

(19) 日本国特許庁(JP)

## (12) 特許公報(B2)

(11) 特許番号

特許第4848425号  
(P4848425)

(45) 発行日 平成23年12月28日(2011.12.28)

(24) 登録日 平成23年10月21日(2011.10.21)

(51) Int.Cl.

HO4L 12/56 (2006.01)

F 1

HO4L 12/56 200Z

請求項の数 10 (全 27 頁)

(21) 出願番号 特願2008-536010 (P2008-536010)  
 (86) (22) 出願日 平成18年10月5日 (2006.10.5)  
 (65) 公表番号 特表2009-513046 (P2009-513046A)  
 (43) 公表日 平成21年3月26日 (2009.3.26)  
 (86) 國際出願番号 PCT/EP2006/067094  
 (87) 國際公開番号 WO2007/045561  
 (87) 國際公開日 平成19年4月26日 (2007.4.26)  
 審査請求日 平成21年6月29日 (2009.6.29)  
 (31) 優先権主張番号 11/256,259  
 (32) 優先日 平成17年10月21日 (2005.10.21)  
 (33) 優先権主張国 米国(US)

(73) 特許権者 390009531  
 インターナショナル・ビジネス・マシーンズ・コーポレーション  
 INTERNATIONAL BUSINESS MACHINES CORPORATION  
 アメリカ合衆国10504 ニューヨーク  
 州 アーモンク ニュー オーチャード  
 ロード  
 (74) 代理人 100108501  
 弁理士 上野 剛史  
 (74) 代理人 100112690  
 弁理士 太佐 種一  
 (74) 代理人 100091568  
 弁理士 市位 嘉宏

最終頁に続く

(54) 【発明の名称】適応型帯域幅制御を行う方法、装置、及びコンピュータ使用可能なコード(適応型帯域幅制御)

## (57) 【特許請求の範囲】

## 【請求項1】

データ転送を適応的に制御するコンピュータ実装方法であつて、  
 選択された優先度を使用してソースから配信されるデータを複数のゲートウェイにおいて受信するステップと、  
 前記選択された優先度を使用して、前記データを前記複数のゲートウェイから複数の受信機に送信するステップと、を含み、

前記複数のゲートウェイは全て、適応型帯域幅制御処理と、前記データが前記選択された優先度で送信されるように前記適応型帯域幅制御処理を制御する各パラメータ・セットと、を有し、

前記選択された優先度で行われる各ゲートウェイからの前記データの送信は、前記複数のゲートウェイのうちの異なるゲートウェイに関する前記パラメータ・セットについて異なる値が設定されている場合には、前記異なるゲートウェイの他のトラフィックに異なる影響を与えることになる、

コンピュータ実装方法。

## 【請求項2】

前記パラメータ・セットは、閾値と、と、最大待ち時間と、を含む、請求項1に記載のコンピュータ実装方法。

## 【請求項3】

前記複数のゲートウェイのうちの特定のゲートウェイに関するパラメータ・セットを構

成するステップを更に含み、前記特定のゲートウェイに関する前記パラメータ・セットは、前記複数のゲートウェイのうちの別のゲートウェイに関する前記パラメータ・セットとは異なる、

請求項 1 に記載のコンピュータ実装方法。

【請求項 4】

前記適応型帯域幅制御処理では、前記複数のゲートウェイのうちのあるゲートウェイから前記配信の受信機までの経路に沿った輻輳が加味される、請求項 1 に記載のコンピュータ実装方法。

【請求項 5】

前記各パラメータ・セットは、前記複数のゲートウェイのうちの各ゲートウェイ毎に異なる、請求項 1 に記載のコンピュータ実装方法。 10

【請求項 6】

前記データは、データ・ファイル、アプリケーション・アップデート、及びウイルス・パッチのうちの 1 つを含む、請求項 1 に記載のコンピュータ実装方法。

【請求項 7】

請求項 1 乃至 6 のいずれか一項に記載の方法の各ステップをコンピュータに実行させるコンピュータ・プログラム。

【請求項 8】

データ処理システムであって、

バスと、

前記バスに接続された通信ユニットと、

前記バスに接続され、1組のコンピュータ使用可能なプログラム・コードを含む記憶デバイスと、

前記バスに接続され、前記 1 組のコンピュータ使用可能なプログラム・コードを実行して請求項 1 乃至 6 のいずれか一項に記載の各ステップを実施する処理ユニットと、を備えるデータ処理システム。

【請求項 9】

カスタマイズ・データ転送をクライアントに提供する方法であって、

ネットワーク特性及びゲートウェイ ID を含むカスタマイズ・データ転送要求を前記クライアントから受信するステップと、 30

それぞれ前記ゲートウェイ ID 及び前記ネットワーク特性に基づいてゲートウェイに割り当てられるカスタム・パラメータ・セットを生成するステップと、

選択された優先度を使用してソースから配信されるデータを複数のゲートウェイにおいて受信するために、コンピュータ使用可能なプログラム・コードと共に使用される前記カスタム・パラメータ・セットを前記クライアントに送信するステップと、

前記選択された優先度を使用して、前記データを前記複数のゲートウェイから複数の受信機に送信するステップと、を含み、

前記複数のゲートウェイは全て、適応型帯域幅制御処理と、前記データが前記選択された優先度で送信されるように前記適応型帯域幅制御処理を制御する各パラメータ・セットと、を有し、 40

前記選択された優先度で行われる各ゲートウェイからの前記データの送信は、前記複数のゲートウェイのうちの異なるゲートウェイに関する前記パラメータ・セットについて異なる値が設定されている場合には、前記異なるゲートウェイの他のトラフィックに異なる影響を与えることになる、

方法。

【請求項 10】

前記送信するステップは、前記コンピュータ使用可能なプログラム・コードと共に前記カスタム・パラメータ・セットを送信するステップを更に含む、請求項 9 に記載の方法。

【発明の詳細な説明】

【技術分野】

10

20

30

40

50

**【0001】**

本発明は一般に改良型データ処理システムに関するものであり、特に、データ転送を行うコンピュータ実装方法及び装置に関するものである。

**【0002】**

より詳細には、本発明は、データ転送に使用される帯域幅を適応的に制御するコンピュータ実装方法、装置、及びコンピュータ使用可能なプログラム・コードに関するものである。

**【背景技術】****【0003】**

ネットワーク及びインターネットの普及は、商用通信に大きな変革をもたらしている。  
10 一般に、ネットワークはデータ転送に使用される。多くの分散型アプリケーションでは、サービス品質を高めるためにバックグラウンド転送が利用されている。これらのタイプのバックグラウンド転送を用いると、ユーザはバックグラウンド転送の完了を待たずに他のアクションを実行することが可能となる。バックグラウンド転送は、例えばデータ・バックアップ、プリフェッヂ、エンタープライズ・データ配信、インターネット・コンテンツ配信、及びピア・ツー・ピア・ストレージを含めた様々なアプリケーション及びサービスで利用されている。上記及び他のタイプのアプリケーションでは、ネットワーク帯域幅の消費量が増加している。こうしたサービスの中には、帯域幅が増加するほどサービス品質が向上し得ることから帯域幅需要が際限なく増加するものもある。

【非特許文献1】Venkataramani等の「TCP Nice: A Mechanism for Background Transfers」、ACM SIGOPS Operating Systems Review、Vol. 36、Issue SI Winter 2002、1~15頁  
20

**【発明の開示】****【発明が解決しようとする課題】****【0004】**

これらのタイプのアプリケーションに関しては、大部分のネットワークにおいてデータ転送に使用可能な帯域幅が限られているという問題がある。アプリケーションの中には重要な機能を実行するものもあればそうでないものもある。典型的には、バックグラウンド転送は重要な機能に入らないが、これによって使用可能な全ての帯域幅が使用されれば、重要なネットワーク・トラフィックの遅延を招く恐れがある。  
30

**【0005】**

ネットワーク状態を自動的に適応させてネットワークへの影響を少なくするために、適応型帯域幅制御が利用されている。現在、様々なアプリケーションで使用される帯域幅量を制御して輻輳を回避する様々な適応型帯域幅制御処理及びアルゴリズムが利用されている。現時点で使用可能な適応型帯域幅制御処理では、サーバ又はクライアント・コンピュータ上のローカル・インターフェース・レベルのネットワーク状態が加味されるが、データ転送時に存在し得る他の状態が加味されることはない。

**【0006】**

したがって、データ転送時の帯域幅使用率を適応的に制御するように改良されたコンピュータ実装方法、装置、及びコンピュータ使用可能なプログラム・コードを有することが有利なはずである。  
40

