダクションの<token>文字列によって識別される変数の値によって置き換えられる。

#### 【0478】

<function>プロダクションに準拠する<expression>は、これらの規則に従って引数の各々を評価し、以下で説明されるように<function>プロダクションの<token>要素に応じてこれらの引数に変換を適用することによって、評価される。10

#### 【0479】

<expression>プロダクションの最後の代替形態に準拠する<expression>は、2つの<expression>要素を評価して、以下で説明されるように<expression>プロダクションの最後の代替形態の<operator>要素に応じて動作をこれらの引数に適用することによって、評価される。

#### 【0480】

上で説明される方法では、複数の変数が定義され得る状況において評価が行われると仮定される。変数は(名前、値)のペアであり、「名前」は<token>プロダクションに準拠する文字列であり、「値」は<literal>プロダクションに準拠する文字列である。いくつかの変数は、評価が開始する前に評価プロセスの外側で定義され得る。他の変数は、評価プロセス自体の中で定義され得る。1つだけの変数が各々のあり得る「名前」とともに存在するという意味で、すべての変数は「グローバル」である。20

#### 【0481】

関数の例は、「printf」関数である。この関数は、1つまたは複数の引数を許容する。第1の引数は、<string>プロダクション(以後「文字列」)に準拠し得る。printf関数は、その第1の引数の変換されたバージョンとして評価される。適用される変換は、C標準ライブラリの「printf」関数と同じであり、<function>プロダクションに含まれる追加の引数が、C標準ライブラリのprintf関数によって要求される追加の引数を提供する。30

#### 【0482】

関数の別の例は、「hash」関数である。この関数は、2つの引数を許容し、第1の引数は文字列であってよく、第2の引数は<number>プロダクション(以後「数」)に準拠してよい。「hash」関数は、ハッシュアルゴリズムを第1の引数に適用し、第2の引数未満の非負の整数である結果を返す。適切なハッシュ関数の例は、図22に示されるC関数において与えられ、このC関数の引数は、入力文字列(囲いの引用符を除く)および数値的な入力値である。ハッシュ関数の他の例は、当業者にはよく知られている。

#### 【0483】

関数の別の例は、1つ、2つ、または3つの文字列引数をとる「Subst」関数である。1つの引数が与えられる場合、「Subst」関数の結果は第1の引数である。2つの引数が与えられる場合、「Subst」関数の結果は、第1の引数内での第2の引数(囲いの引用符を除く)のあらゆる存在をなくし、そのように修正された第1の引数を返すことによって、計算される。3つの引数が与えられる場合、「Subst」関数の結果は、第1の引数内での第2の引数(囲いの引用符を除く)のあらゆる存在を第3の引数(囲いの引用符を除く)によって置き換え、そのように修正された第1の引数を返すことによって、計算される。40

#### 【0484】

演算子のいくつかの例は、<operator>プロダクション「+」、「-」、「/」、「\*」、「%」によってそれぞれ識別される、加算、減算、除算、乗算、およびモジュロ演算子である。これらの演算子は、<operator>プロダクションの両側の<expression>プロダクションが数字として評価されることを要求する。演算子の評価は、適切な算術演算(それぞれ加50

算、減算、除算、乗算、およびモジュロ)を通常の方式でこれらの2つの数に適用し、<number>プロダクションに準拠する形式で結果を返すことを含む。

#### 【0485】

演算子の別の例は、<operator>プロダクション「=」によって識別される割当て演算子である。この演算子は、その内容が<token>プロダクションに準拠する文字列として左側の引数が評価されることを要求する。文字列の内容は、囲いの引用符内の文字列として定義される。等号演算子は、その名前が左側の引数の内容に等しい<token>である変数が、右側の引数を評価した結果に等しい値へと割り当てられるようになる。この値はまた、演算子の式を評価した結果である。

#### 【0486】

演算子の別の例は、<operator>プロダクション「;」によって識別される順序演算子である。この演算子の評価の結果は、右側の引数である。すべての演算子のように、両方の引数が評価され、左側の引数が最初に評価されることに留意されたい。

#### 【0487】

本発明の一実施形態では、ファイルの識別子は、要求されるファイルを識別する入力変数の特定のセットによって、上記の規則に従ったファイル識別子構築規則を評価することによって、取得され得る。入力変数の例は、名前「インデックス」と、プレゼンテーション内のファイルの数値的なインデックスに等しい値とを伴う変数である。入力変数の別の例は、名前「ビットレート」と、プレゼンテーション要求されたバージョンの平均のビットレートに等しい値とを伴う変数である。

#### 【0488】

図23は、ファイル識別子構築規則のいくつかの例を示し、ここで、入力変数は、所望されるプレゼンテーションの表現の識別子を与える「id」、および、ファイルの順序番号を与える「seq」である。

#### 【0489】

本開示を読んだ当業者には明らかなように、上の方法の多数の変形が可能である。たとえば、上で説明された関数および演算子のすべてが与えられるとは限らず、または、追加の関数または演算子が与えられることがある。

#### 【0490】

URL構築規則およびタイミング

30

#### 【0491】

このセクションは、表現およびメディアプレゼンテーションの中で、ファイルまたはセグメントURI、さらには各セグメントの開始時間を割り当てるための、基本的なURI構築規則を提供する。

#### 【0492】

この節では、クライアントにおけるmedia presentation descriptionの利用可能性が仮定される。

#### 【0493】

HTTPストリーミングクライアントが、メディアプレゼンテーション内でダウンロードされるメディアを再生していると仮定する。HTTPクライアントの実際のプレゼンテーション時間は、プレゼンテーション時間がプレゼンテーションの始めに対してどこにあるかについて定義され得る。初期化のときには、プレゼンテーション時間t=0が仮定され得る。

40

#### 【0494】

任意の点tにおいて、HTTPクライアントは、実際のプレゼンテーション時間tより最大でMaximumClientPreBufferTime前にある再生時間tP(プレゼンテーションの始めに対するものもある)を伴う任意のデータ、および、ユーザ対話、たとえば、探索、早送りなどによる、必要とされる任意のデータをダウンロードすることができる。いくつかの実施形態では、MaximumClientPreBufferTimeは、クライアントが制約を伴わずに現在の再生時間tPの前にデータをダウンロードできるという意味で、規定すらされないことがある。

#### 【0495】

50

HTTPクライアントは、不必要的データのダウンロードを避けることができ、たとえば、再生されることが予想されない表現からの任意のセグメントは通常、ダウンロードされなくてよい。

#### 【0496】

ストリーミングサービスを提供する際の基本プロセスは、たとえば、HTTP GET要求またはHTTP partial GET要求を使用することによって、全体のファイル/セグメントまたはファイル/セグメントのサブセットをダウンロードするための、適切な要求の生成によるデータのダウンロードであり得る。この説明は、特定の再生時間tPに対するデータにどのようにアクセスするかを扱うが、一般に、クライアントは、非効率な要求を避けるために、より長い時間範囲の再生時間に対するデータをダウンロードすることができる。HTTPクライアントは、ストリーミングサービスを提供する際に、HTTP要求の数/頻度を最小限にすることができます。10

#### 【0497】

特定の表現の中の、再生時間tP、または少なくとも再生時間tPに近い時間におけるメディアデータにアクセスするために、クライアントは、この再生時間を含むファイルに対するURLを決定し、加えて、この再生時間にアクセスするためにファイル中のバイト範囲を決定する。

#### 【0498】

Media Presentation Descriptionは、たとえば、RepresentationID属性の使用によって、表現id rを、各表現に割り当てることができる。言い換えると、MPDのコンテンツは、取込システムによって書き込まれると、またはクライアントによって読み取られると、割当てが存在するように解釈される。id rを伴う特定の表現の、特定の再生時間tPに対するデータをダウンロードするために、クライアントは、ファイルの適切なURIを構築することができる。20

#### 【0499】

Media Presentation Descriptionは、次の属性を各表現rの各ファイルまたはセグメントに割り当てることができる。

#### 【0500】

(a)  $i=1, 2, \dots, Nr$ である、表現r内のファイルの順序番号i、(b)  $ts(r, i)$ として定義される、プレゼンテーション時間に対する表現id rおよびファイルインデックスiを伴うファイルの相対的な開始時間、(c)  $FileURI(r, i)$ として示される、表現id rおよびファイルインデックスiを伴うファイル/セグメントのファイルURI。30

#### 【0501】

一実施形態では、ファイルの開始時間およびファイルURIは、表現に対して明示的に提供され得る。別の実施形態では、ファイルURIのリストが明示的に提供されてよく、このとき、リスト中の位置に従ってインデックスiを各ファイルURIが固有に割り当てられ、セグメントの開始時間が1からi-1までのセグメントに対するすべてのセグメントの継続時間の合計として導出される。各セグメントの継続時間は、上で論じられた規則のいずれかに従って提供され得る。たとえば、基本的な数学の当業者は、他の方法を使用して、単一の要素または属性、および表現中のファイルURIの位置/インデックスから、容易に開始時間を導出するための方法を導出することができる。40

#### 【0502】

動的なURI構築規則がMPDにおいて提供される場合、各ファイルの開始時間および各ファイルのURIは、構築規則、要求されるファイルのインデックス、および、場合によってはmedia presentation descriptionにおいて提供される何らかの追加のパラメータを使用することによって、動的に構築され得る。情報は、たとえば、FileURIPatternおよびFileInfoDynamicのような、MPDの属性および要素において提供され得る。FileURIPatternは、ファイルインデックス順序番号iおよび表現ID rに基づいてURIをどのように構築するかについての情報を提供する。FileURIFormatは次のように構築される。

#### 【0503】

```
FileURIFormat=sprintf("%s%s%s%s%s.%s",BaseURI,BaseFileName,
【0504】
RepresentationIDFormat,SeparatorFormat,
【0505】
FileSequenceIDFormat,FileExtension);
```

【0506】  
また、FileURI(r,i)は次のように構築される。

【0507】  
FileURI(r,i)=sprintf(FileURIFormat,r,i);  
【0508】

各ファイル/セグメントに対する相対的な開始時間ts(r,i)は、この表現中のセグメントの継続時間を記述する、MPDに含まれる何らかの属性、たとえば、FileInfoDynamic属性によって導出され得る。MPDはまた、上で規定されたのと同じ方法で、メディアプレゼンテーション中のすべてのプレゼンテーションに対して、または、少なくともある期間のすべての表現に対してグローバルである、FileInfoDynamic属性の順序を含み得る。表現r中の特定の再生時間tPに対するメディアデータが要求される場合、このインデックスの再生時間がts(r,i(r,tP))とts(r,i(r,tP)+1)の間にあるように、対応するインデックスi(r,tP)がi(r,t<sub>p</sub>)として導出され得る。セグメントへのアクセスはさらに、上の事例によって制約されることがあり、たとえば、セグメントはアクセス可能ではない。

【0509】

対応するセグメントのインデックスおよびURIが取得されてから正確な再生時間tPにアクセスすることは、実際のセグメントのフォーマットに依存する。この例では、メディアセグメントが一般性を失うことなく0で開始するローカルの時間軸を有すると仮定する。再生時間tPにおいてデータにアクセスしデータを提示するために、クライアントは、i=i(r,t<sub>p</sub>)であるURI FileURI(r,i)を通じてアクセスされ得るファイル/セグメントから、ローカルの時間に対応するデータをダウンロードすることができる。

【0510】

一般に、クライアントは、ファイル全体をダウンロードすることができ、次いで、再生時間tPにアクセスすることができる。しかしながら、必ずしも3GPファイルのすべての情報がダウンロードされる必要はなく、それは、3GPファイルが、ローカルのタイミングをバイト範囲と対応付けるための構造を提供するからである。したがって、十分なランダムアクセス情報が利用可能である限り、再生時間tPにアクセスするための特定のバイト範囲だけで、メディアを再生するには十分であり得る。構造についての十分な情報、バイト範囲の対応付け、およびメディアセグメントのローカルなタイミングも、たとえば、セグメントインデックスを使用して、セグメントの最初の部分で提供され得る。セグメントの最初の、たとえば1200バイトへのアクセス権を有することによって、クライアントは、再生時間tPに対して必要なバイト範囲に直接アクセスするための十分な情報を有し得る。

