

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第6419173号
(P6419173)

(45) 発行日 平成30年11月7日(2018.11.7)

(24) 登録日 平成30年10月19日(2018.10.19)

(51) Int.Cl.

F 1

H04N 21/238 (2011.01)

H04N 21/238

H04N 21/438 (2011.01)

H04N 21/438

請求項の数 17 (全 71 頁)

(21) 出願番号	特願2016-524842 (P2016-524842)
(86) (22) 出願日	平成26年7月11日 (2014.7.11)
(65) 公表番号	特表2016-531466 (P2016-531466A)
(43) 公表日	平成28年10月6日 (2016.10.6)
(86) 国際出願番号	PCT/EP2014/064949
(87) 国際公開番号	W02015/004276
(87) 国際公開日	平成27年1月15日 (2015.1.15)
審査請求日	平成29年7月10日 (2017.7.10)
(31) 優先権主張番号	1312561.2
(32) 優先日	平成25年7月12日 (2013.7.12)
(33) 優先権主張国	英国(GB)
(31) 優先権主張番号	1312547.1
(32) 優先日	平成25年7月12日 (2013.7.12)
(33) 優先権主張国	英国(GB)

(73) 特許権者	000001007 キヤノン株式会社 東京都大田区下丸子3丁目30番2号
(74) 代理人	100126240 弁理士 阿部 琢磨
(74) 代理人	100124442 弁理士 黒岩 創吾
(72) 発明者	ユアン フアブレ フランス国 レンヌー・アタラント, セデ ックス セッソン-セヴィニエ 3551 7, リュ ドウ ラ トゥッシュ-ランペ ール キヤノン リサーチセンター フラ ンス エス. エー. エス. 内

最終頁に続く

(54) 【発明の名称】プッシュメッセージ制御による適応型データストリーミング方法

(57) 【特許請求の範囲】

【請求項 1】

メディアセグメントをクライアントへ送信するサーバであって、
前記クライアントから第1メディアセグメントのリクエストを受信する受信手段と、
前記受信手段によるリクエストの受信に応じて、前記クライアントに対して前記第1メディアセグメントを送信する送信手段と、
前記第1メディアセグメントのリクエストを送信した前記クライアントに対してプッシュすべき第2メディアセグメントを、前記クライアントとの間で共有されたプッシュストラテジーを用いることによって特定する特定手段と、
前記特定手段により特定された前記第2メディアセグメントを前記クライアントに対してプッシュするプッシュ手段とを有するサーバ。

【請求項 2】

前記送信手段は、前記特定手段により特定された前記第2メディアセグメントを識別するための1又は複数のメッセージを前記クライアントに対してさらに送信する請求項1に記載のサーバ。

【請求項 3】

前記クライアントとの間で共有される前記プッシュストラテジーは、前記サーバによる前記第2メディアセグメントの送信順序を規定し、前記送信手段は、前記送信順序に関する情報を前記クライアントに対してさらに送信する請求項1又は2に記載のサーバ。

【請求項 4】

10

20

前記プッシュストラテジーの共有は前記クライアントから前記サーバに対する情報の送信によりなされる請求項 1 乃至 3 のうち何れか 1 項に記載のサーバ。

【請求項 5】

前記プッシュストラテジーの共有は前記サーバから前記クライアントに対する情報の送信によりなされる請求項 1 乃至 3 のうち何れか 1 項に記載のサーバ。

【請求項 6】

前記情報は、H T T P フレームのヘッダに格納される請求項 4 又は 5 に記載のサーバ。

【請求項 7】

サーバからメディアセグメントを受信するクライアントであって、

前記サーバに対して第 1 メディアセグメントのリクエストを送信する送信手段と、

前記送信手段により送信されたリクエストに応答して前記サーバから送信される前記第 1 メディアセグメントを受信する受信手段と、

前記サーバとの間で共有されたプッシュストラテジーを用いることによって、前記サーバが前記第 1 メディアセグメントをリクエストした前記クライアントに対してプッシュすべき第 2 メディアセグメントを決定する決定手段とを有するクライアント。

【請求項 8】

前記プッシュストラテジーを決定する決定手段と、

前記決定手段により決定されたプッシュストラテジーを示す情報を前記共有のために前記サーバへ通知する通知手段とを有する請求項 7 に記載のクライアント。

【請求項 9】

前記プッシュストラテジーの共有は前記サーバから前記クライアントに対する情報の送信によりなされる請求項 7 又は 8 に記載のクライアント。

【請求項 10】

前記情報は、H T T P フレームのヘッダに格納される請求項 8 又は 9 に記載のクライアント。

【請求項 11】

前記決定手段は、前記第 1 メディアセグメントと、前記プッシュストラテジーとに基づいて前記第 2 メディアセグメントを決定する請求項 7 乃至 10 のうち何れか 1 項に記載のクライアント。

【請求項 12】

前記第 2 メディアセグメントを識別するための識別情報と、前記第 2 メディアセグメントの送信順序に関する順序情報とを前記サーバから受信する情報受信手段と、

前記情報受信手段により受信された順序情報に応じた送信順序を変更すべきか否かを判定する判定手段と、

前記判定手段による判定結果に応じて前記サーバに対して変更要求を送信する要求手段とを有する請求項 7 乃至 11 のうち何れか 1 項に記載のクライアント。

【請求項 13】

通信装置から他の通信装置に対するメディアセグメントの送信方法であって、

前記通信装置の受信手段が、前記他の通信装置から第 1 メディアセグメントのリクエストを受信する受信工程と、

前記通信装置の受信手段による前記リクエストの受信に応じて、前記通信装置の送信手段が、前記他の通信装置に対して前記第 1 メディアセグメントを送信する送信工程と、

前記第 1 メディアセグメントのリクエストを送信した前記他の通信装置に対してプッシュすべき第 2 メディアセグメントを、前記通信装置と前記他の通信装置との間で共有されたプッシュストラテジーを用いることによって前記通信装置のプロセッサが特定する特定工程と、

前記通信装置のプロセッサにより特定された前記第 2 メディアセグメントを前記他の通信装置に対してプッシュするプッシュ工程とを有する送信方法。

【請求項 14】

前記特定された前記第 2 メディアセグメントを識別するための 1 又は複数のメッセージ

10

20

30

40

50

を、前記プッシュ工程におけるプッシュ前に前記通信装置の送信手段が前記他の通信装置に対して送信する第2送信工程をさらに有する請求項13に記載の送信方法。

【請求項15】

前記プッシュストラテジーは、前記通信装置から前記他の通信装置に対する情報の通知により共有される請求項13又は14に記載の送信方法。

【請求項16】

コンピュータを請求項1乃至6のうち何れか1項に記載のサーバの各手段として動作させるためのプログラム。

【請求項17】

コンピュータを請求項7乃至12のうち何れか1項に記載のクライアントの各手段として動作させるためのプログラム。 10

【発明の詳細な説明】

【関連出願の相互参照】

【0001】

本出願は、英國特許出願第1312547.1号および第1312561.2号（ともに2013年7月12日）、並びに第1410540.7号（2014年6月12日）からの優先権を主張し、それらは参考文献としてすべて本明細書に援用される。

【技術分野】

【0002】

本発明は、HTTP通信ネットワーク上のデータストリーミングに関する。 20

【0003】

特に、本発明はネットワーク制約を満たすための適応型データストリーミングに関する。本発明は、DASHネットワークにおけるアプリケーションを有してもよい。

【0004】

DASH (Dynamic Adaptive Streaming over HTTP (動的適応型HTTPストリーミング)の頭文字)は、HTTP上のメディアコンテンツストリーミング（通常はオーディオ/ビデオコンテンツ）を可能にする通信規格である。DASHによれば、メディアプレゼンテーションは、「メディアプレゼンテーション記述」ファイル（以下において、MPD）と呼ばれるXMLファイルとして記載される。MPDファイルは、クライアント装置がメディアコンテンツの配信を要求し制御することを可能にする情報を、クライアント装置に提供する。 30

【背景技術】

【0005】

HTTPメディアストリーミングの一般的原理は、図3に示される。適応型HTTPメディアストリーミングのための新たなプロトコルおよび標準のほとんどは、この原理に基づく。

【0006】

メディアサーバ300は、クライアント310に対してデータをストリーミングする。メディアサーバは、メディアプレゼンテーションを保存する。例えば、メディアプレゼンテーション301は、オーディオデータおよびビデオデータを含む。オーディオおよびビデオは、同じファイル内でインターリープされてもよい。メディアプレゼンテーションが構築される方法は、図4aを参照しながら以下に記載される。メディアプレゼンテーションは、独立してアドレス指定されてダウンロードされることができるMP4セグメントなどの、小さな独立した連続する時間セグメント302a、302bおよび302cに一時的に分割される。これらの時間セグメントの各々のためのメディアコンテンツのダウンロードアドレス(HTTP URL)は、クライアントに対してサーバによって設定される。オーディオ/ビデオメディアコンテンツの各時間セグメントは、1つのHTTPアドレスに関連づけられている。

【0007】

メディアサーバは、また、メディアコンテンツ特性（例えば、メディアのタイプ：オー 50

ディオ、ビデオ、オーディオビデオ、テキストなど)、コード化形式(例えば、ビットレート、タイミング情報など)、時間メディアセグメントと関連するURLとのリストを含むメディアプレゼンテーションのコンテンツを記述する(図5を参照ながら以下に記載される)マニフェストファイルドキュメント304を保存する。または、ドキュメントは、時間メディアセグメント及び関連するURLとの明示的なリストを再構築可能にするテンプレート情報を含む。このドキュメントは、エクステンシブルマークアップランゲージ(XML)を用いて記載されてもよい。

【0008】

マニフェストファイルは、クライアントに対して送信される。ステップ305の間にマニフェストファイルの受信をすると、クライアントには、メディアコンテンツの時間セグメントとHTTPアドレスとの間の関連づけが通知される。また、マニフェストファイルは、メディアプレゼンテーションのコンテンツに関する情報(本例ではインターリーブされたオーディオ/ビデオ)をクライアントに提供する。情報は、解像度、ビットレートなどを含んでもよい。

10

【0009】

受信された情報に基づいて、クライアントのHTTPクライアントモジュール311は、マニフェストファイル内に記載されたメディアコンテンツの時間セグメントのダウンロードのためにHTTP要求306を発行することができる。サーバのHTTP応答307は、要求された時間セグメントを伝達する。HTTPクライアントモジュール311は、応答から時間メディアセグメントを抽出し、メディアエンジン312の入力バッファ307に対してそれらを提供する。最終的に、メディアセグメントは、ステップ308および309のそれぞれの間に復号されて表示ができる。

20

【0010】

メディアエンジン312は、次の時間セグメントが適切な時期に発行されるように要求するために、DASH制御エンジン313と情報のやりとりをする。次のセグメントは、マニフェストファイルから識別される。要求が発行される時期は、受信バッファ307が満杯であるか否かに依存する。DASH制御エンジン313は、バッファが過負荷になるかまたは完全に空になるのを防ぐようにバッファを制御する。

【0011】

メディアプレゼンテーションおよびマニフェストファイルの生成を、図4aを参照して説明する。ステップ400および401の間に、オーディオデータおよびビデオデータが取得される。次に、オーディオデータは、402の間に圧縮される。例えば、MP3規格を用いることができる。また、ビデオデータは、ステップ403の間に並行して圧縮される。MPEG4、MPEG/AVC、SVC、HEVC、またはスケーラブルHEVCなどのビデオ圧縮アルゴリズムを用いることができる。オーディオデータおよびビデオデータの圧縮が一旦実行されれば、オーディオビデオエレメンタリーストリーム404およびビデオエレメンタリーストリーム405は、利用可能になる。エレメンタリーストリームは、ステップ406の間に、グローバルメディアプレゼンテーションにカプセル化される。例えば、ISO-BMFF規格(またはISO-BMFF規格をAVC、SVC、HEVC、HEVCのスケーラブル拡張などに拡張する)は、符号化されたオーディオエレメンタリーストリームおよびビデオエレメンタリーストリームのコンテンツをグローバルメディアプレゼンテーションとして記述するために使用することができる。それによって取得されたカプセル化されたメディアプレゼンテーション407は、ステップ408の間に、XMLマニフェストファイル409を生成するために用いられる。ビデオデータ401およびオーディオデータ400のいくつかのリプレゼンテーションは、取得され、圧縮され、カプセル化され、メディアプレゼンテーション407内に記載することができる。

30

【0012】

図4bに示されたMPEG/DASHストリーミングプロトコルの特定のケースについて、マニフェストファイルは、「メディアプレゼンテーション記述」(または「MPD」ファイル)と呼ばれる。ファイルのルート要素は、プロファイルまたはスキーマのような

40

50

D A S H 情報を加えたすべてのプレゼンテーションに適用する属性を含む M P D 要素である。メディアプレゼンテーションは、*P e r i o d* 要素によって表される時間的期間に分割される。M P D ファイル 4 1 0 は、各時間的期間に関連するデータをすべて含む。この情報を受信することによって、クライアントは、各々の期間ごとのコンテンツを認識する。*P e r i o d* 4 1 1 ごとに、*A d a p t a t i o n S e t* (アダプテーションセット) 要素が定義される。

【 0 0 1 3 】

可能な構成は、プレゼンテーション内に含まれるメディアタイプ当たり1つ以上のAdaptation Setを有することである。ビデオに関連するAdaptationset412は、サーバにおいて利用可能な符号化されたビデオの異なる可能的なリプロゼンテーションに関する情報を含む。各リプロゼンテーションは、Representation要素内に記述される。例えば、第1のリプロゼンテーションは、 640×480 の空間解像度で符号化され、且つ500kビット/秒のビットレートで圧縮されたビデオになり得る。第2のリプロゼンテーションは、同じビデオになり得るが、250kビット/秒のビットレートで圧縮され得る。

〔 0 0 1 4 〕

クライアントが各ビデオに関するHTTPアドレスを知っている場合、各ビデオは、その後、HTTP要求によってダウンロードすることができる。各リプレゼンテーションのコンテンツとHTTPアドレスとの間の関連づけは、付加的なレベルの記述：時間セグメントを用いることによって行われる。各ビデオリプレゼンテーションは、時間セグメント413（通常は数秒）に分割される。各時間セグメントは、HTTPアドレス（URLまたは1バイトレンジのURL）を介してアクセス可能なサーバに保存されたコンテンツを備える。いくつかの要素は、MPDファイル：SegmentList（セグメントリスト）、SegmentBase（セグメントベース）またはSegmentTemplate（セグメントテンプレート）内に時間セグメントを記述するために用いることができる。

[0 0 1 5]

さらに、特定のセグメント：イニシャライゼーションセグメントが利用可能である。イニシャライゼーションセグメントは、（ISO BMFF またはその拡張を用いてビデオがカプセル化されているならば）カプセル化されたビデオストリームを記述するMP4イニシャライゼーション情報を含む。例えば、それは、クライアントがビデオに関する復号アルゴリズムをインスタンス化するのを支援する。

[0 0 1 6]

イニシャライゼーションセグメントおよびメディアセグメントのHTTPアドレスは、MPDファイル内に示される。

[0 0 1 7]

図 5 に、例示的な MPD ファイルが示される。2 つのメディアが、図示された MPD ファイル内に記述される。第 1 のものは英語のオーディオストリームであり、第 2 のものはビデオストリームである。英語のオーディオストリームは、Adaptation Set タグ 500 を用いて導入される。2 つの代替的リプレゼンテーションが、このオーディオストリームのために利用可能である。

[0 0 1 8]

・第1のリプレゼンテーション501は、64000ビット／秒のピットレートでMP4にカプセル化されたエレメンタリオーディオストリームである。この（MP4構文解析後の）エレメンタリストリームを取り扱うために用いられるコーデックは、値「mp4a.0x40」を有する属性コーデックによって規格内に定義されている。それは、セグメント階層内のBaseURL（ベースURL）要素：相対的なURIである<BaseURL>7657412348.mp4</BaseURL>の連結によって形成されたアドレスにおける要求を介してアクセス可能である。「http://cdn1.example.com/」によって、または「http://cdn2.example.co

m /」によってMPD要素内のトップレベルにおいて定義される BaseURL（2つのサーバが同じコンテンツのストリーミングのために利用可能である）は、絶対URIである。そして、クライアントは、その要求から、アドレス「`http://cdn1.example.com/7657412348.mp4`」またはアドレス「`http://cdn2.example.com/7657412348.mp4`」に対して、英語のオーディオストリームを要求することができる。

【0019】

・第2のリプレゼンテーション502は、32000ビット/秒のビットレートでMPDにカプセル化されたエレメンタリオーディオストリームである。第1のリプレゼンテーション501と同じような説明を行うことができ、クライアント装置は、したがって、以下のアドレスのいずれか1つにおける要求によって、この第2のリプレゼンテーション502を要求することができる。
10

【0020】

「`http://cdn1.example.com/3463646346.mp4`」または
 「`http://cdn2.example.com/3463646346.mp4`」。
。

【0021】

ビデオに関するアダプテーションセット503は、6つのリプレゼンテーションを含む。これらのリプレゼンテーションは、異なる空間解像度（ 320×240 、 640×480 、 1280×720 ）、および異なるビットレート（ $256000 \sim 2048000$ ビット/秒）のビデオを含む。これらのリプレゼンテーションの各々に対して、それぞれのURLは、BaseURL要素を通じて関連づけられる。そのため、クライアントは、推定された帯域幅、スクリーン解像度などの異なる基準にしたがって同じビデオのこれらの代替リプレゼンテーションの間で選択することができる。（図5において、時間セグメントへのRepresentationの分解は、明瞭にするために図示されていない、ということに留意されたい。）
20

図5aは、DASHクライアントの標準的挙動（standard behavior）を示す。図5bは、図4aに示された方法において用いられる、例示的なマニフェストファイル（記述ファイルまたはMPD）のツリー表現を示す。
30

【0022】

ストリーミングセッションを開始する場合、DASHクライアントは、マニフェストファイルを要求することによって開始する（ステップ550）。サーバの応答を待ってマニフェストファイルを受信した後（ステップ551）、クライアントは、マニフェストファイルを解析し（ステップ552）、その環境に適したAdaptation SetsのセットAS_{i,j}を選択し（ステップ553）、それから、例えば、その帯域幅、復号化、およびレンダリング機能に適したMPD内のRepresentationを各Adaptation Set AS_{i,j}内で選択する（ステップ554）。

【0023】

そして、DASHクライアントは、メディア復号器のためのイニシャライゼーション情報を始めとして、要求するセグメントのリストを予め構築することができる。このイニシャライゼーションセグメントは、それが複数のリプレゼンテーション、アダプテーションセットおよび期間に共通になり得るか、または各Representationに特有になり得るか、または第1のメディアセグメント内にも含まれ得るので、このMPD（ステップ555）内で、識別されなければならない。
40

【0024】

そして、クライアントは、イニシャライゼーションセグメントを要求する（ステップ556）。一旦イニシャライゼーションセグメントが受信されると（ステップ557）、復号器が起動される（ステップ558）。

【0025】

そして、クライアントはセグメント単位で第1のメディアデータを要求し（ステップ560）、復号化を実際に開始して表示する（ステップ563）前に、（ステップ559の条件のために）最小データ量をバッファする。MPDダウンロードと最初に表示されたフレームとの間のこれら複数の要求／応答は、ストリーミングセッションにおける起動遅延（start up delay）を招く。これらの初期ステップの後、DASHストリーミングセッションは、標準的方法で継続する、すなわち、DASHクライアントは、メディアセグメントを順々に適応させてリクエストする。

【0026】

現在のDASHバージョンは、マニフェストファイル内部の関心領域の記述を提供しない。いくつかのアプローチが、このような記述のために、提案してきた。

10

【0027】

特に、メディアコンテンツのコンポーネントは、SubRepresentation（サブリプレゼンテーション）要素を用いて記載することができる。これらの要素は、リプレゼンテーション内に埋め込まれる1つまたはいくつかのコンポーネントのプロパティを記述する。図6に、タイルトラック（tile tracks）をビデオのコンポーネントとして記述するDASHマニフェストファイルの例が示される。簡明に且つおよび明瞭にするために、1つのPeriod 600のみを示す。しかしながら、後続の期間要素は、同じ方式で編成されるだろう。部分601において、第1のアダプテーションセット要素は、スケーラブルビデオのベース層を記述するために用いられる。例えば、ビデオは、SVCまたはスケーラブルなHEVCにしたがって符号化される。部分602において、第2のアダプテーションセットはスケーラブルビデオの最高解像度層を記述するために用いられる。非スケーラブルビデオに対して、第2のアダプテーションセット602のみが、ベース層に依存することなく（すなわちdependency1d属性）、存在するだろう。この第2のアダプテーションセット602において、単一のリプレゼンテーション603、すなわち表示することができるビデオに対応するもの、が記述される。そのリプレゼンテーションは、クライアント要求のためのそれぞれのURLとともにセグメント610のリストとして記載される。

20

【0028】

このように、リプレゼンテーションは、「R1」（dependency1d属性）によって識別された別のリプレゼンテーション、実際には第1のアダプテーションセット601からのベース層リプレゼンテーション、に依存する。依存性は、エンハンスマント層のために現在のセグメントを得る前にベース層のための現在のセグメントを最初に要求することをストリーミングクライアントに強制する。このように参照されるトラックがクライアントによって自動的にロードされるので、これをタイルトラックに対する依存性を表現するためには用いることができない。メディアプレゼンテーションの間にユーザにとって興味のあるタイル（tiles）をいつでも選択することはユーザ次第であるので、これは回避されるべきである。そのため、複合トラックとタイルトラックとの間の依存性を示すために、SubRepresentation要素が用いられる。表示可能なビデオは、サブリプレゼンテーション604～608のリストとして記載される。各サブリプレゼンテーションは、カプセル化されたMP4ファイル内のトラックを実際に表わす。このように、複合トラック608に対して1つのタイルを加えた（本例においては4つのタイル）サブリプレゼンテーション当たり1つのサブリプレゼンテーションがある。各サブリプレゼンテーションは、タイルトラック614、615、616、および617、または複合トラック618に対応するか否かを示すために、コンテンツコンポーネント要素614～618によって記述される。DASH/MPDにおいて利用可能なロール記述子型は、特定のスキームによるタイリングのために用いられる。ロール記述子は、また、フルフレームビデオ内のタイルの位置を示す。例えば、コンポーネント614は、ビデオの左上（先頭行および先頭列の1：1）に位置するタイルを記述する。タイルの寸法、幅および高さは、MPDによって可能になるようなサブリプレゼンテーションの属性として指定される。帯域幅情報も、また、その帯域幅にしたがって、タイルの数およびタイルの選択の

40

50

決定において D A S H クライアントを支援するためにここに設けることができる。複合トラックに関して、ダウンロードの終わりに、復号することができるビデオストリームを構築することができることが義務的であるので、タイルトラックとは異なる方法で、信号で送らなければならない。その目的のために、2つの要素が記述に加えられる。第1に、関連コンテンツコンポーネント 618 内の記述子は、それがすべてのコンポーネントの中の主要なコンポーネントであることを示す。第2に、対応するデータが要求されるべきであることをクライアントに対して示すために、サブリプレゼンテーション内に、新たな「`req u i r e d` (必須)」属性が加えられる。複合トラック、またはタイルトラックの1つ以上に対する要求はすべて、セグメントリスト 610 内に提供される U R L から計算される (1 時間間隔当たり 1 つ)。例では、M P D の先頭において「Base U R L」と組み合わされた「U R L_X」は、H T T P G E T 要求を実行するためにクライアントが用いることができる完全な U R L を提供する。この要求により、クライアントは、複合トラックのためのデータと、すべてのタイルトラックのためのすべてのデータとを得るだろう。要求の代わりに、伝送を最適化するために、クライアントは、`i n d e x _ r a n g e` 属性 620 からの利用可能なデータを用いて、セグメントインデックス情報 (通常は、当業者により周知の I S O B M F F 内の「`s i d x`」情報および / または「`s s i x`」) を第1に要求することができる。このインデックス情報は、コンポーネントの各々のためのバイト単位の範囲を決定することを可能にする。そして、D A S H クライアントは、適切なバイト単位の範囲で、(要求された複合トラックを含む) 選択されたトラックと同数の H T T P G E T 要求を送信することができる。

10

20

【 0 0 2 9 】

ストリーミングセッションを開始するとき、D A S H クライアントは、マニフェストファイルを要求する。一旦受信されれば、クライアントは、マニフェストファイルを解析し、その環境に適した1セットの A d a p t a t i o n S e t を選択する。次に、クライアントは、各 A d a p t a t i o n S e t 内部の M P D において、その帯域幅と互換性をもつ R e p r e s e n t a t i o n 、復号化およびレンダリング機能を選択する。次に、それは、メディア復号器のためのイニシャライゼーション情報を始めとして、要求されるセグメントのリストを予め構築する。イニシャライゼーション情報が復号器によって受信されると、それらはイニシャライズされ、クライアントは、第1のメディアデータを要求し、ディスプレイを実際に開始する前に最小データ量をバッファする。

30

【 0 0 3 0 】

これらの複数の要求 / 応答は、ストリーミングセッションの起動における遅延を招くかもしれない。サービスプロバイダにとってのリスクは、ビデオを見始めることなくサービスを離れるクライアント見ることである。クライアントによって実行される第1のメディアデータチャンクのための初期の H T T P 要求とメディアデータチャンクが実際に再生し始める時間との間の時間を、スタートアップ遅れとよぶことが一般的である。それは、ネットワークラウンドトリップ時間だけでなく、メディアセグメントのサイズにも依存する。

【 0 0 3 1 】

サーバプッシュは、ウェブリソースのローディング時間を減少させるために有用な機能である。このようなサーバは、図 1 a ~ 図 1 e を参照しながら論じる。

40

【 0 0 3 2 】

図 1 b では、H T T P / 2 交換 (H T T P / 2 e x c h a n g e s) において、要求は、必要とされるリソース : (図 1 a に示されるように) リソース R 1 ~ R 4 およびサブリソース A ~ I ごとに送信されなければならない、ということが図示される。しかしながら、サーバによるプッシュ機能を用いるとき、図 1 c に図示されるように、要求の数は、要素 R 1 ~ R 4 に制限される。要素 A ~ I は、図 1 a に示される依存性に基づいてサーバによってクライアントに対して「プッシュされ」、それによって、不必要的関連する要求が生じる。

【 0 0 3 3 】

50

このように、サーバがプッシュ機能を用いる場合、図1bおよび図1cに図示されるように、そのサブリソースと共にリソースをロードするために必要なHTTPラウンドトリップ（要求+応答）の数が減少される。これは、モバイルネットワークなどの高待ち時間のネットワークにとって、とりわけ興味深い。

【0034】

HTTPは、ウェブリソース、通常はウェブページ、を送信するために用いられるプロトコルである。HTTPは、クライアントとサーバとを暗に含む。

【0035】

- ・クライアントは、サーバに対して要求を送信する。

【0036】

・サーバは、クライアントの要求に対して、ウェブリソースのリプレゼンテーションを含む応答により返答する。

【0037】

要求および応答は、様々な部分、特にHTTPヘッダ、を含むメッセージである。HTTPヘッダは、値と共に名称を備える。例えば「Host : en.wikipedia.org」は「Host（ホスト）」ヘッダであり、その値は「en.wikipedia.org」である。それは、問い合わせられたリソースのホストを示すために用いられる（例えばHTTPを記述するウィキペディアページは、http://en.wikipedia.org/wiki/HTTPにおいて利用可能である）。HTTPヘッダは、クライアント要求およびサーバ応答上に出現する。

【0038】

HTTP/2は、ストリームを通じて要求／応答を交換することを可能にする。ストリームは、HTTP要求および応答ごとにHTTP/2接続の内部で生成される。フレームは、要求および応答のコンテンツおよびヘッダを伝達するために、ストリーム内で交換される。

【0039】

HTTP/2は、以下のような、個別の意味をもつフレームの制限されたセットを定義する。

【0040】

- ・HEADERS : HTTPヘッダの伝送のために提供される
- ・DATA : HTTPメッセージコンテンツの伝送のために提供される
- ・PUSH_PROMISE : プッシュされたコンテンツのアナウンスのために提供される
- ・PRIORITY : ストリームのプライオリティを設定するために提供される
- ・WINDOW_UPDATE : 制御フローウィンドウの値を更新するために提供される
- ・SETTINGS : 設定パラメータを伝達するために提供される
- ・CONTINUATION : ヘッダブロックフラグメントのシーケンスを継続するために提供される
- ・RST_STREAM : ストリームの終了またはキャンセルのために提供される

サーバによるプッシュは、サーバがクライアントに対して自発的なウェブリソースリプレゼンテーションを送信することを可能にするために、HTTP/2に導入された。ウェブページなどのウェブリソースは、一般的には、それ自体が他のリソースへのリンクを含んでいる、他のリソースに対するリンクを含む。完全にウェブページを表示するために、リンクされ、サブリンクされたすべてリソースは、一般的には、クライアントによって検索される必要がある。この増分ディスクバリは特に高待ち時間ネットワーク（モバイルネットワーク）上の、ウェブページの低速表示をもたらすであろう。

【0041】

所定のウェブページに対する要求を受信するとき、サーバは、他のどのリソースが要求されたリソースの完全な処理のために必要とされるのかを知っているであろう。要求されたリソースおよびリンクされたリソースを同時に送信することによって、サーバは、ウェ

10

20

30

40

50

ページのロード時間を減少させることを可能にする。このように、プッシュ機能を用いて、サーバは、所定のリソースが要求されたときに、追加リソースリプレゼンテーションを送信してもよい。

【0042】

図1eのフローチャートを参照して、プッシュ機能を実行するサーバの例示的なオペレーションモードを記述する。

【0043】

ステップ100の間、サーバは、初期要求を受信する。次に、サーバは、ステップ101の間に応答の一部としてプッシュするリソースを識別し、ステップ102の間にコンテンツ応答を送信し始める。並行して、サーバは、ステップ103の間にクライアントに対してプッシュプロミスマッセージを送信する。これらのメッセージは、例えば図1aに示される依存性に基づいて、サーバがプッシュすることを意図している他のリソースを識別する。これらのメッセージは、どのプッシュされたリソースが送信されることになるのかをクライアントに予め知らせるために送信される。特に、これは、同時にプッシュされた、またはプッシュされようとしている、リソースに対して、クライアントが要求を送信する、というリスクを低減する。このリスクをさらに低減するために、サーバは、プッシュプロミス内に記述されるリソースを参照する応答のあらゆる部分を送信する前に、プッシュプロミスマッセージを送信するべきである。これは、また、クライアントがそれらのリソースを望まないならば、クライアントがプロミスされたリソース(`promised resources`)のプッシュのキャンセルを要求することを可能にする。次に、サーバは、ステップ104の間に、応答およびすべてのプロミスされたリソースを送信する。プロセスは、ステップ105の間に終了する。