**【課題を解決するための手段】****【0007】**

本発明は、選択された優先度を使用してソースから配信(distribution)されるデータを複数のゲートウェイにおいて受信するコンピュータ実装方法、装置、及びコンピュータ使用可能なプログラム・コードを提供する。前記データは、前記選択された優先度を使用して前記複数のゲートウェイから複数の受信機に送信される。前記複数のゲートウェイは全て、適応型帯域幅制御処理と、前記データが前記選択された優先度で送信されるように前記適応型帯域幅制御処理を制御する各パラメータ・セットと、を有する。  
50

前記選択された優先度で行われる各ゲートウェイからの前記データの送信は、前記複数のゲートウェイのうちの異なるゲートウェイに関する前記パラメータ・セットについて異なる値が設定されている場合には、前記異なるゲートウェイの他のトラフィックに異なる影響を与えることになる。

#### 【0008】

前記パラメータ・セットは、閾値と、と、最大待ち時間と、を含むことができる。前記複数のゲートウェイのうちの特定のゲートウェイに関するパラメータ・セットを構成することができ、前記特定のゲートウェイに関する前記パラメータ・セットは、前記複数のゲートウェイのうちの別のゲートウェイに関する前記パラメータ・セットとは異なるものとすることができます。前記適応型帯域幅制御処理では、前記複数のゲートウェイのうちのあるゲートウェイから前記配信の受信機までの経路に沿った輻輳が加味される。前記各パラメータ・セットは、前記複数のゲートウェイのうちの各ゲートウェイ毎に異なるものとすることができます。各実例において、前記データは、データ・ファイル、アプリケーション・アップデート、及びウイルス・パッチのうちの1つを含む。

10

#### 【0009】

ここでは単なる例示として、本発明の実施形態（単数又は複数）を添付図面を参照しながら説明する。

#### 【発明を実施するための最良の形態】

#### 【0010】

図1及び図2はそれぞれ、本発明の諸実施形態を実施することが可能なデータ処理環境を例示する図である。図1及び図2は例示的なものに過ぎず、本発明の諸態様又は諸実施形態を実施することが可能な環境を限定することを主張するものではなく、そのように暗示することも意図しているわけではないことを理解していただきたい。図示の環境には、本発明の趣旨及び範囲から逸脱しない限り様々な修正を施すことができる。

20

#### 【0011】

ここで添付図面を参照すると、図1には本発明の諸態様を実施することが可能な複数のデータ処理システムから成るネットワークが示されている。ネットワーク・データ処理システム100は、本発明の諸実施形態を実施することが可能な複数のコンピュータから成るネットワークである。ネットワーク・データ処理システム100はネットワーク102を含んでおり、ネットワーク102は、ネットワーク・データ処理システム100内で互いに接続された様々なデバイスとコンピュータとの間の通信リンクを提供するのに使用される媒体である。ネットワーク102は、有線通信リンク、無線通信リンク、光ファイバ・ケーブル等の接続を含むことができる。

30

#### 【0012】

図示の例において、ネットワーク102には、サーバ104及びサーバ106ならびに記憶ユニット108が接続されている。また、ネットワーク102には、クライアント110、1112、及び1114も接続されている。これらのクライアント110、1112、及び1114は、例えばパーソナル・コンピュータであってもネットワーク・コンピュータであってもよい。図示の例において、サーバ104は、ブート・ファイルのようなデータ、オペレーティング・システム・イメージ、及び各種アプリケーションをクライアント110、1112、及び1114に提供する。本例において、クライアント110、1112、及び1114は、サーバ104のクライアントに相当する。ネットワーク・データ処理システム100は、図示されていない追加的なサーバ、クライアント、及び他のデバイスを含むこともできる。

40

#### 【0013】

図示の例において、ネットワーク・データ処理システム100は、伝送制御プロトコル/インターネット・プロトコル(TCP/IP)プロトコル・スイートを使用して互いに通信し合う世界中のネットワーク及びゲートウェイの集合体を表す、ネットワーク102を有するインターネットである。当該インターネットの中心には、データ及びメッセージを送る数千もの民間用、政府用、教育機関用他のコンピュータ・システムから成る、主要

50

なノード又はホスト・コンピュータ間の高速データ通信回線の基幹回線が存在する。言うまでもなく、ネットワーク・データ処理システム100は、例えばインターネットや、ローカル・エリア・ネットワーク( LAN )や、広域ネットワーク( WAN )等のいくつかの異なるタイプのネットワークとして実施することもできる。図1は一例として意図されたものであり、本発明の様々な実施形態に関するアーキテクチャ上の限定として意図されたものではない。

#### 【0014】

ここで図2を参照すると、本発明の諸態様を実施することが可能なデータ処理システムのブロック図が示されている。データ処理システム200は、本発明の諸実施形態に関する処理を実行するコンピュータ使用可能なコード又は命令を設置することができる、図1のサーバ104やクライアント110等のコンピュータの一例である。10

#### 【0015】

図示の例において、データ処理システム200は、ノース・ブリッジ及びメモリ・コントローラ・ハブ( MCH )202と、サウス・ブリッジ及び入力／出力( I/O )コントローラ・ハブ( ICH )204とを含むハブ・アーキテクチャを利用している。プロセッサ206、メイン・メモリ208、及びグラフィックス・プロセッサ210は、ノース・ブリッジ及びメモリ・コントローラ・ハブ202に連結されている。グラフィックス・プロセッサ210は、高速グラフィックス・ポート( AGP )を介してノース・ブリッジ及びメモリ・コントローラ・ハブ202に連結することができる。20

#### 【0016】

図示の例において、ローカル・エリア・ネットワーク( LAN )アダプタ212は、サウス・ブリッジ及びI/Oコントローラ・ハブ204に連結されている。オーディオ・アダプタ216、キーボード及びマウス・アダプタ220、モデム222、読み取り専用メモリ( ROM )224、ユニバーサル・シリアル・バス( USB )ポート及び他の通信ポート232、ならびにPCI / PCI eデバイス234は、バス238を介してサウス・ブリッジ及びI/Oコントローラ・ハブ204に連結されており、ハード・ディスク・ドライブ( HDD )226及びCD - ROMドライブ230は、バス240を介してサウス・ブリッジ及びI/Oコントローラ・ハブ204に連結されている。PCI / PCI eデバイスとしては、例えばイーサネット・アダプタ、アドイン・カード、及びノート型コンピュータ用のPCカード等を挙げることができる。PCIはカード・バス・コントローラを使用する一方、PCIeはこれを使用しない。ROM224は、例えばフラッシュ・バイナリ入力／出力システム( BIOS )等であってもよい。30

#### 【0017】

ハード・ディスク・ドライブ226及びCD - ROMドライブ230は、バス230を介してサウス・ブリッジ及びI/Oコントローラ・ハブ204に連結されている。ハード・ディスク・ドライブ226及びCD - ROMドライブ230は、例えば統合ドライブ・エレクトロニクス( IDE )やシリアル・アドバンスト・テクノロジー・アタッチメント( SATA )・インターフェース等を使用することができる。スーパーI/O( SIO )デバイス236は、サウス・ブリッジ及びI/Oコントローラ・ハブ204に連結させることができる。オペレーティング・システムは、プロセッサ206上で作動して、図2のデータ処理システム200内の様々なコンポーネントの制御を調整し実行するものである。クライアントとしてのオペレーティング・システムは、Microsoft( R )Windows( R )XPのような市販のオペレーティング・システムであってもよい(「Microsoft」と「Windows」は、マイクロソフト・コーポレーションの米国その他の国、又はその両方における商標である)。Java( TM )プログラミング・システムのようなオブジェクト指向のプログラミング・システムは、上記オペレーティング・システムと連動して作動することができ、データ処理システム200上で実行されているJavaプログラム又はアプリケーションから当該オペレーティング・システムへのコールを行うものである(「Java」は、サン・マイクロシステムズの米国その他の国、又はその両方における商標である)。4050

## 【0018】

サーバとしてのデータ処理システム200は、例えば拡張対話式エグゼクティブ(Advanced Interactive Executive: AIX(R))オペレーティング・システム又はLinuxオペレーティング・システムを実行するIBM eServer(TM) pSeries(R)コンピュータ・システムであってもよい(「eServer」、「pSeries」、及び「AIX」は、インターナショナル・ビジネス・マシーンズ・コーポレーションの米国その他の国、又はその両方における商標であり、「Linux」は、Linus Torvalds氏の米国その他の国、又はその両方における商標である)。データ処理システム200は、複数のプロセッサを処理ユニット206内に含む対称型マルチプロセッサ(SMP)システムであってもよい。別法としてシングル・プロセッサ・システムを採用することもできる。10

