

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5920006号
(P5920006)

(45) 発行日 平成28年5月18日(2016.5.18)

(24) 登録日 平成28年4月22日(2016.4.22)

(51) Int.Cl.

F I

G 0 6 F 13/00 (2006.01)

G 0 6 F 13/00 5 5 0 A

G 0 6 F 13/00 5 4 0 A

請求項の数 7 (全 58 頁)

(21) 出願番号 特願2012-111014 (P2012-111014)
 (22) 出願日 平成24年5月14日(2012.5.14)
 (65) 公開番号 特開2013-238993 (P2013-238993A)
 (43) 公開日 平成25年11月28日(2013.11.28)
 審査請求日 平成27年3月19日(2015.3.19)

(73) 特許権者 000005223
 富士通株式会社
 神奈川県川崎市中原区上小田中4丁目1番
 1号
 (74) 代理人 100074099
 弁理士 大菅 義之
 (74) 代理人 100133570
 弁理士 ▲徳▼永 民雄
 (72) 発明者 堀尾 健一
 神奈川県川崎市中原区上小田中4丁目1番
 1号 富士通株式会社内
 (72) 発明者 松井 一樹
 神奈川県川崎市中原区上小田中4丁目1番
 1号 富士通株式会社内

最終頁に続く

(54) 【発明の名称】 画面更新制御プログラム、画面更新制御方法、および情報処理装置

(57) 【特許請求の範囲】

【請求項1】

第1のコンピュータに、
 ネットワークを介して前記第1のコンピュータと接続された第2のコンピュータであっ
 て、

前記第1のコンピュータの入力装置を介して前記第1のコンピュータに与えられた入
 力についての通知を前記第1のコンピュータから受信し、

前記入力に応じて1つ以上のプログラムを実行し、

前記1つ以上のプログラムの実行に応じて、前記第1のコンピュータの画面に表示す
 るための画像を生成し、

前記第1のコンピュータに前記画像を前記画面上に表示させるための画面情報を送信
 するとともに、前記画面情報を送信する第1の時刻を示す時刻情報を送信し、

前記第1のコンピュータから通知される値に基づいて、前記画面情報を送信する頻度
 を調整しながら、前記画像の生成ならびに前記画面情報および前記時刻情報の送信を繰り
 返す

ことを特徴とする第2のコンピュータから、前記画面情報と前記時刻情報を受信し、

前記第1のコンピュータと前記第2のコンピュータとの間の前記ネットワークの可用帯
 域幅の推定値を、

前記第1のコンピュータが前記画面情報を受信した第2の時刻と、

前記推定値と、受信した前記画面情報のサイズと、受信した前記時刻情報により示さ

10

20

れる前記第1の時刻から、前記第1のコンピュータが前記画面情報を受信するものと推定される第3の時刻と

の差に応じて、増加させるか、維持するか、または、ある期間内に1回以上受信した前記画面情報それぞれのサイズと前記期間の長さから算出される第1の受信ビットレートに基づいて減少させ、

少なくとも前記推定値を増加または減少させることで更新した場合には、前記推定値または前記推定値の更新を反映する指標値を前記第2のコンピュータに通知する

ことを含む画面更新制御処理を実行させる画面更新制御プログラム。

【請求項2】

前記推定値を前記差に応じて増加させるか、維持するか、または、減少させる処理は、
前記差が正の第1の閾値より大きいとき、前記第1の受信ビットレートに基づいて前記推定値を減少させ、

前記差が前記第1の閾値未満の第2の閾値以下のとき、前記推定値を増加させ、

前記差が前記第2の閾値より大きく前記第1の閾値以下のとき、前記推定値を維持する

ことを含むことを特徴とする請求項1に記載の画面更新制御プログラム。

【請求項3】

直近に受信された前記画面情報に関しての、

前記第1の時刻と

前記第2の時刻と

前記サイズ

から算出される第2の受信ビットレートが、前記推定値に基づいて前記推定値よりも大きく定められる判定基準値よりも大きい場合、前記第2の受信ビットレートに応じて前記推定値を増加させる

ことを前記画面更新制御処理が含むことを特徴とする請求項1または2に記載の画面更新制御プログラム。

【請求項4】

前記差が所定の閾値以下の場合に、

前記推定値を維持するとともに、

前記推定値よりも大きな値を前記第2のコンピュータに通知する

ことを前記画面更新制御処理が含むことを特徴とする請求項1または3に記載の画面更新制御プログラム。

【請求項5】

第1のコンピュータが、前記第1のコンピュータの入力装置を介して前記第1のコンピュータに与えられた入力を、ネットワークを介して前記第1のコンピュータと接続された第2のコンピュータに通知し、

前記第2のコンピュータが、前記通知された入力に応じて1つ以上のプログラムを実行し、

前記第2のコンピュータが、前記1つ以上のプログラムの実行に応じて、前記第1のコンピュータの画面に表示するための画像を生成し、

前記第2のコンピュータが、前記第1のコンピュータに前記画像を前記画面上に表示させるための画面情報を送信するとともに、前記画面情報を送信する第1の時刻を示す時刻情報を送信し、

前記第1のコンピュータが、前記画面情報と前記時刻情報を受信し、

前記第1のコンピュータが、前記第1のコンピュータと前記第2のコンピュータとの間の前記ネットワークの可用帯域幅の推定値を、

前記第1のコンピュータが前記画面情報を受信した第2の時刻と、

前記推定値と、受信した前記画面情報のサイズと、受信した前記時刻情報により示される前記第1の時刻から、前記第1のコンピュータが前記画面情報を受信するものと推定される第3の時刻

10

20

30

40

50

の差に応じて、増加させるか、維持するか、または、ある期間内に1回以上受信した前記画面情報それぞれのサイズと前記期間の長さから算出される受信ビットレートに基づいて減少させ、

前記第1のコンピュータが、少なくとも前記推定値を増加または減少させることで更新した場合には、前記推定値または前記推定値の更新を反映する指標値を前記第2のコンピュータに通知し、

前記第2のコンピュータが、通知された前記推定値または通知された前記指標値に基づいて、前記画面情報を送信する頻度を調整しながら、前記画像の生成ならびに前記画面情報および前記時刻情報の送信を繰り返す

ことを含む画面更新制御方法。

10

【請求項6】

前記画面更新制御方法は、前記第1のコンピュータが前記差を前記第2のコンピュータに通知することを含み、

前記第2のコンピュータは、通知された前記差にも基づいて前記頻度を調整する

ことを特徴とする請求項5に記載の画面更新制御方法。

【請求項7】

ネットワークを介して接続された第1の情報処理装置と第2の情報処理装置を含むシステムで使われる前記第1の情報処理装置であって、

前記第1の情報処理装置と前記第2の情報処理装置との間の前記ネットワークの可用帯域幅の推定値または前記推定値の更新を反映する指標値を示す通知を、前記第2の情報処理装置に送信し、前記第1の情報処理装置に入力が与えられると、前記入力についての通知を前記第2の情報処理装置に送信する送信部と、

20

前記第2の情報処理装置が、前記入力についての前記通知に基づいて1つ以上のプログラムを実行し、前記1つ以上のプログラムの実行に応じて、前記第1の情報処理装置の画面に表示するための画像を生成し、前記第1の情報処理装置に前記画像を前記画面上に表示させるための画面情報を送信するとともに、前記画面情報を送信する第1の時刻を示す時刻情報を送信し、前記推定値または前記指標値に基づいて前記画面情報を送信する頻度を調整しながら前記画像の生成ならびに前記画面情報および前記時刻情報の送信を繰り返すと、前記第2の情報処理装置から、順次送信される前記画面情報と前記時刻情報をそれぞれ受信する受信部と、

30

前記推定値を、

前記画面情報が前記受信部により受信された第2の時刻と、

前記推定値と、前記受信部が受信した前記画面情報のサイズと、前記受信部が受信した前記時刻情報により示される前記第1の時刻から、前記受信部が前記画面情報を受信するものと推定される第3の時刻と

の差に応じて、増加させるか、維持するか、または、ある期間内に1回以上前記受信部により受信された前記画面情報それぞれのサイズと前記期間の長さから算出される受信ビットレートに基づいて減少させる推定部と、

を備えることを特徴とする第1の情報処理装置。

【発明の詳細な説明】

40

【技術分野】

【0001】

本発明は画面の表示を制御する技術に関する。

【背景技術】

【0002】

ある種のシンクライアントシステムでは、コンピュータの画面の更新を遠隔制御する技術が使われる。例えば、画面の遠隔制御に使われるプロトコルとして、RDP (Remote Desktop Protocol) やRFB (Remote Framebuffer) プロトコルなどのプロトコルが知られている。

【0003】

50

例えば、ある種のシンククライアントシステムでは、サーバは、シンククライアントに画面を描画（render）させるためのデータを生成し、生成したデータをシンククライアントに送信する。サーバは、以上のようなデータの生成と送信を繰り返すことにより、シンククライアントの画面の更新を遠隔制御する。

【0004】

近頃では、コンピュータからの情報の漏洩を防止するという観点や、コンピュータの保守にかかるコストを削減するという観点など、いくつかの観点から、シンククライアントシステムが注目されている。シンククライアントシステムの利用は広まりつつあり、シンククライアントシステムに関する研究も様々に行われている。

【0005】

例えば、シンククライアントの汎用性を維持しつつ、操作レスポンスを向上させることを課題として、以下のような技術が提案されている。

当該技術による情報処理装置は、画像メモリと、変更頻度判別部と、第1の画像送信部と、高頻度変更領域識別部と、送信停止部と、第2の画像送信部とを有する。画像メモリは、コンピュータの実行結果を描画する画像を保持する。変更頻度判別部は、画像メモリに描画された画像を複数の領域に分割し、領域ごとにフレーム間の変更の頻度を判別する。

【0006】

第1の画像送信部は、変更のある領域の画像を送信する。高頻度変更領域識別部は、変更の頻度が閾値を超えた領域を高頻度更新領域として識別する。送信停止部は、識別した領域について、第1の画像送信部での送信を停止する。第2の画像送信部は、識別した領域の画像を、第1の画像送信部よりも圧縮率の高い動画の圧縮処理を行って送信する。

【0007】

ところで、画面の更新の遠隔制御はネットワーク通信をとこなう。例えば、ある種のシンククライアントシステムでは、シンククライアントの画面の更新を遠隔制御するためのデータを、サーバがネットワークを介してシンククライアントに送る。サーバがデータを送信してからシンククライアントがデータを受信するまでの時間の長さは、ネットワーク状況に依存する。したがって、ネットワーク状況によっては、かなり長い表示遅延が生じ得る。

【0008】

一方で、ネットワーク状況を把握するための技術も、様々に研究されている。例えば、ネットワーク状況は、具体的にはネットワークの可用帯域幅により表されてもよいし、ネットワーク上の経路のコストにより表されてもよい。可用帯域幅や経路のコストに関して、例えば以下のような技術が提案されている。

【0009】

ある方法は、ネットワークの可用帯域幅を高速かつ高精度で測定するための方法である。当該方法によれば、複数の周期的なタイムスタンプパケットが、受信側の通信装置へ、現在設定されている試験伝送率で伝送される。そして、受信側の通信装置にタイムスタンプパケットのそれぞれが受信された時刻に基づいて、タイムスタンプパケット間の伝送遅延時間差の変化傾向が検査される。

【0010】

伝送遅延時間差が、予め定められた安定範囲内に含まれず、かつ増加傾向を示すと、試験伝送率の値を減少させた後に、検査が繰り返される。一方、伝送遅延時間差が安定範囲内に含まれず、かつ減少傾向を示すと、試験伝送率の値を増加させた後に、検査が繰り返される。以上のような処理の結果、伝送遅延時間差が安定範囲内に含まれると、現在設定されている試験伝送率が可用帯域幅として決定される。

【0011】

また、経路のコストに関連して、次のようなシステムも提案されている。当該システムは、同一ノードに至る複数の経路が存在する場合に、最適な経路を選択するためのシステムである。具体的には、当該システムは、始点ノードから第1ノードを経由して終点ノードへ至る第1経路と、始点ノードから第2ノードを経由して終点ノードへ至る第2経路と

10

20

30

40

50

を含むマルチノードネットワークにおいて最適な経路を選択するシステムである。当該システムは、記憶手段と経路計測最適化手段を有する。

【0012】

記憶手段は、リンクコストを記憶する。また、経路計測最適化手段は、第1の経路コストと第2の経路コストとを比較し、コストが低い方の経路を選択することによって経路を最適化する。なお、第1の経路コストは、具体的には、始点ノードコスト、第1リンクコスト、第1ノードコスト、第2リンクコスト、および終点ノードコストに基づいて算出される。また、第2の経路コストは、始点ノードコスト、第3リンクコスト、第2ノードコスト、第4リンクコスト、および終点ノードコストに基づいて算出される。

【先行技術文献】

10

【特許文献】

【0013】

【特許文献1】特開2011-238014号公報

【特許文献2】特開2006-74773号公報

【特許文献3】特開2006-237837号公報

【発明の概要】

【発明が解決しようとする課題】

【0014】

ある種のシステムでは、第1のコンピュータの画面に表示される画像が、第2のコンピュータにより生成される。この場合、第1のコンピュータに画像を画面上に表示させるための画面情報を、第2のコンピュータが第1のコンピュータに送信する。そして、第1のコンピュータは、画面情報を受信し、画面情報にしたがって画像を画面上に表示する。以上のような画像の生成、画面情報の送信、および画像の表示は、繰り返し行われ得る。

20

【0015】

具体的には、第2のコンピュータによる画像の生成は、第1のコンピュータの入力装置を介して第1のコンピュータに与えられた入力に基づく。第1のコンピュータは、入力についての通知を第2のコンピュータに送信し、第2のコンピュータは、通知された入力に応じて、1つ以上のプログラムを実行する。第2のコンピュータは、1つ以上のプログラムの実行に応じて、上記のとおり画像を生成する。

【0016】

30

以上のように第1と第2のコンピュータが動作すると、第1のコンピュータのユーザにとっては、次のように認識される。すなわち、ユーザにとっては、ユーザが入力装置を介して第1のコンピュータに与えた入力に応じて、第1のコンピュータの画面が更新されるように認識される。

【0017】

この場合、ユーザが入力装置を介して第1のコンピュータに入力を与えてから、第1のコンピュータが画面情報に応じて画像を画面上に表示するまでにかかる時間の長さが、ユーザビリティに影響する。当該時間の長さは、第1と第2のコンピュータの間のネットワークの状況に依存する。そこで、ネットワーク状況が悪くてもユーザビリティの低下を抑えるために、ネットワーク状況に応じた何らかの制御が行われることが好ましい。

40

【0018】

一方で、仮に、ネットワーク状況に応じた制御のために、ネットワーク状況を把握するための付加的な通信が過剰に行われるとすると、ネットワーク状況が悪化するおそれがある。なぜなら、付加的な通信により、ネットワークの負荷が一層増してしまうからである。そして、ネットワーク状況の悪化は、ユーザビリティの低下を招く。よって、上記のような第1と第2のコンピュータを含むシステムにおけるユーザビリティの観点からは、付加的な通信の量は少ないことが好ましい。

【0019】

本発明は、1つの側面では、ユーザビリティがネットワーク状況に依存する環境においても、ユーザビリティの低下を抑えることを目的とする。

50

【課題を解決するための手段】

【0020】

一態様による画面更新制御プログラムは、第1のコンピュータに、以下に述べる画面更新制御処理を実行させる。

前記画面更新制御処理は、ネットワークを介して前記第1のコンピュータと接続された第2のコンピュータから、画面情報と時刻情報を受信することを含む。

【0021】

前記第2のコンピュータは、以下のようなコンピュータである。すなわち、前記第2のコンピュータは、前記第1のコンピュータの入力装置を介して前記第1のコンピュータに与えられた入力についての通知を前記第1のコンピュータから受信する。また、前記第2のコンピュータは、前記入力に応じて1つ以上のプログラムを実行する。そして、前記第2のコンピュータは、前記1つ以上のプログラムの実行に応じて、前記第1のコンピュータの画面に表示するための画像を生成する。

【0022】

前記画面情報は、前記第1のコンピュータに前記画像を前記画面上に表示させるための情報であり、前記時刻情報は、前記画面情報を送信する第1の時刻を示す情報である。前記第2のコンピュータは、前記画面情報を送信するとともに、前記時刻情報を送信する。また、前記第2のコンピュータは、前記第1のコンピュータから通知される値に基づいて、前記画面情報を送信する頻度を調整しながら、前記画像の生成ならびに前記画面情報および前記時刻情報の送信を繰り返す。

【0023】

そして、前記画面更新制御処理は、前記第1のコンピュータと前記第2のコンピュータとの間の前記ネットワークの可用帯域幅の推定値を、第2の時刻と第3の時刻との差に応じて、増加させるか、維持するか、または、減少させることを含む。ここで、前記第2の時刻は、前記第1のコンピュータが前記画面情報を受信した時刻である。また、前記第3の時刻は、前記推定値と、受信した前記画面情報のサイズと、受信した前記時刻情報により示される前記第1の時刻から、前記第1のコンピュータが前記画面情報を受信するものと推定される時刻である。なお、前記差に応じて前記推定値を減少させる処理は、ある期間内に1回以上受信した前記画面情報それぞれのサイズと前記期間の長さから算出される受信ビットレートに基づく。

【0024】

前記画面更新制御処理は、少なくとも前記推定値を増加または減少させることで更新した場合には、前記推定値または前記推定値の更新を反映する指標値を前記第2のコンピュータに通知することを含む。

【発明の効果】

【0025】

上記の画面更新制御プログラムによれば、ユーザビリティがネットワーク状況に依存する環境においても、ユーザビリティの低下を抑えることができる。

【図面の簡単な説明】

【0026】

【図1】第1実施形態の動作例を示すシーケンス図である。

【図2】画面の更新について説明する図である。

【図3】第1実施形態の効果を説明する図（その1）である。

【図4】第1実施形態の効果を説明する図（その2）である。

【図5】ハードウェア構成例を示す図である。

【図6】システムのブロック構成図である。

【図7】各種データの例を示す図である。

【図8】第2実施形態のクライアントの動作フローチャート（その1）である。

【図9】第2実施形態のクライアントの動作フローチャート（その2）である。

【図10】サーバの動作フローチャート（その1）である。

【図 1 1】サーバの動作フローチャート（その 2）である。

【図 1 2】過剰遅延時間と受信ビットレートの変化の例を示すグラフである。

【図 1 3】第 3 実施形態のクライアントの動作フローチャートである。

【図 1 4】第 4 実施形態のクライアントの動作フローチャートである。

【発明を実施するための形態】

【0027】

以下、実施形態について、図面を参照しながら詳細に説明する。具体的には、図 1～4 を参照して第 1 実施形態について説明する。次に、図 5 を参照して、第 1～第 4 実施形態のいずれにも適用可能なハードウェア構成について説明する。その後、図 6～12 を参照して第 2 実施形態について説明し、図 13 を参照して第 3 実施形態について説明し、図 14 を参照して第 4 実施形態について説明する。また、いくつかの変形例についても説明する。

10

【0028】

図 1 は、第 1 実施形態の動作例を示すシーケンス図である。図 1 には、コンピュータ 101 と、コンピュータ 102 と、ユーザ 103 の動作が例示されている。図 1 にはステップ S101～S124 が例示されているが、これらのステップについて順に説明する前に、まず、コンピュータ 101 と 102 の動作の概要を説明する。

【0029】

コンピュータ 101 と 102 はネットワークを介して接続されている。例えば、コンピュータ 101 は、シンクライアントシステムにおけるクライアントであってもよく、コンピュータ 102 はサーバであってもよい。より具体的には、コンピュータ 102 は、シンクライアントシステムにおいて、コンピュータ 101 の画面の更新を遠隔制御するサーバ（換言すれば、リモートデスクトップサービスを提供するサーバ）であってもよい。

20

【0030】

例えば、図 1 のステップ S104、S109、および S120 に例示するように、コンピュータ 101 には、コンピュータ 101 の入力装置を介して、ユーザ 103 から入力を与えられる。すると、例えばステップ S105、S110、および S121 に例示するように、コンピュータ 101 は、与えられた入力についての通知をコンピュータ 102 に送信し、コンピュータ 102 は当該通知を受信する。

【0031】

コンピュータ 102 は、通知された入力に応じて 1 つ以上のプログラムを実行する。そして、コンピュータ 102 は、それらの 1 つ以上のプログラムの実行に応じて、コンピュータ 101 の画面に表示するための画像を生成する。

30

【0032】

例えば、あるアプリケーションを起動するための入力操作をユーザ 103 が行くと、当該入力操作をコンピュータ 101 がコンピュータ 102 に通知する。よって、コンピュータ 102 は当該アプリケーションの実行を開始する。アプリケーションの実行の開始にともない、例えば、コンピュータ 102 は、あるウィンドウを画面に表示するための処理（つまり、当該ウィンドウを含む画面全体の画像を生成する処理）を行うこともある。

【0033】

また、起動済みのアプリケーションのウィンドウ上でユーザ 103 が入力操作を行うと、当該入力操作をコンピュータ 101 がコンピュータ 102 に通知する。そして、コンピュータ 102 は、当該アプリケーションの実行の過程において、当該入力操作に応じた処理を実行する。

40

【0034】

例えば、テキストエディタがフォアグラウンドで実行されている状況において、ユーザがキー入力操作を行う場合がある。この場合、コンピュータ 102 は、コンピュータ 101 からの通知に応じて、テキストエディタのウィンドウ内のカーソル位置に文字を表示させるための処理（つまり、カーソル位置に新たに文字が追加されているような画像を生成する処理）を実行する。

50

【0035】

起動済みのアプリケーションの種類によっては、コンピュータ101からの入力のお知らせがない間は、コンピュータ102は、当該アプリケーションの実行をサスペンドしてもよい。もちろん、コンピュータ102は、コンピュータ101から入力のお知らせがなくても、プログラムの実行を続行する場合もある。

【0036】

例えば、動画像を視聴するためのメディアプレイヤーが起動されている場合、ユーザ103が再生ボタンをクリックすると、クリック操作がコンピュータ101からコンピュータ102に通知される。すると、コンピュータ102は、動画像を再生するために、メディアプレイヤーのウィンドウ内を更新するための処理を開始する。ユーザ103が再生ボタンをクリックした後、仮に何も操作を行わないとしても、コンピュータ102は、メディアプレイヤーのウィンドウ内を更新し続ける。また、例えば、画面上に時刻を表示する機能をOS (Operating System) が提供する場合、コンピュータ102は、時間の経過に応じて画面上の時刻表示を更新する処理を、入力のお知らせとは関係なく行ってもよい。

10

【0037】

コンピュータ102は、例えば上記に例示したようにして1つ以上のプログラムを実行し、それらの1つ以上のプログラムの実行に応じて、コンピュータ101の画面に表示するための画像を繰り返し生成する。例えば、コンピュータ101の画面の解像度が1280×1024画素の場合、コンピュータ102は、1280×1024画素の画像を繰り返し生成する。

20

【0038】

そして、コンピュータ102は、生成した画像をコンピュータ101の画面上に表示させるための情報（以下、便宜上「画面情報」という）をコンピュータ101に送信する。コンピュータ102は、さらに、コンピュータ102が画面情報をコンピュータ101に送信する第1の時刻を示す時刻情報も、コンピュータ101に送信する。

【0039】

コンピュータ102は、画面情報と時刻情報を1つのパケットに含めてもよい。図1の例では、ステップS101、S106、S111、S114、S117、およびS122において、コンピュータ102がコンピュータ101に画面情報と時刻情報を送信する。