【0044】

図1dのフローチャートは、クライアント側の処理を示す。

【0045】

クライアントがサーバから取り出すリソースを識別すると、対応するデータが既にそのキャッシュメモリ内にあるか否かを、ステップ106の間にまずチェックする。リソースが既にキャッシュメモリ内にある場合(はい(`Yes`))、ステップ107の間に、そこから取り出される。キャッシュされたデータは、先の要求から取りだされたデータ、またはサーバによって予めプッシュされたデータのいずれかであろう。それがキャッシュメモリ内になかった場合(いいえ(`No`))、クライアントは、ステップ108の間に、要求を送信し、サーバの応答を待つ。サーバからのフレームを受信すると、クライアントは、フレームがプッシュプロミスに対応するか否かを、ステップ109の間にチェックする。データフレームがプッシュプロミスに対応するならば(はい(`Yes`))、ステップ110の間、クライアントは、プッシュプロミスを処理する。クライアントは、プッシュされるリソースを識別する。クライアントがリソースを受信しなければ、クライアントは、エラーメッセージをサーバに対して送信してもよく、そうすれば、サーバは、そのリソースをプッシュしない。そうでなければ、クライアントは、対応するプッシュコンテンツを受信するまでプッシュプロミスを保存する。サーバがそれをプッシュしている間に、クライアントがプロミスされたリソースを要求しないように、プッシュプロミスが用いられる。データフレームがプッシュプロミスに対応しない場合(いいえ(`No`))、ステップ111の間に、フレームがプッシュデータに関連するデータフレームであるか否かがチェックされる。それがプッシュデータに関する場合(はい(`Yes`))、クライアントは、ステップ112の間に、プッシュされたデータを処理する。プッシュされたデータは、クライアントのキャッシュ内に保存される。フレームがプッシュデータに関するデータフレームではない場合(いいえ(`No`))、ステップ113の間に、サーバから受信された応答にそれが対応するか否かがチェックされる。フレームがサーバからの応答に対応する場合(はい(`Yes`))、応答は、ステップ114の間に処理される(例えば、アプリケーションに対して送信される)。そうでなければ(いいえ(`No`))、フレームが応答の終了を識別するか否か(はい(`Yes`))が、ステップ115の間にチェックされる。この

10

20

30

40

50

場合、処理は、ステップ116の間に終了する。そうでなければ、処理は、ステップ109に戻る。

【0046】

このように、クライアントが応答およびプロミスされたリソースを受信する、ということが思われる。そのため、取りだされたウェブページを表示するブラウザなどのアプリケーションによって応答が用いられる間に、プロミスされたリソースは、一般的には、クライアントのキャッシュ内に保存される。クライアントアプリケーションがプッシュされたリソースの1つを要求するとき、リソースは、いかなるネットワーク遅延も引き起こさずに、クライアントのキャッシュから直ちに取りだされる。

【0047】

キャッシュ内のプッシュされたリソースの保存は、キャッシュ制御指示を用いて制御される。キャッシュ制御指示は、また、応答の制御のために用いられる。これらの指示は、特に、プロキシに対して適用可能であり、プッシュされた若しくはプッシュされない任意のリソースは、プロキシ若しくはクライアントのみによって保存されてもよい。

【0048】

図1aは、それらの関連性とともにサーバによって所有された1セットのリソースのグラフである。リソースのセットは、以下のように関連し合う。すなわち、R₁、R₂、R₃、およびR₄は、クライアントによって適切に処理されるために一緒にダウンロードされる必要のあるリソースである。さらに、サブリソースA～Hが定義される。これらのサブリソースは、1つ、2つ、または3つのリソースに関連する。例えば、Aは、R₁に対してリンクされ、Cは、R₁、R₂、およびR₄に対してリンクされる。

【0049】

既に上述した図1bは、サーバPUSH機能を用いないHTTP交換を示しており、クライアントは、R₁を要求し、次にR₂、A、B、C、およびDを発見し、それらを要求する。それらを受信した後、クライアントは、R₃、R₄、F、およびGを要求する。最終的に、クライアントは、HおよびIのサブリソースを要求する。これは、リソースの全体のセットを検索するために4つのラウンドトリップを必要とする。

【0050】

既に上述した図1cは、サーバによって直接的に接続されるサブリソースをプッシュする機能を用いたHTTP交換を示す。R₁を要求した後、サーバは、R₁を送信し、A、B、C、およびDをプッシュする。クライアントは、R₂を識別し、それを要求する。サーバは、R₂を送信し、FおよびGをプッシュする。最終的に、クライアントは、R₃、R₄を識別し、これらのリソースを要求する。サーバは、R₃、R₄を送信し、HおよびIをプッシュする。これは、リソースの全体のセットを取りだすために3つのラウンドトリップを必要とする。

【0051】

1セットのリソース、通常はウェブページおよびそのサブリソース、のローディング時間を減少させるために、HTTP/2は、複数の要求および応答のプライオリティを並列に交換することを可能にする。図2に図示されるように、ウェブページは、Java（登録商標）Script、画像などのような、いくつかのリソースのダウンロードを要求してもよい。初期のHTTP交換200の間に、クライアントは、HTMLファイルを検索する。このHTMLファイルは、2つのJava（登録商標）Scriptファイル（JS1、JS2）、2つの画像（IMG1、IMG2）、1つのCSSファイル、および1つのHTMLファイルに対するリンクを含む。交換201の間に、クライアントは、各ファイルに対する要求を送信する。図2の交換201において与えられる順序は、ウェブページの順序に基づき、クライアントは、リンクが見つかるとすぐに、要求を送信する。そして、サーバは、JS1、CSS、IMG1、HTML、IMG2、およびJS2に対する要求を受信し、その順序に従ってこれらの要求を処理する。そして、クライアントは、その順序で、これらのリソースを取りだす。

【0052】

10

20

30

40

50

H T T P プライオリティは、どの要求がより重要であり且つ他の要求より早く処理されるべきであるかを、クライアントに明示することを可能にする。特定のプライオリティの使用を、交換 202 に示す。J a v a (登録商標) S c r i p t ファイルは、最も高いプライオリティを割り当てられる。C S S およびH T M L ファイルは、中プライオリティを割り当てられ、画像は、低プライオリティを割り当てられる。このアプローチにより、ブロッキングファイル、または他のファイルより早く他のリソースに対する参照を含むことができるファイルを受信することができる。これを受け、交換 202 に記述されるように、サーバは、まずJ a v a (登録商標) S c r i p t ファイル、その後C S S およびH T M L ファイル、最後には画像を、送信することを試みることが期待される。サーバは、クライアントプライオリティに従うようには義務付けられていない。

10

【0053】

プライオリティに加えて、H T T P / 2 は、同時に交換されるデータ量を制御することができる、ということを提供する。クライアントおよびサーバは、接続ベース当たりおよびストリームベース当たりのバッファリングができるデータ量を指定することができる。これは、T C P 輻輳制御と同様であり、利用可能なバッファサイズを指定するウインドウサイズは、所定の値にイニシャライズされ、エミッタがデータを送信するたびに、ウインドウサイズは減少し、エミッタは、ウインドウサイズが 0 より下にならないようデーティアの送信を停止しなければならない。受信器は、データを受信し、データがバッファから受信され取り出されたということを通知するメッセージを送信し、メッセージは、バッファから取り出されたデータ量を含んでおり、そして、ウインドウサイズが所定の値から増加され、エミッタは、データを送信することを再開することができる。

20

【0054】

上記を考慮すると、クライアントは実行しているアプリケーションの目的のためにコンテンツのベストのリプレゼンテーションを一般的には選択することができるので、D A S H は、クライアントがストリーミングをリードするという前提に基づくように思われる。例えば、クライアントは、そのフォームファクタおよびスクリーン解像度に基づいて高解像度または小解像度のコンテンツを要求するべきかどうかを認識していくてもよい。

【発明の概要】

【発明が解決しようとする課題】

【0055】

30

サーバベースのストリーミングは、通常は、R T P を用いて実行される。D A S H とは対照的に、R T P は、H T T P を用いず、ウェブインフラストラクチャ、特にプロキシ、キャッシュから直接的に恩恵を受けることができない。ウェブソケットベースのメディアストリーミングは、同じ短所を持つ。H T T P / 1 . 1 では、サーバがクライアント要求に対して一般的には答えることしかできないので、サーバベースのストリーミングは、容易に実行することができない。H T T P / 2 では、特にプッシュ機能の導入では、D A S H ベースのサーバは、ストリーミングを通すことができる。このように、サーバは、ユーザエクスペリエンスを最適化するために、ストリーミングしているコンテンツの特性のサーバの知識を用いることができる。例えば、サーバは、広告は付加的な限定的な量の帯域幅を利用するのでアドバーティスメントをH D としてプッシュすることを除けばフィルムをS D として（制限された帯域幅のために）プッシュするかもしれない。別の例は、低解像度ビデオでの高速起動を実行し始め、且つ一旦帯域幅が十分に推定されれば可能な限りベストのリプレゼンテーションに切り替える、サーバのケースである。

40

【0056】

サーバがストリーミングを通すことを可能にするために、1つのアプローチは、サーバに好ましくはデータ（特にD A S H データ）をプッシュさせることである。そして、クライアントは、ビデオを表示するのに利用することができるいかなるデータも用いる。サーバは、通常は、一度にいくつかのセグメントのプッシュをアナウンスする。そして、サーバは、並行してまたは連続的にセグメントを送信する。

【0057】

50

発生する問題は、クライアントとサーバはプロミスされたデータが望まれる時間に送られ、受信されるかどうか、わからないかもしれないことである：クライアントは、ビデオセグメントが、いつ、どの順序で送信されることになるのかをわからないかもしれない。

【0058】

また、サーバによってプッシュまたはアンウンスされた、プロミスされたデータは、クライアントのニーズにミスマッチするかもしれません、それにより、サーバエンドにおいて特にリソースを無駄にすることになる。

【0059】

このように、特に、DASHベースの通信との関連においてデータストリーミングを強化する必要性がある。10

【課題を解決するための手段】

【0060】

上記の課題を解決するために、本発明のサーバは、例えば、以下の構成を有する。すなわち、メディアセグメントをクライアントへ送信するサーバであって、前記クライアントから第1メディアセグメントのリクエストを受信する受信手段と、前記受信手段によるリクエストの受信に応じて、前記クライアントに対して前記第1メディアセグメントを送信する送信手段と、前記第1メディアセグメントのリクエストを送信した前記クライアントに対してプッシュすべき第2メディアセグメントを、前記クライアントとの間で共有されたプッシュストラテジーを用いることによって特定する特定手段と、前記特定手段により特定された前記第2メディアセグメントを前記クライアントに対してプッシュするプッシュ手段とを有する。20

【0061】

本発明の他の機能および効果は、図1～図6に加えて、添付された以下の図面を参照しながら、非限定的な好ましい実施形態の以下の記述から明らかになる。

【図面の簡単な説明】

【0062】

【図1a】サーバによってプッシュされる要素の依存性を図示する。

【図1b】プッシュ機能HTTP/2交換の例を図示する。

【図1c】プッシュ機能を用いたHTTP交換の例を図示する。30

【図1d】クライアント側の処理のフローチャートを図示する。

【図1e】サーバ側の処理のフローチャートを図示する。

【図2】サーバ・クライアント間のHTTP/2交換の例を図示する。

【図3】HTTPメディアストリーミングの概略的原理を図示する。

【図4a】メディアプレゼンテーションおよびマニフェストファイルの生成例を図示する。40

【図4b】マニフェストファイル例を図示する。

【図5】MPDファイル例を図示する。

【図5a】DASHクライアントの標準的挙動例を図示する。

【図5b】マニフェストファイルのツリー表現例を図示する。

【図6】DASHマニフェストファイル例を図示する

【図7a】実施形態にかかる、順序付けされたメディアセグメントを図示する。

【図7b】実施形態にかかる、順序付けされたメディアセグメントを図示する。

【図8】実施形態にかかる、サーバによって実行される例示的なステップのフローチャートである。

【図9】実施形態にかかる、クライアントによって実行される例示的なステップのフローチャートである。

【図10】実施形態にかかる、プロキシによって実行される例示的なステップのフローチャートである。

【図11】実施形態にかかる帯域幅測定を図示する。50

【図12】実施形態にかかるビデオ再生のイニシャライゼーションを図示する。

【図13】実施形態にかかる装置の概略図である。

【図14a】クライアント側における本発明の一般的なステップを、フローチャートを用いて図示する。

【図14b】サーバ側における本発明の一般的なステップを、フローチャートを用いて図示する。

【図15a】明示的アプローチに基づいて、クライアント側において共有されるプッシュポリシーを決定するステップを、フローチャートを用いて図示する。

【図15b】明示的アプローチが用いられるときに、サーバ側においてプッシュポリシーを決定するステップを、フローチャートを用いて図示する。 10

【図16】サーバによって適用されるプッシュポリシーを指定するためにPush Policyノードが用いられるMPDドキュメントを示す。

【図17】共有されたプッシュポリシー「Push Policy」にしたがってプッシュされる準備ができている、いくつかのセグメントを識別してマークするステップを、フローチャート用いて図示する。

【図18a】HTTP「プッシュポリシー」ヘッダ内で送信されるプッシュポリシーによるサーバとクライアントとの間の通信の例を図示する。

【図18b】プッシュポリシーを変更するクライアントの要求と同じ例を図示する。

【図19】メディアプレゼンテーション記述(MPD)ファイルの例を図示する。

【図20】アナウンスマッセージをマージする、実施形態によるサーバ側における処理のステップを、フローチャートを用いて図示する。 20

【図21】プッシュポリシーを宣言するためにHTTPヘッダを用いる場合のサーバ側における処理のステップを、フローチャートを用いて図示する。

【図22】プッシュポリシーを宣言して共有するためにHTTP要求を用いる場合のクライアント側における処理のステップを、フローチャートを用いて図示する。

【図23】ドキュメントの階層レベルにおいてサーバによって適用されるプッシュポリシーを指定するためにSupplemental Property要素が用いられるMPDドキュメントを示す。

【図24】XPathベースのプッシュポリシーのために例として用いられるMPDドキュメントを示す。 30

【図25】プッシュポリシーを適用する前の、例えばウェブページ内のプライオリティリー内の要素の順序付けを図示する。

【図26】本発明の実施形態による、DASH高速起動を取得するために、サーバによって、およびクライアント装置によって、それぞれ実施された例示的な方法を示す。

【図27】DASH高速起動のためのサーバによって実施される例示的な方法を記述する。

【図28】DASH高速起動のためのクライアント装置によって実施された可能な方法を記述する。

【発明を実施するための形態】

【0063】

以下において、本発明の実施形態は、HTTP 2.0 プロトコルを実施するDASHに基づくネットワークとの関連で記述される。ストリーミングされるデータは、例えばビデオデータである。本発明の実施形態は、DASHネットワークに限定されない。

【0064】

クライアント装置に対してデータをストリーミングする通信ネットワークのサーバ装置は、送信されるデータ要素に対するクライアントからの明示的リクエストを伴わずに、クライアントに対してデータ要素を送信することによる、プッシュ機能を実施する。

【0065】

サーバおよびクライアントは、プッシュプロミスを決定し、且つ対応するデータを実際 50

に送信するように、サーバを駆動するプッシュポリシーを共有してもよい。この共有により、クライアントは、そのようなプッシュをキャンセルするために、いくつかの無用データのプッシュを予期することができる。PUSH_PROMISEフレームは、送信される前にキャンセルされてもよいので、これは、ネットワーク使用と同様に、サーバの処理も減少させることになる。

【0066】

具体的な実施形態において、サーバは、明確に要求されていないデータ要素の送信をアナウンスする、そのプッシュプロミスにおいて、データ要素を送信するよう意図するサーバの順序に関する順序付け情報を示すことができる。データ要素の順序は、プライオリティ値、例えばHTTP/2によるプライオリティ値、を用いて定義されてもよい。

10

【0067】

プッシュプロミスを受信すると、クライアント装置は、サーバによって意図された送信の順序を予め決定することができ、それによって、クライアントは、自身の所望の順序と一致しなかった場合に、提案された順序付けに対応できる。例えば、クライアント装置は、プライオリティ値を更新し、更新されたプライオリティ値をサーバに対して送信することができる。サーバは、したがって、クライアントのニーズと適切に一致させるために、新たなプライオリティ値に基づいて送信順序付けを変更することができる。サーバは、将来のデータ送信を考慮して、更新されたプライオリティを用いることができる。

【0068】

20

実施形態によれば、クライアントは、サーバに対するデータ要素の送信の全面的な順序付けまたは部分的な順序付けを要求しても良い。

【0069】

図7aを参照して、全面的な順序付けを説明する。クライアントは、ステップ700の間に、メディアプレゼンテーション記述（以降、MPD）をサーバに対して要求する。サーバは、ステップ701の間に、クライアントに対して返送するMPDを取得し、プッシュするべき対応するデータ要素を識別する。図7aの例において、サーバは、「データ1.1」、「データ1.2」および「データ1.3」をプッシュするデータ要素として識別する。これらの要素は、例えばデータセグメントである。要素「データX.1」は、データXのためのベース層を表現し、要素「データX.2」は、データXのためのエンハンスマント層を表現し、「データX.3」は、データXのための付加的なエンハンスマント層を表現する。サーバは、データ要素のための具体的な送信の順序を定義する。サーバは、来たるプッシュデータ要素のアナウンスのためにクライアントに対して送信される、PUSH_PROMISEフレームに、それぞれのプライオリティ値を関連づける。そして、サーバは、ステップ702の間に、関連するプライオリティおよびMPDにより、PUSH_PROMISEフレーム「P1.1」、「P1.2」および「P1.3」を送信する。次に、MPDおよびプッシュプロミスを送信した直後、ステップ703の間に、サーバは、クライアントに対して、定義された送信順序における「データ1.1」、「データ1.2」および「データ1.3」に続くセグメントである、「データ1.1」要素に対応するデータフレームと要素「データ2.1」、「データ2.2」および「データ2.3」にそれぞれ対応するPUSH_PROMISEメッセージ「P2.1」、「P2.2」および「P2.3」とを送信する。データフレームの受信およびステップ703のプッシュプロミスと並行して、クライアントは、MPDおよび「P1.1」、「P1.2」および「P1.3」PUSH_PROMISEフレームの受信後に、エンハンスマント層「データ1.2」は付加的なエンハンスマント層「データ1.3」に比較して低いプライオリティである、と判断する。したがって、クライアントは、ステップ704の間に、低い「データ1.2」プライオリティに対してプライオリティ更新フレームを送信する。プライオリティ更新要求を受信すると、サーバは、ステップ705の間に、送信のスケジュールを変更する。その結果、「データ1.3」が送信された後、「データ1.2」の送信は先延ばしされる。さらに、サーバは、「データ1.2」に関連付けられたセグメントをリンクす

30

40

50

るためにMPDを用いる。サーバは「データ2.2」を識別し、そのプライオリティを減じる。

【0070】

図7bを参照して、部分的な順序付けを説明する。図7bのステップ710～714は、図7aのステップ700～704と実質的に同じである。プライオリティ更新フレームの受信の後、サーバの挙動は、先に記載されたステップ705と比べて異なる。ステップ715の間、サーバは、既に「データ1.2」の送信を開始しており、送信をさらに進める。そのセグメントに関して、プライオリティに変更はない。サーバは、それにもかかわらず、連結されたセグメント、すなわち本例における「データ2.2」)のプライオリティを更新する。プライオリティ変更が考慮された、という事実をアナウンスするために、サーバは「データ2.2」のためのプライオリティ更新マッサージ(priority update message)を送信してもよい。クライアントは、したがって、変更を通知されることができる。

【0071】

本発明の実施形態は、ビデオのすべての部分を高品質に再生することができるよう、サーバが予め十分に良い高品質のビデオ部分をプッシュすることができる使用事例において実施されてもよい。例えば、ビデオは、低品質として再生される部分1、高品質として再生される部分2、および低品質として再生される部分3に分割されることができる。クライアントとサーバとの間の帯域幅は、高品質ではなく低品質のリアルタイムストリーミングを可能にする。その場合、サーバは、部分2を強化し、部分1をインターリープしてもよい。一旦部分1が再生されれば、強化された部分2は、また、利用可能であり、サーバは、同じ部分2の強化と共同して高品質として再生される部分2のベース層を送信する。このように、サーバは、すべての部分2が高品質として再生されることを確実にする。部分3は、その後に送信される。ユーザエクスペリエンスを妨害する品質のちらつき(quality flickering)を軽減することができ、品質の切り換え(quality switching)のみが、限られた数のモーメントにおいて行われる。サーバは、ビデオコンテンツを認識して以後、異なる品質レベルにいつ切り替えるべきであるかを認識するために最善位置にある。

【0072】

図8は、実施形態にかかる、プッシュベースのDASHメディアストリーミングを実施するサーバによって実行されたステップのフローチャートである。ステップ800～812は、一般的な原理を記述する。ステップ820～827は、より具体的には、クライアントからのプライオリティのフィードバックの管理に対処する。

【0073】

ステップ800の間、サーバは、クライアントから要求Rを受信する。この要求は、通常はMPDファイルを参照することによって特定のメディアを識別する。次に、サーバは、ステップ801～810を含む反復処理を実行する。その処理は、定義された順序にしたがってデータを送信することを有する。送信の順序は、クライアントのフィードバックにしたがって更新される。一旦データが送信されれば、それらはクライアントによって受信され再生される。次に、サーバは、送信すべき新たなデータを識別し、処理は、そのように継続し続ける。

【0074】

第1の反復は、送信されるデータが識別される間のステップ801で始まる。反復処理の第1の実行の場合、クライアントがビデオ再生を可能な限り迅速に開始することを可能にするために、高速起動アプローチが用いられてもよい。さらに、サーバは、また、チャプターへのメディアの細区分を識別しても良い。クライアントが一般的にはチャプターを用いてナビゲートする、ということをサーバが認識する場合、サーバは、メディアの先頭に対して対応するセグメントだけでなくメディア内の第1のチャプターの開始に対応するセグメントも実際に選択してもよい。反復の第1の実行の後、サーバは、また、接続がメディアの高品質のリプレゼンテーションの送信をサポートすることができる、ということ

10

20

30

40

50

を検出しても良い。このように、サーバは、解像度または品質の切り換えをいつ行うべきであるのかを識別しても良い。

【0075】

一旦サーバがプッシュするセグメントのリストを識別すれば、サーバは、これらのセグメントのための送信順序を定義する。送信順序は、1ステップ802の間に各々のプッシュされたセグメントに対する初期のプライオリティ値を算出するために用いられる。順序付けは、いくつかのパラメータに基づいてもよい。

【0076】

第1のパラメータは、異なるセグメント間の関連性であってもよく、例えば、いくつかのセグメントは、他のセグメントを正確に復号するために利用可能でなければならない。利用可能でなければならないセグメントは、したがって、前記他のセグメントよりも高いプライオリティを割り当てられる。10

【0077】

第2のパラメータは、過去統計値から採集することができる、ビデオセグメントの人気度であってもよい。例として、YouTube（登録商標）のURLにより、ビデオ内の特定の期間をアドレス指定しても良い。これらのURLSに関連するリンク上をクリックすると、特定の時刻にビデオ再生をスタートするために必要とされるビデオのみが取りだされる。さらに、ビデオがチャプター化されている場合、各々のチャプターの先頭は、一般的には多くの場合、チャプター開始の間のセグメントよりもユーザから取りだされる。チャプターの先頭のSegmentsは、したがって、中間のチャプターセグメントインよりも高いプライオリティを割り当てられる。20

【0078】

第3のパラメータは、タイムラインであってもよく、再生されているビデオセグメントにより近いビデオセグメントのプライオリティは、後で再生されることになるビデオセグメントのプライオリティより高い。

【0079】

第4のパラメータは、セグメントを実際に送信するために費やされる推定時間であってもよい。ビデオセグメントが大きい場合、送信するのに長い時間がかかり、そのため、できるだけ早く、すなわち高プライオリティで、送信を開始するべきである。

【0080】

2つのセグメントが同一のプライオリティを有する場合、対応するデータフレームは、送信の間にインターリープすることができる。関心領域がメディアコンテンツ内で識別された場合、帯域幅が高品質のリプレゼンテーションには十分に大きくはないが、低品質のリプレゼンテーションには十分に大きければ、サーバは、関心領域のためにのみエンハンスメント層を選択してもよい。30

【0081】

一旦プライオリティが算出されれば、サーバは、ステップ803の間に、プライオリティ値を含むPUSH_PROMISEフレームを送信する。すべてのセグメントの識別は、PUSH_PROMISEフレームの送信を開始するためには必要でない。MPDがプッシュされるセグメントのために送信される場合（ステップ804）、MPDが送信される（ステップ805）。セグメント送信は、ステップ806の間に、並列的に開始する。40

【0082】

一旦PUSH_PROMISEフレームがクライアントによって受信されれば、サーバは、プライオリティ更新変更を受信し、その後、それに応じて、その送信スケジュールを変更してもよい（ステップ807～808およびステップ820～828）。セグメントを送信する一方で、サーバは、プライオリティ変更メッセージの受信を待つ。プライオリティ変更メッセージが受信された場合（ステップ807）、サーバは、それに応じてセグメントを再順序付けて、セグメント送信を継続する（ステップ808）。一旦すべてのセグメントが送信されると（ステップ809～1）、サーバは、メディアの端部までメディアをストリーミングし続けるために反復処理を再開する。メディアの端部に到達すると（50

ステップ 809-2)、サーバは、別のメディアを自動的にストリーミングし始めるべきか否かをチェックする(ステップ 810)。別のメディアがストリーミングされるべきである場合(はい(Y e s))、サーバは、ストリーミングする新たなメディアを識別し(ステップ 811)、ステップ 801 から処理を再開する。新たなデータがストリーミングされるべきでない場合、処理は停止される(ステップ 812)。

【0083】

クライアントからのプライオリティのフィードバックの、すなわちステップ 808 の管理は、ステップ 820 の間に、プライオリティ更新変更メッセージの受信とともに始まる。クライアントがセグメントプッシュをキャンセルした場合、以下のステップが実行されてもよく、このケースは、そのセグメントに対して最も低いプライオリティを割り当てる 10 ことと実際には同等であると、理解してもよい。

【0084】

プライオリティ更新変更メッセージを受信すると、サーバは、ステップ 821 の間に、関連セグメントを識別する。そして、サーバは、セグメント送信の再順序付けを進める(ステップ 822 および 823)。セグメントが既に送信されていれば、処理は終了する。セグメントが送信されているならば、サーバの実装に応じて、送信を変更することを拒絶してもよいし(例えば複雑すぎるので)、または、送信される残りのデータを実際にスケジュール変更してもよい。

【0085】

データのスケジュール変更は、以下のように実行されてもよい。サーバは、プッシュするビデオセグメント(および/またはプッシュされるビデオセグメント)のリストを保存する。このリストは、サーバによって設定されたプライオリティにしたがって順序付けられる。そして、サーバは、セグメントに新たなプライオリティ値を設定する。そして、リストが再順序付けられ、対応するビデオセグメントの送信は、それに応じて、早くなったり遅くなったりする。 20

【0086】

一旦ビデオセグメントが再順序付けられれば、サーバは、他の関連するビデオセグメントに対して、このプライオリティ変更を適用することを実際に決定してもよい。クライアントがエンハンスマント層の一部であるビデオセグメントのプライオリティを引き上げれば、サーバは、このエンハンスマント層のすべてのセグメントのプライオリティを引き上げてもよい。逆に、クライアントがベースビデオセグメント層のプライオリティを引き下げる場合、このセグメントに一時的に関連するすべてのセグメントのプライオリティが引き下げられてもよい。この処理は、ステップ 824 ~ 827 に記述される。MPD およびスケジュール変更されたビデオセグメントに基づいて、サーバは、関連セグメントのリストを識別する(ステップ 824)。関連性は、時間、空間、品質ベースなどであってもよい。MPD は、潜在的な関連性をより適切に示すために強化されてもよい。特に、(1つ以上のビデオセグメントを再生するのに必要である)イニシャライゼーションセグメントのプライオリティが引き下げられるかまたは引き上げられると、すべての関連セグメントがスケジュール変更されてもよい。これは、ベース層セグメントおよび強化セグメントの同様のケースになり得る。各々の識別された関連セグメントに対して、サーバは、関連セグメントの送信が変更されるべきであるか否かテストする(ステップ 825)。変更されるべきである場合、サーバは、各セグメントに対して新たなプライオリティ値を算出し(ステップ 826)、それに応じて、セグメント送信をスケジュール変更する(ステップ 827)。新たなプライオリティ値は、ステップ 820 の間に受信された新たなプライオリティ値とステップ 821 の間に識別されたセグメントの初期のプライオリティ値との間の差分を古い値に対して加えることによって算出されてもよい。各関連セグメントがテストされると、処理は停止する(ステップ 828)。 30

【0087】

サーバは、また、WINDOW SIZE フレームなどの制御フローメッセージを受信してもよい。これらのメッセージによって、サーバは、クライアントが現在何を再生して 50

いるのかを、識別することが出来るであろう。いくつかの追加的バッファスペースがクライアントの端上で利用可能な場合、いくつかのデータ、通常は最も古いデータ、がバッファから取り出された、ということが推測され得る。サーバが送信されたデータの履歴を保持すれば、サーバは、どのデータが取り出されたのかを識別することができる。このように、もしサーバが、クライアントのキャッシュの順序付けを認識しているならば、サーバは、クライアントが現在どのビデオセグメントを再生しているかを知ることができる。この順序付けは、タイムラインにしたがってキャッシュされたデータを、順序付けることを可能にするMPDに基づいてもよい。そして、サーバは、例えばクライアントの時間スキップを検出することができる。サーバは、クライアントがビデオチャプターをスキップし続けることができるよう、予め次のチャプターの開始を迅速に送信することによって対応してもよい。