【0511】

さらなる例では、以下のように場合によっては「tidx」ボックスとして規定されるセグメントインデックスが、要求される1つまたは複数のフラグメントのバイトオフセットを特定するために使用されると仮定する。Partial GET要求が、要求される1つまたは複数のフラグメントに対して形成され得る。他の代替形態が存在し、たとえば、クライアントは、ファイルに対する標準的な要求を出し、第1の「tidx」ボックスが受信されたときにこれを取り消すことができる。

【0512】

探索

クライアントは、表現中の特定のプレゼンテーション時間tpを探索することを試み得る。MPDに基づいて、クライアントは、表現中の各セグメントのメディアセグメント開始時間およびメディアセグメントURLへのアクセス権を有する。クライアントは、開始時間tS(r,i)がプレゼンテーション時間tp以下となる最大のセグメントインデックスiとして、ブ

10

20

30

40

50

プレゼンテーション時間tpに対するメディアサンプルを含む可能性が最も高いセグメントのセグメントインデックスsegment\_index、すなわち、 $\text{segment\_index} = \max\{i \mid tS(r, i) \leq tp\}$ を得ることができる。セグメントURLは、FileURI(r, i)として取得される。

#### 【0513】

MPD中のタイミング情報は、ランダムアクセスポイントの配置、メディアトラックの整列、およびメディアのタイミングのドリフトに関連する問題が原因で、概略的であり得ることに留意されたい。結果として、上記の手順により識別されるセグメントは、tpのわずかに後の時間で開始することがあり、プレゼンテーション時間tpに対するメディアデータは前のメディアセグメント中にあることがある。探索の場合、探索時間は、取り出されたファイルの最初のサンプル時間に等しくなるように更新されてよく、または、先行するファイルが代わりに取り出されてよい。しかしながら、代替的な表現/バージョン間での切り替えが存在する場合を含めて、連続的な再生の間に、tpと取り出されたセグメントの始めとの間の時間に対するメディアデータはそれでも利用可能であることに留意されたい。10

#### 【0514】

プレゼンテーション時間tpに対する正確な探索のために、HTTPストリーミングクライアントは、ランダムアクセスポイント(RAP)にアクセスする必要がある。3GPP適応HTTPストリーミングの場合に、メディアセグメント中のランダムアクセスポイントを決定するために、クライアントは、たとえば、「tidx」または「sidx」ボックス中の情報を使用して、存在する場合、ランダムアクセスポイントと、メディアプレゼンテーション中の対応するプレゼンテーション時間とを位置決めすることができる。セグメントが3GPPムービーフラグメントである場合、クライアントが、「moof」および「mdat」ボックス内の情報を使用して、たとえば、RAPを位置決めして、ムービーフラグメント中の情報からの必要なプレゼンテーション時間と、MPDから導出されるセグメント開始時間とを取得することも可能である。要求されたプレゼンテーション時間tpよりも前のプレゼンテーション時間を伴うRAPが利用可能ではない場合、クライアントは、前のセグメントにアクセスすることができ、または、探索の結果として最初のランダムアクセスポイントだけを使用することができる。メディアセグメントがRAPで開始する場合、これらの手順は単純である。20

#### 【0515】

メディアセグメントのすべての情報が必ずしもプレゼンテーション時間tpにアクセスするためダウンロードされる必要はないことにも留意されたい。クライアントは、たとえば、バイト範囲要求を使用して、メディアセグメントの始めから「tidx」または「sidx」ボックスを最初に要求することができる。「tidx」または「sidx」ボックスの使用によって、セグメントのタイミングが、セグメントのバイト範囲と対応付けられ得る。30

HTTP要求を継続的に使用することによって、メディアセグメントの関連する部分のみが、改善されたユーザ体験および少ない始動遅延のためにアクセスされ得る。

#### 【0516】

##### セグメントリストの生成

本明細書で説明されるように、durというシグナリングされた概略的なセグメント継続時間を持つ表現に対するセグメントのリストを作成するためにMPDによって提供される情報を使用する、単純なHTTPストリーミングクライアントをどのように実装すべきかは明らかであろう。いくつかの実施形態では、クライアントは、表現の連続的なインデックス*i=1, 2, 3, ...*内にメディアセグメントを割り当てることができ、すなわち、第1のメディアセグメントはインデックス*i=1*を割り当てられ、第2のメディアセグメントがインデックス*i=2*を割り当てられ、以下同様である。次いで、セグメントインデックス*i*を伴うメディアセグメントのリストはstartTime[i]を割り当てられ、URL[i]がたとえば次のように生成される。まず、インデックス*i*が1に設定される。第1のメディアセグメントの開始時間は0として取得され、startTime[1]=0である。メディアセグメント*i*のURL、URL[i]は、FileURI(r, i)として取得される。インデックス*i*を伴うすべての記述されたメディアセグメントに対してこのプロセスが続けられ、メディアセグメント*i*のstartTime[i]は(i-1)\*durとして取得され、URL[i]はFileURI(r, i)として取得される。40

## 【0517】

## 同時のHTTP/TCP要求

ブロック要求ストリーミングシステムにおける1つの課題は、再生のために時間内に完全に受信され得る最高品質のブロックを常に要求することに対する要求である。しかしながら、データ到達レートは事前に知られていないことがあるので、要求されたブロックが再生されるべき時間内に到達しないということが起こり得る。これにより、メディア再生が一時停止しなければならず、悪いユーザ体験をもたらす。この問題は、データ到達レートがブロックの受信の間に落ちた場合でも、時間内に受信される可能性がより高い、より低品質の(および、よってより小さなサイズの)ブロックを要求することによる、要求すべきブロックの選択についての保守的な手法を採用するクライアントアルゴリズムによって、軽減され得る。しかしながら、この保守的な手法は、ユーザまたは宛先デバイスに対して、悪いユーザ体験もあるより低品質の再生を配信する可能性があるという欠点を有する。問題は、複数のHTTP接続が以下で説明されるように異なるブロックをダウンロードするために同時に使用されるときには大きくなることがあり、それは、利用可能なネットワークリソースが複数の接続にわたって共有され、したがって、異なる再生時間を伴うブロックに対して同時に使用されるからである。

10

## 【0518】

クライアントが複数のブロックに対する要求を同時に出すことが有利であることがあり、この文脈では、「同時に」とは、要求に対する応答が重複する時間間隔において行われることを意味し、必ずしも、要求が厳密に、または概略的にすらも同時に行われないことがある。HTTPプロトコルの場合、この手法は、(よく知られているように)TCPプロトコルの挙動が原因で、利用可能な帯域幅の利用率を改善することができる。これは、コンテンツザッピング時間を改善するためには特に重要であり得る。それは、新たなコンテンツが初めて要求されると、それを通じてブロックに対するデータが要求される対応するHTTP/TCP接続が、開始するのが遅いことがあり、したがって、この時点でいくつかのHTTP/TCP接続を使用することで、最初のブロックのデータ配信時間を劇的に高速にすることができるからである。しかしながら、異なるHTTP/TCP接続を通じて異なるブロックまたはフラグメントを要求することは、性能の劣化にもつながり得る。それは、最初に再生されるべきブロックに対する要求が後続のブロックに対する要求と競合して、競合するHTTP/TCPダウンロードが配信速度に関して大きく変動し、したがって、要求の完了時間が大きく変わりやすいことがあり、どのHTTP/TCPダウンロードが迅速に完了し、どれがより遅いかを制御することは一般的に不可能であり、したがって、最初の数ブロックのHTTP/TCPダウンロードの少なくとも一部が完了するのが最後になり、長くおよび変わりやすいチャネルザッピング時間につながる可能性が高いからである。

20

## 【0519】

セグメントの各ブロックまたはフラグメントが、別個のHTTP/TCO接続を通じてダウンロードされ、並列接続の数がnであり、各ブロックの再生継続時間がt秒であり、セグメントと関連付けられるコンテンツのストリーミングレートがSであると仮定する。クライアントが最初にコンテンツのストリーミングを開始するとき、要求は、 $n*t$ 秒のメディアデータを表す最初のnブロックに対して出され得る。

30

## 【0520】

当業者に知られているように、TCP接続のデータレートには大きな変動がある。しかしながら、この議論を簡単にするために、最初のブロックが、要求される他のn-1個のブロックとほぼ同時に完全に受信されるように、理想的にはすべての接続が並列に進行すると仮定する。議論をさらに簡単にするために、n個のダウンロード接続によって利用される合計の帯域幅は、ダウンロードの継続時間全体に対して値Bに固定され、ストリーミングレートSは表現全体を通じて一定であると仮定する。さらに、メディアデータ構造は、ブロック全体がクライアントにおいて利用可能なときにブロックの再生が行われ得るようなものであると仮定し、すなわち、ブロックの再生は、たとえば、背後にあるビデオ符号化的構造が原因で、または、各フラグメントまたはブロックを別々に暗号化するために暗号

40

50

化が利用されており、したがって全体のフラグメントまたはブロックが復号され得る前に受信される必要があるために、ブロック全体が受信された後にのみ開始できる。したがって、以下の議論を簡単にするために、ブロックのいずれかが再生され得る前に、ブロック全体が受信される必要があると仮定する。そうすると、最初のブロックが到達し再生され得るまでに必要とされる時間は、約 $n*t*S/B$ である。

### 【0521】

コンテンツザッピング時間を最小限にすることが望ましいので、 $n*t*S/B$ を最小限にすることが望ましい。 $t$ の値は、背後にあるビデオ符号化構造、および取込方法がどのように利用されるかなどの要因によって決定され得るので、 $t$ はかなり小さいことがあるが、非常に小さな $t$ の値は、過剰に複雑なセグメントマップにつながり、場合によっては、効率的なビデオの符号化および復号が使用される場合はそれらと適合しないことがある。 $n$ の値はまた、 $B$ の値に影響を与えることがある。すなわち、 $B$ は、より大きな接続の数 $n$ に対してはより大きいことがあり、したがって、接続の数 $n$ を減らすことには、利用される利用可能な帯域幅の量 $B$ を減らす可能性があるという、負の副作用があり、よって、コンテンツザッピング時間を減らすという目標を達成するには効果的ではないことがある。 $S$ の値は、ダウンロードおよび再生のためにどの表現が選ばれるかに依存し、理想的には、 $S$ は、所与のネットワーク条件に対するメディアの再生品質を最大限にするために、可能な限り $B$ に近くなければならない。したがって、この議論を簡単にするために、 $S$ はほぼ $B$ に等しいと仮定する。そうすると、チャネルザッピング時間は $n*t$ に比例する。したがって、より多くの接続を利用して異なるフラグメントをダウンロードすることは、通常そうであるように、接続によって利用される合計の帯域幅が接続の数に準線形に比例する場合、チャネルザッピング時間を劣化させ得る。10