## 【0019】

上記オペレーティング・システムに対する命令、上記オブジェクト指向のプログラミング・システム、及びアプリケーション又はプログラムは、ハード・ディスク・ドライブ226のような記憶装置上に設置され、処理ユニット206による実行のためにメイン・メモリ208にロードすることができる。本発明の諸実施形態に関する各処理は、例えばメイン・メモリ208や読み取り専用メモリ224等のメモリ内に設置することも、1つ又は複数の周辺デバイス226及び230内に設置することもできるコンピュータ使用可能なプログラム・コードを使用して、処理ユニット206によって実行される。20

## 【0020】

図1及び図2のハードウェアは実装形態に応じて変更され得ることが当業者には理解されるだろう。図1及び図2に示されるハードウェアに加えて又はその代わりに、フラッシュ・メモリ又は同等の不揮発性メモリ又は光ディスク・ドライブ等、他の内部ハードウェア又は周辺デバイスを使用することもできる。また、本発明の上記処理は、マルチプロセッサ・データ処理システムにも適用することができる。20

## 【0021】

いくつかの実例において、データ処理システム200は、オペレーティング・システム・ファイル又はユーザ生成データあるいはその両方を記憶する不揮発性メモリを提供するフラッシュ・メモリが組み込まれた携帯情報端末(PDA)であってもよい。バス・システムは、図2に示されるバス238やバス240のような1つ又は複数のバスから構成されることもある。言うまでもなく、上記バス・システムは、任意のタイプの通信ファブリック又はアーキテクチャであって、当該ファブリック又はアーキテクチャに取り付けられた様々なコンポーネント又はデバイス間でデータの転送を実施する通信ファブリック又はアーキテクチャを使用して実装することができる。通信ユニットは、図2のモジュール222やネットワーク・アダプタ212等、データの送受信に使用される1つ又は複数のデバイスを含むことができる。メモリは、例えば図2のノース・ブリッジ及びメモリ・コントローラ・ハブ202内で見受けられるようなメイン・メモリ208、読み取り専用メモリ224、キャッシュ等であってもよい。図1及び図2に示される例及び上述の実例は、アーキテクチャ上の限定を暗示することを意図しているわけではない。例えば、データ処理システム200は、PDAの形式をとることに加えて、タブレット・コンピュータ、ラップトップ・コンピュータ、あるいは電話デバイスであってもよい。3040

## 【0022】

本発明は、データ転送時の使用帯域幅を適応的に制御するコンピュータ実装方法、装置、及びコンピュータ使用可能なプログラム・コードを提供する。本発明の諸態様は、現在使用されている適応型帯域幅制御処理がクライアント側以外のネットワーク状態を加味することができない点を認識している。本発明の諸態様は、様々なネットワークが様々な特性を有することがデータ転送に影響を及ぼす可能性がある点を認識している。本発明の諸態様は、そのようなタイプの要因が加味された様々な機構を提供する。本発明の諸態様は、個々のデータ・パケットを適応型帯域幅制御に使用されるネットワーク上にどのように送出するかを制御するパラメータを設定する能力を提供する。50

**【0023】**

本発明の諸態様は、現在使用可能なデータ転送システムではバックグラウンド・モードでの実行速度が非常に低速となるため、実行の完了に非常に長い時間を要する可能性があることを認識している。また、本発明の諸態様は、顧客が適応型機能を所望する一方で配信に長い時間を掛けたくない場合が多いことも認識している。例えば、アプリケーションのアップデートに関しては、ウイルス・パッチ又はウイルス・アップデートの配信が非常に重要なものと見なされている。そのため、本発明の諸態様は、配信単位で優先度を設定する能力を提供する。これらの実例では、「高(h i g h)」、「中(m e d i u m)」、「低(l o w)」の3種類の優先度が設定される。各実例の優先度は、例えば「高」のような選択された優先度に応じて適応型帯域幅制御処理の挙動が変化する、適応型優先度(*adaptive priority*)である。例えば、ソフトウェア・アップデートを「低」優先度として送信し、その後ウイルス・パッチを「高」優先度で送信することができる。したがって、該当する配信物(*d i s t r i b u t i o n*)の重要度に基づいて様々な配信物に様々な優先度を与えることが可能となる。配信物毎に優先度が異なるよう 10 にすることで、適応型帯域幅制御処理の異なるパラメータ・セットが提供され、当該処理の挙動が選択された特定の優先度に応じて変化することになる。

**【0024】**

また、優先度はゲートウェイ単位で様々な形で設定することができる。これらの実例において、ゲートウェイは、1組のデバイスのコンジットとして働くデバイス又はデータ処理システムである。例えば、ゲートウェイは、ローカル・エリア・ネットワーク又は広域ネットワークのポータル又は入口の働きをすることができる。また、ゲートウェイは、無線ネットワークとの間の接続の働きをすることもできる。ゲートウェイは、1組のクライアントを管理するものとして言及されることもある。 20

**【0025】**

各実例はいずれも適応型帯域幅制御を対象としているが、本発明の諸態様は、1つ又は複数のターゲット・データ処理システムへの任意のタイプのバルク・データ転送に適用されてもよい。

**【0026】**

ここで図3を参照すると、本発明の例示的な一実施形態に従って様々な優先度を使用して配信を行うことが可能となる例示的なネットワーク・データ処理システムが示されている。本例において、ネットワーク・データ処理システム300は、ゲートウェイ304、306、及び308に接続されたソース302を含む。ゲートウェイ304は、ルータ318を介したクライアント310、312、314、及び316との間の接続を提供する。ゲートウェイ306は、衛星ルータ328を介したクライアント320、322、324、及び326との間の接続を提供する。ゲートウェイ308は、クライアント330、332、334、及び336との間の接続を提供する。これらの実例において、各クライアントは、無線接続、ダイヤル・アップ接続、ケーブル・モデル、これらに類する他の何らかの接続システム等、様々な手段を介してゲートウェイ308に接続される移動ラップトップ・コンピュータである。 30

**【0027】**

ネットワーク300は、図1のネットワーク・データ処理システム100内に含まれるネットワークの一例に過ぎない。特に、図2のデータ処理システム200と同様のデータ処理システムを使用して様々なクライアント及びゲートウェイを実装することが可能である。

**【0028】**

本例において、ゲートウェイ304は、安全性の高いローカル・エリア・ネットワーク・サーバを管理する。言い換えれば、クライアント310、312、314、及び316は、サーバ・データ処理システムである。ゲートウェイ306は、様々な支店に所在するPOS(point-of-sale:販売時点)システムを管理する。ゲートウェイ308は、移動ラップトップの形をとる各クライアントを管理する。 50

**【0029】**

管理者338は、様々な設定を使用して配信物を送信することができる。これらの実例において、配信(distribution)とは、任意の種類のデータを1つ又は複数のエンドポイント又は受信機に送信することを指す。そのような配信物は、例えばアプリケーション・アップデート、ダイナミック・リンク・ライブラリ・アップデート、ウイルス・パッチ又はウイルス定義アップデート、あるいはデータ・ファイルであることもある。図3の例において、配信物は3種類の優先度レベル、即ち、「高」、「中」、「低」のうちの1つを使用して送信することができる。言うまでもなく、特定の実装形態に応じてこれ以外の数の優先度レベルを実装することもできる。3つの異なる優先度レベルは、本発明の一実施形態を例示するために採用したものである。

10

**【0030】**

したがって、管理者338は、例えばクライアントに配信される1つの配信物に対して「高」のような1つの優先度を付与した上で、当該配信物をゲートウェイ304、306、及び308に送信することができる。次に、管理者338は、例えば「中」のような異なる優先度を後続の配信物に付与した上で、当該配信物をゲートウェイ304、306、及び308に送信することができる。このようにして、特定の配信物に関連する重要度に基づいて、受信機への転送に関する様々な優先度を様々な配信物に与えることができる。したがって、重要度の高い配信物ほどクライアント側に到達するタイミングを早めることができるが、このようなタイプの配信では使用する帯域幅が多くなる。優先度の低い配信物ほど使用する帯域幅が少なくなるが、このようなタイプの配信では配信に要する時間が長くなる。