【0040】

また、コンピュータ101は、画面情報を受信すると、画面情報にしたがって画面を更新する。例えば、図1の例では、コンピュータ101が、ステップS102、S107、S112、S115、S118、およびS123において、受信した画面情報にしたがって画面を更新する。

30

【0041】

なお、画面情報の具体的な形式は、実施形態に応じて様々であってよい。以下に、画面情報の形式の例を3つ挙げる。

【0042】

第1の形式は、画面の少なくとも一部を更新するためのコマンドを1つ以上含む形式である。例えば、第1の形式の画面情報には、以下の(1a)～(1c)のようなコマンドが1つ以上含まれていてもよい。

40

【0043】

(1a) ある点からある点へ線を引くよう命じるコマンドなど、ベクタオブジェクトの描画のためのコマンド。

(1b) ある特定の座標に、ある特定の文字を、ある特定のフォントで、表示するよう命じるコマンド。

(1c) 画面内のある特定の領域を、ある静止画像に置き換えるよう命じるコマンド。

【0044】

第1の形式の画面情報が使われる実施形態においては、コンピュータ101は、画面情

50

報を受信すると、画面情報に含まれるコマンドにしたがって画面を更新する。

【0045】

第2の形式は、コンピュータ101の画面全体を、動画像用の圧縮アルゴリズムにしたがって圧縮したデータを含む形式である。例えば、30fps (frames per second) というフレームレートでコンピュータ101の画面を更新するために、30fpsというフレームレートの動画像のデータが、画面情報として送信されてもよい。例えば、MP EG (Moving Picture Expert Group) 形式などのデータ形式が利用可能である。

【0046】

第2の形式の画面情報が使われる実施形態においては、コンピュータ101は、画面情報を受信すると、画面情報に含まれる圧縮された動画像データをデコードし、フレーム画像を画面に表示する。コンピュータ101は、次々にフレーム画像を画面に表示することにより、画面を更新する。

10

【0047】

第3の形式は、以下に詳しく説明するとおり、第1の形式と第2の形式のハイブリッド形式である。第3の形式には、画面の更新が高頻度で生じても遅延の累積を防げるという効果と、コンピュータ102における圧縮処理の負荷を減らす効果がある。よって、画面情報は、第3の形式で送信されることが好ましい。

【0048】

例えば、メディアプレイヤーが利用されて動画像が再生される場合など、画面の更新が高頻度で生じる場合がある。画面の更新が高頻度で生じる場合、第1の形式では、ある程度大きなサイズの画面情報が、高頻度でコンピュータ102からコンピュータ101へ送信されることになる。

20

【0049】

すると、コンピュータ101と102の間の可用帯域幅によっては、コンピュータ102からコンピュータ101への経路上（例えば、コンピュータ102内の送信バッファや、経路上のルータ内のバッファなど）に画面情報が滞留してしまうことがある。その結果、遅延時間が次第に長くなってゆき、ユーザビリティが低下してしまうおそれがある。

【0050】

他方、第2の形式では、動画像用の圧縮アルゴリズムが使われる。動画像の圧縮では、フレーム間予測や動き補償などが行われる。そのため、画面の更新が高頻度で生じる場合、第2の形式の方が第1の形式よりもデータ量が少なく済むことがほとんどである。よって、第2の形式が使われる場合、上記のような遅延の累積のおそれはあまりない。

30

【0051】

ところが、動画像の圧縮は比較的高負荷の処理である。よって、例えば、多数のクライアントの画面を1台のサーバが遠隔制御するシステムにおいて第2の形式が使われると、サーバの負荷が非常に高くなる場合があり得る。換言すれば、コンピュータ101と同様の多数のコンピュータの画面を1台のコンピュータ102が遠隔制御するシステムにおいて第2の形式が使われると、コンピュータ102の負荷が非常に高くなる場合があり得る。

【0052】

高負荷に対処するため、コンピュータ102には、動画像の圧縮用の専用ハードウェア回路（例えばASIC (Application-Specific Integrated Circuit) など）が設けられてもよい。しかし、専用ハードウェア回路をコンピュータ102に設けることは、コンピュータ102の製造コストの増加につながるもので、あまり好ましくない。

40

【0053】

そこで、遅延の累積を防ぎつつ、コンピュータ102の処理負荷の軽減を図るために、第1の形式と第2の形式を適宜に組み合わせたハイブリッド形式である第3の形式が使われてもよい。第3の形式が使われる場合、具体的には、第1の形式と第2の形式は次のように組み合わせられてもよい。

【0054】

50

画面内において高頻度で更新が生じる領域には、第2の形式が適用される。それにより、遅延の累積を防ぐことができる。

【0055】

他方、更新の頻度が低い領域には、第1の形式が適用される。つまり、第2の形式を適用する範囲が限定される。多くの場合、画面上の小さな範囲を占める動画像を圧縮する処理は、画面上の大きな範囲を占める動画像を圧縮する処理よりも負荷が低い。よって、第2の形式を適用する範囲を限定することで、コンピュータ102の負荷を軽減することができる。

【0056】

動画像の圧縮の負荷がそれほど高くない場合、コンピュータ102は、圧縮用の専用ハードウェア回路なしでも、汎用のプロセッサによって、十分、動画像の圧縮に対処することができる。したがって、第2の形式を適用する範囲を限定することは、コンピュータ102の製造コストの増加を回避することにもつながる。

【0057】

以上のように、第3の形式は、更新が発生する頻度に応じて第1の形式と第2の形式を組み合わせた形式である。ここで、第3の形式の画面情報のより具体的な例について説明するため、図2を参照する。図2は、画面の更新について説明する図である。

【0058】

図2に示すように、画面110は、横方向にH個に分割され、縦方向にV個に分割されてもよい。この場合、画面110は合計で $V \times H$ 個のブロックを含む。図2の例では $V = 6$ かつ $H = 8$ であるが、 $2 \leq V$ かつ $2 \leq H$ であれば、VとHの値は任意である。

【0059】

図2には、参照の便宜上、横方向のh軸と縦方向のv軸が示されている。また、図2には、各ブロックを参照するための、h軸方向のインデックス番号とv軸方向のインデックス番号も示されている。以下では、2つのインデックス番号を使って、例えば「ブロック(1, 2)」などと表記することがある。

【0060】

図2の例は、具体的には、第1の時点の画面110と、第1の時点より後の第2の時点の画面110とを比較すると、以下の(2a)～(2c)に示す領域111～113で変更が検出される場合の例である。

【0061】

(2a) ブロック(4, 1)とブロック(5, 1)とブロック(6, 1)とブロック(4, 2)とブロック(5, 2)とブロック(6, 2)とブロック(4, 3)とブロック(5, 3)とブロック(6, 3)からなる領域111。

(2b) ブロック(1, 2)からなる領域112。

(2c) ブロック(7, 5)からなる領域113。

【0062】

説明の便宜上、第1の時点よりも前のある時点を「第3の時点」といい、以下の(3a)～(3c)のように仮定する。

【0063】

(3a) 第3の時点から第2の時点までの間の期間において、領域111では高頻度で変更が生じている。例えば、領域111に含まれるどのブロックにおいても、当該期間中には、ある閾値以上の回数の変更が生じている。

(3b) 第3の時点から第2の時点までの期間において、領域112で変更が生じた頻度は低い。例えば、ブロック(1, 2)において当該期間中に生じた変更の回数は上記閾値未満である。

(3c) 第3の時点から第2の時点までの期間において、領域113で変更が生じた頻度は低い。例えば、ブロック(7, 5)において当該期間中に生じた変更の回数は上記閾値未満である。

【0064】

図2では、第1の時点から第2の時点までに変更が生じていないブロックを白で示している。また、図2では、第1の時点から第2の時点までに変更が生じているが上記期間中での変更頻度が低いブロックは網点パターンで示している。そして、図2では、第1の時点から第2の時点までに変更が生じており、かつ上記期間中での変更頻度が高いブロックは斜線パターンで示している。

【0065】

上記の(3a)～(3c)の仮定のもとでは、領域111に第2の形式が適用され、領域112と113には第1の形式が適用される。より具体的には、第2の時点の画面110をコンピュータ101に表示させるための第3の形式の画面情報は、以下の(4a)～(4c)のデータを含んでもよい。

10

【0066】

(4a) 動画像用の圧縮アルゴリズムにしたがって領域111を圧縮したデータであって、第2の時点の領域111をフレーム画像(例えば1枚のピクチャ)として含むデータ。

(4b) 第2の時点の領域112の静止画像のデータ。

(4c) 第2の時点の領域113の静止画像のデータ。

【0067】

もちろん、領域112と113に関しては、(4b)と(4c)の静止画像のデータの代わりに、ベクタオブジェクトまたは文字を表示するためのコマンドが使われてもよい。

【0068】

20

なお、第2の形式の適用対象の領域を矩形に整えるために、第1の時点から第2の時点までの間に変更が生じていないブロック、または第3の時点から第2の時点までの期間における変更頻度が低いブロックが、第2の形式の適用対象に含められてもよい。例えば、図2の例では、領域111が9個のブロックを含むが、9個のブロックでの変更頻度が同じであるとは限らない。例えば、以下の(5a)と(5b)がともに成り立つような場合もあり得る。

【0069】

(5a) ブロック(6, 1)では、第1の時点から第2の時点までの間に変更が生じていない。または、ブロック(6, 1)では、第1の時点から第2の時点までの間に変更が生じているが、第3の時点から第2の時点までの期間における変更頻度が低い。

30

(5b) 残りの8つのブロックでは、第1の時点から第2の時点までの間に変更が生じており、しかも、第3の時点から第2の時点までの期間における変更頻度が高い。

【0070】

上記の(5a)と(5b)がともに成り立つ場合、コンピュータ102は、(5b)に述べた8つのブロックにブロック(6, 1)を加えることにより、第2の形式の適用対象の領域を矩形に整えてもよい。そして、コンピュータ102は、矩形の領域111に対して、動画像の圧縮アルゴリズムによる圧縮を適用してもよい。

【0071】

以上のように、画面情報は、第1の形式と第2の形式を変更頻度に応じて適宜に組み合わせた第3の形式であることが好ましい。しかし、画面情報は、第1の形式または第2の形式であってもよい。

40

【0072】

ところで、画面情報がいずれの形式であっても、第1実施形態においては、コンピュータ102が、コンピュータ101から通知される値に基づく頻度で画面情報を送信する。つまり、第1実施形態では、コンピュータ101は、コンピュータ102に通知する値を決めることにより、コンピュータ102が画面情報を送信する頻度を間接的に制御することができる。

【0073】

ここで、コンピュータ102が画面情報を送信する頻度を制御することは、換言すれば、単位時間当たりにコンピュータ102からコンピュータ101に送信される画面情報の

50

量を制御することである。したがって、コンピュータ101は、コンピュータ102に通知する値をネットワークの状況に応じて適切に決めることにより、通信トラフィックの量をネットワークの状況に応じて制御することができる。

【0074】

具体的には、第1実施形態では、コンピュータ101は、コンピュータ101と102との間のネットワークの可用帯域幅の推定値、または、推定値の更新を反映する指標値を、コンピュータ102に通知する。そして、コンピュータ102は、通知された推定値または指標値に基づく頻度で、画面情報を送信する。

【0075】

コンピュータ101は、状況に応じて、可用帯域幅の推定値を決定する。

10

例えば、コンピュータ101は、最初に画面情報を受信したときに、可用帯域幅の推定値を初期化してもよい。より具体的には、コンピュータ101は、最初に画面情報を受信したときに、以下の(6a)と(6b)の値を用いて受信ビットレートを算出し、受信ビットレートを可用帯域幅の推定値の初期値として用いてもよい。

【0076】

(6a) 最初の画面情報をコンピュータ101が受信した時刻と、最初の画面情報に対応する時刻情報が示す時刻（つまり最初の画面情報をコンピュータ102が送信した時刻）との差。

(6b) 最初の画面情報のサイズ。

【0077】

20

また、コンピュータ101は、2回目以降に画面情報を受信したときに、下記の(7a)の時刻と(7b)の時刻の差に応じて、可用帯域幅の推定値を、増加させるか、維持するか、または減少させる。

【0078】

(7a) コンピュータ101が画面情報を実際に受信した時刻。

(7b) コンピュータ101が画面情報を受信するものと推定される (estimated) 時刻。

【0079】

換言すれば、画面情報の送信から受信までに実際にかかる時間と、画面情報の送信から受信までにかかる時間と推定される時間との差に応じて、コンピュータ101は、可用帯域幅の推定値を、増加させるか、維持するか、または減少させる。

30

【0080】

上記(7a)の時刻はネットワーク状況に依存する。また、可用帯域幅の推定値がどの程度良くネットワーク状況を反映しているかに応じて、(7a)の時刻と(7b)の時刻の差も変動する。よって、(7a)の時刻と(7b)の時刻の差に応じて可用帯域幅の推定値を決めることは、換言すれば、ネットワーク状況に応じて可用帯域幅の推定値を決めることである。

【0081】

なお、上記(7b)の時刻は、具体的には、以下の(8a)～(8c)の値から推定される時刻である。

40

【0082】

(8a) 可用帯域幅の現在の推定値。

(8b) コンピュータ101が受信した画面情報のサイズ。

(8c) 当該画面情報に対応する時刻情報により示される時刻。すなわち、当該画面情報がコンピュータ102から送信された時刻。

【0083】

例えば、上記の(7a)の時刻と(7b)の時刻の差が正の第1の閾値より大きい場合、コンピュータ101は、可用帯域幅の推定値を減少させてもよい。なぜなら、推定される受信時刻よりも実際の受信時刻が大幅に遅い場合、可用帯域幅が過大に推定されている可能性が高いからである。よって、この場合、コンピュータ101は、可用帯域幅の推定

50

値を減少させてもよい。

【0084】

また、第1の閾値未満の適宜の値が、第2の閾値として定められているものとする。上記の(7a)の時刻と(7b)の時刻の差が、第2の閾値以下の場合、可用帯域幅が過小に推定されている可能性がある。よって、この場合、コンピュータ101は、可用帯域幅の推定値を増加させてもよい。第2の閾値は、より具体的には、0または負であることが好ましい。

【0085】

そして、上記の(7a)の時刻と(7b)の時刻の差が、第2の閾値より大きく第1の閾値以下の場合、コンピュータ101は、可用帯域幅の推定値を維持してもよい。

10

なお、コンピュータ101は、可用帯域幅の推定値を減少させる場合には、具体的には、以下の(9a)と(9b)の値から算出される受信ビットレートに基づいて、可用帯域幅の推定値を減少させる。

【0086】

(9a) ある期間内にコンピュータ101が1回以上受信した画面情報それぞれのサイズ。

(9b) 上記期間の長さ。

【0087】

コンピュータ101と102の間のポイント・ツー・ポイントの実際の可用帯域幅を正確に観測することは困難である。しかし、ある期間における実際の可用帯域幅の平均値は、少なくとも、当該期間における平均受信ビットレート以上である。よって、コンピュータ101は、当該期間における可用帯域幅の下限值として、当該期間における実際の平均受信ビットレートを利用してもよい。具体的には、コンピュータ101は、上記のように受信ビットレートに基づいて推定値を減少させることが好ましい。

20

【0088】

他方、コンピュータ101は、可用帯域幅の推定値を増加させる場合は、例えば、推定値を所定の割合で増加させてもよい。あるいは、コンピュータ101は、推定値に所定の値を足すことにより、推定値を増加させてもよい。

【0089】

また、以下の(10a)～(10c)の値から算出される直近の受信ビットレートが、ある程度大きい場合、コンピュータ101は、直近の受信ビットレートに応じて可用帯域幅の推定値を増加させてもよい。

30

【0090】

(10a) コンピュータ101が直近に受信した画面情報についての時刻情報により示される送信時刻。

(10b) コンピュータ101が直近に画面情報を受信した時刻。

(10c) コンピュータ101が直近に受信した画面情報のサイズ。

【0091】

例えば、可用帯域幅の推定値に基づいて、推定値よりも大きな判定基準値が定められていてもよい。例えば、1より大きい定数を使って、判定基準値が、現在の推定値の定数倍と定められていてもよい。直近の受信ビットレートが判定基準値よりも大きい場合、実際の可用帯域幅が最近急増した可能性がある。また、直近の受信ビットレートは、直近の可用帯域幅の下限值と見なせる。

40

【0092】

よって、コンピュータ101は、(7a)の時刻と(7b)の時刻の差とは関係なく、直近の受信ビットレートに応じて可用帯域幅の推定値を増加させてもよい。もちろん、コンピュータ101は、(7a)の時刻と(7b)の時刻の差と、直近の受信ビットレートの双方を考慮に入れて、可用帯域幅の推定値を増加させてもよい。

【0093】

コンピュータ101は、少なくとも可用帯域幅の推定値を増加または減少させることで

50

更新した場合には、推定値または推定値の更新を反映する指標値をコンピュータ102に通知する。指標値は、例えば、更新後の推定値と更新前の推定値の差分を示す値でもよい。

【0094】

例えば、図2の例では、ステップS103、S108、S113、S116、S119、およびS124に示すように、コンピュータ101は、画面情報と時刻情報を受信するたびに、推定値をコンピュータ102に通知する。実施形態によっては、コンピュータ101は、推定値を維持する場合には、推定値をコンピュータ102に通知しなくてもよい。

【0095】

また、コンピュータ101は、推定値を増加させた場合に、推定値の増加を強調して示す指標値をコンピュータ102に通知してもよい。例えば、コンピュータ101は、増加させた推定値よりもさらに大きく指標値を定めて、指標値をコンピュータ102に通知してもよい。

【0096】

実施形態によっては、コンピュータ101は、推定値を維持する場合に、推定値自体の代わりに、推定値よりも大きな値をコンピュータ102に通知してもよい。例えば、(7a)の時刻と(7b)の時刻の差が所定の閾値以下の場合、コンピュータ101は、推定値自体は変えずに、推定値よりも大きな値をコンピュータ102に通知してもよい。

【0097】

コンピュータ102は、推定値自体をコンピュータ101がコンピュータ102に通知したのか、それとも推定値とは別の値をコンピュータ101がコンピュータ102に通知したのかを区別しなくてもよい。コンピュータ102は、単に、コンピュータ101から通知された値に応じて、画面情報を送信する頻度を調整してもよい。具体的には、コンピュータ101から通知される値が大きいほど、コンピュータ102は、画面情報を送信する頻度を上げる。ただし、頻度には上限があってもよい。

【0098】

以上のようにして、第1実施形態によれば、ネットワーク状況を反映する可用帯域幅の推定値に応じて、コンピュータ102がコンピュータ101に画面情報を送信する頻度が調整される。つまり、第1実施形態では、通信トラフィックの量がネットワークの状況に応じて調整される。よって、過剰な通信トラフィックのせいで遅延が累積する事態は回避される。つまり、遅延が累積することによるユーザビリティの低下は回避される。

【0099】

また、以上説明したとおり、第1実施形態では、画像（静止画像、動画像、またはその双方）の品質を落とすことで通信トラフィックの量を減らすのではなく、画面情報を送信する頻度を落とすことで通信トラフィックの量を減らすアプローチが採用されている。そのため、個々の瞬間においてコンピュータ101の画面に表示される画像の品質は保たれる。画像の品質を保つことは、ユーザビリティの低下を防止する効果がある。

【0100】

例えば、色深度を減少させたり、圧縮符号化において量子化を粗くしたりするなど、画質を落とすことで通信トラフィックの量の低減を図ることも可能ではある。しかしながら、画質が落ちると、例えば、「コンピュータ101の画面に文字がぼやけて表示されるため、ユーザ103が文字を読み取れない」などの問題が起きるおそれがある。文字が読み取れないことはユーザビリティの低下に直結する。また、高精細な（high-definition）画像（静止画像、動画像、またはその双方）を扱うアプリケーションをユーザ103が使いたい場合も、画質の低下はユーザビリティを大きく損ねる。

【0101】

したがって、ユーザビリティの観点からは、画質を落とすことは好ましくない。換言すれば、第1実施形態のように頻度の調整により通信トラフィックの量の低減を図るアプローチには、ネットワーク状況が悪くてもユーザビリティの悪化を抑える効果がある。

10

20

30

40

50

【0102】

続いて、コンピュータ101と102が以上説明したように動作することによって生じ得る具体的な動作シーケンスの例について、図1を参照しながら説明する。

【0103】

ステップS101より前の時間は図1では省略されているが、上記のとおりコンピュータ102は、コンピュータ101からの通知に応じて1つ以上のプログラムを実行しており、コンピュータ101の画面に表示するための画像を生成する。そして、ステップS101でコンピュータ102は、生成した画像に基づいて画面情報を生成し、画面情報と時刻情報を送信する。

【0104】

すると、ステップS102でコンピュータ101は、画面情報にしたがって画面を更新する。また、ステップS103でコンピュータ101は、上記の(8a)～(8c)の値から(7b)の時刻を推定し、(7a)の時刻と(7b)の時刻の差に応じて、可用帯域幅の推定値を、増加させるか、維持するか、または減少させる。そして、コンピュータ101は、可用帯域幅の推定値をコンピュータ102に通知する。

【0105】

なお、コンピュータ101は、ステップS102とS103のどちらを先に実行してもよい。以下のステップにおいても、コンピュータ101は、画面情報と時刻情報を受信したとき、画面の更新と推定値の通知のどちらを先に実行してもよい。

【0106】

一方、ユーザ103は、コンピュータ101の画面の更新とは独立したタイミングで、入力装置を介してコンピュータ101に入力を与える。例えば、ステップS104でユーザ103は入力操作を行う。すると、ステップS105でコンピュータ101は、入力操作の内容をコンピュータ102に通知する。コンピュータ102は、プログラムを実行する過程において、通知された入力に応じて適宜の処理を行う。

【0107】

また、コンピュータ102は、ステップS103で通知された値に応じた頻度で画面情報をコンピュータ101に送信しようとする。具体的には、ステップS101での送信からある時間A1が経過した後、コンピュータ102は、ステップS106において、コンピュータ101の画面に表示するための画像を生成し、画像に基づいて画面情報を生成し、画面情報と時刻情報を送信する。

【0108】

すると、ステップS107でコンピュータ101は、画面情報にしたがって画面を更新する。また、ステップS108でコンピュータ101は、可用帯域幅の推定値を、増加させるか、維持するか、または減少させる。そして、コンピュータ101は、可用帯域幅の推定値をコンピュータ102に通知する。

【0109】

また、ユーザ103はステップS109においても何らかの入力操作を行う。すると、ステップS110でコンピュータ101は、入力操作の内容をコンピュータ102に通知する。コンピュータ102は、プログラムを実行する過程において、通知された入力に応じて適宜の処理を行う。

【0110】

コンピュータ102は、ステップS108で通知された値に応じた頻度で画面情報をコンピュータ101に送信しようとする。具体的には、ステップS106での送信からある時間A2が経過した後、コンピュータ102は、ステップS111において、コンピュータ101の画面に表示するための画像を生成し、画像に基づいて画面情報を生成し、画面情報と時刻情報を送信する。

【0111】

すると、ステップS112でコンピュータ101は、画面情報にしたがって画面を更新する。また、ステップS113でコンピュータ101は、可用帯域幅の推定値を、増加さ

10

20

30

40

50

せるか、維持するか、または減少させる。

【0112】

例えば、ステップS111で今までよりもネットワーク状況が悪化した場合、ステップS111での画面情報と時刻情報の送信から受信までにかかる時間は、コンピュータ101が可用帯域幅の推定値から推定する時間よりも長い。よって、この場合、コンピュータ101は、ステップS113で可用帯域幅の推定値を減少させてもよく、減少させた推定値をコンピュータ102に通知してもよい。