10

【0088】

プライオリティをもつPUSH_PROMISEフレームの送信が様々な方法で行われてもよい、ということを留意するべきである。PUSH_PROMISEフレームは、クライアントによって起動される開かれたストリーム(opened stream)に関連しなければならない。実施形態によれば、ステップ800の間にクライアントによって行われた初期のストリームは、常に開かれたままであってもよい。他の実施形態によれば、PUSH_PROMISEフレームは、サーバによって開かれたストリーム内で送信される。この場合、それが親クライアントに起動されたストリームによって送信されるよう、クライアントは、PUSH_PROMISEフレームを考慮する。このように、個々のPUSH_PROMISEフレームに対応する仮想要求の正しいヘッダを算出することができる。

20

【0089】

他の実施形態によれば、プライオリティメッセージは、PUSH_PROMISEと共同で送信される。第1の可能性は、PUSH_PROMISEフレーム内部のヘッダとしてそれを送信することである。もう1つの可能性は、対応するPUSH_PROMISEフレームによって保存されたストリームIDを持つPRIORITYフレームを送信することである。第3の可能性は、このPUSH_PROMISEフレーム、その後、(ストリームを開くために)対応するHEADERSフレーム、および新しく開かれたストリーム上のPRIORITYフレームを送信することである。

30

【0090】

さらにクライアントのバッファを制御するために、サーバは、クライアントによってキャッシュされたセグメントの新たなリプレゼンテーションを送信してもよい。この新たなリプレゼンテーションの一部として送信されたヘッダ内で、HTTPキャッシュ指示は、例えばキャッシュすることができるものとしてそれをマークすることによって実際にセグメントを取り出すことをクライアントに要求するために用いられてもよい。これにより、クライアントの端上のバッファスペースを回復することが可能となる。HTTP/2制御フローが用いられてもよい。そして、サーバは、付加データをプッシュすることができる。

【0091】

40

サーバは、各ビデオセグメントのプライオリティ値を送信してもよい。サーバは、また、特定セグメントのためのプライオリティ値を送信してもよい。サーバが、現在のPUSH_PROMISEフレームのプライオリティ値を送信しなかった場合、クライアントは、サーバから送信された最後のプライオリティ値からプライオリティ値を算出することができる。例えば、クライアントは、関連付けられたプライオリティ値がない新たなPUSH_PROMISEフレームが受信されるたびにプライオリティ値を増加させてもよい。従って、特定セグメントのプライオリティの更新をするとまたグループのすべてのセグメントのプライオリティを更新するように、PUSH_PROMISEフレームは、グループ化することができる。

【0092】

50

図9を参照してクライアント側の処理を説明する。

【0093】

クライアントは、所定の時間に利用可能なコンテンツを再生可能であるべきである。しかしながら、クライアントは、潜在的なバッファ制限および処理時間に対処しなければならない。クライアントは、サーバによって提案された送信順序付けがクライアントのバッファにおいて利用可能なメモリースペースと一致し、且つクライアントによって現在再生されているコンテンツと一致するか否かを、チェックしなければならない。

【0094】

最初のステップ900の間に、クライアントは、サーバに接続して、MPDファイルを要求する。そして、クライアントは、ステップ901の間にMPDファイルを取り出し、データの受信を待つ(ステップ902)。データが受信されると、クライアントは、データがプッシュプロミスであるか否かをチェックする(ステップ903)。プッシュプロミスが受信された場合、これは、新たなビデオセグメントがサーバによって送信されている、ということを意味する。クライアントは、プッシュプロミスを処理する。特に、クライアントは、ステップ904の間にサーバによって提案されたプライオリティ値を認証してもよい。クライアントが現在のセグメントまたは別のプロミスされたセグメントのプライオリティ値を変更したい場合(ステップ905)、クライアントは、新たなプライオリティ値を算出して、それをサーバに対して送信する(ステップ906)。

10

【0095】

クライアントがビデオデータを受信すると(ステップ907)、クライアントは、ビデオセグメントをMPDファイルにリンクし(ステップ908)、ビデオデータ(ステップ909)を保存する。ビデオデータをMPDファイルにリンクすると、ビデオの復号のためにさらに用いられる場合にクライアントがビデオセグメントを取りだすことが可能になる(ステップ911)。例えば連続したビデオセグメントがグループ化されると、これは、また、ビデオデータの効率的な保存を提供することができる(ステップ909)。

20

【0096】

バッファ保存制約は、さらにプライオリティを変更することができる。このように、クライアントは、プライオリティ値を変更しなければならないかどうかを再びチェックすることができ、もし必要ならばサーバと通信してもよい(ステップ905および906)。

【0097】

一旦クライアントがビデオを再生し始める、または再生し続ける準備ができれば(ステップ910)、クライアントは、そのキャッシュから次のタイムスロットのビデオセグメントを取り出し(ステップ911)、ビデオを復号して再生する(ステップ912)。ステップ911の一部として、クライアントは、どのビデオセグメントが利用可能であるのかを認識するためにそのキャッシュを問い合わせることができる。デフォルトでは、クライアントは、利用可能なすべてのビデオセグメント、もしあれば特にすべての強化セグメント、を用いてもよい。クライアントは、サーバにコンテンツを選択させることができ、一般的には言えば、すべてのセグメントは、クライアントによって用いられるべきである。(オーディオの英語トラックおよびフランス語トラックのような)いくつかのセグメントを共同で用いることができなければ、クライアントは、まず第1に未使用のセグメントを捨てるべきである。すべてのクライアントがキャッシュ状態に対してアクセスするとは限らないかもしれない、ということに留意するべきであり、特にウェブアプリケーションは、ウェブブラウザのキャッシュに対して通常はアクセスしない。このようなケースにおいて、サーバは、ウェブアプリケーションクライアントに対してプッシュされたセグメントのリストを直接送信してもよい。例えば、この情報は、ウェブソケット接続を用いて、サーバからクライアントに交換されてもよい。

30

【0098】

ビデオが再生されて復号されるとともに、対応するビデオセグメントは、バッファから取り出されるであろう。従って、クライアントは、WINDOW SIZEフレームを用いて、その利用可能なバッファサイズを更新する。クライアントは、ユーザが限られた期

40

50

間の間にビデオを巻き戻すことを可能にするために、最近再生されたビデオセグメントを保持してもよい。ユーザが早送り / タイムスキップを行う場合、フロー制御更新メカニズムも用いられてもよい。クライアントは、新たなコンテンツのために場所を空けるために、前から保存されたビデオコンテンツを取り出してもよく、WINDOW SIZEフレームを用いて、サーバに対してこの変更をアナウンスする。サーバがWINDOW SIZEフレームを受信すると、サーバは、前述のように、どのビデオセグメントが取り出されたのかを算出し、その後、クライアントが実際に何を再生しているのかを識別することができるであろう。

【0099】

以下において、ステップ904をさらに詳細に説明する。

10

【0100】

クライアントは、すべてのプッシュプロミスビデオセグメントのリストを保持する。このリストは、プッシュプロミスフレームにおいて見つかったプライオリティ情報にしたがって順序付けられる。第1に、それは潜在的な凍結されたビデオ問題のためにチェックされる。利用可能な帯域幅の推定および順序付けられたビデオセグメントリストに基づいて、各セグメントの送信の始まりおよび終わりの時刻を推測することができる。これらの時刻に基づいて、各ビデオセグメントはビデオ再生のために用いられる時刻に利用可能であるか否か、をテストすることができる。プロミスされたビデオセグメントがその対応するビデオ再生の使用の後に配送されることが予定される場合、そのプライオリティを上げるべきである。このように、このビデオセグメントは、プッシュプロミスビデオセグメントリスト内の順序が繰り上げられる。正確なプライオリティ値を算出するために、ビデオセグメントを時間通りに配送することを可能にするとともに、現在のビデオセグメントの位置に最も近い、ビデオセグメントリスト内における位置を検索する。そして、プライオリティは、ビデオセグメントの新たな位置の前後にあるリスト内のビデオセグメントのプライオリティ間の値に設定される。

20

【0101】

他のファクタも、また、ビデオセグメントのプライオリティの変更のために、クライアントによって用いられてもよい。例えば、クライアントがいくつかのチャプター切換を行うことを予定していれば、クライアントは、チャプターを開始するすべてのビデオセグメント、特に対応するイニシャライゼーションセグメント、のプライオリティを実際に拡大してもよい。

30

【0102】

実施形態によれば、クライアント側フロー制御は、ストリームごとのフロー制御を停止させて接続ごとのフロー制御のみを保持することを有する。接続ごとのウィンドウサイズは、クライアントが実際にいかなるときにも保存してもよいビデオの最大量を定義する。クライアントおよびサーバは、このウィンドウサイズを減少または拡大するためにイニシャライゼーション時や接続中にネゴシエイトしてもよい。サーバがいくつかのHDコンテンツをプッシュしたければ、サーバは、ウィンドウサイズを拡大することをクライアントに要求してもよい。接続帯域幅が低いならば、サーバは、ビデオの特定の部分のためのHDコンテンツを送り出すことを予め十分に予期する必要があるかもしれません、その場合には、バッファサイズは、より大きくされる。

40

【0103】

バッファが単一のサイズを有する場合、送信の順序は、重要な問題であるであろう。特に、バッファがデータでいっぱいになった時に、プライオリティ順序付けは、より一層重要になる。重要な制約は、ビデオがフリーズしないということである。バッファが大きく空いている限り、サーバは、効率的な早送りまたはチャプタースキップを提供するために、かなり先行したセグメントのような様々なビデオセグメントを予めプッシュしてもよい。一旦バッファがほぼ満タンになれば、プッシュするビデオセグメントは再生されるビデオセグメントになるべく近いるべきである。サーバがクライアントバッファに関係する正確な情報を有していれば、このプッシュ挙動は、サーバによって行われてもよい。また、

50

それは、プライオリティ更新メカニズムを用いて、クライアントによって実施されてもよい。

【0104】

自動化されたビデオ切換の場合には、図9のフローチャートは、プッシュプロミスチェックの一部として新たなMPDのプッシュを検出することによって拡張されてもよい(ステップ903)。MPDプッシュが検出されると、クライアントは、ステップ908の一部として新たなビデオのセグメントを受信し始めてよい。そのため、クライアントは、ビデオデータに関連するMPDを識別しなければならない。一旦ビデオ再生が所定のMPD(ステップ902)のために終了されると、新たなMPDは、ビデオ再生の継続のために用いられる。クライアントは、先のMPDにリンクされたビデオセグメントをすべて実際に消去することができる。10

【0105】

図10を参照して、DASH認識プロキシの挙動を説明する。サーバからプッシュされたセグメントを受信すると、プロキシは、エンドクライアントに対してそれをプッシュするようには要求されない。しかしながらDASHストリーミングの場合には、それは優れた方式(またはデフォルトの挙動)と考えられる。

【0106】

プロキシは、送信されるプッシュされたデータだけでなくプライオリティ処理に関して、サーバおよびクライアントの挙動を調整することができてもよい。プロキシは、実際、クライアントによるプライオリティを、サーバによるプライオリティから独立して取り扱ってもよい。さらに、サーバは所定のクライアントのために必要とされる以上のデータをプッシュしてもよく、プロキシは、他のクライアントからの要求を実現するために付加的なプッシュされたデータを取り込んでもよい。20

【0107】

サーバは、いくつかの理由によりビデオセグメントをプッシュしてもよい。例えば、それがエンドクライアントのために有用であると考えられた場合、ビデオセグメントがプッシュされてもよい。ビデオセグメントは数回使用することが可能で、且つプロキシに対してそれをプッシュする価値があると考えられる場合、ビデオセグメントもプッシュされてもよい。

【0108】

第1のケースにおいて、プロキシは、一般的には、クライアントに対してビデオセグメントを送信する。プロキシは、クライアントまたはプロキシのネットワーク状態(例えばクライアント無線状態)を最適化するためにその送信を先延ばししてもよい。例示的なケースは、高速起動ビデオ再生および帯域幅推定のためのセグメントプッシュであり、その場合には、データはクライアントに対して可能な限り高速に送信されるべきである。サーバがプロキシに対するデータのプッシュに关心がある場合、プロキシは、ビデオセグメントがクライアントに対して有用になることを認識する手段を有する場合以外は、クライアントに対してビデオセグメントを自動的に送信しなくてもよい。クライアントに対して送信されなくてもよいビデオセグメントの識別を可能にするために、特定のプライオリティ値を用いてもよい。プライオリティ値を用いると、到達する様々なフレームの処理を最適化するためにプロキシが常にプライオリティ値をチェックすることができる。3040

【0109】

図10は、3つのフローチャートを含む。1つのフローチャートは、プッシュされたセグメントをフィルタリングする処理に関する(ステップ1000～1008)。もう1つのフローチャートは、セグメントが別のクライアントに対して既にプロミスされている一方で、セグメントがクライアントによって要求された場合に実行される処理に関する(ステップ1010～1015)。もう1つのフローチャートは、プライオリティ変更の管理に関する(ステップ1020～1026)。

【0110】

プッシュされたセグメントをフィルタリングする処理は、プッシュされたデータイベン50

トの受信（ステップ1000）とともに、通常は P U S H _ P R O M I S E フレームまたは関連する D A T A フレームを受信したときに、始まる。プロキシは、データが高プライオリティであるか否かをチェックする（ステップ1001）。データは、それらのプライオリティ値が送信されている他のセグメントのプライオリティ値よりはるかに大きい場合、高プライオリティであると考えられる。データは、また、そのプライオリティ値が高速起動または帯域幅推定などの特別の意味を有するならば、高プライオリティであるとして考えられる。データが高プライオリティである場合、それらはクライアントに対して可能な限り迅速に送信される（ステップ1002）。そして、プロキシは、データを保存するべきであるか否かを決定する（ステップ1003および1004）。プッシュされたデータストリームを開く対応する P U S H _ P R O M I S E フレームまたは対応する H E A D E R S フレームを受信すると、この判定は、一回行われてもよい。この判定は、また、プロキシのキャッシュ状態、ビデオの予見された使用、ビデオソースの人気度、または他の基準に基づいてもよい。セグメントが1つ以上のクライアントによって同時に要求される間にプッシュされれば、プロキシは、ビデオセグメントを保存する。セグメントが高速起動として識別されれば、ビデオセグメントもまた保存されてもよい。
10

【0111】

データが高プライオリティでないならば、プロキシは、それが低プライオリティであるか否かをチェックする（ステップ1005）。低プライオリティのデータは、クライアントに対する送信がスキップされてもよいが、プロキシのようなネットワーク媒介に対して興味深いものとしてサーバによって考慮されるデータかもしれない。プロキシは、まず、クライアントに対してデータを送信するべきか否かを決定する（ステップ1006）。プッシュされたデータストリームを開く対応する P U S H _ P R O M I S E フレームまたは対応する H E A D E R S フレームを受信すると、この判定は、一回行われてもよい。そのように決定されれば、プロキシは、クライアントに対して、対応するフレームを送信する（ステップ1002）。そして、処理は、データを保存するべきか否かを決定した後に停止する。
20

【0112】

サーバとプロキシとの間でネゴシエイトされたプライオリティ値は、また、クライアントとプロキシとの間でネゴシエイトされたプライオリティ値とは異なってもよい。そのため、データが通常のプライオリティである（すなわち、低プライオリティでもなく、高プライオリティでもない）場合、プロキシは、セグメントのプライオリティ値がプロキシによって管理されるか否かをチェックする。図10に図示されるように（ステップ1020～1026）、プロキシは、データが送信されるべき時刻をスケジューリングするためにクライアント～プロキシ間の値を用い、プロキシは、送信されるべきすべてのビデオ関連のフレームのリストを保持する。これらのフレームは、その順序にしたがって送信される前に、プライオリティ値にしたがって順序付けられる。
30

【0113】

プロキシは、プライオリティ更新フレームを受信している場合に（ステップ1010）、プロキシは、関連するビデオセグメントを識別する（ステップ1011）。そのプライオリティ値がプロキシによって管理されていなければ（ステップ1012）、プロキシは、サーバに対してプライオリティ更新フレームを転送する（ステップ1013）。さもなければ、プロキシは、この新たなプライオリティ値を保存し、それに応じて、ビデオセグメントの送信を再順序付ける（ステップ1014）。潜在的な競合が出現した場合、特にサーバからのビデオセグメントの配信がクライアントのニーズに対して遅れすぎると予想される場合、プロキシは、サーバに対してプライオリティ値を転送することができる。
40

【0114】

ステップ1020～1026は、サーバによって既にプロミスされているビデオセグメント（ステップ1020）のクライアントから別のクライアントに対する要求を受信するプロキシのケース（ステップ1021）に関連する。その要求に対して与えられるプライオリティに応じて、プロキシは、クライアントの要求を満たす、プロキシ～サーバ間の最小
50

のプライオリティを算出する（ステップ1022）。この計算は、サーバ～プロキシ間の配信時刻がプロキシ～クライアント間の予定した配信時刻よりも早いということを保証する、プロキシ～サーバ間のプライオリティ値を算出することによって行われる。プライオリティは、算出されたプライオリティが現在設定されているプライオリティを下回る場合に、変更され（ステップ1023）、その場合には、プロキシは、プライオリティ更新メッセージをサーバに対して送信することになり（ステップ1024）、プロキシは、プロキシがそれらのニーズに対して最良の時刻にその2つのクライアントに対してビデオセグメントを送信するように、プロキシによって管理されるようなこのビデオセグメントプライオリティをマークすることになる。この処理と同様に、プロキシは、いくつかのクライアントから同じセグメントに対する、いくつかのプライオリティ更新を受信してもよく、その場合には、プロキシは、すべてのクライアントを満たす最も低いプライオリティ値を実際に送信してもよい。10

【0115】

図11を参照して、プッシュされたデータイベントをクライアントが受信することによる実施形態が説明され、そのデータイベントのプライオリティ値は、サーバが帯域幅の測定のためにそれを用いたいということを示す。帯域幅の測定は、ラウンドトリップ時間を算出するための能動的または受動的な測定を通じてTCP/IPパケットを用いて行われてもよい。ラウンドトリップ時間に基づいて、利用可能な帯域幅は、Saubhasik他の文献「BitVampireにおける帯域幅推定および速度制御（Bandwidth Estimation and Rate Control in BitVampire）」に見出されるように、算出されてもよい。この計算は、場合によっては、HTTP/2制御フローの影響を考慮に入れてもよい。いくつかのデータフレームが可能な帯域幅推定のために用いられる、という通知を行うことによって、HTTP/2制御フローなしで、利用可能な帯域幅を推定することができる。20

【0116】

処理は、プッシュされたデータフレームがサーバから受信される間のステップ1100にて始まる。次に、ストリームの関連付けられたプライオリティが、サーバが帯域幅を測定しているということを示すか否かがチェックされる（ステップ1101）。その場合、専用のバッファは最大化される（ステップ1102）。または、ストリームフロー制御を無効にすることもできる。受信ノードがプロキシであるならば（ステップ1103）、セグメントデータを転送してもよい。さもなければ、クライアントは、セグメントを保存するべきか否かを決定する（ステップ1104）。クライアントは、プッシュされたセグメントを保存する（ステップ1105）。いずれにしても、クライアントは、接続ごとのウィンドウのための肯定応答をWINDOWS（登録商標）_UPDATEの形式でサーバに対して送信する（ステップ1106）。そして、この肯定応答は、接続帯域幅を推定するためにサーバによって用いられることになる。クライアントがプロキシで有る場合、プッシュされたデータを可能な限り迅速に転送する（ステップ1108）。エンドクライアントから肯定応答を受信すると、プロキシは、同様にそれをサーバに返送する（ステップ1109および1110）。30

【0117】

利用可能な帯域幅を推定するために、サーバは、データフレームの送信時刻と肯定応答メッセージの受信時刻との間の差分として算出される、送信されたデータフレームのラウンドトリップ時間を用いてもよく、2者間のペアリングは、例えばウィンドウサイズ更新に等しくなるべきデータフレームサイズに基づく。ラウンドトリップ時間は、1つ以上のビデオセグメントの様々なデータフレームから算出される。正確さを増すために、データフレームは、様々なサイズを有してもよい。異なるサイズのいくつかのDATAフレームにビデオセグメントを分割することは、サーバによって実行できる。サーバは、ネットワーク層が、DATAフレームを、いくつかのTCP/IPパケット（したがって小型のDATAフレーム）に分割しない、または送信されるコンテンツをバッファリングして、いくつかのDATAフレームをTCP/IPパケットにマージしない、ということを保証する4050

ことのみ必要とされる。それらの測定値に基づいて、規格技術は、どのビデオリプレゼンテーションを用いるかを実際に判断するために、サーバが用いてもよい利用可能な帯域幅を算出するために用いることができる（例は上記の文献内で見出すことができる）。

【0118】

図12を参照して、初期のビデオ再生のケースを説明する。サーバは、高速起動プライオリティを用いて、データをプッシュする。データは、通常低ビットレートを有しており、クライアントは、サーバが帯域幅を推定することができ、且つ最適なリプレゼンテーションに切り替えることができるよう、それらのデータを受信して肯定応答をサーバに対して送信することになる、と考えられる。クライアント側の処理はステップ1200～1207に記載される。サーバ側の処理は、ステップ1210～1215に記載される。

10

【0119】

クライアント処理は、プッシュされたデータの受信のステップ1200から始まる。そして、クライアントは、プライオリティが高速起動の値を有するか否かをチェックする（ステップ1201）。その場合、クライアントは、通常は、専用のバッファを最大化する（ステップ1202）。プッシュされたデータのPUSH_PROMISEを受信すると、この最大化が実行される。そして、データが保存され（ステップ1203）、クライアントは、WINDOW_UPDATEフレームを用いて、肯定応答をサーバに送信する（ステップ1204）。そして、クライアントは、ビデオを再生し始めるために十分なデータが利用可能であるか否かをチェックする（ステップ1205）。そうであるならば、ビデオ再生が始まる（ステップ1206）。そうでなければ、データを再生し始めるために十分なデータが利用可能になるまで、クライアントは、さらなるデータを待つ（ステップ1207）。

20

【0120】

サーバ処理は、高速起動プライオリティでセグメントデータフレームを送信するステップ1211から始まる（ステップ1210）。そして、サーバは、利用可能な帯域幅を算出すること（ステップ1212）を可能にする肯定応答を受信する（ステップ1211）。一旦十分な測定値が取得されれば、サーバは、最適なリプレゼンテーションを選択し（ステップ1213）、最適なリプレゼンテーションセグメントをプッシュし始める（ステップ1214）。サーバは、リプレゼンテーションをいつ切り替えるのかを決定する。これには少なくとも2つの利点がある。第1に、クライアントが多少の遅延に対処する必要がある一方で、サーバは、いつ測定値が十分に正確になるのかを認識することができ、このケースであれば直ちに1つの解像度から別の解像度に切り替えることができる。第2に、サーバは、ユーザエクスペリエンスをあまり妨害しない時刻に1つの解像度から別の解像度に切り替えることを決定することができる。実際には、サーバは、ビデオコンテンツの認識を有する。特に、MPDは、解像度の切り替えを最良に予見することができる時刻の情報により強化されてもよい。

30

【0121】

本発明は、強化されたストリーミング方法に関し、サーバ側において、第1のメディアデータに関連する要求がクライアント装置から受信され、要求されていなくともクライアント装置に対して送信される第2のメディアデータが識別され、そして、前記第1のメディアデータに関連する情報が前記要求に応答して前記クライアント装置に対して送信され、クライアント装置に対してアナウンスマッセージ（単数または複数）を送信するために前記第2のメディアデータをそれぞれ識別する少なくとも1つのアナウンスマッセージが準備される。

40

【0122】

クライアント側において、第1のメディアデータに関連する要求は、サーバ装置に対して送信され、前記第1のメディアデータに関連する情報は、前記要求に応答して前記サーバ装置から受信される。

【0123】

50

強化されたストリーミング方法は、いくつかのメディアデータをプッシュするサーバの判定と、このようなデータのためのクライアントのニーズとの間のミスマッチを減少させる。以下から明らかになるように、サーバおよびクライアントは、サーバおよびクライアントが両方ともクライアントによって要求されたいずれかのメディアデータからプッシュされる同じメディアデータを決定するように、プッシュポリシーを共有する。プッシュポリシーは、プッシュするデータを決定する方法を定義しており、要求されたデータが処理された（GET要求の後に）後に、要求されたデータに対してリンクされたどのリソースがプッシュされるのか、および場合によってどのようにプッシュされるか（例えば、どの順序で）を判定するための規則として理解されてもよい。通常、リンクされたリソースは、1つのドキュメント、例えば、（マルチメディアデータのためのDASHコンテキストにおける）MPDファイルなどのマニフェストファイル、またはHTMLドキュメント、を用いて決定される。

【0124】

結果として、共有されたプッシュポリシーに基づいて、クライアントはサーバの挙動を予期し、サーバからの無用のメディアデータの送信を、回避する、そしてより正確にはキャンセルすることができる。クライアントとサーバとの間の通信ネットワーク内の帯域幅の使用が、このようにして減少する。さらに、HTTP要求およびPUSH_PROMISEキャンセルの数が減少し、その結果、特に低遅延のライブビデオストリーミング用のアプリケーションの待ち時間が下がる。

【0125】

本発明によれば、サーバは、サーバ装置がクライアント装置に対する第2の非要求メディアデータの識別および送信を駆動するためにクライアント装置と共有されるプッシュポリシーを、用いることができる。特に、クライアント装置に対して送信される第2の非要求メディアデータサーバ装置が決定するために、クライアント装置と共有されるとともに第2のメディアデータを決定する方法を定義するプッシュポリシーを用いてもよい。それに対応して、クライアント装置によって要求されていないともサーバ装置によって送信される第2のメディアデータをクライアント装置が決定するために、クライアントは、サーバ装置と共有されるとともに第2のメディアデータを決定する方法を定義するプッシュポリシーを用いてもよい。

【0126】

図14aは、クライアント側における本発明の一般的なステップを、フローチャートを用いて示し、その一方で、図14bは、サーバ側における本発明の一般的なステップを、フローチャートを用いて示す。

【0127】

図1dおよび図1eを参照して説明された処理との比較において、付加的なステージ1400および1402は、サーバおよびクライアントが、他方と共有されて、それにより用いられるプッシュストラテジをそれぞれ決定することを可能にする。

【0128】

第1の実施形態によれば、共有されたプッシュポリシーは、共有されるプッシュポリシーが何であるのかを他方に知らせるためにクライアントおよびサーバがポリシーデータを交換しない（明示的でない）ということを意味する、暗示的なプッシュポリシーである。共有されたプッシュポリシーの暗示的アプローチの実施は、サーバ装置およびクライアント装置の両方において、「第2のメディアデータを決定するアルゴリズム」として呼ばれる同じアルゴリズムを用いることを含み、アルゴリズムは、サーバ装置およびクライアント装置が、要求された第1のメディアデータから同じ第2のメディアデータを決定することを可能にする。

【0129】

例えば、アルゴリズムは、クライアントおよびサーバのセットアップの間に、または特定の規格に関連して、予め決定される。アルゴリズムの代表例は、マニフェストファイルの構文解析順序における要求されたリソースに追続くN個のリソースのプッシュする場合で

10

20

30

40

50

あり、ここで、Nは所定数（例えば4）である。

【0130】

図を参照して、ステップ1400および1402は、暗示的なプッシュポリシーのケースでは、プッシュされるリソースを識別する（サーバ側におけるステップ1403）ためにメモリ内の所定のアルゴリズムをロードすることにある。

【0131】

クライアントは、例えばステップ1401において、期待されるPUSH_PROMISEの数を推定し、且つ不要なプッシュデータのためにキャンセルメッセージを準備するよう決してされたプッシュポリシーを効率的に用いてよい。

【0132】

例えば、サーバ装置が対応する用意されたアナウンスマッセージを送信しないように、第2の非要求メディアデータの一部の送信をキャンセルすることを要求するキャンセル要求をサーバがクライアント装置から受信することになる。一方で、クライアントは、第2の非要求メディアデータの一部を識別するアナウンスマッセージを送信しないようにサーバ装置を駆動するために、第2の非要求メディアデータの一部の送信をキャンセルすることを要求するキャンセル要求をサーバ装置に対して送信するであろう。アナウンスマッセージがサーバ装置から送信されるか、またはクライアント装置によって受信される前に、このようなキャンセルが行われることができる、ということをわかるであろう。このアプローチは、例えば、別のバージョンのメディアに切り替えることをクライアントが決定するときに、有用かもしれない。このような状況において、前バージョンのためにプッシュされたセグメントをキャンセルすることを決定することができる。

10

【0133】

アルゴリズムを用いてプッシュされるリソースの認識のために、クライアントは、サーバから対応するPUSH_PROMISEを待つ必要なく後続のリソースを取りだすために、サーバに対して第2の要求を並列に行うことができる、ということにも留意するべきであろう。DASHの場合には、クライアントのためのこの可能性は、第2の要求が後で受信するPUSH_PROMISEに干渉しないことを保証しながら、クライアントの待ち時間を減少させることを可能にする。

【0134】

これらの他の必要とされるリソースがプッシュされようとしている、ということをアルゴリズムの結果から判定すれば、クライアントは、また、必要とする他のリソースを要求してもよい。

30

【0135】

第2の実施形態によれば、共有されたプッシュポリシーは、全体の規則（すなわちアルゴリズムまたはアルゴリズムのパラメータ）を明確に定義するか、または両方の側で事前に定義されたプッシュポリシーを参照することによって、クライアントとサーバとの間の交換において定義される。これは、サーバのプッシュポリシーを記述するプッシュポリシー情報をサーバがまず決定することが必要となる。そして、プッシュポリシー情報は、プッシュポリシーをクライアントと共有するために、クライアントに対して送信される。それに対応して、クライアントは、それにより、共有されたプッシュポリシーを記述するプッシュポリシー情報をサーバ装置から受信する。

40

【0136】

明示的アプローチの1つの効果は、それらの処理特性をより適切に満たすために異なるプッシュポリシーを、各クライアントまたは各マルチメディアプレゼンテーション（例えば各MPD）のためにサーバが用いることができるという事実に依存する。図15aは、明示的アプローチに基づいて、クライアント側において共有されるプッシュポリシーを決定するステップ1400を、フローチャートを用いて示し、その一方で、図15bは、明示的アプローチが用いられるときに、サーバ側においてプッシュポリシーを決定するステップ1402を、フローチャートを用いて示す。