20

**【0031】**

更に、本発明の諸態様は、選択された優先度レベルに応じてゲートウェイ毎に異なる量の帯域幅を使用する能力を、選択される優先度毎に提供する。言い換れば、2つのゲートウェイが同一のネットワーク状態にあると考えた場合に、それらのゲートウェイは、選択された優先度レベル毎に異なる量の帯域幅を使用してデータ転送を行うことができる。これらの実例では、様々なクライアントに対する配信物の送信を行うゲートウェイ上で実行される適応型帯域幅制御処理のパラメータを調整することによって、選択される優先度レベルの様々な効果がゲートウェイ単位で調整される。

30

**【0032】**

例えば、ゲートウェイ304が「高」優先度の配信物をクライアント310、312、314、及び316に送信する際は、帯域幅の70%を使用することができる。ゲートウェイ306が「高」優先度の同一の配信物をクライアント320、322、324、及び326に転送する際は、帯域幅の50%を使用することができる。ゲートウェイ308が「高」優先度の同一の配信物をクライアント330、332、334、及び336に送信する際は、帯域幅の30%まで使用することができる。

**【0033】**

ゲートウェイからアクセスされる特定のネットワーク内の帯域幅使用率に対する実際の影響は、当該ゲートウェイ上で実行される適応型帯域幅制御処理のパラメータを調整することによって調整される。各ゲートウェイは、それらが最初にセット・アップされる時点で事前構成することができる。また、上記の各パラメータは、管理者338によって特定されたネットワーク内の変更あるいは他の変更に基づいて変更することもできる。これらの変更は、後述するユーザ定義設定を介して管理することができる。

40

**【0034】**

ここで図4を参照すると、本発明の例示的な一実施形態に係る適応型帯域幅制御で使用される各コンポーネントが示されている。この例では、適応型送信機(adaptive sender)400がデータを受信機402、404、及び406に送信する。図4の例において、適応型送信機400は、図3のゲートウェイ304のようなゲートウェイとすることができる。特に、適応型送信機400は、図2のデータ処理システム200内で見受けられるようなハードウェアを使用する図1のサーバ106として実装することが

50

できる。受信機 402、404、及び 406 は、図 3 のクライアント 310、312、314 のようなクライアントとすることができる。これらの受信機は、図 2 のデータ処理システム 200 のようなデータ処理システムを使用して実装することもできる。データ送信は、パケット 408 をルータ 410 のようなルーティング機構に送信することによって行われる。ルータ 410 は、パケット 408 内で発見された経路指定データに基づいてパケット 408 を該当する受信機に経路指定し又は送信する働きをするデバイスである。ルータ 410 が過剰な数のパケットの処理を強いられる場合、そのようなルータ 410 をバックログ・ルータと呼ぶ。言い換えれば、バックログ・ルータとは、それ自体のキューに所在するパケットのうちで最も負荷の大きいパケット、即ち最大数のパケットを有するルータを指す。送信機と受信機との間にルータが介在する場合もあれば介在しない場合もある。送信機はパケットの送信元であり、受信機は肯定応答の送信元である。各受信機はパケット 408 を受信すると、肯定応答 412 を適応型送信機 400 に返す。本例において、肯定応答は TCP / IP 通信の一部となる。

#### 【0035】

これらの実例では、適応型送信機 400 を図 1 のサーバ 104 のようなデータ処理システム内の一処理として実施することができる。受信機 402、404、及び 406 は、図 1 のクライアント 108、110、112 のような受信デバイス上で実行することができる。特に、これらの様々な処理は、図 2 のデータ処理システム 200 のようなデータ処理システム上で実施することができる。

#### 【0036】

適応型送信機 400 は、各受信機に送信されたパケット 408 を追跡する。また、肯定応答 412 の受信も追跡され、この情報はラウンド・トリップ時間等のパラメータを特定するのに使用される。ラウンド・トリップ時間とは、パケットが送信されてから肯定応答が受信されるまでの時間を指す。本例において、ラウンド・トリップ時間は適応型送信機 400 の見通しに基づいている。また、この情報はロックを特定する際にも使用される。ロックは任意のパケットが送信される時点を開始点とし、例えば、あるロック内の全てのパケットに関する統計データは、当該ロックの開始点となった最初のパケットに対する肯定応答が受信機から返されるまで保持される。任意のパケットに対する肯定応答が返されたときに、適応型送信機 400 は、複数のパケットから成る当該ロックを対象とする統計値を計算する。言い換えれば、当該ロックの開始点に位置する任意のパケットに対する肯定応答が返されるまでにパケットがいくつ送信されたかに応じて、1 つ又は複数のパケットをロック内に所在させることができる。また、適応型送信機 400 はウィンドウの特定も行う。ウィンドウとは、肯定応答が受信されるまでにネットワーク内で送信されたパケット数を指す。

#### 【0037】

更に、適応型送信機 400 は、ルータ 410 内に所在するパケット数の推定も行う。当該情報は現在のラウンド・トリップ時間を使用して判定され、これにより、肯定応答が返されないことが予想されたパケット数と、肯定応答が実際に返されなかったパケット数との差が計算される。例えば、現在のラウンド・トリップ時間に照らしてネットワーク上に 5 つのパケットが存在することが考えられる場合に、8 つのパケットの肯定応答が返されていないことが特定された場合、適応型送信機 400 は、ルータ 410 上に 3 つのパケットが所在するものと推定することができる。

#### 【0038】

適応型送信機 400 は、ルータ 410 上の選択されたパケット数を維持しようと試みながら、予想されるウィンドウ・サイズを増減することによって速度を制御する。ウィンドウが大きくなるほどルータ 410 が他のネットワーク・トラフィックの処理よりも適応型パケットの処理に費やす時間が増加するため、大きいウィンドウの方がアグレッシブ (aggressive) である。このようにして、適応型送信機 400 は、ルータ 410 に関するパラメータ及びパラメータを調整することができる。は、その数に達した時点でウィンドウが 1 パケット分拡大される、ルータ 410 内の接続単位の最大パケット数

10

20

30

40

50

を示す整数である。 は、その数に達した時点でウィンドウが 1 パケット分縮小される、バックログ・ルータ内の接続単位の最大パケット数を示す整数である。これらのパラメータを用いると、ウィンドウを 及び に基づいて調整することにより速度増減幅を小さくすることが可能となる。別のパラメータ及び閾値を用いると、ブロック内の半数のパケットが当該閾値の条件を満たした時点で速度を減少させることが可能となる。本例において、閾値は、ベース・ラウンド・トリップ時間から最大ラウンド・トリップ時間までの時間に関して構成可能な時間率である。

#### 【 0 0 3 9 】

ここで図 5 を参照すると、本発明の例示的な一実施形態に係る伝送制御プロトコル／インターネット・プロトコル（TCP／IP）及び同様のプロトコルが示されている。TCP／IP 及び同様のプロトコルは、通信アーキテクチャ 500 によって利用される。本例において、通信アーキテクチャ 500 は、4 レイヤ・システムである。当該アーキテクチャは、アプリケーション・レイヤ 502 と、トランポート・レイヤ 504 と、ネットワーク・レイヤ 506 と、リンク・レイヤ 508 と、を含む。各レイヤは、様々な通信タスクの処理を担当する。リンク・レイヤ 508 は、データ・リンク・レイヤ又はネットワーク・インターフェース・レイヤとも呼ばれ、通常、デバイス・ドライバをオペレーティング・システム内に含み、対応するネットワーク・インターフェースをコンピュータ内に含んでいる。当該レイヤは、使用されている光ケーブルやイーサネット・ケーブル等のネットワーク媒体との物理インターフェースに関する全てのハードウェア詳細を取り扱うレイヤである。

#### 【 0 0 4 0 】

ネットワーク・レイヤ 506 は、インターネット・レイヤとも呼ばれ、ネットワーク中のデータ・パケットの移動を処理するレイヤである。例えば、ネットワーク・レイヤ 506 は、当該ネットワークを介して転送される様々なデータ・パケットの経路指定を処理する。TCP／IP スイートのネットワーク・レイヤ 506 は、インターネット・プロトコル（IP）、インターネット制御メッセージ・プロトコル（ICMP）、及びインターネット・グループ管理プロトコル（IGMP）を含めたいくつかのプロトコルから構成される。