【0113】

コンピュータ102は、ステップS113で通知された値に応じた頻度で画面情報をコンピュータ101に送信しようとする。具体的には、ステップS111での送信からある時間A3が経過した後、コンピュータ102は、ステップS114において、コンピュータ101の画面に表示するための画像を生成し、画像に基づいて画面情報を生成し、画面情報と時刻情報を送信する。

10

【0114】

例えば上記のように、ステップS113で今までよりも小さな値がコンピュータ101から通知された場合、コンピュータ102は、画面情報を送信する頻度を下げる。よって、時間A3は、時間A1やA2よりも長い。

【0115】

さて、コンピュータ101は、ステップS114で送信された画面情報と時刻情報を受信すると、ステップS115で、画面情報にしたがって画面を更新する。また、ステップS116でコンピュータ101は、可用帯域幅の推定値を、増加させるか、維持するか、または減少させる。そして、コンピュータ101は、可用帯域幅の推定値をコンピュータ102に通知する。

20

【0116】

ところで、ステップS116の送信からしばらく時間が経った後、ネットワーク状況が好転したとする。ステップS117でコンピュータ102は画面情報と時刻情報を送信するが、当該送信にかかる時間は、ネットワーク状況の好転の結果として、短くて済む。例えば、説明の便宜上、コンピュータ101が可用帯域幅の推定値から推定する時間よりも短い時間で、画面情報と時刻情報はコンピュータ101に到着したとし、その結果、コンピュータ101は推定値を増加させるものとする。

30

【0117】

具体的には、コンピュータ101はステップS118で画面情報にしたがって画面を更新し、ステップS119で上記のように増加させた推定値をコンピュータ102に通知する。

【0118】

また、ユーザ103は好きなタイミングで入力操作を行い得るので、例えばステップS120においてもユーザ103から入力装置を介してコンピュータ101に入力が与えられる。すると、ステップS121でコンピュータ101は、入力操作の内容をコンピュータ102に通知する。コンピュータ102は、プログラムを実行する過程において、通知された入力に応じて適宜の処理を行う。

40

【0119】

そして、コンピュータ102は、ステップS119で通知された値に応じた頻度で画面情報をコンピュータ101に送信しようとする。具体的には、ステップS117での送信からある時間A4が経過した後、コンピュータ102は、ステップS122において、コンピュータ101の画面に表示するための画像を生成し、画像に基づいて画面情報を生成し、画面情報と時刻情報を送信する。

【0120】

例えば、上記のように、ネットワーク状況が悪かった間にコンピュータ101がコンピュータ102に通知していた値よりも大きな値をステップS119でコンピュータ101がコンピュータ102に通知した場合、コンピュータ102は、画面情報を送信する頻度

50

を上げる。よって、時間 A 4 は、時間 A 3 よりも短い。

【0121】

コンピュータ101は、ステップS122で送信された画面情報と時刻情報を受信すると、ステップS123で、画面情報にしたがって画面を更新する。また、ステップS124でコンピュータ101は、可用帯域幅の推定値を、増加させるか、維持するか、または減少させる。そして、コンピュータ101は、可用帯域幅の推定値をコンピュータ102に通知する。

【0122】

例えば以上例示したステップS101～S124のようにして、画面情報の送信頻度はネットワーク状況に応じて調整される。よって、ネットワーク状況がたとえ悪化しても、遅延の累積によるユーザビリティの低下は回避される。また、個々の瞬間における画質は保たれるので、画質の低下によるユーザビリティの低下は生じない。

10

【0123】

ところで、図1には、コンピュータ101が可用帯域幅の推定値をコンピュータ102に通知するステップが破線矢印で描かれており、それ以外のステップが実線矢印で描かれている。上記の説明から明らかなとおり、破線矢印で描かれたステップは、画面情報の送信頻度の調整のために追加的に生じる通信であるが、可用帯域幅の推定値のデータサイズはごくわずかである。また、画面情報の送信頻度の調整のために、時刻情報がコンピュータ102からコンピュータ101へと送信されるが、時刻情報のデータサイズもごくわずかである。

20

【0124】

よって、第1実施形態には、「ネットワーク状況の把握のために追加的に生じる通信トラフィックの量が、ネットワーク状況に影響を与えない程度のごくわずかな量に過ぎない」という利点もある。換言すれば、第1実施形態では、「ネットワークに大きな追加的負荷をかけなくても、ネットワーク状況を推定することができ、推定したネットワーク状況に応じて通信トラフィックの量を調整することができる」という利点がある。

【0125】

以上説明したような種々の利点のある第1実施形態は、シンククライアントシステムに好適である。以下に、シンククライアントシステムへの第1実施形態の適用に関して補足説明を行う。

30

【0126】

シンククライアントシステムでは、クライアントとサーバがネットワークを介して接続される。そのため、クライアントにおける操作レスポンス性能はネットワーク状況に依存する。換言すれば、ユーザビリティがネットワーク状況に依存する。しかし、常にネットワーク状況が良好であるとは限らない。

【0127】

第1実施形態によれば、たとえネットワーク状況が悪くてもユーザビリティの低下を緩和することができるような方法で、クライアントの画面更新が制御される。なぜなら、ネットワーク状況に応じて画面情報の送信頻度が調整され、ネットワーク状況の把握のためにネットワークに追加的に生じる負荷は無視して構わない程度のレベルであり、個々の瞬間での画質の低下は生じないからである。

40

【0128】

また、シンククライアントシステムの利用が広まるのにつれて、ユーザがシンククライアントを行って行う作業の種類も、多様化が進むと予測される。よって、シンククライアントの画面の更新は、作業の種類によらずにユーザが快適にシンククライアントを使って作業を進めることができるように制御されることが好ましい。

【0129】

例えば、ユーザは、テキスト文書の作成や電子メールの送受信などの作業だけでなく、高精細な画像（静止画像、動画像、または双方）を扱うアプリケーションを使った作業も、シンククライアント上で行いたい場合があり得る。高精細な画像を扱うアプリケーション

50

の例として、例えば、C A D（Computer Aided Design）アプリケーションや、フォトレタッチアプリケーションなどがある。

【0130】

高精細な画像のデータサイズは大きいため、ある種の遠隔制御方法では、実用上満足得られる操作レスポンス性能が達成されないことがしばしば起こり得る。つまり、クライアントとサーバ間の可用帯域幅が、実用上満足得られる短い時間内に高精細な画像のデータをサーバからクライアントに送信するには、不十分な場合が多々ある。

【0131】

しかし、第1実施形態では、高精細な画像を扱うアプリケーションが使われる場合でも、ユーザビリティの低下は比較的軽度である。なぜなら、第1実施形態によれば、画面情報のデータサイズが大きくても、画面情報をネットワーク状況に応じた頻度で送信することにより、遅延の累積による操作レスポンス性能の悪化を防ぐことができ、しかも、個々の時点での画質は保たれるからである。

10

【0132】

なお、画面情報が送信される頻度は、換言すれば、ユーザが見ているクライアントの画面が更新される頻度である。そして、当該頻度が低くなることで、多少ユーザビリティが低下する可能性はある。なぜなら、画面の更新頻度が低い場合、動画像の動きがぎくしゃくして見えることがあるからである。

【0133】

しかしながら、遅延の累積による操作レスポンス性能の悪化や、個々の瞬間における画質の低下がユーザビリティに与える悪影響と比べると、画面の更新頻度の低下がユーザビリティに与える悪影響は小さい。よって、第1実施形態は、ユーザビリティの低下を抑制または緩和するという点において、優れている。

20

【0134】

また、仮に高精細な画像を扱うアプリケーションをユーザが使わないとしても、クライアントとサーバ間の可用帯域幅が狭ければ、ある種の遠隔制御方法では、実用上満足得られる操作レスポンス性能が達成されないおそれが高い。そして、スマートフォンやタブレット端末やノート型P C（Personal Computer）などの携帯端末をモバイル環境下で利用するユーザにとっては、可用帯域幅は必ずしも十分に広いわけではない。例えば、社内の有線L A N（Local Area Network）でクライアントとサーバが接続されている環境と比較すると、クライアントが公衆無線L A Nを介してサーバと接続されている環境では、可用帯域幅がかなり狭い。

30

【0135】

しかし、第1実施形態によれば、可用帯域幅が狭ければ、画面情報の送信頻度が、狭い可用帯域幅に応じた小さな値に調整される。その結果、遅延の累積は回避される。よって、第1実施形態によれば、モバイル環境下でも、実用上問題ない操作レスポンス性能が得られる。

【0136】

以上説明したように、第1実施形態は、今後利用の拡大が見込まれるシンクライアントシステムにも好適である。

40

【0137】

続いて、図3と4を参照して、第1実施形態の効果についてさらに説明する。

図3には、5つの模式的なタイミングチャート121～125が示されている。タイミングチャート121は、コンピュータ102が画面情報F1～F7を送信するタイミングを示す。説明の便宜上、図3には、コンピュータ102がある一定の頻度で画面情報F1～F7を送信する場合のタイミングチャート121が示されている。

【0138】

コンピュータ101と102の間の実際の可用帯域幅が十分な場合は、画面情報F1～F7は、タイミングチャート122に示すようにコンピュータ101に受信される。つまり、送信から受信まで（より具体的には、送信の開始から受信の完了まで）、ある程度の

50

遅延は生じるものの、「遅延が累積して次第に遅延時間が長くなる」という事態は生じない。よって、ある程度満足な操作レスポンス性能が得られる。タイミングチャート122に示す受信頻度は、換言すれば、コンピュータ101における表示フレームレートである。

【0139】

他方、タイミングチャート123は、第1実施形態と比較するための比較例において、コンピュータ101と102の間の実際の可用帯域幅が不十分な場合に、コンピュータ101が画面情報F1～F5を受信するタイミングを示す。比較例では、コンピュータ102が画面情報F1～F7を送信する頻度を調整しないものとする。つまり、比較例では、コンピュータ102は、コンピュータ101と102の間の実際の可用帯域幅が不十分な場合にも、タイミングチャート121と同じ頻度で画面情報F1～F5を送信するものとする。

10

【0140】

可用帯域幅が不十分な場合、例えば画面情報F1がコンピュータ102からコンピュータ101への経路上のノードにおいて、当該経路上のリンクが空くのを待つために滞留してしまうことがある。具体的には、画面情報F1は、コンピュータ102内の送信バッファや、経路上のルータ内のバッファなどで、しばらく滞留することがある。同様に、画面情報F2～F5も、コンピュータ102からコンピュータ101への経路上のどこかで滞留することがある。

【0141】

20

このように、比較例においては、可用帯域幅が不十分な場合にコンピュータ102がタイミングチャート121の頻度で画面情報F1～F5を送信すると、コンピュータ101は、タイミングチャート123に示すように画面情報F1～F5を受信する。タイミングチャート121と123を比較すると分かるように、滞留の結果として、画面情報F1～F5の送信間隔よりも画面情報F1～F5の受信間隔の方が長い。

【0142】

そして、受信間隔が長くなると、遅延の累積によって次第にレスポンス性能が悪化し、時間の進み方が遅くなったようにユーザに感じられる。つまり、下記の(11a)～(11e)の遅延時間D1～D5は、 $D1 < D2 < D3 < D4 < D5$ という関係にある。

【0143】

30

- (11a) 画面情報F1の送信から受信までにかかる遅延時間D1。
- (11b) 画面情報F2の送信から受信までにかかる遅延時間D2。
- (11c) 画面情報F3の送信から受信までにかかる遅延時間D3。
- (11d) 画面情報F4の送信から受信までにかかる遅延時間D4。
- (11e) 画面情報F5の送信から受信までにかかる遅延時間D5。

【0144】

なお、遅延時間D1は、より具体的には、画面情報F1の送信が開始されてから画面情報F1の受信が完了するまでにかかる時間である。他の遅延時間D2～D5も同様である。

【0145】

40

タイミングチャート123に示すようにコンピュータ101が画面情報F1～F5を受信する場合、ユーザ103にとっては、時間の進み方が遅くなったように感じられる。例えば、「1秒間かけてユーザ103がマウス操作を行うと、当該マウス操作に応じて画面上のマウスポインタが1.5秒間かけて移動する」といったような事態が生じ得る。つまり、比較例では、「入力と表示の間で時間の進み方が異なる」という違和感をユーザ103に感じさせてしまうおそれが多いにある。

【0146】

さらに、上記のとおり次第に遅延時間が長くなるので、やがては、例えば、「入力操作を行ってから10秒以上もたってから、やっと、当該入力操作に応じて画面が変化する」といった事態も生じ得る。つまり、遅延の累積によって、やがては操作レスポンス性能が

50

非常に悪化してしまいかねない。

【0147】

以上のとおりなので、タイミングチャート123に示すような比較例は、ユーザビリティを大きく悪化させるおそれがある。しかし、第1実施形態では、比較例とは異なり、画面情報の送信頻度が調整される。

【0148】

例えば、コンピュータ101がコンピュータ102に通知する可用帯域幅の推定値に応じて、コンピュータ102は、タイミングチャート121に示す頻度の半分の頻度で画面情報を送信してもよい。タイミングチャート124は、そのように頻度を半減させる調整を第1実施形態においてコンピュータ102が行う場合を示す。

10

【0149】

タイミングチャート121と124を比較すると分かるように、タイミングチャート124の例では、画面情報F2、F4、およびF6は送信されない。タイミングチャート124の例では、コンピュータ102が画面情報の送信頻度を落とすことで、単位時間当たりの通信トラフィックの量が減る。

【0150】

その結果、遅延の累積は回避される。具体的には、コンピュータ101は、タイミングチャート125に示すように画面情報F1、F3、F5およびF7を受信することができる。つまり、各画面情報の送信から受信まで、ある程度の遅延は生じるものの、遅延の累積による遅延の長期化は回避される。よって、ある程度満足な操作レスポンス性能が得られる。

20

【0151】

また、画面情報F1、F3、F5およびF7がそれぞれ示す個々の時点の画面の品質は、第1実施形態では落とされない。よって、高精細な画像や細かい文字でも、くっきりとコンピュータ101の画面上に表示される。

【0152】

続いて、図4を参照して第1実施形態の効果についてさらに説明する。図4には、あるCADアプリケーションが利用される場合についての、遅延時間に関するグラフ131とフレームレートに関するグラフ132が示されている。グラフ131と132において、白い棒グラフは、コンピュータ102が画面情報の送信頻度を調整しない比較例を示し、斜線パターンの棒グラフは、第1実施形態を示す。

30

【0153】

具体的には、図4に関して、比較例と第1実施形態は、次の(12a)～(12c)の点で共通である。

【0154】

(12a) ユーザ103は、CADアプリケーションを起動する入力操作と、CADアプリケーション上での拡大、縮小、回転などのための入力操作を行う。

(12b) コンピュータ102は、コンピュータ101から通知される入力操作の内容に応じて、CADアプリケーションを実行し、コンピュータ101の画面に表示するための画像を生成する。そして、コンピュータ102は、生成した画像をコンピュータ101に表示させるための画面情報を生成し、画面情報をコンピュータ101に送信する。

40

(12c) コンピュータ101は、受信した画面情報にしたがって画面を更新する。

【0155】

しかし、比較例では下記の(13a)～(13c)の処理は行われないのに対して、第1実施形態では、下記の(13a)～(13c)の処理が行われる。

【0156】

(13a) コンピュータ102は、画面情報とともに時刻情報もコンピュータ101に送信する。

(13b) コンピュータ101は、時刻情報を用いて、可用帯域幅の推定値を増加させるか、維持するか、それとも減少させるかを決定する。そして、コンピュータ101は

50

、可用帯域幅の推定値（または推定値の更新を反映する指標値）をコンピュータ102に通知する。

（13c）コンピュータ102は、通知された値に応じて、画面情報の送信頻度を調整する。

【0157】

具体的には、グラフ131には、拡大、縮小、および回転という入力操作の種類ごとに、ユーザ103が入力操作をそれぞれ複数回行ったときの平均遅延時間が示されている。なお、グラフ131における遅延時間は、ユーザ103が入力操作を行った時点から、当該入力操作に応じた画像がコンピュータ101の画面に表示されるまでの時間である。白い棒グラフと斜線パターンの棒グラフを比較すると分かるように、比較例よりも第1実施形態の方が、遅延時間が短い。よって、第1実施形態はユーザビリティの改善に有益である。

10

【0158】

また、グラフ132には、拡大、縮小、および回転という入力操作の種類ごとに、ユーザ103が入力操作をそれぞれ複数回行ったときの平均表示フレームレートが示されている。白い棒グラフが示す比較例での表示フレームレートは、例えば図3のタイミングチャート123に例示するように、遅延の累積の結果として送信フレームレートよりも下がってしまった表示フレームレートを示す。他方、斜線パターンの棒グラフが示す第1実施形態の表示フレームレートは、コンピュータ101から通知される値に応じてコンピュータ102が画面情報の送信頻度を調整した結果として下がったフレームレートを示す。

20

【0159】

グラフ132では、比較例と第1実施形態のフレームレートの差があまりない。このことは、画面情報の送信頻度（換言すれば送信フレームレート）の調整（すなわち可用帯域幅の推定に基づく調整）が適切であり、実際の可用帯域幅が十分に活用されていることを示す。つまり、第1実施形態によれば、フレームレートが過度に下げられてしまうこと（そして、その結果として画面の表示が必要以上にぎくしゃくしてしまうこと）も防げる。

【0160】

さて、図5は、ハードウェア構成例を示す図である。図5に示すハードウェア構成は、第1～第4実施形態のいずれにも適用可能である。図5には、ネットワーク220を介して互いに接続されたコンピュータ200と240が例示されている。

30

【0161】

例えば、第1実施形態に関して図1に示したコンピュータ101は、図5のコンピュータ200であってもよい。この場合、図1のコンピュータ102は、図5のコンピュータ240であってもよい。

【0162】

また、第2～第4実施形態に関して後述する図6のサーバ310は、図5のコンピュータ240であってもよい。この場合、図6のクライアント320は、図5のコンピュータ200であってもよい。

【0163】

コンピュータ200と240の種類は任意である。例えば、コンピュータ200は、エンドユーザが使うユーザ端末でもよい。また、コンピュータ240は、クラウドサービス的一种としてリモートデスクトップサービスを提供するサーバでもよい。ユーザ端末の例としては、デスクトップ型のPC、ノート型のPC、タブレット端末、PDA（Personal Digital Assistant）、スマートフォン、ネットワーク機能を持つメディアプレイヤなどが挙げられる。

40

【0164】

ネットワーク220の種類は任意である。ネットワーク220は、例えば、LAN、WAN（Wide Area Network）、インターネット、および、それらの組み合わせのいずれであってもよい。また、ネットワーク220は、有線ネットワークでもよく、無線ネットワークでもよく、両者の組み合わせであってもよい。

50

【0165】

図5のコンピュータ200は、CPU (Central Processing Unit) 201と、ROM (Read Only Memory) 202と、RAM (Random Access Memory) 203と、通信インタフェース204を有する。コンピュータ200はさらに、入力装置205と、出力装置206と、記憶装置207と、コンピュータ読み取り可能な記憶媒体210の駆動装置208を有する。コンピュータ200の各構成要素は、バス209によって互いに接続されている。

【0166】

CPU 201は、シングルコアまたはマルチコアのプロセッサである。コンピュータ200は、複数のプロセッサを有していてもよい。例えば、コンピュータ200は、2台以上のCPU 201を有していてもよいし、CPU 201に加えてGPU (Graphics Processing Unit) などのプロセッサを有していてもよい。CPU 201は、プログラムをRAM 203にロードし、RAM 203をワーキングエリアとしても利用しながら、プログラムを実行する。

10

【0167】

通信インタフェース204は、例えば、有線LANインタフェース、無線LANインタフェース、またはその組み合わせである。コンピュータ200は、通信インタフェース204を介してネットワーク220に接続される。

【0168】

通信インタフェース204は、具体的には、外付けのNIC (Network Interface Card) でもよいし、オンボード型のネットワークインタフェースコントローラでもよい。例えば、通信インタフェース204は、物理層の処理を行う「PHYチップ」と呼ばれる回路と、MAC (Media Access Control) 副層の処理を行う「MACチップ」と呼ばれる回路を含んでいてもよい。

20

【0169】

入力装置205は、例えば、キーボード、ポインティングデバイス、またはその組み合わせである。ポインティングデバイスは、例えば、マウスでもよいしタッチパッドでもよいしタッチスクリーンでもよい。出力装置206は、ディスプレイ、スピーカ、またはその組み合わせである。ディスプレイはタッチスクリーンであってもよい。

【0170】

記憶装置207は、例えばHDD (Hard Disk Drive) やSSD (Solid-State Drive) などの不揮発性の記憶装置である。

30

【0171】

記憶媒体210の例は、CD (Compact Disc) やDVD (Digital Versatile Disk) などの光ディスク、光磁気ディスク、磁気ディスク、フラッシュメモリなどの半導体メモリカードなどである。

【0172】

CPU 201が実行するプログラムは、予めROM 202または記憶装置207にインストールされていてもよい。あるいは、プログラムは、記憶媒体210に格納されて提供され、記憶媒体210から駆動装置208により読み取られて記憶装置207にコピーされ、その後、RAM 203にロードされてもよい。または、ネットワーク220上のプログラム提供者230から、ネットワーク220と通信インタフェース204を介して、プログラムがコンピュータ200にダウンロードされ、インストールされてもよい。プログラム提供者230は、具体的には、コンピュータ200ともコンピュータ240とも異なる、他のコンピュータである。

40

【0173】

さて、コンピュータ240は、CPU 241と、ROM 242と、RAM 243と、通信インタフェース244と、入力装置245と、出力装置246と、記憶装置247と、コンピュータ読み取り可能な記憶媒体250の駆動装置248を有する。コンピュータ240の各構成要素は、バス249によって互いに接続されている。コンピュータ240の

50

構成はコンピュータ200の構成と類似なので、詳しい説明は省略するが、何点か補足すると下記のとおりである。

【0174】

図5には1台のCPU241のみが描かれているが、コンピュータ240は、コンピュータ200と同様に、複数のプロセッサを有していてもよい。コンピュータ240は、CPU241に加えて、さらに、動画像の圧縮用の専用ハードウェア回路を有していてもよい。

【0175】

CPU241が実行するプログラムは、予めROM242または記憶装置247にインストールされていてもよい。あるいは、プログラムは、記憶媒体250に格納されて提供され、記憶媒体250から駆動装置248により読み取られて記憶装置247にコピーされてもよい。または、プログラム提供者230から、ネットワーク220と通信インタフェース244を介して、プログラムがコンピュータ240にダウンロードされ、インストールされてもよい。

10

【0176】

なお、ROM202、RAM203、記憶装置207、記憶媒体210、ROM242、RAM243、記憶装置247、および記憶媒体250は、いずれも、有形の(tangible)記憶媒体であり、信号搬送波のような一時的な(transitory)媒体ではない。

【0177】

続いて、図6～12を参照して第2実施形態について説明する。詳しくは後述するとおり、図6～8および10～12は、第3～第4実施形態とも関わる。第2～第4実施形態のいずれにおいても、第1実施形態と同様の効果が得られる。

20

【0178】

図6は、システムのブロック構成図である。図6のシステム300は、サーバ310とクライアント320を含む。上記のとおり、サーバ310は例えば図5のコンピュータ240であってもよく、クライアント320は図5のコンピュータ200でもよい。図6ではネットワークが省略されているが、図5に例示したように、サーバ310とクライアント320はネットワークを介して接続されている。

【0179】

サーバ310は、受信部311と、生成部312と、送信部313と、調整部314と、付加部315を有する。また、サーバ310は、送信制御データ316を保持する。

30

【0180】

受信部311は、クライアント320からデータを受信する。クライアント320からサーバ310に送信されるデータには、少なくとも2種類ある。

【0181】

第1の種類のデータは、ユーザからクライアント320への入力をサーバ310に通知するためのデータである。受信部311は、第1の種類のデータを受信すると、受信したデータを生成部312に出力する。