### 【0522】

ある例として、 $t=1$ 秒、 $n=1$ の場合は $B$ の値=500Kbps、 $n=2$ の場合は $B$ の値=700Kbps、かつ $n=3$ の場合は $B$ の値=800Kbpsであると仮定する。 $S=700$ Kbpsを伴う表現が選ばれると仮定する。そうすると、 $n=1$ では、最初のブロックのダウンロード時間は $1*700/500=1.4$ 秒であり、 $n=2$ では、最初のブロックのダウンロード時間は $2*700/700=1$ 秒であり、 $n=3$ では、最初のブロックのダウンロード時間は $3*700/800=2.625$ 秒である。さらに、接続の数が増えるにつれて、(1つの接続でも、何らかの大きな変動がある可能性が高いが)接続の個々のダウンロード速度の変動が大きくなる可能性が高い。したがって、この例では、チャネルザッピング時間およびチャネルザッピング時間の変動は、接続の数が増えるにつれて増加する。直感的に、配信されているブロックは異なる優先順位を有し、すなわち、最初のブロックは最早の配信期限を有し、2番目のブロックは2番目に早い期限を有する、などであるが、ブロックがそれを通じて配信されているダウンロード接続は、配信の間にネットワークリソースをめぐって競合し、したがって、最早の期限を伴うブロックは、より多くの競合するブロックが要求されるとより遅れるようになる。一方、この場合でも、2つ以上のダウンロード接続を最終的に使用することは、持続可能により高いストリーミングレートをサポートすることを可能にし、たとえば、3つの接続によって、最高で800Kbpsのストリーミングレートがこの例ではサポートされ得るが、1つの接続では500Kbpsの1つのストリームしかサポートされ得ない。30

### 【0523】

実際には、上で述べられたように、接続のデータレートは、同じ接続の中で時間とともに、および接続の間で、大きく変わりやすいことがあり、結果として、 $n$ 個の要求されたブロックは一般に同じ時間で完了せず、実際には、あるブロックが別のブロックの半分の時間で完了し得るということが一般に起こり得る。この影響により、いくつかの場合には、最初のブロックが他のブロックよりもはるかに早く完了することがあり、かつ他の場合には最初のブロックが他のブロックよりもはるかに遅く完了することがあり、結果として、再生の開始がいくつかの場合には比較的早く起こることがあり、かつ他の場合には起こるのが遅いがあるので、予測不可能な挙動をもたらす。この予測不可能な挙動は、ユーザを苛立たせるものであることがあるので、悪いユーザ体験であると見なされ得る。4050

**【 0 5 2 4 】**

したがって、必要とされるのは、チャネルザッピング時間およびチャネルザッピング時間の変わりやすさを改善するために複数のTCP接続が利用され得ると同時に、可能な高品質のストリーミングレートをサポートする、方法である。やはり必要とされるのは、必要な場合に、利用可能な帯域幅のより大きな部分が最も近い再生時間を伴うブロックに割り振られ得るように、各ブロックに割り振られた利用可能な帯域幅の部分が、ブロックの再生時間が近づくにつれて調整されることを可能にするための方法である。

**【 0 5 2 5 】****協調的なHTTP/TCP要求**

ここで、協調的な方式で同時のHTTP/TCP要求を使用するための方法について説明する。  
受信機は、たとえば、複数のHTTPバイト範囲要求を使用して、複数の同時の協調的なHTTP/TCP要求を利用できること、各々のそのような要求は、ソースセグメント中のフラグメントの一部、または、ソースセグメントのフラグメントのすべて、または、修復セグメントの修復フラグメントの一部、または、修復セグメントの修復フラグメントのすべてに対するものである。

10

**【 0 5 2 6 】**

FEC修復データの使用を伴う協調的なHTTP/TCP要求の利点は、高速なチャネルザッピング時間を安定して提供するために特に重要であり得る。たとえば、チャネルザッピング時間においては、TCP接続が開始されたばかりであるか、またはある期間の間アイドル状態であった可能性が高く、その場合、輻輳ウインドウcwndはその接続に対する最小の値にあり、したがって、これらのTCP接続の配信速度は、上昇するのにいくつかのラウンドトリップタイム(RTT)がかかり、この上昇時間の間は、異なるTCP接続上の配信速度に大きな変動がある。

20

**【 0 5 2 7 】**

非FEC方法の概要がここで説明され、これは、協調的なHTTP/TCP要求方法であり、ソースブロックのメディアデータのみが、複数の同時のHTTP/TCP接続を使用して要求され、すなわち、FEC修復データが要求されない。非FEC方法では、同じフラグメントの部分が、異なる接続を通じて、たとえば、フラグメントの部分に対するHTTP接続バイト範囲要求を使用して要求されるので、たとえば、各HTTPバイト範囲要求は、フラグメントに対するセグメントマップにおいて示されるバイト範囲の部分に対するものである。個々のHTTP/TCP要求が、いくつかのRTT(ラウンドトリップタイム)を通じて、利用可能な帯域幅を完全に利用するためにその配信速度を上昇させることができるので、配信速度が利用可能な帯域幅よりも小さい、比較的長い期間が存在し、したがって、単一のHTTP/TCP接続がたとえば再生されるべきコンテンツの最初のフラグメントをダウンロードするために使用される場合、チャネルザッピング時間は大きいことがある。非FEC方法を使用して、異なるHTTP/TCP接続を通じて同じフラグメントの異なる部分をダウンロードすることは、チャネルザッピング時間を大きく低下させ得る。

30

**【 0 5 2 8 】**

FEC方法の概要がここで説明され、これは協調的なHTTP/TCP要求方法であり、ソースセグメントのメディアデータおよびメディアデータから生成されたFEC修復データが、複数の同時のHTTP/TCP接続を使用して要求される。FEC方法では、同じフラグメントの部分およびそのフラグメントから生成されたFEC修復データが、異なる接続を通じて、フラグメントの部分に対するHTTP接続バイト範囲要求を使用して要求されるので、たとえば、各HTTPバイト範囲要求は、フラグメントに対するセグメントマップにおいて示されるバイト範囲の部分に対するものである。個々のHTTP/TCP要求が、いくつかのRTT(ラウンドトリップタイム)を通じて、利用可能な帯域幅を完全に利用するためにその配信速度を上昇させることができるので、配信速度が利用可能な帯域幅よりも小さい、比較的長い期間が存在し、したがって、単一のHTTP/TCP接続がたとえば再生されるべきコンテンツの最初のフラグメントをダウンロードするために使用される場合、チャネルザッピング時間は大きいことがある。FEC方法を使用することは、非FEC方法と同じ利点を有し、フラグメントが復元

40

50

され得る前に要求されたデータのすべてが到達する必要がなく、チャネルザッピング時間およびチャネルザッピング時間の変動がさらに減るという追加の利点を有する。異なるTCP接続を通じて要求を行うこと、および、接続の少なくとも1つでFEC修復データも要求することによる再度の要求を行うことによって、たとえば、メディア再生の開始を可能にする最初に要求されたフラグメントを復元するために十分な量のデータを配信するのにかかる時間は、大きく低減され、協調的なTCP接続およびFEC修復データが使用されなかった場合よりもはるかに安定させられ得る。

#### 【0529】

図24(a)～図24(e)は、emulated evolution data optimized(EVDO)ネットワークの同じHTTPウェブサーバからの同じクライアントへの、同じリンク上を通る5つのTCP接続の配信レートの変動を示す。図24(a)～図24(e)では、X軸は秒単位の時間を示し、Y軸は、各接続に対して、1秒という間隔にわたって測定される、5つのTCP接続の各々を通じてクライアントにおいてビットが受信されるレートを示す。この特定の模擬においては、このリンク上を通る全体で12個のTCP接続があったので、ネットワークは示された時間の間は比較的負荷が高く、これは、2つ以上のクライアントがモバイルネットワークの同じセル内でストリーミングしている場合には典型的であり得る。配信レートは時間とともに幾分相關付けられるが、多くの時点において、5つの接続の配信レートには大きな差があることに留意されたい。

#### 【0530】

図25は、サイズが250,000ビット(約31.25キロバイト)であるフラグメントに対する可能な要求構造を示し、フラグメントの異なる部分に対して並列に行われる4つのHTTPバイト範囲要求があり、すなわち、第1のHTTP接続は最初の50,000ビットを要求し、第2のHTTP接続は次の50,000ビットを要求し、第3のHTTP接続は次の50,000ビットを要求し、第4のHTTP接続は次の50,000ビットを要求する。FECが使用されない場合、すなわち、非FEC方法では、これらが、この例でのフラグメントに対するすべての4つの要求である。FECが使用される場合、すなわち、FEC方法では、この例では、フラグメントから生成される、追加の50,000ビットの修復セグメントのFEC修復データを要求する1つの追加のHTTP接続がある。

#### 【0531】

図26は、図24(a)～図24(e)に示される5つのTCP接続の最初の数秒の拡大であり、図26では、X軸は100ミリ秒の間隔で時間を示し、Y軸は100ミリ秒の間にわたって測定された5つのTCP接続の各々でクライアントにおいてビットが受信されるレートを示す。1つの線は、最初の4つのHTTP接続(FECデータがそれを通じて要求されるHTTP接続を除く)からフラグメントに対してクライアントにおいて受信されたビット、すなわち、非FEC方法を使用して到達するものの合計の量を示す。別の線は、5つすべてのHTTP接続(FECデータがそれを通じて要求されるHTTP接続を含む)からフラグメントに対してクライアントにおいて受信されたビット、すなわち、FEC方法を使用して到達するものの合計の量を示す。FEC方法では、フラグメントが要求された250,000ビットのうちの任意の200,000ビットの受信からFEC復号され得ると仮定され、これは、たとえば、リードソロモンFEC符号が使用されれば実現されることが可能であり、たとえば、Luby IVで説明されるRaptorQ符号が使用されれば基本的に実現され得る。FEC方法では、この例では、1秒の後にFEC復号を使用してフラグメントを復元するために十分なデータが受信され、(最初のフラグメントが完全に再生される前に後続のフラグメントに対するデータが要求され受信されると仮定すると)1秒というチャネルザッピング時間を可能にする。非FEC方法では、この例では、4つの要求に対するすべてのデータが、フラグメントが復元され得る前に受信される必要があり、フラグメントの復元は1.7秒の後に起こり、1.7秒というチャネルザッピング時間をもたらす。したがって、図26に示される例では、非FEC方法は、FEC方法よりもチャネルザッピング時間に関して70%悪い。この例においてFEC方法により示される利点の理由の1つは、FEC方法では、要求されたデータの任意の80%の受信がフラグメントの復元を可能にするが、非FEC方法では、要求されたデータの100%の受信が必要とされる、ということである。したがって、非FEC方法は、配信を終えるために最も遅いTCP接続を待機しなければならず、TCP配

10

20

30

40

50

信レートの自然な変動により、平均的なTCP接続と比較して、最も遅いTCP接続の配信速度が大きく変動する傾向がある。FEC方法により、この例では、1つの遅いTCP接続が、フラグメントが復元可能であるときを決定しない。代わりに、FEC方法では、十分なデータの配信は、最悪の場合のTCP配信レートよりも、平均のTCP配信レートにはるかに依存する。

#### 【0532】

上で説明された非FEC方法およびFEC方法の多くの変形がある。たとえば、協調的なHTTP/TCP要求は、チャネルザップが起きてから最初の数フラグメントだけのために使用されてよく、その後、単一のHTTP/TCP要求のみが、さらなるフラグメント、複数のフラグメント、またはセグメント全体をダウンロードするために使用される。別の例として、使用される協調的なHTTP/TCP接続の数は、要求されているフラグメントの緊急性、すなわち、これらのセグメントの再生時間がどれだけ差し迫っているかということと、現在のネットワーク条件の両方に応じたものであり得る。

#### 【0533】

いくつかの変形では、複数のHTTP接続が、修復セグメントから修復データを要求するために使用され得る。他の変形では、たとえば、メディアバッファの現在のサイズおよびクライアントにおけるデータ受信レートに応じて、異なる量のデータが、異なるHTTP接続上で要求され得る。別の変形では、ソース表現は互いに独立しておらず、代わりに階層化されたメディアコーディングを表し、このとき、たとえば、改善されたソース表現は基本ソース表現に依存し得る。この場合、基本ソース表現に対応する修復表現があつてよく、別の修復表現は基本ソース表現と改善ソース表現の組合せに対応する。