#### 【 0 0 4 1 】

次に、トランポート・レイヤ 504 は、2 つのホスト・コンピュータ間のデータ転送を容易にする、ネットワーク・レイヤ 506 とアプリケーション・レイヤ 502 との間のインターフェースを提供する。トランポート・レイヤ 504 は、例えばアプリケーションからそれ自体に渡されたデータを下位のネットワーク・レイヤに適したサイズのチャンクに分割することや、受信パケットの肯定応答を返すこと、送信されたパケットを相手方が受け取ったことを確認するためのタイムアウトを設定すること、等に関与するレイヤである。TCP／IP プロトコル・スイートには、明確に異なる 2 つのトランスポート・プロトコル、即ち、TCP と、ユーザ・データグラム・プロトコル（UDP）とが存在する。TCP は、ドロップアウト検出及び再送信サービスを含めた信頼性サービスを提供することにより、データが 2 つのホスト間で適切に伝送されることを保証するものである。

#### 【 0 0 4 2 】

これとは逆に、UDP は、データが適切に転送されることを保証する機構を何ら提供することなく、単に一方のホストから他方のホストにデータグラムと呼ばれるデータ・パケットを送信するだけという、TCP よりもずっと単純なサービスをアプリケーション・レイヤに提供するものである。UDP が使用される場合には、アプリケーション・レイヤは信頼性機能を実行しなければならない。

#### 【 0 0 4 3 】

アプリケーション・レイヤ 502 は、特定のアプリケーションの詳細を取り扱う。ほぼ全ての実装形態に関して、リモート・ログイン向けの Telnet、ファイル転送プロトコル（FTP）、電子メール向けの簡易メール転送プロトコル（SMT P）、及び簡易ネットワーク管理プロトコル（SNMP）を含めた多くの一般的な TCP／IP の応用例が

10

20

30

40

50

存在する。

#### 【0044】

本発明の諸態様はアプリケーション・レイヤ502において実装され、これにより、ユーザがユーザ設定を入力することができるようデータ転送を適応的に制御することが可能となる。このようにして、ユーザは様々な設定を変更することにより、適応型帯域幅制御で使用されるネットワーク上にパケットをどのように送出するかを制御することができる。ユーザ設定をアプリケーション・レベルから入力できるようにすることで、ユーザは、様々なタイプのネットワークに変更を施すことができ、また、典型的には適応型帯域幅制御処理で加味されることがない様々なネットワーク状態に変更を施すこともできる。このようにして、クライアント側以外の状態をも加味することができる。例えば、本発明の諸態様を用いると、ユーザは、受信機までの経路内のホップ数や、受信機までの経路内のリンクのうち、トラフィック量が多いリンク又は輻輳状態に陥っているリンク等の各種要因を特定することに基づいて設定を変更することが可能となる。10

#### 【0045】

本発明の各実例はアプリケーション・レベルで実装されているが、他のレイヤに様々な処理を実装することもできる。例えば、本発明の諸態様は、特定の実装形態に応じてトランポート・レイヤ504内に実装することもネットワーク・レイヤ506内に実装することもできる。

#### 【0046】

ここで図6を参照すると、本発明の例示的な一実施形態に係る適応型帯域幅制御を実現する際に使用される各ソフトウェア処理及びコンポーネントが示されている。本例において、適応型送信機600は、図4の適応型送信機400内の処理を詳細に示したものである。図6の例において、適応型送信機600は3つのスレッドを含む。送信(send)スレッド602、輻輳制御(congestion control)スレッド604、及びパケット・スニッフィング・スレッド(packet sniffing)606はそれぞれ、データを1つ又は複数の受信機に適応的に送信するのに使用されるコンポーネントである。送信スレッド602は、ソケット・コールを介してデータを送信するのに使用される。ソケットとは、TCP/IPスタック内のTCP/IPプロトコル等のネットワーク・プロトコルにアプリケーションを接続するソフトウェア・オブジェクトを指す。20

#### 【0047】

送信スレッド602は、要求608を輻輳制御スレッド604に送信して、送信スレッド602によって送信可能なデータ量を問い合わせる。輻輹制御スレッド604は、応答610を返すことによって送信可能なデータ量を知らせる。送信スレッド602は、応答610を使用して送信対象のパケット612を送信する。これらの各パケットは、バックログ・ルータによってそれぞれの宛先に経路指定されるまでバックログ・ルータ・キュー614に格納される。バックログ・ルータ・キュー614は、図4のルータ410のようなバックログ・ルータ上に所在する。各パケットが受信されたときは肯定応答616が適応型送信機600に返される。30

#### 【0048】

輻輹制御スレッド604は、適応型帯域幅制御処理を実行することによって送信すべきデータ量を特定する。これらの実例において、輻輹制御スレッド604は、ネットワーク状態の変化に応じて適応型帯域幅制御処理を使用して、ネットワークを介したデータ・パケット伝送の伝送レートを適応的に特定する。これらのネットワーク状態としては、例えば適応型送信機によって処理されているデータ送信ならびに他の様々なデータ送信に起因する、ネットワーク上の輻輹量が挙げられる。言い換えば、データ送信を行うことが可能な転送速度は、ネットワーク状態に応じて変化する。輻輹制御スレッド604は、ユーザによって設定される様々なパラメータに基づいて、ネットワークへの影響を最小限に抑えるために加減される様々な転送速度を特定する。40

#### 【0049】

以下ではネットワーク状態の一例を図7を使用して説明する。FTPサーバ752がデ50

ータを F T P クライアント 7 5 4 に送信する前は、ルータ 7 4 6 はアイドル状態にある。この状況下では、適応型送信機 7 0 0 が非常に高い転送速度で受信機又はエンドポイントへの送信を行うことが可能である。しかしながら、F T P サーバ 7 5 2 が F T P クライアント 7 5 4 へのデータ送信を開始するとすぐに、ルータ 7 4 6 がデータで氾濫してしまう。その後、適応型帯域幅制御処理では、データの優先度に応じてこの状況に対処する。「低」優先度の場合には、適応型帯域幅制御処理の速度は殆ど送信を行うことができない程度まで低下し、これによって F T P 配信への影響が最小限に抑えられる。「高」優先度の場合には、適応型帯域幅制御処理の速度が大幅に増加するが、それによって F T P 配信の速度は低下することになる。

## 【 0 0 5 0 】

以下では、適応型配信 (adaptive distribution) によってネットワーク状態に対処するための別の手法を例示する。ある銀行は、当該銀行内の 10 個のシステムを対象としたネットワーク接続を管理する単一のルータを含んでいる。当該銀行は、当該ルータを介してセントラル・サイト (全国の 300 店の銀行を管理する) に接続されている。午前 6 時 (当該銀行の営業開始時間前) の時点では銀行内に誰もおらず、ネットワーク (特に当該銀行の単一のルータ) はアイドル状態にある。この間に、セントラル・サイトからは大容量の適応型配信物が「低」優先度で送信される。この配信物のトラフィックは当該銀行のルータ上に限られるため、配信速度はルータの帯域幅の 100 % まで高められる。この配信は、顧客が当該銀行の A T M 及び融資業務システム (loan processing system) の使用を開始することに伴って単一のルータとのネットワーク共有が発生することになる午前 8 時まで続けられる。適応型配信では、低速リンク (このシナリオでは当該銀行のルータ) を横切る追加的なトラフィックの存在が直ちに認識される。本例の適応型配信は「低」優先度で行われているため、適応型配信の速度をルータの非常に少ないパーセンテージだけを使用するように低下させることで、A T M 及び融資業務トラフィックへの影響が最小限に抑えられる。本例の適応型配信は午後 5 時の時点でもまだ続けられる。融資業務トラフィックは当該銀行の営業終了と同時に減少する。本例の適応型配信でネットワーク負荷の減少が感知されると、より多くの帯域幅が使用可能となることから配信速度も増加することになる。

## 【 0 0 5 1 】

再び図 6 を参照すると、上記の処理はキュー 6 1 8 内に所在する情報を使用して行われる。特に、パケット・スニッフィング・スレッド 6 0 6 は、ネットワークからのパケット及び肯定応答をグラブし、パケット情報 6 2 0 及び肯定応答情報 6 2 2 をキュー 6 1 8 に入れる。パケット・スニッフィング・スレッド 6 0 6 は、パケット及び肯定応答のフィルタリングを行って輻輳制御スレッド 6 0 4 から必要とされている適切なデータをキュー 6 1 8 に入る。別法として、輻輳制御スレッド 6 0 4 による処理のために全てのパケット及び肯定応答をキュー 6 1 8 に入れることもできる。キュー 6 1 8 内の情報は、パケットが送信されてから当該パケットの肯定応答が受信されるまでのラウンド・トリップ時間のようなパラメータを特定するために輻輳制御スレッド 6 0 4 によって使用される。キュー 6 1 8 内の情報から輻輳制御スレッド 6 0 4 によって特定され維持され得る他のパラメータとしては、ブロック、ウィンドウ、及びバックログ・ルータ内のパケットが挙げられる。

