

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第6054427号
(P6054427)

(45) 発行日 平成28年12月27日 (2016.12.27)

(24) 登録日 平成28年12月9日 (2016.12.9)

(51) Int. Cl. F I
HO 4 N 21/437 (2011.01) HO 4 N 21/437
HO 4 N 21/433 (2011.01) HO 4 N 21/433
HO 4 N 21/442 (2011.01) HO 4 N 21/442

請求項の数 15 (全 71 頁)

(21) 出願番号	特願2014-558949 (P2014-558949)	(73) 特許権者	595020643
(86) (22) 出願日	平成25年2月26日 (2013.2.26)		クアアルコム・インコーポレイテッド
(65) 公表番号	特表2015-511783 (P2015-511783A)		QUALCOMM INCORPORATED
(43) 公表日	平成27年4月20日 (2015.4.20)		ED
(86) 国際出願番号	PCT/US2013/027809		アメリカ合衆国、カリフォルニア州 92
(87) 国際公開番号	W02013/130474		121-1714、サン・ディエゴ、モア
(87) 国際公開日	平成25年9月6日 (2013.9.6)		ハウス・ドライブ 5775
審査請求日	平成28年2月2日 (2016.2.2)	(74) 代理人	100108855
(31) 優先権主張番号	61/603,569		弁理士 蔵田 昌俊
(32) 優先日	平成24年2月27日 (2012.2.27)	(74) 代理人	100109830
(33) 優先権主張国	米国 (US)		弁理士 福原 淑弘
(31) 優先権主張番号	13/745,809	(74) 代理人	100103034
(32) 優先日	平成25年1月20日 (2013.1.20)		弁理士 野河 信久
(33) 優先権主張国	米国 (US)	(74) 代理人	100075672
早期審査対象出願			弁理士 峰 隆司

最終頁に続く

(54) 【発明の名称】 プレイバックレート選択を伴う改良されたDASHクライアントおよび受信機

(57) 【特許請求の範囲】

【請求項1】

受信機のプレゼンテーション要素を使用してプレイアウトするためのメディアを受信する前記受信機によって実行される方法であって、前記プレイアウトの結果としてメディアがプレゼンテーションバッファから1つまたは複数のプレイバックレートで消費され、前記受信機が複数のプレイバックレートから選択するように構成され、前記方法は、

前記プレゼンテーションバッファを監視することであって、前記プレゼンテーションバッファが、少なくとも、メディアデータが受信される時間と、前記受信機に関連付けられたプレゼンテーション要素によって前記メディアデータが消費される時間との間、前記メディアデータを記憶する、監視することと、

バッファレベルのインジケーションを記憶することであって、前記バッファレベルが、前記プレゼンテーションバッファのうちどの程度が、受信されたが前記プレゼンテーション要素によってまだ消費されていないメディアデータによって占有されるかに対応する、記憶することと、

推定ダウンロードレートを決定することと、

目標プレイバックレートを計算するために、前記記憶されたインジケーションおよび前記推定ダウンロードレートを使用することと、

前記目標プレイバックレートに従って、前記複数のプレイバックレートの中からプレイバックレートを選択することと、

を備え、

前記選択されたプレイバックレートが、前記推定ダウンロードレートの乗数以下であり、前記乗数が、前記バッファレベルの増加関数を含み、前記乗数が、前記バッファレベルが増加するとき、増加する、

方法。

【請求項 2】

前記乗数が、前記バッファレベルのアフィン一次関数である、請求項 1 に記載の方法。

【請求項 3】

前記プレゼンテーションバッファの前記バッファレベルがしきい量未満のとき、前記乗数が 1 未満である、請求項 1 に記載の方法。

【請求項 4】

前記プレゼンテーションバッファ中の前記メディアデータのプレゼンテーション持続時間が、あらかじめ設定された最大量のプレゼンテーション時間以上であるとき、前記乗数が 1 以上である、請求項 1 に記載の方法。

【請求項 5】

前記乗数が、前記バッファレベルの区分的一次関数である、請求項 1 に記載の方法。

【請求項 6】

前記乗数が、レート変化に対する反応時間の推定値で除算した前記バッファレベルの増加関数である、請求項 1 に記載の方法。

【請求項 7】

前記反応時間の前記推定値が、前記メディアデータ中の切替え点の間のプレゼンテーション時間に対する上限である、請求項 6 に記載の方法。

【請求項 8】

前記反応時間の前記推定値が、前記メディアデータ中の切替え点の間の平均プレゼンテーション時間である、請求項 6 に記載の方法。

【請求項 9】

前記反応時間の前記推定値が、所定の定数に推定ラウンドトリップ時間（「ERTT」）を乗じたもの以上である、請求項 6 に記載の方法。

【請求項 10】

受信機のプレゼンテーション要素を使用してプレイアウトするためのメディアを受信し、1 つまたは複数のプレイバックレートでメディアデータを消費する受信機であって、複数のプレイバックレートのうち 1 つでのプレイバックを提供するプレゼンテーションインターフェースと、

前記プレゼンテーションインターフェースに結合された、少なくとも、メディアデータが受信される時間と、前記受信機に関連付けられたプレゼンテーション要素によって前記メディアデータが消費される時間との間、前記メディアデータを記憶するプレゼンテーションバッファと、

バッファレベルのインジケーションを含む、プレゼンテーションバッファ容量に関連した変数のためのストレージであって、前記バッファレベルが、前記プレゼンテーションバッファのうちどの程度が、受信されたが前記プレゼンテーション要素によってまだ消費されていないメディアデータによって占有されるかに対応する、ストレージと、

推定ダウンロードレートを決定する推定ダウンロードレート決定器と、

目標プレイバックレートを計算するために前記記憶されたインジケーションおよび前記推定ダウンロードレートを使用する、決定された選択されたプレイバックレートに従って要求をアレンジするための論理と、

を備え、

前記選択されたプレイバックレートが、前記推定ダウンロードレートの乗数以下であり、前記乗数が、前記バッファレベルの増加関数を含み、前記乗数が、前記バッファレベルが増加するとき、増加する、

受信機。

【請求項 11】

10

20

30

40

50

前記乗数が、前記プレゼンテーションバッファ中の前記メディアデータのプレイバック持続時間のアフィン一次関数である、請求項 10 に記載の受信機。

【請求項 12】

前記プレゼンテーションバッファの前記バッファレベルがしきい量未満のとき、前記乗数が 1 未満である、請求項 11 に記載の受信機。

【請求項 13】

前記乗数が、レート変化に対する反応時間の推定値で除算した前記バッファレベルの増加関数である、請求項 10 に記載の受信機。

【請求項 14】

前記反応時間の前記推定値が、前記メディアデータ中の切替え点の間のプレゼンテーション時間に対する上限、または、前記メディアデータ中の切替え点の間の平均プレゼンテーション時間である、請求項 13 に記載の受信機。

10

【請求項 15】

前記反応時間の前記推定値が、所定の定数に推定ラウンドトリップ時間（「ERTT」）を乗じたもの以上である、請求項 13 に記載の受信機。

【発明の詳細な説明】

【技術分野】

【0001】

関連出願の相互参照

[0001]本出願は、参照によってその内容全体がすべての目的のために本明細書に組み込まれている、2012年2月27日に出願された、「Improved DASH Client and Receiver with Rate Adaptation and Downloading for Adaptive Video」と題する米国仮出願第 61/603,569 号の利益を主張する。

20

【背景技術】

【0002】

[0002]DASHは、「動的適応ストリーミングオーバーHTTP（Dynamic Adaptive Streaming over HTTP）」を指す。DASHを使用して、コンテンツプロバイダは、コンテンツを、MPDファイルなど、関連付けられたメタデータとともに、セグメント、フラグメント、リプレゼンテーション（representations）、アダプテーション（adaptations）などにフォーマットし、それらすべてを、標準HTTPサーバまたは特殊化HTTPサーバを経由して、利用可能なファイルとして記憶する。DASHクライアントとは、これらのファイルを必要に応じて取得して、DASHクライアントのユーザにプレゼンテーションを提示する受信機である。

30

【0003】

[0003]ユーザが通常、ネットワークが制限されている環境において、事前告知がほとんどまたはまったくなしで高品質ストリーミングを望むので、DASHクライアントは、厳しい制約を有する。したがって、改良されたDASHクライアントが望まれる。

【発明の概要】

【0004】

[0004]クライアントデバイスは、ストリーミングメディアを提示し、ストリームを制御するためのストリームマネージャと、コンテンツについてのネットワーク要求を行うための要求アクセラレータと、どの要求を行うか決定するための、ストリームマネージャおよび要求アクセラレータに結合されたソースコンポーネントと、ネットワーク接続と、メディアプレーヤとを含む。要求アクセラレータは、要求をバッファリングするための要求データバッファと、アクセラレータが応答することができる各要求に完了応答を戻すための論理とを備える。ストリームマネージャ、要求アクセラレータ、およびソースコンポーネントは、プロセッサ命令またはプログラムコードとして実装されることができ、クライアントデバイスは、プログラムメモリと、ワーキングメモリと、プロセッサと、電力源とをさらに備える。クライアントデバイスはまた、ディスプレイとユーザ入力デバイスとを含み得る。クライアントタスクは、効率的にデータをストリーミングするために、ソースコ

40

50

ンポーネント、ストリームマネージャ、および要求アクセラレータの間でパースされる。

【 0 0 0 5 】

[0005]様々な態様において、本明細書に記載するように、クライアントは、リプレゼンテーションを維持し、または別のリプレゼンテーションに切り替えるべきときを決定するなどの動作を実施し、どのフラグメントを要求するか決定し、メディアプレーヤが、ほとんどの条件において、失速(stalling)することなくストリームを継続するのに十分なデータを確実に取得できるようにすることができる。

【 0 0 0 6 】

レート選択プロセスは、(a)バッファが低レベルにあるときにバッファが満たされ、(b)ダウンロードレート変動が観測される場合でも、バッファを使用して、不規則に変化するレートを回避し、(c)安定レートシナリオでは、正しい安定レートを迅速に選ぶように、レート決定を行うことができる。(a)正確なレート推定を可能にし、(b)ネットワーク遅延およびパケット損失レートが高い場合であってもリンク容量を達成することが可能であり、(c)ストリームの適時配信を達成し、(d)短期変動がほとんどない、比較的安定なダウンロードレートを達成する、HTTPのためのマルチメディアダウンロード戦略が使用される。これを達成するために、受信機は、マルチプルなHTTP接続を使用し、メディア要求をより小さいチャック要求に分解し、TCPフロー制御メカニズムを使用して接続を同期させ、データをバースト的に要求することができる。さらに、受信機は、接続をビジーに保つために、HTTPパイプラインングプロセスを使用することもできる。

【 0 0 0 7 】

受信機のプレゼンテーション要素を使用してプレイアウトするためのメディアを受信する受信機の一実施形態では、プレイアウトにより、メディアがプレゼンテーションバッファからあるプレイバックレートで消費されることになり、受信機は、複数のプレイバックレートから選択を行うように構成され、プレイバックレートを選択する方法、受信機は、少なくとも、メディアデータが受信される時間と、受信機に関連付けられたプレゼンテーション要素によってメディアデータが消費される時間との間、そのメディアデータをプレゼンテーションバッファを監視する。受信機は、また、バッファレベルのインジケーションを記憶し、ここで、バッファレベルは、プレゼンテーションバッファのどの程度が、受信されたがプレゼンテーション要素によってまだ消費されていないメディアデータによって占有されるかに対応し、受信機は、また、推定ダウンロードレートを決定し、目標プレイバックレートを計算するために、記憶されたインジケーションおよび推定ダウンロードレートを使用し、目標プレイバックレートに従って複数のプレイバックレートの中から選択する。

【 0 0 0 8 】

選択されたプレイバックレートは、推定ダウンロードレートの所定の乗数以下であるプレイバックレートであることができ、所定の乗数は、バッファレベルの増加関数である。所定の乗数は、プレゼンテーションバッファ中のメディアデータのプレイバック持続時間のアフィン一次関数であることができ、および/またはプレゼンテーションバッファ中のメディアデータのプレゼンテーション持続時間があらかじめ設定された最大のプレゼンテーション時間以上であるときには1以上であることができる。所定の乗数は、プレゼンテーションバッファ中のメディアデータのプレイバック持続時間の区分的一次関数であることができる。選択されたプレイバックレートは、推定ダウンロードレートの所定の乗数以下であるプレイバックレートであることができ、所定の乗数は、プレゼンテーションバッファ中のメディアデータのバイト数の増加関数である。プレイバックレートは、比例因子に推定ダウンロードレートを乗じたもの以下である複数のプレイバックレートのうち最も大きい利用可能プレイバックレートであることができ、比例因子は、プレゼンテーションバッファ中のメディアデータのプレイバック持続時間を、レート変化に対する反応時間の推定値で除算したものの増加関数である。

10

20

30

40

50

【0009】

反応時間の推定値は、メディアデータ中の切替え点の間のプレゼンテーション時間に対する上限、および/またはメディアデータ中の切替え点の間のプレゼンテーション時間に関する平均、および/または所定の定数に推定ラウンドトリップ時間(「ERTT」)を乗じたもの以上であり得る。

【0010】

受信機は、また、バッファレベルの許容分散を決定し、目標プレイバックレートを計算するために、バッファレベルの記憶されたインジケーションおよびバッファレベルの許容分散を使用し、目標プレイバックレートにしたがって複数のプレイバックレートの中から選択し得る。

10

【0011】

プレイバックレートは、上位比例因子(upper proportional factor)、下位比例因子(lower proportional factor)、ダウンロードレート推定値、現在のプレイバックレート、バッファレベル、およびレート変化に対する反応時間の推定値に基づいて選択され得る。上位比例因子および下位比例因子は両方とも、プレゼンテーションバッファ中のメディアデータのプレイバック持続時間を、レート変化に対する反応時間の推定値で除算したものの増加関数および/または区分的一次関数であり得る。上位比例因子は、下位比例因子よりも大きいものまたは下位比例因子に等しいものであり得る。プレイバックレートは、前のプレイバックレートが、下位比例因子に推定ダウンロードレートを乗じたものと、上位比例因子にダウンロードレート推定値を乗じたものとの間にあるときには、前のプレイバックレートと同じになるように選択され得る。プレイバックレートは、前のプレイバックレートが上位比例因子に推定ダウンロードレートを乗じたものを上回るときには、上位比例因子に推定ダウンロードレートを乗じたもの以下の最も大きい利用可能プレイバックレート、および/または、前のプレイバックレートが下位比例因子にダウンロードレート推定値を乗じたものを下回るときには、下位比例因子に推定ダウンロードレートを乗じたもの以下の最も大きい利用可能プレイバックレートになるように、選択され得る。

20

【0012】

一実施形態では、受信機は、受信機のプレゼンテーション要素を使用してプレイアウトするためのメディアを受信し、あるプレイバックレートでメディアデータを消費し、また受信機は、複数のプレイバックレートのうち1つでのプレイバックをもたらずプレゼンテーションインターフェースと、プレゼンテーションインターフェースに結合された、少なくとも、メディアデータが受信される時間と、受信機に関連付けられたプレゼンテーション要素によってメディアデータが消費される時間との間、メディアデータを記憶するプレゼンテーションバッファと、バッファレベルのインジケーションを含む、プレゼンテーションバッファ容量に関連した変数のためのストレージであって、バッファレベルが、プレゼンテーションバッファのうちどの程度が、受信されたがプレゼンテーション要素によってまだ消費されていないメディアデータによって占有されるかに対応する、ストレージと、推定ダウンロードレート決定器と、目標プレイバックレートを計算するために記憶されたインジケーションおよび推定ダウンロードレートを使用する、決定された選択プレイバックレートに従って要求をアレンジするための論理と、を含む。

30

40

【0013】

[0012]様々な要素は、ネットワーク経路によって結合されたソースと受信機との間のネットワーク経路を介したデータダウンロードを制御するためのプロセッサによって実行されるコンピュータ可読媒体を使用して実装されることもできる。コンピュータ可読媒体は、非一時的コンピュータ可読媒体でありうる。

【0014】

[0013]本発明の他の態様が、本記述から明らかになるはずである。

【図面の簡単な説明】

【0015】

【図1】[0014]記録段階と、コンテンツ準備段階と、コンテンツ配信段階とを伴う、メデ

50

ィア記録がどのようにエンドユーザに到達するかを表示する、DASH展開におけるDASHクライアントを含む様々な要素を示す図。

【図2】[0015]ストリームマネージャ、要求アクセラレータ、ソースコンポーネント、ネットワーク接続、およびメディアプレーヤを含む様々なコンポーネントを備えたDASHクライアントの例示的アーキテクチャを示す図。

【図3】[0016]リプレゼンテーション切替えプロセスを示すタイミング図であり、図3Aは後方参照プロセスについて、図3Bは前方参照プロセスについてのタイミング図。

【図4】[0017]切替え点のアライメントされている場合のリプレゼンテーション切替えプロセスを示すタイミング図。

【図5】[0018]レートエスティメータ、特に、バッファレベルに適合したエスティメータ（pkerタイプのレートエスティメータなど）によって管理される経時レートを示すプロット。

10

【図6】[0019]非適応指数加重移動平均（「EWMA」）フィルタが使用されるとき、ダウンロード時間（r時間）に対するレート増大を示すプロット。

【図7】[0020]非適応EWMAフィルタが使用されるとき、プレイバック時間（p時間）に対するレート増大を示すプロット。

【図8】[0021]可変ウィンドウサイズ加重移動平均（「WMA」）フィルタが使用されるとき、ダウンロード時間（r時間）に対するレート増大を示すプロット。

【図9】[0022]pkerタイプのプロセスが使用されるとき、プレイバック時間（p時間）に対するレート増大を示すプロット。

20

【図10】[0023]セクション2.1のpkerプロセスが使用されるとき、ダウンロード時間に対するレート低下を示すプロット。

【図11】[0024]レートの急上昇に対するpkerプロセスの挙動を示す図。

【図12】[0025]急なレート下落に対するpkerプロセスの挙動を示す図。

【図13】[0026]単純（固定幅）移動ウィンドウ平均と指数加重移動平均との比較を示す図。

【図14】[0027]pkerレート推定プロセスのフローチャート。

【図15】[0028]図16とともに、pkerプロセスによって使用される値Bおよび T_{fast} が、記録された（ T_p , T_r ）値の履歴からどのように決定され得るかを示す図。

【図16】[0029]値を決定する態様を示す図。

30

【図17】[0030]「ウォーターマーク」フェッチングプロセスの挙動を示す図。

【図18】[0031]プレイバックレートを選択するのに使用され得るラムダ関数およびミュー関数の例を示す図。

【図19】[0032]「コンサバティブ」設定を使用する、（ラムダ, ミュー）関数の例示的選択を示す図。

【図20】[0033]「中間」設定を使用する、（ラムダ, ミュー）関数の例示的選択を示す図。

【図21】[0034]「アグレッシブ」設定を使用する、（ラムダ, ミュー）関数の例示的選択を示す図。

【図22】[0035]MLBプロセスをある程度までエミュレートするためのプロセスを使用する、（ラムダ, ミュー）関数の例示的選択を示す図。

40

【図23】[0036]ラムダ設定用の隣り合っている値の例を示す図。

【図24】[0037]ミュー設定用の隣り合っている値の例を示す図。

【図25】[0038]レート推定、次いでレートベースのレート選択、次いでバッファ管理ベースのレート選択のためのプロセスを示す図。

【図26】[0039]要求取消しなしでのレート下落を示す図。

【図27】[0040]要求取消しのあるレート下落を示す図。

【図28】[0041]例示的な要求取消しプロセスを示すフローチャート。

【図29】[0042]要求取消し検出のためのプロセスを示す図。

【図30】[0043]マルチプルなTCP接続を用いるが、受信バッファのチューニングは用

50

いないでフェッチする挙動のプロット。

【図3 1】[0044]マルチプルなTCP接続を用いて、および受信バッファのチューニングを用いてフェッチする他の挙動のプロット。

【図3 2】[0045]例示的な要求アクセラレータプロセスのフローチャート。

【図3 3】[0046]所与のフラグメント要求を生み出すためのいくつかのサブ要求を見つけるためのプロセスを示す図。

【図3 4】[0047]計算されたサイズを有するソース要求の分離区間となるように選ばれた個々の要求を選択するためのプロセスを示す図。

【図3 5】[0048]時間オフセットと、時間オフセットによって決まる修復セグメントについてのフラグメント構造との例を示す図。

【図3 6】[0049]レート選択におけるラムダおよびミューに使用することができる値のテーブル。

【詳細な説明】

【0016】

[0050]本明細書で説明されるDASHクライアントは、図2に示すように、ストリームマネージャ(SM)と、要求アクセラレータ(RA)と、ソースコンポーネント(SC)と、ネットワーク接続と、メディアプレーヤを含む。DASHクライアントは、1つまたは複数のメディアデータバッファも含み得る。いくつかの実装形態では、RA、SCおよびメディアプレーヤはすべて、それら自体のデータバッファ、または1つの大きいデータバッファの論理区画を有し得る。他の実装形態では、おそらくRAのみが、要求をバッファリングするためのデータバッファを有し、その結果、RAは、それが応答し得るあらゆる要求に対して完了応答(complete response)を戻すことができるようになり、メディアプレーヤは、SCがセットアップしたデータバッファはどれでも使用する。SMは、決定を行うのに必要とされるメタデータを記憶するための、それ自体の(物理的または論理的)ローカルストレージを有し得る。

【0017】

[0051]図1は、DASHクライアントをもつDASH展開を示す。

【0018】

[0052]図2は、様々なコンポーネントをもつDASHクライアントの例示的アーキテクチャを示す。SM、RA、SCおよびメディアプレーヤは、ハードウェア、ソフトウェアまたは何らかの組合せで実装され得ることを理解されたい。したがって、機能がコンポーネントによるものである場合、機能は、プロセッサ命令、プログラムコードなどとして実装されてよく、この場合、それらの命令を実行するための必要なハードウェア(プログラムメモリ、ROM、RAM、プロセッサ、電力源、コネクタ、回路板など)が含意される。ネットワーク機能が記述される場合、ネットワーク接続は、存在するものと理解されるべきであり、ワイヤード、光学、ワイヤレスなどであってよく、ユーザインタラクションが含意される場合、ユーザインターフェイス能力(ディスプレイ、キーボード、タッチパッド、スピーカ、マイクロフォンなど)も含意される。

【0019】

[0053]DASHクライアントは、2つのクロック、またはそれらの論理的等価物を維持する。一方のクロックは、クライアント中で稼動するローカルクロックの時間を示すリアルタイムクロック回路またはソフトウェアであり、他方のクロックは、メディアコンテンツの開始に対する、メディアコンテンツのプレゼンテーションの時間を表すプレゼンテーション時間である。本明細書において、リアルタイムクロック時間は「r時間」と呼ばれ、「p時間」は、プレゼンテーション時間を示す記述子である。

【0020】

[0054]リプレゼンテーションとは、同じコンテンツについての、異なるビットレートまたは他の違いで符号化されたメディアストリームである。したがって、ユーザは通常、1つのリプレゼンテーションを必要とするだけであるが、クライアントは、条件および/または要件が変わると、あるリプレゼンテーションから別のリプレゼンテーションに切り替

10

20

30

40

50

えうる。たとえば、帯域幅が高い場合、ストリーミングクライアントは、高品質、高ビットレートのリプレゼンテーションを選びうる。帯域幅が低減された場合、クライアントは、より低品質のより低いビットレートリプレゼンテーションに切り替えることによって、これらの条件に適應することができる。

【0021】

[0055]切替え点(またはランダムアクセスポイント)は、ストリームに先行するデータを知る必要なしに、メディアサンプルの復号がそこから始まり得る、リプレゼンテーション中のサンプルである。特に、ビデオリプレゼンテーションでは、サンプル(フレーム)は概して以前のフレームに依存するので、あらゆるサンプルがランダムアクセスポイントであるわけではない。ストリーミングクライアントは、リプレゼンテーションを切り替えたいときには、無駄な作業を避けるために、必ず切替え点において新たなリプレゼンテーションを復号し始めるべきである。いくつかのケースでは、切替え点は、ストリーミングクライアントに対してセグメントインデックス(sidex)中でシグナリングされる。

10

【0022】

[0056]リプレゼンテーショングループ(単にグループと略記されることがある)とは、切替え可能なリプレゼンテーションのセットである。メディアプレゼンテーションは、1つよりも多くのリプレゼンテーショングループを含み得る。メディアプレゼンテーションは、たとえば、様々なビットレートでのビデオリプレゼンテーションについて1つのリプレゼンテーショングループと、オーディオビットレートについて別のリプレゼンテーショングループとを有し得る。DASH規格では、リプレゼンテーショングループは、適応セットと呼ばれることもある。

20

【0023】

[0057]セグメントとは、複数のリプレゼンテーションのうち1つのリプレゼンテーションの少なくとも一部分についてのメディアデータを含むファイルである。フラグメントとは、フラグメントの開始p時間からセグメント内のフラグメントのバイト範囲へのマッピングが利用可能である、セグメントの一部である。フラグメントの代わりにサブセグメントという用語が使用されることもあり、これらは等価であると見なされ得る。一部のメディアコンテンツは、フラグメントに分割されず、そのようなケースにおいて、「フラグメント」は、セグメント自体を指す場合がある。

【0024】

[0058]図3は、2つの可能なリプレゼンテーション切替えプロセスを示すタイミング図である。切替えは、後方参照(backward looking)(図3Aの第1のプロセス)であることができ、この場合、switch-fromリプレゼンテーション中ですでに要求されているp時間ストレッチを見ることによって、およびこのストレッチの終了に最も近い、switch-toリプレゼンテーションから、p時間において後退する前の切替え点を選ぶことによって、switch-toリプレゼンテーションにおける切替え点が見つかる。第2のプロセス(図3B)は前方参照(forward looking)であり、switch-fromリプレゼンテーション中の、最後の要求されるp時間から始まる、switch-toリプレゼンテーション中の、p時間において前進する次の切替え点を見つける。

30

【0025】

[0059]図4は、切替え点のアライメントされたとき、および切替え点が最後の要求フラグメントの後にすぐに続くときの、切り替えるためのプロセスを示すタイミング図である。この図は、前方参照方法と後方参照方法の両方の挙動を示すが、それは、これらの2つのプロセスが、そのような設定において同一に振る舞うからである。したがって、切替え点のアライメントされるときには、どちらのプロセスも、重複データをダウンロードする必要はない。

40

【0026】

[0060]プレゼンテーション時間とは、メディアが、一般には通常速度で、プレイアウトまたはプレイバックすると予想される時間期間である。たとえば、30分のビデオプレゼンテーションであれば、30分間再生する。ユーザは、早送りまたは巻き戻しをする場合

50

があるので、実際にかかる時間は変わるが、プレゼンテーションは依然として、30分のビデオプレゼンテーションであることを理解されたい。プレゼンテーション要素が、プレゼンテーション時間に、ユーザにプレゼンテーションを提示する。プレゼンテーション要素の例としては、視覚ディスプレイおよびオーディオディスプレイ、またはストリームを提示することができるデバイスにパイピングされるビデオ/オーディオストリームがある。「プレイバック」とは、メディアの消費を記述するのに使用される用語である。たとえば、スマートフォンであれば、プレゼンテーションのプレゼンテーション時間(p時間)にわたってプレゼンテーションを表すメディアデータをダウンロードまたは取得し、バッファリングすることができ、メディアプレーヤは、そのメディアを「消費する」と言われ、好ましくは、バッファが少なくともプレゼンテーション時間の終了までは完全には空にならないように消費し、そうすることによってユーザは、受信機がより多くのデータを取得するのを待っている間、プレゼンテーションの失速を受けない。当然ながら、「プレイバック」または「プレイアウト」は、メディアが一度よりも多く再生されることは含意しない。多くの事例において、メディアは一度消費されると、再度使用されることはない。

【0027】

[0061]プレゼンテーションバッファとは、受信機、メディアプレーヤ内にあるか、または一方もしくは両方にとってアクセス可能なメモリ要素である。説明を簡単にするために、「プレゼンテーションバッファ」、「バッファ」、「メディアバッファ」および「プレイバックバッファ」という用語を互換的に使用するが、これは、ダウンロードされているがまだプレイアウト、すなわち消費されていないデータ、すなわち通常はメディアデータを備える論理バッファであることは理解している。プレゼンテーションバッファを成すデータが、デバイスにおいて、様々なコンポーネントの間で区分される場合があり、すなわち、ダウンロードされたデータのいくつかの部分が、1つのプロセス、たとえば、デバイス内の受信プロセスによって保持され、他の部分は、別のプロセス、たとえば、デバイス内のプレイアウトプロセスにすでに渡されている場合がある。プレゼンテーションバッファを成すデータの少なくとも一部は、異なるプロセスの異なるバッファにわたって少なくとも部分的に複製される場合もあり得る。いくつかのケースでは、ダウンロードされているがまだプレイアウトされていないデータすべてが、依然としてプレゼンテーションバッファ内にあると見なされるわけではなく、たとえば、いくつかのケースでは、メディアコンテンツは、メディアプレーヤにパスされると、もはやプレゼンテーションバッファにあると見なすことはできない。概して、メディアデータが存在する場合、ダウンロードされているがまだプレイアウトされておらず、さらにプレゼンテーションバッファ内にあると見なされないメディアデータの量は、非常に少ない。

【0028】

[0062]プレゼンテーションバッファは、メディアを受信しプレイバックし、受信メディアデータを、消費されるまで記憶する際の不均一に適應する。メディアデータは、消費された後、構成に応じて、消去されることもでき、または記憶され続けることもできる。いくつかの実装形態では、プレゼンテーションバッファのサイズ(プレゼンテーションバッファに記憶することができるデータバイトの数で測ることができる)は、時間とともに変わり得る。たとえば、プレゼンテーションバッファは、必要に応じて共有メモリから動的に割り振られ得る。

【0029】

[0063]本明細書において詳細に説明する多くの例では、プレゼンテーションバッファは、サイズによって特徴付けられると仮定され得る。プレゼンテーションバッファに専用の固定メモリサイズのケースでは、そのサイズは、利用可能メモリに記憶することができるバイトの数で測ることができる。プレゼンテーションバッファが動的に割り振られる場合、プレゼンテーションバッファに起因する「サイズ」は、プレゼンテーションバッファに現時点で割り振られているバイトの数、プレゼンテーションバッファに割り振ることができ得るバイトの最大数、または何らかの他の適切な測度に等しいものであり得る。プレゼンテーションバッファサイズは、プレゼンテーションバッファ中の現在利用可能なメディ