#### 【0534】

追加の全体的な要素が、上で開示された方法により実現できる利点を増やす。たとえば、使用されるHTTP接続の数は、メディアバッファ中の現在のメディアの量、および/またはメディアバッファへの受信レートに応じて変化し得る。FEC、すなわち上で説明されたFEC方法およびその方法の変形を使用する協調的なHTTP要求は、メディアバッファが比較的空いているときには積極的に使用されてよく、たとえば、より多くの協調的なHTTP要求が最初のフラグメントの異なる部分に対して並列に行われ、ソースフラグメントのすべて、および対応する修復フラグメントからの修復データの比較的大きな部分を要求し、次いで、メディアバッファが増えるに従って、より少数の同時のHTTP要求に移行し、要求当たりより多くの部分のメディアデータを要求し、修復データのより小さな断片を要求し、たとえば、1つ、2つ、または3つの同時のHTTP要求に移行し、要求ごとに全体のフラグメントまたは複数の連続的なフラグメントに対する要求を行うことに移行し、修復データを要求しないことに移行する。

#### 【0535】

別の例として、FEC修復データの量は、メディアバッファサイズに応じて変化してよく、すなわち、メディアバッファが小さい場合、より多くのFEC修復データが要求されてよく、メディアバッファが増えるにつれて、要求されるFEC修復データの量は減少してよく、メディアバッファが十分に大きくなつた何らかの時点で、修復データは要求されなくてよく、ソースプレゼンテーションのソースセグメントからのデータのみが要求されてよい。そのような改善された技法の利点は、より高速でより安定したチャネルザッピング時間と、起こり得るメディアの詰まりまたはストールに対するより大きな復元力とを可能にしつつ、同時に、要求メッセージのトラフィックとFEC修復データの両方を減らすことによって、ソースセグメント中のメディアを配信することのみによって消費されるであろう量を超えて使用される追加の帯域幅の量を最小限にして、同時に、所与のネットワーク条件に対して可能な最高のメディアレートのサポートを可能にできることである。

#### 【0536】

同時のHTTP接続を使用したときの追加の改善

HTTP/TCP要求は、適切な条件が満たされると廃棄されてよく、廃棄された要求において要求されたデータを置き換えるデータをダウンロードするために、別のHTTP/TCP接続が行われてよく、第2のHTTP/TCP要求は、元の要求の中にあるデータと厳密に同じデータ、

10

20

30

40

50

たとえばソースデータ、または重複するデータ、たとえば同じソースデータの一部と第1の要求において要求されなかった修復データ、または完全にはらばらのデータ、たとえば第1の要求で要求されなかった修復データを要求することができる。適切な条件の例は、与えられた時間内にブロックサーバインフラストラクチャ(BSI)からの応答がなかったこと、またはBSIへのトランスポート接続の確立に失敗したこと、またはサーバからの明示的な失敗メッセージの受信、または別の失敗の条件により、要求が失敗することである。

#### 【0537】

適切な条件の別の例は、予想される接続速度、または、含まれるメディアデータの再生時間もしくはその時間に依存する別の時間の前に応答を受信するために必要とされる接続速度の推定値との、接続速度の尺度(問題の要求に応答したデータ到達レート)の比較に従って、データの受信が異常に遅く進んでいるということである。10

#### 【0538】

この手法は、BSIが失敗を起こす、またはその性能が悪いことがある場合に利点を有する。この場合、上の手法は、BSI内の失敗または悪い性能にもかかわらず、クライアントがメディアデータの信頼性のある再生を継続できる確率を高める。いくつかの場合には、そのような失敗または悪い性能を時々示すような方法でBSIを設計することが有利であることがあり、たとえば、そのような設計は、そのような失敗または悪い性能を示さない、またはこれらを示すことがより少ない代替的な設計よりも、コストが低いことがある。この場合、本明細書で説明される方法は、ユーザ体験の当然の劣化を伴わずに、BSIに対するそのようなより低コストの設計の利用を可能にするという、さらなる利点を有する。20

#### 【0539】

別の実施形態では、所与のブロックに対応するデータに対して出される要求の数は、ブロックに関する適切な条件が満たされたかどうかに依存し得る。条件が満たされない場合、クライアントは、ブロックに対するすべての現在は未完了のデータ要求の完了に成功することが、高い確率でブロックの復元を可能にする場合、ブロックに対するさらなる要求を行うことを制約され得る。条件が満たされる場合、ブロックに対するより多数の要求が出されてよく、すなわち、上の制約は当てはまらない。適切な条件の例は、ブロックのスケジューリングされた再生時間までの時間、またはその時間に依存する別の時間が与えられた閾値を下回ることである。ブロックを含むメディアデータの再生時間が近いために、ブロックの受信がより緊急になっているときに、ブロックに対するデータのための追加の要求が出されるので、この方法は有利である。HTTP/TCPのような一般的なトランスポートプロトコルの場合、これらの追加の要求は、問題のブロックの受信に寄与する、データに専用の利用可能な帯域幅の占有率を上げるという効果を有する。これは、ブロックを復元して完了するための十分なデータの受信に必要とされる時間を減らすので、ブロックを含むメディアデータのスケジューリングされた再生時間までにブロックが復元できない確率を下げる。上で説明されたように、ブロックを含むメディアデータのスケジューリングされた再生時間までにブロックが復元できない場合、再生は一時停止して悪いユーザ体験をもたらし得るので、ここで説明された方法は、有利なことに、この悪いユーザ体験の確率を下げる。30

#### 【0540】

本明細書の全体で、ブロックのスケジューリングされた再生時間への言及は、一時停止を伴わずにプレゼンテーションの再生を達成するために、ブロックを含む符号化メディアデータがクライアントにおいて初めて利用可能になり得る時間を指すことを理解されたい。メディアプレゼンテーションシステムの当業者には明らかなように、この時間は、実際には、再生のために使用される物理的な変換器(スクリーン、スピーカーなど)におけるブロックを含むメディアの出現の実際の時間よりもわずかに前であり、それは、ブロックの実際の再生を行うために、いくつかの変換機能がブロックを含むメディアデータに適用される必要があることがあり、これらの機能は、完了するにある時間を必要とし得るからである。たとえば、メディアデータは一般に、圧縮された形式でトランスポートされ、解凍変換が適用され得る。4050

**【 0 5 4 1 】**

協調的なHTTP/FEC方法をサポートするファイル構造を生成するための方法

協調的なHTTP/FEC方法を利用するクライアントによって有利に使用され得るファイル構造を生成するためのある実施形態が、ここで説明される。この実施形態では、各ソースセグメントに対して、次のように生成される対応する修復セグメントがある。パラメータRは、平均的にどれだけのFEC修復データがソースセグメント中のソースデータに対して生成されるかを示す。たとえば、 $R=0.33$ は、ソースセグメントが1,000キロバイトのデータを含む場合、対応する修復セグメントが約330キロバイトの修復データを含むことを示す。パラメータSは、FECの符号化および復号のために使用されるシンボルサイズをバイト単位で示す。たとえば、 $S=64$ は、ソースデータおよび修復データが、FECの符号化および復号の目的で各々64バイトのサイズのシンボルを含むことを示す。

10

**【 0 5 4 2 】**

修復セグメントは、次のようにソースセグメントに対して生成され得る。ソースセグメントの各フラグメントは、FEC符号化の目的ではソースブロックとして見なされるので、各フラグメントは、修復シンボルの生成元の、ソースブロックのソースシンボルの列として扱われる。最初の*i*個のフラグメントに対して生成される修復シンボルの総数は、 $TNRS(i) = ceiling(R*B(i)/S)$ として計算され、ここで、 $ceiling(x)$ は少なくともxである値を有する最小の整数を出力する関数である。したがって、フラグメント*i*に対して生成される修復シンボルの数は、 $NRS(i) = TNRS(i) - TNRS(i-1)$ である。

**【 0 5 4 3 】**

20

修復セグメントは、フラグメントに対する修復シンボルの連結物を含み、修復セグメント内での修復シンボルの順序は、修復シンボルの生成元のフラグメントの順序であり、フラグメント内では、修復シンボルは符号化シンボル識別子(ESI)の順序である。修復セグメント生成器2700を含む、ソースセグメント構造に対応する修復セグメント構造が図27に示される。

**【 0 5 4 4 】**

上で説明されたようにフラグメントに対する修復シンボルの数を定義することによって、すべての前のフラグメントに対する修復シンボルの総数、およびしたがって、修復セグメントの中のバイトインデックスは、R、S、B(*i*-1)、およびB(*i*)のみに依存し、ソースセグメント内のフラグメントの以前のまたは後続の構造のいずれにも依存しないことに留意されたい。これは、クライアントが修復セグメント内の修復ブロックの始めの位置を迅速に計算すること、さらに、修復ブロック内の修復シンボルの数を迅速に計算することを、修復ブロックの生成元のソースセグメントの対応するフラグメントの構造についてのローカル情報のみを使用して可能にするので、有利である。したがって、クライアントがソースセグメントの中央部からのフラグメントのダウンロードおよび再生を開始すると決断すると、クライアントはまた、対応する修復セグメント内からの対応する修復ブロックを迅速に生成しそれにアクセスすることができる。

30

**【 0 5 4 5 】**

フラグメント*i*に対応するソースブロック中のソースシンボルの数は、 $NSS(i) = ceiling((B(i)-B(i-1))/S)$ として計算される。B(*i*)-B(*i*-1)がSの倍数ではない場合、最後のソースシンボルは、FECの符号化および復号のために0のバイトをパディングされ、すなわち、最後のソースシンボルがFECの符号化および復号のためにSバイトのサイズとなるように、最後のソースシンボルはパディングされるが、これらの0のパディングバイトは、ソースセグメントの一部として記憶されない。この実施形態では、ソースシンボルのESIは0、1、…、 $NSS(i)-1$ であり、修復シンボルのESIはNSS(*i*)、…、 $NSS(i)+NRS(i)-1$ である。

40

**【 0 5 4 6 】**

この実施形態の修復セグメントのURLは、たとえば拡張子「.repair」をソースセグメントのURLに単に追加することによって、対応するソースセグメントのURLから生成され得る。

**【 0 5 4 7 】**

50

本明細書で説明されるように、修復インデクシング情報および修復セグメントのFEC情報は、対応するソースセグメントに対するインデクシング情報によって、ならびにRおよびSの値から、暗黙的に定義される。時間オフセット、および修復セグメントを含むフラグメント構造は、対応するソースセグメントの時間オフセットおよび構造によって決定される。フラグメント $i$ に対応する修復セグメント中の修復シンボルの終わりに対するバイトオフセットは、 $RB(i)=S*\text{ceiling}(R*B(i)/S)$ として計算され得る。そうすると、フラグメント $i$ に対応する修復セグメント中のバイトの数は $RB(i)-RB(i-1)$ なので、フラグメント $i$ に対応する修復シンボルの数は $NRS(i)=(RB(i)-RB(i-1))/S$ として計算される。フラグメント $i$ に対応するソースシンボルの数は、 $NSS(i)=\text{ceiling}((B(i)-B(i-1))/S)$ として計算され得る。したがって、この実施形態では、修復セグメント内の修復ブロックに対する修復インデクシング情報、および対応するFEC情報は、R、S、および対応するソースセグメントの対応するフラグメントに対するインデクシング情報から暗黙的に導出され得る。