【0182】

第2の種類のデータは、後述するように可用帯域幅と過剰遅延時間をサーバ310に通知するためのデータである。過剰遅延時間は、実際の受信時刻が、可用帯域幅から推定される受信時刻よりもどの程度遅延しているかを示す。受信部311は、第2の種類のデータを受信すると、受信したデータを調整部314に出力する。

40

【0183】

生成部312は、クライアント320の画面に表示するための画像を生成する。具体的には、生成部312は、クライアント320への入力の通知(つまり上記の第1の種類のデータ)に基づいて、1つまたは複数のプログラムを実行し、プログラムの実行に応じて画像を生成する。そして、生成した画像に基づいて、生成部312は、クライアント320に画像を画面上に表示させるための画面情報(具体的には後述の描画コマンド)を生成する。送信部313は、画面情報をクライアント320に送信する。

50

【0184】

より具体的には、画面情報がクライアント320に送信される頻度は、調整部314により調整される。換言すれば、調整部314は、生成部312が画面情報を出力する頻度を制御する。調整部314は、送信制御データ316を用いて頻度の調整を行う。送信制御データ316の具体例は図7とともに後述する。

【0185】

また、画面情報がクライアント320に送信されるときには、現在時刻を示す時刻情報が付加部315により画面情報に付加され、画面情報は時刻情報とともに送信部313から送信される。すなわち、画面情報は、画面情報が送信される時刻を示す時刻情報とともに、送信される。

10

【0186】

なお、サーバ310が図5のコンピュータ240により実現される場合、受信部311は、CPU241と通信インタフェース244により実現されてもよい。同様に、送信部313も、CPU241と通信インタフェース244により実現されてもよい。生成部312と調整部314と付加部315は、いずれも、CPU241により実現されてもよい。送信制御データ316は、RAM243に記憶されることが好ましいが、記憶装置247に記憶されてもよい。

【0187】

一方、クライアント320は、入力部321と、表示部322と、送信部323と、受信部324と、描画部325と、推定部326を有する。また、クライアント320は、受信履歴データ327と、時刻差データ328と、可用帯域幅データ329を保持する。

20

【0188】

入力部321は、ユーザからの入力を受け付ける。例えば、入力部321は図5の入力装置205であってもよい。

【0189】

表示部322は、画面を表示する。例えば、表示部322は図5の出力装置206であってもよい。

【0190】

送信部323は、入力部321を介してユーザからクライアント320に与えられた入力をサーバ310に通知するためのデータや、可用帯域幅と過剰遅延時間をサーバ310に通知するためのデータを、サーバ310に送信する。送信部323は、例えば、図5のCPU201と通信インタフェース204によって実現されてもよい。

30

【0191】

受信部324は、サーバ310から、時刻情報とともに画面情報を受信する。また、詳しくは後述するように、サーバ310は、クライアント320の時計が示す時刻と、サーバ310の時計が示す時刻との時刻差を算出するためのデータ（以下では説明の便宜上、「時刻合わせコマンド」ともいう）も送信する。よって、受信部324は、画面情報だけでなく時刻合わせコマンドも受信する。

【0192】

受信部324は、サーバ310からデータを受信するたびに、受信履歴データ327を更新する。また、受信部324は、時刻合わせコマンドを受信すると、クライアント320の時計が示す時刻と、サーバ310の時計が示す時刻との時刻差を算出し、算出結果を時刻差データ328として記憶する。

40

【0193】

受信履歴データ327と時刻差データ328の具体例は図7とともに後述する。なお、クライアント320が図5のコンピュータ200により実現される場合、受信履歴データ327と時刻差データ328は、RAM203に記憶されることが好ましいが、記憶装置207に記憶されてもよい。なお、送信部323と同様に、受信部324も、CPU201と通信インタフェース204によって実現されてもよい。

【0194】

50

描画部 325 は、受信部 324 が受信した画面情報に基づいて、画面の一部または全部を描画する。描画部 325 は、CPU 201（または、不図示の GPU などの他のプロセッサ）によって実現されてもよい。

【0195】

推定部 326 は、受信履歴データ 327 と時刻差データ 328 を用いて、サーバ 310 とクライアント 320 の間のネットワークの可用帯域幅を推定する。可用帯域幅の推定値は、可用帯域幅データ 329 として記憶される。可用帯域幅データ 329 の具体例は図 7 とともに後述する。

【0196】

より詳しくは、可用帯域幅の推定に際して、推定部 326 は、過剰遅延時間を算出する。過剰遅延時間は、実際に受信部 324 が画面情報を受信した時刻と、現在の可用帯域幅データ 329 から推定される受信時刻との差である。推定部 326 は、過剰遅延時間に応じて、可用帯域幅の推定値を増加させるか、維持するか、または減少させる。

【0197】

また、推定部 326 は、可用帯域幅と過剰遅延時間を送信部 323 に出力する。すると、送信部 323 は、可用帯域幅と過剰遅延時間をサーバ 310 に通知するためのデータ（上記の第 2 の種類のデータ）をサーバ 310 に送信する。

【0198】

なお、推定部 326 は、図 5 の CPU 201 によって実現されてもよい。また、可用帯域幅データ 329 は、RAM 203 に記憶されることが好ましいが、記憶装置 207 に記憶されてもよい。

【0199】

続いて、図 7 を参照して、図 6 中の各種データの詳細について説明する。図 7 には、参照の便宜のため、後述の数式で用いる記号も併記してある。

なお、以下では説明の便宜上、サーバ 310 における時刻は、サーバ 310 が起動してから経過した時間の長さにより示されるものとする。同様に、クライアント 320 における時刻は、クライアント 320 が起動してから経過した時間の長さにより示されるものとする。そのため、図 7 では、時刻が ms (millisecond) 単位で示されている。

【0200】

サーバ 310 における時刻は、例えば、サーバ 310 を実現するコンピュータ 240 の CPU 241 に内蔵されたタイマにより、カウントされてもよい。同様に、クライアント 320 における時刻は、例えば、クライアント 320 を実現するコンピュータ 200 の CPU 201 に内蔵されたタイマにより、カウントされてもよい。

【0201】

さて、サーバ 310 が保持する送信制御データ 316 は、具体的には、可用帯域幅と、過剰遅延解消時刻と、送信フレームレートと、最終送信時刻を含む。図 7 の例では、可用帯域幅が 3500 kbps (kilobit per second)、過剰遅延解消時刻が 101000 ms、送信フレームレートが 30 fps、最終送信時刻が 100010 ms である。

【0202】

送信制御データ 316 中の可用帯域幅は、クライアント 320 から通知された推定値を示す。可用帯域幅の初期値は、例えば 0 kbps などの無効な値である。

【0203】

また、過剰遅延解消時刻は、「過剰遅延を解消するためには、サーバ 310 がいつまで画面情報の送信を控えるのが好ましいか」という目安を示す時刻である。後述するとおり、過剰遅延解消時刻は、クライアント 320 から通知される過剰遅延時間に基づいて算出される。過剰遅延解消時刻の初期値は、例えば 0 ms などの無効な値である。

【0204】

送信フレームレートは、画面情報を送信する頻度の、現在の値を示す。送信フレームレートの初期値は、例えば 30 fps などの所定の値である。送信フレームレートの初期値は、実施形態に応じた適宜の値でよい。

10

20

30

40

50

【0205】

最終送信時刻は、サーバ310が最後にクライアント320に画面情報を送信した時刻である。最終送信時刻の初期値は、例えば0msなどの無効な値である。

【0206】

さて、クライアント320が保持する受信履歴データ327は、1つ以上のエントリを含み得る。各エントリは、受信時刻と、当該受信時刻に受信部324が受信したデータのサイズとのペアである。図7の例は、受信部324が、1200msという時刻に1000バイトのデータを受信し、1240msという時刻に1500バイトのデータを受信し、1270msという時刻に1200バイトのデータを受信したことを示す。

【0207】

受信履歴データ327は、初期状態ではエントリを1つも持たない。また、現在時刻から所定の範囲よりも受信時刻が古くなったエントリは、受信部324により削除される。以下では説明の便宜上、上記「所定の範囲」を、「現在から6秒間以内」という範囲だとするが、所定の範囲は実施形態に応じて適宜決められてよい。

【0208】

また、クライアント320が保持する時刻差データ328は、クライアント320の時計が示す時刻と、サーバ310の時計が示す時刻との時刻差を示す。第2～第4実施形態では、具体的には、時刻差の近似値が使われる。図7の例では、時刻差は-98740msである。

【0209】

ところで、クライアント320が保持する図6の可用帯域幅データ329は、第2～第3実施形態では、具体的には図7の可用帯域幅データ329aのように、推定部326が推定した可用帯域幅の推定値を含むデータである。図7の例では、可用帯域幅データ329aは、3500kbpsという推定値を含む。

【0210】

他方、後述の第4実施形態では、図6の可用帯域幅データ329は、具体的には図7の可用帯域幅データ329bのように、推定部326が推定した可用帯域幅の推定値だけでなく、さらに、前回通知帯域幅を含む。図7の例では、可用帯域幅は3500bpsであり、前回通知帯域幅は3850bpsである。前回通知帯域幅の詳細については、第4実施形態とともに後述する。

【0211】

さて、図8～9は、第2実施形態のクライアント320の動作フローチャートである。クライアント320は、電源が入れられて起動すると、図8～9の処理を開始する。

【0212】

ステップS201で、クライアント320は、サーバ310に接続要求を送信してサーバ310との間の接続を確立し、リモートデスクトップ初期化処理を行う。リモートデスクトップ初期化処理は、サーバ310からクライアント320がリモートデスクトップサービスを受けるための初期化処理である。例えば、ステップS201でクライアント320は、表示部322の解像度や、使用するインプットメソッドなどの情報をサーバ310に通知してもよい。

【0213】

次に、ステップS202でクライアント320は、イベントの発生を待つ。

そして、入力部321を介してユーザから入力与えられるという入力イベントが発生すると、処理はステップS203に移行する。入力イベントの具体例は、例えば、キーボードのキーの押鍵、マウスポインタの移動、ドラッグ、クリック（またはタップ）、ダブルクリック（またはダブルタップ）、フリック、ピンチイン、ピンチアウト、などの操作である。

【0214】

他方、受信部324がサーバ310から何らかのデータを受信するというデータ受信イベントが発生すると、処理はステップS204に移行する。

【0215】

ステップS203で送信部323は、入力部321を介して与えられた入力の内容を、サーバ310に通知する。そして、処理はステップS202に戻る。

【0216】

ステップS204で受信部324は、受信時刻とデータサイズの組を受信履歴データ327に追加する。つまり、受信部324は、現在時刻を取得し、サーバ310から受信したデータのサイズを調べ、現在時刻と調べたサイズを組にした新たなエントリを受信履歴データ327に追加する。

【0217】

なお、サーバ310がクライアント320に送信するデータの形式は実施形態に応じて適宜決められていてよい。例えば、TCP/IP (Transmission Control Protocol/Internet Protocol) 上でリモートデスクトップサービスが提供される場合、サーバ310がクライアント320に送信するデータは、以下の(14a)～(14d)のフィールドを含むパケットであってもよい。

【0218】

(14a) IPヘッダ。

(14b) TCPヘッダ。

(14c) リモートデスクトップサービス用のアプリケーション層のヘッダ。

(14d) リモートデスクトップサービス用のアプリケーション層のペイロード。

【0219】

例えば(14a)～(14d)のフィールドを含むパケットを受信部324が受信した場合、受信履歴データ327に記録されるデータサイズは、(14a)～(14d)のフィールドを含むパケット全体の長さである。

【0220】

次に、ステップS205で受信部324は、受信したデータの種別を判定する。例えば、(14c)のアプリケーション層のヘッダが、データの種別を示すフィールドを含んでもよい。その場合、受信部324は、アプリケーション層のヘッダを参照することにより、データの種別を判定することができる。以下では説明の便宜上、サーバ310がクライアント320に送信するデータには、少なくとも、「時刻合わせコマンド」と「描画コマンド」の2種類があるものとする。

【0221】

時刻合わせコマンドは、クライアント320の時計が示す時刻と、サーバ310の時計が示す時刻との時刻差を算出するためのデータである。具体的には、時刻合わせコマンドは、サーバ310が時刻合わせコマンドを送信した時刻を示すタイムスタンプを含む。

【0222】

なお、(14c)のアプリケーション層のヘッダの形式は、実施形態に応じて様々であってもよい。(14c)のヘッダの形式に応じて、タイムスタンプは、(14c)のヘッダに含まれていてもよいし、(14d)のペイロードに含まれていてもよい。

【0223】

描画コマンドは、サーバ310がクライアント320に画像を画面上に表示させるための画面情報を含むデータである。詳しくは後述するように、描画コマンドのペイロードは、画面の一部または全部を静止画像形式で圧縮したデータ、画面の一部または全部を動画形式で圧縮したデータ、または両者の組み合わせを含む。換言すれば、描画コマンドは、第1実施形態に関して説明した第3の形式の画面情報を含む。

【0224】

例えば、静止画像用の形式としては、JPEG (Joint Photographic Experts Group) 形式が利用可能であり、動画用の形式としては、MPEG形式が利用可能である。もちろん、実施形態に応じて、他の画像圧縮形式が使われてもよい。

【0225】

また、描画コマンドも、サーバ310が描画コマンドを送信した時刻を示すタイムスタ

10

20

30

40

50

ンプを含む。アプリケーション層のヘッダの形式に応じて、タイムスタンプは、ヘッダに含まれていてもよいし、ペイロードに含まれていてもよい。

【0226】

受信したデータが時刻合わせコマンドの場合、処理はステップS207に移行する。受信したデータが描画コマンドの場合、処理はステップS210に移行する。受信したデータが他の種類のデータである場合、処理はステップS206に移行する。

【0227】

ステップS206でクライアント320は、受信したデータの種類に応じた適宜の処理を行う。そして、図8～9の処理はステップS202に戻る。

【0228】

ステップS207で受信部324は、受信した時刻合わせコマンドから、時刻合わせコマンドの送信時刻を取得する。取得される送信時刻は、サーバ310での時計で表された時刻である。以下では説明の便宜上、ステップS207で取得される送信時刻を「T_{sent}」と表記することがある。

【0229】

次に、ステップS208で受信部324は、現在時刻を取得し、現在時刻とステップS207で取得した送信時刻の差を算出する。そして、受信部324は、算出した時刻差を時刻差データ328として記憶する。

【0230】

ステップS208で取得される現在時刻は、クライアント320の時計で表された時刻である。以下では説明の便宜上、ステップS208で取得される現在時刻を「T_{now_c}」と表記し、ステップS208で算出される時刻差を「T_{diff}」と表記する。すなわち、式(1)の時刻差T_{diff}が時刻差データ328として記憶される。

$$T_{diff} = T_{now_c} - T_{sent} \quad (1)$$

【0231】

式(1)から明らかなように、第2実施形態における時刻差T_{diff}は、クライアント320の時計で表される時刻とサーバ310の時計で表される時刻との差そのものを厳密に示す値ではなく、当該時刻差の近似値である。時刻合わせコマンドのような短いデータに関しては、サーバ310が当該データを送信し始めてから、クライアント320が当該データを受信し終わるまでの転送時間も短い。よって、短い転送時間を無視することによって、クライアント320の時計で表される時刻とサーバ310の時計で表される時刻との差は、式(1)のように近似される。

【0232】

なお、参照の便宜上、図7には時刻がミリ秒単位で示されているが、式(1)および後述の各数式における時刻と時間の単位は秒であるものとする。

【0233】

以上のようにして受信部324が時刻差データ328を更新した後、処理はステップS202に戻る。

【0234】

さて、ステップS209では、描画部325が、受信部324により受信された描画コマンドにしたがって、画面上の更新された領域を描画する。

【0235】

具体的には、描画コマンドのペイロードには、画面の中で更新された領域を描画部325に描画させるためのデータが含まれる。例えば、表示部322の画面は、図2に例示するように、横方向にH個に分割され、縦方向にV個に分割されてもよい(2 ≤ Hかつ2 ≤ V)。この場合、画面は合計でH × V個のブロックを含む。これらのH × V個のブロックのすべてに変化が生じる場合もあるし、どのブロックにも変化が生じない場合もあるし、1個以上のあるブロックには変化があり残りのブロックには変化がない場合もある。

【0236】

少なくとも1個のブロックに変化がある場合、描画コマンドのペイロードには、変化し

10

20

30

40

50

たブロックを静止画像形式で圧縮したデータ、変化したブロックを含む矩形の領域を動画画像形式で圧縮したデータ、またはその両者が含まれる。よって、描画部 3 2 5 は、描画コマンドのペイロードに含まれるデータをデコードし、更新された各領域を画面上に描画する。

【0 2 3 7】

例えば、図 2 の例では、描画コマンドのペイロードには、領域 1 1 1 を動画画像形式で圧縮したデータと、領域 1 1 2 を静止画像形式で圧縮したデータと、領域 1 1 3 を静止画像形式で圧縮したデータが含まれる。よって、描画部 3 2 5 は、描画コマンドのペイロードに含まれるデータをデコードし、領域 1 1 1 ～ 1 1 3 を更新する。例えば、描画部 3 2 5 は、V R A M (Video Random Access Memory) 上に記憶されている、領域 1 1 1 ～ 1 1 3 中の各画素のデータを、デコード結果に応じて更新する。それにより、描画部 3 2 5 は、画面上の領域 1 1 1 ～ 1 1 3 を更新する。

10

【0 2 3 8】

次のステップ S 2 1 0 では、推定部 3 2 6 が、受信履歴データ 3 2 7 を用いて、最近の 6 秒間以内の期間における平均受信ビットレートを算出する。図 7 に関して説明したとおり、第 2 実施形態では、受信時刻から 6 秒が経過した後は、受信履歴データ 3 2 7 からエントリが削除される。よって、受信履歴データ 3 2 7 のどのエントリの受信時刻も、現在から 6 秒以内である。

【0 2 3 9】

説明の便宜上、受信履歴データ 3 2 7 のエントリの個数を R とする ($1 \leq R$)。また、 $1 \leq j \leq R$ なる各 j について、受信履歴データ 3 2 7 の j 番目のエントリにおける受信時刻とデータサイズをそれぞれ「 T_rcv_j 」と「 $Size_rcv_j$ 」と表記する。なお、1 番目のエントリの受信時刻が最も古く、R 番目のエントリの受信時刻が最も新しいものとする。

20

【0 2 4 0】

ステップ S 2 1 0 で推定部 3 2 6 は、より具体的には、例えば式 (2) にしたがって最近の 6 秒間以内の期間における平均受信ビットレート (以下では「 BPS_avr 」と表記する) を算出してもよい。

【数 1】

$$BPS_avr = \frac{\sum_{j=1}^R 8 \cdot Size_rcv_j}{T_now_c - T_rcv_1} \quad (2)$$

30

【0 2 4 1】

なお、式 (2) における 8 の乗算は、バイトからビットへの換算を示す。また、式 (2) における「 T_now_c 」は、推定部 3 2 6 がステップ S 2 1 0 で取得する現在時刻を示す。なお、参照の便宜上、図 7 には帯域幅が k b p s 単位で示されているが、式 (2) および後述の各数式における、帯域幅および受信ビットレートの単位は、b p s (bit per second) であるものとする。

40

【0 2 4 2】

実施形態によっては、上記の「最近 6 秒間」という所定の期間の長さに合わせて、式 (2) の分母は、「6」という定数に置き換えられてもよい。もちろん、所定の期間の長さ自体も、実施形態に応じて適宜変更されてよい。

【0 2 4 3】

続いて、ステップ S 2 1 1 で推定部 3 2 6 は、可用帯域幅データ 3 2 9 を参照し、可用帯域幅が初期値 (例えば 0 k b p s) であるか否かを判断する。可用帯域幅が初期値の場合、処理はステップ S 2 1 2 に移行し、可用帯域幅が初期値ではない場合、処理はステップ S 2 1 3 に移行する。

【0 2 4 4】

50

ステップS 2 1 2で推定部3 2 6は、ステップS 2 1 0で算出した平均受信ビットレートB P S _ a v rを可用帯域幅として記憶する。すなわち、推定部3 2 6は、可用帯域幅データ3 2 9を平均受信ビットレートB P S _ a v rで上書きする。

【0 2 4 5】

なお、可用帯域幅データ3 2 9は、サーバ3 1 0とクライアント3 2 0の間のエンド・ツー・エンドの可用帯域幅の推定値を示す。そして、実際の可用帯域幅は時間とともに変化し得るが、ある期間における可用帯域幅の実際の値の平均値は、当該期間における平均受信ビットレート以上であると推定される。よって、ステップS 2 1 2では、平均受信ビットレートB P S _ a v rが可用帯域幅の推定値として使われる（すなわち、平均受信ビットレートB P S _ a v rが可用帯域幅データ3 2 9として記憶される）。

10

【0 2 4 6】

以下では説明の便宜上、可用帯域幅（より正確には可用帯域幅の推定値）を「E s t B W」と表記することがある。具体的には、表記の簡単化のために単に「可用帯域幅E s t B W」と表記する場合もあるし、可用帯域幅の実際の値と推定値との区別を明示するために「可用帯域幅の推定値E s t B W」のように表記する場合もある。

【0 2 4 7】

ステップS 2 1 2の処理は、式（3）のように表される。ある観点から見れば、ステップS 2 1 2の処理は、無効な値に設定されている可用帯域幅E s t B Wを、式（3）により有効な値に初期化する処理である。

$$\text{EstBW} = \text{BPS_avr} \quad (3)$$

20

【0 2 4 8】

以上のようにして可用帯域幅が更新された後、処理はステップS 2 1 3に移行する。

ステップS 2 1 3で推定部3 2 6は、受信部3 2 4が今回受信したデータ（つまり描画コマンド）のサイズと可用帯域幅から、今回受信したデータの転送にかかった時間の推定値を算出する。以下では説明の便宜上、ステップS 2 1 3で算出される時間を「推定転送時間」といい、「E s t T _ t r a n s」と表記する。

【0 2 4 9】

具体的には、推定部3 2 6は、受信履歴データ3 2 7の最新のエントリに記憶されているデータサイズS i z e _ r c v _ Rと、可用帯域幅データ3 2 9として記憶されている可用帯域幅E s t _ B Wを用いて、式（4）の計算を行う。

30

$$\text{EstT_trans} = 8 \times \text{Size_rcv}_R / \text{EstBW} \quad (4)$$

【0 2 5 0】

そして、次に図9のステップS 2 1 4で推定部3 2 6は、受信部3 2 4により受信された描画コマンドから、描画コマンドがサーバ3 1 0から送信された時刻を示す時刻情報（すなわち上記のタイムスタンプ）を取得する。

【0 2 5 1】

以下ではステップS 2 1 4で取得される時刻情報が示す送信時刻を、説明の便宜上、「T _ s e n t」とも表記する。送信時刻T _ s e n tは、サーバ3 1 0の時計で表される時刻である。推定部3 2 6は以上のようにして送信時刻T _ s e n tを取得する。

【0 2 5 2】

40

次のステップS 2 1 5で推定部3 2 6は、式（5）に示すとおり、送信時刻T _ s e n tに時刻差T _ d i f fを足し、推定送信時刻（以下では「E s t T _ s e n t」と表記する）を算出する。

$$\text{EstT_sent} = \text{T_sent} + \text{T_diff} \quad (5)$$

【0 2 5 3】

式（1）から分かるように、サーバ3 1 0の時計で表された時刻に時刻差T _ d i f fを足すことで、サーバ3 1 0の時計で表された時刻は、クライアント3 2 0の時計で表された時刻に換算される。また、時刻差T _ d i f fは上記のとおり近似値である。よって、式（5）により算出される推定送信時刻E s t T _ s e n tは、描画コマンドがサーバ3 1 0から送信された時刻T _ s e n tをクライアント3 2 0の時計で示した時刻の推定