10

20

30

40

50

アのプレゼンテーション時間ブレイアウト持続期間によって測定されることもある。

【 0 0 3 0 】

[0064] プレゼンテーションバッファは、別の特性、すなわちバッファの「レベル」または「充満レベル」も有する。プレゼンテーションバッファのレベルは、どれだけ多くの未消費メディアデータがプレゼンテーションバッファ中に存在するかを表し、たとえばバイトまたはプレゼンテーション持続時間で測定される。レベルは、メディアデータが受信されると上がり、メディアデータが消費されると下がると予想される。レベルは、論理的なものにすぎない可能性があり、たとえば、プレゼンテーションバッファは、メディアデータで常に満杯（full）であり得るが、メディアの一部、たとえば、すでに消費されたメディアデータは、新規メディアデータが受信されると、上書きのためにマーキングされる。いくつかの受信機は、「空きバッファ」が、未消費のメディアデータがゼロであるコンディションであり、「満杯バッファ」が、プレゼンテーションバッファの100%が未消費メディアデータで満たされているコンディションであるようにプログラムされ得る。他の受信機は、プレゼンテーションバッファサイズの0%～100%よりも小さい範囲にレベルが及ぶように、他の限度を有し得る。未消費メディアデータがバッファに記憶されたときに共有メモリが使われ、プレゼンテーションバッファしか割り振られたことがないケースでは、プレゼンテーションバッファは、定義により、常に満杯になるので、プレゼンテーションバッファのメモリの動的に割り振られたサイズを、レベル比を示すときの水準として使用するのとは意味がない場合がある。そうではなく、プレゼンテーションバッファのレベルは、プレゼンテーションバッファ中の未消費メディアデータの量を、プレゼンテーションバッファについての最大許容サイズで除算した比として測ることができる。

10

20

【 0 0 3 1 】

1. クライアントコンポーネントの概説

[0065] 図1～図2を再度参照すると、例示的クライアントの様々なコンポーネントが示されている。

【 0 0 3 2 】

[0066] SCは、どのようなリプレゼンテーションが利用可能であるか、およびリプレゼンテーションのフラグメントが何であるかについての情報などのメタデータを追跡する。SCは、ネットワークを介して受信されたメディアデータのバッファリング、およびメディアプレーヤへのハンドオフも担う。SMは、どのようなリプレゼンテーションがどの時点でダウンロードされるべきかを決定すること、およびレート切替え決定を行うことを担う。最終的に、RAは、SCによって与えられる正確なURLおよびバイト範囲情報を所与として、メディアフラグメントのダウンロードを担当する。

30

【 0 0 3 3 】

[0067] SMは、レート切替え決定を担うソフトウェアコンポーネントである。SMの目標の1つは、所与の状況向けの最良のコンテンツを選び出すことである。たとえば、たくさんの帯域幅が利用可能である場合、高ダウンロードレートが達成され得るので、SMは、高レトリプレゼンテーションを選び出すべきである。ダウンロードレートが大幅に下落した場合、選ばれた高いリプレゼンテーションはそれ以上持続可能でなくなるので、SMは、条件により適した、より低いリプレゼンテーションレートに切り替えるべきである。SMは、プレイバックバッファを完全にドレインするのを避け（プレイバック失速を引き起こすので）、ただし同時にあまりに急いで、またはあまりに頻りに切り替えようとはしないように、十分高速にレートを切り替えるべきである。さらに、ネットワークを通じてダウンロードし、失速することなくプレイバックすることができる、最も高品質のコンテンツを要求することを目指すべきである。SMは、意思決定プロセスにおいて、ダウンロード速度以外の要因を考慮に入れるように拡張することができる。可能性としては、リプレゼンテーション決定を行うときには、バッテリー寿命、ディスプレイサイズ、および他の要因などに配慮することができる。そのようなさらなる制約は、フィルタとしてSMに追加することができ、本明細書において記述する基本レート決定計算には影響しない。

40

【 0 0 3 4 】

50

[0068]典型的な高レベルのクライアント動作について、次に説明する。ユーザが、ライブのスポーツ放送、あらかじめ録画された映画、オーディオストリーム、または、ビデオおよびオーディオ以外のメディアタイプを伴い得る他のオーディオビジュアルもしくは他のコンテンツなど、特定のメディアコンテンツを要求すると想定する。クライアントは、その要求を、おそらくユーザインターフェースまたはコンピュータインターフェースを通して、SMに供給する。SMは、SCに対して要求を行い、どのリプレゼンテーションが利用可能であるか、どのp時間帯がどのフラグメントによってカバーされるか、およびリプレゼンテーション中の切替え点がどこにあるかについてのインジケーションを受信する。それに加え、SMは、後で説明するように、自由にできる短期ダウンロードレートについての何らかの情報を有することができ、RAは、このデータをSCに報告し、SCは、このデータをSMに報告または提供する。

10

【0035】

[0069]SMは、その情報を、過去の履歴と一緒に使用して、持続可能レートを推定し、リプレゼンテーション内の適切な切替え点と、その切替え点において始まるそのリプレゼンテーションからダウンロードすべきメディアコンテンツの量とを選ぶ。ダウンロードが進行中であり、メディアコンテンツがプレイバックされるとき、SMは、供給された情報を使用して、レート切替えが順序通りであるかどうかを決定する。レート切替えが順序通りでない場合、SMは、現在のリプレゼンテーションからフラグメントをフェッチし続けるよう、SCに伝える。レート切替えが順序通りである場合、SMは、潜在的切替え点を見て、どのリプレゼンテーションにあるどのフラグメントが、所望の切替えを行うためにフェッチされる必要があるか決定する。SMは次いで、その情報をSCに渡す。SCとSMとの間のこの交換は、ダウンロードされるべきビデオの次のセクションに対する決定が行われるべきであるときはいつでも、定期的に行われる。良好な決定を行うために、SMはバッファレベルを監視し、いくつかのケースでは、SMは、バッファが十分に満杯であり、いかなるフラグメントも、ある程度の期間ダウンロードされる必要はないと決定し得る。

20

【0036】

[0070]SMが、ダウンロードすべきフラグメントを決定すると、SCは、RAに、実際にフラグメントをダウンロードさせ、ダウンロードされたフラグメントをメディアバッファに保たせ、最終的に、メディアデータをプレイアウトする時間になったとき、メディアバッファ中のメディアデータをメディアプレーヤにハンドオーバーさせるのを担当する。

30

【0037】

[0071]SMはもはや、SCに、ダウンロードするよう伝えたフラグメントに能動的に関与していない。ただし、SMは、所与のフラグメントのダウンロードがすでに始まった後でも、考えを変え、以前発行したフラグメント要求を取り消すことができる。この機能は、ダウンロードレートが劇的に下落し、ダウンロードされているフラグメントが、メディアバッファが完全にドレインされた時間に、利用可能である見込みがなくなったケースにおいて有用である。その条件が起きた場合、SMはその条件を検出し、要求を取り消し、代わりにより適切なレートに切り替える。

40

【0038】

[0072]SCは、SMから、フェッチすべきフラグメントハンドルを受信すると、対応するフラグメントのURLとバイト範囲とを、そのデータ構造中でルックアップし、それを使用して要求を作成し、その要求をRAにハンドオーバーする。それはまた、RAから応答データを取り出し、受信メディアフラグメントを、再生可能なストリームに変換するのも担う。最終的に、SCは、MPDから取得したデータ、セグメントインデックス(sidx)ボックス、またはAppleのHTTPライブストリーミング(HLS)のケースでは、プレイリストなどのメタデータをパースし、追跡するのを担当する。

【0039】

[0073]RAは、SCから受信したフラグメントおよびメタデータ要求を引き受け、対応するHTTP要求を作成し、それらをネットワーク接続を通じて送出し、対応する応答を

50

取り出し、S Cに返すコンポーネントである。ネットワーク接続は、インターネット接続、セルラーベースの接続、W i F i (登録商標)接続またはH T T P要求と応答とを扱うことが可能な他のネットワーク接続であることができる。ネットワーク接続は、単一デバイスの内部にあることができ、すなわち、デバイス内にすでにキャッシュされているメディアデータとの内部インターフェースであることができる。また、多くの組合せがあってもよく、すなわち、メディアコンテンツの一部は、ワイヤードインターネット接続から、一部はセルラーベースの接続を介して、一部はW i F i接続を介して、一部はローカルキャッシュからダウンロードすることができる。いくつかのケースでは、メディアデータがダウンロードされる接続は混合されてよく、すなわち、いくつかの部分はセルラーを介し、いくつかの部分はW i F iを介し、いくつかの部分はワイヤードを介するなどである。特定の要求は、いくつかの事例ではH T T P以外であってよいが、メディアコンテンツをサービスするサーバがH T T Pサーバである場合はH T T Pが好まれる。

10

【 0 0 4 0 】

[0074]その最も単純な形において、R AはH T T Pクライアントである。ただし、R Aは、一般的なH T T Pクライアントより効率的であることが望ましい場合もある。R Aの1つの目標は、十分に高いダウンロード速度を達成することであり、選択されたプレイバックメディアレートよりも大幅に速いダウンロードを目指すべきである。一方、未加工スループットのために適時性を損なわないように、慎重でもあるべきであり、すなわち、間もなくプレイアウトされるフラグメントは、さらに後にくる他のものよりも緊急であり、R Aは、それらを時間内に受信しようとするべきである。したがって、適時性のためにある程度のスループットを犠牲にすることが必要となり得る。R Aは、すべての妥当なネットワーク条件でうまく機能するように設計されるべきである。

20

【 0 0 4 1 】

[0075]R Aの基本設計は、いくつかの接続と、最良の結果を取得するために、できればさらにF E C (前方誤り訂正)も使用するものである。したがって、R Aは通常、2つ以上のオープンなH T T P接続を管理する必要がある。R Aは、それらの接続に要求をディスパッチする。R Aは、いくつかの状況において、要求を、より小さい要求のセットに分割する。対応する応答を受信すると、R Aは次いで、データをコヒーレント応答にリアセンブルする。言い換えると、R Aは、送出すべきH T T P要求の粒度と、どの接続に要求をディスパッチすべきかとを決定し、ソースフラグメントまたは修復セグメントのどの部分を要求すべきか決定する役目を果たす。それらの要求の粒度は、たとえばバッファレベル、要求の緊急度、利用可能な接続の数など、いくつかのことに依存し得る。

30

【 0 0 4 2 】

[0076]R Aによって送出される各要求は、メタデータについての、またはS CによってR Aにパスされたフラグメント要求の一部もしくは全部についてのH T T P要求である。ソースメディアデータまたはソースメディアデータから生成された修復データのいずれかについての要求でありうる。S Cフラグメント要求から生成されたR A要求への応答は、ほとんどの場合、R Aがフラグメント要求中のメディアデータすべてを再構築するのに十分なはずであり、R Aは次いで、このデータをS Cに返す。したがって、R Aは、メディアフラグメント要求に関連付けられたR A要求からの応答を、S Cに与えられるフラグメント要求への応答に逆アセンブルするのを担う。R Aによるアセンブルは、たとえばF E C修復データについてのいくつかのR A要求がある場合、F E C復号を含み得る。

40

【 0 0 4 3 】

[0077]H T T P要求の管理に加え、R Aは、短期の期間にわたる、すなわちある程度のサンプリングレートの時間スライスにわたるダウンロード速度を測定する。例示的サンプリングレートは1 0 0 m sであり、すなわち、R Aは、1 0 0 m sの期間にわたってダウンロード速度を測定する。このデータは、S Mによって、S Mのダウンロード速度推定値を計算し、最終的にはレート決定を行うのに使用される。他のサンプリングレートも可能である。

【 0 0 4 4 】

50

[0078] R A は、D A S H メディアプレゼンテーション記述 (M P D) などのメタデータについても、セグメント構造についても知る必要はない。特定の実装形態において、R A は、H T T P スタック実装形態のいくつかの同時事例を使用して、いくつかの接続にわたって、いくつかのケースでは同様または異なるサーバへの異なるタイプの接続にもわたって、H T T P 検索を実装する。

【 0 0 4 5 】

[0079] R A は、新規要求がいつ受諾され得るかを S C に知らせることを担う。S C は、S M に、要求すべき次のフラグメントを決定するよう求め、適切な要求を R A に与える。R A は、何らかのステータス情報も与える。R A は、短期ダウンロード速度と、ダウンロードに費やされる総時間とを、S C を経由して S M に定期的に与え得る。S M は、また、この情報について、S C を経由して間接的に R A をポーリングし得る。それに加え、R A は、各個々の要求がすでに完了された割合についても S M に知らせる。この情報は、S M がそれを取り出すために呼び出す A P I を用いて同様に与えられる。

10

【 0 0 4 6 】

[0080] R A と、S C と、実際のメディアパイプラインとの間には、非常にタイトなデータフローがあるはずであり、R A または S C 内にバッファリングされるデータは、(意図的なメディアバッファは別にして) 可能な限り少ない。同じことが、様々な形の H T T P 要求についても成り立ち、S M は、実際の対応する H T T P 要求がネットワークを介して送出される時よりも早く、わずかな時間量だけで、要求すべきフラグメントを決定しなければならないはずである。1 つの理由は、S M が要求を事前に決定しなければならない程、その情報は正確でも最新でもなくなり、したがって、その決定はより低い品質となる。

20

【 0 0 4 7 】

[0081] S M は、発行されるべき要求を一度に 1 つ提出する。ただし、S M は、以前のすべての要求が完了されているわけではない場合にも新規要求を発行することができ、同時要求が許容される。S C は、S M が要求を発行した順序で、要求を R A にパスする。R A は次いで、同時処理を引き受け、受信データを S C に必ず返す。

【 0 0 4 8 】

[0082] 同時要求により、R A は、H T T P パイプライン化を実装することが可能になる。実際、複数の接続を利用する R A でさえも、この方式に適合する。

30

【 0 0 4 9 】

1 . 1 . ストリームマネージャ (S M)

[0083] S M は、ユーザのアクションと、ネットワーク条件と、他の要因との組合せに応じて、フラグメントをいつ要求すべきか、およびどのフラグメントを要求すべきかを決定する。ユーザがコンテンツを見始めると決定すると、S M は、ユーザによって、または提供されるサービスによって指定された、p 時間から始まるそのコンテンツについて要求すべき第 1 のフラグメントを決定するのを担う。たとえば、いくつかのライブストリーミングサービスは、すべてのユーザが、メディアコンテンツの同じ p 時間部分を同じ r 時間に閲覧していることを要求し得るが、他のライブストリーミングおよびオンデマンドサービスは、どの p 時間をどの r 時間にプレイバックするかについて、柔軟性をエンドユーザまたはアプリケーションに認め得る。メディアバッファが満杯になると、S M は、さらなるフラグメント要求を与えるのを一時的にサスペンドする。S M は、ネットワーク条件およびたとえばディスプレイのサイズ、残存バッテリー寿命など、他の要因に依存して、p 時間中の各時点においてコンテンツをどの品質でプレイバックすべきか決定するのを担う。

40

【 0 0 5 0 】

[0084] S M が、フラグメント要求を提供するのが適切であると考えるとき、S M は、R A がフラグメント要求を受信および処理する準備ができていない場合にしか要求を提供することができない。S C は、R A をポーリングすることによって、準備ができていないときを決定し、この情報を S M にフォワーディングする。

50

【 0 0 5 1 】

[0085] R A が次の要求を受信する準備ができているときには、S M は、新規要求が発行されるべきかどうか決定し、要求するべき次のフラグメントを選ぶ。S M は、メディアデータについての要求を、一度に1つのフラグメントだけ行う。S M は、コンテンツの適時およびシームレスなプレイバックを可能にするフラグメントを要求するのを担う。リプレゼンテーション中のプレイバック変更は概して、切替え点においてのみ起こり得、2つの連続する切替え点の間に複数のフラグメントがあってもよく、S M はその制約を尊重する。

【 0 0 5 2 】

[0086] 概して、S M は、円滑なプレイバックのために時間内に受信されると信じるのが妥当なフラグメントを要求することだけを試みる。ただし、ネットワーク条件が、時には非常に急速に劇的に変化しうることを考えると、このことは、すべての状況において保証されるわけではない。したがって、S M は、要求を取り消すことも可能である。S M は、輻輳が検出されるとともに、何のアクションもとられなかった場合に失速する危険性が大きい場合、要求を取り消す。失速とは、何のアクションもとられない場合、たとえば、フラグメント要求が発行された後間もなく、ネットワーク条件の悪化により、急激に、ダウンロードレートが突然下落する場合の可能性である。

【 0 0 5 3 】

[0087] S M は、最も直近の以前選ばれたフラグメントのリプレゼンテーション、すなわち R と、終了 p 時間、すなわち E とを追跡する。S M は通常、 $E' = E$ という開始 p 時間を有する次のフラグメントを要求することを選ぶ。いくつかの変形体は、バッファレベルおよび現在のプレイバック時間から決定された開始時間を有し得る。

【 0 0 5 4 】

[0088] S M は、切替え点における潜在的重複が破棄された場合、円滑にプレイバックすることができるストリームを生成することを意図した、一連の要求を生成する。S M が要求を作り出す順序は、R A が要求を（必ずしも発行するわけではないが）優先するべき順序と同じである。この順序は、また、R A が S C に受信データを返し、S C がデータをプレイアウトするべきであるのと同じ順序でもある。

【 0 0 5 5 】

[0089] S M が、レートを切り替える必要があると決定した場合、一般的ケースでは、切替えのための2つのプロセスがある。1つのプロセスでは、S M は、E 以下の p 時間をもつ新規の（「switch-to」）リプレゼンテーション中の切替え点（「ランダムアクセスポイント」または「RAP」と呼ばれることもある）Pを探し、そのような点が識別されると、S M は、新規リプレゼンテーション中のフラグメントを要求し始める。第2のプロセスは、E のものよりも後または等しい p 時間をもつ切替え点 P を探すこと、および P を超える終了時間をもつフラグメントが要求されるまで、古い（「switch-from」）リプレゼンテーション中のフラグメントを要求し続けることのうちの1つである。いずれのケースでも、切替えを S C にシグナリングすることが有用であり得る。

【 0 0 5 6 】

[0090] これらのプロセスは両方とも、いくらかの重複データがダウンロードされなければならない可能性があるという特性を有することに留意されたい。switch-from リプレゼンテーションと switch-to リプレゼンテーションの両方についてデータがダウンロードされる必要があり得る、p 時間のストレッチがある。

【 0 0 5 7 】

[0091] これらの切替えプロセスのうちどちらが好ましいかは、状況に依存する。たとえば、ある特定の状況では、プロセスのうち一方についての重複が不当に大きく、他方については極めて短い場合がある。すべてのフラグメントが複数のリプレゼンテーションにわたってアライメントされ、すべてのフラグメントが R A P で始まる単純なケースでは、これらの切替えプロセスは、より簡単な方法に縮小し、この方法において、S M は、switch-from リプレゼンテーションではなく switch-to リプレゼンテーションに対して次のフラグメントを要求することによって切り替えるだけである。この場合、

10

20

30

40

50

どの重複データもダウンロードされる必要はないことにも留意されたい。

1.1.1. SMフラグメント決定プロセス

[0092]このセクションは、どのフラグメントを要求するようSCに伝えるべきか決定するためのSMフラグメント決定プロセスについて記載する。これらの例では、単一リプレゼンテーショングループが仮定されるが、これらの例は、複数のリプレゼンテーショングループを使用するプロセスに対処するように拡張することができ、たとえば、ビデオリプレゼンテーショングループからビデオリプレゼンテーションを、およびオーディオリプレゼンテーショングループからオーディオリプレゼンテーションを選ぶ。

【0058】

[0093]SMによって選ばれる次のフラグメントは通常、前のフラグメント要求の終了p時間である開始p時間を有する。以下で、要求するべき次のフラグメントを選ぶための、SM内に実装することができるある程度の詳細な論理について説明する。

【0059】

[0094]以下の例では、フラグメントが、RAPで始まり、リプレゼンテーションの間でアライメントされると仮定する。これが成り立たない場合、この記述の変形形態が可能である。それらの条件が存在する場合、SMのフラグメント決定は、レート決定に縮小し、すなわち、SMは、現在のリプレゼンテーションにとどまるべきか、それとも異なるリプレゼンテーションに切り替えるべきかを決定する。フラグメントが、複数のリプレゼンテーションにわたって必ずしもアライメントされず、RAPで始まらない、より一般的なケースでは、決定は同様であるが、切替えコストが高くなるので、このことを考慮に入れられる。

【0060】

[0095]SMリプレゼンテーションプロセスは、2つの論理的に別個のプロセスを備え、第1のプロセスは、RAが与える短期サンプルから概算持続ダウンロードレートを計算するレートエスティメータであり、第2のプロセスは、この推定値を利用して切替え決定を行う決定プロセスである。

【0061】

2. レート推定プロセス

[0096]適応ビットレートストリーミングクライアントは概して、正しいビットレートメディアを選ぶためのレート決定モジュールによって後で使用されるダウンロードレートエスティメータモジュールを使用する。この手法を用いると、ダウンロードレートが大きいとき、より高品質のメディアがストリーミングされ得る。ダウンロードレートの変化は、リプレゼンテーション切替えをトリガすることができる。レート推定値の品質は、ストリーミングクライアントの品質に対して大きな影響をもつ。

【0062】

[0097]適応ビデオストリーミングデバイスについての良好なレートエスティメータは、いくつかの特性を有するべきである。第1に、短期ダウンロードレートが大きく変わる場合であっても、分散はほとんどあるべきでない。第2に、基底チャネル(underlying channel)におけるレート変化に素早く適応するべきである。チャネルレートが大幅に下落すると、推定値は、その事実を素早く反映するべきであり、そうすることによってデバイスは、それに従って、失速することなく品質を調整することができる。それに応じて、ビデオ品質の向上が素早く観測されるはずであり、そうすることによってより良好な品質のコンテンツがフェッチされ得る。

【0063】

[0098]それらの2つの要件を満足するには、トレードオフが求められる場合がある。通常、分散が小さいエスティメータは、大きい反応時間を有し、逆もまた同様である。たとえば、デバイス内で使用することができる単純なエスティメータを検討する。そのエスティメータは、何らかの固定Xについて、ダウンロードの最後のX秒にわたって移動平均をとる。大きいX、たとえば、X=30秒(s)を選び出すと、分散がほとんどない比較的平滑な推定値が得られるが、ダウンロードレート変化に対してゆっくりとしか反応しない

10

20

30

40

50

。そのようなエスティメータがレート決定に使われた場合、結果として得られるプレーヤは、帯域幅下落に対して頻繁に失速し、またはそうすることが安全に可能であるときにより高いビットレートに適時に切り替えそこねる場合がある。これらの理由により、ある実装形態は、より小さな X 、たとえば $X = 3$ sを選び出す場合がある。そのような選択は、はるかに迅速なレート調整をもたらすが、安定性を犠牲にする。レート推定値は大きく変わり、プレーヤはしたがって、ビデオプレイバックレートを非常に頻繁に変化させて、劣悪なユーザエクスペリエンスをまねきうる。

【 0 0 6 4 】

[0099]図5において、凹凸の多い曲線は、たくさんの短期ゆらぎをもつ、未加工ダウンロードレートである。レートエスティメータは、凹凸の多いダウンロードレートの平滑化バージョンである。レートが変化すると、新規持続レートに収束し、レートが変わらない限りは同様のままである。

10

【 0 0 6 5 】

[0100]所望の特性のうち1つは、バッファレベルがわずかな場合、調整は迅速であり、これがレートの高速適応を引き起こし、その結果、ダウンロードレートが下落中であるときには、調整前にプレゼンテーションバッファが空にならない。一方、メディアバッファ内にたくさんのメディアデータがある場合、レート推定値はより平滑であり、調整はより遅い。メディアバッファ中により多くのメディアデータがあるときには、プレイアウトレートは、ダウンロードレートが下落中であるときには、メディアバッファ中のメディアデータがより少ないときよりも、より長い時間期間、より高いままである傾向にある。

20

【 0 0 6 6 】

[0101]これ以降で提示するレート推定プロセスは、 $pker$ 、 $pker$ プロセス、または $pker$ タイプのプロセスと呼ばれ、レート変化に迅速に反応するが、安定してもいるので、低い分散および高い反応性についての要件の両方を満足する。

【 0 0 6 7 】

2.1. $pker$ プロセス

[0102]このセクションは、本明細書において $pker$ 、 $pker$ タイプのプロセスまたは単に「 $pker$ プロセス」と呼ばれるレート推定プロセスについて記載する。基本レートエスティメータは、その推定値を、そこからより長い移動平均を計算するためにある方法または別の方法を使用した、短期レート測定値にのみ基づかせる。上述した基本移動ウィンドウ平均（「MWA」）は、そのようなプロセスの一例である。

30

【 0 0 6 8 】

[0103]図6～図7は、レート選択目的で非適応（固定係数）指数加重平均を使用することの効果を示す。これらのプロットは、簡単のために、新規レート推定値が新規ダウンロード選択を直ちにトリガし（すなわち、フラグメントが比較的小さい）、新規レート選択は単にレート推定であると仮定する。

【 0 0 6 9 】

[0104]図6は、 r 時間態様を示す。図に示すように、 x 軸はダウンロード時間（リアルタイム）である。時間 T_1 において劇的なレート増大が起こるとき、バッファは、非常に速く増大し始め、それは、ビデオデータがプレイアウトされるよりもはるかに速くそのビデオデータがダウンロードされているからである。EWMA推定値は、徐々に真のレートに収束する。

40

【 0 0 7 0 】

[0105]図7は、同じイベントの p 時間態様を示す。図において、線702は、スクリーンに表示されるビットレートを示す。レートは、図6の r 時間ピクチャにおけるよりもはるかにゆっくり適応する。 r 時間と比較した、 p 時間に対する収束の速度は、はじめは NR/OR の割合で減速される（プレーヤが、その時点で、ダウンロードの1秒あたりに約 NR/OR 秒分のビデオを受信したからである）。したがって、正味の効果は、メディアが、このタイプのレートエスティメータを使用するときには、多大な量の p 時間についてのダウンロードレートよりもはるかに低いレートでプレイアウトできることである。

50

【 0 0 7 1 】

[0106]レートが、メディアをストリーミングする目的で推定される場合、エスティメータは、他の関連情報を利用することができる。特に、メディアプレーヤのバッファ、または一般的に、バッファリングされたまたはすでにプレイアウトされた各メディアセグメントをダウンロードするのにどれだけ時間がかかったかについての情報を含む、メディアプレーヤのダウンロード履歴（現在のバッファにあるものよりも過去にさかのぼる）が関心事である。

【 0 0 7 2 】

[0107]ある実装形態は、たとえば、MWAエスティメータを使用することができるが、メディアバッファに応じたウィンドウサイズを選ぶことができる。

10

【 0 0 7 3 】

[0108]メディアプレーヤのバッファレベルが高い場合、プレーヤは、ただちに失速する危険状態にはないので、大きいウィンドウを使用して長期推定値をとることができ、結果としてより安定した推定値になる。一方、バッファレベルが低い場合、プレーヤは急速に反応するはずであり、このことは、この場合に、より短い平均化ウィンドウがより良好な選択であることを示唆する。

【 0 0 7 4 】

[0109]したがって、レート推定プロセスの実装形態は、変化するウィンドウ幅を使用することができるが、現在のメディアバッファ中の p 時間の量（つまり、ダウンロードされ、まだプレイアウトされていない現在の p 時間の量）に比例する r 時間ウィンドウ幅を使用し得る。

20

【 0 0 7 5 】

[0110]別の実装形態は、メディアバッファに現在含まれるバイト数に比例するようにウィンドウ幅を選ぶことができる。

【 0 0 7 6 】

[0111]ある実装形態は、単にバッファのレベルではなく、バッファ自体のコンテンツを検査することもできる。たとえば、バッファの大部分が、その同じコンテンツのプレイバック持続期間よりもはるかに短い時間でダウンロードされたと決定した場合、このことは、ダウンロードバッファが急速に増大している最中であり、したがってレートエスティメータは、推定値が調整される必要があると結論づけ得ることを示唆する。

30

【 0 0 7 7 】

[0112]同様に、レートエスティメータは、バッファレベルの変更レートを追跡し、バッファレベルの高速変化を、レート推定値が急速に調整される必要があるというインジケーションとみなすこともできる。

【 0 0 7 8 】

[0113]図 8 ~ 図 9 は、可変ウィンドウサイズ加重移動平均（「WMA」）フィルタが使用されるとき、図 6 ~ 図 7 と同じシナリオにおける挙動を示す。この例では、「 $p k e r$ 」プロセスは、そのような可変ウィンドウサイズWMAフィルタのように、プログラミングコードとして説明される。 $p k e r$ プロセスは、プロセッサによって実行されるプログラム命令として実施することができる。

40

【 0 0 7 9 】

[0114]図 8 において、線 8 0 2 は、基底チャネルが、レートOR（古いレート）からレートNR（新規レート）への急激なレート増大を有する場合の $p k e r$ レート推定値である。レート選択が新規レートに適應するのにかかる r 時間の量は、 OR / NR に比例する。増大が大きい程、適應はリアルタイムで速く起こる。図に示すように、時間 $T 2$ において、 $B u f f @ T 2 = 2 * B u f f @ T 1$ および $T_{f a s t} = OR / NR * B u f f @ T 1$ である。

【 0 0 8 0 】

[0115]図 9 は、 p 時間におけるプレイバック挙動を示す。 $p k e r$ エスティメータは、新規レートに適應するのに約 1 バッファ持続期間（レート増大が起きたとき、バッファ中

50

にあった p 時間の量) がかかり、すなわち、 p $k e r$ エスティメータは、メディアバッファが、 p 時間の持続時間 B がメディアバッファに追加された、メディアコンテンツの量を有するときには、新規レートに適応済みであり、ここで B は、新規レートまでのレート増大時におけるメディアバッファ中のメディアコンテンツの p 時間の持続時間である。

【 0 0 8 1 】

[0116] 上記を行う特定のプロセスについて、ここで説明する。このプロセスは、プレイバックバッファの最後の T 比分 (T -fraction) をダウンロードするのに、どれだけの r 時間がかかったか決定し、ここで T は適切に選ばれた定数である。たとえば、これは、現在のプレイバックバッファ全体 ($T = 1$) をダウンロードするのにかかった完了時間であり得、またはプレイバックバッファの後半分 ($T = 0.5$) をダウンロードするのにかかった時間であり得る。 $T > 1$ であることもあり得る。 T_{fast} を、プレイバックバッファの最後の T 比分をダウンロードするのにかかった r 時間の量とする。推定ダウンロードレートは、ダウンロード時間の前の T_{fast} 秒にわたるダウンロードレートを推定することによって計算されることができ、 T の他の値も可能であることに留意されたい。本明細書に記載するように、異なる値が、異なる目標に役立ち得る。