【0137】

50

図15bに示されるように、サーバは、ステップ1504において、プッシュポリシーを宣言するメッセージを生成し、その後、それを共有するために、ステップ1505においてクライアントに対してそれを送信する。宣言メッセージ内のプッシュポリシーを記述する情報は、プッシュポリシー情報と称される。

【0138】

以下に記載される図16～図18は、プッシュポリシーがどのように宣言され、クライアントに対して送信されるのかについて例示的な詳細を示す。

【0139】

ステップ1402において決定されるようなプッシュポリシーを用いてプッシュされるリソースは、ステップ1504において生成されるプッシュポリシー宣言メッセージ内に定義されたセレクションアルゴリズム（または第2のメディアデータを決定するアルゴリズム）により、ステップ1403において識別される。10

【0140】

クライアント側では、図15aに示されるように、クライアントは、同じ選択アルゴリズムを適用することによって、所定のリソース要求のためにプッシュされるリソースを、事前に識別することができる。これは、サーバによってプッシュされるデータをクライアントが予め決定することを可能にし、したがって、プッシュデータの効率的な管理と、必要であればGET要求の数の削減とを保証することを可能にする。

【0141】

同じ選択アルゴリズムを適用するために、クライアントは、サーバによって適用されたプッシュポリシーを記述するプッシュポリシー情報を受信する。20

【0142】

様々なプッシュポリシーの宣言方法が用いられてもよい。

【0143】

1つの実施形態において、プッシュポリシー宣言は、要求Rと、プッシュされるリソースを含むドキュメント（通常はDASHのためのマニフェストファイル）に対応するDOMツリーとを入力パラメータとして利用して、プッシュされるリソースの順序付きリストを出力するJava（登録商標）Scryptプログラムのおかげで共有される。この実施形態において、プッシュポリシー情報は、サーバ装置からクライアント装置に対して送信されるウェブページ内に埋め込まれるJava（登録商標）Scryptプログラムを含む。30

【0144】

他の実施形態において、プッシュポリシーは、マニフェストファイル内に記述される。それは、共有されたプッシュポリシーを用いて、サーバ装置からクライアント装置に対して送信される記述ファイル内の共有されたプッシュポリシーが挿入されることを記述するプッシュポリシー情報である。記述ファイルは、第1のメディアデータを含むメディアデータに関係し、プッシュされる第2の非要求メディアデータを決定するために両方の側によって用いられる記述情報を含む。

【0145】

DASHにおいて、記述ファイルは、例えばMPDファイルである。以下の記述は、主として、DASHおよびMPDファイルに基づく。しかしながら、同じアプローチは、スマートなストリーミングまたはHTTPライブストリーミングのような他のマニフェストベースのストリーミング方法に適用できる。40

【0146】

具体的な定の実施形態によれば、プッシュポリシー情報は、識別される第2の非要求メディアデータの量を定義する第1のプッシュ属性を記述ファイル内に含む。これは、1つの要求Rがクライアントから受信された後、プッシュされるセグメントの数を指定することを可能にする。

【0147】

これは、サーバによって適用されるプッシュポリシーを指定するためにPushPol50i

c y ノード 1 6 0 0 が用いられる M P D ドキュメントを示す図 1 6 によって示される。

【 0 1 4 8 】

この例において、P u s h P o l i c y ノード 1 6 0 0 は、G E T 要求が受信された後にプッシュされるセグメントの数を宣言するために、プッシュ属性（すなわち「S e g m e n t 1 d x 」）を含む。例えば、クライアントがそのG E T 要求内のセグメント 1 6 0 1 を必要とするならば、M P D ドキュメントの順序を解析する際に、次の2つのセグメントのためのP U S H _ P R O M I S E フレームを応答として、受信することになる。この例において、第1のプッシュ属性は、記述ファイル内の要求された第1のメディアデータに対して第2の非要求メディアデータを識別する。さらに一般的には、プッシュされるK個のセグメントの所定数は、プッシュポリシー値を定義するために用いられる。結果的に、クライアントによって要求される各セグメントに対して、サーバは、Kの次のセグメントをプッシュすることになる。
10

【 0 1 4 9 】

図 1 6 の例 1 6 0 0 は、単一のプッシュ属性を示すが、複数のプッシュ属性があつてもよい。各プッシュ属性は、プッシュされるセグメントを選択するために、マニフェストを表現するD O M (ドキュメントオブジェクトモデル (D o c u m e n t O b j e c t M o d e l))ツリーのノード上の制約を表現してもよい。図 4 b の先の例を参照すると、プッシュポリシーノード 1 6 0 0 は、メディアデータが属するP e r i o d 要素を参照する期間属性「P e r i o d 1 d x 」と、メディアデータのA d a p t a t i o n S e t 要素を参照するアダプテーション属性「A d a p t a t i o n S e t 1 d x 」と、R e p r e s e n t a t i o n 要素を参照するリプレゼンテーション属性「R e p r e s e n t a t i o n 1 d x 」、すなわちメディアデータの符号化バージョン（特定のコーデック、解像度、またはビットレート…）と、所定のリプレゼンテーション内のセグメントを参照するセグメント属性「S e g m e n t 1 d x 」と、を含むメディアデータ属性（M P D 要素および / または属性）を用いて、記述ファイル（M P D ファイル）内に記述されるメディアデータを参照することができる。
20

【 0 1 5 0 】

これらの既存のメディアデータ属性に基づいて、プッシュポリシー情報は、第2の非要求メディアデータを識別するために、メディアデータ属性（单数または複数）上の制約を定義する少なくとも第2のプッシュ属性を含んでもよい。
30

【 0 1 5 1 】

例えば、S e g m e n t 1 d x 属性に関連する上記の第1のプッシュ属性に加えて、プッシュするセグメントを選択するための期間上の制約を指定するために、プッシュ属性は、P e r i o d 1 d x 属性に関連付けられてもよく、別の1つは、適応上の制約を指定するためにA d a p t a t i o n S e t 1 d x 属性に関連付けられてもよく、別の1つは、リプレゼンテーション上の制約を指定するためにR e p r e s e n t a t i o n 1 d x 属性に関連付けられてもよい。

【 0 1 5 2 】

プッシュ属性が存在しないかまたは無効のとき、関連するメディアデータ属性は、非制約として考えられねばならない。
40

【 0 1 5 3 】

プッシュ属性の値は、以下の構文を用いてよい。

【 0 1 5 4 】

プッシュ属性 = [オペレータ] オペランド (p u s h a t t r i b u t e = [o p e r a t o r] o p e r a n d)

ここで、「オペレータ」は、オプションであり、要求されたセグメントに対して相対的にプッシュされるセグメントを定義するために、値「+」または「-」をとり（「+」は後を意味し「-」は先を意味する）、ここで、「オペランド」は、ワイルドカードパラメータとして 0 または「< * >」よりも上位の整数値若しくは等しい整数値のいずれかである。

【0155】

図17は、共有されたプッシュポリシー「Push Policy」にしたがってプッシュされる準備ができている、いくつかのセグメントを識別してマークするステップを、フローチャート用いて示す。このフローチャートは、ステップ1403を示す。

【0156】

まず、サーバは、ステップ1700において、マニフェストファイル内の要求されたセグメントを識別する。要求は、このセグメントの識別された「req Seg1dX」を含む。

【0157】

マニフェストファイルMPD内の各々のノード種別に対して、指標値が、各ノードに対して属性付けられる。値は、マニフェストファイル内での出現順において各々のノードごとに増加される。10

【0158】

次に、要求されたセグメント（すなわちGET要求において指定されたセグメント）に対応するPeriod、AdaptationSet、Representation、およびSegmentURLのインデックスは、要求されたセグメントに到達されるまで、全体のMPDを解析することによって取りだされる。

【0159】

プッシュポリシー内に定義されたプッシュ属性のオペレータおよびオペランドの値は、（「+」または「-」オペレータに関連付けられている場合に、プッシュされるセグメントの量を定義するSegment1dX属性を除いて）プッシュされるセグメントがどのノード内で定義されるのかを識別するために用いられる。20

【0160】

オペレータが指定されない場合は、オペランド値は、プッシュされるデータを検索しなければならないNodeのインデックスを識別する。例えば、第1のプッシュ属性「Segment1dX」がオペレータを有しない場合には、それは、プッシュされる特定セグメントの、記述ファイル内部の、識別子である。1つの代替案において、オペレータが指定されないと、オペランド値は、範囲値を識別してもよく、例えば、「Segment1dX = 2 - 5」は、2、3、4および5に等しいインデックスでセグメントを返すだろう。

【0161】

そうでなければ（オペレータが指定されている）、オペランド値は、要求されたセグメント（ステップ1700において取得された「req Seg1dX」）のインデックスに対して適用するために（「idxFOffset」と命名される）オフセット値を表現する。そのような場合、プッシュされるセグメントは、もしオペレータが「+」ならば[req Seg1dX, req Seg1dX + idxFOffset]範囲、そしてもしオペレータが「-」ならば[reg Seg1dX - idxFOffset, reg Seg1dX]内に備えられたインデックスとともにノード内にあるはずである。オペレータの使用は、記述ファイル内部の第1のメディアデータの対応するメディアデータ属性（単数または複数）に対する第2の非要求メディアデータのメディアデータ属性（単数または複数）を定義することを可能にする。30

【0162】

例えば、以下のプッシュポリシーを考えてみよう。

【0163】

```
1. <PushPolicy Representation1dX = "-1" Segment1dX = "2" />
2. <PushPolicy Period1dX = "+1" Segment1dX = "+2" />
3. <PushPolicy Period1dX = "+0" Segment1dX = "+2" />
```

PushPolicy #1は、要求されたセグメントのリプレゼンテーションノードに先40

行するリプレゼンテーションノード内のインデックス 2 のセグメントをサーバがプッシュすることになる、ということを指定する。

【0164】

`PushPolicy # 2` により、サーバは、現在の期間または以降において、要求されたセグメントに続く 2 つのセグメントをプッシュすることになる。例えば、図 24 のセグメント 2401 を要求すると、セグメント 2405 および 2402 がプッシュされることになる。

【0165】

`PushPolicy # 3` は、`PushPolicy # 2` に対して非常に類似し、主な差は、要求されたセグメントが `Period` の最後から 2 番目の時のものである。例えば、2401 を要求する場合、現在の期間における最後セグメント 2405 のみが（2 つのセグメントの代わりに）プッシュされることになる。`PushPolicy # 3` により、`Period1dx` は、要求されたセグメントの `Period` ノードに対してセグメント検索を制限しており、したがって、（要求されたセグメントが `Period` 内の最後から 2 番目のセグメントであるので）`Period` の最後のセグメントのみがプッシュされる。それどころか、`PushPolicy # 2` により、セグメントは、次の期間から取りだすことができる。

10

【0166】

代替案において、またはオプションの値として、オペランドの値は、また、あらゆるセグメントがプッシュされるべきであるということを意味する「`<*>`」（ワイルドカードを意味する）であってもよい。オペレータ「`+`」（それぞれ「`-`」）に関連付けられていると、それは、要求されたものに対して後続する（各々が先行する）すべてのセグメントがプッシュされるべきであるということを意味する。

20

【0167】

この代替案は、クライアントが `PushPolicy : <PushPolicy Period1dx = "+0" Segment1dx = "+<*>">` により、例えば 1 つの `Period` のセグメントをすべて取りだす、単一の HTTP 要求のみを送信することを可能にする。

【0168】

これらの例において、要求された第 1 のメディアデータに対する（プッシュされる）第 2 のメディアデータを識別する `Segment1dx` 属性の使用は、第 2 のメディアデータが第 1 のメディアデータに隣接することを必要とする。実施形態において、`Segment1dx` 属性は、要求されたセグメントのインデックスに適用するために（オペランドに加えて）オフセットを含んでもよい。これは、所定量のセグメントをプッシュしなければならない基準セグメントのインデックスをシフトする。例として、`Segment1dx` 属性の構文は、次のとおりであっても良い。

30

【0169】

プッシュ属性： [オペレータ] オペランド [, オフセット] (`push attribute : [operator] operand [, offset]`)

ここで「オフセット」は、要求されたセグメントインデックスに適応するために 0 とは異なる正または負の整数である。そのような場合、サーチ範囲は、オペレータが「`+`」であれば `[reqSeg1dx + offset, reqSeg1dx + idxOffset + offset]` 、およびオペレータが「`-`」であれば `[reqSeg1dx - idxOffset + offset, reqSeg1dx + offset]` である。

40

【0170】

プッシュポリシーの構文は、また、プッシュされているプレゼンテーション内に最大サイズのデータまたは時間のような（非制限的な）条件をそれぞれ含むことができる。例えば

、
`<PushPolicy Segment1dx = '+<*>[size<500000]>` は、多くとも 500 キロバイトのセグメントデータをプッシュするプッシュポリシ

50

ーを定義する。

(0 1 7 1)

`<PushPolicy Segment1dx = ' + <*> [time<0:01:30]>`は、多くとも1分30秒の次のセグメントデータを転送するプッシュポリシーを定義する。

[0 1 7 2]

上記の例が、どのセグメントをプッシュしなければならないのかを判定するプッシュポリシーを宣言する方法を示している一方で、セグメントがプッシュされることになる好ましい順序を指定する必要もあるかもしれない。この情報も、クライアントとサーバとの間で共有されるべきである。

10

【 0 1 7 3 】

例として、図7～図12を参照して上記したプッシュされたセグメントの送信の順序の宣言を、適用することができる。

(0 1 7 4)

プッシュされたセグメントの送信の順序の1つの代替の実施形態において、記述ファイル内の記述情報は、メディアデータに関連付けられたプライオリティ属性を含み、各々のメディアデータごとに1つのプライオリティ属性（例えば「priority1d×」）があり、第2のメディアデータの送信の順序は、関連するプライオリティ属性に基づく。記述ファイルの送信のおかげで、クライアントは、また、これらのプライオリティ属性によって得られた値に気づいており、したがって、意図された送信の順序を決定することができる。

30

[0 1 7 5]

図 16 の例に示されるように、マニフェストファイル内に記述される各セグメント（例えば、1つの Segment URL ノードによって識別される）は、セグメントのプッシュ順序を指定する priority_ldx 属性（1604）を含む。図 16 の例において、セグメント 1603 は、セグメント 1602 の前にプッシュされる。これらのプライオリティは、サーバ側においてメディアセグメントの準備の間に算出される。異なるプライオリティ値が使用出来る：32ビット数として、Period プライオリティのための4最上位ビット、AdaptationSet プライオリティ値のための次の4MSB、リプレゼンテーションプライオリティ値のための次の8ビット、およびセグメントプライオリティのための最下位 16 有効ビットをもつ（図 16 のような）所定の Representation 内の相対プライオリティ値または絶対プライオリティ値のいずれかを用いることができる。絶対プライオリティ値を示す別な方法は、プライオリティ値のコンマ区切りリストを用いることであり、上記の引用されたレベルの各々の1つは、例えば、Period プライオリティ、AdaptationSet プライオリティ、Representation プライオリティ、そしてセグメントプライオリティを連続的に定義する priority_ldx = '1, 1, 2, 1' である。第 1 の実施形態は、以下の 32 のビット値（バイナリ型）により与えるだろう。

30

[0 1 7 6]

```
priority_idx = '00010001000000010000000000000000  
00001'
```

10

`priority` `ldx` 値を用いる主な効果は、個別の `Representation`（通常は、ビデオの代替ビューなどの関連リプレゼンテーション）からセグメント間のプライオリティ順を定義することを可能にすることである。プッシュポリシーが個別の `Representation` セットのセグメントの送信にあると有用である。1つの層からのセグメントが1つ以上の他の層をもつセグメントによりインターリープされる場合、通常の使用事例は、層状ビデオ（マルチビューにおけるビューである層またはスケーラブルビデオにおけるスケーラビリティ層）のストリームのためのものである。

(0 1 7 7)

図1.7に示すように、サーバは、MPDファイル内に定義されるようなプッシュポリシーに基づいて音楽を配信する。

50

いて、ステップ1701において、プッシュされるセグメントの数を決定する。この数は、Segment1d属性値から直接推測される。すなわち、オペレータが属性値内で用いられなければ、この数は1に等しく、そうでなければ（オペレータは「-」または「+」である）、数は、オペランド値に等しく、オペランドが「<*>」であれば無限である（但し他の制約によって、および既存のセグメントの数によって、限定される）、と想定される。

【0178】

次に、プッシュするセグメントの数がプッシュされるセグメントの各々をマークするところに達する（テスト1702）まで、ステップ1702～1705から構成される反復処理は、ストリーミングサーバによって適用される。10

各々の反復に対して、サーバは、ステップ1703において、Push Policy制約（Adaptation Set、Representation、Period、およびSegment制約およびオプションの条件）に関する、MPDファイル内に定義されたセグメントのリストを取りだす。

【0179】

セグメントのリストが空であるか、またはすべてのセグメントが既にマークされていれば（テスト1704）、処理は終了し、サーバはクライアントの要求に対して応答を送信し始める（上記のステップ102）。

【0180】

そうでなければ、リストの第1のセグメントは、ステップ103（PUSH_PROMISE）および104（プロミスされたセグメント）の間にプッシュされるのと同様に、ステップ1705においてマークされる。20

【0181】

プッシュポリシーを宣言する、これらのMPDに基づく例において、1つのプッシュポリシーは、Push Policy要素を用いて定義される（図16の1600を参照）。

【0182】

記述ファイルが、複数のメディアデータ属性レベル（すなわち、以上で定義されたPeriod、Adaptation Set、Representation要素）を用いて、メディアデータを記述することが、ここで想起される。

【0183】

上記のものに対するわずかな変形として、様々な共有されたプッシュポリシーは、様々なそれぞれのレベルの記述ファイルにおいて定義されてもよい。これは、メディアストリームのコンテンツに対してプッシュ戦略を適応するために、関係するレベル（Adaptation Set、Representation、Period）に依存する様々なプッシュポリシーを定義することができる、ということである。30

【0184】

これは図23を通じて示され、ここで、Representationレベルにおいて、プッシュポリシーが、所望のレベルにて、例えば「Supplemental Property」記述子を用いて定義される。

【0185】

<MPD>レベルごとにプッシュポリシーを用いることは、メディアにわたって定数および同じプッシュ戦略を有することを可能にする。40

【0186】

<Period>レベルごとにプッシュポリシーを用いることは、時間に沿って変化できるプッシュ戦略を有することを可能にする。

【0187】

<Adaptation Set>レベルごとにプッシュポリシーを用いることは、メディアに適したプッシュ戦略を有することを可能にする。

【0188】

<Representation>レベルごとにプッシュポリシーを用いることは、メデ50

イア特性（帯域幅…）に適することができるプッシュ戦略を有することを可能にする。

【0189】

図23の例において、Representationレベルにおいて指定されるプッシュポリシーは、プッシュデータとともにあまりにも多くの帯域幅を用いないようするために、高いビットレートのビデオ（2301）よりも低ビットレートビデオセグメント（2300）のために、よりさらにセグメントをプッシュするように構成される。

【0190】

プッシュ属性の構文に関しての上記の説明も、また、このわずかな変形に適用されてもよい、ということに留意されたい。特に、プッシュポリシーは、新たな要素（図16のように）としてマニフェストにおいて、または（図23のように）新たなscheme1d Uriにより既存の記述子を用いて、または（表現されていない）新たな記述子として、またはMPDスキーマ若しくはMPDスキーマの拡張ポイントに適合するいずれかの手段として、信号で送ることができる。

【0191】

MPDは、また、各自が一意の識別子を有する、代替的なPUSHポリシーのリストを含むこともできる（以下のリストに関するさらなる説明を参照）。

【0192】

他の代替の実施形態において、プッシュポリシーは、相補的なRepresentationのためのセグメントが系統的にプッシュされる、ということを、例えば以下の構文を用いて定義してもよい。

【0193】

<push_policy Segments='+complementary'>または、DASH記述子を用いる場合は、value='complementary'層状ビデオの場合には、これは、要求されたビデオセグメントに対して、相補的なRepresentationとして宣言されるすべてのRepresentationから同時に各セグメントも（通常は、異なるRepresentation間の依存性を信号で送るMPD内のdependency1d属性を通じて）プッシュされるだろう、ということを意味する。

【0194】

別のプッシュポリシーは、また、@association1d属性またはrole='supplementary'により信号で送られる、関連するRepresentationからのセグメントのプッシュにある。

【0195】

完全にサーバで駆動されるストリーミングの場合には、プッシュポリシーは、サーバ挙動が「積極的」（若しくは「楽観的」）または「保守的」でなければならないか否かの情報、すなわち、それぞれ高品質のセグメントをプッシュしようとする、または同じ品質レベル（帯域幅を維持する）でプッシュしようとするか否かの情報を提供することができるであろう。

【0196】

他の実施形態において、プッシュポリシーは、専用のHTTPヘッダとして見なされる「プッシュポリシー」ヘッダ内で送信される。すなわち、共有されるプッシュポリシーを記述するプッシュポリシー情報は、サーバ装置からクライアント装置に対して送信されたHTTPフレームのヘッダ内に埋め込まれる。

【0197】

これらの実施形態は、上記のように、それらがもはやMPDファイルの送信に依存せず、クライアントおよびサーバがHTTP/2プロトコルを用いて交換するので、プッシュポリシーを経時的に変化させることを可能にする。

【0198】

図18は、HTTP「プッシュポリシー」ヘッダ（ヘッダ名「プッシュポリシー」は、単なる例である）内で送信されるプッシュポリシーでの、サーバとクライアントとの間の通

10

20

30

40

50

信の例である。

【0199】

プッシュポリシーへッダは、プッシュ属性のリスト（プッシュされるデータ上の制約を各々定義する）を含む。特に、先に記述された Push Policy の構文は、HTTP へッダ構文に書き換えられてもよい。

【0200】

図 18 aにおいて、クライアント（矢印 1800）からの MPD 要求に応答してサーバは、プッシュポリシーを共有するために、送信された MPD に伴う HTTP へッダ内のプッシュポリシーを送信する（ステップ 1801）。

【0201】

例えば、プッシュポリシーは、要求されたセグメントを追従するセグメントがプッシュされることになる、ということを指定する。結果として、クライアントがセグメントデータ 1.1 を要求する（矢印 1802）と、サーバは、セグメントデータ 2.1 のための PUSH_PROMISE と（矢印 1803）、そしてセグメントデータ 1.1（矢印 1804）のデータを送信する。

【0202】

後続のセグメント要求のためにどのデータが送信されようとするのかを定義するために、任意の構文：MPD 特有のものまたは DOM ツリーノード トラバースに基づいたより抽象的なものを用いることができる。

【0203】

動的に共有されたプッシュポリシーに専用の特定の実施形態において、クライアントは、特定のプッシュポリシーを要求してもよい。すなわち、例えば現在共有されたプッシュポリシーがそのニーズに適していないければ、共有されたプッシュポリシーを更新してもよいし、または改善してもよい。

【0204】

それは、クライアント装置が HTTP フレームのヘッダ内に埋め込まれたプッシュポリシー更新情報をサーバ装置に対して送信する、ということを意味する。それに対応して、サーバ装置は、HTTP フレームのヘッダ内に埋め込まれたプッシュポリシー更新情報をクライアント装置から受信する。サーバ装置は、したがって、クライアント装置（例えば次の要求のための）によって要求された他のメディアデータから非要求メディアデータを決定する前に、共有されたプッシュポリシーをそれに応じて更新してもよい。

【0205】

実施形態において、クライアントからのプッシュポリシー要求は、「プッシュポリシー要求」という名の（ここでの名称は単なる例である）HTTP へッダまたは要求内で伝達される。

【0206】

図 18 b は、クライアントが新たなプッシュポリシーを要求した場合のクライアント・サーバ間の例示的な交換を示す。

【0207】

交換の始まりは、図 18 a と同じである。

【0208】

セグメントデータ 2.1 を受信した後、クライアントは、例えば利用可能な帯域幅が、セグメント要求に応答してさらなるセグメントをサーバにプッシュさせるのに十分に安定しているので、現在のプッシュポリシーが修正されるべきである、ということを識別する。

【0209】

結果として、クライアントは、ステップ 1805 において、各々の新たな要求毎に、さらなるセグメント（1 の代わりに 3）をプッシュするようにサーバに依頼するプッシュポリシー要求を送信する。

【0210】

サーバは、ステップ 1806 において、OK 200 によりこのプッシュポリシー要求に肯

10

20

30

40

50

定的に回答する。この肯定的な回答は、同じクライアントからいずれかの新たな要求に対するプッシュポリシー要求内に記述された新たなプッシュポリシーをサーバが用いることになる、ということを意味する。

【0211】

サーバがそのプッシュポリシーを変更したくないならば、それはプッシュポリシー要求が拒否されている、ということをクライアントに通知するためのエラーコード回答を返す。

【0212】

次に、クライアントが、ステップ1807において、次のセグメントデータ3.1を要求すると、サーバは、ステップ1808において、次の3つのセグメントデータ4.1、データ5.1、およびデータ6.1に対してPUSH_PROMISEにより回答する。

10

【0213】

図21は、プッシュポリシーの共有のためにHTTP要求を用いる場合のサーバ側における処理のステップを、フローチャート用いて示し、その一方で、図22は、プッシュポリシーを共有するためにHTTP要求を用いる場合のクライアント側における処理のステップを、フローチャートを用いて示す。

【0214】

図14の処理との比較において、サーバは、クライアントからのプッシュポリシー要求を取り扱い、かつまた初期のプッシュポリシーを送信してそれを更新する、新たな処理ステップ(2100～2105)を含む。

【0215】

サーバによって受信された要求がクライアントからのプッシュポリシー要求であるならば(テスト2100)、サーバは、まず、クライアントによって提案されたデータプッシュの制約を抽出するために、ステップ2101において、プッシュポリシー要求を解析する。

20

【0216】

このステップの間、サーバは、クライアントによって要求されたプッシュポリシーに従うことを決定してもよい。そのような場合、サーバは、その内部のプッシュポリシーを更新し(ステップ2102)、提案されたプッシュポリシーを検証するために、ステップ2103において、クライアントに対してOK200の応答を送信する。

【0217】

そうでなければ、サーバがプッシュポリシーを廃棄すると(例えば提案されたポリシーがリソースの観点からコストのかかりすぎる、または適用することができないので)、ステップ2102ではサーバにおいて内部プッシュポリシーを修正しない。また、エラーコードが、ステップ2103において、クライアントに対して送信される。

30

【0218】

具体的な実施形態によれば、サーバは、クライアントの要求とは無関係に、そのプッシュポリシーをさらに更新してもよい。そのような場合、サーバは、ステップ1402の間に、プッシュポリシーを決定し、その特性(例えばクライアントおよびネットワークの特性によって実行された要求の解析による)を変更することを決定しても良いし、または、決定されたプッシュポリシーが現在のものとは異なるということを理解しても良い。このような状況において、サーバは、後者がそれにまだ気づいていなければ(テスト2104)、クライアントと新たなプッシュポリシーを共有しなければならず、その場合には、新たなプッシュポリシーは、ステップ2105において、HTTPヘッダ内で送信される。

40

【0219】

クライアント側の対応する処理は、図22を参照して説明される。サーバ処理に関して、新たな処理ステップ(2200～2204)は、図14の処理との比較において、プッシュポリシーメッセージを処理し、且つプッシュポリシー要求を実行するために追加される。

【0220】

ステップ1400において、現在の共有されたプッシュポリシー(すなわちサーバのプ

50

シュポリシー)を決定した後に、クライアントは、例えば、メディアストリームのセグメントを取りだすために送信するHTTP要求の数を低減するために、新たなプッシュポリシーを要望してもよい。したがって、新たなプッシュポリシーがクライアントによって要求されると(テスト2200)、クライアントは、ステップ2201において、先に記述されたような「プッシュポリシー要求」によりHTTP要求を送信する。

【0221】

この要求に対する応答は、OK200の応答またはそうでなければエラーコードを返すことによって、サーバが要求を有効にするか否かをクライアントがチェックするステップ2204において処理される。

【0222】

サーバがOK200の応答を返せば、ステップ1400において、決定された現在のプッシュポリシーが、要求されたポリシーと置き換えられる。そうでなければ、それはそのまま変更されない。

【0223】

図14の処理に加えて、クライアントがサーバから新たなプッシュポリシーでフレームを受信すると(テスト2202)、プッシュポリシーは、ステップ1400の次のオカラムスで取りださるために解析され、メモリ内に保存される(ステップ2203)。

【0224】

プッシュポリシー要求がまた他のデータ(例えばメディアデータ)も含むフレーム内にあると、他のデータが、ステップ109~111~113~115を通じて処理される、ということに留意すべきである。

【0225】

上記のHTTPに基づいた例は、適用されるプッシュポリシーを完全に定義するためにHTTP要求を用いるが、1つの具体的な実施形態は、クライアントとサーバ側の両方において1セットの予め定義された同じプッシュポリシーを有し、各々が一意の識別子を有することを定義することに依存してもよい。この場合、HTTP要求は、セットの中から用いられるプッシュポリシーの識別子を指定するためにのみ用いられる。この具体的な実施形態は、HTTP要求のサイズを減少させる。

【0226】

1つの実施形態において、プッシュポリシー要求は、サーバリソースの1つを要求するために用いられるHTTP要求の1つの付加的なHEADERとして送信され、通常は、プッシュポリシー要求は、MPDファイルに対するGET要求における「受理プッシュポリシー」HTTPヘッダ内で送信される。

【0227】

別の実施形態において、クライアントは、クライアントによって支持される(または必要とされる)プッシュポリシーのリストを示すために1つのHTTP要求においていくつかの「Accept-Push-Policy」を指定する。HTTP要求に応答して、サーバは、提案されたリスト内のプッシュポリシーの1つを選択し、そして、HTTP応答内のプッシュポリシーを指定してもよいし、または、どれもサポートされなければ、新たなプッシュポリシーによって応答してもよい。

【0228】

さらに別の実施形態において、プッシュポリシー要求は、サーバによって認識されるあらゆるリソースの独立した専用のHTTP要求において送信される。例えば、GET(またはPOST)要求は、ウェブページのリソースのどれにも対応しないURL、例えばhttp://server/push_policyと、さらに少なくとも1つのAccept-Push-Policyヘッダとにより形成される。