## 【 0 0 5 2 】

更に、本発明の諸態様は、ユーザ定義パラメータ 6 2 6 を設定するのに使用され得るユーザ・インターフェース 6 2 4 も含む。ユーザ・インターフェース 6 2 4 は、パラメータ・スレッド 6 2 8 によって提供される。典型的には、各パラメータのセット・アップは、適応型処理がシステム上で最初にセット・アップされるときに行われる。これらのパラメータは、ユーザ・インターフェース 6 2 4 及びパラメータ・スレッド 6 2 8 を介して変更することができる。ユーザ定義パラメータ 6 2 6 は、ディスク等の不揮発性メモリに記憶される。当該パラメータは、後に適応型帯域幅処理が開始されるときに読み出すことができる。図 6 の例において、当該パラメータは、優先度毎に値が異なるパラメータ・セット

10

20

30

40

50

を含む。これらの実例において、配信物の優先度レベルに基づいて変更される各パラメータ・セットは、閾値、データ、及び最大待ち時間である。ユーザは優先度レベル毎に異なる値を入力することができるが、適応型送信機 600 によって使用されるパラメータを直接設定することもできる。ユーザ・インターフェース 624 を用いると、ユーザは、適応型帯域幅制御処理を実行する際に輻輳制御スレッド 604 によって使用される様々なパラメータを選択し、入力することが可能となる。

#### 【0053】

これらの実例において、本発明の諸態様は、5つのパラメータの値を入力又は選択することを可能にする。これらの5つのパラメータは、閾値、ラウンド・トリップ時間の最大変化 (round trip time maximum change) 、ラウンド・トリップ時間の平滑化 (round trip time smoothing) 、及び最大待ち時間に関するパラメータを含む。ラウンド・トリップ時間の最大変化は、事前に測定されたラウンド・トリップ時間と比較して、現時点で測定されるラウンド・トリップ時間の変化量を制限するのに使用されるパラメータである。この処理では、現在のラウンド・トリップ時間、最大ラウンド・トリップ時間、及びベース・ラウンド・トリップ時間が追跡される。

#### 【0054】

輻輳制御スレッド 604 は、ラウンド・トリップ時間をソケット単位で連続的に追跡する。1つ目のパケットのラウンド・トリップ時間を「10」とし、2つ目のパケットのラウンド・トリップ時間を「20」とした場合、当該2つ目のパケットのラウンド・トリップ時間は、ラウンド・トリップ時間の最大変化の値「1.5」によって制限される。実際のラウンド・トリップ時間が「20」であっても、ラウンド・トリップ時間の値は「15」( $10 \times 1.5$ )として記録される。その後、後続のパケットのラウンド・トリップ時間は、最大で  $1.5 \times 15$  (第2のパケットに関して記録されたラウンド・トリップ時間)となる。

#### 【0055】

ネットワークは、典型的にはある程度のランダム性を有する。当該パラメータを用いると、過剰に大きい又は過剰に小さいラウンド・トリップ時間を適応型帯域幅制御処理で無視することが可能となる一方で、大きいラウンド・トリップ時間が十分に高い頻度で発生する場合には、それらを許容することも可能となる。当該パラメータを用いると、ユーザは許容される変更の程度を構成することが可能となる。

#### 【0056】

これらの実例におけるラウンド・トリップ時間の平滑化パラメータは、ラウンド・トリップ時間の測定値を指数関数的減衰によってどの程度平滑化すべきかを示す上で使用される整数である。指数関数的減衰は、事前測定値の平均をとることによって実行される。当該パラメータは、ラウンド・トリップ時間の変動の過補償によって適応型帯域幅制御処理が不規則な挙動を示す場合に役立つ可能性がある。平滑化は、典型的には最大変更パラメータが加味される前に行われる。

#### 【0057】

最大待ち時間は、接続を断念して再設定を行う前に当該接続が送信待機状態におかれるラウンド・トリップ時間が、最大ラウンド・トリップ時間の何倍に相当するかを示すパラメータである。肯定応答はネットワーク上で失われることがあり、それによって送信機が追加的なデータを送信するまでに長時間待たされることもある。リセット値は、データが送信されるまでに時間がかかりすぎた場合に適応型帯域幅制御処理をリセットする値である。その閾値パラメータに関しては、パケットのラウンド・トリップ時間がソケット上に送信される場合、当該ラウンド・トリップ時間は選択された範囲内に収まる傾向がある。この状況は特にネットワークがアイドル状態にあるときにあてはまる。

#### 【0058】

当該閾値は、最小ラウンド・トリップ時間と最大ラウンド・トリップ時間との差のパーセンテージである。例えば、閾値が 20 パーセントであれば、当該閾値によって、ベース

10

20

30

40

50

・ラウンド・トリップ時間と最大ラウンド・トリップ時間との差が20パーセントであることが示される。ラウンド・トリップ時間は、ネットワークのタイプに応じて変動が大きくなることも小さくなることもある。典型的には、様々な変動によって様々な閾値の値が必要となる。例えば、良好な挙動のローカル・エリア・ネットワークのラウンド・トリップ時間の変動は非常に小さく、その結果閾値の値を低くすることが可能となる。これに比べ、広域ネットワークのラウンド・トリップ時間の変動は大きい。このタイプのネットワークでは、より高い閾値が必要とされる。このような特定のパラメータは、データ送信の際に遭遇する可能性がある様々なタイプのネットワークを加味するようにユーザ構成可能なパラメータである。その閾値パラメータに関しては、パケットのラウンド・トリップ時間がソケット上に送信される場合、当該ラウンド・トリップ時間は選択された範囲内に収まる傾向がある。この状況は特にネットワークがアイドル状態にあるときにはある。

#### 【0059】

パケットのラウンド・トリップ時間が閾値の値よりも低速である場合には、当該パケットは低速と見なされる。ロック内のパケットの50%が低速と見なされた場合には、ウインドウを半分に分割することによって配信速度を大幅に減少させることができる。したがって、上記の閾値の値に関するユーザ構成を可能にすることによって、様々なネットワーク・タイプ及びネットワーク状態を加味することが可能となる。

#### 【0060】

本発明の諸態様は、各実例の3つのパラメータを修正することによって優先度を変更する。各実例において、優先度毎に修正されるパラメータは、閾値、  
、及び最大待ち時間である。

#### 【0061】

ここで図7を参照すると、本発明の例示的な一実施形態に係る優先度を使用して、配信物を適応型送信機から受信機に送信するネットワークが示されている。本例において、適応型送信機700は、ネットワーク702を介してクライアント704、706、708、710、712、714、716、718、720、722、724、726、728、730、732、734、736、738、740、及び742に配信物を送信する。750、748、746、及び744はルータであり、746はバックログ・ルータである。上記の各クライアントはルータ744に接続され、ルータ744はルータ746に接続されている。ルータ746はルータ748に接続されている。適応型送信機700は、サーバ750を介してネットワーク702に接続されている。本例において、ネットワーク702は20回のホップを含む。ファイル転送プロトコル(FTP)サーバ752は、配信物と共にデータをFTPクライアント754に送信することができる。本例において、FTPサーバ752は、配信物が適応型送信機700から各クライアントに送信されている間に、データをFTPクライアント754に送信する。本実例では、FTPサーバによって高需要のトラフィックが生成され、その結果、ルータ746のリンクが氾濫することになる。優先度を設定することにより、適応型送信機700は、配信物を各クライアントに送信する際に様々な帯域幅量を使用することができるようになる。

#### 【0062】

ここで図8を参照すると、本発明の例示的な一実施形態に係るネットワーク内の帯域幅使用率が示されている。図8のグラフは、図7に示されるようなネットワークのデータ転送で使用されるネットワーク帯域幅のパーセンテージを示している。本例において、線802は、様々なタイプの配信に使用される帯域幅量を示している。セクション804では、図7のFTPサーバ752からFTPクライアント754へのFTP転送だけが示されている。この配信では帯域幅の100%が使用されている。図7の適応型送信機700から各クライアントへの配信に関して「低」優先度設定が用いられる場合には、図7のルータ744は、セクション806に示されるように帯域幅の約80%をFTP転送用に割り当てる。「中」優先度が用いられる場合には、セクション808に示されるように帯域幅の約50%がFTP転送に使用される。図示のとおり、優先度が高くなるほど適応型送信機による配信に対して割り当てられる帯域幅が増加するため、FTP転送に割り当てられ