10

【 0 0 8 2 】

[0117] T_{fast} 幅のウィンドウにわたるこの種類のウィンドウ化平均は、レート増大を迅速に検出するという、注目すべき特性を有する。実際、 T_{fast} を決定するために値 $T < 1$ が使用される場合、エスティメータは、メディアバッファ中のメディアコンテンツの p 時間の持続時間が B である特定の瞬間においてどれだけレートが増大する場合でも、レートエスティメータが増大したレートに収束する前にバッファが最大でも B の有限倍数まで増大するという特性を有する。

20

【 0 0 8 3 】

[0118] より精巧なレート推定方法は、上で言及した 2 つの手法を組み合わせることができる。この方法は特に、バッファレベル B の最小値、および平均化ウィンドウ幅としての T_{fast} 、すなわち、ダウンロードレートを平均すべき r 時間の量を使用することができる。より一般的には、ダウンロードレートは、 $B \cdot B$ の最小値および T_{fast} の前の r 時間にわたって平均されることができ、ここで B は、適切に選ばれた定数である。そのような選択は、失速する危険をもってレートが下落したときに迅速に反応するという特性を有しており、というのは、それらのケースにおいて、 B は最小値であり、平均化は、メディアバッファ中のメディアコンテンツの p 時間の持続時間に比例する r 時間にわたり、したがって、メディアバッファが半ばまでドレインするときには、レート推定値は新規レートになるからである。たとえば、レート低下時において、メディアバッファ中のメディアコンテンツ持続期間は B であり、ダウンロードレートが、ダウンロードレート低下前の、選択されたプレゼンテーションのプレイバックレートの比 < 1 となるように、ダウンロードレートが低下し、悲観的には、選択されたりプレゼンテーションのプレイバックレートは、レート推定値が新規ダウンロードレートに減少するまで低下しないと想定する。次いで、ダウンロードは、レート低下が起こるときを超えて x の r 時間だけ継続するので、バッファレベルは、 $B' = B - x + \cdot x$ 、すなわち、メディアバッファから x 回の p 時間ドレインとなり、 $\cdot x$ がメディアバッファにダウンロードされる。レート推定値は、 $x = B'$ であるような時点で、すなわち、 p 時間におけるメディアバッファレベルが、ダウンロードが新規レートであった r 時間に等しくなる時点で、新規レートになる。というのは、この時点で、前の r 時間のダウンロードにわたる推定値が新規レートになるからであり、それは、この時間全体にわたって、ダウンロードが新規レートで行われたからである。式 $x = B' = B - x + \cdot x$ を x について解くと、 $x = B' = B / (2 - \cdot)$ となり、すなわち、バッファ B' が依然として少なくとも $B / 2$ であるとき、レート推定値は新規レートに達する。そうではなく、レートがどこかの時点で大幅に増大する場合、 T_{fast} は最小値になり、前の T_{fast} r 時間にわたる平均ダウンロードレートは、前の B r 時間にわたる平均よりも大幅に高くなる。

30

40

【 0 0 8 4 】

50

[0119]ここで、この構成に基づいて、p k e rレート推定プロセスの例について詳細に説明する。このプロセスは、要求アクセラレータ（R A）などのダウンロードモジュールから取得することができる短期レート測定値と、バッファ情報とを使用して、推定値を計算する。バッファ情報は、短期レート測定値が有用な推定値になるためのウィンドウ幅を決定するのに使用される。

【 0 0 8 5 】

[0120]図 1 0 は、ダウンロードレートが急激に下落するとき、p k e rレートエスティメータがどのように発展するかを示す。レートが下落するとすぐに、バッファレベルが下落し始める。レート推定値も、適応し始める。レート推定値は、遅くともバッファレベルが2分の1に下落したときには、新規レート（N R）に達する。この例では、中間レート決定が行われないので、B u f fは線形に下落する。中間決定が行われる場合、B u f fの降下は徐々に減速する。

10

【 0 0 8 6 】

[0121]p k e rプロセスの設計目標は、十分に大きい平均化ウィンドウを使用して、ノイズな数をもつのを回避し、反応するのに十分に短い数をもつことである。p k e rプロセスは、動的に変化するウィンドウサイズをもつウィンドウ化平均を使用することによって、この目標を達成する。R Aは、B、すなわちプレイバックバッファのレベル（p時間中）と、プロセスパラメータ B および T と、 T_{fast} 、すなわちバッファの（p時間中）最後の T 比분을ダウンロードするのにかかった r 時間についての保存された値と、R、すなわち r 時間中の最後のC持続期間のダウンロードにわたる平均ダウンロード速度とを含む、p k e rプロセスによって使用するためのいくつかの変数をメモリ中に維持し、ここで $C = \max(S T P, \min(B \cdot B, T_{fast}))$ であり、S T Pは最小許容可能ウィンドウサイズであり、Cはサンプル時間期間（たとえば100msなど）を超えるはずである。いくつかの実施形態では、 $B = 1$ および $T = 0.5$ であるが、他の値も可能であり、結果としては、両方が正であり $T < 1$ である限り、質的に同様の挙動となる。小さい B は、p k e rプロセスを、レート削減に対して素早く反応させ、小さい T は、p k e rプロセスをレート増大に対して素早く反応させる。

20

【 0 0 8 7 】

[0122]本明細書に記載するように、Cという持続期間にわたるダウンロード速度を計算するために、S Mは、R Aによって定期的に与えられるダウンロード速度情報を使用する。その目的のために、S Mは、R Aによって与えられるダウンロード速度情報の履歴を維持し得る。平均がとられる持続期間は、ほとんどの B バッファ持続時間であり、このことが、メディアバッファレベルに対して上限があるときに、どれだけの履歴が維持される必要があるかを効果的に制限する。

30

【 0 0 8 8 】

[0123]選択されたプレイアウトレートがダウンロードレートにほぼ等しい場合、ストリームをダウンロードするには、そのストリームをプレイアウトするのと同じ時間量だけかかるとすると、 $T_{fast} = T \cdot B$ となるので、バッファリング値Cは、バッファ持続期間程度であることに留意されたい。 r 時間中のバッファレベルの程度のものを選ぶことは、ダウンロードレート推定値についての平滑化間隔に対する当然の選択である、というのは、それは、ストリーミングクライアントが、失速を避けたい場合にもたなければならない見通し量だからである。

40

【 0 0 8 9 】

[0124]単純な一実装形態において、平均化ウィンドウ幅は、B、すなわちビデオバッファに含まれるp時間の量に比例する。そのような選択は、失速をうまく防ぐが、欠点がある。すなわち、ダウンロードレートが、選択されたメディアのレートのk倍である場合、毎秒のダウンロードの結果、p時間のうちk秒間分のメディアがダウンロードされ、レート推定を本当にゆっくり適応させることになる。たとえば、 $k = 10$ であり、10秒間分のバッファがある場合、レートエスティメータは、適応する前にp時間のうち約 $k \cdot 10 s = 100 s$ 分をダウンロードするが、これは非常に長い時間である。このことが、p k

50

er方法への T_{fast} パラメータの導入の理由となる。実際、平滑化に指数加重移動平均が使用される場合、そのようなフィルタは無限インパルス応答を有するので、事態はある程度悪化し得る。この理由により、pkerプロセスは、代わりに有限インパルス応答フィルタを使用する。プレーンな移動平均は機能するが、一実装形態は、また、より精巧な加重移動平均を使用し得る。

【0090】

[0125]図13は、この最後の点を示す。この図は、単純(固定幅)移動ウィンドウ平均と指数加重移動平均の比較を示す。グラフは、レート変化が見られるときには、固定ウィンドウ移動平均が最初によりゆっくりと新規レートに収束し得るが、1つのウィンドウ持続期間中に収束することを示す。指数加重移動平均は、はじめは急速に動く傾向があるが、後の段階では、ゆっくりと収束するだけである。ウィンドウ化移動平均とは異なり、固定ウィンドウ内では収束しないが、代わりに、レート変化のマグニチュードの対数の時間をかけて収束する。

10

【0091】

[0126] $B = 1$ および $T = 0.5$ なので、pkerプロセスは、様々な保証を与え得る。1つには、ダウンロード速度がどれだけ下落した場合でも、推定値は、バッファがその元の持続期間の半分に縮むのにかかる時間内に新規ダウンロード速度に調整される。もう1つには、ダウンロード速度がどれだけ増大した場合でも、追加のp時間に値する多くとも1つのバッファが、pkerプロセスが新規レートに収束する前にダウンロードされる。直接的な算出は、同様の固定比保証が、 $0 < B$ および $0 < T < 1$ のどの選択にも成り立つことを示す。

20

【0092】

[0127]バッファレベル、すなわちBを計算する1つの手法は、次のようになる。Tは、メディアプレーヤの現在のプレイバックp時間とし、 $F_{i,1}, \dots, F_{i,n}$ は、開始時間の昇順でソートされたリプレゼンテーショングループi中の、ダウンロード済みまたはダウンロード中であって、まだプレイアウトされていないフラグメントとする。依然としてダウンロード中であるグループiのどのフラグメントも $F_{i,1}, \dots, F_{i,n}$ の中にある。($F_{i,j}$)は、すでにダウンロードされたフラグメント $F_{i,j}$ のバイト数を、バイトでのフラグメント $F_{i,j}$ のサイズで除算したものなど、ダウンロード済みのフラグメント $F_{i,j}$ の部分とする。様々なiおよびjについての($F_{i,j}$)の値は、RAによって算出され、SMにパスされ得る。所与のグループiについて、ダウンロードされたp時間の現在の総量を、式1にあるように定義する。

30

【数1】

$$T_{p,i} := \text{starttime}(F_{i,1}) + \sum_{j=1}^{N_i} \text{duration}(F_{i,j}) \cdot \alpha(F_{i,j}) \quad (\text{式1})$$

40

【0093】

[0128]式1の結果から全体的 T_p 値を計算するために、DASHクライアントは、MPD(メディアプレゼンテーション記述メタデータ)から決定される、各グループの重みづけ因子、すなわち w' と、リプレゼンテーショングループの数、すなわちGとを考慮し、式2の計算を実施する。バッファレベルBはこれで、 $B := T_p - T$ と定義される。

【数2】

$$T_p := \sum_{i=1}^G w'_i \cdot T_{p,i} \quad (\text{式2})$$

50

【 0 0 9 4 】

[0129]式 2 は、現在プレイアウトされているフラグメントに属するバッファの部分も取り込む。この定義は、いくつかのフラグメントが一度にダウンロードされる場合にも機能することに留意されたい。

【 0 0 9 5 】

[0130] T_{fast} を計算するために、一般的ケースにおいて、SMは何らかの履歴を維持する。 T_r を、RAがメディアをダウンロードする(ダウンロードしようとする)のに費やした r 時間の総量とし、 Z を、RAによってダウンロードされたバイトの総量とする。 T_r の値は、RAによって計算される。SMは、 $i = 1, 2, \dots, K$ について規則的な間隔で(たとえば、100msおきに)サンプリングされたタプル

10

【数 3】

$$(T_r^i, Z^i, T_p^i)$$

【 0 0 9 6 】

の履歴、すなわち H を維持し、ここで第 K の観察が最後の観察である。履歴は観察順に記憶されると仮定するので、

【数 4】

$$T_{p,j}^1 \leq T_{p,j}^2 \leq \dots \leq T_{p,j}^K$$

20

【 0 0 9 7 】

ならびに

【数 5】

$$T_r^1 \leq T_r^2 \leq \dots \leq T_r^K$$

【 0 0 9 8 】

および

【数 6】

$$Z^1 \leq Z^2 \leq \dots \leq Z^K$$

30

【 0 0 9 9 】

となる。

【 0 1 0 0 】

[0131]ここで、 T_{fast} を計算するために、 B は、上に挙げた方法ですでに計算されると仮定する。次いで、 RA は、たとえばバイナリサーチで履歴をサーチすることによって、式 3 の不等式が満たされるように j を決定する。

【数 7】

40

$$T_p^K - T_p^{j+1} < \gamma_T \cdot B \leq T_p^K - T_p^j \quad (\text{式 3})$$

【 0 1 0 1 】

[0132]すると、

【数 8】

$$T_{\text{fast}} := T_r^K - T_r^J$$

【0102】

となる。無限履歴をずっと維持する必要はなく、最大バッファ持続期間の B 分を超えて広がる T_i 値についてだけで十分である。

【0103】

[0133] 図 15 は、図 16 の拡大版とともに、p k e r プロセスによって使用される値 B および T_{fast} が、記録された (T_p, T_r) 値の履歴からどのように決定され得るかを示す。この図は、 r 時間および p 時間が等しく高速に進行し（ダウンロード割込みはない）、したがってプレイバック時間（ p 時間）が、ダウンロード時間（ r 時間）の 45 度の傾斜線であるケースを示す。 (T_p, T_r) 値の履歴はグラフ中にプロットすることができ、プレイバック失速が起こらない場合、厳密にプレイバックタイムラインの上にある曲線が得られる。バッファレベル B はしたがって、プレイアウト時間に対する最後の記録された T_p 値の差である。 T_{fast} の値は、現在の（最後の） T_p 値を下回る $T_r \cdot B$ のレベルでの、 (T_p, T_r) 曲線までの水平距離を測定することによって、このグラフ中に見ることができる。

10

【0104】

[0134] 図 11 は、図 15 ~ 図 16 と同じ種類のプレゼンテーションを使用して、レート

の急上昇に対する p k e r プロセスの反応を示す。 T_{fast} は、プレーヤがまだ反応していない急上昇を受信レートが受けたときには、比較的小さい。これは、高受信レートに対する高速反応を示す。平均化ウィンドウは、比較的狭いので、完全にグラフの高レート部分内にあることに留意されたい。したがって、この時点で、p k e r 推定値は、より長いレートにすでに収束している。

20

【0105】

[0135] 図 12 はやはり、図 15 のプレゼンテーションを使用して、レート下落に対する可変ウィンドウサイズ WMA フィルタ（たとえば、p k e r）反応を示す。この場合、 T_{fast} は比較的大きくなるが、バッファがドレインするので、 B は小さくなり、ある程度のドレイン時間の後、平均化ウィンドウを、完全に低レートエリア内に収まらせる。図に示すように、平均化ウィンドウの幅、すなわち B は、 B が T_{fast} よりも小さくなるようなものであるが、推定値は依然として、バッファが完全にドレインされる前に、より低い新たなレートに収束する。

30

【0106】

[0136] 図 14 は、p k e r レート推定プロセスのフローチャートである。

【0107】

[0137] T_{fast} および B の値が計算されると、 C の値が容易に続き、最後のステップは、持続期間 C の過去のウィンドウにわたるレート R を計算するものである。その目的のために、履歴中の Z^i および

【数 9】

$$T_r^i$$

【0108】

値が使用される。

【0109】

[0138] 間隔 C にわたるレートを計算するために、SM または RA は、以下のことを行う：
（1）

40

【数 1 0】

$$T_r^K - T_r^j \geq C$$

【0 1 1 0】

となるような最も大きい j を見つけ、次いで、(2)式4にあるように平均ダウンロードレートを計算する。そのような j が第1のステップにおいて存在しない場合、SMまたはRAは、 $j := 0$ 、すなわち、最も古い既知の観察結果を設定する。 j の値は、バイナリサーチによって効率的に決定することができる。

【数 1 1】

10

$$R := \frac{z^K - z^j}{T_r^K - T_r^j} \quad (\text{式 4})$$

【0 1 1 1】

[0139]各グループは、そのグループが消費すると予想される総帯域幅の割合に対応する関連付けられた重み、すなわち w を有する。これは、好ましくは非使用可能リプレゼンテーションがフィルタ除去された後の、MPDによって与えられる情報の関数である。本明細書において、グループ g の重み w の提案される定義は、 $w(g) := \max rate(g) + \min rate(g)$ であり、ここで、 $\max rate()$ は、グループ g 中の最大プレイバックレートであり、 $\min rate()$ は、最小プレイバックレートである。

20

【0 1 1 2】

[0140]重み w から、SMまたはRAは、標準化重み w' を次のように計算することができる。クライアントが、グループ $1, \dots, G$ をストリーミングしたいと想定すると、標準化重みは、式5にあるように、すべての重みの合計で除算された重みである。

【数 1 2】

$$w'_i = \frac{w_i}{\sum_{j=1}^G w_j} \quad (\text{式 5})$$

30

【0 1 1 3】

[0141]標準化は、実際にストリーミングされる重みにわたって行われることを意図している。たとえば、ストリーミング中ではないグループがある場合、そのグループは考慮されるべきでない。

【0 1 1 4】

[0142]この $pker$ プロセスの動作において、いくつかの仮定が行われる。たとえば、個々のリプレゼンテーショングループのバッファレベルは、互いに比較的近く保たれるべきである。 $pker$ プロセスは、そうすることによってより良好に機能する。たとえば、あるグループが、非常に大きいバッファを有し、別のグループは非常に小さいバッファを有し、両方が同様の重みを有すると想定する。そのようなケースでは、小さいバッファにとっては条件が変わったときに失速を避けることが必要なので、レート推定値の迅速な調整を行うことが必要となる。しかし、 $pker$ プロセスは依然として、その推定値を、はるかに大きいバッファ向けに作用しているかのように満足に平滑化する。逆に、より大きいバッファにとっては、測定値は、バッファレベルが許容する、ある程度高い分散を有するので、ナーバスなレート決定となる。

40

【0 1 1 5】

50

[0143]いくつかのケースでは、バッファレベルが大きく相違するリプレゼンテーショングループを有することは不可避である。この理由により、別の実装形態は、いくつかのバッファが非常に小さいときに、より迅速にレートを調整し、したがってそのようなケースにおいてビットを失速からより良好に保護する、p k e r 方法の変形体を使用することができる。そのような実装形態は、以前と同じように T_{fast} を計算することができるが、ウィンドウサイズを $C = \max(S T P, \min(T_{fast}, T_{p,1} - T, T_{p,2} - T, \dots, T_{p,N} - T))$ に設定する。

【0116】

[0144]これらのダウンロードレート推定の他の変形体は、各リプレゼンテーショングループについての独立 p k e r 推定値を使用して、そのグループについて決定を行うことを含む。

10

【0117】

3. フェッチング戦略

[0145]ストリーミングビデオプレーヤは概して、制限付きメディアバッファを有する。したがって、通常動作において、最終的にはバッファ満杯状態に達し得ると予想される。バッファが満杯状態に達すると、ストリーミングモジュールは、バッファのオーバーフローリング (overfilling) を避けるために、メディア入力を抑える (throttle) べきである。これを行うための容易な方法は、バッファが満杯になったときは常に、バッファが次のフラグメントを保持することができるのに十分にドレインされるまで待ち、次いで、フェッチングを再開することである。

20

【0118】

[0146]この方法の効果は、各フラグメントが個々にフェッチされ、各フラグメント要求の間に時間ギャップがある、すなわち、次のフラグメントが収まり、要求され得るように、バッファから十分ドレインするのにかかる時間量があることである。

【0119】

[0147]TCP プロトコルは、そのダウンロードレートを、現在のネットワーク条件に基づいて自動調整する。TCP 接続を介してダウンロードが開始されたときには、初期ダウンロードレートは、通常、非常に遅く、より高いダウンロードレートを達成することができるかどうかを TCP プロトコルが調べると増大する。どれだけ高速に TCP がダウンロードレートを増大させるか、および TCP が概してエンドツーエンドの TCP 接続のプロパティにどのように反応するかは、極めて複雑であり、固有のエンドツーエンドのネットワークレイテンシ、TCP 配信および確認応答経路に沿ったネットワーク要素のバッファ容量、これらの経路に沿った競合トラフィック、TCP のどのような変形体が使用されるかなどを含む、いくつかの因子に依存する。概して、TCP は、遅いダウンロードレートで始まり、そのダウンロードレートを時間とともに増大させ、したがって、ダウンロード時間全体にわたる TCP 接続の平均ダウンロードレートは、ダウンロード時間全体が相当なものであるときの持続可能 TCP ダウンロードレートに近づくだけである。たとえば、持続可能 TCP ダウンロードレートが 1 メガビット / 秒であり、TCP 接続が、事実上ゼロのダウンロードレートで始まり、時間とともに 1 秒間にわたって 1 メガビット / 秒まで線形に増大した場合、最初の 1 秒にわたる平均ダウンロードレートは 500 キロビット / 秒であり、平均ダウンロードレートが持続可能ダウンロードレートの 95% を達成するには、10 秒間のダウンロードが必要である。この理由により、要求の間の多くのダウンロードギャップを有するフェッチング戦略は理想的ではなく、ダウンロードギャップは、あるダウンロード要求の完了と、次のダウンロード要求の開始との間の時間期間である。ダウンロード要求の間のギャップがゼロであるときでも、通常、TCP は、前の要求の完了の後、次の要求のためにダウンロードレートを増加させるのにある程度の期間かかるので、理想的ではない。各ギャップの後、持続可能スループットは再び達成されなければならない場合があり、これにより達成される全体的な平均ダウンロードレートが低減される。

30

40

【0120】

50

[0148]そのような低減されたレートは、より小さいレート推定値をまねき、したがってより小さいメディアレートの選択をまねき得る。その結果、通常、(バイトでのサイズが)より小さいメディアフラグメントがダウンロードされることになり、これによりギャップの相対的な大きさがさらに増大し、より一層小さいプレイバックレートが選択されることになり得る。言い換えると、その効果は自己増幅型である。

【0121】

[0149]したがって、DASHクライアント実装形態が、この問題の影響を最小限にするプロセスを使用することが有利である。

【0122】

[0150]ある実装形態は、メディアデータを絶え間なくダウンロードし、次いで、バッファレベルを次のように定期的にドレインすることができる。要求されたがまだプレイアウトされていないp時間の量が、あらかじめ設定された高いウォーターマーク、すなわち M_h を超える場合はいつでも、SMは、バッファレベルが低いウォーターマーク M_l を下回るまで、いかなる要求もそれ以上発行しない。特定の実装形態では、 $M_h = 20$ 秒および $M_l = 10$ 秒であるが、他の実装形態では、それらの値はより低くてもより高くてもよい。低いウォーターマークを下回った後、通常動作が再開され、SMは、フラグメント決定を再び発し始める。

10

【0123】

[0151]別の実装形態は、プレゼンテーション時間ではなくバイトで指定されたウォーターマークを使用して、同様の効果を達成することもできる。

20

【0124】

[0152]バッファが定期的にドレインしているという事実は、システムの他の部分によって、有利に利用することができる。たとえば、セクション6.1.2で説明するように、RTTの最新の推定値を取得するのに利用することができる。

【0125】

[0153]図17は、「ウォーターマーク」フェッチングプロセスの挙動を示す。上のグラフは、ドレイン期間とフェッチング期間の交替パターンが見られるバッファレベルグラフである。ダウンロードレートは、下のグラフに表示されている。各フェッチング期間の最初において、TCPは、持続可能最大速度になるのにある程度の時間をかけるので、(フェッチング期間中の)平均ダウンロードレートは、最大達成可能ダウンロードレートよりも小さい。低いウォーターマークと高いウォーターマークとの間の差が大きいほど、フェッチング期間が長くなり、平均レートが高くなる。

30

【0126】

4. レート選択プロセス

[0154]メディアデータを要求し始めるとき、ストリーミングモジュール(SM)は、何らかの方法を使用して、第1のプレイアウトレート選択を行う。SMは、最も低い利用可能レートをとる場合もあり、あるいは、たとえば、ネットワーク条件の履歴を維持し、次いで、この履歴に基づいて、失速なしで持続される可能性が高い、選ぶべきプレイアウトレートの推定値を決定する場合もある。SMが、すでにデータを受信中であり、したがって自由にレート推定値R(たとえば、セクション2にある方法で計算されたレート推定値のうちの一つなど)を有するとき、SMは、そのレートのまま続けるか、それともリプレゼンテーションを変えるかの決定を行う。

40

【0127】

[0155]単純なレート決定プロセスについて、ここで説明する。受信機は、推定ダウンロードレートRよりも低いプレイバックレートをもつ最も高い帯域幅リプレゼンテーションを決定し、データをプレイアウトする(プレイバックする)ためのリプレゼンテーションとして選び出す。直接的ではあるが、この手法には、いくつかの問題がある。第1に、その手法は、当然ながら、小さいメディアバッファを増大させないので、ダウンロードレートがほとんど変わらないときでも、失速を受けやすい。第2に、変動する推定値Rは、急速に変化するレート決定をまねき、これは、必要ではない場合があり、視覚的に邪魔にな

50

り得る。第3に、少なくともおおよそフラグメントの持続期間、したがって一般的に数秒の、スタートアップ時間につながる。

【0128】

[0156] DASHクライアントはしたがって、そのレート決定を、ダウンロード推定値Rだけではなく、バッファレベルB（つまり、バッファリングされ、まだプレイアウトされていないp時間の量）、および概して2つの連続する切替え点の間のp時間の持続時間の推定値である変化レートDなどのコンテンツに依存する変数にも基づかせる、レート決定プロセスを実装することができる。

【0129】

[0157]したがって、一実装形態は、決定レートとして、Rに比例する最も大きいプレイバックレートを選び出せばよく、ここで比例因子はバッファレベルの関数である。

10

【0130】

[0158]通常、比例因子は、バッファレベルの増加関数である。ある実装形態は、を、たとえば、バッファレベルのアフィン関数にすることができる。

【0131】

[0159] がバッファレベルの関数である場合、ある実装形態は、バッファが空または小さいときには小さくなるように を選ぶことができる。そのような選択は、小さいバッファを増大させ、ダウンロードレートが正確に予測されないときに失速に対するある程度の安全性も与えるので、有利である。

【0132】

20

[0160]より大きいバッファレベルの場合、ある実装形態は、1に近い、1に等しい、または1を超える の値を選ぶことができる。こうすることにより、失速する即時の危険性がないときには確実に高プレイアウトレートがダウンロードされるために選ばれることになり、定常状態において高品質メディアがストリーミングされることになる。

【0133】

[0161]レート決定プロセスは、単なる単純なアフィン関数ではなく、Bの区分的アフィン関数である を実装することができる。区分的アフィン関数は、任意の連続関数を、任意の所望の程度の精度まで概算することができ、そうすることにより連続関数が適切な選択となる。代わりに、同じプロパティをもつ、他の任意のパラメータ化可能クラスの関数が選ばれることもできる。

30

【0134】

[0162]別の実装形態は、 を、p時間で表されるバッファレベルではなく、バイトで表されるバッファレベルの関数にすることもできる。

【0135】

[0163]さらに別の実装形態は、 を、バッファレベルBだけではなく、バッファレベルBと切替え機会の頻度の両方の関数にする。そのようにする理由は、レートを変える機会がより少ないプレーヤが、変える機会がより頻繁なものよりも、さらに今後は各決定に関わるからである。したがって、前者のケースでは、各決定は、より大きいタイムスパン、およびより高い危険性への関与である。このことは、バッファレベルBと推定ダウンロードレートRが同じであるとき、失速する危険性を小さく保つために、後者よりも前者のケースではより低いレートを選び出す方がよいであろうということを示唆する。

40

【0136】

[0164]レート切替え機会の頻度を考慮に入れるレート選択プロセスのための具体的方法は、次のようになる。Dを、ストリーム中の2つの連続する切替え点の間のp時間の典型的な量とする。Dの値は、符号化ビデオに依存し、たとえば、2つの連続する切替え点の間のp時間中の最大距離、または2つの連続する切替え点の平均距離、または2つの連続する切替え点の90パーセントイル距離、またはメディア中の2つの連続する切替え点のp時間距離の他の任意の適切な測度となるようにとられ得る。そのようなDを所与とすると、方法は、 B/D の区分的アフィン関数、またはその変形体、たとえば $B/\max(u, D)$ もしくは $B/(D+u)$ などとなるように を選ぶことを含むことができ、ここで

50

値 u は、要求を発行する際に誘発されるオーバーヘッドを考慮に入れるために追加される。 u の値は、小さい一定の時間量（たとえば 100ms など）であることができる。さらなる改良として、ある実装形態は、 u を、推定 RTT の小さい倍数とすることができる。

【0137】

[0165] 上述した方法など、そのレート決定を単に $\mu \cdot R$ に基づかせるプロセスには、 R の比較的小さい変動でさえも、多くのレート切替えをまねき得るという欠点がある。これは、望ましくない場合がある。十分なバッファがあるときには、 R の小さな変化には直ちに反応せず、それに応じてバッファレベルを変わらせる方がよい場合がある。

【0138】

[0166] そのような挙動を得るために、プロセスは、両方とも同じ量（たとえば、上で説明したように、 B 、 B/D または $B/\max(100\text{ms}, D)$ ）の関数である値 μ および μ を使用することができる。これらの値は、現在のレートとともに、新規レート決定を選び出すためのものである。これらの関数は、 $\mu \cdot R$ が低い許容可能レート選択となり、 $\mu \cdot R$ が高い許容可能レート選択となるように選ばれるべきである。プロセスは次いで、それら2つの値を良好なレート決定のためのガイドとして使用するよう設計され得る。

10

【0139】

[0167] そのような設定において、関数は、概して μ となるように選ばれるべきである。

【0140】

[0168] レート決定プロセスは、前の選択がすでに $\mu \cdot R \sim \mu \cdot R$ の範囲内だった場合、レートを同じに保つことを決定し得る。前の選択が $\mu \cdot R$ 未満の場合、 $\mu \cdot R$ 以下の、最も大きい利用可能プレイバックレートが選択される。前の選択が $\mu \cdot R$ よりも大きい場合、 $\mu \cdot R$ 以下の、最も大きい利用可能プレイバックレートが選択される。

20

【0141】

[0169] ある実装形態は、関数 μ および μ がハードコードされることを選ぶことができる。あるいは、状況に応じて、より入念な方法で関数を選択することもできる。具体的には、一実装形態は、クライアントが最大限で行うバッファリングの量に応じて、適切な μ および μ 関数を選択することができる。オンデマンドコンテンツの場合、クライアントは、たくさんのデータ、可能性としては数分間のメディアデータをプリバッファリングすることを選ぶことができる。低遅延ライブコンテンツの場合、クライアントは、多くとも、おそらく数秒間のみの、エンドツーエンドのレイテンシによって規定されるメディア量をバッファリングしさえすればよい。バッファリングがほとんどないコンテンツの場合、クライアントは、よりコンサバティブ (conservative) な、すなわち、より小さい値を有する、 μ および μ 関数を選び出すと決定し得る。