【0229】

さらに別の具体的な実施形態において、代替的なプッシュポリシーのセットは、各々が一意の識別子を有する、サーバとクライアントとの間で交換されたMPDファイル内で定義されてもよい。プッシュポリシーの1つは、サーバによって選択されるデフォルトプッシ

10

20

30

40

50

ュポリシーとしてマークされてもよい。クライアントは、デフォルトプッシュポリシーの交換に用いられるプッシュポリシーの識別子を含む新たなプッシュポリシー要求を送信することによって、どのプッシュポリシーが用いられるべきであるのかを指定してもよい。

【0230】

1つの実施形態において、特定のプッシュポリシーは、高速起動のためにMPDドキュメントに対する要求直後に、どのセグメントがプッシュされるのかを示すために定義される。

【0231】

ハイブリッドアプローチにおいて、共有されるプッシュポリシーを記述するプッシュポリシー情報は、第1のプッシュポリシー部および第2のプッシュポリシー部によって定義され、第1のプッシュポリシー部は、記述ファイル(MPD)内に挿入され、第2のプッシュポリシー部は、サーバ装置からクライアント装置に対して送信されるHTTPフレームのヘッダ内に埋め込まれる。10

【0232】

例えば、MPDは、プッシュポリシーHTTP要求のおかげでサーバによってその後定義される(またはさらにオーバーロードされる)テンプレート引数によりプッシュポリシーを定義してもよい。例として、MPDファイル内に定義されたプッシュポリシーは、<PushPolicy Segment1dx="parameter7">であってもよく、変数「パラメータ」の値は、プッシュポリシーHTTP要求において定義されてもよい。この例において、第2のプッシュポリシー部は、第1のプッシュポリシー部内に定義された1つ以上の関連変数に対する1つ以上の値(のみ)を有する。20

【0233】

上記のプッシュポリシー識別子ベースのアプローチを用いて、記述ファイルは、複数の候補プッシュポリシーの記述を含んでもよく、第2のプッシュポリシー部は、それにより前記複数からの候補プッシュポリシーの識別子を備えてもよく、識別された候補プッシュポリシーは、それによって第1のプッシュポリシー部を形成する。

【0234】

クライアントに対してプッシュポリシーを宣言するための別の実施形態において、プッシュポリシーは、プッシュデータが選択されることになるリプレゼンテーションを示すMPD内に定義されたRole記述子に依存する。通常は、プッシュポリシーは、プッシュ戦略が「代替」または「追加」のロール値をもつRepresentation内のセグメントを用いることになる、ということを指定してもよい。30

【0235】

別の実施形態において、例えばストリーミングマニフェスト、HTMLページなどのリソースのドキュメントは、GET要求が受信された後にプッシュされるリソースを決定するために閲覧されるプライオリティツリーに変換される。プライオリティツリー内部のナビゲーションは、XPath要求のおかげで実行することができる。このアプローチにおいて、プッシュポリシー情報は、第2の非要求メディアデータを識別するリソースのドキュメントのツリー表現上で評価される、XPath式を含む。40

【0236】

例えば、ストリーミングマニフェストにおいて、「following[name() = "SegmentURL"] [2]」のXPath式は、プッシュされるセグメントとして、GET要求内のクライアントによって要求されたセグメントに続く2つのセグメントを選択するために用いることができる。また、チャプターを切り替える使用事例に関して、「((following[name() = "Period"] // SegmentURL) [2])」のXPath式は、各チャプターの最初の2つのセグメントを予めローディングするために後続のPeriodの最初の2つのセグメントを選択することを可能にする。例えば、クライアントが図24のMPDファイル内のセグメント2401を要求すると、後続のPeriodのセグメント2402および2403も、プッシュされたデ

ータとしてサーバによって送信される。

【0237】

さらに、高度なプッシュポリシールールのために書く X P a t h 式を単純化するために、例えば X S L T 命令を用いて、プライオリティツリーをまず再順序付けることができるかもしだい。X S L T 命令は、プッシュポリシーを適用する前にツリーを再編成することを可能にする。X P a t h 式は、好ましくは、クライアントに対して、例えば 1 つの H T T P ヘッダ内で送信され、X S L T スタイルシートは、ウェブページ内で定義される。これは、例えばドキュメント内の宣言するすべての画像、D O M ツリーの同じレベルにおける連続するノードのようなすべての C S S リソース、をグループ化するために、特に H T M L ドキュメントに適用する。

10

【0238】

例えば、図 2 5 のツリー 2 5 0 1 は異なるタイプの異なるリソースも持つ H T T P 頁を表し、ハッシュされたノード (2 5 1 1 ~ 2 5 1 4) は、画像リソースに対応し、単色のノード (2 5 2 1 - 2 5 2 4) は、スクリプトリソース (C S S または J a v a (登録商標) s c r i p t) である。ツリー 2 5 0 2 は、タイプによってリソースをグループ化する X S L T 変換結果の例である (2 5 3 0 における画像および 2 5 4 0 におけるスクリプトリソース)。簡単な X P a t h 式は、したがって、一旦この所定のタイプの第 1 のリソースが要求されれば、所定のタイプのためのいくつかのリソースがプッシュされることになる、ということを示すために、定義することができるかもしだい。

20

【0239】

以上に記述されたすべての実施形態において、いくつかのセグメントがプッシュされることをプッシュポリシーが要求するならば、各々のクライアントに対していくつかの P U S H _ P R O M I S E でのサーバ返信を要求する、という可能性が非常に高い。

【0240】

例えば、図 1 9 の M P D 1 9 0 0 は、要求されたセグメントに追従する 3 つのセグメントがプッシュされることになるということを示すプッシュポリシーを有する (< P u s h P o l i c y > 要素を参照)。従って、クライアントが 0 ~ 9 9 9 に等しいバイト単位の範囲でメディア 1 9 0 1 に対する G E T 要求によりイニシャライゼーションセグメントを要求すれば、サーバは、ステップ 1 0 3 の間に、3 つの P U S H _ P R O M I S E メッセージに 1 9 0 2 を送信することになる。

30

【0241】

1 つの実施形態において、識別された第 2 のメディアデータの各々がアナウンスマッセージ (すなわち P U S H _ P R O M I S E) を要求する複数のメディアセグメントを備える場合、対応する複数のアナウンスマッセージは、クライアント装置に対して送信される单一のアナウンスマッセージにマージされてもよい。

【0242】

図 2 0 に示されるように、この状況を実現するために、サーバにおける処理は、好ましくは、図 1 4 の一般的な処理と比較して、ステップ 1 0 3 においてプッシュプロミスを送信する直前に前処理ステップ 2 0 0 0 を含む。前処理ステップは、アナウンスマッセージの上記のマージを実行しようと努める。

40

【0243】

プッシュプロミスが 1 9 0 2 のようにバイト単位の範囲の要求を含むと、プッシュプロミスのリスト 1 9 0 2 は、連続するバイト単位の範囲のアドレスを含むプッシュプロミス 1 9 0 3 の減少されたセットを生成するために閲覧される。次に、プッシュプロミス 1 9 0 2 の各セットは、プッシュプロミスセット内のバイト単位の範囲の連結に等しい連続したバイト単位の範囲でプッシュプロミスの減少されたセットと、または例えば 1 9 0 5 の不連続のバイト単位の範囲のリストでの单一のプッシュプロミスと置き換えられる。

【0244】

例えば、3 つのプッシュプロミス 1 9 0 2 は、図 1 9 に示される单一のプッシュプロミス 1 9 0 3 と置き換えられる。

50

【0245】

プッシュプロミスをマージするこのアプローチは、クライアントがより簡単な方法で、およびより低い帯域幅および処理コストで、プッシュデータの送信をキャンセルすることを可能にする。これは、マージされていない各々のプッシュプロミスの幾つかのストリームを閉じる代わりに、クライアントが単一のプッシュプロミスのための単一のストリームを閉じるだけで良いからである。

【0246】

代替案において、プッシュプロミスが別々なバイト単位の範囲の間隔を有したとしても、プッシュプロミスは、すべて、バイト単位の範囲のリストと置き換えられてもよい（ここで連続するバイト単位の範囲の間隔は連鎖される）。 10

【0247】

さらに、プッシュプロミスがバイト単位の範囲ではなく、異なる Segment URL 値を含む場合、プッシュプロミスも、また、以下の通り単一のプッシュプロミスマッセージを生成するために連鎖されてもよく、生成されたプッシュプロミスマッセージの方法は MGET (複数の GET のための) として定義される。また、パスフィールドは、1904 に表現されるようなセグメント URL のリストである。先の実施形態と同様に、クライアントは、すべてのセグメントのプッシュをキャンセルするために生成されたプッシュプロミスに対応する単一のストリームを閉じなければならない。

【0248】

クライアントが各プッシュされたセグメントを解析し識別することができるることを保証するために、サーバは、その後に送信されるデータ内の各々のセグメントの端部において END_SEGMENT フラグを含んでもよい、ということに留意されたい。 20

【0249】

さらに、HTTP / 2 の SETTINGS フレームは、プッシュプロミスのグループ化がストリーミングセッションのために許可されるか否かを示すことを可能にする、新たな SETTINGS_ENABLE_GROUP_PUSH_PROMISE パラメータを含むために拡張される。

【0250】

本発明の実施形態は、1回または複数回のラウンドトリップを回避することができるので、DASH 高速起動を有することを可能することができる。ここで、図 26 ~ 図 28 を参照して、本発明のこの態様を、説明する。 30

【0251】

DASH 高速起動機能は、図 7 ~ 図 12 および図 14 ~ 図 25 のすべてまたは一部を参照して、上記された何れの通信アプローチとともに用いられてもよい。

【0252】

図 26 は、DASH 高速起動を取得するために、本発明の教示によるサーバによって、およびクライアント装置によって、それぞれ実施される例示的な方法を示す。

【0253】

まさに記載された標準プロセスに関して、第 1 のステップは、クライアントが記述ファイル、ここでは MPD ファイル、を要求するためにある（ステップ 2650）。そして、クライアントは、サーバの応答を待つ（ステップ 2651）。 40

【0254】

その間に、サーバは、以下に説明されるように、特にクライアントがより高速に開始するのを支援するイニシャライゼーションデータを識別するために（ステップ 2653）MPD ファイルを解析する（ステップ 2652）。ステップ 2653 の例示的な実施形態は、図 27 を参照して、以下に説明する。

【0255】

一旦イニシャライゼーションデータがサーバによって識別されれば、クライアントの要求を待たずに、イニシャライゼーションデータをプッシュするその意図を示すために、ステップ 2654 において、クライアントに対して PUSH_PROMISE フレームを送信

10

20

30

40

50

する。

【0256】

場合によっては、それは、また、クライアントが関係するリソース、すなわち、`:scheme`、`:host`、および`:path`などの関係する初期のメディアデータ、を識別することを可能にするヘッダ部を含む別の`PUSH_PROMISE`フレームを送信することによって、初期のメディアデータ（ステップ2656）をプッシュすることになる、ということをさらに示す。

【0257】

イニシャライゼーションデータのための`PUSH_PROMISE`フレーム、および初期のメディアデータのための`PUSH_PROMISE`フレームの場合、他のヘッダ部も、また、プッシュすることを決定したデータにおいてサーバがどれほど確信しているのかを示すために、サーバによって追加され、本実施形態において、`confidence_level`パラメータは`PUSH_PROMISE`フレームに関連付けられる（すなわち、ヘッダ内に含まれる）。図27を参照して、`confidence_level`パラメータの決定を以下に説明する。サーバは、また、プッシュするつもりのセグメントを明白に示すために特定の`DASH`ヘッダを挿入することができる。

10

【0258】

プッシュされるべきイニシャライゼーションデータおよび第1のメディアデータのためにクライアントが要求することのリスクを最小化するために、`PUSH_PROMISE`フレームは、応答において、いずれかのコンテンツに先立って送信されるべきである。すなわち、ステップ2654およびステップ2656は、サーバからクライアント装置までMPDファイルを送信するステップ2655の前に行われるべきである。

20

【0259】

したがって、`PUSH_PROMISE`フレームがクライアント装置に対して送信されると、サーバは、ステップ2655において、クライアント装置に対してMPDファイルを送信する。

【0260】

その間にサーバがクライアント装置から`CANCEL`または`ERROR`のメッセージを受信していないければ、イニシャライゼーションデータ（ステップ2657）および第1のメディアデータ（ステップ2658）をプッシュし始める。

30

【0261】

`PUSH_PROMISE`フレームおよびサーバからクライアント装置へのデータのプッシュは、例えばドキュメント「ハイパーテキスト転送プロトコルバージョン2.0（`HyperText Transfer Protocol version 2.0`）、`draft-ietf-httpbis-http2-latesf`、`HTTPbis`作業部会、インターネットドラフト、2013年6月24日（例えば`http://http2.github.io/http2-spec/`において利用可能）内に記述されるような、例えば`HTTP2.0`のフレームにおいて展開される、対応する機能にしたがって実行される。

40

【0262】

クライアント装置が受信すると、イニシャライゼーションデータは、復号器を設定する（ステップ2659）ためにクライアントによって用いることができ、画面停止なしで十分なデータ量が復号化およびレンダリング（例えば、表示）のために利用可能になるまで、第1のメディアデータは、バッファリングされる（ステップ2660）。

【0263】

クライアントがMPDファイルを完全に受信し、完全にデータをバッファリングすると（ステップ2661）、それを解析して（ステップ2662）復号化し表示し始め（ステップ2663）。そうでない場合は、クライアント装置が、さらなるセグメントが送信されることを、サーバによって送信された`PUSH_PROMISE`フレーム（ステップ2656を参照）から認識するならば、ステップ2664において、サーバからの第1のメデ

50

イアデータのプッシュの完了を待つ。このアイドルステップ 2664 の間、クライアント装置は、既に上で説明されたように、標準的なクライアント制御の DASH (ステップ 2665)において発行される後続のセグメントのために次の要求を準備してもよい。対応する PUSH_PROMISE フレーム (上記ステップ 2656 を参照) 内のプッシュされる (またはプッシュされている) 初期のメディアデータの受信情報をクライアント装置が有するので、これは可能であり、したがって、サーバによってプッシュされると意図された最後の時間セグメントを直後に続く時間セグメントのための要求を準備することができる。

【0264】

クライアント装置は、MPD を全て受信すると、この初期のメディアデータがバッファを満たすか否かをチェックするために、そうでなければ、ステップ 2661 の前の標準的なクライアント制御の DASH 処理にしたがって後続のメディアデータ (例えば初期のメディアデータによって表現される時間セグメントに続く時間セグメントに対応するメディアデータ) に対する要求を送信するために、ステップ 2656 において受信された初期のメディアデータの情報を用いてもよい (プッシュされた初期のメディアデータがバッファを満たすケースを示す図 26 に示されるものとは逆に)。これは、クライアントが、プッシュする大量の第 1 のメディアデータの上のサーバから不良な推定を訂正することを可能にする。

10

【0265】

この処理は、ストリーミングクライアントが、標準的マニフェストベースのストリーミングよりも早くメディアを表示し始めることを可能にする。実際は、ネットワーク上の HTTP ラウンドトリップの数がイニシャライゼーションデータおよび / または初期のメディアデータを得るために減少されるので、起動遅延は減少する。

20

【0266】

但し、この処理は、以下の理由から、現在の DASH 規格に準拠したままである。

【0267】

- ・MPD ファイルの修正がない。その送信は軽量で高速のままである。

【0268】

・標準 DASH クライアントの挙動は不变のままである (すなわち、本発明の教示から恩恵を受けない)。このようなクライアント装置は、認識されない HTTP ヘッダを無視するだろう。そして、プッシュ機能を受け入れなければ、さらなる要求 / 応答を単純に実行しなければならないだろう。したがって、表示を開始するために多くの時間を費やすだろう。

30

【0269】

図 27 は、クライアント装置からのマニフェスト (または記述ファイル) に対する要求に追従するサーバ側において実施される例示的な方法説明する。

【0270】

この方法は、クライアントがメディアプレゼンテーションの表示を急速に開始することができるよう、予めプッシュする最も関連する初期データを識別しようと努める。

【0271】

ステップ 2700 において、マニフェストに対する要求が受信される。そして、サーバは、クライアント装置が要求内にいくつかのプレファレンスを挿入したか否かを、ステップ 2701 において、チェックする。例えばメディアプレゼンテーションのための送信速度およびオーディオストリームのための好適な言語を表現するためなど、これは、以下の専用の HTTP ヘッダなどを介して行われてもよい。

40

【0272】

```
GET http://myserver.com/presentation/press1.mpd\r\n
Preferred-MediaRange: bw=2000:lang=FR\r\n
\r\n
```

50

要求がプレファレンスを含むならば（テスト2701で真）、サーバは、クライアントのプレファレンスを解析し（ステップ2703）、そのconfidence_levelパラメータを値「高」に設定する（ステップ2704）。

【0273】

表示が要求において提供されないならば（テスト2701で偽）、サーバは、このクライアントのサービス利用情報（ログ）（すなわちユーザまたはクライアント装置とサーバとの間の従来の交換に基づく統計値または使用データ）またはUser-Agentヘッダからの情報を既に示しているか否かを、ステップ2702において。チェックする。実際は、User-Agentヘッダは、RFC2616（例えばhttp://www.ietf.org/rfc/rfc2616.txtを参照）内のHTTPヘッダとして定義され、例えばオペレーティングシステム、ブラウザタイプ、アプリケーション名などの情報を交換するアプリケーションのための手段を提供する。例えば、DASHサーバは、サービスを用いることができるようになる前にクライアントに対する認証スキームを有してもよく、変形物において、それは、サービスに対してアクセスする前にロギングするユーザであり得る。このような手段により、サーバは、メディアパラメータを、接続されたユーザまたは装置に対してリンクすることができる。

10

【0274】

前の利用情報（ログ）が、関係するクライアント装置またはユーザが利用可能であると（テスト2702で真）、ステップ2705においてログを解析することによって、サーバは、所定のクライアントまたはユーザにとっての最も頻度の高い利用方法を推定することができる。例えば、ユーザまたはクライアント装置がHD（高解像度）のフランス語によるオーディオストリームおよびビデオストリームを常に選択する、ということを推定することができる。さらに、サーバは、これがオープンTCP接続における第1の要求であるか否か（サービスに対して接続され、且つ第2のメディアプレゼンテーションを要求するクライアント）を知ることができる。この場合、帯域幅推定は、さらに正確で且つさらに確実になり得るし、TCP輻輳ウィンドウは、第1の要求に対するものよりも大きくなるであろう。これは、適したRepresentationの観点から、サーバによって行われる選択に影響を与えることができる。

20

【0275】

DASH品質メトリックを登録することによって、サーバは、そのログ内にユーザ／クライアントが通常実行する様々なリプレゼンテーションの中の変更を持つことができる。このことから、サーバは、通常の挙動を、「積極的（aggressive）」または変更の頻度に依存する定数（constant depending on the frequency of changes）の間で決定する（いかなる基準：帯域幅、解像度、フレームレート等でも、変更によって、我々は他のRepresentationに切り換えることを意味する。）。積極的なクライアント（aggressive client）は、そのコンテキストが変化すると、異なるリプレゼンテーションに自動的に切り替えるDASHクライアントである。例として、帯域幅またはバッファ占有期間を監視すると、積極的なクライアントは、現在のRepresentationと比較して、新たなRepresentationがクライアントのコンテキストに近い特性を有すると直ちに、異なる帯域幅でRepresentationを要求することになる。反対に、一定のクライアントは、安定した品質および表示速度を維持するために、頻繁なRepresentationの切り換えを回避しようとするだろう。ユーザ／クライアント装置挙動が適応の観点からいくぶん積極的であると、サーバは、それがストリーミングを開始するために初期のリプレゼンテーションとして何を選択よりも、クライアントは、続く最初の数秒または数分間のストリーミングにおいて適応しようとする、ということを知る。

30

【0276】

プレファレンスがログから推定されると、サーバは、ステップ2706において、そのconfidence_levelパラメータを値「中」に設定する。実際は、この情報は、クライアントそれ自体によって信号で送る明示的なプレファレンスよりも関連性が低い

40

50

かもしれない（テスト2701で真）。

【0277】

利用可能なログ情報がない場合（テスト2702で偽）、サーバは、その時、ステップ2707において、そのconfidence_levelパラメータを最低値「低」にする。これは、決定するべき演繹的な情報がないので、サーバが、プッシュする情報上でベストの推測を実行していることを示す。この場合のさらなる処理を、以下に説明する記載する（ステップ2711を参照）。

【0278】

このconfidence_levelパラメータ計算と並行して、サーバは、ステップ2708において、マニフェストを解析してもよい。マニフェストが頻繁に変更しがちではない（ライブサービスとは逆に、特にオンデマンドのサービスに対して）場合において、マニフェストの解析は、ルックアップテーブル内の様々なRepresentationの記述を登録することによって、今回限りで、オフラインで実行することができる。このルックアップテーブルは、また、クライアントのログをメディアプレゼンテーションのいくつかの部分に対してリンクするために、サーバによって用いられてもよい。これによって、いくつかのクライアントのプレファレンスを推定するために、より高速のログ処理（上記のステップ2705を参照）を行うことが出来る。

10

【0279】

マニフェストの解析（ステップ2708）は、ストリーミングを開始するために初期のRepresentation（すなわち初期のメディアデータ）として適したRepresentationを選択するときに、サーバに対して情報を提供する。

20

【0280】

（要求においてプレファレンスをそれぞれ取得する、または、先の交換からの使用データに基づく）両方のステップ2703および2705は、クライアント装置／ユーザからプレファレンスまたは利用をMPD属性と一致する具体的なパラメータ内に変換することにある。例えば、それは、帯域幅、ビデオの幅および高さ、使用中のコーデックの種類、字幕のための言語、またはオーディオストリームであり得る。そして、これらのパラメータのために取得された値から、サーバは、クライアントに対してプッシュするべき最も都合のよいRepresentationを、ステップ2709において識別するために、マニフェスト内の値と比較する。

30

【0281】

このステップ2709は、通常は、クライアント装置がDASHのような動的および適応型ストリーミングプロトコルで継続的に実行するものである、ということに留意されたい。ここで、同じステップは、MPD構文解析手段により、ストリーミングセッションの先頭において、サーバによって実行される。

【0282】

適したRepresentationを2709において推定することができない場合、テスト2710は偽であり、サーバは、そのconfidence_levelパラメータを（前述のステップ2707において）「低」値にする。

40

【0283】

confidence_valueパラメータが「低」値を有する場合（プレファレンスを決定することができなかったので、または適したRepresentationがプレファレンスに基づいて見つからないので）、サーバは、最も簡単なRepresentationを選択することをステップ2711において判断する。ビデオに関して、例えば、最も簡単なRepresentationは、最低の空間解像度によるRepresentationであってもよいし、最低の帯域幅用に設計されてもよい。

【0284】

（図27には表現されない）可能な補完的な特徴によれば、コーデック上にアンビギュイティがない場合（すなわち、ビデオRepresentationは、すべてコーデック

50

属性と同じ値を有し、すなわち、同じコーデック、例えばHEVCは、ビデオRepresentationをすべて符号化するために用いられている)、confidence_levelパラメータは、値「中」に引き上げられてもよい。

【0285】

ステップ2711の次のステップは、または適したRepresentationが見出された場合(テスト2710で真)は、イニシャライゼーションデータを識別することにある(ステップ2712)。実際に、DASHマニフェスト(または記述ファイル)において、イニシャライゼーション情報は、異なる方法で、信号で送ることができ、それは、イニシャライゼーションデータに対して直接的なURLを提供するSegmentBase、SegmentListまたはSegmentTemplate要素のイニシャライゼーション要素内に明示的に加えることができる。10

【0286】

この場合、このURLは、(変数:scheme、:host、:path、および最終的に:Rangeを指定することによって)プッシュされることをプロミスされたリソースをクライアントが識別することを可能にするPUSH_PROMISEフレームのヘッダ部内に加えられる(図26を参照しながら上記されたステップ2654を参照)。

【0287】

イニシャライゼーションデータが明示的に記述されない場合、これは、メディアセグメントがセルフイニシャライズされる、ということを意味する。そのような場合、サーバは、セグメントの最初(例えばmp4形式のセグメントのためのセグメントインデックス情報ボックス)を解析しなければならない。この解析に基づいて、PUSH_PROMISEフレーム内にヘッダとして加えられる対応するURLを適切なバイト単位の範囲で構築することができる。20

【0288】

一旦識別されれば、イニシャライゼーションデータのためのPUSH_PROMISEフレームは、クライアントに対して直ちに送信され(図26におけるステップ2654に対応するステップ2713)、その直後にイニシャライゼーションデータのプッシュにより送信される(図26におけるステップ2657に対応するステップ2717a)。イニシャライゼーションデータが受信されると、クライアントは、その後、そのメディア復号器をイニシャライズすることができる(ステップ2717b)。30

【0289】

随意に、PUSH_PROMISEフレームを処理するときにクライアント装置によってセグメント信号送信およびその後の識別を向上させるために(以下に記載のステップ2806を参照)、サーバは、ステップ2713において、プッシュされたデータの性質と、イニシャライゼーションまたはメディア若しくは両方(自分でイニシャライズするセグメントの場合)と、URLテンプレートのパラメータまたは図5bのMPDリプレゼンテーションツリーにおけるパスのようなセグメントの表示(例えば:P2A\$21R211S1、すなわち、識別子に続く要素型の連結)を示すことができる。これは、MPDを受信するようにクライアント装置に要求する、ということに留意されたい。そして、サーバは、クライアント装置によってMPD受信の後に処理されることになると考えるPUSH_PROMISEメッセージ内にのみ、この特定情報を追加することを決定することができる。MPDの受信および解析の前にPUSH_PROMISEを受け入れるかまたは受け入れないかをクライアント装置で決定することを支援するために、サーバは、MPD内のセグメントパスの代わりに、プッシュされたセグメントの定性的情報(例えばそれがベース層またはエンハンスマント層からのセグメントであるか否かなど)を示すことができ、また、別の例によれば、サーバは、それらの値で選択されたRepresentationの属性をヘッダ内に配置することができる。40

【0290】

(図27には表現されない)可能な実施形態によれば、ステップ2708においてマニフェストの解析が、イニシャライゼーションデータがマニフェストのトップレベル要素内に50

あることを判定するときに、（すなわち、例えば依存する Representation の場合には、いかなる Representation でも、イニシャライゼーションデータは、すべてのリプレゼンテーションに共通である）、サーバは、プッシュされたデータとクライアントが選択したであろうものとの間のミスマッチのリスクがないので、「高」値に設定された confidence_level パラメータセットによりイニシャライゼーションデータを指定する PUSH_PROMISE フレームを直ちに（すなわちステップ 2708 と同時に）送信することができる。例えば HTTP ヘッダとして PUSH_PROMISE フレームとともに confidence_level パラメータを送信する利点は、プッシュプロミスを受け入れるかまたはキャンセルする際に、それがクライアント装置を支援することができるということである（以下に説明する図 28 の記述を参照）。 10

【0291】

この機能のおかげで、クライアントは、（PUSH_PROMISE フレームが初期に送信されるので）その復号器を設定するのに必要なイニシャライゼーションデータを一層早く受信することになる。イニシャライゼーションデータが所定のメディアタイプごとにユニークであるときに（例えば、この Adaptation Set 内のリプレゼンテーションの数はいくつであっても Adaptation Set 当たり 1 つの単一の Initialization Segment である）、これもまた動作する。この一層高速のプッシュは、マニフェストの解析（上記のステップ 2708）の直後に行われることになり、従ってログまたはプレファレンスを処理する（上記のステップ 2701、2703、および 2705）前に行われる。 20

【0292】

そして、先にサーバによって決定された confidence_level パラメータが、値「中」以上であれば（テスト 2714）、サーバは、クライアントのために適しているものとして考えられる第 1 のメディアデータをプッシュするイニシアチブをとる。

【0293】

これは、2 つステップにおいて反復して行われ、最初に PUSH_PROMISE フレームが送信され（図 26 のステップ 2656 に対応するステップ 2715）、そして、第 1 のメディアデータのプッシュがステップ 2719 において開始する。これは、ステップ 2709 においてプッシュされるように選択された各第 1 のメディアデータセグメントに対して繰り返される。 30

【0294】

可能な実施形態によれば、連続するメディアセグメントがプッシュされることがプロミスされるときに（すなわち、複数の PUSH_PROMISE がそれぞれのメディアセグメントのために送信される）、現在のメディアセグメントに関連付けられた PUSH_PROMISE は、先の PUSH_PROMISE のチャイルドまたはフォロアとしてマークされる（ステップ 2716）。サーバがステートレスである場合、これは、PUSH_PROMISE フレーム内に新たな HTTP ヘッダとして加えることができ、また、サーバがステートフルである場合テーブル内に保持されることができる。この関連性の保持は、プッシュプロミスに対して階層的なキャンセルを実行するのに有用かもしれない（図 28 を参照して以下で説明するように）。 40

【0295】

データの様々な送信が可能なスケジュールは、以下の通りである。すなわち、実際に第 1 のメディアデータをプッシュする前に、サーバは上記のステップ 2717a においてイニシャライゼーションデータをプッシュし始める。第 1 のメディアデータおよびイニシャライゼーションデータに関する PUSH_PROMISE フレームの送信と並行して、サーバは、また、ステップ 2718 において、MPD ファイル（マニフェスト）を送信し、プッシュされたデータが完全に送信されるまで、ストリームのオープンを保持する。

【0296】

別の実施形態において、テスト 2714 は、いかなる信頼度であっても、第 1 のメディア 50

データをプッシュするために回避することができる。但し、confidence_level パラメータが「低」に設定されている場合、サーバは、第 1 の（または初期の）メディアデータを実際にプッシュする前に、クライアントからの可能性のある CANCEL を待ってもよい。