#### 【0548】

ある例として、バイトオフセット $B(1)=6,410$ で開始しバイトオフセット $B(2)=6,770$ で終了するフラグメント2を示す、図28に示される例を考える。この例では、シンボルサイズは $S=64$ バイトであり、垂直方向の点線は、 $S$ の倍数に対応するソースセグメント内のバイトオフセットを示す。ソースセグメントサイズの断片としての全体の修復セグメントのサイズは、この例では $R=0.5$ に設定される。フラグメント2に対するソースブロック中のソースシンボルの数は、 $NSS(2)=\text{ceiling}((6,770-6,410)/64)=\text{ceil}(5.625)=6$ として計算され、これらの6つのソースシンボルはそれぞれESI<sub>s</sub> 0, ..., 5を有し、最初のソースシンボルは、ソースセグメント内のバイトインデックス6,410で開始するフラグメント2の最初の64バイトであり、2番目のソースシンボルは、ソースセグメント内のバイトインデックス6,474で開始するフラグメント2の次の64バイトであり、以下同様である。フラグメント2に対応する修復ブロックの終了バイトオフセットは、 $RB(2)=64*\text{ceiling}(0.5*6,770/64)=64*\text{ceiling}(52.89...)=64*53=3,392$ として計算され、フラグメント2に対応する修復ブロックの開始バイトオフセットは、 $RB(1)=64*\text{ceiling}(0.5*6,410/64)=64*\text{ceiling}(50.07...)=64*51=3,264$ として計算されるので、この例では、ESI 6および7を伴うフラグメント2に対応する修復ブロック中に、それぞれ、修復セグメント内のバイトオフセット3,264で開始しバイトオフセット3,392で終了する、2つの修復シンボルがある。

#### 【0549】

図28に示される例では、 $R=0.5$ であっても、フラグメント2に対応する6つのソースシンボルがあり、修復シンボルの数は、ソースシンボルの数を単に使用して修復シンボルの数を計算した場合に予測され得る3ではなく、代わりに、本明細書で説明される方法によれば2として計算されることに留意されたい。フラグメントのソースシンボルの数を単に使用して修復シンボルの数を決定することとは対照的に、上で説明された実施形態は、対応するソースセグメントの対応するソースブロックと関連付けられるインデックス情報のみから、修復セグメント内の修復ブロックの配置を計算することを可能にする。さらに、ソースブロック中のソースシンボルの数 $K$ が増えるにつれて、対応する修復ブロックの修復シンボル $KR$ の数は、ほぼ $K*R$ によって近似される。それは、一般に、 $KR$ は最大でも $\text{ceil}(K*R)$ であり、 $KR$ は少なくとも $\text{floor}((K-1)*R)$ であるからであり、ここで $\text{floor}(x)$ は最大でも $x$ である最大の整数である。

#### 【0550】

当業者が認識するように、協調的なHTTP/FEC方法を利用するクライアントによって有利に使用され得るファイル構造を生成するための上の実施形態の多くの変形がある。代替的な実施形態の例として、表現に対する元のセグメントが $N>1$ 個の並列セグメントへと区分されてよく、ここで、 $i=1, \dots, N$ に対して、元のセグメントの規定された断片 $F_i$ は $i$ 番目の並列セグメントに含まれ、 $i=1, \dots, N$ に対して、 $F_i$ の合計は1に等しい。この実施形態では、上で説明された実施形態で修復セグメントマップがソースセグメントマップから導出される方法と同様に、並列セグメントのすべてに対するセグメントマップを導出するために使用される1つのマスターセグメントマップがあり得る。たとえば、マスターセグメント

10

20

30

40

50

マップは、ソースメディアデータのすべてが並列セグメントに区分されず、代わりに1つの元のセグメントに含まれる場合、フラグメント構造を示すことができ、次いで、 $i$ 番目の並列セグメントに対するセグメントマップは、元のセグメントのフラグメントの最初のプレフィックス中のメディアデータの量が $L$ バイトである場合に、最初の $i$ 個の並列セグメント全体で、このプレフィックスのバイトの総数が $\text{ceil}(L^*G_i)$ であることを計算することによって、マスターセグメントマップから導出されることが可能であり、ここで、 $G_i$ は $F_j$ の $j=1, \dots, i$ にわたる合計である。代替的な実施形態の別の例として、セグメントは、各フラグメントに対する修復データが直後に続く、各フラグメントに対する元のソースメディアデータの組合せで構成されてよく、ソースメディアデータ、およびそのソースメディアデータからFEC符号を使用して生成された修復データの混合物を含むセグメントをもたらす。代替的な実施形態の別の例として、ソースメディアデータおよび修復データの混合物を含むセグメントは、ソースメディアデータおよび修復データの混合物を含む複数の並列セグメントへと区分され得る。

#### 【0551】

低レイテンシストリーミングを処理するための方法

いくつかの展開シナリオでは、ライブサービスのための低レイテンシストリーミングが望ましいことがある。たとえば、スポーツイベントまたはコンサートのような、イベントの局所的なその場での配信の場合、生の動作とクライアント上でのライブサービスのプレゼンテーションとの間の遅延は、可能な限り短いことが望ましい。たとえば、最大で1秒の遅延が望ましいことがある。

#### 【0552】

上で説明されたように、メディアプレゼンテーションのセグメントを記憶する各ファイルがランダムアクセスポイント(RAP)で開始するように並べることが有利であり得る。いくつかのプロファイル、具体的には、ISOベースのメディアファイルフォーマットのライブプロファイルは、各メディアセグメントがRAPで開始することを必要とする。

#### 【0553】

しかしながら、端末間レイテンシの少ない配信が必要とされる環境では、各セグメントの継続時間は、生の動作とクライアント上でのライブイベントのプレゼンテーションとの間の遅延を最小限にするために、短くなければならない。低レイテンシストリーミングのために使用されるべき各セグメントに、RAPを挿入するのは避けることが望ましい。たとえば、ビデオ中のRAPは、IDRフレームによって通常は実現される。符号化の効率は、低レイテンシストリーミングのために望ましい短いセグメント内でIDRフレームの使用を避けることによって、改善され得る。

#### 【0554】

ある実施形態によれば、メディアプレゼンテーションの、ライブプロファイルに準拠する表現および低レイテンシ表現が生成される。ライブプロファイルに準拠する表現は、比較的長いメディアセグメント継続時間有する。ライブプロファイルに準拠する表現の各メディアセグメントは、メディアセグメントの始めにRAPを有する。低レイテンシ表現は、RAPを含まないことがある、比較的短いセグメント(「メディアフラグメント」と呼ばれる)を有する。低レイテンシストリーミングをサポートするクライアントは、メディアプレゼンテーションの低レイテンシ表現に対して生成されるメディアフラグメントを受信することができるが、低レイテンシストリーミングをサポートしないクライアントは、メディアプレゼンテーションのライブプロファイルに準拠する表現のために生成されたメディアセグメントを受信することが可能であり得る。

#### 【0555】

図30は、低レイテンシストリーミングのためのメディアフラグメントとメディアフラグメントとの関係を示す。ライブプロファイルストリーミングのために生成されたメディアセグメント3002は、メディアデータ(「mdat」)の始めにRAP 3004を含む。対照的に、低レイテンシストリーミングのために生成されたメディアフラグメント3004、3006、および3008のうち、メディアフラグメント3004のみがRAPを含む。

10

20

30

40

50

**【0556】**

メディアフラグメントは、その場で生成され、HTTPを介したクライアントによるダウンロードが可能である。メディアフラグメントは、必要とされるメディアフラグメントに対する修正を何ら伴わずに、ISOベースのメディアファイルフォーマットのライブプロファイルに準拠するメディアセグメントに蓄積され得る。たとえば、メディアフラグメントは、メディアセグメントへと連結され得る。

**【0557】**

メディアセグメントおよびメディアフラグメントは両方、同じ符号化プロセスを使用して作成され得る。このようにして、メディアは、端末間の低レイテンシを要求する環境において動作するクライアント、および各セグメント中にRAPを必要とするプロトコルを使用するクライアントによる消費のために、効率的に符号化され得る。10

**【0558】**

いくつかの実施形態では、セグメントインデックス(SIDX)が、各メディアフラグメントに対して生成される。IDXは、メディアセグメント内のプレゼンテーション時間範囲と、メディアフラグメントにより専有されるメディアセグメントの対応するバイト範囲とを含み得る。いくつかの実施形態では、IDXは、RAPがフラグメント内に存在するかどうかを示す。図30では、メディアフラグメント3004のIDXボックスのcontains\_RAPフィールドは1に設定され、メディアフラグメント3004がRAPを含んでいることを示す。メディアフラグメント3006および3008のIDXボックスのcontains\_RAPフィールドは0に設定され、メディアフラグメント3006および3008がRAPを含んでいないことを示す。IDXはさらに、フラグメント内の最初のRAPのプレゼンテーション時間を示し得る。20

**【0559】**

ある実施形態によれば、メディアサーバは、低レイテンシストリーミングのためのフラグメントを生成し、フラグメントをキャッシュにプッシュすることができる。キャッシュは、フラグメントを連結して、ライブプロファイル適合メディアセグメントを生成することができる。メディアセグメントが生成された後、キャッシュは、メディアセグメントを生成するために連結されたメディアセグメントをページすることができる。

**【0560】**

単一のmedia presentation description(MPD)は、メディアプレゼンテーションのライブプロファイル適合メディアセグメントを有する第1の表現、および、低レイテンシストリームのメディアフラグメントを有する第2の表現についての情報を記憶し得る。タイムシフトバッファリングのためのメディアセグメントと、ストリーミングのライブに近い端部における視聴を処理するためのメディアフラグメントとを使用して、タイムシフト視聴が実現され得る。クライアントは、たとえば、タイムシフトバッファで開始し、メディアプレゼンテーションのセクションを飛ばすことによって、ライブの端部に近いところに移動して、これらの表現を切り替えることができる。MPDの各表現は、単一のメディアプレゼンテーションに利用可能な表現のアレイを表すための属性を割り当てられ得る。30

**【0561】**

メディアセグメントを有する第1の表現およびメディアフラグメントを有する第2の表現についての情報を記憶するMPDでは、第2の表現のどのメディアフラグメントがRAPで開始するかを示す情報を提供することが有利であり得る。たとえば、MPDは、複数のメディアフラグメント内でのRAPの発生の頻度を示すための属性を含み得る。一実施形態では、MPDは、フラグメントの数に関して頻度を示す属性を含む(すなわち、x個のメディアフラグメントごとに1個のメディアフラグメントがRAPを含む)。別の実施形態では、属性は、隣接するRAPの間の時間的な距離に関して頻度を示す。40

**【0562】**

あるいは、メディアフラグメントについての情報は第1のMPDに記憶されてよく、メディアセグメントについての情報は第2のMPDに記憶されてよい。

**【0563】**

いくつかの実施形態では、MPDは、プレゼンテーションのメディアセグメントまたはメ50

ディアフラグメントの最大継続時間のような、特定の表現に適用可能な特定のパラメータをシグナリングし得る。

【0564】

本開示を読んだ後で、当業者はさらなる実施形態を想起し得る。他の実施形態では、有利には、上で開示された発明の組合せまたは副次的な組合せが作られ得る。コンポーネントの例示的な構成は例示のために示されており、組合せ、追加、再構成などが、本発明の代替的な実施形態で考慮されることを理解されたい。したがって、本発明は、例示的な実施形態に関して説明されてきたが、当業者は、多くの修正が可能であることを認識するだろう。

【0565】

たとえば、本明細書で説明されたプロセスは、ハードウェアコンポーネント、ソフトウェアコンポーネント、および/またはこれらの任意の組合せを使用して実施され得る。いくつかの場合には、ソフトウェアコンポーネントは、メディア内に設けられた、またはメディアとは別々のハードウェア上の実行のために、有形な非一時的媒体上で提供され得る。したがって、本明細書および図面は、限定ではなく例示であると見なされるべきである。しかし、特許請求の範囲において述べられる本発明のより広範な趣旨および範囲から逸脱することなく、様々な修正および変更が行われてよく、本発明は、以下の特許請求の範囲内にあるすべての修正および等価物を包含することが意図されることが、明らかであろう。