30

【0142】

[0170] 具体的実装形態は、たとえば、2つの極値関数 μ_1 と μ_2 との間で関数を線形に補間することができる。ここで選択される補間点は、低バッファウォーターマーク M_1 (セクション3参照) である。したがって、この実装形態は、2つのハードコードされた関数、すなわち μ_1 と μ_2 とを有し、 μ_1 は、ある m_1 未満の、 M_1 の小さい値用に使われ、 μ_2 は、ある値 m_1 、 m_2 について $M_1 \sim m_2$ であるときに使われ、ここで $m_1 < m_2$ である。 $m_1 \sim m_2$ の範囲内の値について、関数 $\mu(x) := \mu_1(x) (m_2 - M_1) / (m_2 - m_1) + \mu_2(x) (M_1 - m_1) / (m_2 - m_1)$ が使用される。

40

【0143】

[0171] ここで、上記記述にしたがう、レート決定プロセスの詳細な例を挙げる。このために、いくつかの表記法を採り入れる。

【0144】

[0172] 1) S_1, S_2, \dots, S_L を、リプレゼンテーショングループの (昇順で与えられる) L 個の利用可能リプレゼンテーションのストリームレートとする。

【0145】

[0173] 2) $\mu(x)$ を、負でないスカラーを入力としてとり、負でない実スケーリン

50

グ係数を戻す区分的一次関数とする。関数 $\lambda(x)$ は、コンパイル時に、または構成ファイルにより設定可能であるべきである。大きい x について、たとえば、 M_1 よりも大きい x について、 $\lambda(x)$ は不変であるべきである。

【 0 1 4 6 】

[0174]ここで、そのような関数がどのように実装され得るかについての一例を挙げる。端点 $(0, \lambda_0)$, (x_1, λ_1) , ..., (x_N, λ_N) が与えられ、ここで x_i は昇順である。 $\lambda(x)$ を評価するために、 $x_i \leq x$ であるような最も大きい i を見つける。次いで、式 6 を使用して、受信機はこの関数を評価することができる。

【 数 1 3 】

$$\lambda(x) = \begin{cases} \lambda_i + (\lambda_{i+1} - \lambda_i) \cdot \frac{x - x_i}{x_{i+1} - x_i}, & \text{if } i < N \\ \lambda_N, & \text{if } i = N \end{cases} \quad \text{(式 6)}$$

10

【 0 1 4 7 】

[0175]そのような $\lambda(x)$ 関数についての適切な例は、例示的パラメータ $N = 1$ 、 $[(0, 0.5), (3, 1)]$ によって定義されるもの、つまり、 $x = 0$ において 0.5 に等しく、 x が 3 に達するまで線形に増大する関数であり、 x が 3 に達した時点で、関数は 1 に等しく、その後は 1 のままである。

20

【 0 1 4 8 】

[0176] $\mu(x)$ を、別のそのような区分的一次関数とする。そのような関数の一例は、 $x = 0$ において 0 になり、 $x = 3$ において 1.5 に達し、その後は一定のままであるものである。

【 0 1 4 9 】

[0177] D を、(以前指定された)ある切替え点から次の切替え点までの p 時間における持続期間の推定値とする。

【 0 1 5 0 】

[0178] $x := \min\{(T_d - T), M_1\} / \max\{D, 1 \text{ 秒}\}$ とし、ここで T は現在のプレイバック p 時間であり、 T_d は、レート決定が行われる p 時間であり、 D は、上に挙げた通りであり、 M_1 はバッファレベル低マーク(セクション 3 参照)である。

30

【 0 1 5 1 】

[0179] $CURR$ を、現在選択されているリプレゼンテーション(すなわち、最後のフラグメント要求において使われたもの)とする。 UP を、レートが最高でも $\lambda(x) \cdot R$ である最も高いビットレートリプレゼンテーションのプレイアウトレートとし、そのようなリプレゼンテーションがない場合、 UP は、最も低いビットレートリプレゼンテーションのプレイアウトレートである。 $DOWN$ を、最高でも $\mu(x) \cdot R$ のレートの最も高いビットレートリプレゼンテーションのプレイアウトレートとし、そのようなリプレゼンテーションがない場合、 $DOWN$ は、最も低いビットレートリプレゼンテーションのプレイアウトレートである。概して $\lambda(x) \leq \mu(x)$ なので、概して $DOWN \leq UP$ である。

40

【 0 1 5 2 】

[0180]次いで、レート決定プロセスは、次のフラグメントのレート $NEXT$ を次のように選び出す。(1) $UP < CURR$ の場合、 $NEXT := \min(DOWN, CURR)$ であり、(2) それ以外の場合、 $NEXT := UP$ である。

【 0 1 5 3 】

[0181]上のステップ 5 で単に D ではなく $\max\{D, 1 \text{ 秒}\}$ を使用する理由は、 RTT ゆえであり、1 の役割は、 RTT の上限として作用することである。

【 0 1 5 4 】

50

[0182]関数 $\lambda(x)$ および $\mu(x)$ は、 x に応じて増加していくことが好ましい。 および μ 関数は、小さい x については < 1 であることが好ましく、それにより確実に、選ばれたプレイアウトレートが R 未満になり、小さいバッファレベルについてバッファ増大が引き起こされる。選択されたプレイバックレートは、最高でも $\max(B / \max\{D, 1\})$ 、 $\mu(B / \max\{D, 1\}) \cdot R$ に等しく、 $(B / \max\{D, 1\})$ と $\mu(B / \max\{D, 1\})$ の両方が 1 未満であるバッファレベル B すべてについてバッファ増大が保証されることに留意されたい。

【0155】

[0183]より簡単なプロセスは、 $(B) \cdot R$ 未満のプレイバックレートをもつ最良リプレゼンテーションとなるように、新規リプレゼンテーションを直接選び出してよい。このリプレゼンテーションは、バッファが空になりそうなときに、バッファは満杯になる傾向にあるというプロパティを依然として有する。ただし、 R は大きく変動し得るので、たくさんのリプレゼンテーション切替えも引き起こす。本明細書において記述する、より洗練されたレート選択プロセスは、切替えを避けようとし、代わりに、より低いプレイバックレートに切り替える前に、バッファをある程度までドレインさせる。これが機能するためには、関数 μ および λ は、中間から大までのバッファレベルについて、 μ が λ を超えるように選ばれるべきであり、ここで、選択されたプレイバックレートが $CURR$ であり、測定されたレートが R である場合、式 7 が満足される限りレート変化は起こらず、レート切替えなしで受信レートのある程度変動させることに留意されたい。

【数 14】

$$\frac{CURR}{\mu(B / \max\{D, 1\})} \leq R \leq \frac{CURR}{\lambda(B / \max\{D, 1\})} \tag{式 7}$$

【0156】

[0184]いくつかのバージョンでは、 λ および μ は単に、比 $B / \max\{D, 1\}$ ではなくバッファレベル B の関数である。後者を導入する動機は、次のようになる。

【0157】

[0185] λ を、選択されたリプレゼンテーションのプレイバックレートと、ダウンロードレートとの比を示すものとする。良好な λ を決定することが望まれる。次の切替え点までダウンロードするのに、ほぼ $\lambda \cdot D$ の r 時間だけかかる。受信データがバッファに追加されるすぐ前に、バッファは、 $B - \lambda \cdot D$ までドレインされている。失速を避けるために、その量を正にしたいが、セーフティクッションとして、それは、ダウンロードされると、バッファに追加されるフラグメントのプレイバック持続期間 D に比例するべきであるので、ある $\lambda > 0$ について、少なくとも $\lambda \cdot D$ であるべきである。要するに、 $B - \lambda \cdot D > 0$ としたい。

【0158】

[0186] λ の値を求めると、 $B / D - \lambda$ が得られる。これは、リプレゼンテーション選択プロセスが、 $B / D - \lambda$ を超えない、プレイバックとダウンロードレートの比を選ぶべきであることを示唆する。関数 $\lambda(x)$ および $\mu(x)$ は、許容可能なそのような比に対する限度であるので、 $\lambda(x) > B / D - \lambda$ を超えない $x = B / D$ の関数であるべきである。

【0159】

[0187]実際には、1つのフラグメントを送信する、 RTT の追加コストを考慮に入れるために、 B / D を、 $B / \max\{D, 1\}$ で置き換える。より一般的には、 λ は、 RTT の近似のある程度の倍数、またはサーバからメディアデータのダウンロードを開始するためのプロセスの反応時間を考慮に入れる他のパラメータで置き換えることができる。

【0160】

[0188]図 18 は、プレイバックレートを選択するのに使用され得る λ 関数および μ 関数の例を示す。 x 軸は、 D の単位で表したバッファレベルであり、 y 軸は、受信比 (receiv

10

20

30

40

50

e fraction)、すなわち、プレイバックリプレゼンテーションレートを現在の受信またはダウンロードレートで除算したものである。線 1802 によって図に示すように、受信比が 1 未満の場合、バッファは増大し、1 よりも大きい場合は縮む。3つのエリアが識別される。第 1 に、プレーヤは、決定点において 曲線 1804 を下回る場合、高いレートに切り替える。 曲線 1804 と μ 曲線 1806 との間である場合、選択されたレートのままである。 μ 曲線 1806 を上回る場合、低いレートに切り替える。

【0161】

[0189]図 19 は、「コンサバティブ (conservative)」設定を使用する、(, μ) 関数の例示的選択を示す。この設定は、利用可能な帯域幅すべてを使用するわけではないが、引き換えに、非常にまれにしか失速しないという点で「コンサバティブ」である。

10

【0162】

[0190]図 20 は、「中間 (moderate)」設定を使用する、(, μ) 関数の例示的選択を示す。この設定は、コンサバティブ設定よりも多い帯域幅を使用するが、わずかに失速しやすいという点で「中間」である。

【0163】

[0191]図 21 は、「アグレッシブ (aggressive)」設定を使用する、(, μ) 関数の例示的選択を示す。この設定は、利用可能帯域幅をすべて積極的に (aggressively) に使おうとするという点で「アグレッシブ」である。この設定は、他の 2 つの提示した例示の設定よりも頻繁に失速し得る。

【0164】

20

[0192]図 22 は、MLB プロセス、すなわち、米メジャーリーグ (MLB) で働いている何人かの研究者によって提案されたものと同様のプロセスをある程度までエミュレートするためのプロセスを使用する、(, μ) 関数の例示的選択を示す。(, μ) 関数は、メディアバッファ充満度に基づいて変わるのではないことに留意されたい。

【0165】

[0193]図 23 は、 設定および μ 設定用の隣り合っている値の例を示す。

【0166】

[0194]図 24 は、 設定および μ 設定用の隣り合っている値の例を示す。

【0167】

[0195]図 36 は、レート選択において および μ に使用することができる値のテーブルを備える。

30

【0168】

[0196]図 25 は、レート推定、次いでレートベースのレート選択、次いでバッファ管理ベースのレート選択のためのプロセスを示す。この例示的プロセスにおいて、本明細書において記述する手法のうち 1 つまたは複数が、レート推定を実施するのに使用される。その推定値に基づいて、新規プレイバックレートが選択され、バッファ管理ルールに基づいて調整されることが可能である。

【0169】

5. 要求取消し

[0197]いくつかのケースでは、優れたレート選択プロセスでさえも、単独ではビデオプレイバックの失速を防止することができない。たとえば、要求が行われた後であるが完了される前に、ダウンロードレートが急激に下落した場合、選択されたビットレートは大きすぎた可能性があり、遅いダウンロードレートは、プレイバックレートを変えるための次の切替え機会に達する前であってもプレイバック失速につながり得る。

40

【0170】

[0198]別の例として、たとえば、セルラー接続から Wi-Fi 接続への移行により、利用可能帯域幅が劇的に増大するときには、メディアバッファは、比較的低いプレイバックレートメディアで満杯になり得る。この場合、すでにダウンロードされたがまだプレイアウトされていないメディアの大部分を破棄し、破棄された p 時間部分を再度ダウンロードするが、今度はダウンロードするべきより高いプレイバックレートリプレゼンテーションを

50

選ぶことが有利であり得る。したがって、すでにダウンロードされた低プレイバックレートメディアは取り消され、別のリプレゼンテーションからのより高いプレイバックレートのメディアが、プレイアウトされるべき場所にダウンロードされるので、より高品質のユーザエクスペリエンスがもたらされる。

【 0 1 7 1 】

[0199]この理由により、ストリーミングモジュール実装形態は、ダウンロードレートを監視するモジュールを実装することができ、いくつかの状況では以前の決定を取り消し得る。要求が取り消された場合、ストリーミングモジュールは、ダウンロードレートのより新しい、より適切な推定値に基づいて新規要求を発行するべきである。この監視モジュールを、本明細書では要求取消しプロセスと呼ぶ。

10

【 0 1 7 2 】

[0200]要求取消しプロセスは、様々な理由で要求を取り消すことができる。たとえば、ダウンロードレートが激しく下落し、プレイバックが、データが十分高速に受信されていないことにより、失速する危険があるという理由で、要求を取り消すことができる。取り消す別の理由は、プレイバックのために時間内により高品質のメディアが選択され、取り出され得ると決定された場合である。取り消すさらに別の理由は、受信機が何をしたか、および保留要求の完了の許容に対して取消しが失速期間を短縮するかどうかという推定にかかわらず、失速が起こると受信機が決定した場合である。受信機は次いで、可能性としてはプレイバックされるべきメディアリプレゼンテーションの品質も考慮に入れて、推定されたより短い失速とともに進むアクションを選ぶ。当然ながら、失速があるかどうか、および失速がある場合には失速の持続期間は、推定とは異なり得る。

20

【 0 1 7 3 】

[0201]実際の取消しは、取消しが決定された後には、要求がその上で発行されたTCP接続をクローズすることによって達成することができる。クローズは、取り消されたフラグメントについてのデータを送り続けないうサーバに伝えるという効果を有し、したがって、クローズされた接続によって使用される帯域幅が、置換えデータをフェッチするために利用可能になる。

【 0 1 7 4 】

[0202]ストリーミングモジュールは次いで、取り消された部分を置き換えるための要求を発行することができる。この目的のために新規TCP接続をオープンすることが必要な場合がある。

30

【 0 1 7 5 】

[0203]ある実装形態は、置換え要求を選ぶいくつかのオプションを有する。どれが最も適切なオプションであるかは、プレイアウトされているコンテンツのタイプに依存し得る。

【 0 1 7 6 】

[0204]それは、ストリームのシームレスなプレイバックを可能にする置換え要求を選び出す場合がある。一般的ケースでは、このことは、置換え要求が、前のダウンロードされたフラグメントの終了時に、またはその前に、切替え点をもたなければならないことを意味する。

40

【 0 1 7 7 】

[0205]その場合、プレーヤは、取消しなしでダウンロードを続けるときに失速が予測され、置換えセグメントの取消しおよび選択を用いると失速を回避でき、または少なくとも失速の持続時間を短縮できることが予測される場合、取り消すべきである。プレーヤは次いで、置換え要求についてのそのプロパティをもつ最も高品質のメディア要求を選び出すことができる。

【 0 1 7 8 】

[0206]レート取消しプロセスは、失速を次のように予測することができる。このプロセスは、フラグメント中の未処理バイトの数をダウンロードレートの推定値で除算することによって、発行された要求の推定完了時間を計算することができる。その時間が、円滑な

50

プレイバックのためにフラグメントが必要とされるデッドラインよりも後の場合、失速が予測される。

【0179】

[0207]切迫した失速が予測される場合、要求取消しプロセスは、レートの切替えが事態を改善する見込みがあるかどうかを決定し、改善が見込まれるときのみ、取り消すという決定が行われる。

【0180】

[0208]一実装形態は、候補置換えフラグメントのレート推定値およびサイズにのみ基づいて、置換えフラグメントをロードするのにかかる時間を推定することができる。

【0181】

[0209]別の実装形態は、取消しによる追加オーバーヘッドも考慮に入れることができる。すなわち、既存の要求を取り消し、新規要求を発行するのに必要とされる時間を考慮するために、推定RTTの倍数を加算することができる。取り消された要求からネットワーク上での配信のためにキューイングされているがまだ宛先には届いていないデータは、追加遅延に寄与し得る。クライアントは、この遅延を、TCP受信ウィンドウサイズを推定レートで除算することによって推定することができる。遅延の別の推定値は、推定帯域幅遅延積に基づき得る。クライアントは、2つの推定値のうちの最大値など、2つの推定値を組み合わせることができる。

【0182】

[0210]要約すれば、クライアントは、置換えフラグメント全体をダウンロードするのに必要とされる時間と、通常はRTTに比例する量と、キューイング遅延の推定値との合計を計算する。失速が予測され、その時間が現在のフラグメントをダウンロードするための推定残り時間よりも小さい場合、取消しが発行される。

【0183】

[0211]要求取消しプロセスは、また、初期のレート選択が正確でなかったために、第1のフラグメントをダウンロードするのに所望されるよりも長くかかることにプレーヤが気づいたときには、スタートアップ時に取り消すこともできる。

【0184】

[0212]別のレート取消し実装形態は、また、シームレスなプレイバックを認めず、いくつかのフレームをスキップすることを含意する、置換え要求を選び出すこともできる。これは、エンドツーエンドのレイテンシが小さく保たれることを要求するライブコンテンツを再生するときに必要なとなり得る。

【0185】

[0213]フレームスキップを用いて取消しを行う実装形態は、フレームスキップが可能な限り小さくなるような方法で置換えフラグメントを選び出すことができる。

【0186】

[0214]この実装形態は、置換え要求として、指定された失速持続期間またはスキップフレーム持続期間を超えることなく持続可能にダウンロードすることができる、最も高品質の要求を選ぶことができる。

【0187】

[0215]すでにダウンロードされたフラグメントについては、異なる種類の取消しが実装され得る。プレーヤは、プレイアウトされる予定のいくらかのメディアをすでにバッファリングしている場合、ネットワークを通じてより高品質のリプレゼンテーションをフェッチし、それをストリーミングすることを決定することができ、以前バッファリングされた、より低品質バージョンは破棄し得る。

【0188】

[0216]その取消しプロセスは、失速することなくプレイアウトすることができるように時間内により良好な品質のビデオを受信することができると決定した場合、これらの置換え動作を行うことを決定し得る。

【0189】

10

20

30

40

50

[0217]図 2 6 は、時間 T_1 における新規フラグメント要求の直後に起こる、ダウンロードレートの強烈な下落を示す。この要求まで、受信レートは OR であり、次いで、 NR に下落する。バッファレベルは今では下落している。新たに要求されたフラグメントは、約 $T_2 = T_1 + OR / NR * \text{フラグメント持続期間の時間}$ に、完全にダウンロードされる。 OR / NR が大きい場合、この値は、時間 T_1 におけるバッファ中のメディアコンテンツの p 時間の持続時間よりも大きくなり得、このことは、要求されたフラグメントは、失速なしではプレイバックされないことを意味する。 $pkerr$ エステイメータは、はるかに速くレート NR に収束しているが、 T_1 よりも前に要求が行われたので、フラグメントのダウンロードは、推定値が新規レート NR に収束する機会がないうちに行われることに留意されたい。失速を避け、訂正された推定値をもつ新規要求を発行するために、要求を取り消し、より適切なビットレートで要求を再発行することが必要である。

10

【0190】

[0218]図 2 7 は、要求取消しのあるケースを示す。ダウンロードレートの鋭い下落（線 2702）の後、バッファはドレインし始め、推定ダウンロードレート（たとえば、 $pkerr$ プロセス）は、新規ダウンロードレートに収束し始める。どこかの時点で、ストリームマネージャは、失速なしでプレイバックのための時間内にフラグメントが受信されないことに気づく。その点は、図 2 7 のプロットにおいて、「取消し点」2704としてマークされている。その時点で、部分的に受信されているフラグメントは、取り消され、バッファから追い出される（したがって、バッファレベルがさらに下落する）。しかしその後、正しいレートをもつフラグメントが要求され得るので、バッファレベルはそれ以上下落しない。実際、自明でない（nontrivial）レート選択プロセスが使用される場合、レベルは再度増大し得る。

20

【0191】

[0219]図 2 8 は、例示的要求取消しプロセスを示すフローチャートである。

【0192】

[0220]図 2 9 は、要求取消し検出のためのプロセスを示す。

【0193】

[0221]ここで、上記詳細に基づいて、要求取消し実装形態について説明する。

【0194】

[0222]このセクションにおいて、 N_i は、要求されたが、まだ完全には受信されていない、リプレゼンテーショングループ i 中のフラグメントの数を示す。これらは、 $F_{i,1}, \dots, F_{i,N_i}$ として参照される。さらに、 $F_{i,j}$ は、開始 p 時間の昇順でソートされると仮定し、 $(F_{i,j})$ は、要求されたフラグメント $F_{i,j}$ についてすでにダウンロードされたバイトの量を、バイトで表されたフラグメントのサイズで除算したものである。変数 T は、現在のプレイバック p 時間を示す。要求取消し検出プロセスは、図 2 9 の擬似コードによって示すように進行し得る。

30

【0195】

[0223]要求取消し検出プロセスは、実行されると、ニル（nil）を返すことができ、この場合はいかなるアクションもとられることはなく、あるいは、取り消すべきグループ中のフラグメントを識別する。そのようなフラグメントが識別された場合、このことは、このフラグメント、および（ p 時間順で）その後にくる、同じグループ中のあらゆるものが、取り消され、バッファからフラッシュされるべきであることを意味する。 SM は次いで、そのレート決定プロセスを再度呼び出し、セクション用に新規代替要求を発行すべきである。

40

【0196】

[0224]このプロセスを説明するために、差し当たり、単一の要求のみが今のところ未処理であると仮定する。その場合、 R を、ダウンロードレートの正確な推定値とし、 d_{avail} を、問題となっているフラグメントがプレイアウトされることになるまで、依然として受信することができるバイト数とする。量 d_{need} は、そのフラグメント中で依然として欠けているバイト数である。したがって、 $d_{avail} < d_{need}$ の場合、プレーヤは、フラグメ

50

ント $F_{i,j}$ を再生する前に失速すると予測する。このことは、以上のプロセス中の最初の「if」条件について説明している。

【0197】

[0225]失速が予測される場合であっても、取消しにより失速を避けることになる場合、または少なくともその持続期間を短縮する場合にのみ、取り消すことに意味がある。取消しの後には、新たなフラグメントが選択され、最初からダウンロードされなければならない。ただ1つのリプレゼンテーショングループがあり、レート決定プロセスが正しいレートを選んだ場合、これには持続期間 ($F_{i,j}$) のほぼ 倍の時間がかかり、ここでは、現在の適切なラムダ因子である。一方、SMが切り替えないと決定した場合、現在のフラグメントダウンロードの終了には、 $d_{need} \cdot R^{-1}$ の時間がかかる。簡単のために、 $\lambda = 1$ と仮定すると、他の因子をもち得る、第2の条件が得られる。

10

【0198】

6. 要求アクセラレータ

[0226]ストリーミングメディアクライアント向けの直接的なやり方は、単一HTTP接続を用いてメディアをフェッチすることである。そのようなクライアントは、フラグメント要求を連続して処理することになる。そのような手法には、ビデオストリーミングにおいていくつかの欠点がある。第1に、一般的ネットワーキングソフトウェアはしばしば、長いダウンロードにわたる最大スループットについてのみ合わせられる。これは、大きいファイルを受信するには良いが、安定受信レートなど、他の重要なストリーミング目標とは相反する。第2に、TCPの性質により、リンクの満杯容量は、必ずしもそのようなHTTP接続とともに使用することができるわけではない。チャンネルが、ある程度の遅延とパケット損失とを受けると、TCPは、達成され得る実際のスループットを制限し、こうすることにより、ストリーミングクライアントが、良好な品質のメディアをストリーミングするのが妨げられる場合がある。

20

【0199】

[0227]これらの問題を避けるために、本明細書において要求アクセラレータ (RA) と呼ぶ、特殊HTTPクライアントが実装されることが出来る。要求アクセラレータは、前述した問題を回避または軽減するための特殊プロセスを有する。要求アクセラレータのある実装形態は、いくつかの重要な要素を利用して、その目標を達成することができる。それは、いくつかのTCP接続を使用して、データを受信することができる。それらの接続は、並行してアクティブであることができる。データ要求を、より小さいチャック要求に分割することができ、それらのチャック要求は、異なる接続上で個々にダウンロードされ、要求アクセラレータにおいて、1つの大きな部分に再アセンブルされ得る。それは、接続が相互に公平 (fair) になるように、TCP接続パラメータ (具体的には、TCP受信ウィンドウサイズなど) を調節し、比較的安定したデータ受信を行うことができる。それは、使用するべきTCP接続の数を、測定されたネットワーク条件および目標プレイバックレートに基づいて動的に調整することができる。

30

【0200】

[0228]使用するべきTCP接続の理想的な数は、ネットワーク条件、特に、ラウンドトリップ時間 (RTT) およびパケット損失挙動に依存する。RAはしたがって、これらの量を推定するための方法を使用することができる。

40

【0201】

[0229]RAは、HTTP要求の発行から、応答が着信し始めるまでかかる時間をサンプリングすることによって、RTTを推定することができる。一実装形態は、一定期間、たとえば最後の数秒間にわたって取得されたすべてのそのようなサンプルの最小値をとることによって得られるRTTの推定値を使用することができる。別の実装形態は、最後のN個の取得サンプルの最小値を使用することができ、ここでNは、推定値としての何らかの整数である。

【0202】

[0230]TCPプロトコルは、パケット損失を扱い、連続するデータプリフィックスをア

50

アプリケーションに届けるので、TCP層より上でのパケット損失の測定値を取得することは、しばしば困難である。したがって、代わりに、パケット損失についての妥当な値をRAプロセスへの入力として固定することが有用なことがある。ある実装形態は、損失を一定であると推定し得る。いかなるパケット損失測定値ももたないので、RAは、損失を1%であると推定する場合もあり、または、RAは、損失を0.1%であると推定する場合もある。推定値は、接続のタイプによって決定されることができ、たとえば、推定値は、Wi-Fi接続については0.1%に設定され、セルラー接続については1%に設定され得る。RTTにおける分散など、他の方法も、RAによってパケット損失を間接的に推論するために使用されることができ、あるいは、ある実装形態は、パケット損失推定値を、それに関する情報についてオペレーティングシステムカーネルを照会することによって取得することができる。

10

【0203】

[0231]別の実装形態は、アプリケーション自体における損失を推定することができる。そうするために、その実装形態は、ネットワークソケットからのデータが概して最大セグメントサイズの(MSS)チャンクの形で受信されるが、パケット損失は、はるかに大きいチャンクを受信、すなわち近似的にTCP受信ウィンドウ全体のサイズのバーストを引き起こす、という観察に基づいた以下の手順を使用することができる。Mを、バイトで表されたMSSとすると(良好な推測は、 $M = 1500$ である)、nバイトが受信された場合、送られるパケットの数は約 n/M である。zを、 $k \cdot M$ バイトを超える読み込みを生じたソケット読み込みの数とする、ここでkは、ある小さい整数である。kは、アプリケーションの2つのネットワーク読み込みの間にk個以上のパケットが到着した見込みがないように、十分に大きいように選ばれと仮定する。ソケット上でコンスタントに待機するアプリケーションについては、 $k = 3$ がよい。すると、 $p = z \cdot M / n$ が、パケット損失確率の推定値である。所望の開始点から、zとnとをカウントすることによって、この手順は、時間の任意の所望の範囲にわたるパケット損失レートを推定することができる。

20

【0204】

[0232]RTTの推定値およびパケット損失確率が与えられると、アプリケーションは、必要とされるかなりの数の接続を計算することができる。このプロセスは特に、目標ダウンロードレートがその接続数で達成され得るように十分に大きい、接続の数を選ぶことができる。単一レートの達成可能レートは概して、達成可能スループットに関するTCP式によって制限され、この式によると、おおまかにいえば、単一のTCP接続が、 $T = MSS / (RTT \cdot p)$ の平均ダウンロードレートを達成することができる。したがって、このプロセスは、目標ダウンロードレートをTで除算したものに比例するように、接続の数を選べばよい。

30

【0205】

[0233]RAは、使用するべきTCP接続の数に対して、現実的な理由により、下限と上限とを課してもよい。たとえば、RAは、RAがオープンする接続の最大数を8に、および接続の最小数を2に制限してよい。

【0206】

[0234]帯域幅、損失確率、およびRTTは、変わることがある。要求アクセラレータは、それらの量を監視し、接続の数を動的に変える。

40

【0207】

[0235]要求アクセラレータは、HTTP要求を、より小さいサブ要求に分割し、あらゆるサブ要求についての戻されたデータ応答を、元の要求に対応するコピーレス応答に再アSEMBLすることができる。要求をサブ要求に分割するには、いくつかの利点がある。第1に、利用可能TCP接続を使用するために、それらの接続すべての上で要求を発行することが必要である。メディアストリーミングプレーヤは、接続すべてを使用するような十分な要求を与えるわけではない場合がある。要求分割は、より多くの数のサブ要求を生じ、サブ要求は様々な接続上で発行され得るので、この問題を緩和する。第2に、要求分割はより短い要求を生じ、そうすることによって時機を逸したデータ配信の危険性が

50

低減され、すなわち、一部のTCP接続が、他の接続よりも時間的に遅い場合、それらの一部の接続は、依然として短い要求とともに使用されることができる。それらは、速い方の接続よりも応答を配信するのが遅いが、要求が小さいので、全体的要求を完了するための追加的な相対遅延は通常、それ程大きくない場合がある。

【0208】

[0236]概して、より多くの接続が使用されている場合、1つの要求につき、より多くのサブ要求を作成することが好ましい。これを達成するために、要求アクセラレータは、 n 個の接続があるとき、各要求を、 n 個のサブ要求に分割することができる。

【0209】

[0237]別の実装形態は、要求サイズに応じて、1つの要求当たりのサブ要求の数を選び出す。サブ要求サイズが、ダウンロードするのに一定の時間量（たとえば、2秒）かかることが予測されるサイズとなるように選ばれた場合、要求は、より多くの接続がある場合はより多くのサブ要求に分割され、所望の効果を達成する。

【0210】

[0238]分割ルールは、不必要に小さいサブ要求がないようにすべきである。たとえば、RA実装形態は、その分割プロセスにおいて最小サブ要求サイズを課し、最小値が満たされない場合はより少ないサブ要求に分割することができる。

【0211】

[0239]マルチプルなTCP接続が使用されるとき、それらの接続は、帯域幅について競合する可能性がある。大きい時間スケールにおいて、各接続は、他の接続と同じ量を受信するが、2～3秒間など、より小さいスケールでは、一部のTCP接続は、他の接続よりも著しく遅い場合がある。これは、いくつかのサブ要求が、他の要求よりもはるかに長くかかる場合があり、それがプレイバック失速につながり得ることを含意するので、ストリーミングについての問題を起こす。