【0297】

第 1 のメディアデータをプッシュすると、サーバは、プッシュするデータの全体的な量を決定し、使用する速度を決定する（フロー制御）。

【0298】

第 1 の態様に関して、サーバは、例えばマニフェストの先頭における前述の minBuffferTime 属性などのマニフェストからの情報を活用することができる。この属性を用い、ステップ 2709 または 2711 において選択された Representation 考慮し、所定のセグメント期間属性がマニフェスト内にあるとすると、サーバは、minBuffferTime 制約を実現するためにプッシュするセグメントの数（すなわち、セグメントの量、したがってプッシュされる初期のメディアデータを形成するデータの量）を容易に決定できる。有なことに、マニフェストの解析をオフラインで実行する場合（ステップ 2708）、第 1 のメディアセグメントのこの数を、サーバのメモリ内のテーブル内に記録することができる。

10

【0299】

第 2 の態様に関して、セグメントの期間および選択された Representation の帯域幅を与えられると、必要なビットレートの推定は、サーバによって取得されるであろう。これは、主としてビデオセグメントのために、用いる送信速度を提供する。例えば、5 秒の期間のセグメントを有する、1.6 メガビット / 秒に等しい帯域幅により圧縮されたビデオリプレゼンテーションに対して、各セグメントは、送信する 1 メガバイトのデータを表現するだろう。デフォルトでは、HTTP v2.0 におけるフロー制御は、65535 バイトにほぼ等しいストリームウィンドウサイズを提供する。したがって、我々の例において、これは、プッシュされた 65536 バイトの各パケットに対して、クライアントは、サーバに戻す肯定応答を送信しなければならず、そのため、我々の例においては、1 セグメント当たり 15 回を超える！ということを意味する。開発中の HTTP 2.0 下でプッシュ機能を用いる場合にネットワーククラウンドトリップおよびトラフィックを減少させることを目指すので、我々は、（ネットワークトラフィックを減少させることによって）DASH 高速起動を可能にするためにデフォルト挙動（実際にはデフォルトの輻輳ウィンドウサイズ）を改善する必要がここにある、ということを明確に認識する。

20

【0300】

クライアント装置が、マニフェストに対する要求内に含まれたプレファレンスを送信する場合、要求の直後に SETTINGS フレームが送信されるべきである、ということを、また、示すことができ、この SETTINGS フレームは、そのバッファリング容量にしたがって、例えば初期ウィンドウサイズ（SETTINGS_INITIAL_WINDOW_SIZE）を指定する。可能な変形物によれば、この SETTINGS フレームは、接続設定時間に送信することができる。別な可能性は、クライアント装置が、第 1 のプッシュデータを通知する場合に適切なサイズの WINDOW_UPDATE を送信することである。

30

【0301】

図 28 は、本発明の教示による、例えば図 27 に記載されたような方法を実行するサーバとデータを交換する場合に、クライアント装置によって実施される可能な方法を記述する。

40

【0302】

この方法の可能なアプリケーションによれば、クライアント装置は、ビデオ・オン・デマンドサービスから恩恵を得るために、サーバに対して接続する。クライアントとサーバとの間の接続確立は、従来のものである。本例において、クライアント装置およびサーバの両方は、例えば既に前述したドキュメント「ハイパーテキスト転送プロトコルバージョン

50

2.0 (Hypertext Transfer Protocol version 2.0)、draft-ietf-hhtpbis-http2-latesf」に記述されたHTTP/2.0プロトコルを用いてメッセージを交換することができる。

【0303】

ある時に(例えばクライアント装置におけるユーザが所定のビデオを選択するときに)、クライアント装置は、メディアプレゼンテーション(ここではユーザが見たいビデオ)を記述するマニフェストのアドレス(例えばURL)上のサーバから情報を得る。

【0304】

そして、クライアント装置は、マニフェストをダウンロードする要求を準備する(ステップ2800)。好みの実施形態において、クライアントは、HTTPヘッダを通じてビデオ解像度、コーデック、それを支援する帯域幅に、いくつかのプレファレンスを追加する(ステップ2801)。そして、クライアント装置は、サーバに対してその要求を送信する(ステップ2802)。

10

【0305】

本実施形態において、クライアント装置は、そして、そのバッファリング容量に従って初期ウィンドウサイズ SETTINGS_INITIAL_WINDOW_SIZE)を示すために、ステップ2803において、HTTP/2.0 SETTINGSフレームを送信する(上記のドキュメント「ハイパーテキスト転送プロトコルバージョン2.0(Hypertext Transfer Protocol version 2.0)、draft-ietf-hhtpbis-http2-latesf」のセクション3.8.5を参照)。

20

【0306】

ステップ2804において、クライアント装置は、様々なサーバ応答を処理し始める。すなわち、マニフェストを形成するデータを受信し、そのマニフェストを解析する(ステップ2805)だけでなく、サーバによって送信されるPUSH_PROMISEフレームも受信し、解析する(ステップ2806)。

【0307】

PUSH_PROMISEフレーム(単数または複数)内の指定されたプッシュ(単数または複数)を受け入れるかキャンセルするかを決定する前に、クライアントは、サーバがプッシュするように意図するリソースのURLを構築し(ステップ2806)、サーバによってPUSH_PROMISEフレーム内に含まれているconfidence_levelパラメータをチェックする(ステップ2807)。

30

【0308】

並行して、そしてマニュフェスト(記述ファイル)が完全に受信されると、クライアント装置は、入手したい所望のメディアセグメントのリスト(すなわちそのニーズに最も適する各セグメントのバージョンのリスト)を構築し(ステップ2808)、現在のsegment_index変数を0にイニシャライズする(ステップ2809)。PUSH_PROMISEの処理において第1のステップは、confidence_levelパラメータをチェックすることにある(ステップ2810a)。そして、例えば(事前に定義された)クライアントセッティングまたはユーザのプレファレンスに依存して、クライアントは、例えば、PUSH_PROMISEフレームが「低」値のconfidence_levelパラメータを含むPUSH_PROMISEなどの、一定レベルの信頼度下のPUSH_PROMISEを拒否することを決定することができる。

40

【0309】

クライアントが(前述のステップ2808においてマニフェストから導き出されたような)所望のセグメントのURLとPUSH_PROMISEフレーム内の前述のURLとをマッチさせることができる(ステップ2810b)ならば、それらの送信状態により送信されている保留中のセグメントのリストのためのテーブルをイニシャライズする(ステップ2811)。ステップ2810bにおいて、所望のメディアセグメントのリスト内のサーバによってプッシュされるように意図されたセグメントをクライアントが識別すること

50

ができなければ、サーバに対して適切な CANCEL 命令を送信することによってプッシュをキャンセルする（ステップ 2812）。

ステップ 2810b におけるセグメント識別を容易にするために、クライアントは、例えば、MPD ツリー表現内のパス（図 5b を参照）として、プッシュされたセグメントのインデックス、または記述ファイル（すなわち MPD ファイルまたはマニフェスト）が segmentTemplate に依存する場合には URL テンプレートパラメータのような、付加的なヘッダ情報を活用することができる。

【0310】

これは、PUSH_PROMISE を構築する際にサーバによって挿入された階層関係を用いるので（上記図 27 の記述を参照）、ここでは特定の CANCEL メッセージであり（ステップ 2812）、クライアントは、後続の 1 つを加えた現在の PUSH_PROMISE のキャンセルをもたらす再帰的な CANCEL を送信することができる。 10

【0311】

可能な実施形態によれば、クライアント装置がプッシュプロミスを解釈することができない場合、メディアのリソースの次の時間セグメントに対応するメディアデータのすべてのプッシュを、デフォルトによって停止する。

【0312】

CANCEL 命令のこの新たな利用により、メディアセグメント識別の観点から、一旦それがサーバと非同期にされると、クライアントが CANCEL メッセージを繰り返すことを回避する。そのような場合、クライアントは、プルモードに戻ることになる。 20

【0313】

そして、プッシュによってサーバから受信されるセグメントが所望のセグメントに対応する場合（テスト 2810b で真）、クライアントは、PUSH_PROMISE フレームの処理を継続する（テスト 2813 およびステップ 2806 上でのループ）。

PUSH_PROMISE フレームがすべて処理されると、クライアント装置は、受理した PUSH_PROMISE に対応するデータを受信してバッファリングすることを（ステップ 2814）予期し、開始する。

【0314】

十分なメディアセグメントがクライアントの受信バッファ内に受信されると（テスト 2815）、それらはクライアントによって処理される（2816）。そして、現在の segment_index 变数は、リスト内の第 1 のセグメントの順序付け数で更新される（ステップ 2817）。すべてのクライアントがクライアントのバッファに対してアクセスするとは限らないかもしれない、ということに留意するべきである。例えば、特にウェブアプリケーションは、ウェブブラウザのキャッシングに対して通常はアクセスしない。このようなケースにおいて、サーバは、ウェブアプリケーションクライアントに対してプッシュされたセグメントのリストを直接送信してもよい。この情報は、例えばウェブソケット接続を用いて、サーバからクライアントまで交換ができる。 30

【0315】

プッシュされたメディアセグメントがすべて処理されると、クライアントは、次に、標準的なプルベースの DASH（ステップ 2818）に戻り、変数 segment_index + 1 によって指定された、次のセグメントに対応するデータの要求を開始する。並行して、プッシュされたセグメントデータは、選択されたビデオの復号化および表示を開始するためには用いられる。 40

【0316】

図 13 は、実施形態に従う装置の概略図である。装置は、サーバ、クライアント、またはプロキシであってもよい。装置は、実施形態による方法を実施するように構成された制御部 1301 のためにワークメモリとして用いられる RAM メモリ 1302 を備える。例えば、制御部は、ROM メモリ 1303 からロードされたコンピュータプログラムの命令を実行するように構成されるであろう。プログラムは、また、ハードドライブ 1306 から 50

ロードしても良い。例えば、コンピュータプログラムは、図8～図12、図14、図15、図17、図20～図22、および図26～図28のフローチャートおよび上記の記述に基づいて設計される。

【0317】

装置は、また、単一のネットワークインターフェースであってもよいネットワークインターフェース1304を備えるか、または、1セットのネットワークインターフェース（例えば、いくつかの無線インターフェース、またはいくつかのタイプの有線若しくは無線インターフェース）を備える。装置は、ユーザに対する情報を表示するための、およびユーザから入力を受信するための、ユーザインターフェース1305を備えてよい。

【0318】

装置は、また、外部装置から／外部装置に、データを受信および／または送信するための入出力モジュール1307を備えてもよい。

【0319】

本発明が図面および前述の記述に詳細に図示され記述されたが、このような例証および記述は、例証または例示的であって限定的ではなく、本発明が、開示された実施形態に限定されない、ということを考慮すべきである。開示された実施形態にする他の変形物は、図面、開示物、および添付の請求項に関する研究から、特許請求された発明を実施する際に、当業者によって理解され、達成され得る。

なお、本実施形態は以下のように捉えることもできる。

すなわち、サーバの観点に対応する本発明の第1の態様によれば、サーバ装置によってメディアデータをクライアント装置に対してストリーミングする方法は、

・クライアント装置から第1のメディアデータに関連する要求を受信するステップと、・要求されていなくともクライアント装置に対して送信される第2のメディアデータを識別するステップと、

・前記要求に応答して、前記クライアント装置に対して前記第1のメディアデータに関連する情報を送信し、クライアント装置に対してアンウンスメッセージ（単数または複数）を送信するために前記第2のメディアデータをそれぞれ識別する少なくとも1つのアンウンスメッセージを準備するステップと

を備え、

前記方法は、サーバ装置がクライアント装置に対する第2の非要求メディアデータの識別または送信を駆動するためにクライアント装置と共有されるプッシュポリシーを用いるステップをさらに備える。

クライアントの観点に対応する本発明の第2の態様によれば、サーバ装置によってストリーミングされたメディアデータにクライアント装置によってアクセスする方法は、

・第1のメディアデータに関連する要求をサーバ装置に対して送信するステップと、・前記要求に応答して、前記第1のメディアデータに関連する情報を前記サーバ装置から受信するステップと

を備え、

前記方法は、クライアント装置が、クライアント装置によって要求されることなくサーバ装置によって送信される第2のメディアデータを決定するために、またはサーバ装置によってその送信の順序を決定するために、サーバ装置と共有されるプッシュポリシーを用いるステップをさらに備える。

特に、共有されるプッシュポリシーは、クライアント装置に対してサーバ装置によって送信される第2の非要求メディアデータを決定する装置のために、第2のメディアデータを決定する方法を定義してもよい。

このアプローチにより、サーバのプッシュされるメディアデータに関する判定とクライアントのニーズとの間のミスマッチを減少させることができ、それにより、リソースを節約することができる。

これは、クライアントが、サーバの挙動と、それによりこれからプッシュされる第2のメディアデータを予期することを可能にする共有されるプッシュポリシーを用いることによ

10

20

30

40

50

つて、実現される。いくつかのクライアントの後続の要求のために用いられる共有されるプッシュポリシーにより、クライアントは、要求がサーバに対して送信される前でさえも、サーバの挙動を予期することができる。

予期の結果として、クライアントは、サーバによるアナウンスに関して予期される方式で、必要でないこのようないくつかの第2のメディアデータのキャンセルを準備し要求することができる。

第1のメディアデータに関連する要求は、第1のメディアデータおよび／またはこの第1のメディアデータに関連する他のデータに関係してもよい。

第2のメディアデータは、例えばサーバ装置によって、前記第1のメディアデータに関連づけられてもよい。

本発明の実施形態は、サーバ誘導 (server-guided)ストリーミングのための軽量のメカニズムを提供する。実施形態は、DASHネットワークとの関連で実施されてもよい。

サーバ装置は、クライアント装置に対してコンテンツ推薦を行うことができる。また、それらはネットワーク利用を最適化することができる。

本発明の実施形態は、既存のHTTP/2機能との互換性をもつ。これらの機能は、本発明の実現実施形態のために有利に用いることができる。

ネットワークの性能は、一般的には増加する。

それに対応して、本発明は、また、クライアント装置に対してメディアデータをストリーミングするためのサーバ装置に関係し、サーバ装置は、

・クライアント装置から第1のメディアデータに関連する要求を受信するように構成された受信器と、

・要求されることなくクライアント装置に対して送信される第2のメディアデータを識別するように構成された制御部と、

・前記要求に応答して、前記第1のメディアデータに関連する情報を前記クライアント装置に対して送信し、クライアント装置に対してアナウンスマッセージ（単数または複数）を送信するために前記第2のメディアデータをそれぞれ識別する少なくとも1つのアナウンスマッセージを準備するように構成された送信器と
を備え、

前記制御部は、クライアント装置に対する第2の非要求メディアデータの識別または送信を駆動するためにクライアント装置と共有されるプッシュポリシーを用いるようにさらに構成される。

本発明は、また、サーバ装置によってストリーミングされたメディアデータへのアクセスのためのクライアント装置に関係し、前記装置は、

・サーバ装置に対して第1のメディアデータに関連する要求を送信するように構成された送信器と、

・前記要求に応答して、前記サーバ装置から前記第1のメディアデータに関連する情報を受信するように構成された受信器と
を備え、

クライアント装置は、クライアント装置によって要求されることなくサーバ装置によって送信される第2のメディアデータを決定するために、またはサーバ装置によってその送信の順序を決定するために、サーバ装置と共有されるプッシュポリシーを用いるように構成される。

サーバおよびクライアント装置は、上記の対応する方法と同じ利点を有する。

方法および装置のオプション機能は、従属請求項において定義される。それらのいくつかは、方法に関して以下に説明される。しかしながら、それらは、また、対応する装置に対して適応することができる。

明示的なアプローチに関して以下に参照された、いくつかの実施形態において、サーバの観点からの方法は、

サーバ装置によってプッシュポリシーを決定するステップと、

クライアント装置とプッシュポリシーを共有するために決定されたプッシュポリシーを記述するプッシュポリシー情報を、サーバ装置からクライアント装置に対して送信するステップと

をさらに備える。

それに対応して、クライアント側において、前記方法は、共有されたプッシュポリシーを記述するプッシュポリシー情報を、サーバ装置から受信するステップをさらに備えてよい。

以下のいくつかの例に記述されるように、共有されたプッシュポリシーを記述するプッシュポリシー情報は、サーバ装置からクライアント装置に対して送信される記述ファイル内に挿入され、記述ファイルは、第1のメディアデータを含むメディアデータに関する記述情報を含み、前記方法は、共有されたプッシュポリシーを用いて、前記記述ファイルに基づいて第2の非要求メディアデータを決定するステップをさらに備える。

10

具体的な実施形態において、記述ファイルは、複数のメディアデータ属性レベルを用いてメディアデータを記述し、様々な共有されるプッシュポリシーは、様々なそれぞれのレベルの記述ファイルにおいて定義される。

他の例において、共有されるプッシュポリシーを記述するプッシュポリシー情報は、サーバ装置からクライアント装置に対して送信されたHTTPフレームのヘッダ内に埋め込まれる。

具体的な機能によれば、前記方法は、サーバ装置において、クライアント装置からHTTPフレームのヘッダ内に埋め込まれたプッシュポリシー更新情報を受信し、それに応じて、クライアント装置によって要求された他のメディアデータから非要求メディアデータを決定する前に共有されるプッシュポリシーを更新するステップをさらに備えてよい。

20

それに対応して、前記方法は、クライアント装置において、HTTPフレームのヘッダ内に埋め込まれたプッシュポリシー更新情報を、サーバ装置に対して送信するステップをさらに備えてよい。

複合型アプローチによれば、共有されるプッシュポリシーを記述するプッシュポリシー情報は、第1のプッシュポリシー部および第2のプッシュポリシー部によって定義され、第1のプッシュポリシー部は、サーバ装置からクライアント装置に対して送信される記述ファイル内に挿入され、記述ファイルは、第1のメディアデータを含むメディアデータに関する記述情報を含み、前記方法は、共有されたプッシュポリシーを用いて、前記記述ファイルに基づいて第2の非要求メディアデータを決定するステップをさらに備え、

30

第2のプッシュポリシー部は、サーバ装置からクライアント装置に対して送信されるHTTPフレームのヘッダ内に埋め込まれる。

例えば、第2のプッシュポリシー部は、第1のプッシュポリシー部内に定義された1つ以上の関連変数に対する1つ以上の値を備えてもよい。

また、記述ファイルは、複数の候補プッシュポリシーの記述を含んでもよく、第2のプッシュポリシー部は、それにより前記複数からの候補プッシュポリシーの識別子を備えてもよく、識別された候補プッシュポリシーは、それによって第1のプッシュポリシー部を形成する。

他の実施形態において、プッシュポリシー情報は、サーバ装置からクライアント装置に対して送信されたウェブページ内に埋め込まれたJava(登録商標)Scriptプログラムを含む。

40

さらに他の実施形態において、前記方法は、構造化ドキュメント(上記の記述ファイルまたは以下の例に導入されるHTMLページなど)に基づいて第2の非要求メディアデータを決定するステップをさらに備え、構造化ドキュメントは、第1のメディアデータを含むメディアデータに関する記述情報を含み、

プッシュポリシー情報は、第2の非要求メディアデータを識別する構造化ドキュメントのツリー表現上で評価される、XPath式を含む。

前記プッシュポリシー情報の構文に関して、実施形態は、プッシュポリシー情報が記述ファイル内に識別された第2の非要求メディアデータの量を定義する第1のプッシュ属性を

50

含む、ということを提供し、

前記記述ファイルは、第1のメディアデータを含むメディアデータに関する記述情報を含み、前記方法は、共有されたプッシュポリシーを用いて、前記記述ファイルに基づいて第2の非要求メディアデータを決定するステップをさらに備える。

具体的な機能によれば、第1のプッシュ属性は、記述ファイル内の要求された第1のメディアデータに相対する第2の非要求メディアデータを識別する。これは、以下に記載されるようなオペレータを用いて行われてもよい。

変形において、第1のプッシュ属性は、記述ファイル内部の具体的なメディアデータの識別子である。

具体的な機能によれば、記述ファイル内の記述情報は、メディアデータが属する期間を定義する期間属性と、メディアデータのメディアタイプを定義するアダプテーション属性と、メディアデータの符号化バージョン（例えば、ビットレート、フレームレート、フレーム解像度、タイミング情報、など）を定義するリプレゼンテーション属性と、定義するセグメント属性との中の少なくとも1つのメディアデータ属性にしたがってメディアデータを記述し、

前記プッシュポリシー情報は、第2の非要求メディアデータを識別するために、メディアデータ属性（単数または複数）上の制約を定義する少なくとも第2のプッシュ属性を含む。

これは、記述ファイルを通じて非常に選択的なプッシュポリシーを有することを可能にする。

10

特に、プッシュ属性（単数または複数）は、記述ファイル内部の第1のメディアデータの対応するメディアデータ属性（単数または複数）に相対する第2の非要求メディアデータのメディアデータ属性（単数または複数）を定義してもよい。

または、プッシュ属性（単数または複数）は、第2の非要求のメディアデータが検索されなければならない記述ファイル内のノードを識別してもよい。

20

いくつかの実施形態において、記述ファイル内の記述情報は、メディアデータに関連付けられたプライオリティ属性を含み、各々のメディアデータごとに1つのプライオリティ属性があり、第2のメディアデータの送信の順序は、関連するプライオリティ属性に基づく。これは、プッシュデータの送信の順序を定義するためである。

実施形態において、共有されるプッシュポリシーは、要求された第1のメディアデータから第2のメディアデータを識別する。

30

暗示的アプローチに関して以下に参照された実施形態において、共有のプッシュポリシーは、サーバ装置およびクライアント装置の両方において同じ第2のメディアデータを決定するアルゴリズムを用いて実施され、アルゴリズムは、サーバ装置およびクライアント装置が、要求された第1のメディアデータから同じ第2のメディアデータを決定することを可能にする。

暗示的アプローチおよび明示的アプローチの双方に適応した、いくつかの実施形態において、識別された第2のメディアデータが、各々がアナウンスマッセージを要求する複数のメディアセグメントを備えるならば、前記方法は、クライアント装置に対して送信される单一のアナウンスマッセージに、対応する複数のアナウンスマッセージをマージするステップをさらに備えてもよい。これは、より少量のアナウンスマッセージが送信されるので、帯域幅の消費量を低減するためである。

40

共有されたプッシュポリシーおよびクライアント装置によるプッシュの結果として生ずる予期を実際に活用するために、前記方法は、サーバ装置が対応する準備したアナウンスマッセージを送信しないように、第2の非要求メディアデータの一部の送信をキャンセルすることを要求するキャンセル要求を、クライアント装置から受信するステップをさらに備えてよい。

それに対応して、クライアントにおいて、前記方法は、第2の非要求メディアデータの一部を識別するアナウンスマッセージを送信しないようにサーバ装置を駆動するために、第2の非要求メディアデータの一部の送信をキャンセルすることを要求するキャンセル要求

50

を、サーバ装置に対して送信するステップをさらに備えてよい。

本発明の実施形態において、第2の非要求メディアデータは、サーバ装置によって準備され（および多分受信され）、少なくとも1つのアナウンスマッセージから独立してクライアント装置によって決定され、要求されることなくサーバ装置がクライアント装置に対して送信する予定である第2の非要求メディアデータを識別する。ここで「独立して（independentlly）」は、このような非要請データの今後の送信のクライアント装置に通知する専用のこののようなアナウンスマッセージ（すなわち、PUSH_PROMISE）を認識することなく、クライアント装置が第2の非要請データの決定を行うことができる、ということを意味する。

本発明の他の実施形態において、同じ共有されたプッシュポリシーは、それぞれの第1のメディアデータに関連する複数の要求からそれぞれの非要求メディアデータを決定するために用いられる。経時的に同じプッシュポリシーと連続的な要求とを用いることによって、クライアントは、サーバによる無用データの送信をさらに一層効率的に予期しやすいポジションにあり、それにより、それらの送信および対応するアナウンスマッセージの送信を効率的にキャンセルしやすいポジションにある。

サーバからクライアントに対するプッシュデータの送信の順序の通知に関して、サーバ装置によってクライアント装置に対してメディアデータをストリーミングする方法は、

- ・クライアント装置から第1のメディアデータに関連する要求を受信するステップと、
- ・要求されていなくともクライアント装置に対して送信される第2のメディアデータを識別するステップと、

- ・前記要求に応答して、前記第1のメディアデータに関連する情報と前記第2のメディアデータをそれぞれ識別する少なくとも1つのアナウンスマッセージとを前記クライアント装置に送信するステップと

を備えてよく、

前記方法は、

- ・サーバ装置によって第2のメディアデータ（これは、共有されたプッシュポリシーの一部またはすべてを形成する）の送信の順序を定義するステップと、

- ・送信の順序に関する情報であってクライアント装置がサーバによって定義された送信の順序を決定することを可能にする前記情報を、前記アナウンスマッセージとともに送信するステップと

をさらに備える。

例えば、前記第2のメディアの送信の順序は、クライアント装置によるプライオリティ値にしたがって定義され、メディアデータは、最初に送信される最も高いプライオリティ値を有する。

前記プライオリティ値は、HTTP/2プロトコルにしたがって定義されてもよい。

実施形態によれば、少なくとも1つのプライオリティ値は、ネットワーク帯域幅推定メカニズムに関連付けられ、方法は、

- ・前記メカニズムに関連付けられたプライオリティ値をもつ第2のメディアデータをクライアント装置に対して送信するステップと、

- ・前記第2のメディアデータに応答して、クライアント装置から少なくとも1つの制御フローメッセージを受信するステップと、

- ・受信された前記少なくとも1つの制御フローメッセージに基づいて、利用可能な帯域幅を推定するステップと

をさらに備える。

例えば、サーバ装置は、それぞれのサイズおよび個別のサイズを有する複数のデータフレームにしたがって前記第2のメディアデータを送信する。

前記方法は、前記帯域幅推定に基づいて、第2のメディアデータの送信の更新された順序をサーバ装置によって定義するステップをさらに備えてよい。

実施形態によれば、クライアント装置からの前記要求は、前記第1のメディアデータを備えるメディアデータに関連する記述ファイルの受信を求める要求を備え、記述ファイルは

10

20

30

40

50

、前記第1のメディアデータに関する記述情報を含み、方法は、前記記述ファイルに基づいて第2の非要求メディアデータを決定するステップをさらに備える。

例えば、要求された第1のメディアデータは、ビデオセグメントである。

ストリーミングは、D A S H 規格にしたがって実行されてもよい。

例えば、前記方法は、

- ・クライアント装置から順序更新要求を受信するステップと
- ・前記順序更新要求に基づいて、第2のメディアデータの送信の新たな順序を定義し、第2のメディアデータの送信の前記新たな順序に関する情報を更新するステップと、
- ・送信の順序に関連する前記更新通知にしたがってクライアントに対して前記第2のメディアデータを送信するステップと

10

をさらに備える。

前記方法は、クライアント装置に対して順序更新確認メッセージを送信するステップをさらに備えてもよい。

例えば、前記更新された順序は、前記順序更新要求の受信時刻にクライアント装置に対する送信が始まっていない、第2のメディアデータのために定義される。

例えば、前記順序更新要求は、少なくとも第2のメディアデータの一部のための順序値を備える。

実施形態によれば、前記第2のメディアの送信の順序は、プライオリティ値にしたがって定義され、プライオリティ値が少なくとも第1のメディアデータの一部のために更新されると、要求されていなくともクライアント装置に対して送信される少なくとも第2のメディアデータの一部のためのおよび少なくとも前記第1のメディアデータの一部に関連するプライオリティ値は、それに応じて更新される。

20

例えば、前記第1のメディアおよび第2のメディアは、時間的関係、空間的関係、および品質関係の少なくとも1つにしたがって関連付けられる。

実施形態によれば、

- ・前記第2のメディアデータは、第1のメディアデータの品質を強化するためのエンハンスマントデータを備え、
- ・プライオリティ値が前記エンハンスマント層のメディアデータのために更新されるとき、プライオリティ値は、前記エンハンスマント層のすべてのメディアデータのために更新される。

30

例えば、第1のメディアデータおよび第2のメディアデータは、ビデオ時間セグメントを備え、強化メディアデータの開始時間は、第1のメディアデータのビデオコンテンツに関連する情報に基づく。

例えば、第1のメディアデータのビデオコンテンツに関連する前記情報は、前記記述ファイル内に保存される。

例えば、前記送信の順序は、少なくとも第1のメディアデータと第2のメディアデータとの間の復号化関連性に基づく。

例えば、前記送信の順序は、少なくともメディアデータの統計的人気度に基づく。

例えば、前記送信の順序は、少なくともクライアント装置の端部上のメディアデータの再生時間に基づく。

40

例えば、前記送信の順序は、少なくともメディアデータの推測された送信時刻に基づく。

例えば、前記送信の順序は、少なくともメディアデータに対するユーザ定義の関心に基づく。

前記方法は、

- ・制御メッセージであって、現在再生されているメディアデータをサーバ装置が識別することを可能にする前記制御メッセージをクライアント装置から受信するステップと、
- ・第2のメディアデータの更新される送信の順序を、前記制御メッセージに基づいて、サーバによって定義するステップと、
- ・前記更新された送信の順序にしたがってクライアントに対して前記第2のメディアデータを送信するステップと

50

をさらに備えてよい。

前記方法は、順序更新確認メッセージをクライアント装置に対して送信するステップをさらに備えてよい。

例えば、前記制御メッセージは、クライアント装置のバッファメモリの使用に関連し、前記バッファメモリは、クライアントによって再生されるそれらのメディアデータを保存する。

例えば、サーバ装置は、送信された第1の要求メディアデータの記録を保持し、第2のメディアデータの識別は、バッファメモリおよび前記記録の前記使用に基づいて実行される。

例えば、前記送信の順序情報は、前記アナウンスマッセージ内で送信される。

10

例えば、前記送信の順序情報は、前記アナウンスマッセージの後の専用のメッセージ内で送信される。

クライアントの観点から、サーバ装置によってストリーミングされたメディアデータをクライアント装置によってアクセスする方法は、

- ・第1のメディアデータに関連する要求をサーバ装置に対して送信するステップと、
 - ・前記要求に応答して、前記第1のメディアデータに関連する情報と、要求されていなくともクライアント装置に対して送信される第2のメディアそれぞれ識別する少なくとも1つのアナウンスマッセージとを、前記サーバ装置から受信するステップと
- を備えてよい、

前記方法は、前記アナウンスマッセージとともに第2のメディアデータの送信の順序に関連する情報を受信するステップをさらに備え、前記情報（すなわち共有されたプッシュボリシー）は、サーバによって定義された第2のメディアデータの送信の順序を決定することをクライアント装置が可能にする。