50

値を示す。

【0254】

そして、次にステップS216で推定部326は、式(6)に示すとおり、受信履歴データ327の最新のエントリに記憶されている受信時刻 T_{rcv_R} から、推定送信時刻 $EstT_{sent}$ と推定転送時間 $EstT_{trans}$ を引く。式(6)の減算により、推定部326は、過剰遅延時間を算出する。なお、式(6)に示すように、以下では便宜上、過剰遅延時間を「 $ExtraDelay$ 」とも表記する。

$$ExtraDelay = T_{rcv_R} - EstT_{sent} - EstT_{trans} \quad (6)$$

【0255】

過剰遅延時間 $ExtraDelay$ は、可用帯域幅 $EstBW$ から推定される受信時刻($EstT_{sent} + EstT_{trans}$)よりも実際の受信時刻 T_{rcv_R} がどれくらい遅延しているかを示す。

10

【0256】

可用帯域幅データ329に基づいて推定される受信時刻よりも早くに描画コマンドが受信された場合は、過剰遅延時間 $ExtraDelay$ は負である。逆に、可用帯域幅データ329に基づいて推定される受信時刻よりも遅くに描画コマンドが受信された場合は、過剰遅延時間 $ExtraDelay$ は正である。

【0257】

続いて、ステップS217で推定部326は、過剰遅延時間 $ExtraDelay$ が300msより長いのか、0msより長く300ms以下なのか、それとも0ms以下なのかを判断する。

20

【0258】

なお、ステップS217における「300ms」という第1の閾値は、実施形態に応じて、適宜、他の正の値に置き換えられてよい。また、ステップS217における「0ms」という第2の閾値も、実施形態に応じて、適宜、他の値に置き換えられてよい。

【0259】

第2の閾値は、ゼロか負の値であることが望ましいが、ゼロに近い小さな正の値が第2の閾値として使われてもよい。ただし、いずれにせよ、第2の閾値は、第1の閾値より小さい。

【0260】

30

なお、2つの閾値は、ユーザ定義可能な変数であってもよいし、推定部326が可用帯域幅 $EstBW$ に応じて変更する可変の閾値であってもよい。しかし、以下では説明の簡単化のため、固定的な2つの閾値が使われるものとする。

【0261】

これらの2つの閾値の意味を説明すれば、以下のとおりである。第1と第2の閾値は、以下に説明する意味をよく反映するように、例えば予備実験などに基づいて、適切に定められることが好ましい。

【0262】

実際の可用帯域幅は、時間とともに変動し得る。例えば、サーバ310とクライアント320以外の他のコンピュータによるネットワーク通信の量に応じて、実際の可用帯域幅は変動し得る。また、サーバ310とクライアント320の間のネットワークの少なくとも一部が無線ネットワークである場合、遮蔽物や干渉源など、環境中のいくつかの要因によっても、実際の可用帯域幅は変動し得る。

40

【0263】

実際の可用帯域幅の変動に応じて、描画コマンドの受信時刻が影響を受けるので、過剰遅延時間 $ExtraDelay$ も実際の可用帯域幅の変動に応じて変動し得る。しかし、偶発的に起こり得る小さな変動は無視しても差し支えない。

【0264】

式(6)から分かるように、可用帯域幅データ329として記憶されている推定値が実際の可用帯域幅に近ければ、過剰遅延時間 $ExtraDelay$ の絶対値も比較的小さい

50

。よって、過剰遅延時間 *Extra Delay* の絶対値が比較的小さい場合、過剰遅延時間 *Extra Delay* は、偶発的に起こり得る小さな変動に起因すると見なして差し支えない。つまり、この場合、可用帯域幅データ 329 として記憶されている推定値は、更新されなくてもよい。

【0265】

しかし、もし過剰遅延時間 *Extra Delay* が第2の閾値以下ならば、可用帯域幅データ 329 として記憶されている推定値は、更新されることが好ましい（具体的には、大きくされることが好ましい）。なぜなら、過剰遅延時間 *Extra Delay* が第2の閾値以下ならば、「可用帯域幅データ 329 として記憶されている推定値が過小である」という可能性が残っているからである。つまり、実際の可用帯域幅の一部が活用されずに遊んでいる可能性がある。

10

【0266】

すなわち、過剰遅延時間 *Extra Delay* が第2の閾値以下ならば、実際の可用帯域幅によっては、サーバ 310 からクライアント 320 に描画コマンドが送信される頻度を上昇させる余地があり得る。換言すれば、クライアント 320 における表示フレームレートを上げて、画面表示をより滑らかにし、ひいてはユーザビリティをさらに向上させることが可能な場合もあり得る。

【0267】

よって、過剰遅延時間 *Extra Delay* が第2の閾値以下ならば、以上のような可能性が実際に成り立つか否かを試すために、推定部 326 は可用帯域幅の推定値を増加させてもよい。クライアント 320 は、より大きな可用帯域幅の推定値をサーバ 310 に通知することで、間接的に、サーバ 310 からクライアント 320 に描画コマンドが送信される頻度を上昇させることができる。

20

【0268】

つまり、第2の閾値は、可用帯域幅データ 329 を更新しないで推定値を維持する場合と、上記のような可能性が成り立つか否かを積極的に試す場合とを分かつための閾値である。

【0269】

逆に、もし過剰遅延時間 *Extra Delay* が非常に長ければ（具体的には、第1の閾値より長ければ）、可用帯域幅データ 329 として記憶されている推定値が過大だと考えられる。また、この場合、「偶発的に起こり得る小さな変動に起因して、過剰遅延時間 *Extra Delay* が、偶然、正になった」という蓋然性は低い。むしろ、この場合、「描画コマンドの送信が繰り返されるうちに遅延が次第に累積してゆく過程（例えば図3のタイミングチャート 123 のような過程）に起因して、過剰遅延時間 *Extra Delay* が第1の閾値を超えた」という蓋然性が高い。

30

【0270】

よって、過剰遅延時間 *Extra Delay* が第1の閾値より長ければ、可用帯域幅データ 329 として記憶されている推定値を減少させる（すなわち推定値を実際の可用帯域幅に近づける）ことが適切である。つまり、第1の閾値は、偶然起こり得る小さな変動と、可用帯域幅の過大な推定に起因する遅延の累積とを見分けるための閾値である。

40

【0271】

さて、ここでステップ S217 の説明に戻る。過剰遅延時間 *Extra Delay* が 300ms より長ければ、可用帯域幅 *Est BW* を減少させるために、処理はステップ S218 に移行する。過剰遅延時間 *Extra Delay* が 0ms より長く 300ms 以下であれば、推定部 326 は、可用帯域幅 *Est BW* を変えずにそのまま維持し、処理はステップ S220 に移行する。過剰遅延時間 *Extra Delay* が 0ms 以下ならば、可用帯域幅 *Est BW* を増加させるために、処理はステップ S221 に移行する。

【0272】

ステップ S218 で推定部 326 は、過大な推定値を小さくするために、可用帯域幅 *Est BW* の現在の値と、ステップ S210 で算出した平均受信ビットレート *BPS_{av}*

50

rとの平均値を、新たに可用帯域幅 $EstBW$ として記憶する。すなわち、推定部 326 は、式 (7) にしたがって可用帯域幅データ 329 を更新する。式 (7) の右辺の「 $EstBW$ 」は可用帯域幅の現在の値を示し、左辺の「 $EstBW$ 」は更新後の可用帯域幅の値を示す。

$$EstBW = (EstBW + BPS_{avr}) / 2 \quad (7)$$

【0273】

なお、実施形態によっては、推定部 326 は、式 (7) 以外の式にしたがって可用帯域幅データ 329 を更新してもよい。例えば、推定部 326 は、 $0 < C_1 \leq 1$ なる適宜の定数 C_1 を用いて、式 (8) にしたがって可用帯域幅データ 329 を更新してもよい。

$$EstBW = (1 - C_1) \times EstBW + C_1 \times BPS_{avr} \quad (8)$$

10

【0274】

式 (7) は、式 (8) における定数 C_1 が $1/2$ の場合の例である。定数 C_1 が小さいほど、可用帯域幅 $EstBW$ の変化は緩やかになる。

【0275】

続いて、ステップ S219 で推定部 326 は、ステップ S218 で更新した可用帯域幅 $EstBW$ を用いて、過剰遅延時間 $ExtraDelay$ を再計算する。

【0276】

具体的には、推定部 326 は、更新後の可用帯域幅 $EstBW$ を用いて、式 (4) により推定転送時間 $EstT_{trans}$ を再計算する。そして、推定部 326 は、再計算した推定転送時間 $EstT_{trans}$ を用いて、式 (6) により過剰遅延時間 $ExtraDelay$ を再計算する。

20

【0277】

過剰遅延時間 $ExtraDelay$ の再計算の後、処理はステップ S220 に移行する。なお、ステップ S216 で算出される過剰遅延時間 $ExtraDelay$ と、ステップ S219 で再計算される過剰遅延時間 $ExtraDelay$ の差は、それほど顕著でない場合も多い。よって、処理の簡素化のため、実施形態によっては、ステップ S219 が省略されてもよい。

【0278】

ステップ S220 では、推定部 326 が可用帯域幅 $EstBW$ と過剰遅延時間 $ExtraDelay$ を送信部 323 に出力する。すると、送信部 323 は、可用帯域幅 $EstBW$ と過剰遅延時間 $ExtraDelay$ を含む通知をサーバ 310 に送信する。そして、処理は図 8 のステップ S202 に戻る。

30

【0279】

ステップ S221 で推定部 326 は、可用帯域幅 $EstBW$ の値を増加させる。具体的には、推定部 326 は、比較的小さな正の適宜の定数 C_2 を用いて、式 (9) により、新たな可用帯域幅 $EstBW$ の値を算出し、可用帯域幅データ 329 を更新する。式 (9) の右辺の記号「 $EstBW$ 」は、可用帯域幅の現在の推定値を示し、左辺の記号「 $EstBW$ 」は、可用帯域幅の新たな推定値を示す。

$$EstBW = EstBW \times (1 + C_2) \quad (9)$$

【0280】

40

図 9 には、 $C_2 = 0.1$ の場合が例示されている。上記のとおり、ステップ S221 が実行されるのは過剰遅延時間 $ExtraDelay$ が $0ms$ 以下の場合であり、この場合、フレームレートを上げる余地があり得る。よって、ステップ S221 では、フレームレートを上げる余地が実際にあるか否かを試すため、推定部 326 が可用帯域幅 $EstBW$ の値を増加させる。

【0281】

なお、推定部 326 は、式 (9) のように 1 より大きい所定の値を可用帯域幅の現在の推定値に掛ける代わりに、所定の正の値を可用帯域幅の現在の推定値に足すことで、可用帯域幅の新たな推定値を算出してもよい。

【0282】

50

次に、ステップS 2 2 2で推定部3 2 6は、ステップS 2 2 1で更新した可用帯域幅E s t B Wを用いて、過剰遅延時間E x t r a D e l a yを再計算する。

【0 2 8 3】

具体的には、推定部3 2 6は、更新後の可用帯域幅E s t B Wを用いて、式(4)により推定転送時間E s t T _ t r a n sを再計算する。そして、推定部3 2 6は、再計算した推定転送時間E s t T _ t r a n sを用いて、式(6)により過剰遅延時間E x t r a D e l a yを再計算する。

【0 2 8 4】

過剰遅延時間E x t r a D e l a yの再計算の後、処理はステップS 2 2 0に移行する。なお、ステップS 2 1 6で算出される過剰遅延時間E x t r a D e l a yと、ステップS 2 2 2で再計算される過剰遅延時間E x t r a D e l a yの差は、それほど顕著でない場合も多い。よって、処理の簡素化のため、実施形態によっては、ステップS 2 1 9が省略されてもよい。

10

【0 2 8 5】

さて、図1 0～1 1は、第2～第4実施形態に共通のサーバ3 1 0の動作フローチャートである。サーバ3 1 0は、クライアント3 2 0からの接続要求を受信すると、図1 0～1 1の処理を開始する。

【0 2 8 6】

ステップS 3 0 1でサーバ3 1 0は、クライアント3 2 0との間の接続を確立し、リモートデスクトップ初期化処理を行う。例えば、サーバ3 1 0は、クライアント3 2 0から表示部3 2 2の解像度や、使用するインプットメソッドなどの情報を受信してもよい。そして、サーバ3 1 0は、クライアント3 2 0用に1つ以上のプログラムを実行するための実行環境(runtime environment)をサーバ3 1 0内に作成し、クライアント3 2 0から受信した情報に応じて、実行環境の環境設定を行ってもよい。

20

【0 2 8 7】

サーバ3 1 0内に生成される上記実行環境は、実施形態に応じて、例えば、サーバ3 1 0上のプロセスであってもよいし、サーバ3 1 0(すなわち物理マシン)上の仮想マシンであってもよい。また、クライアント3 2 0用にサーバ3 1 0が実行する上記1つ以上のプログラムは、具体的には、OSを含み、さらにアプリケーションプログラムを含んでもよい。

30

【0 2 8 8】

次に、ステップS 3 0 2で付加部3 1 5は、現在時刻を取得し、現在時刻を示す時刻情報(すなわちタイムスタンプ)を含む時刻合わせコマンドを生成する。そして、送信部3 1 3が時刻合わせコマンドをクライアント3 2 0に送信する。

【0 2 8 9】

次に、ステップS 3 0 3でサーバ3 1 0は、イベントの発生を待つ。

そして、受信部3 1 1がクライアント3 2 0から入力通知を受信したというイベントが発生すると、処理はステップS 3 0 4に移行する。入力通知は、図8のステップS 2 0 3でクライアント3 2 0から送信される通知である。

【0 2 9 0】

あるいは、受信部3 1 1がクライアント3 2 0から可用帯域幅と過剰遅延時間の通知を受信したというイベントが発生すると、処理はステップS 3 0 5に移行する。可用帯域幅と過剰遅延時間の通知は、図9のステップS 2 2 0でクライアント3 2 0から送信される通知である。

40

【0 2 9 1】

ところで、サーバ3 1 0の生成部3 1 2は、所定の間隔で発火する(expire)画面取得タイマを有する。画面取得タイマは、具体的には、図5のC P U 2 4 1の内蔵タイマにより実現されてもよい。

【0 2 9 2】

第2～第4実施形態では、送信フレームレートの最大値が予め決められており、画面取

50

得タイマが発火する間隔は、送信フレームレートの最大値に応じて予め設定される。以下では説明の便宜上、送信フレームレートの最大値が30fpsであるものとする。したがって、画面取得タイマの発火間隔は、1/30秒である。

【0293】

画面取得タイマの発火というイベントが発生すると、処理はステップS303からステップS308へと移行する。

【0294】

さて、入力通知が受信された場合、ステップS304で生成部312は、受信部311により受信された入力通知の内容に応じた入力割り込みを発生させる。そして、処理はステップS303に戻る。

10

【0295】

なお、説明の簡略化のため図10～11では省略しているが、ステップS301でのリモートデスクトップ初期化処理の完了後、生成部312は、上記実行環境上でOSの実行を開始する。生成部312がステップS304で発生させる入力割り込みは、アプリケーションプログラムを起動するトリガになる場合もあり得る（例えば、所定のアイコン上でのダブルクリック操作が通知された場合など）。この場合、生成部312は、発生させた入力割り込みに応じて、新たにアプリケーションプログラムの実行を開始する。

【0296】

あるいは、生成部312がステップS304で発生させる入力割り込みは、起動済みのプログラム（例えば、OSでもよいし、アプリケーションプログラムでもよい）に対する入力を表す場合もあり得る。この場合、生成部312は、起動済みのプログラムを実行する過程において、入力割り込みに応じた処理を行う。

20

【0297】

例えば、テキストエディタが起動済みで、かつ、入力割り込みがキー入力に対応する場合があります。この場合、生成部312は、テキストエディタのプログラムを実行する過程において、入力割り込みに応じて、カーソル位置に文字を表示させるための処理を行ってもよい。

【0298】

また、CADアプリケーションが起動済みで、かつ、入力割り込みが、CADアプリケーション内の「回転」などのコマンドの呼び出しのための入力（例えばメニューバーからのメニュー選択操作）に対応する場合があります。この場合、生成部312は、CADアプリケーションのプログラムを実行する過程において、入力割り込みに応じて、CADアプリケーション内のコマンドを呼び出してよい。

30

【0299】

また、入力割り込みが、起動済みのアプリケーションプログラムを終了させるための入力（例えば「終了」メニューを選択する操作）に対応する場合があります。この場合、生成部312は、入力割り込みに応じて、当該アプリケーションプログラムの実行を終了する。

【0300】

例えば上記に例示したようにして、生成部312は、クライアント320からの入力通知の内容に応じて、入力割り込みを発生させ、入力割り込みの内容に応じて、プログラムを実行する。

40

【0301】

ここで、図10～11の説明に戻る。ステップS305で調整部314は、受信部311により受信された可用帯域幅と過剰遅延時間の通知から、可用帯域幅を読み取る。そして、調整部314は、通知された可用帯域幅を記憶する。つまり、調整部314は、送信制御データ316中の可用帯域幅を、通知された値に書き換える。

【0302】

次に、ステップS306で調整部314は、可用帯域幅と過剰遅延時間の通知から過剰遅延時間を読み取り、通知された過剰遅延時間が正か否かを判断する。過剰遅延時間が正

50

ならば、処理はステップS307に移行する。逆に、過剰遅延時間が0以下ならば、処理はステップS303に戻る。

【0303】

ステップS307で調整部314は、現在時刻を取得し、現在時刻に過剰遅延時間を足して、過剰遅延解消時刻を算出する。そして、調整部314は、送信制御データ316中の過剰遅延解消時刻を、算出した値に書き換える。

【0304】

ステップS307で取得される現在時刻は、サーバ310の時計で表された時刻である。以下では説明の便宜上、ステップS307で取得される現在時刻を「T_now_s」と表記し、ステップS307で算出される過剰遅延解消時刻を「T_solved」と表記する。ステップS307での過剰遅延解消時刻T_solvedの算出は、式(10)のとおりである。

$$T_{solved} = T_{now_s} + \text{ExtraDelay} \quad (10)$$

過剰遅延解消時刻T_solvedの算出後、処理はステップS303に戻る。

【0305】

さて、画面取得タイマが発火した場合は、ステップS308で生成部312が画面全体の画像（すなわちフレーム画像）を取得する。

【0306】

上記のとおり生成部312は1つ以上のプログラムの実行を続けており、プログラムの実行に応じて、クライアント320の表示部322の画面に表示するための画像を生成する。例えば、画像のデータは、サーバ310を実現するコンピュータ240のRAM243内の第1のメモリ領域に、無圧縮のビットマップ形式で記憶されていてもよい。第1のメモリ領域は、具体的には、クライアント320用に割り当てられたフレームバッファ中の領域であり、ある観点から見れば、クライアント320のVRAMをエミュレートするためのメモリ領域である。

【0307】

生成部312は、プログラムを実行する過程において、プログラムの実行の進捗に応じて第1のメモリ領域内の一部または全部のデータを書き換えることで、画像を更新してもよい。生成部312は、ステップS308では、第1のメモリ領域から画像のデータを読み出すことで、画面全体の画像を取得してもよい。

【0308】

そして、次のステップS309で生成部312は、今回ステップS308で取得した画像と、前回画面取得タイマが発火したときに取得した画像との差分を検出する。

【0309】

例えば、RAM243内において第1のメモリ領域とは別の第2のメモリ領域に、前回画面取得タイマが発火したときに生成部312が取得した画像のデータが格納されていてもよい。生成部312は、第1のメモリ領域と第2のメモリ領域の画像同士を比較することで、今回取得した画像と前回の画面の画像との差分を検出してもよい。なお、生成部312は、後述の動画圧縮におけるフレーム間予測や動き補償のために、第1と第2のメモリ領域とは別のメモリ領域に、さらに過去のフレーム画像を保持していてもよい。

【0310】

次に、ステップS310で生成部312は、動画像用の圧縮アルゴリズムで圧縮する対象の領域（以下、「動画領域」という）を検出する。図2の例では領域111が動画領域である。動画領域を検出する方法は公知なので、詳しい説明は省略するが、例えば、生成部312は以下のようにして動画領域を検出してもよい。

【0311】

図2に例示するように、画面がH×V個のブロックに分割されていてもよい。この場合、生成部312は、H×V個のブロックのそれぞれについて、ステップS309において当該ブロックで差分が検出された回数をカウントし、カウント結果を所定の閾値と比較してもよい。生成部312は、カウント結果が閾値以上のブロックを、高頻度で変更が生じ

10

20

30

40

50

ているブロックだと判断し、カウント結果が閾値未満のブロックを、変更が生じる頻度が低いブロックだと判断してもよい。

【0312】

例えば、生成部312は、各ブロックについて、ステップS308の直近のQ回（ $Q > 1$ ）の実行においてそれぞれ差分が検出されたか否かを記憶してもよい。そして、生成部312は、各ブロックについて、Q回のうち何回差分が検出されたかをカウントして、上記の閾値と比較してもよい。

【0313】

以上のようにして、高頻度で変更が生じていることが検出されるブロックの個数は、0個の場合もあり得るし、1個の場合もあり得るし、複数個の場合もあり得る。

10

【0314】

生成部312は、複数のブロックにおいて高頻度で変更が生じていることを検出した場合は、それら複数のブロックの各々を個別に動画領域と決定してもよい。

【0315】

あるいは、高頻度で変更が生じていることが検出されたいくつかのブロック同士が、隣接または近接している場合もあり得る。この場合、生成部312は、隣接または近接するそれらのブロックを包含する矩形領域（例えば、隣接または近接するそれらのブロックを包含する最小の矩形領域）を、動画領域と決定してもよい。

【0316】

例えば以上のようにして、生成部312は、ステップS310で動画領域を検出する。したがって、場合によっては、1つも動画領域が検出されないこともあり得るし、1つの動画領域が検出されることもあり得るし、離れ離れの複数の動画領域が検出されることもあり得る。

20

【0317】

次に、ステップS311で調整部314は、送信制御データ316を参照して、現在の送信フレームレート（以下では「FPS_cur」と表記する）を取得する。そして、調整部314は、現在の送信フレームレートFPS_curから、式(11)により、送信時間間隔（以下では「Int」と表記する）を算出する。

$$\text{Int} = 1/\text{FPS_cur} \quad (11)$$

【0318】

次に、ステップS312で調整部314は、現在時刻（以下では「T_now_s」と表記する）を取得する。また、調整部314は、送信制御データ316を参照して、最終送信時刻（以下では「T_last」と表記する）を取得する。そして、調整部314は、現在時刻T_now_sと最終送信時刻T_lastの差が、ステップS311で算出した送信時間間隔Intより大きいかな否かを判断する。すなわち、調整部314は、不等式(12)が成り立つかな否かを判断する。

30

$$T_now_s - T_last > \text{Int} \quad (12)$$

【0319】

不等式(12)が成り立たない場合は、まだクライアント320に描画コマンドを送信しなくてよい。よって、処理はステップS303に戻る。逆に、不等式(12)が成り立つ場合、クライアント320に描画コマンドを送信するか否かを決定するために、処理は図11のステップS313に移行する。

40

【0320】

ステップS313で調整部314は、過剰遅延解消時刻T_solve_dを算出済みかな否かを判断する。具体的には、調整部314は、送信制御データ316を参照して、過剰遅延解消時刻T_solve_dとして有効な値（すなわち初期値とは別の値）が記憶されているかな否かを判断する。