【0212】

[0240]これを避けるために、RAは、TCPフロー制御を使用して接続を「制御（tame）」すればよい。こうすることにより、各TCP接続の最大受信ウィンドウを十分に制限することができるので、接続は、その公平なスループット分を超えて使用することはない。TCP接続を介して飛行中の（送られたが、まだ受信確認されていない）データの量はおおよそ、ダウンロードレートをRTTで除算したものである。したがって、TCP受信ウィンドウが、接続についての目標ダウンロードレートを推定RTTで除算したものにおおまかに、またはそれよりもわずかに大きく設定された場合、ダウンロードレートは、目標ダウンロードレートにおおまかに、またはわずかに大きく制限される。したがって、TCP受信ウィンドウサイズの設定は、調速機（governor）として作用することができ、所与のTCP接続が、他のTCP接続にはるかに低いレートでダウンロードするよう強制したような高いレートでダウンロードすることがないようにする。そのようなメカニズムが整っているので、接続は、おおよそ同じ速度でフェッチする傾向にあり、というのは、遅い接続は、公平な分担まで速度を上げるのに利用可能な帯域幅を有するが、同時に、接続は、少なくとも総目標受信レートであり、またはそれよりもわずかに高い、総ダウンロードレートを達成し得るからである。

【0213】

[0241]RAは、受信バッファを調整することによって、クライアントにおける受信ウィンドウを調整することができる。RAは、連続する要求の間でこれらの設定を常に再調整する。

【0214】

[0242]一実装形態は、各接続のTCP受信ウィンドウを、推定RTTと、目標ダウンロードレートを接続の数で除算したものの積よりもわずかに大きく設定し得る。

【0215】

[0243]目標ダウンロードレートは、たとえば、プレイバックすることを目指すメディアレートから決定され得る。別の実装形態は、現在のプレイバックレートに基づいて（たと

10

20

30

40

50

えば、現在のダウンロードレートの2倍)目標レートを設定し得る。

【0216】

6.1 RAの実施形態

[0244]ここで、上述した要素を組み込む要求アクセラレータの実施形態について説明する。

【0217】

[0245]図30は、マルチプルなTCP接続を用いてフェッチする挙動のプロットである。図30～図31は、様々な条件下での挙動を示す。この例において、ウェブサーバへの接続は、毎秒2メガビット(「mbps」)に制限された帯域幅であり、ラウンドトリップ時間は150msであり、0.1%の packets 損失があった。4つの接続アクティブフェッチフラグメントがあった。図30～図31のプロットは、4つの接続の瞬間レート、ならびに総合レート、ならびにクライアントにおいて取得されたRTT推定値を示す。

10

【0218】

[0246]図30において、接続の受信バッファは制限されていない。図31では、受信バッファは帯域幅と遅延との積の約2倍に制限されている。

【0219】

[0247]図30および図31の例において、両方の方法は、安定して2mbpsの総スループットを達成する。接続が制限された受信ウィンドウを有するケース(図31)では、接続の間の配信は、いっそう一様であり、すなわち、ほとんどの時間、接続はほぼ同じレートで受信する。これは、無制限ウィンドウをもつ接続(図30)にはまったく当てはまらず、この場合、いくつかの接続は、長い時間ストレッチにわたって、他の接続よりも遅い。

20

【0220】

[0248]不均一な接続速度は、ストリーミングアプリケーションにとって問題があり、というのは、何らかの緊急データが、(遅い接続上で)非常にゆっくりとしか着信しておらず、帯域幅は、緊急には必要とされないデータをフェッチすることができるより速い接続の方に転用されることを意味し得るからである。

【0221】

[0249]無制限ウィンドウと制限受信ウィンドウとの間の別の違いは、クライアントが動作するときのRTTである。制限が整っているので、RTTは、低いままであり、伝播遅延に近い。受信ウィンドウ制限がないので、飛行中のデータの量が、基底の伝播遅延に接続の容量を乗じたものを超えると、キューイング遅延は非常に大きくなり、高RTTを引き起こし得る。多くのイベントに対するクライアントの反応時間は概してRTTの倍数であるので、高RTTは、メディアストリーミングクライアントにとって望ましくない。たとえば、新規メディアコンテンツをダウンロードさせるユーザ探究イベント、または要求取消もしくはリプレゼンテーションの切替えを引き起こすダウンロード速度の低減に対するクライアント反応時間は概して、現在のRTTの何倍にもなり、したがってそのようなイベントに対するクライアントの一般的応答性は、RTTが大きいとき低下する。

30

【0222】

[0250]図32は、要求アクセラレータプロセスのフローチャートである。

40

【0223】

[0251]図33は、所与のフラグメント要求を生み出すためのいくつかのサブ要求を見つけるためのプロセスを示す。

【0224】

[0252]図34は、計算されたサイズを有するソース要求の分離区間となるように選ばれた個々の要求を選択するためのプロセスを示す。このプロセスにおいて、サブ要求サイズは意図的にランダム化され、そうすることによって、接続がアイドルである時間は、接続によって変化する。こうすることにより、すべての接続が同時にアイドルになることが避けられ、その結果、より良好なチャネル使用となる。要求サイズも順序付けられ、そうすることによって、より大きい要求がより早く出され、制限されるアイドル時間の差を保つ

50

のを助ける。

【0225】

[0253] 図35は、時間オフセットと、時間オフセットによって決定される修復セグメントについてのフラグメント構造との例を示す。

【0226】

[0254] 動作に際して、要求アクセラレータは、HTTP要求（各要求は、URLおよびバイト範囲である）をSCから受信する。

【0227】

[0255] 要求アクセラレータは、HTTPを介して、要求されたバイト範囲をダウンロードし、データが完全に受信されると、SCにデータを返す。RAは、十分に大きいダウンロード速度を達成するが、同時に、各フラグメントがその最終期限時間前に必ず受信されるようにすることを目指す。高ダウンロード速度により、高品質のビデオプレゼンテーションを選ぶことが可能になるとともに、期限を尊重することによって、プレイバックが失速なしで必ず進むようになる。

10

【0228】

[0256] 高ダウンロード速度という目標を達成するために、RAは、変化する数のオープンなTCP接続を管理し、これらの接続はすべて、HTTPを介してデータを受信するのに使用される。RAは、何個の接続を使用すべきか、必要な場合はそれらの接続のオープンまたは再オープン、ならびにどのように要求を接続にディスパッチするかの詳細について引き受ける。

20

【0229】

[0257] RAは、いくつかのケースでは、ソース要求を、より小さいいわゆるRA要求に分割することを決定し、RA要求は次いで、様々な接続にディスパッチされ、その応答データは、到着すると、RAによってトランスペアレントに再アセンブルされる。たとえば、何らかのファイルの最初の64キロバイトを備えるソース要求について、RAは、2つのRA要求を作成することができ、1つはそのファイルの32キロバイトチャンクについてであり、もう1つは、第2の32キロバイトチャンクについてである。RAは次いで、それらの2つのチャンクを2つの異なる接続上で並列に要求し、2つの32キロバイトチャンクが受信されると、元の要求について、コピーレントの64キロバイト応答を作成することができる。

30

【0230】

[0258] RAは、ソース要求の単なるプレーンな部分的範囲以上のものであるRA要求を発行し得る。たとえば、プレーンなビデオデータに加え、フラグメントのFECデータについての要求を発行することができる。その場合、RAは、FEC情報が受信されると、その情報をトランスペアレントに復号し、最終的な復号フラグメントのみをソースに提示することになる。

【0231】

[0259] RAは、RA自体を、ネットワーク条件に自動調節する。たとえば、RTTが大きい場合、RAは、要求の間のたくさんのアイドル時間を避けるように、より大きいチャンク要求を発行することを決定し得る。自動調節の別の例は、RAが、その要求の適時性を確実にするように、個々の接続の速度を同様に保とうとすることである。それらのことをできるようにするために、RAは好ましくは、その接続のソケットへの直接アクセスを有する。たとえば、Unix（登録商標）のような環境では、`setsockopt()`関数を使用してソケットオプションを設定することが可能である。

40

【0232】

[0260] RAは、ネットワーク状態を測定および追跡し、これは特に、ダウンロードレートと推定ラウンドトリップ時間（RTT）とを測定することを含む。RAがこの情報を収集する理由は、第1に、接続の自動調節がそれらのアベイラビリティに依存するからであり、第2に、帯域幅情報がSMにパスされる必要があるからであり、SMは、その情報を使用してSMレート推定値を計算する。

50

【 0 2 3 3 】

[0261] R A が S M に (S C を経由して) フォワードする別の情報ピースは、未処理要求についての進行情報、すなわち、所与の要求のうちどの程度のデータがすでに受信されているかである。S M は、その情報を、S M レート推定値と、要求取消し決定の両方に対して使用する。

【 0 2 3 4 】

[0262] R A は、S M によって、帯域幅推定を行うのに必要とされる情報を追跡する。この情報は、ダウンロードに費やされる r 時間の総量、すなわち T_r 、およびダウンロードされたバイトの総量、すなわち Z である。これらの数は両方とも、単調に増加しており、S M によって頻繁にポーリングされる。 T_r タイマは、少なくとも1つの接続がアクティブである場合のみ動作している。接続は、H T T P 要求を送り出し、または応答データが着信するのを待っている場合、アクティブと見なされる。 Z カウンタは、着信バイトをカウントし、すべての接続にわたって集計される。

10

【 0 2 3 5 】

6 . 1 . 1 R A ダウンロードレート履歴

[0263] 要求アクセラレータは、履歴順に記憶される、(T_r , Z) ペアの増大アレイを保つことによって、レートの何らかの履歴を追跡する。このアレイを、 mapTrZ と呼ぶ。 mapTrZ の更新は、頻繁に、少なくとも一定時間間隔で (たとえば、100ms おきに)、場合によっては新規データが受信されたときにも、起こる。

【 0 2 3 6 】

[0264] R A は、 mapTrZ を利用して、ウィンドウ化帯域幅推定値を次のように計算することができる。幅 t である、関心のウィンドウを検討し、 $\text{mapTrZ}[\text{last}]$ を、 mapTrZ 中の最後のエン트리とする。次いで、 $\text{mapTrZ}[i].T_r$ 、 $\text{mapTrZ}[\text{last}].T_r - t$ であるような、最も大きいインデックス i を見つける。 i は、バイナリサーチで効率的に見つけられ得ることに留意されたい。レート平均は次いで、式 8 に示すようになる。

20

【 数 1 5 】

$$R := \frac{\text{mapTrZ}[\text{last}].Z - \text{mapTrZ}[i].Z}{\text{mapTrZ}[\text{last}].T_r - \text{mapTrZ}[i].T_r} \quad (\text{式 } 8)$$

30

【 0 2 3 7 】

[0265] 式 8 は、後続 T_r における差が、 t と比較して小さいと仮定する。このことは、十分頻繁にサンプリングし、小さいウィンドウ幅 t を決して選び出さないことによって保証される。

40

【 0 2 3 8 】

[0266] 実際に、任意に増大するアレイは厄介である。過去について考察される最大持続期間は上限を定められ得るので、 mapTrZ を、代わりに固定サイズのリングバッファとして実装するやり方がある。これは、次のように行われ得る。 mapTrZ アレイがアップデートされるべきであり、 mapTrZ アレイが少なくとも2つのペアをすでに含むときは常に、 $T_r - \text{mapTrZ}[\text{last} - 1].T_r < 100\text{ms}$ である場合は最後のエントリを置き換え、新規エントリを他のやり方で追加する。

【 0 2 3 9 】

6 . 1 . 2 ラウンドトリップ時間 (「 R T T 」) 推定値

[0267] R A は、帯域幅推定値を収集する。アプライオリな R T T サンプルを受け取るため

50

の簡単なやり方は、HTTP GET要求がアイドル接続上で送出され、応答が着信し始めるときの時間の差を測定することである。

【0240】

[0268]ただし、そのような測定値は、キューイング遅延を実際に含む。すなわち、クライアントが他のオープンなアクティブ接続を有する場合、クライアントにデータを送る最後のホップは、クライアントへのそのリンクが、データを受信することができるレートよりも低いレートを有する場合、いくつかのパケットをバッファリングし得る。その場合、パケットは、本来よりも長い遅延で配信され得る。

【0241】

[0269]我々のケースでは、クライアント自体の活動によって誘発されるキューイング遅延を無視するRTTを知ることが望ましい。その量の推定値を受け取るために、次のように進める。

【0242】

[0270]各活動期間中、前述したタイミング方法を用いてRTTサンプルを収集し、各GETの結果、サンプルが得られる。すると、現在の推定値は、すべてのそれらのサンプルのうち最小である。サンプルのリストは、RAが非アクティブになるときは常にフラッシュされる。(クライアントは、たとえば、高ウォーターマークのセクション3を超え、開始されたダウンロードが終了すると、非アクティブになる)。非アクティブ期間中、またはどのRTTサンプルも受信される前のアクティブ期間中、RTT推定値は、最終的な既知の推定値である。

【0243】

[0271]RTTエスティメータは、象徴的な「どのRTT推定値も既知でない」という値を戻すこともでき、この値は、たとえばクライアントスタートアップ時に使用することができる。

【0244】

6.1.3 TCP接続の数の調整

[0272]TCPフロー制御の調節により、RAは、様々な接続における帯域幅を、おおよそ同じに保つことができる。いくつかの構成可能な調節定数は、 k_R (RTT中で測定されたレート測定値ウィンドウであって、奨励値は30)と、 k_N (比例因子であって、奨励値は8192バイト)と、 N_{min} (N_{target} 下限であって、奨励値は1)と、 N_{max} (N_{target} 上限であって、奨励値は8)とを含み得る。

【0245】

[0273]推定の帯域幅遅延積 (BDP: bandwidth-delay-product) は、 $BDP := RTT \cdot R$ となるように定義され、ここでRTTは推定RTT(上記のように)であり、ここでRは、最後の $k_R \cdot RTT$ 時間(ウィンドウ方法で推定される)にわたる平均受信レートである。

【0246】

[0274]目標接続数は次いで、式9にあるように定義され、ここで k_N は構成可能定数である。

【数16】

$$N_{target} := \max(N_{min}, \min(N_{max}, \lceil BDP/k_N \rceil)) \quad (\text{式9})$$

【0247】

[0275] N_{target} の値は、定期的に計算し直される。現在オープンな接続の数が N_{target} 未満である場合、 N_{target} に一致するように、新規接続が直ちにオープンされる。一方、 N_{target} が、現在オープンな接続の数未満の場合、いかなる即時アクションもとられない。そうではなく、RA要求が終了されると常に、RAは、オープンである接続の数が多すぎるかどうかをチェックし、多すぎる場合、アイドルになったばかりの接続をクローズす

10

20

30

40

50

る。

【 0 2 4 8 】

6 . 1 . 4 接続における T C P 受信ウィンドウの調整

[0276] R A は、各接続の T C P 受信ウィンドウサイズを、

【 数 1 7 】

$$\lceil c_w \cdot BDP / N_{target} \rceil$$

【 0 2 4 9 】

に設定する。ここで、 c_w は、設定可能なハードコードされた定数であり、たとえば $c_w = 3$ である。R A は、その接続上で次の H T T P 要求を発行しようとしているときには常に接続の T C P 受信ウィンドウサイズを設定する。

【 0 2 5 0 】

6 . 1 . 5 要求分割プロセス

[0277] R A に渡された各ソース要求は、おそらく 2 つ以上の R A 要求に分割され、これらの R A 要求は各々、要求された範囲の異なる部分に対応する。所与のソース要求に対応する R A 要求がすべて完了されると、受信データは、R A によって完了フラグメントに再アセンブルされ、完了フラグメントは次いで、S C に戻される。

【 0 2 5 1 】

[0278] 所与の H T T P 要求について、R A は、いくつかの調節可能値に依存するプロセスを使用して、R A 要求の数 n を決定する。 n の値は、 T_{wn} (レート推定値ウィンドウ幅であって、奨励値は 4 s)、 D_{min} (最小フェッチ持続期間であって、奨励値は 2 s)、および c_s (R T T 中の最小フェッチ持続期間であって、奨励値は 6) という調節可能な定数に依存する。

【 0 2 5 2 】

[0279] すると、サブ要求の数 n を見つけて、所与のフラグメント要求を行うためのプロセスは、図 3 3 の擬似コードに示すようになる。

【 0 2 5 3 】

[0280] 個々の要求は次いで、たとえば、計算されたサイズを有する、図 3 4 に示すプロセスを使用して、ソース要求の分割間隔となるように選ばれる。

【 0 2 5 4 】

6 . 1 . 6 要求ディスパッチプロセス

[0281] 要求アクセラレータは、R A 要求のセットを維持する。接続が、次の要求を発行する準備ができたときには常に、キューが空いていない場合は要求が R A キューからデキューされ、アイドル接続上で発行される。キューが空の場合、新規フラグメント要求が S C から取得される。その要求は次いで、R A 要求に分割され、R A キュー上でキューイングされる。キューイングは好ましくは、サブ要求の数を見つけて所与のフラグメント要求を行うためのプロセスによってスライスが戻される順序で行われる。

【 0 2 5 5 】

[0282] H T T P 接続は、様々な理由で、たとえば、ウェブサーバタイムアウトが起こり、または単一接続上で発行され得る要求の数を越えたことが原因で、シャットダウンを受け得る。R A は、このステータスを適切およびトランスペアレントに扱うべきである。接続がシャットダウンされるときは常に、R A は、接続を自動的にオープンし直す。要求は、クローズされた接続上で要求が進行中の場合、接続からデキューされ、未受信部分についての新規 R A 要求が、R A キューの前に置かれる。

【 0 2 5 6 】

[0283] この手順により、クローズされた接続は確実に、パフォーマンスに対して最小限の影響しかもたないことになる。

【 0 2 5 7 】

10

20

30

40

50

6.1.7 特定の実施形態におけるRAパラメータ選択

[0284] TCP接続は、そのフロー制御によって制限される。すなわち、広告される受信ウィンドウは、どの時点でも受信確認されないことが許容されるデータの量に上限を定める。したがって、Wが、受信ウィンドウのサイズを示し、bdpがその接続の帯域幅遅延積を示す場合、 $bdp \leq W$ (条件1)を得る。セクション6.1.4の方法は、 $c_w > 1$ であるとする、この条件(1)が満たされるような受信ウィンドウサイズを選ぶことを記述している。この方法により、個々の接続が、利用可能帯域幅のその公平比率を実質的に超える比率をとる可能性は確実になくなる。レート増大を可能にし、レート下方スパイラルを避けるために、1よりもある程度大きい c_w 、たとえば、 $c_w = 2$ または $c_w = 4$ を選ぶことが好ましい。この値が大きい程、レートはより速くなることができ、相互に対して接続が公平でなくなる。

10

【0258】

[0285]別の制限は、TCP輻輳制御プロセスによって課される。pがパケット損失確率を示し、MがTCP最大セグメントサイズを示す場合、単一接続のレートrは、式10によって示すように制限される。

【数18】

$$r \leq \frac{M}{RTT \cdot \sqrt{p}} \quad (\text{式10})$$

20

【0259】

[0286]ここで、この式を、BDPおよび接続の数Nによって($bdp = r \cdot RTT$ と $BDP = N \cdot bdp$ とを使用する)書き換えると、式11に示すものを得る。

【数19】

$$BDP \frac{\sqrt{p}}{M} \leq N \quad (\text{式11})$$

【0260】

[0287]この式は、 k_N が、式11中の不等式が確実に成り立つように、式9中の

30

【数20】

$$M/\sqrt{p}$$

【0261】

未満のビットになるように選ばれるべきであることを示唆する。Mについての典型的な値は1キロバイトであり、 $p = 0.01$ と設定した場合、

【数21】

$$M/\sqrt{p} = 10 \text{ キロバイト}$$

40

【0262】

となる。したがって、この例では、式9におけるNを設定するために、セクション6.1.3において提案されたように $k_N = 8, 192$ バイトと設定することにより、式11の不等式が確実に満足される。受信機は、適切に構成またはプログラムされ得る。

【0263】

[0288]ここで、所与のソース要求についてのRA要求の数nを計算するための、上のセクション6.1.3のプロセスに移る。アプライオリに、小さいスライスはいくつかの利点を呈するのでスライスを可能な限り小さくしたい。そのような利点として、ある接続が他

50

の接続に比べて遅い場合、そのことが、小さい要求に関する問題を引き起こす可能性は比較的低い、というのは、小さい要求は、遅い接続上であっても迅速に終了するからである。したがって、小さいスライス設定では、遅い接続は本質的に、比較的少ない要求をサービスする結果になる。小さいスライスの別の利点は、RAに、バッファの時間内の比較的短いセクションに対して作用させるので、RAは、最も緊急な作業エリアに対するその作業を強化する傾向があることである。

【0264】

[0289]ただし、スライスを小さくすることには、代償が伴う。すなわち、第1に、各要求が、アップリンクとダウンリンクの両方においていくらかのオーバーヘッドを招く。第2に、1つの要求を終了した後、接続は、約1RTTだけ、アイドルのままである。したがって、要求分割プロセスは、理想的には、あまりにも多くのアップリンクトラフィックを引き起こすことも、各利用可能リンクの容量を実質的に利用しきれないこともない、可能な限り小さいチャックを選ぶことを試みるべきである。好ましいプロパティはしたがって、次のようになる。

【0265】

[0290]1.リアルタイムの D_{min} ごとに、1つの接続につき多くとも1つの要求を目指す。こうすることにより、アップリンクトラフィックは、最悪のケースにおける N_{target} に比例する値を限度とする。

【0266】

[0291]2. c_s . RTTおきに、1つの接続につき多くとも1つの要求を目指す。こうすることにより、接続の活動時間が、少なくとも約 $c_s / (c_s + 1)$ になるようにされ、すなわち、中間の c_s では1に近づく。

【0267】

[0292] D_{min} の良好な選択は、使用ケースに依存する。所望のエンドツーエンドの遅延と同程度（ただしそれ未満）の D_{min} を選び出すことは、一般に、フラグメントの典型的な持続期間である。エンドツーエンドの遅延が大きくなるべき場合、より大きいバッファが使用されることができ、より大きいスライスの悪影響はより小さくなる。一方、短いエンドツーエンドの遅延において、バッファは小さいので、スライスは、失速を引き起こす遅い接続を避けるように小さくあるべきである。そのシナリオにおいて、より高いコストの、より小さい要求は、バッファレベルでの、得られる安定性に値する。

【0268】

[0293]使用されるパラメータは、MPD（メディアプレゼンテーション記述）中のプロファイルインジケータが、クライアントに対する、ストリーミングされるメディアのプロパティの要約なので、そのインジケータに従って調節され得る。あらゆるメディアセグメントをダウンロードし、それらをエンドユーザに見せるのではなく、クライアントは、MPD内部のプロファイルにある様々な使用ケースに基づいて、セグメントを「スキップする」ことを選ぶことができる。

【0269】

[0294] c_s の選択に対する下限は、次のように考えればよい。N個のオープンな接続があり、RAがアクティブである場合、平均で、約 $N \cdot c_s / (c_s + 1)$ 個のアクティブな接続がある。すべてのN個の接続の受信ウィンドウが、全体として、総計目標レートを維持させるのに十分に大きくなるのを確実にするために、 $c_w \cdot c_s / (c_s + 1)$ が少なくとも1であることが望ましい。

【0270】

[0295]この限度はコンサーバティブである。アクティブ接続の推定数 $N \cdot c_s / (c_s + 1)$ は、ある程度の分散がある可能性が高くても、分散を考慮に入れていない平均にすぎない。実際に、 c_s を、上記限度によって奨励される値の約2~3倍にすることが有利であり、たとえば、 $c_w = 3$ および $c_s = 6$ のとき、 $c_w \cdot c_s / (c_s + 1)$ は少なくとも2.5である。

【0271】

10

20

30

40

50

6.2. 前方誤り訂正をもつ R A

[0296]データがいくつかの T C P 接続を介して受信されるとき、これらの接続は、時間的に異なるダウンロードレートを有することがある。フラグメントの要求がいくつかのサブ要求に分割されたとき、フラグメント全体は、最後のサブ要求応答（チャンク）が受信されたときに受信されるだけである。フラグメントが緊急に受信される必要があるとき、サブ要求のうち1つが遅い接続上で扱われる場合があり、フラグメントが素早く受信されるのを妨げるので、これは問題となる場合がある。

【0272】

[0297]コンテンツプロバイダは、ビデオデータに加え、各フラグメントについての追加の前方誤り訂正（「F E C」）修復データを与えることができ、クライアントはこのデータをフェッチして、元のフラグメントを再構築するのを助けることができる。たとえば、クライアントが、4つの接続を有し、4000バイトのサイズのフラグメントを緊急に受信する必要があると仮定する。クライアントの要求アクセラレータは、フラグメントを、各々が1000バイトからなる4つの範囲に分割し、4つの接続の各々において1つの要求を発行することができる。接続1は高速であり、接続4は適度に高速であるが、第2および第3の接続ははるかに遅い場合がある。したがって、総ダウンロードレートが、原則として、フラグメント全体を時間内にダウンロードするのに十分に高い場合であっても、フラグメントは、接続2および3が行き詰まっているので、非常に遅れてしか届かない。

【0273】

[0298]この問題を避けるために、クライアントは、それ自体のサブ要求についてデータフェッチが行われるとすぐに、接続1を使用して、接続2または3と同じデータをフェッチしようとする場合がある。これは役に立ち得るが、R A は、どの接続がより多くの助けを必要とするか、その接続が2それとも3であるか決定を行わなければならない。R A が誤った予測を行った場合、R A は、複製データを不必要にダウンロードしている場合があり、フラグメントは依然として、時間内に届くことができない。

【0274】

[0299]より優れた要求アクセラレータは、代わりに接続1を使用して、何らかの修復データをフェッチすることができる。修復（F E Cコード化された）データは、ダウンロードに成功した場合、要求2または3からのデータが欠けているかどうかにかかわらず、欠けているデータを再構築するのに使用することができる。唯一の制約は、受信されるデータの量は、フラグメントを再構築するのに十分なことである。言い換えると、我々の例では、修復バイトの数に、受信されたフラグメントバイトの数を加えると、4000以上になるはずである。

【0275】

[0300]ある実装形態では、コンテンツプロバイダは、コード化ビデオセグメント用の F E C 修復データへのアクセスを与える。プロバイダは、修復データを、元のビデオデータと同様にして、利用可能にすることができる。たとえば、各メディアセグメントファイルについて、修復情報を含む追加 F E C ファイルを与えることができる。コンテンツプロバイダは、メディアプレゼンテーション記述において F E C を使用するための、必要な情報とパラメータとを与えることができる。別の実装形態では、メディアプレゼンテーション記述は、F E C についてのいかなる情報も含まないが、クライアントは、セグメント U R L から F E C 修復 U R L の名称をどのようにして導出するかについてのルールなど、一般的な協定を使用して、その情報にアクセスすることができる。

【0276】

[0301]クライアント実装形態は、修復データをいつどのようにして要求するかに関するプロセスを実装することができる。要求される修復データの量は、どれだけ多くのデータが未処理であるかに依存し得る。さらに、どれだけすぐにフラグメントが利用可能になる必要があるかに依存し得る。たとえば、十分すぎる時間が残されている場合、ソースデータすべてを時間内に受信することを望むので、どの修復を要求することも、おそらく余分である。一方、フラグメントが緊急になりつつある場合、クライアントが、そのフラグメ

10

20

30

40

50

ントについての十分なデータを時間内に受け取るのに失敗した場合は失速が切迫している
ので、たくさんの修復データを要求したいと思うかもしれない。したがって、ある実装形
態は、要求される修復データの量を $(B)S$ となるように設定することができ、ここで
 S は未処理ソースデータの量であり、 (B) はバッファレベルの減少関数である。

【0277】

[0302] 別の実装形態は、未処理データの量を、未処理の総量ではなく、最も不完全な要
求中の未処理データの量に比例させ得る。

【0278】

6.2.1 修復セグメントジェネレータの実装形態

[0303] DASH規格がどのようにFECを使用するか、特に、FEC用のRaptor
Qを使用するかに関する、以下の算出はすべて、好ましくは、固定小数点/整数演算を使
用して実施される。この算出は、リプレゼンテーションのフラグメント内のソースシンボ
ルの数および位置を算出することを含み、修復セグメント内のフラグメント用の修復シン
ボルの数および位置を算出することは、固定小数点演算を使用して行われるべきである。
というのは、受信FEC修復フラグメントとソースフラグメントの組合せを使用してソー
スフラグメントを復号するRAプロセスとまったく同じ結果が、ソースセグメントからF
EC修復フラグメントを生じる取込み(ingestion)プロセスによって達成される必要が
あり、したがってこれらの算出はまさに同じ出力結果をもたなければならないからである
。固定小数点演算ではなく浮動小数点算出を使用すると、異なるプラットフォーム上の異
なる浮動小数点実装形態の異なるコーナーケース挙動により、追跡するのが難しく、両方
のエンドポイントがまさに同じ算出結果を生じなければならない規格において許容可能で
ない、わずかなバグのある挙動が時として生じ得る。

【0279】

[0304] 修復セグメント内のフラグメントについての修復シンボルの数と位置とを算出す
ることを伴わない、後で説明する他のすべての算出は、取込みと、正確に同じ結果を算出
するためのRAプロセスとの間に依存がないので、所望される場合、(固定小数点でも問
題はないが)浮動小数点を用いて行うことができる。

【0280】

[0305] 修復セグメントは、 sid テーブルを含む、すでに処理されたソースセグメン
トに基づいて、別個のプロセスにおいて生成することができる。ソースセグメント自体に
加え、プロセスへの2つの入力、修復比(repair fragment) R およびシンボルサイズ
 S である。セグメント内の修復フラグメントの修復シンボルの数および位置の算出に固定
小数点演算を使用することを円滑にするために、 R の値は、パーミルで表されることが
でき、すなわち、 $R = 500$ は、比が $1/2$ であることを意味する。