【符号の説明】

【0566】

- 100 ブロックストリーミングシステム
- 101 ブロックサーピングインフラストラクチャ
- 102 コンテンツ
- 103 コンテンツ準備(メディア取込システム)
- 104 HTTPストリーミングサーバ
- 106 HTTPキャッシュ
- 108 HTTPストリーミングクライアント
- 110 コンテンツ記憶装置
- 112 要求
- 114 要求
- 122 ネットワーク
- 123 ブロック選択器
- 124 ブロック要求器
- 125 ブロックバッファ
- 126 バッファモニタ
- 127 メディアデコーダ
- 128 メディアトランステューサ
- 300 バス
- 302 取込システム
- 304 メモリ
- 306 ディスク記憶装置
- 308 ビデオディスプレイ
- 310 英数字入力デバイス
- 312 ネットワークインターフェース
- 400 バス
- 402 クライアントプロセッサ
- 404 メモリ
- 406 ディスク記憶装置
- 408 ビデオディスプレイ

10

20

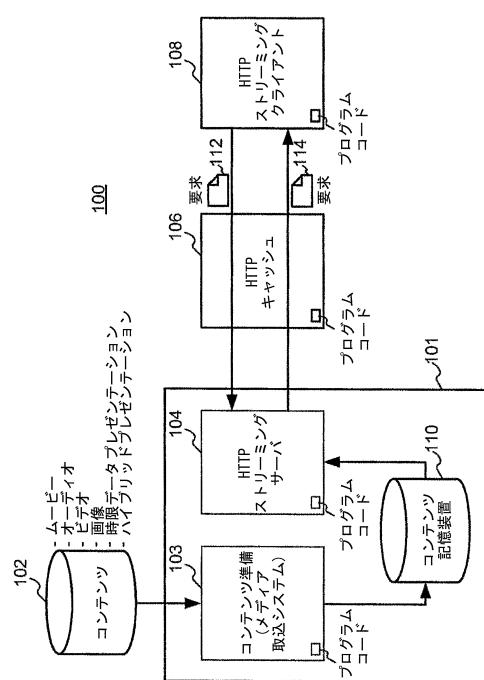
30

40

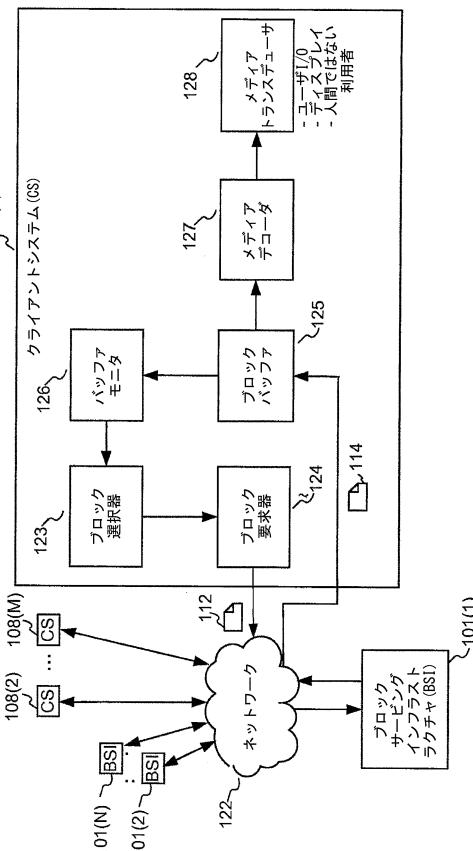
50

|      |                        |    |
|------|------------------------|----|
| 410  | 英数字入力デバイス              |    |
| 412  | ネットワークインターフェース         |    |
| 500  | MPD                    |    |
| 501  | 期間記録                   |    |
| 502  | 表現記録                   |    |
| 503  | セグメント情報                |    |
| 504  | 初期化セグメント               |    |
| 505  | メディアセグメント              |    |
| 510  | ソースセグメント               |    |
| 512  | 修復セグメント                | 10 |
| 700  | 単純なインデックス              |    |
| 712  | 階層的なインデックス             |    |
| 900  | メタデータテーブル              |    |
| 902  | HTTPストリーミングクライアント      |    |
| 904  | ブロック                   |    |
| 906  | HTTPストリーミングサーバ         |    |
| 1000 | ビデオストリーム               |    |
| 1200 | 送信機                    |    |
| 1202 | メタデータ                  |    |
| 1204 | スケーラブルレイヤ1             | 20 |
| 1206 | スケーラブルレイヤ2             |    |
| 1208 | スケーラブルレイヤ3(完全には受信されない) |    |
| 1210 | 受信機                    |    |
| 1212 | メディアプレゼンテーション          |    |
| 3002 | メディアセグメント              |    |
| 3004 | メディアフラグメント             |    |
| 3006 | メディアフラグメント             |    |
| 3008 | メディアフラグメント             |    |