10

20

30

40

50

る帯域幅は減少する。セクション 810 では、各クライアントへの配信に「高」優先度が与えられる。図示のとおり、ここでの F T P 転送は、帯域幅の 30 % 前後のレベルにまで落ち込んでいる。セクション 812 では、適応型帯域幅制御処理が使用されていない。したがって、この特定のセクションでは本明細書に記載の輻輳制御が使用されないことになる。

#### 【0063】

ここで図 9 を参照すると、本発明の例示的な一実施形態に従ってパラメータに関するユーザ設定を構成する処理のフローチャートが示されている。図 9 に示される処理は、図 6 のパラメータ・スレッド 628において実行することができる。この処理を使用すると、ユーザは、適応型帯域幅制御で使用されるパラメータを定義又は変更することが可能となる。これらの実例において、パラメータは、閾値、ラウンド・トリップ時間の最大変化、ラウンド・トリップ時間の平滑化、最大待ち時間、及びである。特に、図 9 に示される処理は、「高」、「中」、「低」等の様々な優先度レベルで使用される様々なゲートウェイに関するパラメータを設定するのに使用することができる。これらの実例において、「高」優先度では を「7」に、閾値を「99」に、最大待ち時間を「20」に設定することができる。「中」優先度では ベータを「5」に、閾値を「40」に、最大待ち時間を「30」に設定することができる。「低」優先度では を「3」に、閾値を「25」に、最大待ち時間を「40」に設定することができる。これらの特定の設定は、それぞれの優先度レベルに応じて使用される。各設定は、提供されるユーザ・インターフェースを介して設定することができる。10 20

#### 【0064】

更に、本発明の諸態様では、各実例におけるいずれかのルータ又はサーバあるいはゲートウェイからエンドポイントに対する影響も加味される。このような能力は、適応型帯域幅制御処理でラウンド・トリップ時間を使用することによって実現される。

#### 【0065】

図 9 の処理は、ユーザ設定を提示することから開始する(ステップ 900)。これらの設定は、図 6 のユーザ・インターフェース 624 のようなユーザ・インターフェース内で提示することができる。その後、ユーザ入力が受信され(ステップ 902)、当該ユーザ入力によってパラメータ設定が変更されたかどうかが判定される(ステップ 904)。当該ユーザ入力によって設定が変更されている場合には、古い設定が新しい設定に置き換える(ステップ 906)。その後、ステップ 900 に戻って各設定がユーザに提示される。30

#### 【0066】

再びステップ 904 を参照すると、ユーザ入力によって設定が変更されていない場合には、当該ユーザ入力がユーザ設定の変更処理を終了するものであるかどうかが判定される(ステップ 908)。当該ユーザ入力が変更処理を終了するものである場合には、ステップ 900 に戻る。そうでない場合には、各ユーザ設定が保存された後(ステップ 910)、図 9 の処理は終了する。これらの実例において、各設定は図 6 のユーザ定義パラメータ 626 として保存される。

#### 【0067】

次に図 10 を参照すると、本発明の例示的な一実施形態に係る送信スレッドの処理に関するフローチャートが示されている。図 10 に示される処理は、図 6 の送信スレッド 602 のような送信スレッドにおいて実行することができる。40

#### 【0068】

図 10 の処理は、輻輳制御スレッド 604 に送信可能なデータ量の確認要求を送信することから開始する(ステップ 1000)。その後、応答が受信される(ステップ 1002)。当該応答は、輻輳制御スレッドによって実行される適応型帯域幅制御処理の結果として送信可能なデータ量を含む。当該応答が受信されたことに応じて、許容可能なデータ量だけを送信するよう求めるコールがソケットに対して送信される(ステップ 1004)。その後、送信すべきデータがそれ以上存在するかどうかが判定される(ステップ 1006)50

)。データが存在する場合には、ステップ1000に戻る。データが存在しない場合には、図10の処理は終了する。

#### 【0069】

次に図11を参照すると、本発明の例示的な一実施形態に係るパケット・スニッフィング・スレッドの処理に関するフローチャートが示されている。図11に示される処理は、図6のパケット・スニッフィング・スレッド606において実行することができる。

#### 【0070】

図11の処理は、パケットが検出されたかどうかを判定することから開始する(ステップ1100)。パケットが検出された場合には、当該パケットからパケット識別子及びタイムスタンプが抽出される(ステップ1102)。その後、当該データがキューに格納される(ステップ1104)。当該キューは、輻輳制御スレッドがラウンド・トリップ時間を判定し適応型帯域幅制御処理を実行する際に上記データを使用することができるよう、輻輳制御スレッドからアクセス可能な状態になっている。

#### 【0071】

再びステップ1100を参照すると、パケットが検出されない場合には、肯定応答が検出されたかどうかが判定される(ステップ1106)。肯定応答が検出されない場合には、ステップ1100に戻る。ステップ1106で肯定応答が検出された場合には、当該肯定応答のパケット識別子及びタイムスタンプが抽出される(ステップ1108)。その後、上述のステップ1104に進む。

#### 【0072】

ここで図12を参照すると、本発明の例示的な一実施形態に係る輻輳制御スレッドの処理に関するフローチャートが示されている。本例において、図12に示される処理は、図6の輻輳制御スレッド604において実行することができる。図12の処理は、送信スレッドから要求を受信することから開始する(ステップ1200)。これらの実例において、輻輳制御スレッドによって使用される適応制御処理は、ゲートウェイ内に所在する。各ゲートウェイは個別に、各優先度について使用される様々なパラメータの値を有する。送信スレッドは配信物の送信を開始する際、各値と共に現在の配信優先度を輻輳制御スレッドに引き渡す。輻輳制御スレッドは、現在の配信優先度に対応する値を使用する。この情報は、配信物の送信要求が最初に行われるステップ1200で受信することができる。当該要求は、送信可能なデータ量の特定を要求するものである。その後、データがキューからプルされる(ステップ1202)。このデータには到達時間及びパケット識別子が含まれる。その後、キュー内のデータからラウンド・トリップ時間が特定される(ステップ1204)。次に、配信の優先度に基づいてパラメータが取得される(ステップ1206)。これらのパラメータは、図6のユーザ定義パラメータ626に由来するものである。特に、ユーザ定義パラメータは、選択された優先度に関連する特定のパラメータ・セットに関するパラメータである。これらの実例において、優先度毎に様々な設定を有するパラメータは、 、閾値、及び最大待ち時間である。その後、適応型帯域幅制御処理が実行される(ステップ1208)。この処理は、例えば輻輳制御スレッドに含まれるステップとしてもよい。別法として、輻輳制御スレッドは、ステップ1208で関数又は外部プロセスをコールすることもできる。その後、結果が取得され(ステップ1210)、送信可能なデータ量と共に応答が返された後(ステップ1212)、図12の処理は終了する。

#### 【0073】

次に図13を参照すると、本発明の例示的な一実施形態に係る適応型帯域幅制御を実行する処理のフローチャートが示されている。図13に示される処理は、図12のステップ1208をより詳細に説明するものである。

#### 【0074】

図13の処理は、最小ラウンド・トリップ時間及び最大ラウンド・トリップ時間を特定することから開始する(ステップ1300)。その後、ウィンドウが特定される(ステップ1302)。その後、閾値を上回るパケット数及び閾値を下回るパケット数が肯定応答単位で特定される(ステップ1304)。次に、ブロック内のパケットの50%が閾値を

10

20

30

40

50

上回るかどうかが判定される(ステップ1306)。ブロック内のパケットの50%が閾値を上回っていない場合には、予想スループットが計算される(ステップ1308)。ステップ1308において、予想スループットは以下のように計算される。

E = W / min RTT

#### 【0075】

Eは予想スループット、Wはウィンドウであり、min RTTは、当該キューから取得されるラウンド・トリップ時間から見た最小ラウンド・トリップ時間である。その後、実際のスループットが特定される(ステップ1310)。実際のスループットは以下の式を使用して特定される。

A = W / observed RTT

10

#### 【0076】

Aは実際のスループット、Wはウィンドウであり、observed RTTは、パケットが送信された時点と肯定応答が受信された時点との時間差を使用して測定される値である。したがって、アイドル状態のネットワークでは最小ラウンド・トリップ時間が常にパケットの送信結果となることが予想されるため、予想スループットは最小ラウンド・トリップ時間に基づくものとなる。実際のスループットは、現在ラウンド・トリップ時間が最小ラウンド・トリップ時間よりも大きくなる現在のネットワーク状態に基づくものとなる。これらの実例において、最小ラウンド・トリップ時間は、ベース・ラウンド・トリップ時間と同一である。次に、バックログ・ルータ上のパケット数が計算される(ステップ1312)。実際のパケット数は以下の式を使用して推定される。