【0281】

[0306] 各セグメント内で、ソースセグメントの始めにおいて、時間/バイトオフセット
セグメントマップを備えるセグメントインデクシング情報も存在する。時間/バイトオフ
セットセグメントマップは、時間/バイトオフセットペア $(T(0), B(0))$ 、 $(T(1), B(1))$ 、 \dots 、 $(T(i), B(i))$ 、 \dots 、 $(T(n), B(n))$
のリストであり、ここで $T(i-1)$ は、すべてのメディアセグメントの中のメディア
の初期開始時間に対する、メディアの第 i のフラグメントのプレイバック用のセグメン
ト内の開始時間を表し、 $T(i)$ は、第 i のフラグメントについての終了時間(およびした
がって、次のフラグメントについての開始時間)を表し、バイトオフセット $B(i-1)$
は、メディアの第 i のフラグメントがソースセグメントの最初のものに相対して始まる、
このソースセグメント内のデータの最初のものに対応するバイトインデックスであり、 $B(i)$
は、第 i のフラグメントまでの、およびそのフラグメントを含むセグメント中の対
応するバイト数である(したがって、 $B(i)$ は、フラグメント $i+1$ の第1のバイトの
インデックスである)。セグメントが複数のメディア成分を含む場合、 $T(i)$ および $B(i)$
は、絶対的な方法でセグメント中の各成分に対して与えられることができ、または
、基準メディア成分をサービスする別のメディア成分に対して表されることができる。い

10

20

30

40

50

ずれの場合にも、 $B(0)$ は、セグメント中の第1のフラグメントの開始バイトインデックスであり、このインデックスは、セグメント中の第1のフラグメントに先行する $sidx$ 情報により、ゼロよりも大きいことがある。 $B(0)$ がゼロでない場合、修復セグメントの最初には、 $sidx$ に対応するいくつかの修復シンボルがある。実装形態によっては、これらの第1の修復シンボルは、第1のフラグメントの最初まで、セグメント中のデータを保護することもでき、使用されない、ゼロをパディングしたデータバイトであってもよい。

【0282】

[0307]修復比 R は、修復セグメントメタデータとともに MPD に入れてシグナリングされても、他の手段 (TBD) によって取得されてもよい。 R についての値の例として、 $R = 500$ の場合、修復セグメントサイズは、それが生成される元のソースセグメントの対応するサイズの 0.5 倍として、(非常に厳密に) 概算され、ソースセグメント内の、ソースフラグメントに対応する、修復セグメントの修復フラグメントのサイズのサイズも、ソースセグメントのサイズの 0.5 倍として (非常に粗く) 概算される。たとえば、ソースセグメントが $1,000$ キロバイトのデータバイトを含む場合、対応する修復セグメントは、ほぼ 500 キロバイトの修復データを含む。

10

【0283】

[0308] S の値は、修復セグメントメタデータとともに MPD においてシグナリングされる、または他の手段によって取得されることができる。たとえば、 $S = 64$ は、ソースデータおよび修復データが、 FEC 符号化および復号の目的のために、各々が 64 バイトのサイズのシンボルを備えることを示す。 S の値は、関連付けられたソースセグメントのリプレゼンテーションのストリーミングレートに比例するように選ぶことができる。たとえば、ストリーミングレートが 100 Kbps の場合、 $S = 12$ バイトが適切であり得、ストリーミングレートが 1 Mbps の場合、 $S = 120$ バイトが適切であり得、ストリーミングレートが 10 Mbps の場合、 $S = 1,200$ バイトが適切であり得る。1つの目標は、どの程度の粒度のフラグメントがシンボルに区分されるかと、ストリーミングレートと比較した FEC 復号についての処理要件と、の間の良好なトレードオフを有することであり得る。たとえば、 1 Mbps のストリーミングレート、および約 500 ms のサイズのフラグメントにおいて、各フラグメントは約 64 KB のデータであり、 $S = 120$ の場合、フラグメントは、ほぼ 500 個のソースシンボルからなり、このことは、各シンボルが、ソースブロックを回復するのに必要とされるデータの約 0.2% であることを意味し、このことは、シンボル粒度により必要とされる余剰受信が、フラグメントが受信されている最中の $HTTP$ 接続の数に 0.2% を乗じたものを上限とすることを意味する。たとえば、 $HTTP$ 接続の数が 6 の場合、シンボル粒度受信オーバーヘッドは、 1.2% を限度とする。

20

30

【0284】

[0309] 修復セグメントは、ソースセグメントについて次のように生成することができる。ソースセグメントの各フラグメントは、 FEC 符号化目的のためのソースブロックと見なされ、したがって各フラグメントは、修復シンボルがそこから生成されるソースブロックの一連のソースシンボルとして扱われる。第1の i 個のフラグメント用に生成された修復シンボルの総数は、 $TNRS(i) = \text{divceil}(R * B(i), S * 1000)$ として算出され、ここで $\text{divceil}(I, J)$ は、少なくとも I を J で除算したものである値、すなわち、 $\text{divceil}(I, J) = (I + J - 1) \text{div} J$ をもつ、最も小さい整数を出力する関数であり、ここで div は、結果が、最も近い整数に切り捨てられる固定小数点除算である。したがって、フラグメント i について生成される修復シンボルの数は、 $NRS(i) = TNRS(i) - TNRS(i - 1)$ である。

40

【0285】

[0310] 修復セグメントは、フラグメントについての修復シンボルの連結を備え、修復セグメント内の修復シンボルの順序は、フラグメントが生成される順序であり、フラグメント内で、修復シンボルは、それらの符号化シンボル識別子 (ESI) の順序である。

50

【 0 2 8 6 】

[0311] 上述したように、フラグメント用の修復シンボルの数を定義することによって、すべての前のフラグメント用の修復シンボルの総数、ならびにしたがって修復フラグメント i のシンボルについてのバイトインデックスおよびバイト範囲は、 R 、 S 、 $B(i-1)$ および $B(i)$ にのみ依存し、ソースセグメント内のフラグメントの前または後続構造のいずれにも依存しないことに留意されたい。これは、クライアントに、修復セグメント内の修復ブロックの開始の位置を素早く計算させ、修復ブロックがそこから生成されるソースセグメントの対応するフラグメントの構造についてのローカル情報のみを使用して、その修復ブロック内の修復シンボルの数も素早く計算させるので、有利である。したがって、クライアントは、ソースセグメントの中央からのフラグメントのダウンロードとプレイバックとを始めることを決定した場合、対応する修復セグメント内にあるフラグメントに対応する、対応する修復ブロックを素早く生成し、アクセスすることもできる。

10

【 0 2 8 7 】

[0312] フラグメント i に対応するソースブロック内のソースシンボルの数は、 $NSS(i) = \text{divceil}(B(i) - B(i-1), S)$ として算出される。最後のソースシンボルは、 $B(i) - B(i-1)$ が S の倍数でない場合、FEC符号化および復号の目的のために、ゼロバイトでパッドアウトされ、すなわち、最後のソースシンボルは、FEC符号化および復号の目的のためにサイズが S バイトになるように、ゼロバイトでパッドアウトされるが、これらのゼロパディングバイトは、ソースセグメントの一部として記憶されない。本実施形態では、ソースシンボルについてのESIは、 $0, 1, \dots, NSS(i) - 1$ であり、修復シンボルについてのESIは、 $NSS(i), \dots, NSS(i) + NRS(i) - 1$ である。

20

【 0 2 8 8 】

[0313] 本実施形態における修復セグメントについてのURLは、たとえば添字「.repair」をソースセグメントのURLに単に追加することによって、対応するソースセグメントについてのURLから生成することができる。

【 0 2 8 9 】

[0314] 修復セグメントは、たとえば、末尾に付加される、対応するソースセグメントの一部でもあり得る。組み合わせられたセグメントの構造は、ソースフラグメントおよび修復フラグメントが、組み合わせられたセグメント内で連続し、すなわち、組み合わせられたセグメントが、第1のソースフラグメント、それに続く第1の修復フラグメント、それに続く第2のソースフラグメント、それに続く第2の修復フラグメントなどを備えるようなものでもあり得る。当業者には認識されるように、上述した方法およびプロセスは、そのような組み合わせられたセグメントに適用するために、容易に採用することができる。

30

【 0 2 9 0 】

6.2.2 修復セグメントを使用する要求アクセラレータの実施形態

[0315] 修復セグメントについての修復インデクシング情報およびFEC情報は、対応するソースセグメントについてのインデクシング情報によって、 R および S の値から暗黙的に定義され、ここで R は、パーミルを示す 0 と 1000 との間の整数として表され、 S は、バイトで表される。時間オフセットおよび修復セグメントを備えるフラグメント構造は、対応するソースセグメントの時間オフセットおよび構造によって決定される。フラグメント i に対応する修復セグメント中の修復シンボルの冒頭および末尾へのバイトオフセットは、それぞれ、 $RB(i-1) = S * \text{divceil}(R * B(i-1), S * 1000)$ および $RB(i) = S * \text{divceil}(R * B(i), S * 1000)$ として算出され得る。フラグメント i に対応する修復セグメント中のバイト数はすると、 $RB(i) - RB(i-1)$ であり、したがってフラグメント i に対応する修復シンボルの数は、 $NRS(i) = (RB(i) - RB(i-1)) / S$ として算出される。(分子が S の倍数であることが保証されるので、ここでは divceil 演算は必要ないが、 divceil はここで使用することができ、結果は依然として正しくなることに留意されたい)。フラグメント i に対応するソースシンボルの数は、 $NSS(i) = \text{divceil}(B(i)$

40

50

) - $B(i - 1), S$)として算出することができ、ここで最後のソースシンボルは、必要な場合、符号化について記述したのと同じく、復号目的のためにゼロでパディングされる。したがって、修復セグメント内の修復ブロックについての修復インデクシング情報、および対応するFEC情報は、対応するソースセグメントの対応するフラグメントについてのR、Sおよびインデクシング情報から暗黙的に導出することができる。

【0291】

[0316]一例として、バイトオフセット $B(1) = 6, 410$ で始まり、バイトオフセット $B(2) = 6, 770$ で終わるフラグメント2を示す、図35に示す例を検討し、すなわち、フラグメント2はサイズが $6, 770 - 6, 410$ バイトであり、 $6, 770$ は、フラグメント3の開始バイトインデックスである。この例では、シンボルサイズは $S = 64$ バイトであり、垂直点線は、 S の倍数に対応する、ソースセグメント内のバイトオフセットを示す。ソースセグメントサイズの割合としての全体的な修復セグメントサイズは、この例では $R = 500$ パーミルに設定される(修復は、ソースのほぼ $1/2$ である)。フラグメント2用のソースブロック中のソースシンボルの数は、 $NS S(2) = \text{div ceil}(6, 770 - 6, 410, 64) = (6, 770 - 6, 410 + 64 - 1) \text{div } 64 = 6$ として算出され、これらの6つのソースシンボルは、それぞれESI0、...、5を有し、ここで第1のソースシンボルは、ソースセグメント内のバイトインデックス $6, 410$ で始まるフラグメント2の最初の64バイトであり、第2のソースシンボルは、ソースセグメント内のバイトインデックス $6, 474$ で始まるフラグメント2の次の64バイトであり、以下同様である。フラグメント2に対応する修復ブロックの終了バイトオフセットは、 $RB(2) = 64 * \text{div ceil}(500 * 6, 770, 64 * 1, 000) = 64 * (3, 385, 000 + 64, 000 - 1) \text{div } 64, 000 = 64 * 53 = 3, 392$ として算出され、フラグメント2に対応する修復ブロックの開始バイトオフセットは、 $RB(1) = 64 * \text{div ceil}(500 * 6, 410, 64 * 1, 000) = 64 * (3, 205, 000 + 64, 000 - 1) \text{div } 64, 000 = 64 * 51 = 3, 264$ として算出され、したがってこの例では、それぞれ、修復セグメント内のバイトオフセット $3, 264$ で始まり、バイトオフセット $3, 392$ で終わる、ESI6および7をもつフラグメント2に対応する修復ブロック中に2つの修復シンボルがある。

【0292】

[0317]このことは、図35に示されている。図35に示すこの例において、 $R = 500$ である(修復は、ソースのほぼ $1/2$ である)とともに、フラグメント2に対応する6つのソースシンボルがあるとしても、修復シンボルの数を算出するのにソースシンボルの数を単に使用している場合に期待し得るように、修復シンボルの数は3ではなく、2となるように算定されることに留意されたい。修復シンボルの数を決定するのに、フラグメントのソースシンボルの数を単に使用するのは反対に、ここで行われるやり方は、対応するソースセグメントの対応するソースブロックに関連付けられたインデックス情報のみから、修復セグメント内の修復ブロックの位置決めを算出することを可能にする。これを、取込みプロセスにおける、およびRAプロセス内での無矛盾の算出にするために、修復セグメント内の修復フラグメントについての修復シンボルの数および位置の算出が、固定小数点演算を使用して算出されることが重要である。さらに、ソースブロック中のソースシンボルの数、すなわち K が増大すると、対応する修復ブロックの修復シンボルの数、すなわち KR は、概して、 KR は多くとも $\text{div ceil}(K * R, 1, 000)$ であり、 KR は、少なくとも $\text{div floor}((K - 1) * R, 1, 000)$ であるので、 $K * R / 1, 000$ によって厳密に概算され、ここで $\text{div floor}(I, J) = I \text{div } J$ である。

【0293】

7. 例示した実施例

[0318]図25は、レート選択プロセスを示す。および μ についての設定が高い程、設定はアグレッシブになる。図23は、パラメータについての様々な値を示す。図24は

、パラメータ μ についての様々な値を示す。ハイブリッド設定が、2つのメインメカニズムによって、レートゆらぎを削減しようとする。第1のメカニズムは、Bが比較的大きいとき、レートを増大させることにより慎重であることによるものであり、第2のメカニズムは、Bが比較的小さいとき、より懸命に、現在のレートに留まろうとするものである。

【0294】

[0319] $pker_{x,y} : C = x * \min(y * Td1, B)$ についての例示的設定は、8.1、4.2、2.4、4.4または他の x, y 値に設定された x, y であり得る。 $pker$ の実際の平均化ウィンドウは、ダウンロード中断期間のスキップにより、Cよりも長いことに留意されたい。EWMAのスキップはなく、ダウンロード中断期間中のレートは、最後のダウンロード間隔のものと同じであると仮定する。

10

【0295】

[0320] MWA (移動ウィンドウ平均) に対して、 $H(z) = (1/D) * ((1 - z^{-D}) / (1 - z^{-1}))$ であり、ここでDはウィンドウサイズである。 $X_i = \min\{R_k : k \leq i\}$ であり、ここで R_k は、重み W_k をもつレートのEWMAであり、 $W_1 < W_2 < W_3 < \dots$ である。EWMAについて、 $H(z) = ((1 - \alpha) / (1 - \alpha z^{-1}))$ であり、ここで α は前の平均の重みである。MWAおよびEWMAは、いくつかのケースではおおよそ均等である。

【0296】

[0321] 適合型エスティメータは、より長い平均化ウィンドウを有する場合、ライブストリーミング用にほぼ同じ平均レートを維持したまま、レート切替え頻度を削減する。異なる設定が、異なるシナリオについてうまく機能する。アグレッシブ設定は、より安定したシナリオに対してうまく機能し、あまりアグレッシブでない設定は、より変動しやすいシナリオにより適している。帯域幅が、かなりの時間部分(たとえば、20秒間の平均がレートキャップよりも高い時間の%)に対して、特定のマージンだけ、最も高いリプレゼンテーションレートよりも高い場合、よりアグレッシブ設定で進めることが有益である。理想的には、デバイスは、シナリオタイプを検出し、適切な設定を適用することができるべきである。シナリオ検出は、無線技術タイプ、特定の単位時間内のレート変化の回数、移動速度などのような要因に基づき得る。より単純な戦略は、上記観察に基づき、すなわち、「全体的」帯域幅がレートキャップよりも高いとき、よりアグレッシブ設定を使用することができる。

20

30

【0297】

8. レート選択パラメータの設定

[0322] このセクションでは、レート選択パラメータを設定する例が挙げられる。

【0298】

[0323] MLBについて、 $EFF = 1 - Rv / Rd1$ であり、ここで Rv は、選択されたリプレゼンテーションの現在のレートであり、 $Rd1$ は現在のダウンロードレートである。提案されるルールは、次の通りである。

[0324] $EFF < 0$ の場合、おそらく2つ以上のレートだけ下がる

[0325] $0 < EFF < 0.1$ の場合、1レート下がる

[0326] $0.1 < EFF < 0.6$ の場合、現在のレートに留まる

[0327] $0.6 < EFF < 0.8$ の場合、1レート上がる

[0328] $0.8 < EFF < 1$ の場合、おそらく2つ以上のレートだけ上がる

[0329] $\alpha = Rv / Rd1$ とする。するとこれは、おおよそ次のようになる。

40

[0330] $\alpha < 0.4$ の場合、少なくとも1つのレートだけ上がる

[0331] $0.4 < \alpha < 0.9$ の場合、同じレートに留まる

[0332] $0.9 < \alpha$ の場合、少なくとも1レートだけ下がる

[0333] これを、DASHクライアントレート選択プロセスのコンテキストに当てはめる。

【0299】

50

[0334] RUPを、UPに対応するリプレゼンテーションのレートとし、RDOWNを、DOWNに対応するリプレゼンテーションのレートとし、上記のように、Rvを、現在選ばれているリプレゼンテーションのレートとする。RUPは、 $RUP = \lambda(t) * Rd1$ となるように可能な限り大きくなるように選ばれ、そのRDOWNが、 $RDOWN = \mu(t) * Rd1$ となるように可能な限り大きくなるように選ばれる。パラメータ $t = B / (D + \Delta)$ であり、ここでBは、メディアバッファにおける現在のプレゼンテーション時間量であり、Dは、現在の決定が行われている時点を超える、次の可能切替え点までの時間に対する限度であり、 Δ は、ネットワーク待ち時間とラウンドトリップ時間とを考慮に入れる小さいパラメータであり、たとえば、 Δ は、近似として1秒または2秒に設定されてもよく、 Δ は、現在のRTTに対する測定された上限に従って設定されてもよい。

10

【0300】

[0335] 次のレートRNEXTの全体的選択は、次のようになる。

【0301】

[0336] $RUP < Rv$ の場合、 $RNEXT = \min\{Rv, RDOWN\}$ であり、そうでない場合、 $RNEXT = RUP$ である。

【0302】

[0337] 上記MLBパラメータは、すべてのtについて $\lambda(t) = 0.4 * R$ および $\mu(t) = 0.9$ と設定することによって概算することができ、ここでRは、次に高いリプレゼンテーションのレートと、現在のリプレゼンテーションのレートのものとの比である。たとえば、現在のレートが500Kbpsであり、次に高いレートが750Kbpsである場合、 $R = 1.5$ となり、したがって $\lambda(t) = 0.6$ となる。これは、MLBプロセスを次のように近似する。

20

【0303】

[0338] 決定点において、 $EFF = 0.6$ 、すなわち、 $\alpha = 0.4$ の場合、 $Rv < 0.4 * Rd1$ であり、この場合、(すべてのtについて $\lambda(t) = 0.4 * R$ であるので) RUPは少なくとも $Rv * R$ となり、したがって $RNEXT = RUP$ となり、すなわち、レートはレート $Rv * R$ において次に高いリプレゼンテーションまで上がってよく、 $Rd1$ が $0.4 * Rv$ よりもはるかに大きい場合、RUPは、(リプレゼンテーションレートの粒度に依存して) $Rv * R$ よりも大きくなり、RUPは、 EFF がたとえば 0.8 よりも大きい場合、 $Rv * R$ を1レート分超えることになる。 $EFF < 0.1$ の場合、 $Rv > 0.9 * Rd1$ であり、この場合、RDOWNは、 Rv 未満になり ($RDOWN = 0.9 * Rd1$ なので)、レートは下がり、すなわち、 $RNEXT < Rv$ になる。 EFF が 0.1 と 0.6 との間の場合、 $RUP = Rv * R$ および $RDOWN = Rv$ になり、この場合、 $RNEXT$ は、 Rv と等しくなるように選ばれる。

30

【0304】

9. レート選択パラメータセット

[0339] 以下のテーブルは、いくつかの可能なレート選択パラメータセットを指定する。以下のテーブルに示されていないtの中間値についての λ および μ の値は、周辺値の間を線形補間することによって算出されるべきである。以下のテーブルに示すものを超えるtの値についての λ および μ の値は、示されているtの最大値についての λ および μ 値に設定されるべきである。

40

【0305】

[0340] $\mu(t) < t$ および $\lambda(t) < t$ という制約がすべてのtについて満たされる場合、理論的には、プレイバックにおける失速はないが、現実的な点からは、まったく失速しないがはるかに削減されたレートでプレイアウトし続けるよりもむしろ、プレイバックにおいて小さい失速を有することが好ましい場合があり、たとえば、1Mbpsから20Kbpsに急変するのは、その間に1秒間の休止があつて1Mbpsから250Kbpsに急変するよりも悪い経験であり得る。 λ および μ の最小値は、図36のテーブルにおいて設定され、 $\mu(t) > t$ および/または $\lambda(t) > t$ の値に

50

ついて、失速が起こる見込みが高いことを注記しておく（ただし、失速は、ラムダ（ t ）およびミュー（ t ）の設定によらず、バッファが空であるときは、いずれの場合にも起こり得る）。

【0306】

[0341]ここまで説明してきたように、クライアントデバイスは、HTTPを介した適合型ビデオストリーミングのためのレート適合プロセスとダウンロードプロセスとを提供することができる。インターネット（および他のネットワーク）を介してビデオをストリーミングするクライアントは、変動帯域幅の問題に直面する。高品質ビデオがストリーミングされる場合、リンクが十分に高速でないときがあり、プレーヤに割り込ませ、再バッファリングさせる。他のケースでは、低品質ビデオは、はるかに少ない帯域幅を使用するが、より劣ったユーザエクスペリエンスとなる。1つの解決策は、ビデオ品質を適応的に調整することであり、すなわち、スループットが高いときは、より良好な品質を選び、自動的に下方切替えることである。

10

【0307】

[0342]ただし、適応ビデオストリーミングは、いくつかの課題を提起する。すなわち、（1）ビデオレート（品質）を選ぶためのプロセスまたはアルゴリズムは、レート下落ならびにレート増大に適応するように、十分に素早く作用するべきである。同時に、早まった、または一貫性のない決定を避け、不必要なレート切替え決定を避けるべきである。クライアントは、高品質ビデオが達成され得るように、十分に高いレートでデータをフェッチすることを目指すべきである。同時に、ダウンロードプロセスは、データが適時に受信されることを確実にすべきである。各フレームは、プレイアウトされる前に全体が受信されるべきである。これらの目標を、不必要に大きいプレイバックバッファを必要とせずに達成することができるべきである。大きいバッファのいくつかの問題は、ライブイベントの場合、バッファ中のビデオの量が、目標エンドツーエンド遅延によって制限され、これらのケースにおける可能プレイバックバッファを厳しく制限することである。また、大きいバッファへの依存は、バッファが、事前充满される必要があるため、プレイバック開始またはシーク時に望ましくない遅延を引き起こし得る。また、大きいプレイバックバッファは、たくさんのメモリを使い、モバイル電話および他のクライアントデバイスではメモリが乏しい場合がある。

20

【0308】

[0343]これらの問題を解決するために、受信レート変化に素早く反応するレート推定のためのプロセス。レート推定は、ビデオをストリーミングする際の使用のために特別に調整された、適応ウィンドウ化平均（adaptive windowed average）であることができる。レートエスティメータは、ウィンドウ幅（およびしたがって測定値分散）を大きく保ちながら、必要な場合にレートが十分に高速に調整することを保証するような形で、ビデオバッファレベルと、ビデオバッファレベルの変化とを考慮に入れる。プロセスによって与えられる保証は、（a）Bが、レート下落が起きたときのバッファ中のビデオデータの量（プレイバック時間の秒数）である場合、エスティメータが、バッファがB/2までドレインするのにかかる時間内にそのレート推定値を調整してしまうこと、および（b）Bが、レート増大が起きている間のバッファ中のデータの量である場合、レートエスティメータが、原則として多くとも $3 * B$ の時間内に見られ得るように、新規レートに十分に迅速に調整する（スマートなレート変更プロセスを仮定する）ことである。

30

40

【0309】

[0344]レート決定プロセスは、（a）バッファが低レベルにあるときにバッファが満たされ、（b）小さいダウンロードレート推定値が観測される場合であっても、不規則に変化するレートを回避するためにバッファを使用し、（c）安定レートシナリオでは、正しい安定レートを素早く選ぶように、レート決定を行うことができる。（a）正確なレート推定を可能にし、（b）ネットワーク遅延およびパケット損失レートが高い場合であってもリンク容量を達成することが可能であり、（c）ストリームの適時配信を達成する、HTTPのためのマルチメディアダウンロード戦略が使用される。これを達成するために、

50

マルチプルなHTTP接続を使用し、ネットワーク条件に依存して、メディア要求をより小さいチャンク要求に分解し、TCPフロー制御メカニズムを使用して接続を同期させ、データをバースト的に要求することができる。また、接続をビジーに保つために、HTTPパイプラインングプロセスを使用することもできる。

【0310】

[0345]いくつかの特徴、態様および詳細について、今まで説明してきた。説明したように、様々な実施形態において、方法ステップは、対応するプログラムされた要素、プロセッサに与えられる命令、ハードウェアまたは当業者には明らかであろう他の装置によって実施することができる。同様に、要素は、プロセスまたはプログラム要素によって有効にすることができる。実施形態の要素の構造は、プロセッサによって実行されるが、本明細書では対応する方法ステップとして記述される命令セットを単に備え得る。

10

【0311】

[0346]様々な実施形態において、ダウンロードレート加速は、使われても使われなくてもよい。ダウンロードレート加速の例は、TCP接続を介したHTTP要求を使用することによってダウンロードを加速する方法または装置である。TCP接続は、特定のウィンドウサイズを有し、TCP接続の端部にあるノードは、ウィンドウサイズについての設定を変えることができる。1つの新規性は、連続するHTTP要求用にウィンドウサイズを設定することであり、ここでサイズは、目標ダウンロードレートの関数である。したがって、目標ダウンロードレートが変わると、TCPウィンドウサイズが変わり得る。

【0312】

20

[0347]一実施形態では、ネットワーク経路によって結合されたソースと受信機との間のネットワーク経路を介したデータダウンロードを制御するための方法および/または装置もしくはコンピュータ可読媒体が使われ、この方法は、ソースと受信機との間の複数のTCP接続の各々について、そのTCP接続用のTCP受信機ウィンドウサイズを決定することによって、ソースと受信機との間のTCP接続が直接接続または間接接続であることができる、決定することと、メディアコンテンツ用の目標ダウンロードレートを決定することによって、目標ダウンロードレートが、少なくとも2つの連続するHTTP要求についての少なくとも2つの値の間で変わる、決定することと、複数のTCP接続の各TCP接続を使用して、ダウンロードされるべきメディアコンテンツの複数のメディアデータをダウンロードすることとを備え、メディアコンテンツが、複数のHTTP要求への応答の一部または全部であり、所与のTCP接続用の決定されたTCP受信機ウィンドウサイズは、目標ダウンロードレートに少なくとも部分的に基づいて決定され、決定されたTCP受信機ウィンドウサイズは、少なくとも2つの連続するHTTP要求についての少なくとも2つの値の間で変わる。

30

【0313】

[0348]現在のTCP接続のための決定されたTCP受信機ウィンドウサイズは、乗数レートで乗算された、現在のTCP接続についての現在の推定ラウンドトリップ時間(「ERTT」)の積に少なくとも部分的に基づいて決定されることができ、乗数レートは、現在のTCP接続用の目標ダウンロードレートと、目標ダウンロードレートよりも所定の量だけ高いレートとによって制限される範囲内である。現在のERTTは、たとえば1秒、10秒、50秒など、直前の測定期間にわたる最小の観測されたRTTの測度によって決定されることができ、現在のERTTは、休止期間の終了における測度によって決定されることができ、休止期間は、ダウンロード期間に続く、TCP接続にわたるアクティブHTTP要求が所定の継続時間期間だけ存在していない期間である。目標ダウンロードレートは、使用されるすべてのTCP接続にわたる現在の総ダウンロードレートを、使用されるTCP接続の数で除算したものに比例し、たとえば現在の総ダウンロードレートの2倍または3倍になり得る。目標ダウンロードレートは、メディアコンテンツのプレイバックレートに比例し得、プレイバックレートは、使用されるすべてのTCP接続に及ぶ総計にわたるレートを、使用されるTCP接続の数で除算したものである。各メディアデータ要素は、所定の分散範囲内のサイズを有するいくつかのチャンクに分割され、そのような

40

50

チャンクの数、使用されるTCP接続の数に基づく。そのようなチャンク数は、現在のTCP接続についての現在の推定ラウンドトリップ時間(「ERTT」)、現在のダウンロードレート、および/または要求されるメディアフラグメントのサイズのうち少なくとも1つにさらに基づき得る。所定の分散範囲はゼロであってよく、したがって各チャンクは、フラグメント要求ごとに同じサイズを有し、チャンク数は、使用されるTCP接続の数に、所定の因子を乗じたものに等しい。各チャンクは、最小バイト数以上のサイズを有し得る。後続メディアデータ要素についての後のHTTP要求が、第1の利用可能TCP接続に割り当てられ得る。

【0314】

[0349]制御することはまた、ソースと受信機との間で使用するべきいくつかのTCP接続を決定することであって、その数が1よりも大きく、使用するべきTCP接続の数が、少なくとも部分的に、決定された少なくとも1つのネットワーク条件に基づいて決定される、決定することと、いくつかのTCP接続の各々を使用して、ダウンロードされるべきメディアコンテンツの複数のメディアデータ要素をダウンロードすることを含み、メディアコンテンツは、複数のHTTP要求への応答の一部または全部である。使用されるTCP接続の数は、TCP接続についての推定ラウンドトリップ時間(「ERTT」)、目標ダウンロードレート、および損失レートの推定値に基づき得る。損失レートは、1%または0.1%であると推定され得る。使用するべきTCP接続の数は、両端値を含む2と16の間であり、および/または(a)目標ダウンロードレート、(b)ERTT、および(c)推定損失レートの平方根の積に比例し得る。TCP接続の各々について、TCP受信機ウィンドウサイズは、そのTCP接続用に、目標ダウンロードレートに基づいて決定することができ、決定されたTCP受信機ウィンドウサイズは、少なくとも2つの連続するHTTP要求についての少なくとも2つの値の間で変わる。

【0315】

[0350]一実施形態では、プレゼンテーションバッファに注目し、バッファがどの程度大きいか/満杯であるか/空であるか、すなわち、バッファのレベルがどこにあるかに基づいて、ダウンロードレートの推定を行うダウンロードレートを推定するための方法および/または装置もしくはコンピュータ可読媒体が使用される。たとえば、有限帯域幅を有するネットワーク経路によってデータソースに結合された受信機においてダウンロードレートを推定することであって、ダウンロードレートが、受信機においてネットワーク経路を介してデータが受信され得るレートであることは、受信機のプレゼンテーションバッファを監視することであって、プレゼンテーションバッファが、少なくともメディアデータが受信される時間と、受信機に関連付けられたプレゼンテーション要素によってメディアデータが消費される時間との間メディアデータを記憶する、監視することと、ダウンロードレートの推定値がそれに基づくべき非ゼロ推定期間を決定することと、推定期間にわたるバッファレベルのインジケーションを記憶することであって、所与の時間におけるバッファレベルが、プレゼンテーションバッファのうちどの程度が、少なくとも近似的に、受信されたがプレゼンテーション要素によってまだ消費されていないメディアデータによってその時間に占有されるかに対応する、決定することと、記憶されたインジケーションを、推定ダウンロードレートの測度の一部として使用することとを備え得る。