【0321】

過剰遅延解消時刻T_solve_dとして有効な値が記憶されていれば、過剰遅延解消時刻T_solve_dは算出済みである。この場合、クライアント320に描画コマンド

50

を送信するか否かは、過剰遅延解消時刻 T_solved による。よって、描画コマンドを送信するか否かを過剰遅延解消時刻 T_solved に応じて決定するために、処理はステップ S 3 1 4 に移行する。

【0322】

逆に、過剰遅延時刻として無効な値が記憶されていれば、過剰遅延時刻はまだ算出されていない。この場合、クライアント 3 2 0 に描画コマンドを送信するか否かは、単に現在時刻 T_now_s と最終送信時刻 T_last の差による。また、不等式 (12) が成り立つことは、ステップ S 3 1 2 において確認済みである。よって、描画コマンドの生成と送信のために、処理はステップ S 3 1 5 に移行する。

【0323】

ステップ S 3 1 4 で調整部 3 1 4 は、現在時刻 T_now_s が過剰遅延解消時刻 T_solved 以降であるか否かを判断する。現在時刻 T_now_s が過剰遅延解消時刻 T_solved よりも前であれば、まだクライアント 3 2 0 に描画コマンドを送信しない方がよいので、処理はステップ S 3 0 3 に戻る。逆に、現在時刻 T_now_s が過剰遅延解消時刻 T_solved 以降であれば、描画コマンドの生成と送信のために、処理はステップ S 3 1 5 に移行する。

【0324】

ところで、以下では説明の便宜上、ステップ S 3 0 9 で生成部 3 1 2 が差分を検出した領域を「差分領域」という。例えば、ステップ S 3 1 0 に関して例示したように、画面が $H \times V$ 個のブロックに分割されている場合、ステップ S 3 0 9 では、 $H \times V$ 個のブロックのうち、0 個、1 個、または複数個のブロックにおいて、差分が検出される。差分が検出された各ブロックが、1 つの差分領域である。

【0325】

ステップ S 3 1 5 で生成部 3 1 2 は、各差分領域について、当該差分領域を動画像用の圧縮アルゴリズムで圧縮するか、それとも静止画像用の圧縮アルゴリズムで圧縮するかを、ステップ S 3 1 0 での動画領域の検出結果に基づいて決定する。具体的には、差分領域が動画領域に含まれる場合、生成部 3 1 2 は、当該差分領域を動画像用の圧縮アルゴリズムで圧縮することに決める。逆に、差分領域が動画領域に含まれない場合、生成部 3 1 2 は、当該差分領域を静止画像用の圧縮アルゴリズムで圧縮することに決める。

【0326】

例えば、図 2 の例では、領域 1 1 2 と 1 1 3 は、差分領域である。しかし、領域 1 1 2 と領域 1 1 3 は、動画領域である領域 1 1 1 に含まれない。よって、生成部 3 1 2 は、領域 1 1 2 と 1 1 3 の各々を静止画像用の圧縮アルゴリズムで圧縮することに決める。

【0327】

そして、次のステップ S 3 1 6 で生成部 3 1 2 は、各動画領域を動画像用の圧縮アルゴリズムで圧縮し、静止画像用の圧縮アルゴリズムで圧縮することに決めた各差分領域を、静止画像用の圧縮アルゴリズムで圧縮する。

【0328】

なお、ステップ S 3 1 0 に関して例示したように、いくつか差分領域を包含する矩形領域が動画領域として決定される場合もある。この場合、差分の検出されていないブロックが動画領域に含まれることもあり得るが、生成部 3 1 2 は矩形の動画領域全体を動画像用の圧縮アルゴリズムで圧縮する。例えば、図 2 の例では、生成部 3 1 2 は、領域 1 1 1 を動画像用の圧縮アルゴリズムで圧縮する。

【0329】

続いて、ステップ S 3 1 7 で調整部 3 1 4 は、送信制御データ 3 1 6 を参照して、可用帯域幅の通知をクライアント 3 2 0 から受信済みか否かを判断する。可用帯域幅として有効な値（すなわち初期値以外の値）が記憶されていれば、可用帯域幅の通知は受信済みなので、処理はステップ S 3 1 8 に移行する。逆に、可用帯域幅として無効な値が記憶されていれば、可用帯域幅の通知はまだ受信されていないので、処理はステップ S 3 2 0 に移行する。

10

20

30

40

50

【0330】

ステップS318で調整部314は、送信制御データ316を参照して、可用帯域幅EstBWの値を取得する。また、調整部314は、生成部312がステップS316で圧縮したデータの合計サイズを認識する。例えば、生成部312が調整部314に圧縮後のデータの合計サイズを通知してもよい。

【0331】

そして、調整部314は、圧縮後のデータの合計サイズから、当該データをペイロードに含む描画コマンドのサイズを算出する。具体的には、描画コマンドの形式に応じて、調整部314は、各種ヘッダの長さなどを圧縮後のデータの合計サイズに足すことで、描画コマンドのサイズを算出する。以下では説明の便宜上、算出されるサイズを「Size_cmd」と表記し、当該サイズはバイト単位で表されるものとする。

10

【0332】

ここで、以下の(15a)と(15b)の2点を仮定した場合に、可用帯域幅EstBWの範囲内で各フレームに対応する描画コマンドの送信を繰り返すことが可能な最大のフレームレートのことを、「送信可能フレームレート」ということにする。

【0333】

(15a) 各フレームに対応する描画コマンドのサイズが一定である。

(15b) 具体的には、当該一定のサイズが、上記のようにして算出されたサイズSize_cmdである。

【0334】

20

調整部314は、可用帯域幅EstBWと描画コマンドのサイズSize_cmdから、送信可能フレームレート（以下、「FPS_feasible」と表記する）を算出する。上記の定義より、送信可能フレームレートFPS_feasibleは、不等式(13)を満たす最大の値である。よって、調整部314は、式(14)にしたがって送信可能フレームレートFPS_feasibleを算出する。

$$8 \times \text{Size_cmd} \times \text{FPS_feasible} \leq \text{EstBW} \quad (13)$$

$$\text{FPS_feasible} = \text{EstBW} / (8 \times \text{Size_cmd}) \quad (14)$$

【0335】

以上のようにしてステップS318で送信可能フレームレートFPS_feasibleが算出された後、処理はステップS319に移行する。すると、ステップS319で調整部314は、送信制御データ316を参照して、現在の送信フレームレートFPS_curを読み取る。そして、調整部314は、送信フレームレートFPS_curと送信可能フレームレートFPS_feasibleに基づいて、新たな送信フレームレートを設定する。例えば、調整部314は以下のようにして新たな送信フレームレートを設定してもよい。

30

【0336】

ステップS308に関して説明したように、第2～第4実施形態では、送信フレームレートの最大値が予め決められている。以下では説明の便宜上、送信フレームレートの最大値を「FPS_max」と表記する。調整部314は、送信フレームレートの現在の値FPS_curと送信可能フレームレートFPS_feasibleの平均値と、送信フレームレートの最大値FPS_maxのうち、小さい方の値を、新たに送信フレームレートとして設定してもよい。

40

【0337】

すなわち、調整部314は、式(15)の右辺にしたがって新たな送信フレームレートを算出し、算出した値を、送信制御データ316中の送信フレームレートとして記憶してもよい。式(15)の左辺の「FPS_cur」という記号は、算出された新たな送信フレームレートを示す。

$$\text{FPS_cur} = \min(\text{FPS_max}, (\text{FPS_cur} + \text{FPS_feasible}) / 2) \quad (15)$$

【0338】

あるいは、調整部314は、 $0 < C_3 \leq 1$ なる適宜の定数 C_3 を用いて、式(15)の

50

代わりに式(16)にしたがって、送信フレームレートFPS_{cur}を更新してもよい。式(15)は、定数C₃が1/2の場合の例である。

$$\text{FPS}_{\text{cur}} = \min(\text{FPS}_{\text{max}}, (1 - C_3) \times \text{FPS}_{\text{cur}} + C_3 \times \text{FPS}_{\text{feasible}}) \quad (16)$$

【0339】

また、ステップS320では、付加部315が現在時刻を取得する。付加部315はさらに、生成部312により圧縮されたデータに現在時刻を付加し、描画コマンドを生成する。

【0340】

なお、上記のとおり、実施形態に応じて描画コマンドの形式は適宜定められてよい。例えば、アプリケーション層のヘッダに時刻情報を含むよう定義された形式の描画コマンドが使われる実施形態では、付加部315は、ステップS320でアプリケーション層のヘッダ中の時刻情報用の所定のフィールドに、現在時刻を書き込む。あるいは、ペイロードに時刻情報を含むよう定義された形式の描画コマンドが使われる実施形態では、付加部315は、ステップS320でペイロード中の時刻情報用の所定のフィールドに、現在時刻を書き込む。

【0341】

そして、ステップS321で付加部315は、送信制御データ316中の最終送信時刻を、ステップS320で現在時刻として取得した時刻で上書きする。

【0342】

さらに、ステップS322で送信部313は、付加部315により生成された描画コマンドをクライアント320に送信する。そして、処理は図10のステップS303に戻る。なお、ステップS321とS322の実行順序は入れ替えられてもよい。

【0343】

以上説明した第2実施形態によれば、クライアント320とサーバ310の間のネットワークの可用帯域幅が狭くても、可用帯域幅に応じたフレームレートで画面情報が送信される。よって、遅延の累積を回避することができ、ある程度満足な操作レスポンス性能が得られる。つまり、ネットワーク状況が悪くても、ユーザビリティの低下は抑えられる。このような第2実施形態の利点について、図12を参照してさらに説明する。

【0344】

図12は、過剰遅延時間と受信ビットレートの変化の例を示すグラフである。図12のグラフ400には、過剰遅延時間を示す曲線401と、クライアント320における受信ビットレートを示す曲線402が描かれている。

【0345】

図12の例では、受信ビットレートはあまり大きく変動はしていない。しかし、例えばデフォルトの送信フレームレートが実際の可用帯域幅に対して過大である場合などに、曲線401に示すように、過剰遅延時間が次第に増加していく事態に陥ることがあり得る。図12の例では、時刻T1において、過剰遅延時間は閾値(図9の例では300ms)を超える。

【0346】

しかし、第2実施形態によれば、過剰遅延時間が閾値を超えると、クライアント320がサーバ310に通知する可用帯域幅の推定値が下がるので、それに応じて送信フレームレートも下がる。また、サーバ310は、クライアント320から通知された過剰遅延時間に応じて、画面情報の送信を省略することがある(つまり、処理が図11のステップS314から図10のステップS303へと戻ることがある)。

【0347】

よって、時刻T1の後、少し時間が経つと、サーバ310からクライアント320への通信トラフィックも減り、次第に過剰遅延時間は短くなってゆく。そして、曲線401に示すように、過剰遅延時間は、やがて小さな値で安定する。つまり、第2実施形態によれば、ネットワーク状況によって仮に一時的に過剰遅延時間が長くなってユーザビリティが多少悪化するとしても、フレームレートの調整により、ユーザビリティは、ある程度満足

10

20

30

40

50

なレベルにまで回復する。

【0348】

続いて、第3実施形態について説明する。図13は、第3実施形態のクライアントの動作フローチャートである。なお、図8のステップS201～S213は第3実施形態にも共通である。図13と図9を比較すれば分かるように、第3実施形態では、ステップS215とS216の間にステップS401～S404が追加されている。

【0349】

推定部326は、ステップS214で送信時刻 T_sent を取得し、ステップS215で式(5)により推定送信時刻 $EstT_sent$ を算出する。ステップS214とS215は図9と同様である。

10

【0350】

その後、ステップS401で推定部326は、以下の(16a)～(16c)の値を用いて、直近の受信ビットレートを算出する。

【0351】

(16a) 受信部324が、今回データ(つまり描画コマンド)を受信した時刻。換言すれば、受信履歴データ327の最新のエントリに記憶されている受信時刻 T_rcv_R 。

(16b) ステップS215で算出した推定送信時刻 $EstT_sent$ 。

(16c) 受信部324が今回受信したデータのサイズ。換言すれば、受信履歴データ327の最新のエントリに記憶されているデータサイズ $Size_rcv_R$ 。

20

【0352】

以下では説明の便宜上、ステップS401で算出される直近の受信ビットレートを「 BPS_latest 」とも表記する。推定部326は、具体的には、式(17)にしたがって直近の受信ビットレート BPS_latest を算出する。

$$BPS_latest = 8 \times Size_rcv_R / (T_rcv_R - EstT_sent) \quad (17)$$

【0353】

次に、ステップS402で推定部326は、直近の受信ビットレート BPS_latest が現在の可用帯域幅 $EstBW$ と比べて十分に大きいかな否かを判断する。具体的には、推定部326は、不等式(18)が成立するかな否かを判断する。

$$BPS_latest > (1+C_4) \times EstBW \quad (18)$$

30

【0354】

なお、不等式(18)における「 C_4 」は、適宜の正の定数である。図13には、定数 C_4 が0.1の場合が例示されている。図13に示すように、定数 C_4 が例えば0.1の場合、推定部326は、具体的には、直近の受信ビットレート BPS_latest が現在の可用帯域幅 $EstBW$ の1.1倍を超えるかな否かを判断する。

【0355】

不等式(18)が成り立つ場合、処理はステップS403に移行する。不等式(18)が成り立たない場合、処理はステップS216に移行する。

【0356】

なお、不等式(18)では、可用帯域幅 $EstBW$ に1より大きい所定の値を掛けた値が、直近の受信ビットレート BPS_latest と比較するための判定基準値として使われている。しかし、実施形態によっては、可用帯域幅 $EstBW$ に所定の正の値を足した値が、判定基準値として使われてもよい。

40

【0357】

ステップS403で推定部326は、直近の受信ビットレート BPS_latest に応じて、可用帯域幅 $EstBW$ の値を大きくする。例えば、式(19)に示すように、推定部326は、直近の受信ビットレート BPS_latest を新たな可用帯域幅 $EstBW$ として記憶してもよい。

$$EstBW = BPS_latest \quad (19)$$

【0358】

50

あるいは、式(20)に示すように、推定部326は、現在の可用帯域幅EstBWと直近の受信ビットレートBPS_{latest}との平均値を、新たな可用帯域幅EstBWとして記憶してもよい。

$$\text{EstBW} = (\text{EstBW} + \text{BPS}_{\text{latest}}) / 2 \quad (20)$$

【0359】

なお、推定部326は、 $0 < C_5 \leq 1$ なる適宜の定数 C_5 を用いて、式(21)にしたがって新たな可用帯域幅EstBWを算出してもよい。

$$\text{EstBW} = (1 - C_5) \times \text{EstBW} + C_5 \times \text{BPS}_{\text{latest}} \quad (21)$$

【0360】

式(19)は、式(21)における定数 C_5 が1の場合の具体例である。また、式(20)は、式(21)における定数 C_5 が $1/2$ の場合の具体例である。

10

【0361】

定数 C_5 の値が1に近いほど、実際の可用帯域幅が急激に増加した場合に、可用帯域幅の推定値EstBWを実際の可用帯域幅の増加に素早く追従させることができる。例えば式(19)が使われる場合、ネットワーク状況の好転に素早く対応して、フレームレートを上昇させ、フレームレートの上昇によってクライアント320の表示部322の表示をより滑らかにすることが可能である。

【0362】

また、定数 C_5 が小さいほど、可用帯域幅EstBWの変化は緩やかになる。例えば式(20)を使うことには、可用帯域幅の変動が激しい場合にも安定した制御を可能とする効果がある。

20

【0363】

なお、不等式(18)が成り立つので、式(19)～(21)のいずれが使われる場合も、可用帯域幅EstBWの値は増加する。具体的には、可用帯域幅EstBWの値は、今までの値の $(1 + C_4 \times C_5)$ 倍の値を超えることになる。

【0364】

続いて、ステップS404で推定部326は、ステップS403で更新した可用帯域幅EstBWを用いて、式(4)により推定転送時間EstT_{trans}を再計算する。そして、処理はステップS216に移行する。

【0365】

30

ステップS216で推定部326は、式(6)により、過剰遅延時間ExtraDelayを算出する。式(4)と(6)から分かるように、上記のステップS403で可用帯域幅EstBWが更新された場合は、ステップS216で算出される過剰遅延時間ExtraDelayにも可用帯域幅EstBWの更新が反映される。

【0366】

そして、ステップS217で推定部326は、過剰遅延時間ExtraDelayが300msより長いのか、0msより長く300ms以下なのか、それとも0ms以下なのかを判断する。

【0367】

過剰遅延時間ExtraDelayが300msより長ければ、推定部326は、ステップS218で式(7)にしたがって可用帯域幅データ329を更新する(すなわち可用帯域幅EstBWの値を減少させる)。実施形態によっては、式(7)の代わりに式(8)が使われてもよい。そして、推定部326は、ステップS219で過剰遅延時間ExtraDelayを再計算する。その後、ステップS220で可用帯域幅EstBWと過剰遅延時間ExtraDelayが送信部323からサーバ310に通知される。

40

【0368】

過剰遅延時間ExtraDelayが0msより長く300ms以下であれば、ステップS220で可用帯域幅EstBWと過剰遅延時間ExtraDelayが送信部323からサーバ310に通知される。

【0369】

50

過剰遅延時間 $ExtraDelay$ が $0ms$ 以下ならば、推定部 326 は、ステップ S 221 で式 (9) にしたがって可用帯域幅データ 329 を更新する（すなわち可用帯域幅 $EstBW$ の値を増加させる）。そして、推定部 326 は、ステップ S 222 で過剰遅延時間 $ExtraDelay$ を再計算する。その後、ステップ S 220 で可用帯域幅 $EstBW$ と過剰遅延時間 $ExtraDelay$ が送信部 323 からサーバ 310 に通知される。

ステップ S 220 での通知の後、処理は図 8 のステップ S 202 に戻る。

【0370】

なお、図 13 から分かるように、第 3 実施形態では、以下の (17a) ~ (17c) のいずれの場合もあり得る。以下の (17a) ~ (17c) のいずれの場合も、推定部 326 が可用帯域幅 $EstBW$ を増加させるという点と、増加した可用帯域幅 $EstBW$ がサーバ 310 に通知されるという点では共通である。

10

【0371】

(17a) ステップ S 403 と S 221 の両方が実行される場合。

(17b) ステップ S 403 は実行され、ステップ S 221 は実行されない場合。

(17c) ステップ S 403 は実行されず、ステップ S 221 が実行される場合。

【0372】

また、ステップ S 402 の式 (18) における定数 C_4 と、ステップ S 403 の式 (21) における定数 C_5 と、ステップ S 221 の式 (9) における定数 C_2 は、実施形態に応じて適宜に決められてよい。例えば、3つの定数は、不等式 (22) を満たすように決められていてもよい。

20

$$C_2 \leq C_4 \times C_5 \quad (22)$$

【0373】

不等式 (22) は、可用帯域幅の推定値 $EstBW$ の算出において、ステップ S 221 に相当する下記 (18a) の判断よりも、ステップ 403 に相当する下記 (18b) の追従の方を重視することを意味する。

【0374】

(18a) 過剰遅延時間が所定の閾値（具体的には、ステップ S 217 で使われる $0ms$ という値）以下の場合における、「可用帯域幅の実際の値は、現在の推測値よりも大きい可能性がある」という判断。

30

(18b) 実際の直近のビットレートの増加への追従。

【0375】

なお、第 3 実施形態は以下のように変形されてもよい。具体的には、ステップ S 403 が実行された場合、ステップ S 404 と S 216 の実行後、推定部 326 は、ステップ S 217 の判断を省略して、可用帯域幅 $EstBW$ と過剰遅延時間 $ExtraDelay$ を送信部 323 に出力してもよい。そして、ステップ S 220 で送信部 323 が可用帯域幅 $EstBW$ と過剰遅延時間 $ExtraDelay$ をサーバ 310 に通知してもよい。

【0376】

続いて、第 4 実施形態について説明する。以下に述べるように、第 4 実施形態には、可用帯域幅の推定精度を向上させる効果がある。

40

【0377】

具体的には、第 4 実施形態では、過剰遅延時間 $ExtraDelay$ が所定の閾値（例えば $0ms$ ）以下の場合に、クライアント 320 は、すぐに可用帯域幅の推定値 $EstBW$ を増加させる代わりに、今までの推定値よりも大きな値をサーバ 310 に通知する。クライアント 320 は、大きな値をサーバ 310 に通知することにより、間接的に送信フレームレートを制御する（具体的には、上昇させる）。

【0378】

サーバ 310 が送信する描画コマンドのデータサイズは、実行中のアプリケーションやユーザからの入力に応じて変動する。しかし、大まかな傾向としては、送信フレームレートが高いほど、単位時間あたりにサーバ 310 がクライアント 320 に送信するデータの

50

量も多い。

【0379】

よって、実際の可用帯域幅が可用帯域幅の現在の推定値よりも大きければ、送信フレームレートが上昇しても経路上でデータが滞留することはないと考えられる。つまり、実際の可用帯域幅が可用帯域幅の現在の推定値よりも大きければ、送信フレームレートの上昇に応じて受信ビットレートも上昇すると考えられる。

【0380】

よって、クライアント320は、上記のように大きな値をサーバ310に通知した後に、実際にクライアント320における受信ビットレートの上昇を確認したら、そのときに可用帯域幅の推定値EstBWを増加させる。つまり、クライアント320は、送信フレームレートの上昇に対処するだけの余裕が実際の可用帯域幅にあったことを確認してから、可用帯域幅の推定値EstBWを増加させる。

10

【0381】

以上のように、第4実施形態では、実際の受信ビットレートの上昇に基づいて可用帯域幅の推定値EstBWをクライアント320が増加させるため、可用帯域幅の推定精度が向上する。可用帯域幅の推定精度の向上により、過剰遅延時間ExtraDelayの精度も向上する。よって、第4実施形態では、より実際の可用帯域幅に即した頻度で、画面情報（具体的には描画コマンド）が送信される。

【0382】

なお、第4実施形態では、上記のように可用帯域幅の推定値EstBWよりも大きな値をクライアント320がサーバ310に通知する場合もあるし、可用帯域幅の推定値EstBW自体をクライアント320がサーバ310に通知する場合もある。いずれの場合も、クライアント320は、通知した値を記憶する。

20

【0383】

具体的には、第4実施形態においては、図6の可用帯域幅データ329は、図7の可用帯域幅データ329bのように、可用帯域幅だけでなく前回通知帯域幅を含む。クライアント320は、サーバ310に通知した値を、前回通知帯域幅として記憶する。以下では便宜上、前回通知帯域幅を「PrevRepBW」と表記することがある。

【0384】

続いて、図14を参照して第4実施形態についてさらに詳しく説明する。図14は、第4実施形態のクライアントの動作フローチャートである。なお、図8のステップS201～S213は第4実施形態にも共通である。

30

【0385】

図14と図13を比較すれば分かるように、第4実施形態では、図13のステップS217とステップS220で挟まれた処理が、図14の処理に置き換わる。つまり、第4実施形態では、図8のステップS213の後、図13のステップS214、S215、S401、およびS402が実行される。さらに、ステップS402の判定結果によっては、ステップS403とS404も実行される。その後、ステップS216が実行されて過剰遅延時間ExtraDelayが算出されてから、図14のステップS217が実行される。