20

前記方法は、サーバ装置によって定義された第2のメディアデータの送信の順序がクライアント装置の端部におけるストリーミング制約を満たすか否かをクライアント装置によって判定し、前記制約が満たされなければ、順序更新要求をサーバ装置に対して送信するステップをさらに備えてよい。

例えば、前記第2のメディアデータの送信の順序は、クライアント装置によるプライオリティ値にしたがって定義され、メディアデータは、最初に送信される最も高いプライオリティ値を有する。

30

例えば、前記プライオリティ値は、HTTP/2プロトコルにしたがって定義される。

実施形態によれば、少なくとも1つのプライオリティ値は、ネットワーク帯域幅推定メカニズムに関連付けられ、前記方法は、

- ・前記メカニズムに関連付けられたプライオリティ値をもつ第2のメディアデータをサーバ装置から受信するステップと、
- ・前記第2のメディアデータに応答して、少なくとも1つの制御フローメッセージを前記サーバ装置に対して送信し、それによって、送信された前記少なくとも1つの制御フローメッセージに基づいてサーバ装置が利用可能な帯域幅を推定することを可能にするステップと

をさらに備える。

40

例えば、クライアント装置は、それぞれのサイズおよび個別のサイズを有する複数のデータフレームにしたがって前記第2のメディアデータを受信する。

例えば、第2のメディアデータの更新される送信の順序は、前記帯域幅推定に基づいて、サーバ装置によって定義される。

例えば、クライアント装置からの前記要求は、前記第1のメディアデータを備えるメディアデータに関連する記述ファイルの受信を求める要求を備え、記述ファイルは、前記第1のメディアデータに関係する記述情報を含み、前記方法は、前記記述ファイルに基づいて第2の非要求メディアデータを決定するステップをさらに備える。

例えば、要求された第1のメディアデータは、ビデオセグメントである。

例えば、前記ストリーミングは、DASH規格にしたがって実行される。

50

方法は、サーバ装置によって定義された第2のメディアデータの新たな送信の順序に関する更新情報にしたがってサーバ装置から前記第2のメディアデータを受信するステップをさらに備えてよい。

前記方法は、順序更新確認メッセージをサーバ装置から受信するステップをさらに備えてよい。

実施形態によれば、前記更新された順序は、前記順序更新要求の受信時刻に送信が始まっている、第2のメディアデータのためにサーバ装置によって定義される。

実施形態によれば、前記順序更新要求は、少なくとも第2のメディアデータの一部のための順序値を備える。

実施形態によれば、前記第2のメディアの送信の順序は、プライオリティ値にしたがって定義され、プライオリティ値が少なくとも第1のメディアデータの一部のために更新されると、要求されていなくともクライアント装置に対して送信される少なくとも第2のメディアデータの一部のための且つ少なくとも前記第1のメディアデータの一部に関連付けられたプライオリティ値は、それに応じて更新される。 10

例えば、前記第1のメディアおよび第2のメディアデータは、時間的関係、空間的関係、および品質関係の少なくとも1つにしたがって関連づけられる。

実施形態によれば、

・前記第2のメディアデータは、第1のメディアデータの品質の強化するためのエンハンスメントデータを備え、

・プライオリティ値がエンハンスメント層の第1のメディアデータの少なくとも一部のために更新されるとき、プライオリティ値は、前記エンハンスメント層のすべてのメディアデータのために更新される。 20

例えば、第1のメディアデータおよび第2のメディアデータは、ビデオ時間セグメントを備え、強化メディアデータの開始時間は、第1のメディアデータのビデオコンテンツに関連する情報に基づく。

実施形態によれば、第1のメディアデータのビデオコンテンツに関連する前記情報は、前記記述ファイル内に保存される。

実施形態によれば、前記送信の順序は、少なくとも第1のメディアデータと第2のメディアデータとの間の復号化関連性に基づく。

実施形態によれば、前記送信の順序は、少なくともメディアデータの統計的人気度に基づく。 30

実施形態によれば、前記送信の順序は、少なくともクライアント装置の端部上のメディアデータの再生時間に基づく。

実施形態によれば、前記送信の順序は、少なくともメディアデータの推測された送信時刻に基づく。

実施形態によれば、前記送信の順序は、少なくともメディアデータに対するユーザ定義の関心に基づく。

前記方法は、

・制御メッセージであって、現在再生されているメディアデータをサーバ装置が識別することを可能にする前記制御メッセージをサーバ装置に送信するステップと、 40

・前記制御メッセージに基づいて、サーバ装置によって、定義された更新された送信の順序にしたがってサーバ装置から前記第2のメディアデータを受信するステップとを備えてよい。

前記方法は、順序更新確認メッセージをサーバ装置から受信するステップを備えてよい。

例えば、前記制御メッセージは、クライアント装置のバッファメモリの使用に関連し、前記バッファメモリは、クライアント装置によって再生されるそれらのメディアデータを保存する。

実施形態によれば、サーバ装置は、送信された第1の要求メディアデータの記録を保持し、現在再生されているメディアの識別は、バッファメモリおよび前記記録の前記使用に基 50

づいて実行される。

例えば、前記送信の順序情報は、前記アナウンスマッセージ内で受信される。

例えば、前記送信の順序情報は、前記アナウンスマッセージの後の専用のメッセージ内で受信される。

さらに送信の順序を参照して、プロキシサーバによる、クライアント装置とサーバ装置との間のデータ交換を管理する方法は、

・送信の順序の通知に関して以上に定義されるような方法を実行するサーバから、クライアント装置に対して再送信されるメディアデータを受信するステップと

・メディアデータの送信の順序に基づいて、メディアデータのための再送信プライオリティを決定するステップと、

・決定された前記送信プライオリティに基づいて、クライアント装置に対する受信されたメディアデータの再送信を実行するステップと

を備えてもよい。

前記方法は、決定された前記再送信プライオリティに基づいて、受信された前記メディアデータを保存するステップをさらに備えてもよい。

前記方法は、

・第2の態様による方法を実施するクライアント装置から順序更新要求を受信するステップと、

・前記要求が再送信されるメディアデータに関連する場合、前記順序更新要求にしたがって前記再送信プライオリティを更新するステップと、

・更新された再送信プライオリティにしたがってメディアデータの再送信を実行するステップと

をさらに備えてもよい。

前記方法は、

・メディアデータのため、第1のサーバ装置に対する要求を、第1のクライアント装置から受信するステップであって、前記メディアデータは、第2のサーバ装置から第2のクライアント装置に対する再送信のためにプロキシサーバによって保存されるステップと、

・第1のサーバ装置および第2のサーバ装置によって前記メディアデータにそれぞれ関連付けられたプライオリティ値を決定するステップと、

・第1のクライアント装置および第2のクライアント装置に対するそれぞれのストリーミング制約にしたがって前記プライオリティ値を更新するステップと、

・前記更新されたプライオリティ値にしたがって、前記第1のクライアント装置および第2のクライアント装置に対して前記メディアデータを再送信するステップと

をさらに備えてもよく、

前記第1のサーバ装置および第2のサーバ装置は、第1の態様による方法を実施し、前記第1のクライアント装置および第2のクライアント装置は、第2の態様による方法を実施する。

前記方法は、更新されたプライオリティ値に関連する更新通知を第1のサーバ装置および第2のサーバ装置に対して送信するステップをさらに備えてもよい。

本発明の別の態様によれば、サーバ装置とクライアント装置との間でデータをストリーミングする方法であって、

・サーバ装置によって第1の態様による方法を実行するステップと、

・クライアント装置によって第2の態様による方法を実行するステップと

を備える方法が提供される。

本発明のさらに別の態様によれば、プログラミング可能な装置のコンピュータ手段にコードされ実行されたときに、以上に定義されるような方法を実施するための命令を備えるコンピュータプログラムおよびコンピュータプログラムプロダクトが提供される。

本発明のさらに別の態様によれば、第1の態様による方法を実施するように構成されたサーバ装置が提供される。

本発明のさらに別の態様によれば、第2の態様による方法を実施するように構成されたク

10

20

30

40

50

ライアント装置が提供される。

サーバからクライアント装置に対するメディアデータの適応型ストリーミングにための解決手段は、関係のあるクライアント装置の機能に対して、およびサーバとクライアント装置との間の接続を提供するネットワークの特性に対して、クライアント装置に送信されるデータのタイプおよび量を特に適応するために提案されてきた。

これに関連して、DASH（動的適応型HTTPストリーミング）規格などのいくつかの解決手段は、配信されるリソース（またはコンテンツ）の複数のバージョンを保存し、且つこれらのバージョンに対してリソースおよびそれぞれのポインタ（例えばURL）を表現する様々なバージョンの記述を含む記述ファイルを、リソースを要求するクライアント装置に対して送信することを提案する。

10

そして、記述ファイルに基づいて、クライアント装置は、対応するポインタを用いて、このバージョンのニーズおよび要求にベストマッチするリソースのバージョンを選択することができる。

この解決手段は、記述ファイルがメディアデータを含まない（但しメディアデータに対するポインタのみ含む）光である、という点で有利である。その利用のために関連するバージョンをクライアントに選択させることによって、クライアント装置に適さないメディアデータの交換を回避する。さらに、それはHTTPに基づく現在のウェブアーキテクチャに適合し、既に配備されたキャッシュメカニズムを利用することができる。

その代わりに、しかしながら、メディアデータがクライアント装置において受信されて、その後、復号され表示され得る前に、この解決手段は、クライアント装置とサーバとの間のいくつかの交換（またはラウンドトリップ）を必要とし、それは開始遅延をもたらす。実施形態において、本発明は、メディアアイテムを表現するデータを保存するサーバからメディアアイテム（例えばビデオ）を表現するメディアデータを提供する方法を提供するのであって、少なくともその時間セグメントは、複数のバージョンによって表現され、方法は、サーバによって実施される以下のステップ、

20

- ・時間セグメントを表現するバージョンに対する時間セグメントおよびそれぞれのポインタを表現するバージョンの記述を含む記述ファイルを求める要求をクライアント装置から受信するステップと、

- ・記述ファイル内でポインタが指すデータのセットの中のデータを選択するステップと、
- ・クライアント装置に対して記述ファイルを送信するステップと、

30

- ・クライアント装置に対して選択されたデータをプッシュするステップとを備える。

適切な方式において選択されたデータをプッシュ（すなわち、クライアント装置によって請求されないが、以下にさらに説明するようにサーバによって選択されたデータの送信）することによって、1つまたは複数のラウンドトリップを回避することができ、それにより、メディアデータの復号化および表示をさらに高速に開始することができる。

メディアアイテムは、例えば、ビデオ、または例えばオーディオトラックなどのオーディオアイテムであってもよい。

上述のデータのセットは、時間セグメントを表現するバージョンを含むが、以下に説明されるようにイニシャライゼーションデータなどの他のデータを含んでもまたよい、ということに注目されたい。

40

まさに述べた如く、選択されたデータは、クライアント装置の復号器のためのイニシャライゼーションデータを含んでもよい。したがって、復号器は、クライアント装置がイニシヤライゼーションデータを特に、要求する必要なしに、イニシャライズでき、従ってより高速にイニシャライズできる。

上述のように、選択されたデータは、また、時間セグメントを表現する前記バージョンの1つの少なくとも一部を含んでもよい。

データを選択するステップは、プッシュされるデータ（例えばビデオデータ）の量を推定するステップを含んでもよく、それは、どのデータが選択されるのかを決定するときに用いられてもよい。量は、記述ファイル内に定義されたバッファ時間に基づいて、および／

50

または、サーバによって決定された帯域幅推定に基づいて推定されてもよい。

データを選択するステップは、要求に含まれる少なくとも1つのプレファレンスに基づいて、および／またはサーバとクライアント装置との間の従来の交換から導き出される使用データに基づいて、および／またはサーバによる記述ファイルの解析に基づいて、および／またはサーバ内に保存され、記述ファイルに関連するテーブルに基づいて、実行されてもよい。

可能な実施形態によれば、選択されたデータをプッシュするステップに関連するとともに、そのステップに先立って、プッシュプロミスを送信するステップを備えてもよい。したがって、これらのデータを実際に受信する前に、プッシュされるデータがクライアント装置に通知されてもよい。

10

プッシュプロミスを送信するステップは、記述ファイルを送信するステップの前に実行されてもよく、それは、初期のステージにおいてクライアント装置に通知することを可能にする。

プッシュプロミスは、例えば選択されたデータの識別を含む。

提案された実施形態によれば、サーバは、選択されたデータに関連する信頼水準を決定し、プッシュプロミスは、決定された信頼水準を含む。

以下に述べる詳細説明において説明する可能なインプランテーションによれば、サーバは、選択されたデータを形成するデータのブロックの階層的リプレゼンテーションを保存してもよい。

このようなケースにおいて、

20

- ・データのブロックをプッシュしないための命令をクライアント装置から受信するステップと、

- ・前記データのブロック、および階層的リプレゼンテーションにおける前記データのブロックに連結されたデータのブロックのプッシングをキャンセルするステップとが備えられていてもよい。

提案された方法は、選択されたデータに関連した信頼度を決定するステップを含んでもよく、そして、

- ・決定された信頼度が所定の閾値未満であれば、選択されたデータをプッシュするステップは、クライアント装置の復号器のイニシャライゼーションデータのみをプッシュするステップを含み、

30

- ・決定された信頼度が所定の閾値を上回れば、選択されたデータをプッシュするステップは、クライアント装置の復号器のためのイニシャライゼーションデータと、時間セグメントを表現する前記バージョンの1つの少なくとも一部をプッシュするステップを含む。

本発明の実施形態は、また、メディアアイテムを表現するデータを保存するサーバからメディアアイテム（例えばビデオ）を表現するメディアデータを受信するための方法を提供するのであって、少なくともその時間セグメントは、複数のバージョンによって表現され、方法は、クライアント装置によって実施される以下のステップ、

- ・時間セグメントを表現するバージョンに対する時間セグメントおよびそれとのポイントを表現するバージョンの記述を含む記述ファイルを求める要求をサーバに送信するステップと、

40

- ・記述ファイルをサーバから受信するステップであって、前記記述ファイルはデータのセットに対するポインタを含むステップと、

- ・サーバから、非要請データを受信するステップであって、前記非要請データは前記データのセットに属するステップとを有する。

上述のように、非要請データは、クライアント装置の復号器のためのイニシャライゼーションデータ（その場合には前記非要請データにより復号器をイニシャライズするステップが提供されてもよい）、および／または、少なくとも時間セグメントを表現する前記バージョンの1つの一部（その場合には非要請データの少なくとも一部を復号するステップが提供されてもよい）を含んでもよい。

50

前記要求は、プッシュされるメディアデータを決定する際にサーバを支援することができる、クライアント装置における復号化を定義する少なくとも1つのプレファレンスを含んでもよい。

前記要求は、また、クライアント装置がプッシュされたデータを受け入れるというインディケータを含んでもよく、それに基づいて、サーバは、データをプッシュすることを効率的に決定することができる。

上記したように、非要請データを受信するステップに関連するとともに、そのステップに先立って、プッシュプロミスを受信するステップを備えてよい。このプッシュプロミスを受信するステップは、記述ファイルを受信するステップに先立って行われてもよい。

プッシュプロミスは、非要請データおよび／または非要請データに関連付けられた信頼度の識別を含んでもよい。

クライアント装置において、以下のステップ、

・プッシュプロミス内に含まれるデータに基づいてプッシュプロミスの受理または拒否を決定するステップと、

・拒否の場合には前記非要請データをプッシュしないための命令を送信するステップとが有していてもよい。

以下のステップ、

・非要請データに関連付けられ且つプッシュプロミス内に含まれた信頼度に基づいて、プッシュプロミスの受理または拒否を決定するステップと、

・拒否の場合には前記非要請データをプッシュしないための命令を送信するステップとが用いられてもよい。

受信に際して前記非要請データをバッファするステップは、これらのデータを復号する前に用いられてもよい。

プッシュされたデータがイニシャライゼーションデータおよび／または初期のメディアデータに対してのみ対応するように意図されるように以下のステップ、

・記述ファイルに、およびプッシュプロミス内に含まれるデータに基づいて要求される（すなわち、プッシュされるようには計画されない）データ（例えばビデオデータ）を決定するステップと、

・決定されたデータに対する要求を、サーバに対して送信するステップと

が実施されてもよい。

本発明の実施形態は、また、クライアント装置に対して、メディアアイテムを表現するデータを保存するサーバからメディアアイテム（例えばビデオ）を表現するメディアデータをストリーミングする方法を提案し、メディアアイテムの少なくとも時間セグメントは、複数のバージョンによって表現され、前記方法は、

・前記クライアント装置が、時間セグメントを表現するバージョンに対する時間セグメントおよびそれぞれのポインタを表現するバージョンの記述を含む記述ファイルを求める要求をサーバに送信するステップと、

・サーバが、クライアント装置から要求を受信するステップと、

・サーバが、記述ファイル内でポインタが指すデータのセットの中のデータを選択するステップと、

・サーバが、クライアント装置に対して記述ファイルを送信するステップと、

・サーバが、クライアント装置に対して選択されたデータをプッシュするステップと、

クライアント装置が、サーバから記述ファイルを受信するステップと、

・クライアント装置が、サーバから選択されたデータを受信するステップとを有する。

本発明の実施形態は、また、サーバからメディアアイテム（例えばビデオ）を表現するメディアデータを提供するための装置を提供するのであって、サーバは、メディアアイテムを表現するデータを保存し、少なくともその時間セグメントは、複数のバージョンによって表現され、前記装置は、

・時間セグメントを表現するバージョンに対する時間セグメントおよびそれぞれのポイン

10

20

30

40

50

タを表現するバージョンの記述を含む記述ファイルを求める要求をクライアント装置から受信するように構成された受信器と、

・記述ファイル内でポインタが指すデータのセットの中のデータを選択するように構成された選択モジュールと、

・クライアント装置に対して記述ファイルを送信するように構成されたモジュールと、・クライアント装置に対して選択されたデータをプッシュするように構成されたモジュールと

を有する。

本発明の実施形態は、また、メディアアイテムを表現するデータを保存するサーバからメディアアイテム（例えばビデオ）を表現するメディアデータを受信するための装置を提供するのであって、少なくともその時間セグメントは、複数のバージョンによって表現され、前記装置は、

・時間セグメントを表現するバージョンに対する時間セグメントおよびそれぞれのポインタを表現するバージョンの記述を含む記述ファイルを求める要求をサーバに送信するように構成されたモジュールと、

・サーバから記述ファイルを受信するように構成されたモジュールであって、前記記述ファイルがデータのセットに対するポインタを含むモジュールと、

・サーバから非要請データを受信するように構成されたモジュールであって、前記非要請データは前記データのセットに属するモジュールと

を有する。

最後に、本発明の実施形態は、メディアアイテムを表現するデータを保存するサーバからメディアアイテム（例えばビデオ）を表現するメディアデータをクライアント装置に対してストリーミングするためのサーバおよびクライアント装置を有するシステムを提供し、

メディアアイテムの少なくとも時間セグメントは、複数のバージョンによって表現され、

・前記クライアント装置は、時間セグメントを表現するバージョンに対する時間セグメントおよびそれぞれのポインタを表現するバージョンの記述を含む記述ファイルを求める要求をサーバに送信するように構成されたモジュールを有し、

・前記サーバは、クライアント装置から要求を受信するように構成されたモジュールと、記述ファイル内でポインタが指すデータのセットの中のデータを選択するように構成された選択モジュールと、クライアント装置に対して記述ファイルを送信するように構成されたモジュールと、クライアント装置に対して選択されたデータをプッシュするように構成されたモジュールと、を有し、

・クライアント装置は、サーバから記述ファイルを受信するように構成されたモジュールと、サーバから選択されたデータを受信するように構成されたモジュールと、を有する、を前記システムは有する。

メディアデータを提供する方法、およびメディアデータを受信する方法のために以上に提案されたオプション機能は、また、メディアデータをストリーミングする方法に対して、および前述の各種装置およびシステムに対して適用する。

【 0 3 2 0 】

特許請求の範囲の単語「備える（c o m p r i s i n g ）」は、他の要素またはステップを除外せず、また、不定冠詞「a」または「a n」は、複数のものを除外しない。単一のプロセッサまたは他のユニットは、特許請求の範囲内の列挙された、いくつかのアイテムの機能を実現してもよい。個別の機能が個別の従属請求項内に相互に列挙されるという單なる事実は、これらの機能の組み合わせを有利に用いることができない、ということを示すものではない。特許請求の範囲内の任意の参照符号も、また、本発明の範囲を制限するものとして解釈するべきでない。

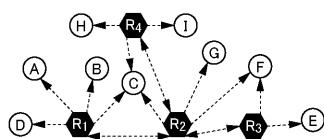
10

20

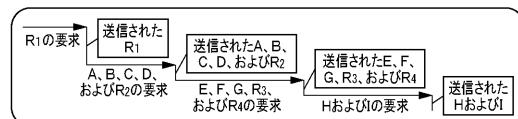
30

40

【図 1 a】



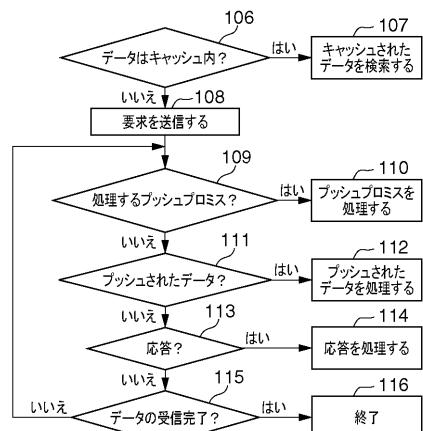
【図 1 b】



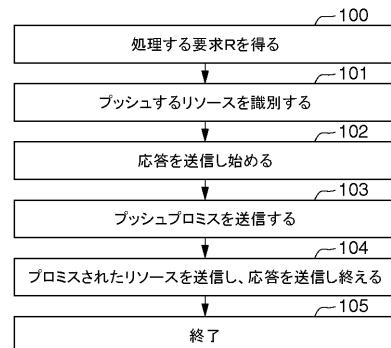
【図1c】



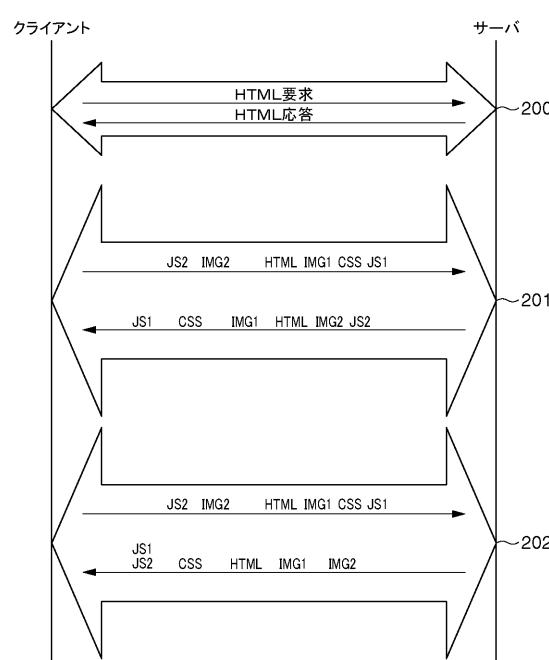
【図1d】



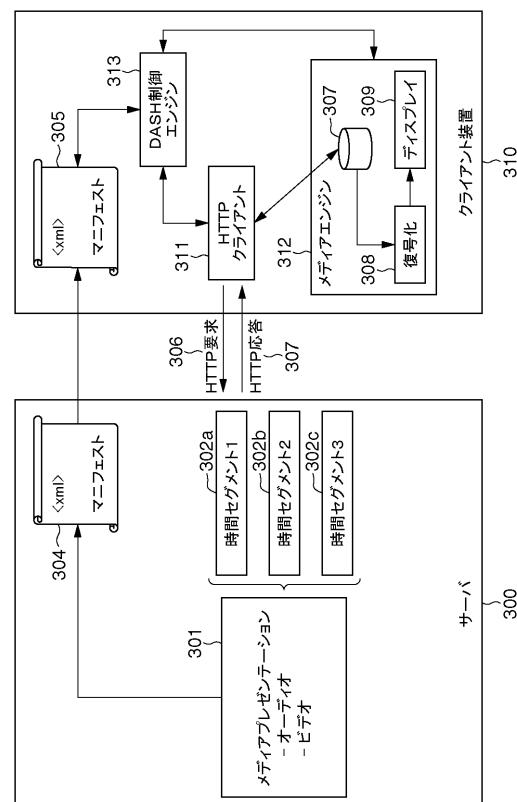
【 図 1 e 】



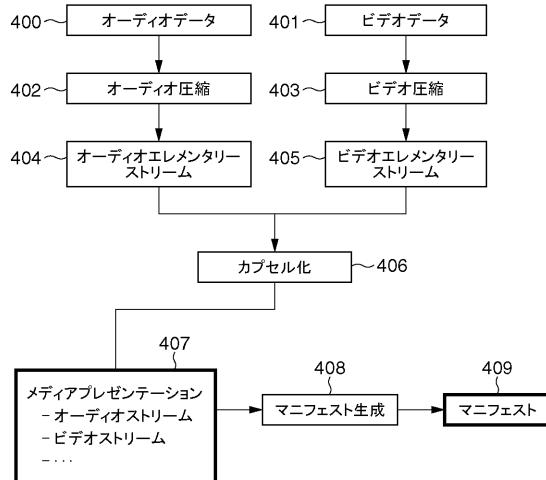
【図2】



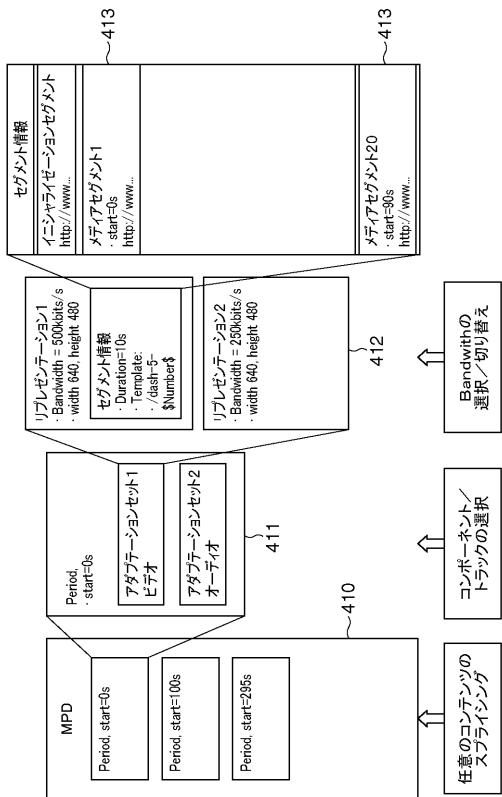
【図3】



【図4 a】



【図4 b】

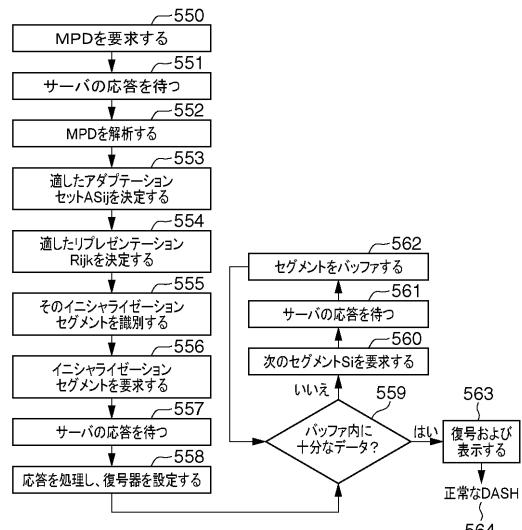


【図5】

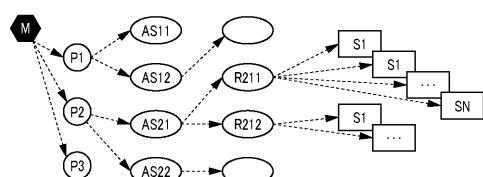
```

<?xml version="1.0"?>
<MPD xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns="urn:mpeg:DASH:schema:MPD-2011"
      xsi:schemaLocation="urn:mpeg:DASH:schema:MPD-2011 DASH-MPD.xsd"
      type="static"
      mediaPresentationDuration="PT3256S"
      minBufferTime="PT12S"
      profiles="urn:mpeg:dash:profile:isoff-on-demand:2011">
  <BaseURL>http://cdn1.example.com/</BaseURL>
  <BaseURL>http://cdn2.example.com/</BaseURL>
  <Period>
    <AdaptationSet mimeType="audio/mp4" codecs="mp4a.0x40" lang="en">
      <segmentAlignment>true</segmentAlignment>
      <ContentProtection schemeIdUri="urn:uuid:706D6953-656C-5244-4D48-656164657221"/>
      501～<Representation id="1" bandwidth="64000">
        <BaseURL>7657412348.mp4</BaseURL>
      </Representation>
      502～<Representation id="2" bandwidth="32000">
        <BaseURL>3463646346.mp4</BaseURL>
      </Representation>
    </AdaptationSet>
    <AdaptationSet mimeType="video/mp4" codecs="avc1.4d0228">
      <segmentAlignment>true</segmentAlignment>
      <ContentProtection schemeIdUri="urn:uuid:706D6953-656C-5244-4D48-656164657221"/>
      503～<Representation id="6" bandwidth="256000" width="320" height="240">
        <BaseURL>8563456473.mp4</BaseURL>
      </Representation>
      <Representation id="7" bandwidth="512000" width="320" height="240">
        <BaseURL>56363634.mp4</BaseURL>
      </Representation>
      <Representation id="8" bandwidth="1024000" width="640" height="480">
        <BaseURL>562465736.mp4</BaseURL>
      </Representation>
      <Representation id="9" bandwidth="1384000" width="640" height="480">
        <BaseURL>41325645.mp4</BaseURL>
      </Representation>
      <Representation id="A" bandwidth="1536000" width="1280" height="720">
        <BaseURL>89045625.mp4</BaseURL>
      </Representation>
      <Representation id="B" bandwidth="2048000" width="1280" height="720">
        <BaseURL>23536745734.mp4</BaseURL>
      </Representation>
    </AdaptationSet>
  </Period>
</MPD>
  