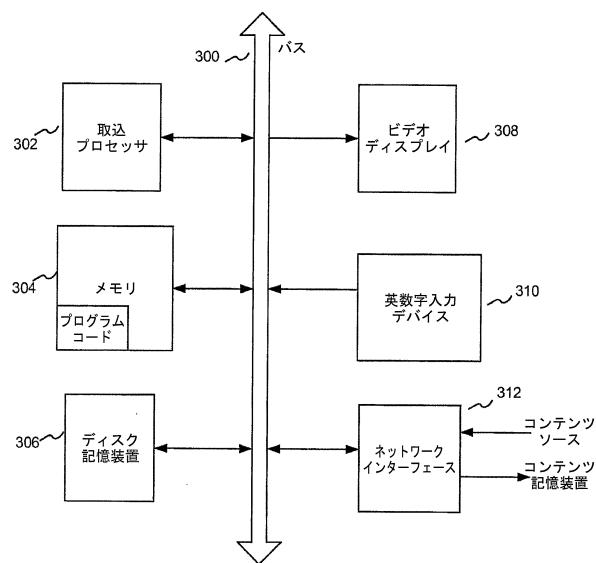
【図1】



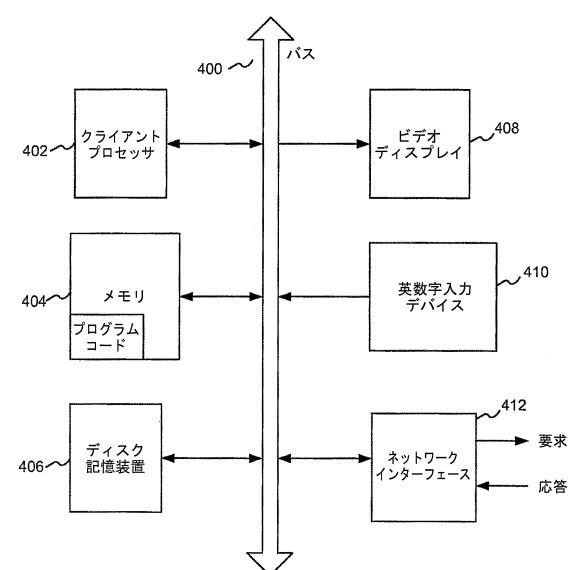
【図2】



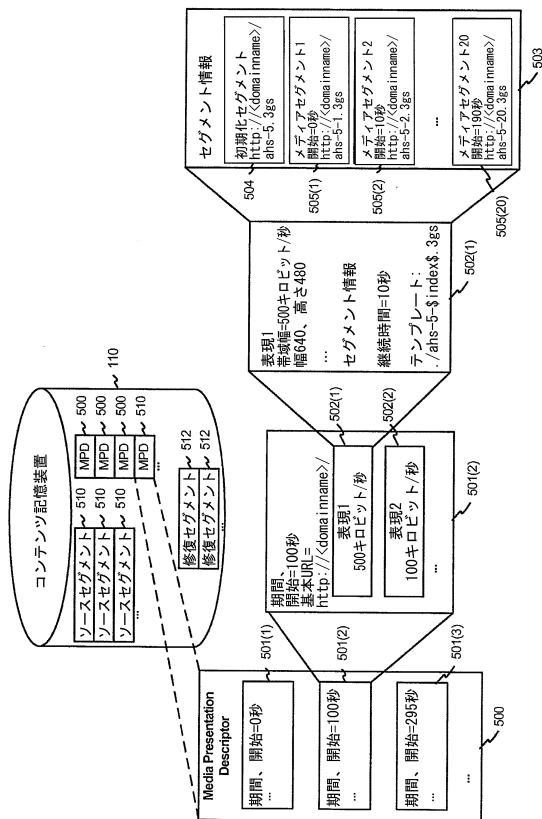
【 四 3 】



【図4】



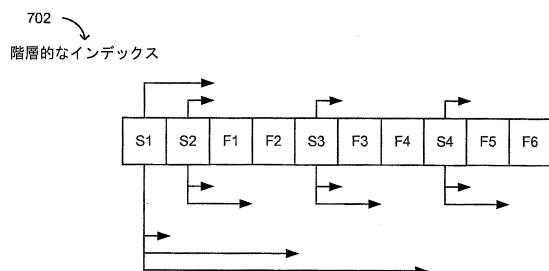
【図5】



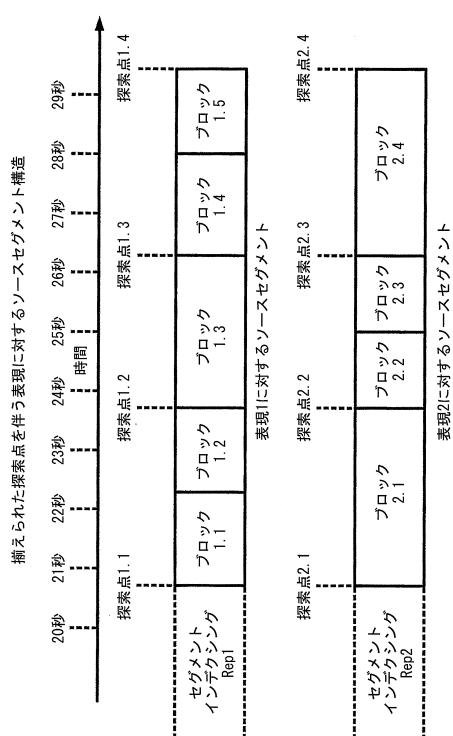
【図6】



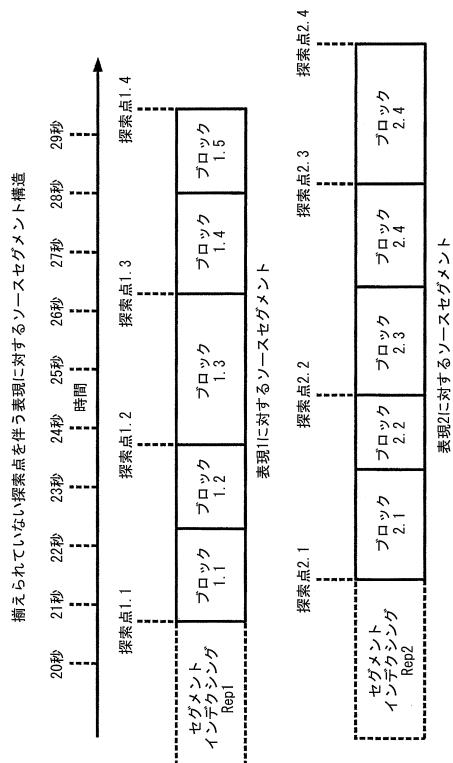
【図7(a)】



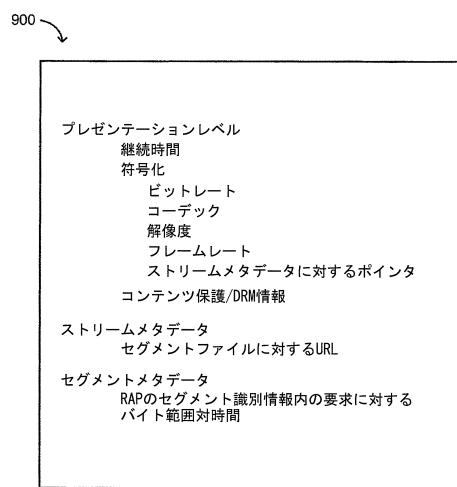
【図7(b)】



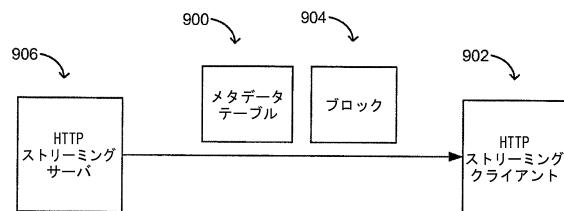
【図 8 ( b )】



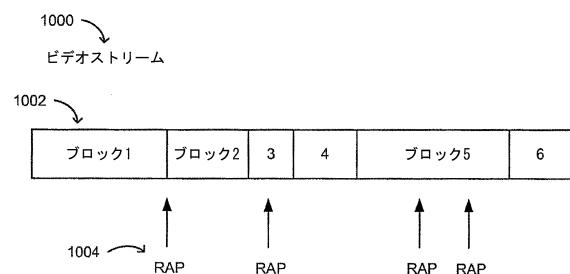
【図 9 ( a )】



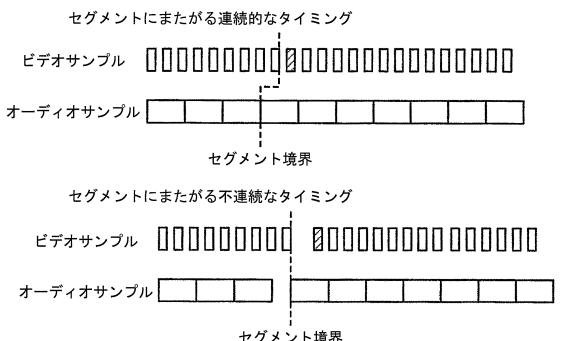
【図 9 ( b )】



【図 10】



【図 11】



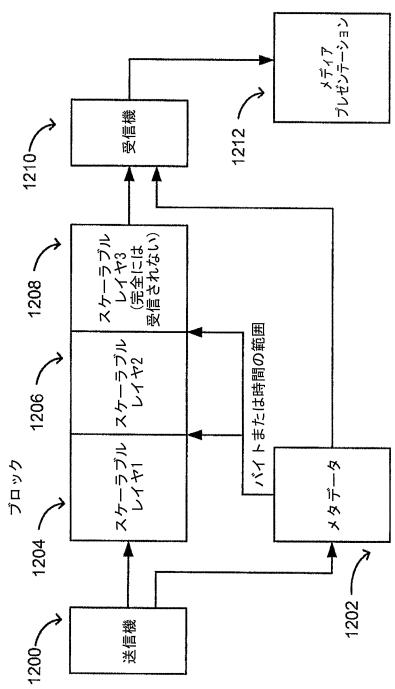
仮定:

- 独立の理由で境界点として選ばれた斜線付きのフレーム
- オーディオは決してビデオよりも遅く開始してはならない

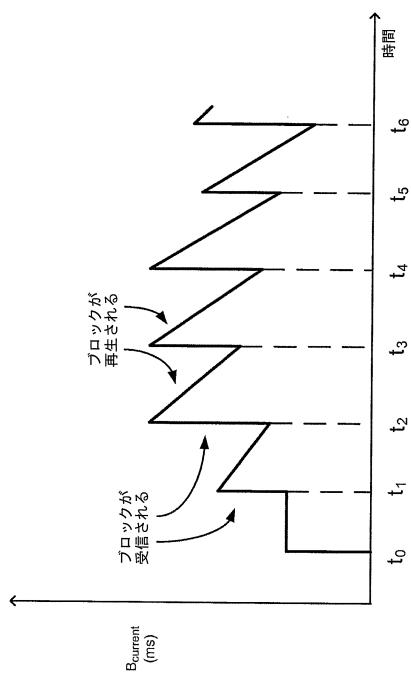
注意:

- 連続的な場合、2番目のセグメントは最初のセグメントと同じコンテンツである
- 不連続の場合、2番目のセグメントは最初のセグメントとは異なるコンテンツである

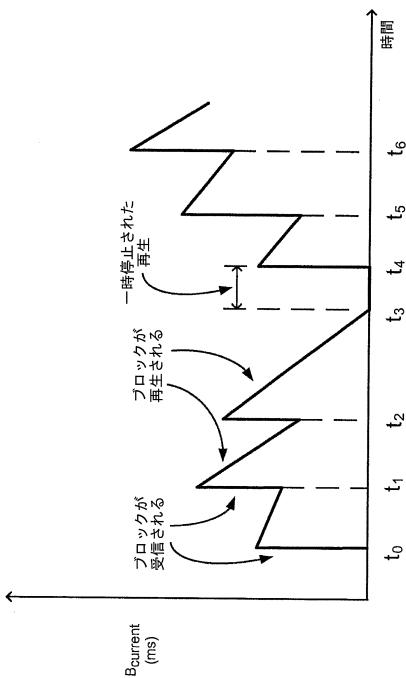
【図 1 2】



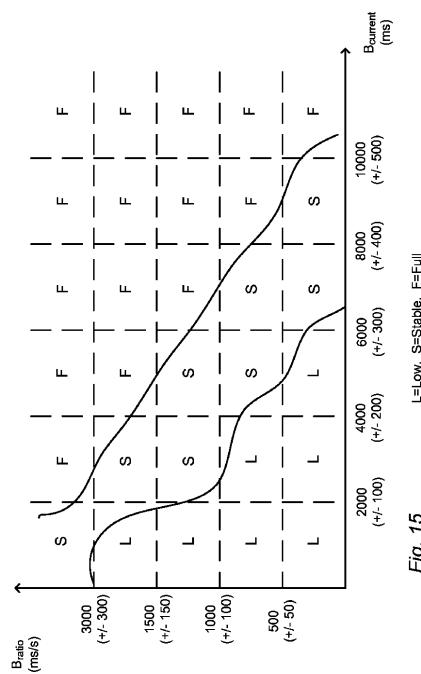
【図 1 3】



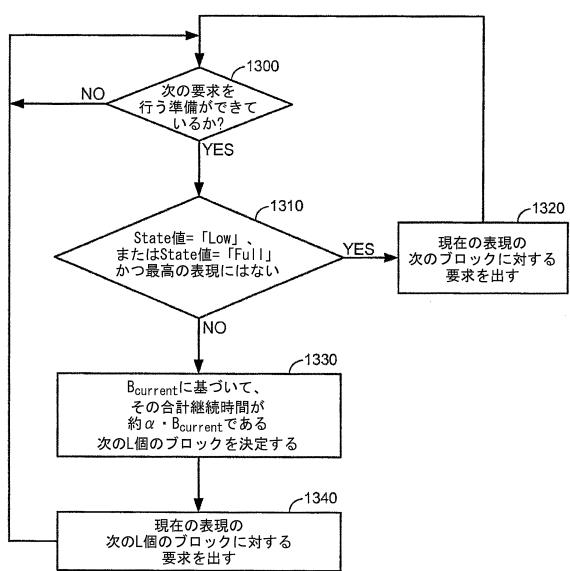
【図 1 4】



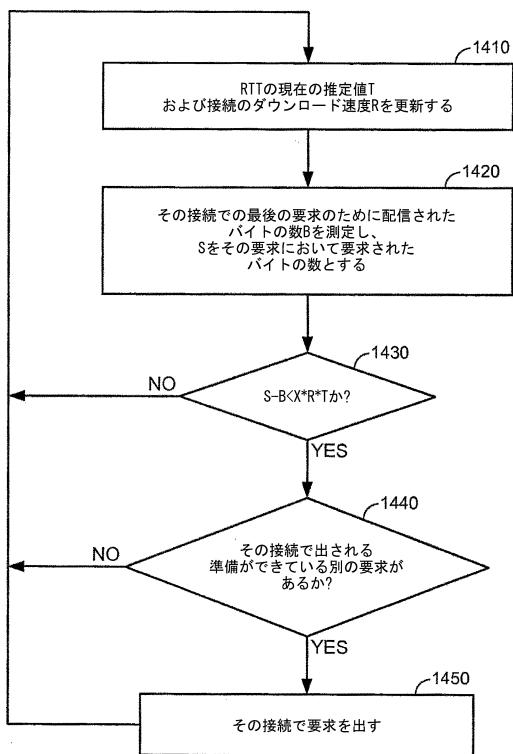
【図 1 5】

Fig. 15  
L=Low, S=Stable, F=Full

【図16】



【図17】



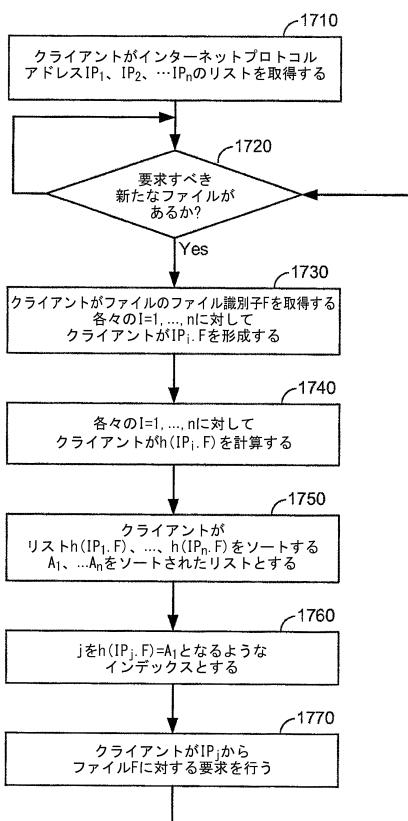
【図18】

| 候補要求リスト | 優先順位 | 接続1に対して接続OKか? | 接続2に対して接続OKか? | 接続3に対して接続OKか? |
|---------|------|---------------|---------------|---------------|
| 要求A     | 0    | X             |               |               |
| 要求B     | 0    | X             | X             |               |
| 要求C     | 1    | X             |               |               |
| 要求D     | 1    |               |               |               |
| 要求E     | 2    |               | X             |               |
| 要求F     | 3    |               | X             | X             |

【図19】

| 候補要求リスト | 優先順位 | 接続1に対して接続OKか? | 接続2に対して接続OKか? | 接続3に対して接続OKか? |
|---------|------|---------------|---------------|---------------|
| 要求A     | 5    | X             |               |               |
| 要求G     | 0    |               | X             |               |
| 要求C     | 4    | X             |               |               |
| 要求D     | 1    |               |               | X             |
| 要求E     | 2    |               |               | X             |
| 要求F     | 3    |               |               | X             |

【図20】



【図21】

```

<expression> ::= <literal> |
    <variables> |
    <function> |
    ('<expression>' ) |
    <expression><operator><expression>

<literal> ::= <string> | <number>

<variable> ::= <token>

<function> ::= <token> '(' [ <expression> * ',' <expression> ) ] ')'

<operator> ::= 1*<opchar>

<token> ::= <tokenchar> *(<tokenchar> | <digit>)

<string> ::= "" *<char> ""

<number> ::= '[' . 1*<digit> [ . 1*<digit> ]

<digit> ::= '0'-'9'

<char> ::= <any ASCII char except ""> | '/' ```

<tokenchar> ::= 'A'-'Z' | 'a'-'z' | '_'

<opchar> ::= '+' | '*' | '=' | '/' | '%' | '^' |

```

*FIG. 21*

【図22】

```
unsigned long Hash( const char *p, unsigned long max )
{
    unsigned long hash = 0;
    while( *p != 0 )
        hash = ( hash << 5 ) ^ ( ( hash & 0xf8000000 ) >> 27 ) ^ *p++;
    return hash % max;
}
```