【0316】

[0351]プレゼンテーション要素は、ディスプレイとオーディオ出力とを備えることができる。推定期間は、所定の比例因子による、測定されたバッファレベルに比例する持続期間を有し得る。推定期間の持続期間は、測定時間におけるプレゼンテーションバッファ中の未消費のメディアデータのバイト数に比例し、および/またはプレゼンテーションバッファにメディアが追加される追加レートの関数であり、および/またはプレゼンテーションバッファの所定の部分をダウンロードするのに使用される時間に比例するようにとられ得る。所定の持続時間は、プレゼンテーションバッファのコンテンツの所定の比率がダウンロードされた持続時間に対応して得る。推定期間は、プレゼンテーションバッファのコンテンツの所定の比率がダウンロードされた時間、およびメディアデータがプレゼンテ

ションバッファ中に存在するプレゼンテーション時間のうち、より短い方であることができる。

【0317】

[0352]一実施形態では、プレイバックレート選択のための方法および/または装置もしくはコンピュータ可読媒体が使用され、プレイバックレートは、メガビット/秒などのメモリ単位/時間で測定される、プレゼンテーションバッファからメディアが消費されるレートである。受信機が、いくつかのメディアについての要求を行うとき、そのメディアのためのプレイバックレートがある。おそらく常にはないがしばしば、より高品質のメディアが、より高いプレイバックレートを有し、したがってトレードオフを提示する。どのプレイバックレートを使用/要求するべきかは、少なくとも時には、どれだけ多くのメディアがプレゼンテーションバッファにあるかに応じて決まる。受信機は、受信機のプレゼンテーション要素を使用してプレイアウトするためのメディアを受信することもでき、ここで、プレイアウトの結果、メディアがプレゼンテーションバッファからあるプレイバックレートで消費されることになり、受信機は、複数のプレイバックレートから選択を行うように構成され、プレゼンテーションバッファを監視することであって、プレゼンテーションバッファは、少なくともメディアデータが受信される時間と受信機に関連付けられたプレゼンテーション要素によってメディアデータが消費される時間との間メディアデータを記憶する、監視することと、バッファレベルのインジケーションを記憶することであって、バッファレベルは、プレゼンテーションバッファのどの程度が、受信されたがプレゼンテーション要素によってまだ消費されていないメディアデータによって占有されるかに
10
20

【0318】

[0353]選択されたプレイバックレートは、推定ダウンロードレートの所定の乗数以下であることができ、所定の乗数は、バッファレベルの増加関数である。所定の乗数は、プレゼンテーションバッファ中のメディアデータのプレイバック持続時間のアフィン一次関数であることができ、所定の乗数は、プレゼンテーションバッファのバッファレベルがしきい値未満であるときには、1未満であることができる。所定の乗数は、プレゼンテーションバッファ中のメディアデータのプレゼンテーション持続時間が、あらかじめ設定された最大量のプレゼンテーション時間以上であるときには、1以上であることができる。所定の乗数は、プレゼンテーションバッファ中のメディアデータのプレイバック持続時間の区分的一次関数であることができる。選択されたプレイバックレートは、推定ダウンロードレートの所定の乗数、およびプレゼンテーションバッファ中のメディアデータのバイト数の増加関数の所定の乗数以下であり得る。プレイバックレートは、比例因子にダウンロードレート推定値を乗じたもの以下である複数のプレイバックレートのうち最も大きい利用可能プレイバックレートとなるように選択されればよく、比例因子は、プレゼンテーションバッファ中のメディアデータのプレイバック持続時間を、レート変化に対する反応時間の推定値で除算したものの増加関数である。反応時間は、メディアデータ中の切替え点の間のプレゼンテーション時間に対する上限であり得、および/または反応時間の推定値は、メディアデータ中の切替え点の間のプレゼンテーション時間に関する平均であり得る。反応時間の推定値は、所定の定数に推定ラウンドトリップ時間(「E R T T」)を乗じたもの以上であり得る。
30
40

【0319】

[0354]受信機のプレゼンテーション要素を使用してプレイアウトするためのメディアを受信する受信機であって、プレイアウトの結果、メディアがプレゼンテーションバッファからプレイバックレートで消費され、受信機は、複数のプレイバックレートから選択するように構成され、プレゼンテーションバッファを監視することであって、プレゼンテーションバッファは、少なくともメディアデータが受信される時間と、受信機に関連付けられ
50

たプレゼンテーション要素によってメディアデータが消費される時間との間のメディアデータを記憶する、監視することと、バッファレベルのインジケーションを記憶することであって、バッファレベルが、プレゼンテーションバッファのうちどの程度が、受信されたがプレゼンテーション要素によってまだ消費されていないメディアデータによって占有されるかに対応する、記憶することと、バッファレベルの許容分散を決定することと、記憶されたバッファレベルインジケーションとバッファレベルの許容分散とを使用して、目標プレイバックレートを計算することと、目標プレイバックレートに従って複数のプレイバックレートの中から選択することと、を行うための方法または装置を備え得る。

【0320】

[0355]プレイバックレートは、上位比例因子 (upper proportional factor)、下位比例因子 (lower proportional factor)、ダウンロードレート推定値、現在のプレイバックレート、バッファレベル、およびレート変化に対する反応時間の推定値に基づいて選択することができる。上位比例因子および下位比例因子は両方とも、プレゼンテーションバッファ中のメディアデータのプレイバック持続時間を、レート変化に対する反応時間の推定値で除算したものの増加関数および/または区分的一次関数であることができ、上位比例因子は、下位比例因子よりも大きいものまたは下位比例因子に等しいものである。プレイバックレートは、前のプレイバックレートが、下位比例因子に推定ダウンロードレートを乗じたものと、上位比例因子にダウンロードレート推定値を乗じたものとの間にあるときには、前のプレイバックレートと同じになるように選択されることができる。プレイバックレートは、前のプレイバックレートが、上位比例因子にダウンロードレート推定値を乗じたものを上回るときには、上位比例因子に推定ダウンロードレートを乗じたもの以下の最も大きい利用可能プレイバックレートになるように選択されることができる。プレイバックレートは、前のプレイバックレートが、下位比例因子にダウンロードレート推定値を乗じたものを下回るときには、下位比例因子に推定ダウンロードレートを乗じたもの以下の最も大きい利用可能プレイバックレートになるように選択されることができる。

【0321】

[0356]一実施形態では、要求を行うためであるが、プロセス要求において取り消すかどうか決定するためでもある方法および/または装置もしくはコンピュータ可読媒体が使用される。受信機は、メディアのセグメント/部分/フラグメントについての要求を行い、要求に対する応答を受信し、応答からのメディアを記憶し、可能性として別の要求を行うとき、要求を取り消し、異なる要求を発行すること好ましいと決定してよい。メディアのプレイバックレートは、最もアグレッシブであるとともに、プレゼンテーションバッファ中のメディアを、消費されるときに使い果たすことなく取得することを期待する最も高いプレイバックレートを選択する受信機によって決定することもできる。ダウンロードレートが予期せぬほど下がった場合、受信機は、その現在の要求を取り消し、より低いプレイバックレートメディアについての新規要求を行うか、それとも現在の要求をプレイアウトさせるか決定する。高プレイバックレート要求を取り消し、より低いプレイバックレート要求で置き換えると、プレゼンテーションバッファのコンテンツがより長く続くことになる場合があるが、要求ミッドストリームを取り消すと、その要求についてのどの部分的に受信されたメディアの損失も引き起こす場合がある。

【0322】

[0357]そのような一実施形態において、受信機は、受信機のプレゼンテーション要素を使用してプレイアウトするためのメディアを受信し、プレイアウトの結果、メディアは、プレゼンテーションバッファから一定のプレイバックレートで消費され、受信機は、複数のプレイバックレートから選択をするように構成される。要求アクションを決定することは、プレゼンテーションバッファを監視することであって、プレゼンテーションバッファが、少なくともメディアデータが受信される時間と、受信機に関連付けられたプレゼンテーション要素によってメディアデータが消費される時間との間、メディアデータを記憶する、監視することと、バッファレベルのインジケーションを記憶することであって、バッファレベルが、プレゼンテーションバッファのうちどの程度が、受信されたがプレゼンテ

10

20

30

40

50

ーション要素によってまだ消費されていないメディアデータによって占有されるかに対応する、記憶することと、選択された第1のメディアデータチャンクをダウンロードするために、および発行された要求が未処理のときに、発行された要求の状態を維持することと、ネットワーク条件および発行された要求の状態に基づいて、要求を続けるか、それとも要求を取り消すかを決定することとを備える。

【0323】

[0358]要求を続けるか、それとも要求を取り消すかを決定することは、第1のメディアデータがブレイアウトされる前に、要求についてのダウンロードを完了するのに十分な時間があるかどうか決定することと、十分な時間がない場合、要求を取り消すこととを備えることができる。要求を続けるか、それとも要求を取り消すかを決定することは、選択された第1のチャンクまたは選択された第2のチャンクのいずれかがブレイアウトされることになる前に、より高レートの第2のチャンクをダウンロードするのに十分な時間があるかどうか決定することと、十分な時間がある場合、要求を取り消し、第2のチャンクについての要求を発行することとをさらに備えることができる。要求を続けるか、それとも要求を取り消すかを決定することは、ダウンロードレートおよびメディア消費レートに基づいて、失速が起ることを検出することと、プレゼンテーション要素が、消費されるメディアによって指示されたレートでメディアデータを消費することができない時間と、プレゼンテーション要素が、消費されるメディアによって指示されたレートでメディアデータの消費を再開することができる時間との間の失速期間を推定することと、継続または取消しが失速期間に対して与える効果を決定することと、要求の取消しが失速期間を短縮することになる場合、要求を取り消すこととをさらに備えることができる。

【0324】

[0359]他の特徴は、第2のメディアデータチャンクを選択することであって、第2のメディアデータチャンクが、開始プレゼンテーション時間を有し、その開始プレゼンテーション時間が、第1のメディアデータチャンクと同じ開始プレゼンテーション時間である、選択することと、第2のメディアデータチャンクのダウンロードを要求することと、第2のメディアデータチャンクを選択することであって、第2のメディアデータチャンクが、開始プレゼンテーション時間を有し、その開始プレゼンテーション時間が、第1のメディアデータチャンクの開始プレゼンテーション時間よりも後である、選択することと、第2のメディアデータチャンクのダウンロードを要求することとを含んでもよい。第2のメディアデータチャンクは、受信機にとって利用可能な第1のチャンクの開始プレゼンテーション時間のものと比較した、その開始プレゼンテーション時間が、最も低い差となるように、および/またはそのプレイバックが、その開始プレゼンテーション時間と第1のメディアデータチャンクの開始プレゼンテーション時間との間の所定の最大ギャップをもつ最大プレイバックレートとなるように、受信機によって選ぶこともできる。

【0325】

[0360]いくつかの実施形態は、第1のメディアデータチャンクのうち残っている部分のダウンロードが、プレイバック用の時間内に完了できないかどうか決定することと、第2のメディアデータチャンクのダウンロードが、プレイバック用の時間内に完了され得るかどうか決定することと、要求を続けるか、それとも第1のメディアデータチャンクについての要求を取り消し、代わりに第2のメディアデータチャンクを要求するかの決定を、第1のメディアデータチャンクのうち残っている部分のダウンロードが、プレイバック用の時間内に完了できないかどうか、および第2のメディアデータチャンクのダウンロードが、プレイバック用の時間内に完了され得るかどうかに基づかせることとを含み得る。第2のデータチャンク中のメディアデータのプレイバックレートは、受信機においてサポートされる最も高いプレイバックレートとなるように選ばれることができる。受信機は、すでにプレゼンテーションバッファ中にある少なくともいくつかのメディアデータのプレゼンテーション時間をカバーするメディアデータを要求し、要求されたメディアデータをダウンロードし、要求されたメディアデータをブレイアウトし、すでにプレゼンテーションバッファ中にある対応するメディアデータのうち少なくともいくつかを破棄してもよい。要

10

20

30

40

50

求されたメディアデータのプレイバックレートは、プレゼンテーションバッファから破棄される対応するメディアデータの最大プレゼンテーション持続時間に対する制約を受ける、最大のプレイバックレートであり得る。要求されるメディアデータは、その開始プレゼンテーション時間が、受信機にとって利用可能な最も早い開始プレゼンテーション時間になるように選ばれ得る。

【0326】

[0361]いくつかの受信機において、ダウンロードは、バッファレベルに依存し、受信機は、高ウォーターマークおよび低ウォーターマークの概念を使用する。そのような受信機において、メディアデータは、ソースからダウンロードされ、受信機のプレゼンテーションバッファに記憶される。プレゼンテーションバッファの充満レベル（または単に「レベル」）が決定され、ここで充満レベルは、プレゼンテーションバッファのどのような部分が、プレゼンテーション要素によってまだ消費されていないメディアデータを含むかを表す。充満レベルが高充満しきい値（「高ウォーターマーク」）を上回る場合、ダウンロードは停止し、充満レベルが低充満しきい値（「低いウォーターマーク」）を下回る場合、ダウンロードはリスタートする。充満レベルは、プレゼンテーション要素によってメディアデータが消費されると、アップデートされ得る。充満レベルは、メモリ記憶容量の単位および/またはプレゼンテーション時間の単位で測ることができる。ダウンロードは、推定ラウンドトリップ時間（「ERTT」）に基づくことができ、ERTTは、メディアデータダウンロードがリスタートされるとリセットされる。ダウンロードが複数のTCP接続を介して起きた場合、メディアデータダウンロードがリスタートされると、使用されるいくつかのTCP接続はリセットされ得る。高充満しきい値および低充満しきい値は、時間とともに変化しうる。

【0327】

[0362]本開示を読んだ後には、当業者にはさらなる実施形態が想起され得る。他の実施形態では、上で開示された発明の組合せまたは副次的な組合せが、有利に行われ得る。コンポーネントの例示的な構成は例示を目的に示され、組合せ、追加、再構成などが、本発明の代替的な実施形態において考えられることを理解されたい。したがって、本発明は、例示的な実施形態に関して説明されてきたが、多数の修正が可能であることを当業者は認識するだろう。

【0328】

[0363]たとえば、本明細書で説明される処理は、ハードウェアコンポーネント、ソフトウェアコンポーネント、および/またはこれらの任意の組合せを使用して実施され得る。したがって、本明細書および図面は、限定的な意味ではなく例示的であると解釈されるべきである。しかしながら、特許請求の範囲において述べられるような本発明のより広い趣旨および範囲から逸脱することなく、様々な修正および変更を行うことができ、本発明は、以下の特許請求の範囲内にあるすべての修正と等価物とを包含することが意図されることは、明らかである。

以下に、出願当初の特許請求の範囲に記載された発明を付記する。

[C1]

受信機のプレゼンテーション要素を使用してプレイアウトするためのメディアを受信する受信機であって、前記プレイアウトの結果としてメディアがプレゼンテーションバッファからあるプレイバックレートで消費され、前記受信機が複数のプレイバックレートから選択するように構成された受信機において、プレイバックレートを選択するための方法であって、前記方法は、

前記プレゼンテーションバッファを監視することであって、前記プレゼンテーションバッファが、少なくとも、前記メディアデータが受信される時間と、前記受信機に関連付けられたプレゼンテーション要素によって前記メディアデータが消費される時間との間、メディアデータを記憶する、監視することと、

バッファレベルのインジケーションを記憶することであって、前記バッファレベルが、前記プレゼンテーションバッファのうちどの程度が、受信されたが前記プレゼンテーショ

10

20

30

40

50

ン要素によってまだ消費されていないメディアデータによって占有されるかに対応する、記憶することと、

推定ダウンロードレートを決定することと、

目標プレイバックレートを計算するために、前記記憶されたインジケーションおよび前記推定ダウンロードレートを使用することと、

前記目標プレイバックレートに従って、前記複数のプレイバックレートの中から選択することと、を備える方法。

[C 2]

前記選択されたプレイバックレートが、前記推定ダウンロードレートの所定の乗数以下のプレイバックレートであり、前記所定の乗数が、前記バッファレベルの増加関数である、上記 C 1 に記載の方法。

10

[C 3]

前記所定の乗数が、前記プレゼンテーションバッファ中の前記メディアデータのプレイバック持続時間のアフィン一次関数である、上記 C 2 に記載の方法。

[C 4]

前記プレゼンテーションバッファの前記バッファレベルがしきい量未満のとき、前記所定の乗数が 1 未満である、上記 C 2 に記載の方法。

[C 5]

前記プレゼンテーションバッファ中の前記メディアデータのプレゼンテーション持続時間が、あらかじめ設定された最大量のプレゼンテーション時間以上であるとき、前記所定の乗数が 1 以上である、上記 C 2 に記載の方法。

20

[C 6]

前記所定の乗数が、前記プレゼンテーションバッファ中の前記メディアデータのプレイバック持続時間の区分的一次関数である、上記 C 2 に記載の方法。

[C 7]

前記選択されたプレイバックレートが、前記推定ダウンロードレートの所定の乗数以下の前記プレイバックレートであり、前記所定の乗数が、前記プレゼンテーションバッファ中のメディアデータのバイト数の増加関数である、上記 C 1 に記載の方法。

[C 8]

前記プレイバックレートが、比例因子に前記推定ダウンロードレートを乗じたもの以下である前記複数のプレイバックレートのうち最も大きい利用可能プレイバックレートとなるように選択され、前記比例因子が、前記プレゼンテーションバッファ中の前記メディアデータのプレイバック持続時間を、レート変化に対する反応時間推定値で除算したものの増加関数である、上記 C 1 に記載の方法。

30

[C 9]

前記反応時間の前記推定値が、前記メディアデータ中の切替え点の間のプレゼンテーション時間に対する上限である、上記 C 8 に記載の方法。

[C 1 0]

前記反応時間の前記推定値が、前記メディアデータ中の切替え点の間のプレゼンテーション時間に対する平均である、上記 C 8 に記載の方法。

40

[C 1 1]

前記反応時間の前記推定値が、所定の定数に推定ラウンドトリップ時間（「ERTT」）を乗じたもの以上である、上記 C 8 に記載の方法。

[C 1 2]

受信機のプレゼンテーション要素を使用してプレイアウトするためのメディアを受信する受信機であって、前記プレイアウトの結果としてメディアがプレゼンテーションバッファからあるプレイバックレートで消費され、前記受信機が複数のプレイバックレートから選択するように構成された受信機において、プレイバックレートを選択するための方法であって、前記方法は、

前記プレゼンテーションバッファを監視することであって、前記プレゼンテーションバ

50

ッファが、少なくとも、前記メディアデータが受信される時間と、前記受信機に関連付けられたプレゼンテーション要素によって前記メディアデータが消費される時間との間、メディアデータを記憶する、監視することと、

バッファレベルのインジケーションを記憶することであって、前記バッファレベルが、前記プレゼンテーションバッファのうちどの程度が、受信されたが前記プレゼンテーション要素によってまだ消費されていないメディアデータによって占有されるかに対応する、記憶することと、

前記バッファレベルの許容分散を決定することと、

目標プレイバックレートを計算するために、バッファレベルの前記記憶されたインジケーションおよび前記バッファレベルの前記許容分散を使用することと、

前記目標プレイバックレートに従って、前記複数のプレイバックレートの中から選択することと、を備える方法。

[C 1 3]

前記プレイバックレートが、上位比例因子、下位比例因子、推定ダウンロードレート、現在のプレイバックレート、前記バッファレベル、およびレート変化に対する反応時間の推定値に基づいて選択される、上記 C 1 2 に記載の方法。

[C 1 4]

前記上位比例因子および前記下位比例因子が両方とも、前記プレゼンテーションバッファ中の前記メディアデータのプレイバック持続時間を、レート変化に対する前記反応時間の前記推定値で除算したものの増加関数および/または区分的一次関数である、上記 C 1 3 に記載の方法。

[C 1 5]

前記上位比例因子が前記下位比例因子以上である、上記 C 1 3 に記載の方法。

[C 1 6]

前のプレイバックレートが、前記下位比例因子に前記推定ダウンロードレートを乗じたものと、前記上位比例因子に前記ダウンロードレート推定値を乗じたものの間であるとき、前記プレイバックレートが、前記前のプレイバックレートと同じになるように選択される、上記 C 1 3 に記載の方法。

[C 1 7]

前のプレイバックレートが、前記上位比例因子に前記ダウンロードレート推定値を乗じたものを上回るとき、前記プレイバックレートが、前記上位比例因子に前記推定ダウンロードレートを乗じたもの以下である最大の利用可能プレイバックレートになるように選択される、上記 C 1 3 に記載の方法。

[C 1 8]

前のプレイバックレートが、前記下位比例因子に前記ダウンロードレート推定値を乗じたものを下回るとき、前記プレイバックレートが、前記下位比例因子に前記推定ダウンロードレートを乗じたもの以下である最大の利用可能プレイバックレートになるように選択される、上記 C 1 3 に記載の方法。

[C 1 9]

受信機のプレゼンテーション要素を使用してプレイアウトするためのメディアを受信し、あるプレイバックレートでメディアデータを消費する受信機であって、

複数のプレイバックレートのうち1つでのプレイバックを提供するプレゼンテーションインターフェースと、

前記プレゼンテーションインターフェースに結合された、少なくとも、前記メディアデータが受信される時間と、前記受信機に関連付けられたプレゼンテーション要素によって前記メディアデータが消費される時間との間、メディアデータを記憶するプレゼンテーションバッファと、

バッファレベルのインジケーションを含む、プレゼンテーションバッファ容量に関連した変数のためのストレージであって、前記バッファレベルが、前記プレゼンテーションバッファのうちどの程度が、受信されたが前記プレゼンテーション要素によってまだ消費さ

10

20

30

40

50

れていないメディアデータによって占有されるかに対応する、ストレージと、
 推定ダウンロードレートを決定する推定ダウンロードレート決定器と、
 目標プレイバックレートを計算するために前記記憶されたインジケーションおよび前記
 推定ダウンロードレートを使用する、決定された選択プレイバックレートに従って要求を
 アレンジするための論理と、を備える受信機。

[C 2 0]

前記選択されたプレイバックレートが、前記推定ダウンロードレートの所定の乗数以下
 の前記プレイバックレートであり、前記所定の乗数が、前記バッファレベルの増加関数で
 あり、前記所定の乗数が、前記プレゼンテーションバッファ中の前記メディアデータのプ
 レイバック持続時間のアフィン一次関数である、上記 C 1 9 に記載の受信機。

10

[C 2 1]

前記プレゼンテーションバッファの前記バッファレベルがしきい量未満のとき、前記所
 定の乗数が 1 未満である、上記 C 2 0 に記載の受信機。

[C 2 2]

前記選択されたプレイバックレートが、前記推定ダウンロードレートの所定の乗数以下
 の前記プレイバックレートであり、前記所定の乗数が、前記プレゼンテーションバッファ
 中のメディアデータのバイト数の増加関数である、上記 C 2 0 に記載の受信機。

[C 2 3]

前記プレイバックレートが、比例因子に前記ダウンロードレート推定値を乗じたもの以下
 である前記複数のプレイバックレートのうち最も大きい利用可能プレイバックレートと
 なるように選択され、前記比例因子が、前記プレゼンテーションバッファ中の前記メ
 ディアデータのプレイバック持続時間を、レート変化に対する反応時間推定値で除算したも
 の増加関数である、上記 C 1 9 に記載の受信機。

20

[C 2 4]

前記反応時間の前記推定値が、前記メディアデータ中の切替え点の間のプレゼンテーシ
 ョン時間に対する上限、または前記メディアデータ中の切替え点の間のプレゼンテーシ
 ョン時間に対する平均である、上記 C 2 3 に記載の受信機。

[C 2 5]

前記反応時間の前記推定値が、所定の定数に推定ラウンドトリップ時間 (「 E R T T 」
) を乗じたもの以上である、上記 C 2 3 に記載の受信機。

30

[C 2 6]

前記受信機のプレゼンテーション要素にメディアをプレイアウトするための受信機のプ
 ロセッサによる実行用の非一時的コンピュータ可読媒体であって、前記プレイアウトの結
 果、メディアが、あるプレイバックレートでプレゼンテーションバッファから消費され、
 前記受信機が、複数のプレイバックレートから選択するように構成され、前記メディアが

、
 前記プレゼンテーションバッファを監視するためのプログラムコードであって、前記プ
 レゼンテーションバッファが、少なくとも、メディアデータが受信される時間と、前記受
 信機に関連付けられたプレゼンテーション要素によって前記メディアデータが消費される
 時間との間、前記メディアデータを記憶する、プログラムコードと、

40

バッファレベルのインジケーションを記憶するためのプログラムコードであって、前記
 バッファレベルが、前記プレゼンテーションバッファのうちどの程度が、受信されたが前
 記プレゼンテーション要素によってまだ消費されていないメディアデータによって占有さ
 れるかに対応するプログラムコードと、

前記バッファレベルの許容分散を決定するためのプログラムコードと、

目標プレイバックレートを計算するために、バッファレベルの前記記憶されたインジ
 ケーションおよび前記バッファレベルの前記許容分散を使用するためのプログラムコードと

、
 前記目標プレイバックレートに従って前記複数のプレイバックレートの中から選択する
 ためのプログラムコードと、を備えるプログラムコードを含む、非一時的コンピュータ可

50

読媒体。

[C 2 7]

上位比例因子、下位比例因子、ダウンロードレート推定値、現在のプレイバックレート、前記バッファレベル、およびレート変化に対する反応時間の推定値に基づいてプレイバックレートを選択するためのプログラムコードをさらに備える、上記 C 2 6 に記載の非一時的コンピュータ可読媒体。

[C 2 8]

前記上位比例因子および前記下位比例因子が両方とも、前記プレゼンテーションバッファ中の前記メディアデータのプレイバック持続時間を、レート変化に対する前記反応時間の前記推定値で除算したものの増加関数および / または区分的一次関数である、上記 C 2 7 に記載の非一時的コンピュータ可読媒体。

10

[C 2 9]

前記上位比例因子が前記下位比例因子以上である、上記 C 2 7 に記載の非一時的コンピュータ可読媒体。

[C 3 0]

前記プレイバックレートを比較し、前のプレイバックレートが、前記下位比例因子に前記推定ダウンロードレートを乗じたものと、前記上位比例因子に前記ダウンロードレート推定値を乗じたものの間であるときの前記前のプレイバックレートと同じになるように選択するためのプログラムコードをさらに備える、上記 C 2 7 に記載の非一時的コンピュータ可読媒体。

20

[C 3 1]

前記プレイバックレートを比較し、前のプレイバックレートが、前記上位比例因子に前記ダウンロードレート推定値を乗じたものを上回るとき、前記上位比例因子に前記推定ダウンロードレートを乗じたもの以下である最大の利用可能プレイバックレートになるように選択するためのプログラムコードをさらに備える、上記 C 2 7 に記載の非一時的コンピュータ可読媒体。

[C 3 2]

前記プレイバックレートを比較し、前のプレイバックレートが、前記下位比例因子に前記ダウンロードレート推定値を乗じたものを下回るとき、前記下位比例因子に前記推定ダウンロードレートを乗じたもの以下である最大の利用可能プレイバックレートになるように選択するためのプログラムコードをさらに備える、上記 C 2 7 に記載の非一時的コンピュータ可読媒体。

30

【 図 1 】

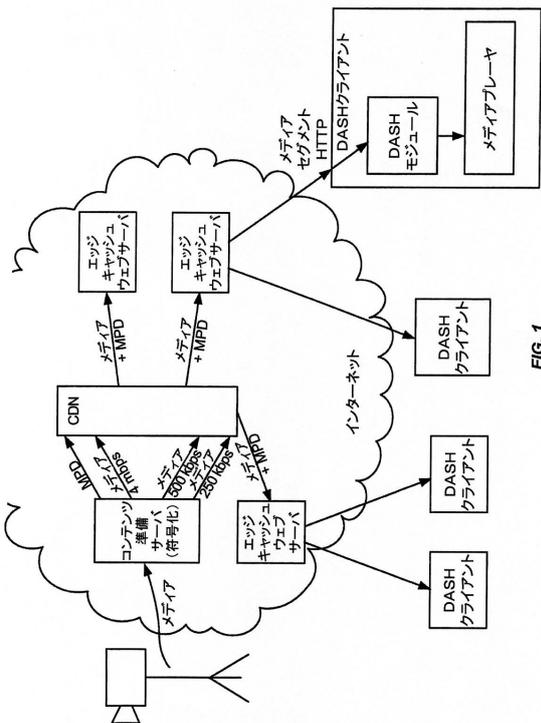


FIG. 1

【 図 2 】

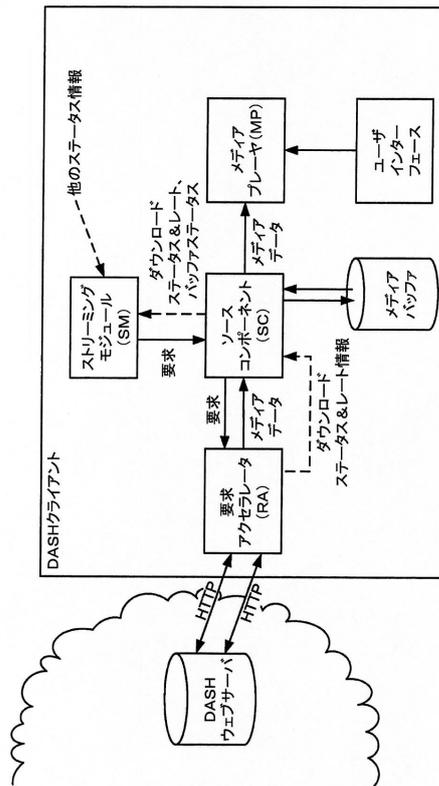


FIG. 2

【 図 3 】

図 3

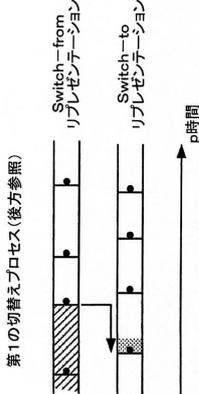
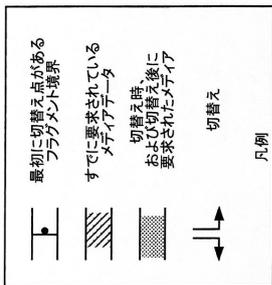