40

【0386】

よって、図14のステップS217が実行される時点における可用帯域幅EstBWの値は、以下の(19a)～(19c)のいずれかである。

【0387】

(19a) ステップS212で式(3)により算出された値。

(19b) ステップS403で式(21)により更新された値。

(19c) 今回の描画コマンドが受信された時点で既に可用帯域幅データ329bに可用帯域幅として記憶されていたままの値。

【0388】

また、ステップS217において参照される過剰遅延時間ExtraDelayは、式

50

(4)と(6)に示すように、可用帯域幅 $EstBW$ に依存する。ステップ $S217$ では、第2～第3実施形態と同様に、推定部 326 が、過剰遅延時間 $ExtraDelay$ が $0ms$ 以下なのか、 $0ms$ より長く $300ms$ 以下なのか、それとも $300ms$ より長いのかを判断する。

【0389】

過剰遅延時間 $ExtraDelay$ が $0ms$ 以下の場合、処理はステップ $S501$ に移行する。過剰遅延時間 $ExtraDelay$ が $0ms$ より長く $300ms$ 以下の場合、処理はステップ $S502$ に移行する。過剰遅延時間 $ExtraDelay$ が $300ms$ より長い場合、処理はステップ $S503$ に移行する。

【0390】

10

ステップ $S501$ で推定部 326 は、図9や図13のステップ $S221$ のように可用帯域幅 $EstBW$ の値を増加させる代わりに、可用帯域幅 $EstBW$ よりも大きな値を通知帯域幅として設定する。以下では便宜上、通知帯域幅を「 $RepBW$ 」とも表記する。

【0391】

通知帯域幅 $RepBW$ は、クライアント 320 がサーバ 310 に通知する値である。後述のように、可用帯域幅 $EstBW$ 自体が通知帯域幅 $RepBW$ として使われる場合もある。

【0392】

具体的には、ステップ $S501$ で推定部 326 は、比較的小さな正の適宜の定数 C_6 を用いて、式(23)により通知帯域幅 $RepBW$ の値を決定する。図14には、 $C_6 = 0$

20

$$RepBW = (1+C_6) \times EstBW \quad (23)$$

【0393】

なお、上記(19b)に示すとおり、式(23)の可用帯域幅 $EstBW$ は、ステップ $S403$ で増加した値の場合もあり得る。この場合、ある観点から見れば、式(23)により算出される通知帯域幅 $RepBW$ は、可用帯域幅 $EstBW$ の増加を強調して示す1種の指標値でもある。

【0394】

実施形態によっては、推定部 326 は、式(23)のように1より大きい所定の値を可用帯域幅 $EstBW$ に掛ける代わりに、所定の正の値を可用帯域幅 $EstBW$ に足すこと

30

【0395】

なお、ステップ $S501$ が実行されるのは過剰遅延時間 $ExtraDelay$ が $0ms$ 以下の場合である。この場合、「可用帯域幅の現在の推定値 $EstBW$ が実際の値と比べて過小である」という可能性がある。よって、この可能性が実際に成り立つのか否かを検証するため、推定部 326 は、ステップ $S501$ で通知帯域幅 $RepBW$ を可用帯域幅 $EstBW$ よりも大きく設定する。ステップ $S501$ の実行後、処理はステップ $S505$ へと移行する。

【0396】

ステップ $S502$ で推定部 326 は、可用帯域幅 $EstBW$ の現在の値を通知帯域幅 $RepBW$ として設定する。つまり、ステップ $S502$ は式(24)のように表される。ステップ $S502$ の実行後、処理はステップ $S505$ へと移行する。

40

$$RepBW = EstBW \quad (24)$$

【0397】

ステップ $S503$ で推定部 326 は、可用帯域幅データ $329b$ を参照し、可用帯域幅 $EstBW$ が前回通知帯域幅 $PrevRepBW$ 未満か否かを判断する。すなわち、推定部 326 は、不等式(25)が成り立つか否かを判断する。

$$EstBW < PrevRepBW \quad (25)$$

【0398】

不等式(25)が成り立つ場合、処理はステップ $S502$ に移行する。逆に、不等式(

50

25) が成り立たない場合、処理はステップ S 2 1 8 に移行する。なお、ステップ S 5 0 3 での判断の意味については、ステップ S 5 0 6 まで説明してから述べる。

【0399】

ステップ S 2 1 8 で推定部 3 2 6 は、以下に再掲する式 (7) にしたがって (あるいは、式 (7) を一般化した式 (8) にしたがって)、可用帯域幅 $E_{st} BW$ を更新する。図 1 4 には、ステップ S 2 1 8 で式 (7) が使われる場合が例示されている。

$$E_{st} BW = (E_{st} BW + BPS_{avr}) / 2 \quad (7)$$

$$E_{st} BW = (1 - C_1) \times E_{st} BW + C_1 \times BPS_{avr} \quad (8)$$

【0400】

そして、次のステップ S 2 1 9 で推定部 3 2 6 は、ステップ S 2 1 8 で更新した可用帯域幅 $E_{st} BW$ を用いて、過剰遅延時間 $Extra Delay$ を再計算する。第 2 実施形態に関して説明したように、再計算による値の変化は比較的小さいので、ステップ S 2 1 9 での再計算は省略されてもよい。

【0401】

さらに、次のステップ S 5 0 4 で推定部 3 2 6 は、ステップ S 2 1 8 で更新した可用帯域幅 $E_{st} BW$ の値を、通知帯域幅 $Rep BW$ として設定する。ステップ S 5 0 4 は式 (26) のように表される。

$$Rep BW = E_{st} BW \quad (26)$$

【0402】

通知帯域幅 $Rep BW$ の設定後、処理はステップ S 5 0 5 に移行する。なお、実施形態によっては、ステップ S 2 1 9 と S 5 0 4 の実行順序が入れ替えられてもよい。

【0403】

ステップ S 5 0 5 では、推定部 3 2 6 が通知帯域幅 $Rep BW$ と過剰遅延時間 $Extra Delay$ を送信部 3 2 3 に出力する。すると、送信部 3 2 3 は、通知帯域幅 $Rep BW$ と過剰遅延時間 $Extra Delay$ を含む通知をサーバ 3 1 0 に送信する。

【0404】

なお、サーバ 3 1 0 は、クライアント 3 2 0 から通知された通知帯域幅 $Rep BW$ を可用帯域幅 $E_{st} BW$ として扱う。よって、クライアント 3 2 0 が通知した通知帯域幅 $Rep BW$ は、図 10 のステップ S 3 0 5 において、サーバ 3 1 0 の送信制御データ 3 1 6 における可用帯域幅 $E_{st} BW$ として記憶される。

【0405】

さて、ステップ S 5 0 5 の実行後、ステップ S 5 0 6 で推定部 3 2 6 は、通知帯域幅 $Rep BW$ を、可用帯域幅データ 3 2 9 b 中の前回通知帯域幅 $Prev Rep BW$ として記憶する。そして、処理は図 8 のステップ S 2 0 2 に戻る。なお、実施形態によっては、ステップ S 5 0 5 と S 5 0 6 の実行順序が入れ替えられてもよい。

【0406】

ここで、ステップ S 5 0 3 の判断の意味を説明する。ステップ S 5 0 3 で不等式 (25) が成り立つ場合とは、具体的には、前回ステップ S 5 0 1 で可用帯域幅 $E_{st} BW$ よりも大きく設定された値が、前回通知帯域幅 $Prev Rep BW$ として記憶されている場合である。また、ステップ S 5 0 3 が実行されるのは、過剰遅延時間 $Extra Delay$ が 300ms より長い場合である。

【0407】

したがって、ステップ S 5 0 3 において不等式 (25) が成り立つことは、「可用帯域幅の推定値 $E_{st} BW$ が実際の値と比べて過小なのか否かを検証したら、可用帯域幅の推定値 $E_{st} BW$ が過小でないことが判明した」ということを意味する。より詳しくは以下のとおりである。

【0408】

上記のように、ステップ S 5 0 1 で、推定部 3 2 6 は、可用帯域幅の推定値 $E_{st} BW$ が実際の値と比べて過小なのか否かを検証するために、可用帯域幅の推定値 $E_{st} BW$ よりも大きく通知帯域幅 $Rep BW$ を設定することがある。クライアント 3 2 0 は、こうし

10

20

30

40

50

て大きく設定した通知帯域幅 $R_{ep BW}$ をサーバ 310 に通知する。

【0409】

その結果、サーバ 310 が送信フレームレートを上昇させ、単位時間当たりの通信トラフィックが増える。一方、もし可用帯域幅の推定値 $E_{st BW}$ が実際の値と比べて過小でないならば、以上のような単位時間当たりの通信トラフィックの増加の結果として、サーバ 310 からクライアント 320 への経路上でデータの滞留が生じ、過剰遅延時間も長くなる。

【0410】

このようにして過剰遅延時間が 300ms という閾値を超えてしまう場合、可用帯域幅 $E_{st BW}$ は、クライアント 320 が以前サーバ 310 に通知した通知帯域幅 $R_{ep BW}$ より小さい。なぜなら、推定部 326 はまだ可用帯域幅 $E_{st BW}$ を増加させてはいないからである。

10

【0411】

また、このようにして過剰遅延時間が 300ms を超えてしまう場合は、クライアント 320 が前回サーバ 310 に通知した通知帯域幅 $R_{ep BW}$ は、実際の可用帯域幅と比べて過大である。つまり、現在可用帯域幅データ 329b に記録されている前回通知帯域幅 $P_{rev Rep BW}$ は過大なのであって、むしろ可用帯域幅 $E_{st BW}$ が適正である。換言すれば、可用帯域幅 $E_{st BW}$ は過小ではない。

【0412】

ステップ S503 からステップ S502 への処理の流れは、以上のようにして可用帯域幅 $E_{st BW}$ が過小ではなく適正であることが確認された場合に、その適正な可用帯域幅 $E_{st BW}$ をクライアント 320 がサーバ 310 に通知しなおすことを意味する。

20

【0413】

他方、たとえステップ S501 が実行されることがないとしても、ネットワークの状況の変化によっては、過剰遅延時間が 300ms を超えてしまう場合もあり得る。その場合は、現在の可用帯域幅 $E_{st BW}$ ですら過大であるから、クライアント 320 は、現在の可用帯域幅 $E_{st BW}$ よりも小さな値をサーバ 310 に通知することが望ましい。

【0414】

ステップ S503 からステップ S218 への処理の流れは、以上のようにクライアント 320 によるステップ S501 の処理とは無関係に起こり得るネットワーク状況の悪化に起因する過剰遅延時間の増大への対処を意味する。

30

【0415】

クライアント 320 の動作とは関係なくネットワーク状況が悪化した場合は、以前ステップ S502 または S504 で設定された通知帯域幅 $R_{ep BW}$ が前回通知帯域幅 $P_{rev Rep BW}$ として現在記憶されている。よって、推定部 326 は、ステップ S503 で可用帯域幅 $E_{st BW}$ が前回通知帯域幅 $P_{rev Rep BW}$ 以上であると判断し、ステップ S218 で可用帯域幅 $E_{st BW}$ を減少させる。そして、推定部 326 は、ステップ S504 で、減少させた可用帯域幅 $E_{st BW}$ を、新たに通知帯域幅 $R_{ep BW}$ として設定する。その結果、ステップ S505 では、減少させた可用帯域幅 $E_{st BW}$ がサーバ 310 に通知される。

40

【0416】

なお、可用帯域幅の推定値 $E_{st BW}$ が実際の値と比べて過小なのか否かを検証するために推定部 326 がステップ S501 で通知帯域幅 $R_{ep BW}$ を大きく設定した後、可用帯域幅の推定値 $E_{st BW}$ が過小だと判明する場合もあり得る。具体的には、次のような場合である。

【0417】

もし可用帯域幅の推定値 $E_{st BW}$ が過小ならば、ステップ S501 で可用帯域幅の推定値 $E_{st BW}$ よりも大きく設定された通知帯域幅 $R_{ep BW}$ がサーバ 310 に通知されると、クライアント 320 における実際の受信ビットレートが上昇する。つまり、ステップ S501 の実行後の受信ビットレートの上昇により、可用帯域幅の推定値 $E_{st BW}$ が

50

過小であることが判明する。

【0418】

具体的には、ステップS402で受信ビットレートの上昇が検出される場合、可用帯域幅の推定値EstBWが過小であることが判明する。よって、この場合、ステップS403で推定部326は、上昇した実際の受信ビットレートに基づいて、可用帯域幅の推定値EstBWを増加させる。

【0419】

以上のように、第4実施形態では、ステップS402の判断は、ステップS501での通知帯域幅RepBWの設定と関連する場合がある。よって、ステップS402の式(18)の定数 C_4 と、ステップS501の式(23)の定数 C_6 は、定数 C_4 が定数 C_6 より

10

もわずかに小さくなるように定められていることが好ましい。

【0420】

ところで、以上説明した第4実施形態では、図7の可用帯域幅データ329bに含まれる通知帯域幅の履歴は、過去の1つの通知帯域幅（すなわち前回通知帯域幅PrevRepBW）のみである。しかし、実施形態によっては、過去の2つ以上の通知帯域幅を可用帯域幅データ329bが含んでいてもよい。この場合、ステップS503で推定部326は、例えば、過去の2つ以上の通知帯域幅のうち、1つでも可用帯域幅より大きいものがあるか否かを判断してもよい。

【0421】

そして、過去の2つ以上の通知帯域幅のうち、1つでも可用帯域幅より大きいものがあるれば、処理がステップS503からステップS502へと移行する。逆に、過去の2つ以上の通知帯域幅のすべてが可用帯域幅以下ならば、処理はステップS503からS218へと移行する。

20

【0422】

あるいは、ステップS503で推定部326は、1より大きい所定の定数 C_7 を用いて、過去の2つ以上の通知帯域幅のうち、1つでも、定数 C_7 を可用帯域幅に乗じた基準値より大きいものがあるか否かを判断してもよい。そして、過去の2つ以上の通知帯域幅のうち、1つでも基準値より大きいものがあるれば、処理がステップS503からステップS502へと移行する。逆に、過去の2つ以上の通知帯域幅のすべてが基準値以下ならば、処理はステップS503からS218へと移行する。

30

【0423】

なお、所定の定数 C_7 は、例えば、ステップS501で使われる定数 C_6 との間に不等式(27)の関係が成り立つように定められていてもよい。

$$1 < C_7 < (1+C_6) \quad (27)$$

【0424】

また、過去の2つ以上の通知帯域幅を可用帯域幅データ329bに記憶するために、リングバッファが使われてもよい。ステップS506では、推定部326は、今回設定した通知帯域幅を、1番近い過去の通知帯域幅として記憶すればよい。

【0425】

以上のように、過去の2つ以上の通知帯域幅を含む可用帯域幅データ329bを利用する実施形態は、通知帯域幅の上昇から送信フレームレートの上昇を経て遅延の累積に至るまでにある程度のタイムラグが生じる場合にも好適である。

40

【0426】

なお、本発明は上記の第1～第4実施形態に限られるものではなく、上記の第1～第4実施形態は様々に変形可能である。以下に、変形に関するいくつかの観点を例示する。例えば、上記の第1～第4実施形態は、下記の観点から様々に変形することができ、これらの変形は、相互に矛盾しない限り、任意に組み合わせることが可能である。

【0427】

上記の第1～第4実施形態におけるいくつかの処理は、閾値との比較を含む。例えば、図9のステップS217、図10のステップS306とS312、図11のステップS3

50

14、図13のステップS402とS217、図14のステップS217とS503では、閾値との比較が行われる。

【0428】

閾値との比較は、実施形態により「比較対象の数値が、閾値を超えるか否か」を判断する処理でもよいし、「比較対象の数値が、閾値以上か否か」を判断する処理でもよい。また、上記の説明では様々な用途の閾値を例示したが、各閾値の具体的な値は、実施形態に応じて適宜任意に決められてよい。

【0429】

図7にはテーブル形式で各種データを示したが、実施形態に応じて他のデータ形式が使われてもよい。

10

【0430】

また、第2～第4実施形態では時刻合わせコマンドが使われるが、実施形態によっては、時刻合わせコマンドが使われなくてもよい。代わりに、NTP (Network Time Protocol) またはSNTP (Simple Network Time Protocol) などのプロトコルにしたがって、クライアント320の時計をサーバ310の時計に同期させる処理が事前に行われてもよい。例えば、図8のステップS201と図10のステップS301に示すリモートデスクトップ初期化処理が、NTPまたはSNTPによる同期処理を含んでいてもよい。

【0431】

また、時刻合わせコマンドが使われる実施形態において、サーバ310は、時刻合わせコマンドを1回だけ送信するのではなく、定期的に、または不定期に、複数回、時刻合わせコマンドを送信してもよい。その場合、クライアント320の受信部324は、時刻合わせコマンドを受信するたびに時刻差データ328を更新する。

20

【0432】

なお、図6には1台のクライアント320が描かれているが、サーバ310は、複数のクライアント320のそれぞれに対して画像を生成し、画面情報と時刻情報を含む描画コマンドを送信してもよい。その場合、サーバ310は、クライアント320ごとに送信制御データ316を管理する。クライアント320ごとの管理のために、送信制御データ316は、クライアント320を識別する識別情報をさらに含んでいてもよい。

【0433】

また、クライアント320が複数台の場合、サーバ310は、複数のクライアント320のそれぞれに対応して独立に図10～11の処理を実行する。例えば、サーバ310は、複数のスレッドで並行してそれぞれ図10～11の処理を実行してもよい。つまり1つのスレッドが1台のクライアント320に対応していてもよい。

30

【0434】

また、図8～11、13、および14のように、いくつか具体的なフローチャートを例示したが、矛盾が生じない限り、実施形態に応じて、ステップの実行順序が入れ替えられてもよい。また、入れ替え可能なステップ同士（すなわち互いに独立したステップ同士）は、並列に実行されてもよい。

【0435】

ステップの実行順序の入れ替えやステップの省略については、いくつかの例を上記の説明においても述べたが、さらに、例えば以下のような変更も可能である。

40

例えば、クライアント320は、第2～第4実施形態のように過剰遅延時間をサーバ310に通知することが好ましいが、過剰遅延時間をサーバ310に通知しなくてもよい。過剰遅延時間が通知されない実施形態では、ステップS307、S313、およびS314は省略される。

【0436】

また、クライアント320は、第2～第4実施形態のように、描画コマンドの受信のたびに、可用帯域幅と過剰遅延時間の算出と通知を行ってもよい。しかし、可用帯域幅と過剰遅延時間の通知に起因するネットワーク負荷を減少させるため、クライアント320は、描画コマンドをN₁回受信するごとに1回だけ、可用帯域幅と過剰遅延時間の算出と通

50

知を行ってもよい ($N_1 > 1$)。クライアント 320 が可用帯域幅と過剰遅延時間の算出と通知を行わない場合には、図 8 のステップ S 209 の実行後、処理はステップ S 202 に戻る。

【0437】

あるいは、クライアント 320 は、描画コマンドの受信のたびに可用帯域幅と過剰遅延時間を算出し、直近の N_2 回の描画コマンドの受信のときに算出した可用帯域幅と過剰遅延時間の値の履歴を保持してもよい ($N_2 \geq 1$)。そして、クライアント 320 は、履歴を参照して、可用帯域幅の変化の大きさまたは変化の傾向を判断してもよく、過剰遅延時間の変化の大きさまたは変化の傾向を判断してもよい。

【0438】

クライアント 320 は、可用帯域幅の変化の大きさもしくは変化の傾向が所定の基準を満たす場合か、または、過剰遅延時間の変化の大きさもしくは変化の傾向が所定の基準を満たす場合にだけ、可用帯域幅と過剰遅延時間をサーバ 310 に送信してもよい。例えば、上記の所定の基準は、以下の (20a) ~ (20c) に例示するような基準であってもよいし、その他の基準であってもよい。

【0439】

(20a) 履歴中の N_2 個の値のうちのいずれか (例えば最大値または最小値) と比べて、今回算出された値は、所定のパーセンテージ以上に大きく変化している。

(20b) 履歴中の N_2 個の値の平均値と比べて、今回算出された値は、所定のパーセンテージ以上に大きく変化している。

(20c) 値が N_2 回連続して増加しているか、または、 N_2 回連続して減少している。

【0440】

例えば以上のようにして、可用帯域幅と過剰遅延時間を通知する場合を制限することにより、クライアント 320 は、可用帯域幅と過剰遅延時間の通知に起因するネットワーク負荷を抑えてもよい。

【0441】

また、ステップ S 209 での描画処理と、ステップ S 210 以降の処理は、並列に実行されてもよい。ステップ S 210 以降の処理 (つまり、ステップ S 220 または S 506 までの処理) の後に、ステップ S 209 が実行されてもよい。

【0442】

あるいは、ステップ S 210 がステップ S 204 の直後に移動されてもよい。ステップ S 213 がステップ S 215 と S 216 の間に移動されてもよい。

【0443】

送信制御データ 316 中の送信フレームレートの更新に関わるステップ S 317 ~ S 319 の処理は、描画コマンドの生成と送信に関わるステップ S 315 ~ S 316 および S 320 ~ S 322 の処理とは独立している。よって、これら 2 つの処理は、並列に実行されてもよい。

【0444】

また、図 10 ~ 11 に示すように、第 2 ~ 第 4 実施形態のサーバ 310 は、所定の間隔でステップ S 308 を実行し、画面全体の画像を取得する。しかし、ステップ S 312 または S 314 の判断結果によっては、サーバ 310 は、取得した画像に対応する描画コマンドを送信しない場合がある。

【0445】

このように、「描画コマンドを送信しない条件が成り立つにもかかわらず、画像が取得される」という場合がある理由は、ステップ S 310 での動画領域の検出が、固定された間隔での画面の変化に基づいて行われるようにするためである。しかし、実施形態によっては、ステップ S 308 ~ S 310 は、例えばステップ S 315 の直前に移動されてもよい。つまり、可変の送信フレームレートに応じた間隔で、動画領域の検出が行われてもよい。

10

20

30

40

50

【0446】

最後に、上記の種々の実施形態に関して、さらに下記の付記を開示する。

(付記1)

第1のコンピュータに、

ネットワークを介して前記第1のコンピュータと接続された第2のコンピュータであって、

前記第1のコンピュータの入力装置を介して前記第1のコンピュータに与えられた入力についての通知を前記第1のコンピュータから受信し、

前記入力に応じて1つ以上のプログラムを実行し、

前記1つ以上のプログラムの実行に応じて、前記第1のコンピュータの画面に表示するための画像を生成し、

前記第1のコンピュータに前記画像を前記画面上に表示させるための画面情報を送信するとともに、前記画面情報を送信する第1の時刻を示す時刻情報を送信し、

前記第1のコンピュータから通知される値に基づいて、前記画面情報を送信する頻度を調整しながら、前記画像の生成ならびに前記画面情報および前記時刻情報の送信を繰り返す

ことを特徴とする第2のコンピュータから、前記画面情報と前記時刻情報を受信し、

前記第1のコンピュータと前記第2のコンピュータとの間の前記ネットワークの可用帯域幅の推定値を、

前記第1のコンピュータが前記画面情報を受信した第2の時刻と、

前記推定値と、受信した前記画面情報のサイズと、受信した前記時刻情報により示される前記第1の時刻から、前記第1のコンピュータが前記画面情報を受信するものと推定される第3の時刻と

の差に応じて、増加させるか、維持するか、または、ある期間内に1回以上受信した前記画面情報それぞれのサイズと前記期間の長さから算出される第1の受信ビットレートに基づいて減少させ、

少なくとも前記推定値を増加または減少させることで更新した場合には、前記推定値または前記推定値の更新を反映する指標値を前記第2のコンピュータに通知する

ことを含む画面更新制御処理を実行させる画面更新制御プログラム。

(付記2)