FIG. 22

【図23】

|                          |   |
|--------------------------|---|
| <u>例1</u>                |   |
| <b>タイトル:</b>             | 単純な固定されたドメインマッピング                                   |
| <b>ファイル識別子<br/>構築規則:</b> | printf("http://<domainname>/%s_%3d.dfx", id, seq)   |
| <b>ファイルID:</b>           | SPEED_PAO_S2_E1_5gsyd75                             |
| <b>順序番号:</b>             | 22  |
| <b>結果:</b>               | http://<domainname>/SPEED_PAO_S2_E1_5gsyd75_022.dfx |

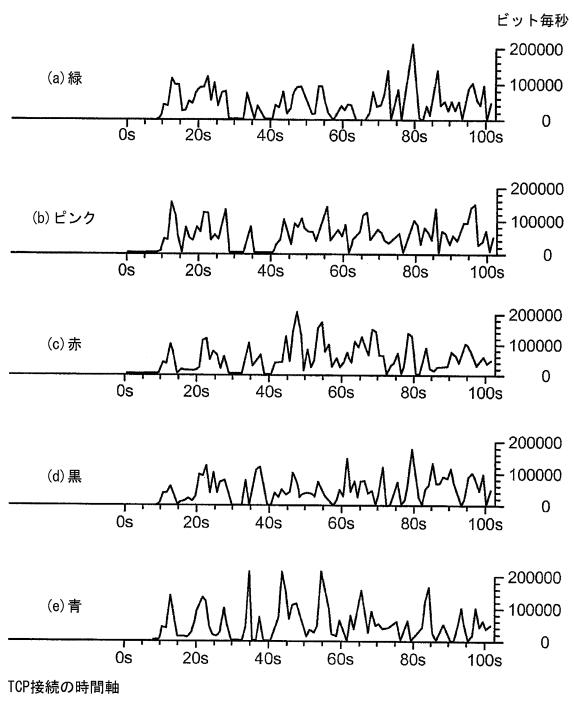
## 例2

|                  |   |
|------------------|---|
| タイトル:            | ファイルIDに基づく1000個のドメインにわたるハッシュ  |
| ファイル識別子<br>構築規則: | printf("http://%3d.<domainname>/%s_%3d.dfx", hash(id, 1000), id, seq) |
| ファイルID:          | SPEED_PAO_S2_E1_5gsy whole string                                     |
| 順序番号:            | 22  |
| 結果:              | http://564.<domainname>/SPEED_PAO_S2_E1_5gsy whole string .dfx        |

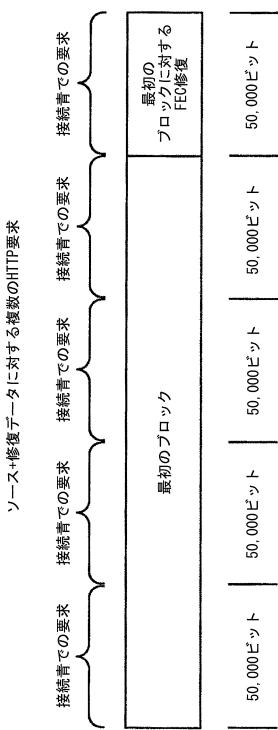
### 例3

|                  |  |
|------------------|--|
| タイトル:            | 10個のドメインおよび1000個のディレクトリにわたるハッシュ  |
| ファイル識別子<br>構築規則: | printf("http://%2d.<domainname>%3d/%s_%3d.dfx", hash(id, 10), hash(id, 1000), id, seq) |
| ファイルID:          | SPEED_PAO_S2_E1_5gsyd75  |
| 順序番号:            | 22   |
| 結果:              | http://4.<domainname>/564/SPEED_PAO_S2_E1_5gsyd75_022.dfx                              |

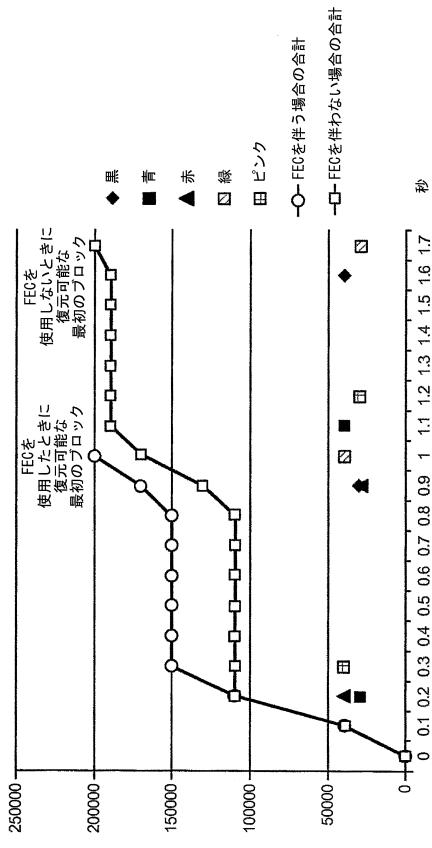
【図24】



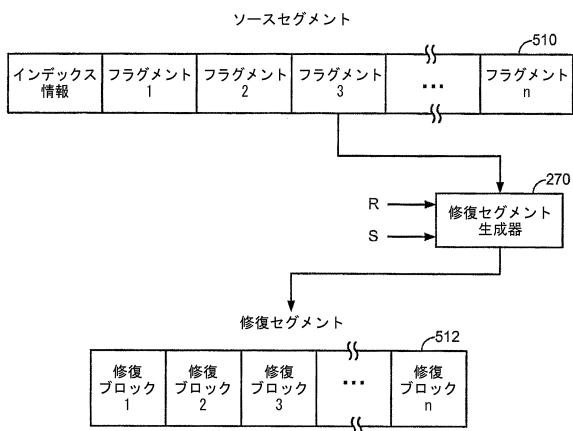
【図25】



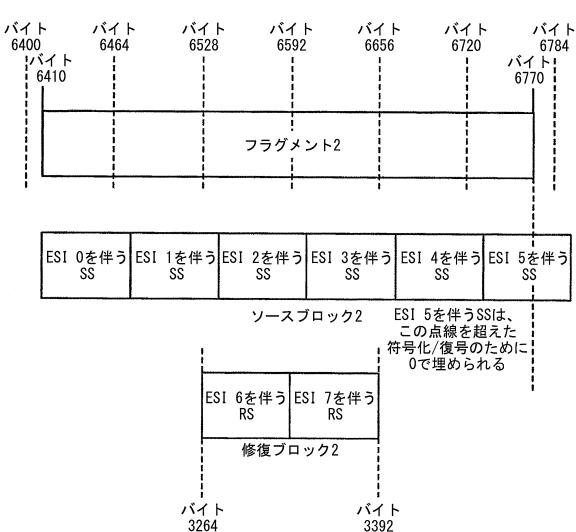
【図26】



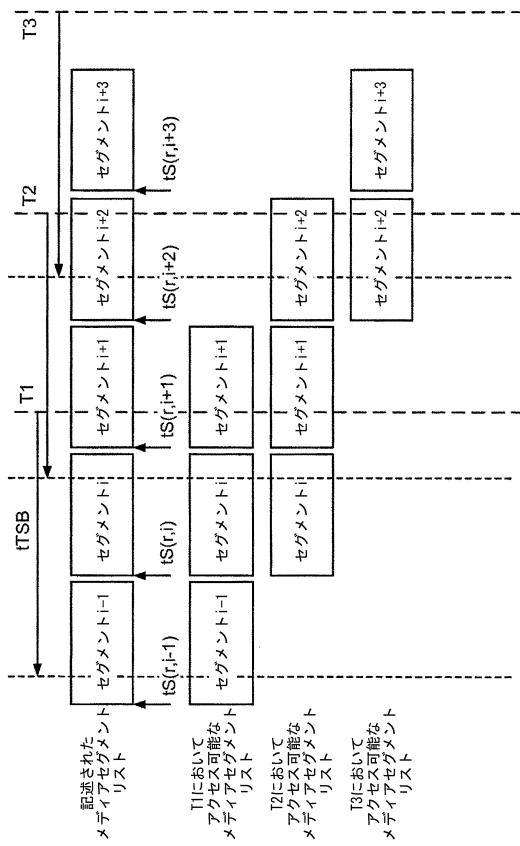
【図27】



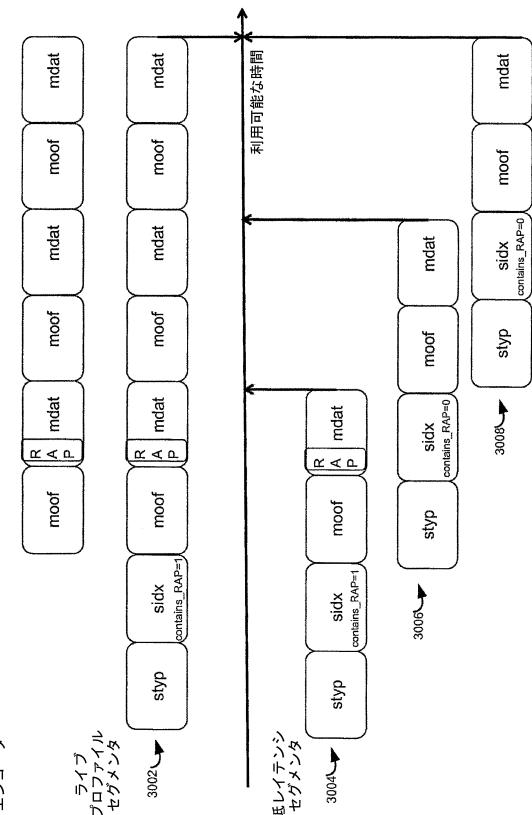
【図28】



【図29】



【図30】



---

フロントページの続き

- (72)発明者 マーク・ワトソン  
アメリカ合衆国・カリフォルニア・92121-1714・サン・ディエゴ・モアハウス・ドライヴ・5775
- (72)発明者 ロレンツォ・ヴィシザーノ  
アメリカ合衆国・カリフォルニア・92121-1714・サン・ディエゴ・モアハウス・ドライヴ・5775
- (72)発明者 パヤム・パクザド  
アメリカ合衆国・カリフォルニア・92121-1714・サン・ディエゴ・モアハウス・ドライヴ・5775
- (72)発明者 ピン・ワン  
アメリカ合衆国・カリフォルニア・92121-1714・サン・ディエゴ・モアハウス・ドライヴ・5775
- (72)発明者 イン・チェン  
アメリカ合衆国・カリフォルニア・92121-1714・サン・ディエゴ・モアハウス・ドライヴ・5775
- (72)発明者 トーマス・ストックハンマー  
アメリカ合衆国・カリフォルニア・92121-1714・サン・ディエゴ・モアハウス・ドライヴ・5775
- (72)発明者 ジャバー・モハンマド・ボラン  
アメリカ合衆国・カリフォルニア・92121-1714・サン・ディエゴ・モアハウス・ドライヴ・5775

審査官 久保 光宏

- (56)参考文献 國際公開第2011/038034 (WO, A1)  
米国特許出願公開第2012/0047280 (US, A1)  
Qualcomm Incorporated, "Discussion Paper on DASH Industry Profile including codecs and other interoperability components", [online], 3GPP TSG-SA4 #66, 2011年11月 5日, Document: S4-110880, [平成28年9月23日検索], インターネット, URL, [http://www.3gpp.org/ftp/tsg\\_SA/WG4\\_CODEC/TSGS4\\_66/Docs/S4-110880.zip](http://www.3gpp.org/ftp/tsg_SA/WG4_CODEC/TSGS4_66/Docs/S4-110880.zip)  
"ISO/IEC 23009-1:2012(E)", 2012年 4月 1日, First edition, 第1節~第5.2.2節, 第6.1節~第6.3.5.2節

- (58)調査した分野(Int.Cl., DB名)  
H04N21/00-21/858  
H04N5/91  
CSDB (日本国特許庁)  
IEEE Explorer (IEEE)