FIG. 3A

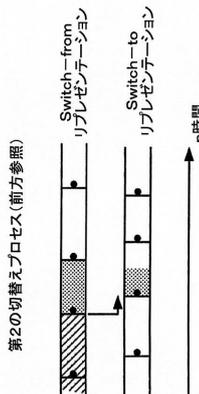


FIG. 3B

【 図 4 】

図 4



FIG. 4

【 図 5 】

図 5

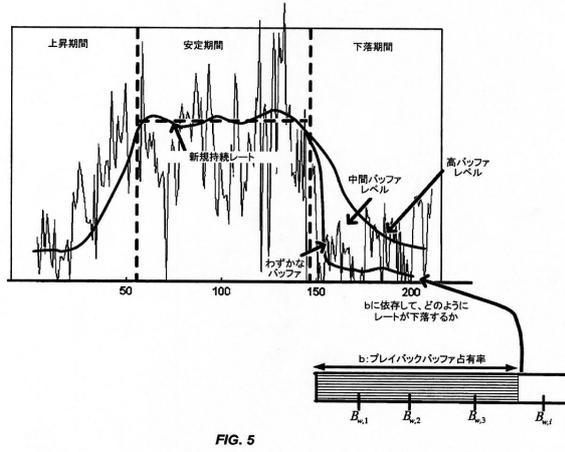


FIG. 5

【 図 6 】

図 6

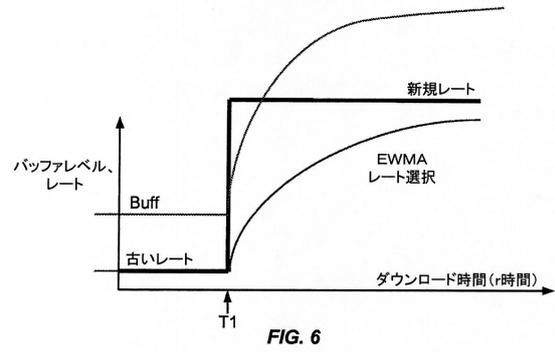


FIG. 6

【 図 7 】

図 7

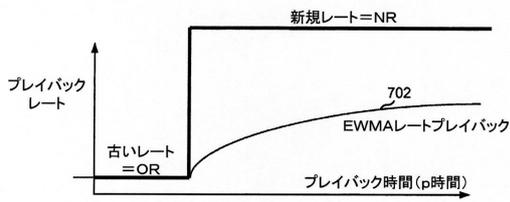
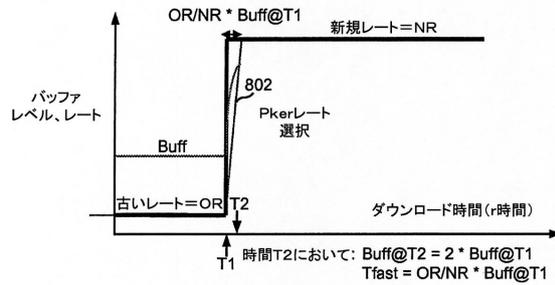


FIG. 7

【 図 8 】

図 8



新規レート=NR

FIG. 8

【 図 9 】

図 9

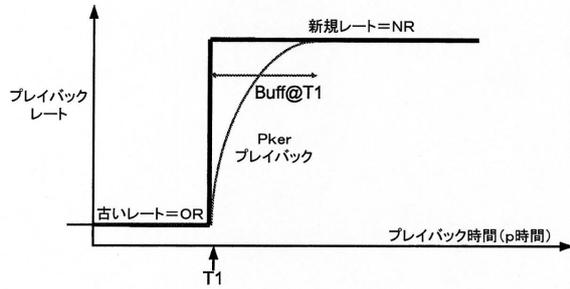


FIG. 9

【 図 10 】

図 10

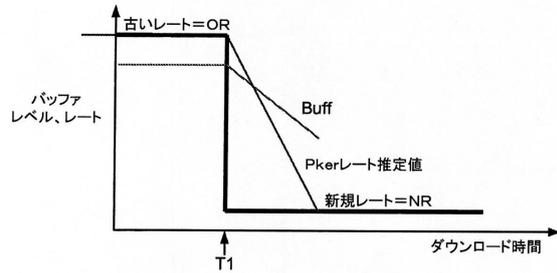


FIG. 10

【 図 11 】

図 11

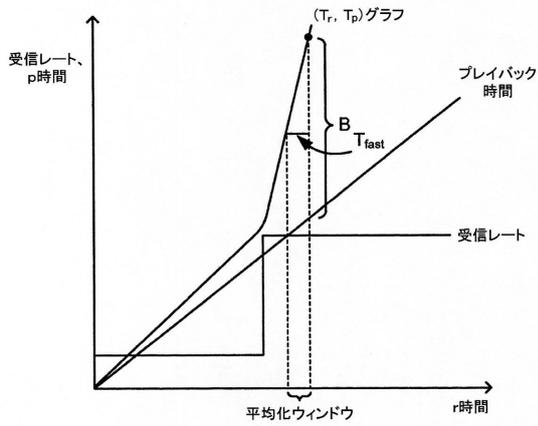


FIG. 11

【 図 12 】

図 12

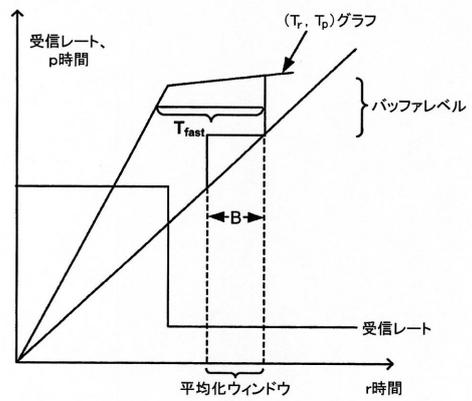


FIG. 12

【 図 1 3 】

図 13

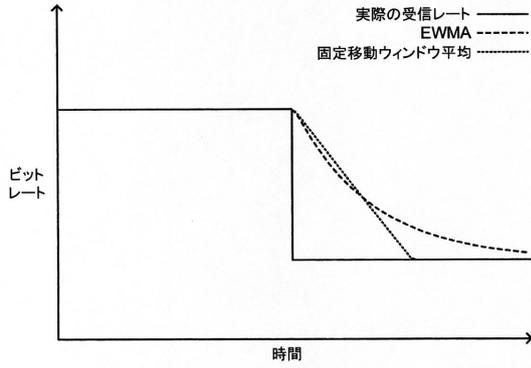


FIG. 13

【 図 1 4 】

図 14

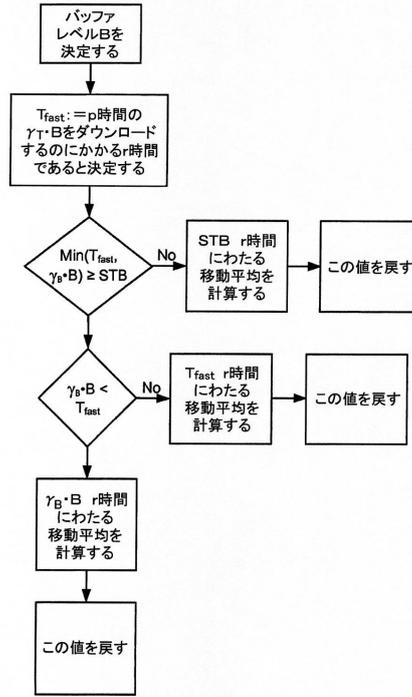


FIG. 14

【 図 1 5 】

図 15

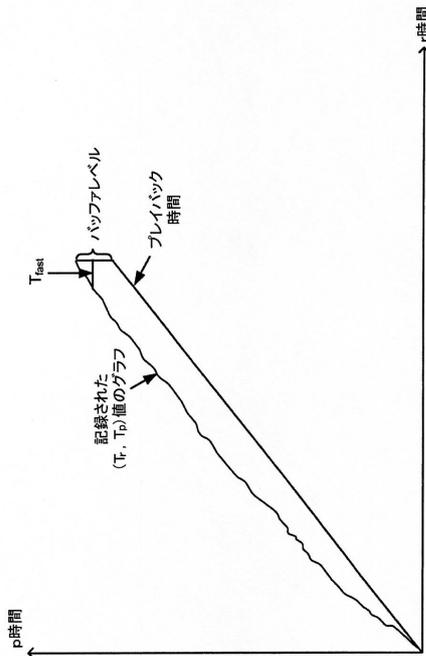


FIG. 15

【 図 1 6 】

図 16

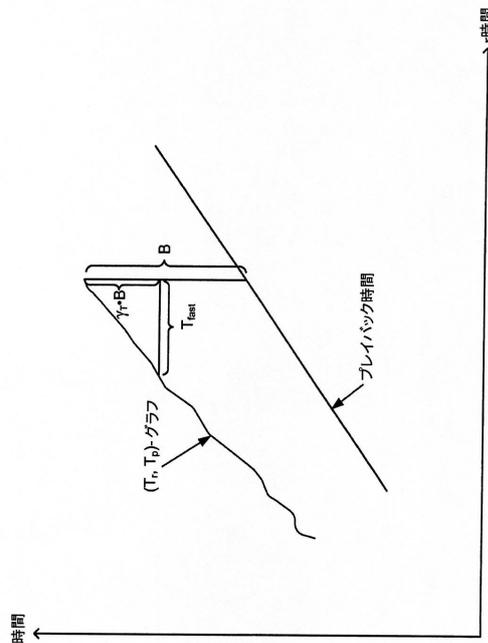


FIG. 16

【 図 17 】

図 17

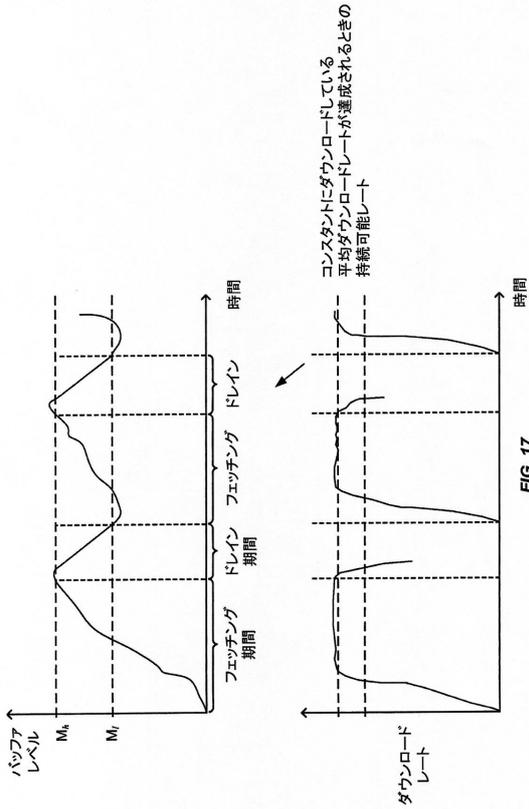


FIG. 17

【 図 18 】

図 18

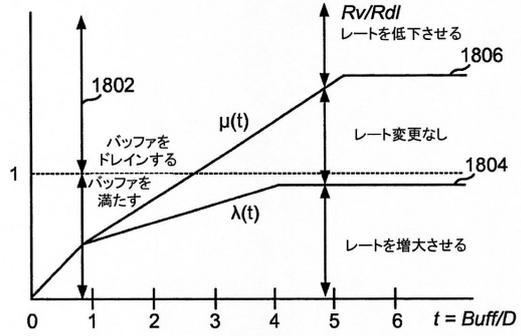


FIG. 18

【 図 19 】

図 19

コンサーバティブ設定

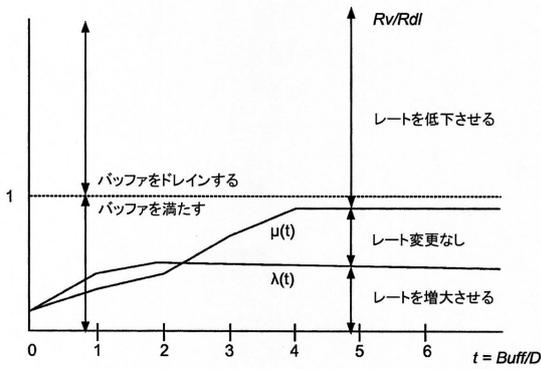


FIG. 19

【 図 20 】

図 20

中間設定

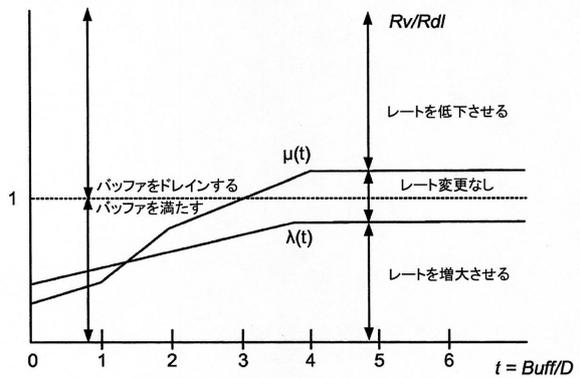


FIG. 20

【 図 2 1 】

図 21

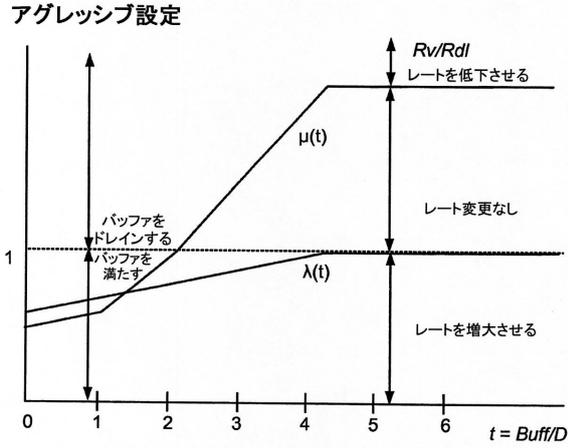


FIG. 21

【 図 2 2 】

図 22

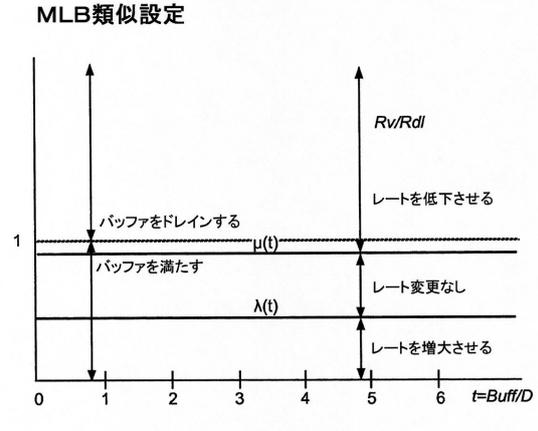


FIG. 22

【 図 2 3 】

図 23

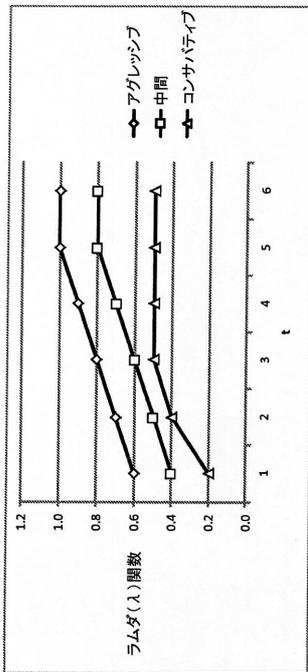


FIG. 23

【 図 2 4 】

図 24

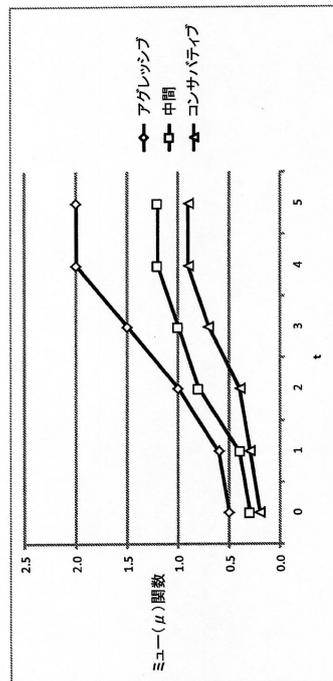


FIG. 24

【 図 25 】

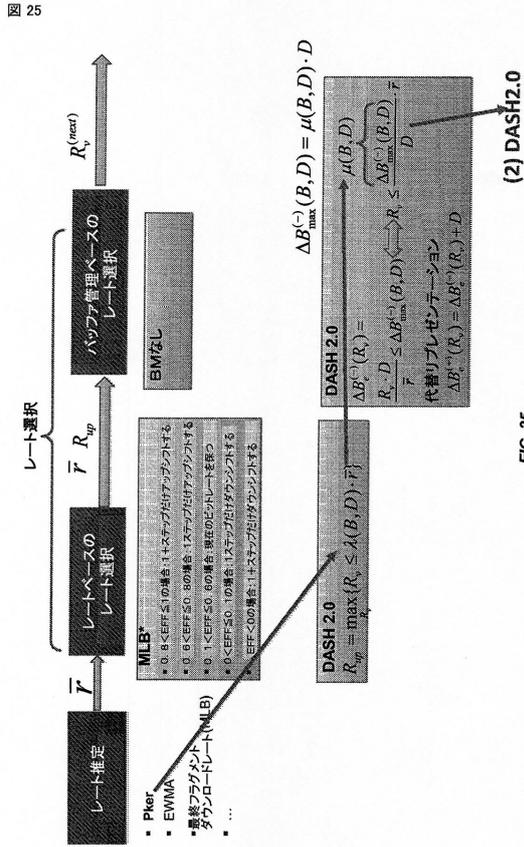


FIG. 25

【 図 26 】

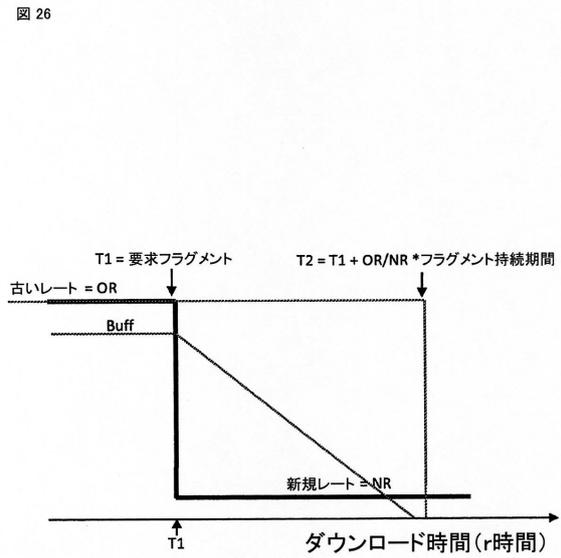


FIG. 26

【 図 27 】

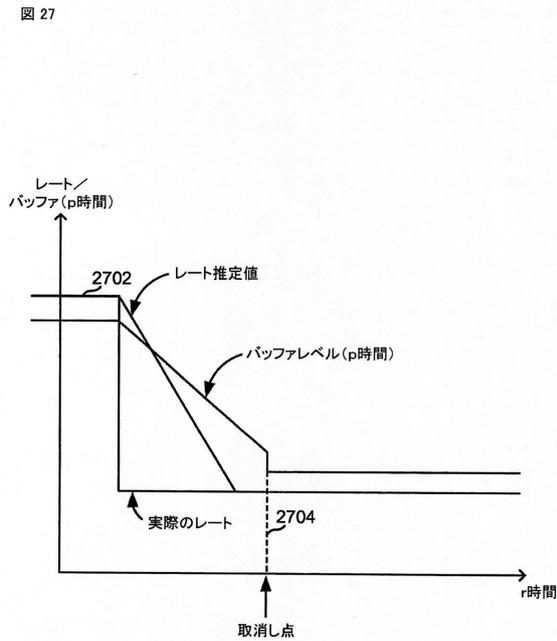


FIG. 27

【 図 28 】

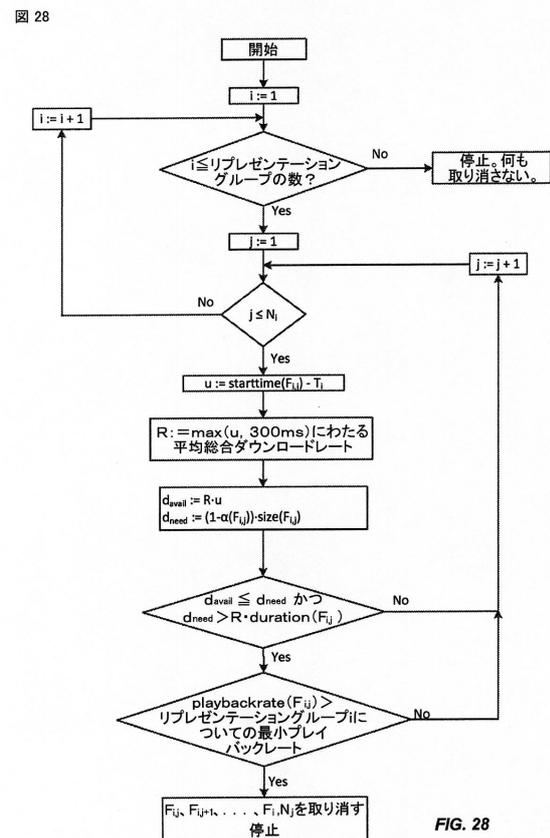


FIG. 28

【 図 29 】

図 29

```

for あらゆるリプレゼンテーショングループ i do
  m ← このグループについて最も小さい利用可能レート
  for j = 1, ..., Ni do
    if representation-rate(Fij) = m then
      continue
    end if
    u ← starttime(Fij) - T
    R ← max(w, 300ms)にわたる総合レート
    if Rを計算することができない then
      continue
    end if
    davail ← R · u
    dneed ← (1 - α(Fij)) · size(Fij)
    if davail < dneed and dneed > R · duration(Fij) then
      return (i, j)
    end if
  end for
end for
return nil

```

FIG. 29

【 図 30 】

図 30

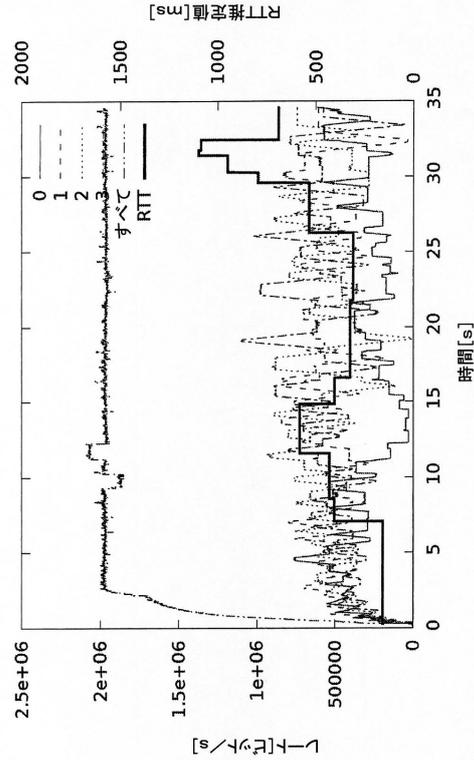


FIG. 30

【 図 31 】

図 31

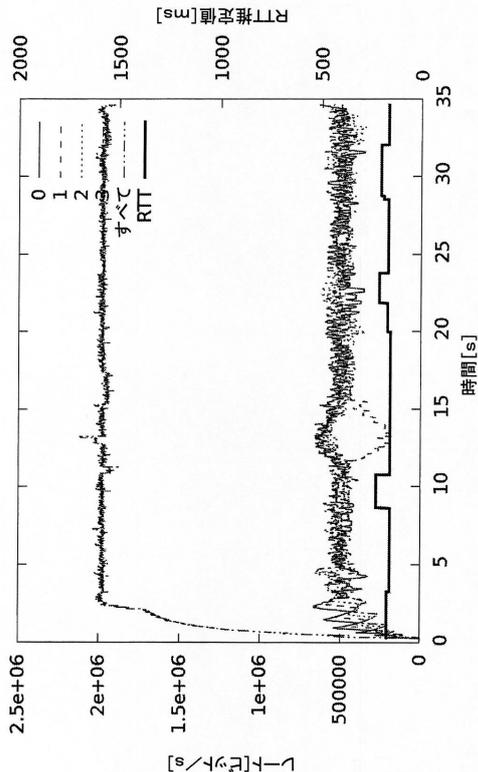


FIG. 31

【 図 32 】

図 32

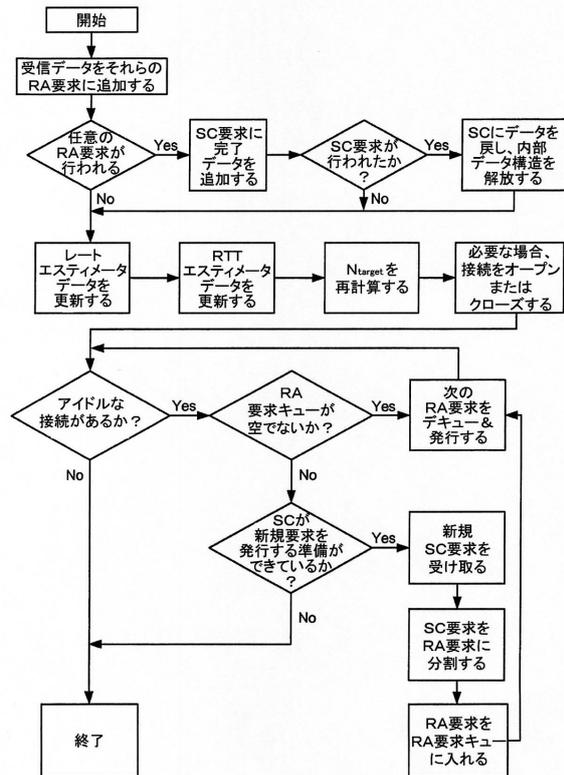


FIG. 32

【 図 3 3 】

図 33

```

S ← 元の要求のサイズ
Ntarget ← 接続の目標数
R ← 最後のTwaitにわたるレート推定値
RTT ← RTT推定値
if 有効なRTT推定値がないまたは有効なレート推定値が既知でない then
  n ← Ntarget
  return n
end if
D ← max(Dmin, cs · RTT)
if S < R · D then
  n ← ⌊  $\frac{N_{target} \cdot S}{R \cdot D}$  ⌋
else
  n ← Ntarget
end if
return n

```

FIG. 33

【 図 3 4 】

図 34

```

a := 1/(n + 1)
v[1], ..., v[n-1]を、範囲[0, 1]内で均一に選ばれるiidランダム変数とする
v[]を昇順でソートする
for i = 1, ..., n - 1:
  w[i] := (1 - a)*i/n + a*v[i]
w[n] := 1
prev := 0
for i = 1, ..., n
  x := floor(w[i] * S)
  s[i] := x - prev
  prev := x
ベクトルs[]をサイズ降順でソートする
return s[1], ..., s[n]

```

FIG. 34

【 図 3 5 】

図 35

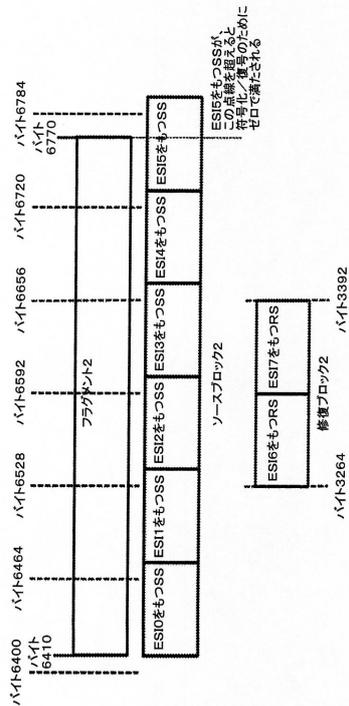


FIG. 35

【 図 3 6 】

図 36

MLBパラメータセット:

T	ラムダ	ミュー
0	0.4*R	0.9

コンサバティブパラメータセット:

t	ラムダ	ミュー
0	0.2	0.2
1	0.4	0.3
2	0.5	0.4
3	0.5	0.7
4	0.5	0.9

中間パラメータセット:

t	ラムダ	ミュー
0	0.4	0.3
1	0.5	0.4
2	0.6	0.8
3	0.7	1.0
4	0.8	1.2

アグレッシブパラメータセット:

t	ラムダ	ミュー
0	0.6	0.5
1	0.7	0.6
2	0.8	1.0
3	0.9	1.5
4	1.0	2.0

FIG. 36

フロントページの続き

- (74)代理人 100153051
弁理士 河野 直樹
- (74)代理人 100140176
弁理士 砂川 克
- (74)代理人 100158805
弁理士 井関 守三
- (74)代理人 100179062
弁理士 井上 正
- (74)代理人 100124394
弁理士 佐藤 立志
- (74)代理人 100112807
弁理士 岡田 貴志
- (74)代理人 100111073
弁理士 堀内 美保子
- (72)発明者 ガオ、キアン
アメリカ合衆国、カリフォルニア州 9 2 1 2 1 - 1 7 1 4、サン・ディエゴ、モアハウス・ドライブ 5 7 7 5
- (72)発明者 ルビー、マイケル・ジョージ
アメリカ合衆国、カリフォルニア州 9 2 1 2 1 - 1 7 1 4、サン・ディエゴ、モアハウス・ドライブ 5 7 7 5
- (72)発明者 マオ、イーニアン
アメリカ合衆国、カリフォルニア州 9 2 1 2 1 - 1 7 1 4、サン・ディエゴ、モアハウス・ドライブ 5 7 7 5
- (72)発明者 ミンダー、ロレンツ・クリストフ
アメリカ合衆国、カリフォルニア州 9 2 1 2 1 - 1 7 1 4、サン・ディエゴ、モアハウス・ドライブ 5 7 7 5
- (72)発明者 フォール、ケビン・ローランド
アメリカ合衆国、カリフォルニア州 9 2 1 2 1 - 1 7 1 4、サン・ディエゴ、モアハウス・ドライブ 5 7 7 5

審査官 福西 章人

- (56)参考文献 国際公開第2010/111261(WO, A1)
特開2011-091709(JP, A)
特開平10-336626(JP, A)
特開2007-089137(JP, A)
国際公開第2011/038021(WO, A1)
特表2009-505502(JP, A)
国際公開第2011/018868(WO, A1)
特表2008-537393(JP, A)

(58)調査した分野(Int.Cl., DB名)

H04N 21/00 - 21/858
H04N 13/00