前記推定値を前記差に応じて増加させるか、維持するか、または、減少させる処理は、

前記差が正の第1の閾値より大きいとき、前記第1の受信ビットレートに基づいて前記推定値を減少させ、

前記差が前記第1の閾値未満の第2の閾値以下のとき、前記推定値を増加させ、

前記差が前記第2の閾値より大きく前記第1の閾値以下のとき、前記推定値を維持する

ことを含むことを特徴とする付記1に記載の画面更新制御プログラム。

(付記3)

前記推定値を増加させる処理が、前記推定値を所定の割合で増加させるかまたは所定の値だけ増加させる処理である

ことを特徴とする付記1または2に記載の画面更新制御プログラム。

(付記4)

直近に受信された前記画面情報に関しての、

前記第1の時刻と

前記第2の時刻と

前記サイズ

から算出される第2の受信ビットレートが、前記推定値に基づいて前記推定値よりも大きく定められる判定基準値よりも大きい場合、前記第2の受信ビットレートに応じて前記推定値を増加させる

ことを前記画面更新制御処理が含むことを特徴とする付記1から3のいずれか1項に記

10

20

30

40

50

載の画面更新制御プログラム。

(付記 5)

前記差が所定の閾値以下の場合に、

前記推定値を維持するとともに、

前記推定値よりも大きな値を前記第 2 のコンピュータに通知する

ことを前記画面更新制御処理が含むことを特徴とする付記 1、3、または 4 に記載の画面更新制御プログラム。

(付記 6)

前記画面更新制御処理は、前記差を前記第 2 のコンピュータに通知することを含む

ことを特徴とする付記 1 から 5 のいずれか 1 項に記載の画面更新制御プログラム。

10

(付記 7)

最初に受信された前記画面情報に関しての、

前記第 1 の時刻と

前記第 2 の時刻と

前記サイズ

から算出される第 3 の受信ビットレートをを用いて、前記推定値を初期化することを前記画面更新処理が含む

ことを特徴とする付記 1 から 6 のいずれか 1 項に記載の画面更新制御プログラム。

(付記 8)

前記ある期間は、現在から所定範囲内に含まれる期間である

20

ことを特徴とする付記 1 から 7 のいずれか 1 項に記載の画面更新制御プログラム。

(付記 9)

第 1 のコンピュータが、前記第 1 のコンピュータの入力装置を介して前記第 1 のコンピュータに与えられた入力を、ネットワークを介して前記第 1 のコンピュータと接続された第 2 のコンピュータに通知し、

前記第 2 のコンピュータが、前記通知された入力に応じて 1 つ以上のプログラムを実行し、

前記第 2 のコンピュータが、前記 1 つ以上のプログラムの実行に応じて、前記第 1 のコンピュータの画面に表示するための画像を生成し、

前記第 2 のコンピュータが、前記第 1 のコンピュータに前記画像を前記画面上に表示させるための画面情報を送信するとともに、前記画面情報を送信する第 1 の時刻を示す時刻情報を送信し、

30

前記第 1 のコンピュータが、前記画面情報と前記時刻情報を受信し、

前記第 1 のコンピュータが、前記第 1 のコンピュータと前記第 2 のコンピュータとの間の前記ネットワークの可用帯域幅の推定値を、

前記第 1 のコンピュータが前記画面情報を受信した第 2 の時刻と、

前記推定値と、受信した前記画面情報のサイズと、受信した前記時刻情報により示される前記第 1 の時刻から、前記第 1 のコンピュータが前記画面情報を受信するものと推定される第 3 の時刻

の差に応じて、増加させるか、維持するか、または、ある期間内に 1 回以上受信した前記画面情報それぞれのサイズと前記期間の長さから算出される受信ビットレートに基づいて減少させ、

40

前記第 1 のコンピュータが、少なくとも前記推定値を増加または減少させることで更新した場合には、前記推定値または前記推定値の更新を反映する指標値を前記第 2 のコンピュータに通知し、

前記第 2 のコンピュータが、通知された前記推定値または通知された前記指標値に基づいて、前記画面情報を送信する頻度を調整しながら、前記画像の生成ならびに前記画面情報および前記時刻情報の送信を繰り返す

ことを含む画面更新制御方法。

(付記 10)

50

前記画面更新制御方法は、前記第１のコンピュータが前記差を前記第２のコンピュータに通知することを含み、

前記第２のコンピュータは、通知された前記差にも基づいて前記頻度を調整することを特徴とする付記９に記載の画面更新制御方法。

(付記１１)

前記差が所定の閾値以下の場合に、前記第１のコンピュータが、

前記推定値を維持するとともに、

前記推定値よりも大きな値を前記第２のコンピュータに通知する

ことを特徴とする付記９または１０に記載の画面更新制御方法。

(付記１２)

ネットワークを介して接続された第１の情報処理装置と第２の情報処理装置を含むシステムで使われる前記第１の情報処理装置であって、

前記第１の情報処理装置と前記第２の情報処理装置との間の前記ネットワークの可用帯域幅の推定値または前記推定値の更新を反映する指標値を示す通知を、前記第２の情報処理装置に送信し、前記第１の情報処理装置に入力が与えられると、前記入力についての通知を前記第２の情報処理装置に送信する送信部と、

前記第２の情報処理装置が、前記入力についての前記通知に基づいて１つ以上のプログラムを実行し、前記１つ以上のプログラムの実行に応じて、前記第１の情報処理装置の画面に表示するための画像を生成し、前記第１の情報処理装置に前記画像を前記画面上に表示させるための画面情報を送信するとともに、前記画面情報を送信する第１の時刻を示す時刻情報を送信し、前記推定値または前記指標値に基づいて前記画面情報を送信する頻度を調整しながら前記画像の生成ならびに前記画面情報および前記時刻情報の送信を繰り返すと、前記第２の情報処理装置から、順次送信される前記画面情報と前記時刻情報をそれぞれ受信する受信部と、

前記推定値を、

前記画面情報が前記受信部により受信された第２の時刻と、

前記推定値と、前記受信部が受信した前記画面情報のサイズと、前記受信部が受信した前記時刻情報により示される前記第１の時刻から、前記受信部が前記画面情報を受信するものと推定される第３の時刻と

の差に応じて、増加させるか、維持するか、または、ある期間内に１回以上前記受信部により受信された前記画面情報それぞれのサイズと前記期間の長さから算出される受信ビットレートに基づいて減少させる推定部と、

を備えることを特徴とする第１の情報処理装置。

【符号の説明】

【０４４７】

１０１、１０２、２００、２４０ コンピュータ

１０３ ユーザ

１１０ 画面

１１１～１１３ 領域

１２１～１２５ タイミングチャート

F１～F７ 画面情報

１３１、１３２、４００ グラフ

２０１、２４１ CPU

２０２、２４２ ROM

２０３、２４３ RAM

２０４、２４４ 通信インタフェース

２０５、２４５ 入力装置

２０６、２４６ 出力装置

２０７、２４７ 記憶装置

２０８、２４８ 駆動装置

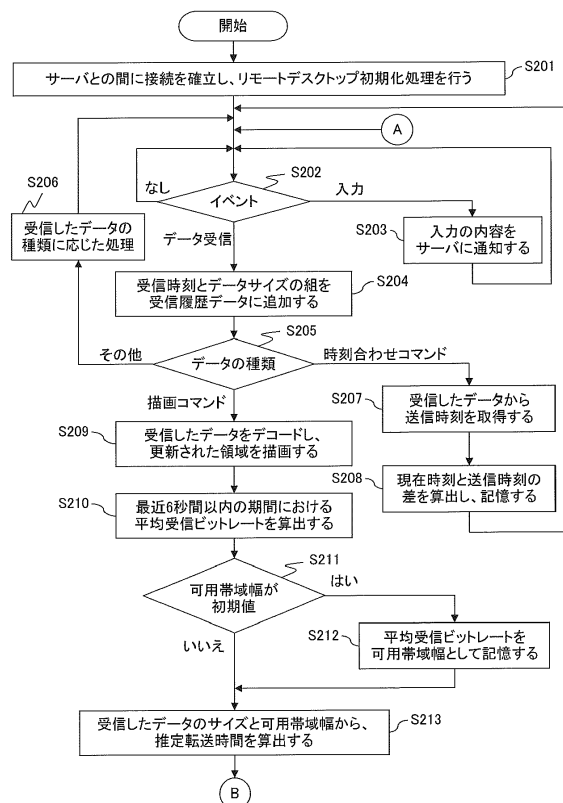
209、249 バス
 210、250 記憶媒体
 220 ネットワーク
 230 プログラム提供者
 300 システム
 310 サーバ
 311 受信部
 312 生成部
 313 送信部
 314 調整部
 315 付加部
 316 送信制御データ
 320 クライアント
 321 入力部
 322 表示部
 323 送信部
 324 受信部
 325 描画部
 326 推定部
 327 受信履歴データ
 328 時刻差データ
 329、329 a、329 b 可用帯域幅データ
 401、402 曲線

10

20

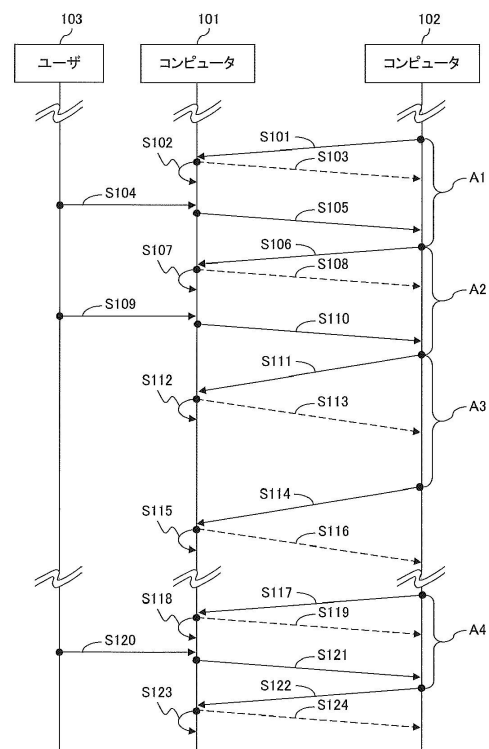
【図 8】

第2実施形態のクライアントの動作フローチャート(その1)



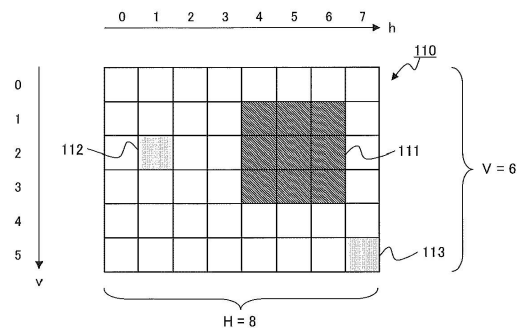
【図 1】

第1実施形態の動作例を示すシーケンス図



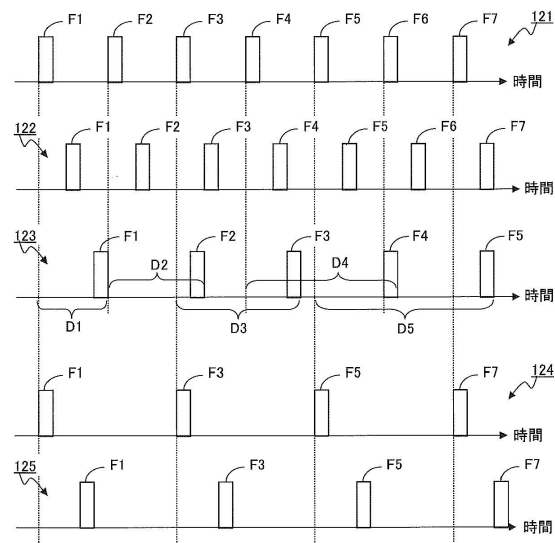
【図 2】

画面の更新について説明する図



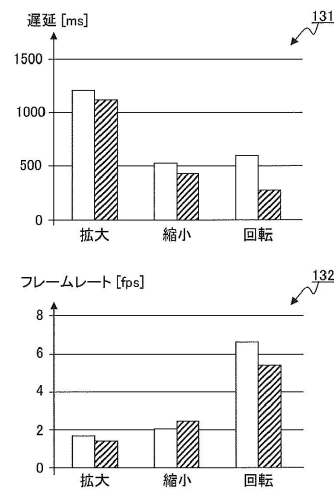
【図 3】

第1実施形態の効果を説明する図 (その1)



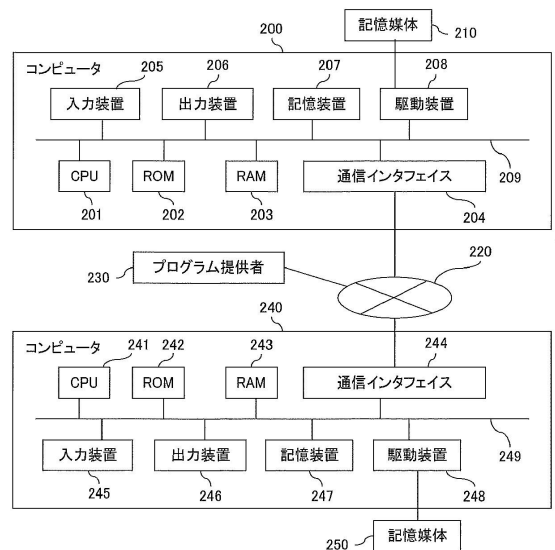
【図 4】

第1実施形態の効果を説明する図 (その2)



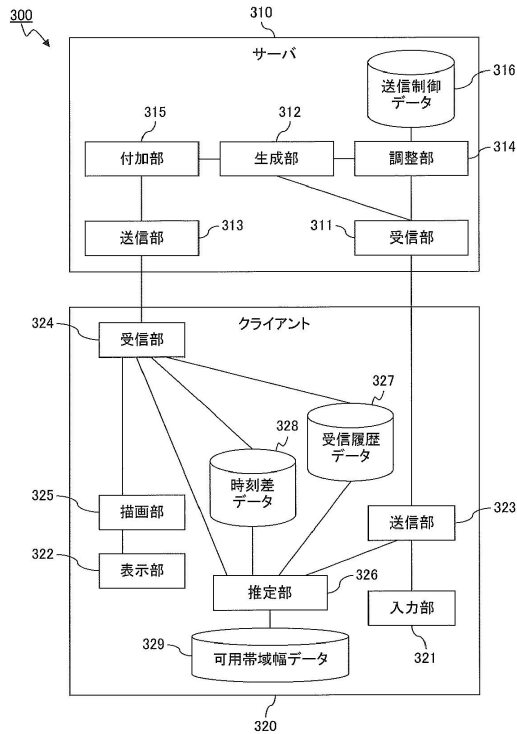
【図 5】

ハードウェア構成例を示す図



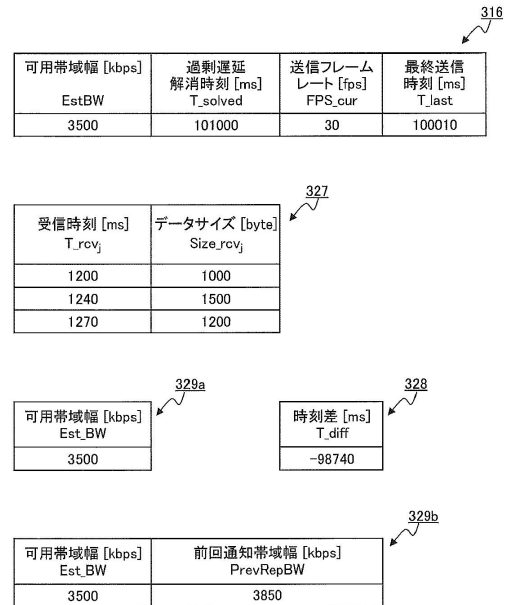
【図 6】

システムのブロック構成図



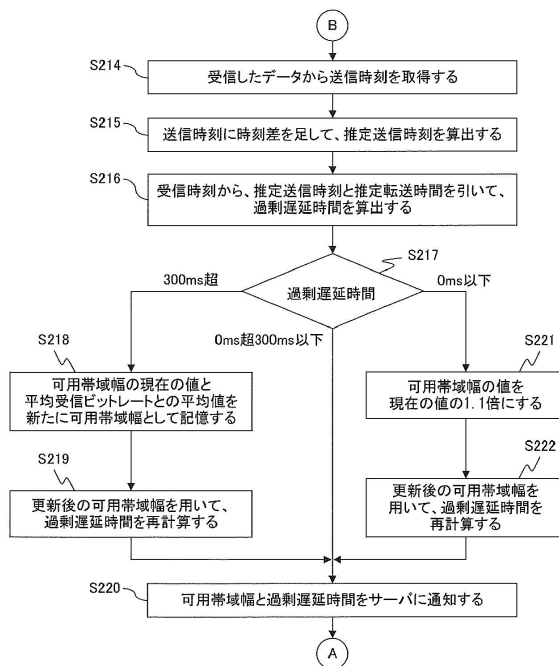
【図 7】

各種データの例を示す図



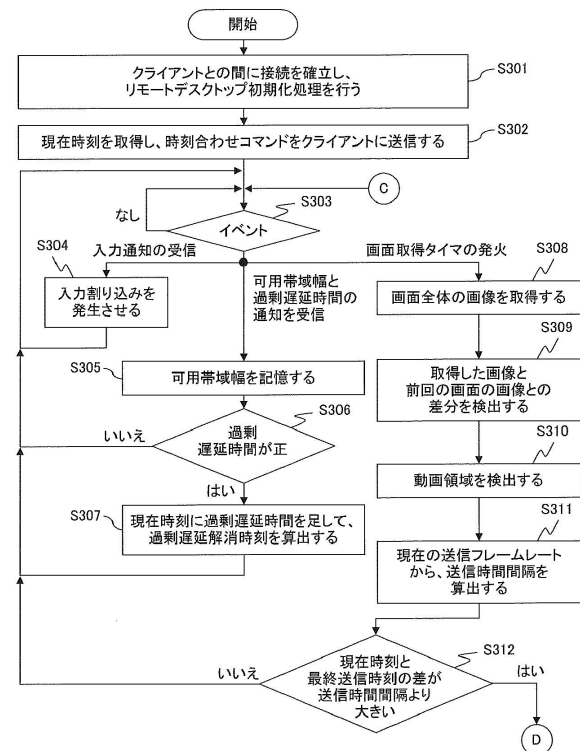
【図 9】

第2実施形態のクライアントの動作フローチャート(その2)



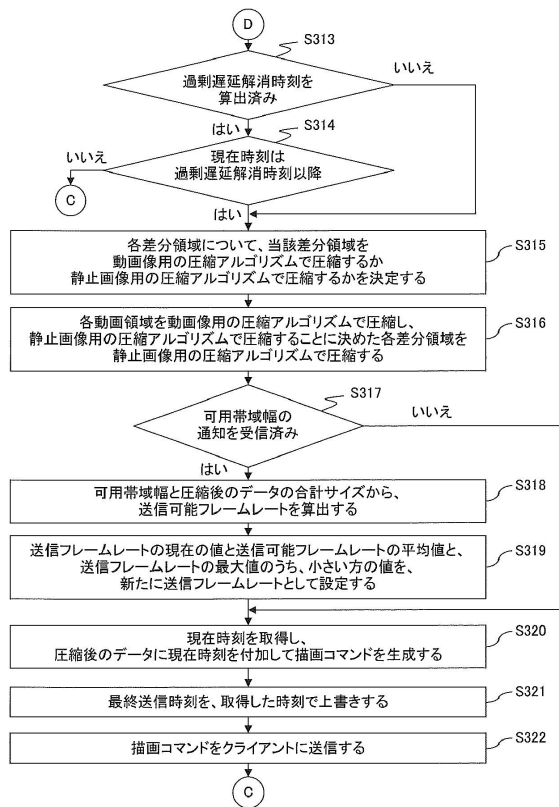
【図 10】

サーバの動作フローチャート(その1)



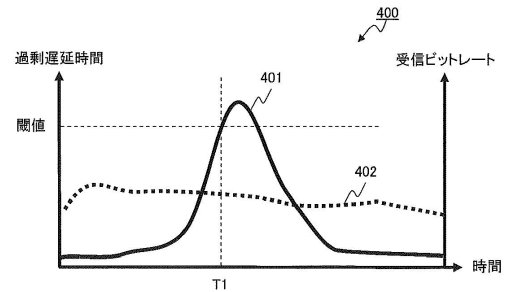
【図 1 1】

サーバの動作フローチャート (その2)



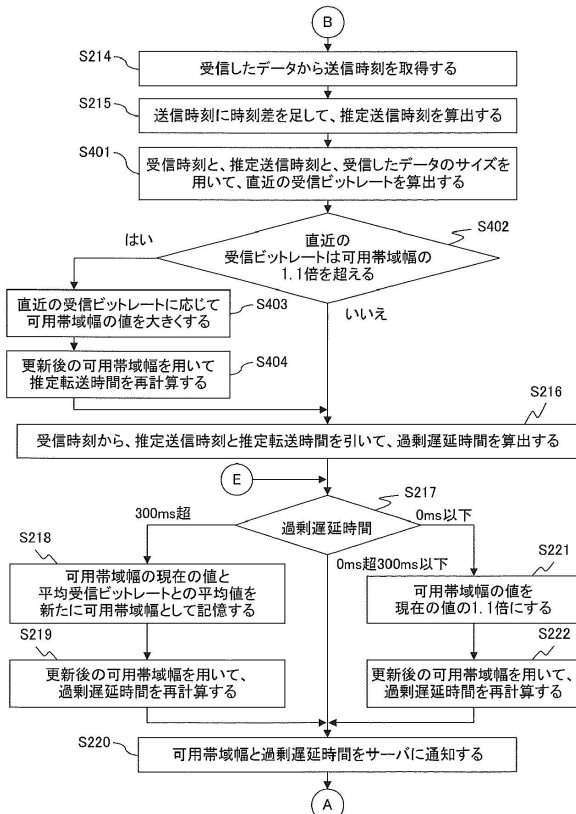
【図 1 2】

過剰遅延時間と受信ビットレートの変化の例を示すグラフ



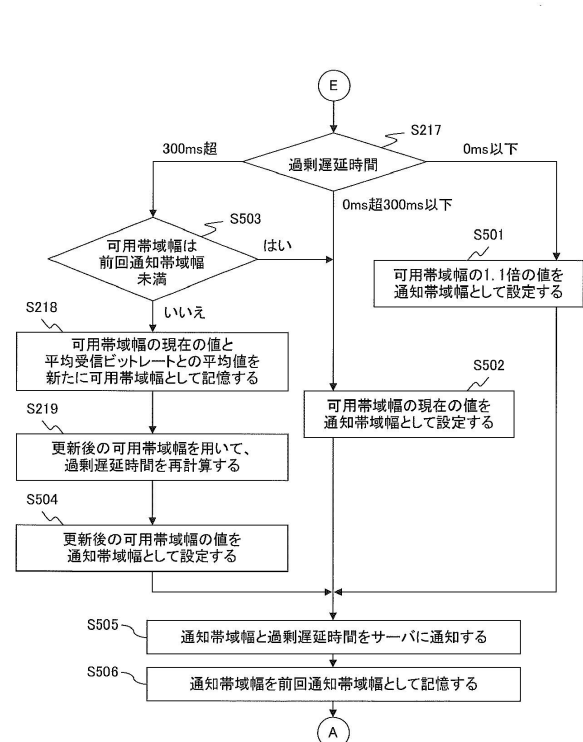
【図 1 3】

第3実施形態のクライアントの動作フローチャート



【図 1 4】

第4実施形態のクライアントの動作フローチャート



フロントページの続き

審査官 小林 義晴

(56)参考文献 特開2008-234389 (JP, A)
国際公開第2011/142311 (WO, A1)

(58)調査した分野(Int.Cl., DB名)
G06F 13/00