```

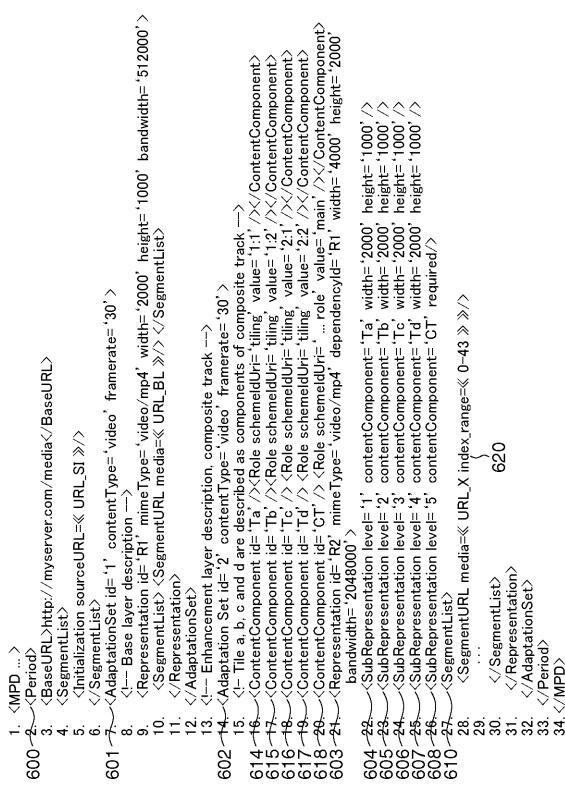
【図5 a】



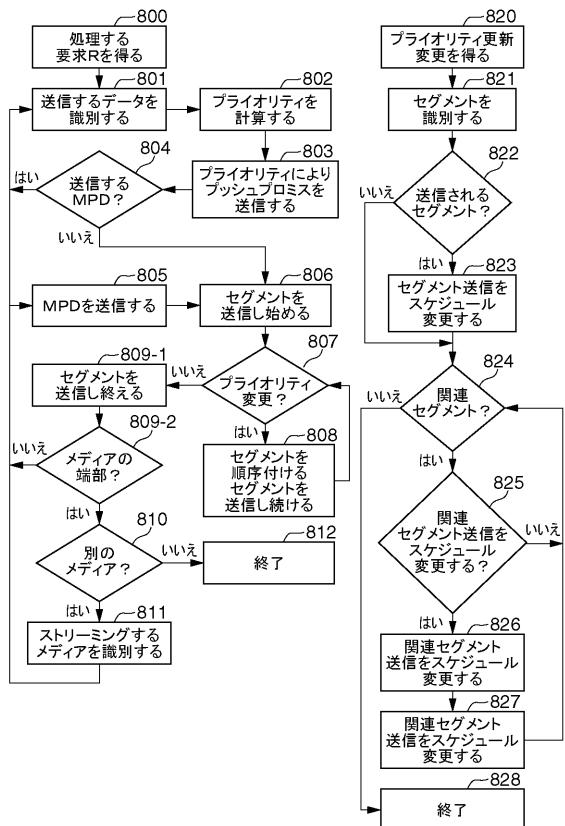
【図5 b】



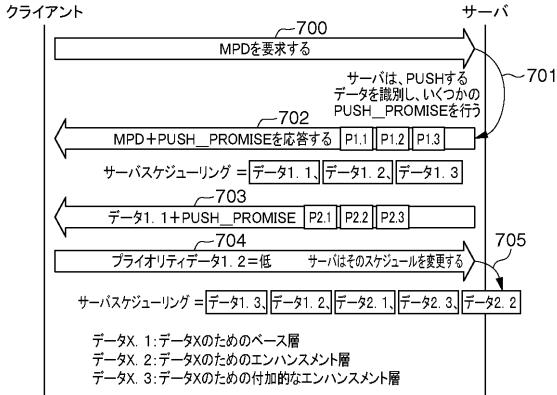
【 义 6 】



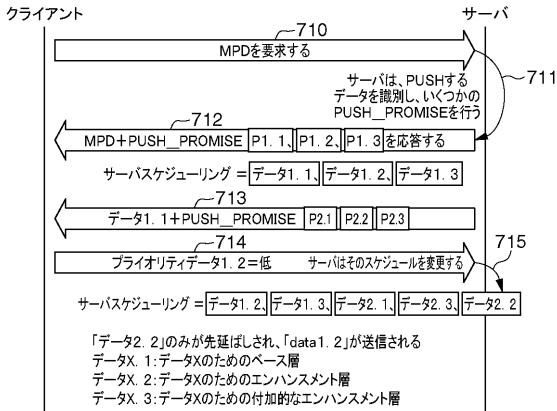
【 义 8 】



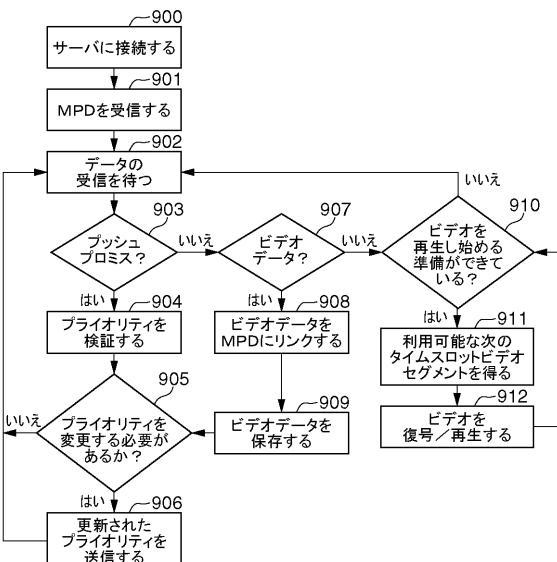
【図 7 a】



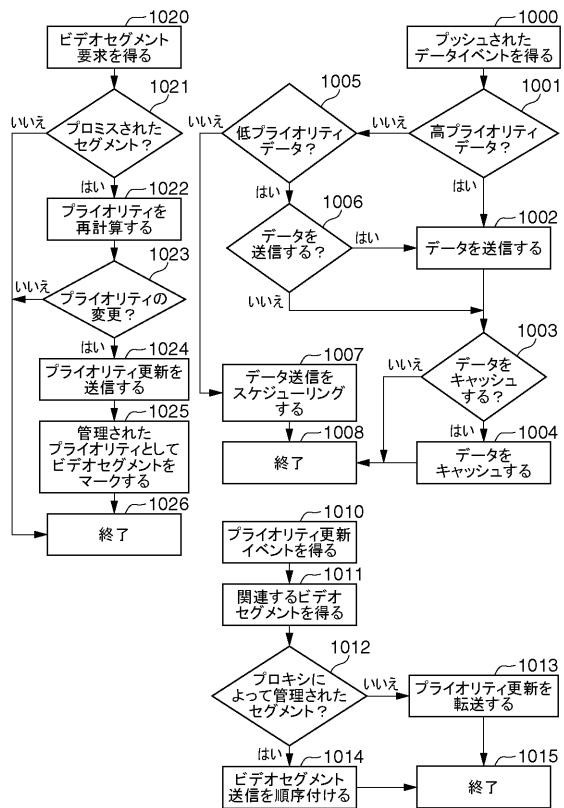
【図 7 b】



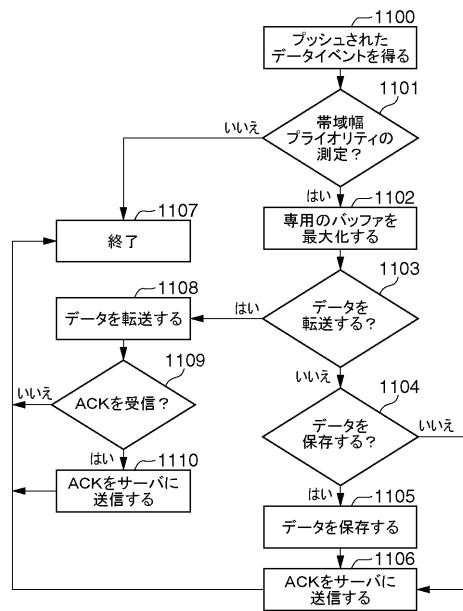
【図9】



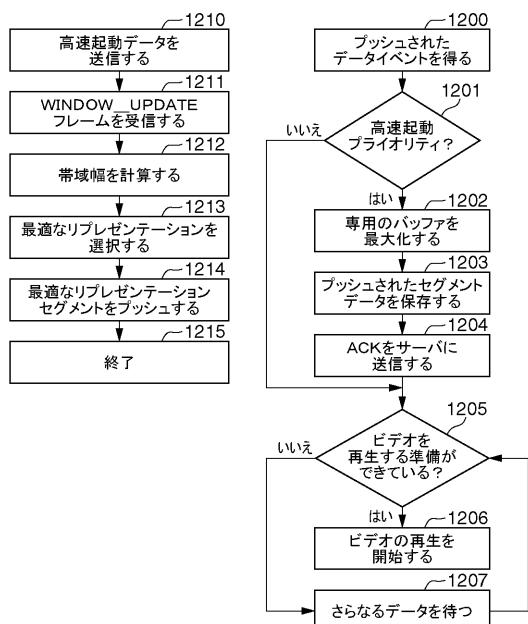
【図10】



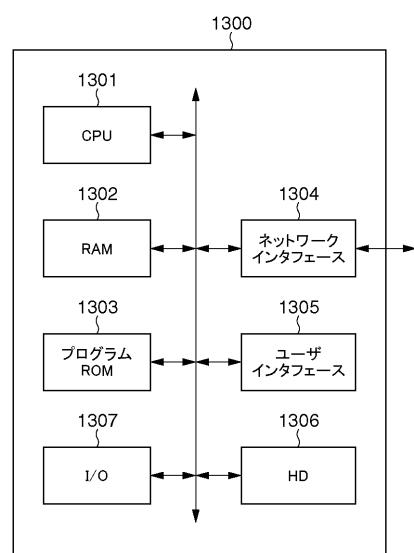
【図11】



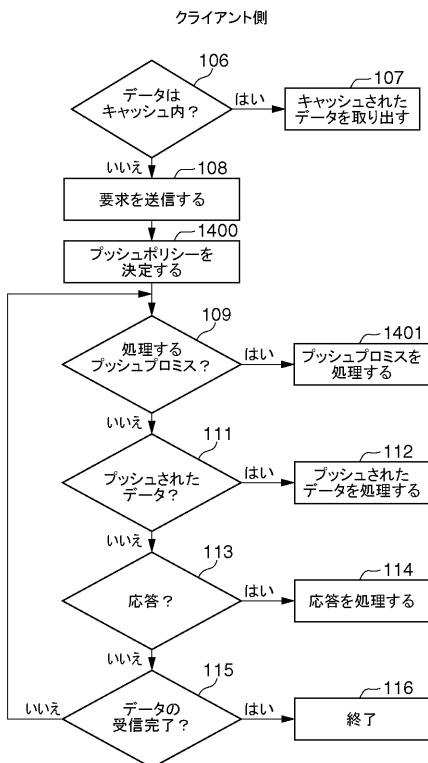
【図12】



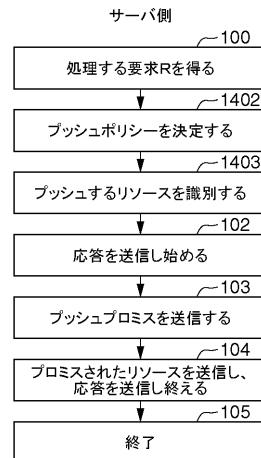
【図13】



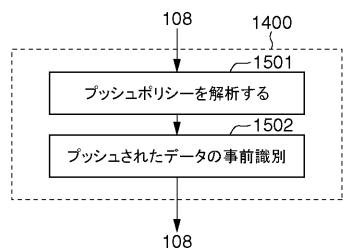
【図14a】



【図14b】



【図15a】

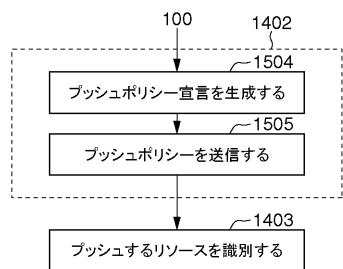


【図16】

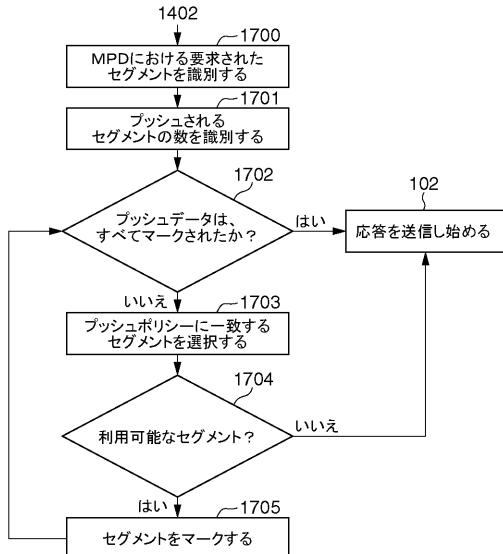
```

<MPD ...>
  <PushPolicy SegmentIdx="2"/> ~ 1600
  <Period>
    <BaseURL>http://myserver.com/media</BaseURL>
    <SegmentList>
      <Initialization sourceURL=<< URL_SI >>>>
    </SegmentList>
    <AdaptationSet id="AS1" mimeType="video/mp4" codecs="hev1">
      <Representation id="R0" width="1920" height="1080" frameRate="30" ....bandwidth="256000">
        <Role schemeURI="urn:mpg:DASHrole:2011" value="main"/>
      <SegmentList duration="10">
        <SegmentURL media="seg-full-AS1R0-1.mp4" priorityIdx="1"/>
        <SegmentURL media="seg-full-AS1R0-2.mp4" priorityIdx="3"/>
        <SegmentURL media="seg-full-AS1R0-3.mp4" priorityIdx="2"/>
        ...
      </SegmentList>
    </Representation>
    <Representation id="R1" width="1920" height="1080" frameRate="30" ....bandwidth="256000">
      <Role schemeURI="urn:mpg:DASHrole:2011" value="associate"/>
      <SegmentList duration="10">
        <SegmentURL media="seg-full-AS1R1-1.mp4"/>
        <SegmentURL media="seg-full-AS1R1-2.mp4"/>
        <SegmentURL media="seg-full-AS1R1-3.mp4"/>
        ...
      </SegmentList>
    </Representation>
  </AdaptationSet>
  <AdaptationSet id="AS2" mimeType="video/mp4" codecs="hev1">
    <Representation id="R1" width="2000" height="1000" bandwidth="512000">
      <SegmentList>
        ...
      </SegmentList>
    </Representation>
  </AdaptationSet>
  ...
</Period>
</MPD>
  
```

【図15b】



【図17】

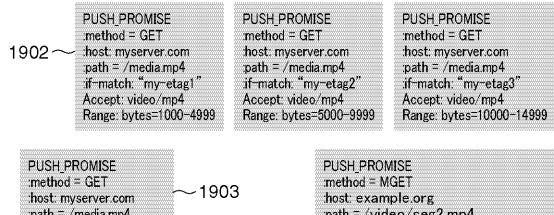


【図19】

1900

```

<MPD ...>
  <PushPolicy segmentIdx="+3"/>
  <Period>
    <AdaptationSet id="AS1" mimeType='video/mp4' codecs='hev1'>
      <Representation id="R0" width='1920' height='1080' frameRate='30' bandwidth='256000'>
        <BaseURL>http://myserver.com/media.mp4</BaseURL> ...
      <SegmentList>
        <InitializationRange="0-999"/>
        <SegmentURL mediaRange="1000-4999" indexRange="1000-2000"/>
        <SegmentURL mediaRange="5000-9999" indexRange="5000-7000"/>
        <SegmentURL mediaRange="10000-14999" indexRange="10000-12000"/>
        <SegmentURL mediaRange="15000-19999" indexRange="15000-17000"/>
      ...
      <SegmentList>
    </Representation>
  </AdaptationSet>
  </Period>
  ...
</Period>
</MPD>
  
```

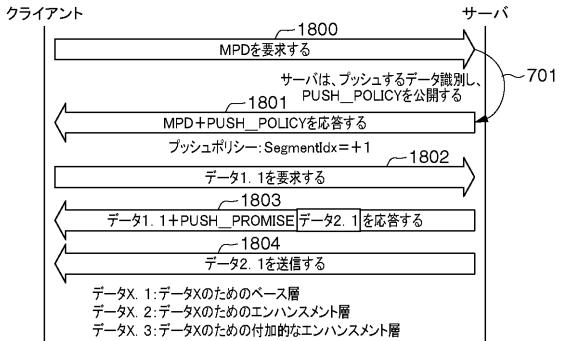


1905

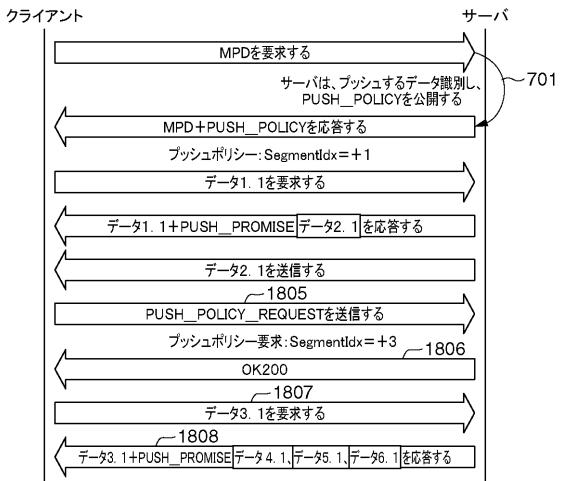
```

PUSH_PROMISE
method = GET
host: example.org
path = /video/segments.mp4
if-match: "my-etag1"
Accept: video/mp4
Range: bytes=123-456, 789-1223, 4567-6546
  
```

【図18 a】



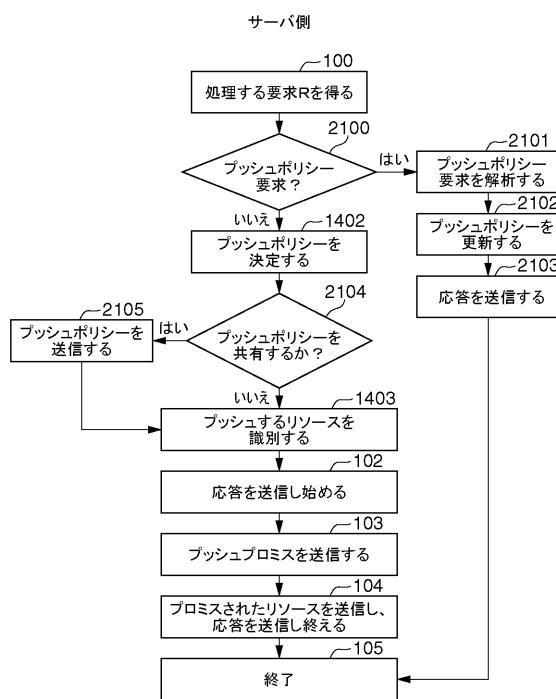
【図18 b】



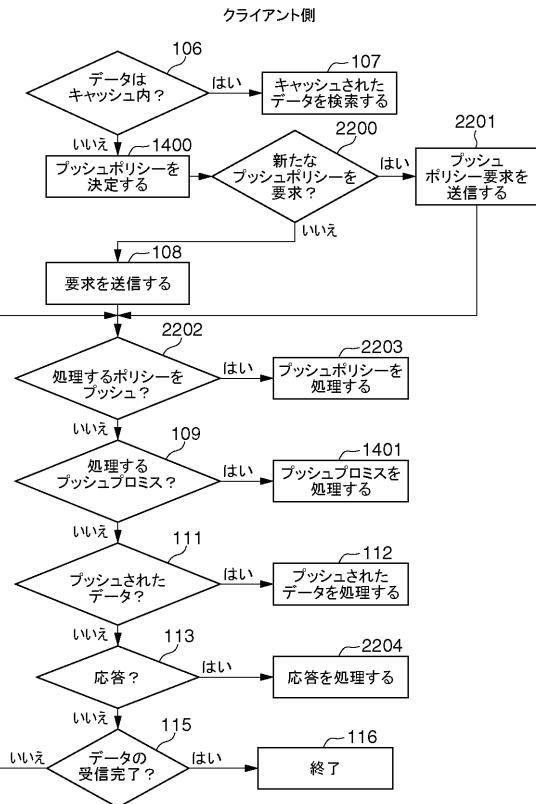
【図20】



【図21】



【図22】



【図23】

```

<MPD ...>
<Period>
<BaseURL>http://myserver.com/media/</BaseURL>
<SegmentList>
    <Initialization sourceURL=< URL_SI >/>
<SegmentList>
<AdaptationSet id="AS1" mimeType='video/mp4' codecs='hev1'>
    <Representation id="R0" width='1920' height='1080' frameRate='30' ....bandwidth='256000'>
        <SupplementalProperty schemeURI='urn:mpeg:dash:push:policy' value='next3' />～2300
        <Role schemeURI='urn:mpeg:DASHrole:2011' value='main' />
    <SegmentList duration='10'>
        <SegmentURL media='seg-full-AS1R0-1.mp4' />
        <SegmentURL media='seg-full-AS1R0-2.mp4' />
        <SegmentURL media='seg-full-AS1R0-3.mp4' />
    ...</SegmentList>
    </Representation>
</AdaptationSet>
<Representation id="R1" width='1920' height='1080' frameRate='30' ....bandwidth='2256000'>
    <SupplementalProperty schemeURI='urn:mpeg:dash:push:policy' value='next1' />～2301
    <Role schemeURI='urn:mpeg:DASHrole:2011' value='associate' />
    <SegmentList duration='10'>
        <SegmentURL media='seg-full-AS1R1-1.mp4' />
        <SegmentURL media='seg-full-AS1R1-2.mp4' />
        <SegmentURL media='seg-full-AS1R1-3.mp4' />
    ...</SegmentList>
    </Representation>
</Adaptation Set id="AS2" mimeType='video/mp4' codecs='hev1'>
    <Representation id="R1" width='2000' height='1000' bandwidth='512000'>
        <SegmentList>
        ...
        </SegmentList>
    </Representation>
</AdaptationSet>
</Period>
</Period>
...
</MPD>

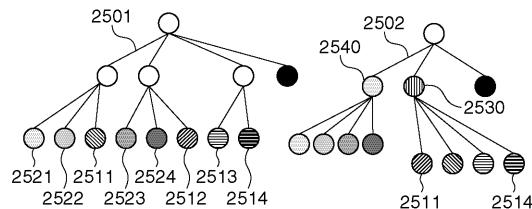
```

```

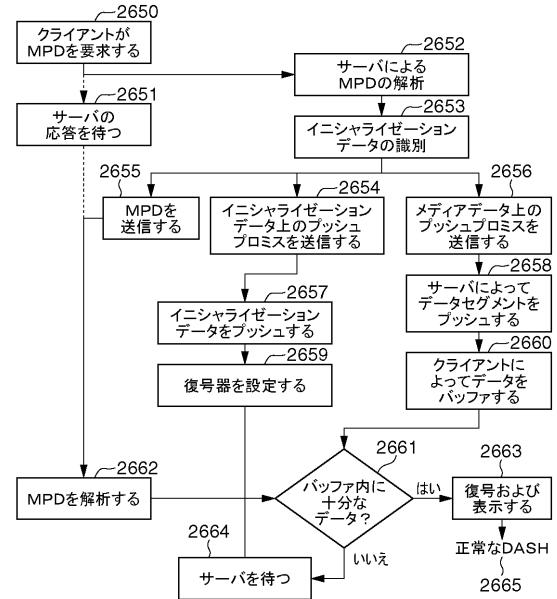
<MPD ...>
<Period id='1'>
<BaseURL>http://myserver.com/media/</BaseURL>
<SegmentList>
<Initialization sourceURL=< URL_SI >/>
</SegmentList>
<AdaptationSet id="AS1" mimeType='video/mp4' codecs='hev1'>
    <Representation id="R0" width='1920' height='1080' frameRate='30' ....bandwidth='256000'>
        <SegmentList duration='10'>
            <SegmentURL media='seg-full-AS1R0-1.mp4' />～2400
            <SegmentURL media='seg-full-AS1R0-2.mp4' />
            <SegmentURL media='seg-full-AS1R0-3.mp4' />
        </SegmentList>
    </Representation>
</AdaptationSet>
<Representation id="R1" width='1920' height='1080' frameRate='30' ....bandwidth='2256000'>
    <SegmentList duration='10'>
        <SegmentURL media='seg-full-AS1R1-1.mp4' />
        <SegmentURL media='seg-full-AS1R1-2.mp4' />
        <SegmentURL media='seg-full-AS1R1-3.mp4' />
    ...</SegmentList>
    </Representation>
</Adaptation Set id="AS2" mimeType='video/mp4' codecs='hev1'>
    <Representation id="R1" width='2000' height='1000' bandwidth='512000'>
        <SegmentList>
        ...
        </SegmentList>
    </Representation>
</AdaptationSet>
</Period>
...
</Period>
</MPD>

```

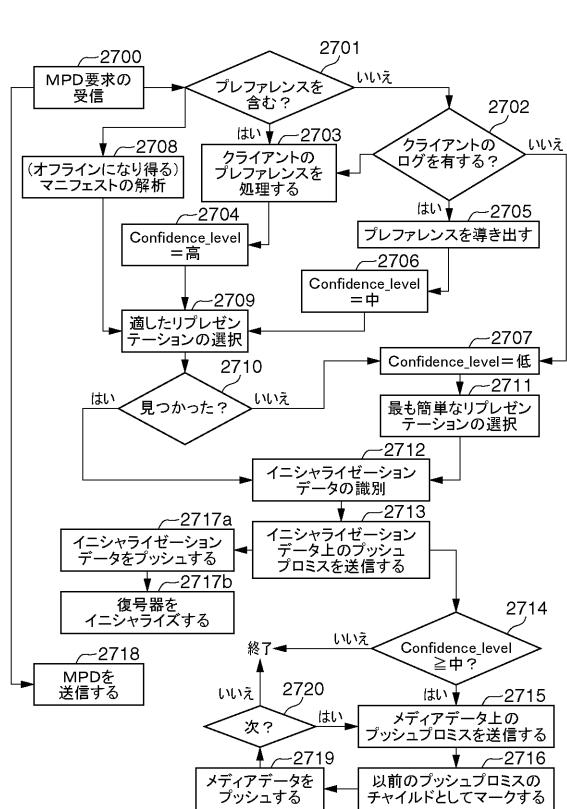
【図25】



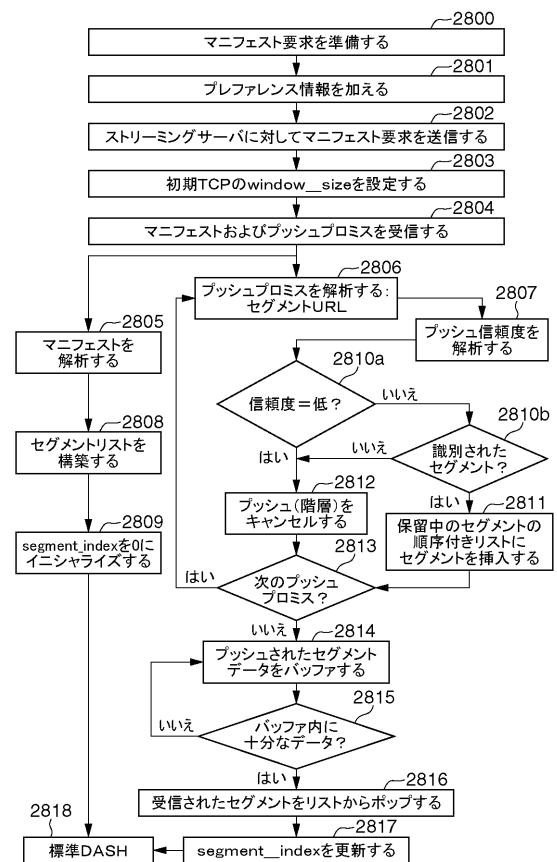
【図26】



【図27】



【図28】



フロントページの続き

(31)優先権主張番号 1410540.7

(32)優先日 平成26年6月12日(2014.6.12)

(33)優先権主張国 英国(GB)

(72)発明者 ロマン ベルソール

フランス国 レンヌ - アタラント , セデックス セッソン - セヴィニエ 35517 , リュ ド
ウ ラ トゥッシュ - ランペール キヤノン リサーチセンター フランス エス. エー. エ
ス. 内

(72)発明者 フレデリック マゼ

フランス国 レンヌ - アタラント , セデックス セッソン - セヴィニエ 35517 , リュ ド
ウ ラ トゥッシュ - ランペール キヤノン リサーチセンター フランス エス. エー. エ
ス. 内

(72)発明者 ナエル ウエドラオゴ

フランス国 レンヌ - アタラント , セデックス セッソン - セヴィニエ 35517 , リュ ド
ウ ラ トゥッシュ - ランペール キヤノン リサーチセンター フランス エス. エー. エ
ス. 内

(72)発明者 フランク ドゥヌアル

フランス国 レンヌ - アタラント , セデックス セッソン - セヴィニエ 35517 , リュ ド
ウ ラ トゥッシュ - ランペール キヤノン リサーチセンター フランス エス. エー. エ
ス. 内

(72)発明者 エルヴェ ルーラン

フランス国 レンヌ - アタラント , セデックス セッソン - セヴィニエ 35517 , リュ ド
ウ ラ トゥッシュ - ランペール キヤノン リサーチセンター フランス エス. エー. エ
ス. 内

審査官 後藤 嘉宏

(56)参考文献 特表2014-533003(JP,A)

欧州特許出願公開第02487609(EP,A1)

国際公開第2013/098319(WO,A1)

特表2009-522897(JP,A)

M. BELSHE , HYPERTEXT TRANSFER PROTOCOL VERSION 2.0; DRAFT-UNICORN-HTTPBIS-HTTP2-00.TXT
, IETF STANDARD-WORKING-DRAFT , スイス , IETF , 2013年 7月 2日 , PAGE(S):1 - 52

M. BELSHE , SPDY PROTOCOL; DRAFT-IETF-HTTPBIS-HTTP2-00.TXT , IETF STANDARD-WORKING-DRAFT
, スイス , IETF , 2012年11月28日 , PAGE(S):1 - 46

(58)調査した分野(Int.Cl., DB名)

H04N 21/00 - 21/858