Diff = (E - A) / min RTT

20

#### 【0077】

Diffはバックログ・ルータ上のパケット数、Eは予想スループット、Aは実際のスループット、min RTTは認められる最小ラウンド・トリップ時間である。

#### 【0078】

次に、バックログ・ルータ上のパケット数がよりも小さいかどうかが判定される(ステップ1314)。上述のように、は、ウィンドウがパケット1つ分増加する前にバックログ・ルータ内に存在するはずの、接続単位のパケット数を示す整数である。バックログ・ルータ上のパケット数がよりも小さい場合には、ウィンドウが1つ増分される(ステップ1322)。その後、ウィンドウ・サイズを上限としてデータ送信が可能であることが指示され(ステップ1318)、その後処理は終了する。

30

#### 【0079】

再びステップ1314を参照し、バックログ・ルータ上のパケット数が以上である場合には、パケット数がよりも大きいかどうかが判定される(ステップ1316)。パケット数がよりも大きい場合には、ウィンドウ・サイズが1つ減分される(ステップ1324)。その後上述のステップ1318に進む。パケット数がを超えていない場合には、ウィンドウ・サイズの変更を行わずにステップ1318に進む。

#### 【0080】

ステップ1306に戻り、ブロック内のパケットの50%が閾値を上回っている場合には、ウィンドウ・サイズが半分に縮小される(ステップ1320)。その後上述のステップ1318に進む。

40

#### 【0081】

図13に示した上記の処理は、Venkataramani等の「TCP Nice: A Mechanism for Background Transfers」、ACM SIGOPS Operating Systems Review、Vol. 36、Issue SI Winter 2002、1~15頁に記載されているNiceアルゴリズム等、現在利用可能な各種帯域幅制御アルゴリズムの修正版に基づいている。

#### 【0082】

本発明の諸態様では、図13に記載される適応型帯域幅制御処理の実行に関与するいくつかのパラメータが特定されている。各パラメータとそれらに関する説明とを以下に列挙

50

する。

#### 【0083】

NICE\_ALPHA : ウィンドウが増加する前の、バックログ・ルータ内の接続単位の最小パケット数を示す整数。デフォルトは「1」である。

NICE\_BASE\_DISCARD : 接続単位で廃棄を行うべき当初ベースRTT測定数を示す整数。これは、配信が最初に開始された時点でのネットワークがまだ飽和状態にはない場合には、当初過剰に低いベースRTT測定値が発生し得ることに依拠する。デフォルトは「3」である。

#### 【0084】

NICE\_BASE\_SCALE : NICEによって維持されるグローバル最小RTT (v\_baseRTT) に適用される倍率を指定する浮動小数点数である。このパラメータを1.1のような小さい正数に設定すると、ネットワークが非定型の短いRTTを許可する場合に役立つ可能性がある。デフォルトは「0.0」である。10

#### 【0085】

NICE\_BELOW\_ONE : 低いウィンドウについて接続をアイドル状態にすることが可能な最大RTT数に相当するv\_cwnd\_below\_oneの下限を示す整数。デフォルトは「48」である。

#### 【0086】

NICE\_BETA : ウィンドウが減少する前の、バックログ・ルータ内の接続単位の最大パケット数を示す整数。この値は のデフォルト値である。デフォルトは優先度に基づく。20

#### 【0087】

NICE\_CLAMP : このパラメータが設定されると、ウィンドウ・サイズのクランプ (snd\_cwnd) は、現時点でネットワーク内に所在するパケット数よりも大きいパケット数の「4」を超えないように制限される。デフォルトは設定済みである。

#### 【0088】

NICE\_COND : このパラメータが設定されると、送信スレッドは、任意の時間待機する代わりに、ackスレッドによってシグナリングされる状態で待機する。デフォルトは未設定である。

#### 【0089】

NICE\_CONG\_RTX\_Q : このパラメータが設定されると、cong拡張スレッドは、各発信パケットの推定値をrtx\_qに更新する。rtx\_qは、RTTの計算に使用される。congスレッドには、発信時間の推定が正確であるという利点があるが、パケットをドロップする可能性がある。デフォルトは設定済みである。30

#### 【0090】

NICE\_DYNAMIC\_MSS : このパラメータが設定されると、適応型帯域幅制御処理は、低いMSS値から開始され、より大きいMSSを有する発信パケットのスニッフィングが行われるたびに、当該MSS値を増分させる。このようにして、MSSは、接続に使用されるMSSに急速に近付くことになる。デフォルトは設定済みである。

#### 【0091】

NICE\_FAST : 各適応型帯域幅制御処理ソケットが高速始動フェーズに入るミリ秒数を指定する整数。高速始動フェーズが長くなるほど、maxRTTの推定値は正確になる。ただし、このパラメータを大きい値に設定すると、その期間の間適応型帯域幅制御処理が効果的にディセーブルされることになる。デフォルトは「5000」即ち「5秒」である。40

#### 【0092】

NICE\_FAST\_RETURN : このパラメータが設定されると、高速始動フェーズは、EWOULDBLOCKエラーによってsend()が最初に失敗したときに終了する。これによって、ソケットの発信キューが非常に早い段階で一杯になることからゲートウェイの足かせとなる高速始動モードに費やされる時間が、最小限に抑えられることに50

なる。デフォルトは設定済みである。

#### 【0093】

NICE\_INTERFACE : N i c e によって使用されることになるインターフェース（ネットワーク・カード識別子）である。適応型帯域幅制御処理では、この時点で正しいインターフェースを動的に判定することはできない。したがって、インターフェースが最初のアクティブ・インターフェースでない場合には、インターフェースを手動で設定する必要がある。典型的には、このインターフェースは「e t h 1」等に設定されることになる。デフォルトは未設定である。

#### 【0094】

NICE\_MAX\_MULT : maxRTT の最小値をベースRTT の倍数として指定する浮動小数点数である。この浮動小数点数が設定されると、maxRTT 値が過剰に低くなることを防止するとともに、それに付随して生じる傾向がある低スループットの発生を防止するにも役立つ。デフォルトは「0 . 0」である。10

#### 【0095】

NICE\_MIN\_MSS : 使用すべき最小MSS を示す整数。効率化のために、オペレーティング・システムから提供される値のうちこのパラメータよりも低くなる値は、NICE\_DEFAULT\_MSS を優先して無視される。デフォルトは「1000」である。

#### 【0096】

NICE\_MIN\_PACKET : 一度に送信すべき最小データ量を指定する整数。このパラメータは、ok\_to\_send が指定値よりも小さい場合はそれ自体が0 に設定される点でNICE\_MIN\_SEND と異なる。これは、輻輳制御スレッドが送信スレッドに対して1バイトのような非常に小さい値を送信するよう指令し続けるような状況を回避するためである。この値を10 に設定すれば、輻輳制御スレッドは、少なくとも10 バイトが送信可能な状態になるまで待機する（計算結果が10 になるまで0 を返す）ことになる。デフォルトは「0」である。20

#### 【0097】

NICE\_MIN\_SEND : ok\_to\_send の最小値を指定する整数。つまり、輻輳制御スレッドは常に、送信スレッドに対して少なくともこのパラメータのデータ量を送信するように指令する。このパラメータを10 に設定すれば、適応型計算によって3 が指定されている場合でも、輻輳制御スレッドからは10 が返されることになる。デフォルトは「0」である。30

#### 【0098】

NICE\_NANO\_FIXED : 内部コールバック関数、即ちselect\_delay() の使用時に当該関数をスリープ状態にすべきマイクロ秒数を指定する整数。

#### 【0099】

NICE\_NANO\_FIXED は、コールバックがselect\_delay() 以外のものに設定されたときは無効となる。このパラメータが設定されない場合には、最終ブロックのスループットによって示される（最終ブロックにおける最速パケットのRTT 及び当該ブロックのサイズによって判定される）、パケットを送信するのに十分なスペースが設けられるまでにどれほどの時間が掛かるかに基づいて、遅延が動的に計算される。このパラメータが1234 に設定された場合には、nanosleep() のコールは行われない。このパラメータが設定されない場合には、遅延が動的に計算される。このパラメータが0 に設定された場合には、nanosleep() が0 の値でコールされ、いくつかのシステム上で10ミリ秒程度の遅延が存在する可能性がある。このパラメータが1234 に設定された場合には、nanosleep() のコールは行わない。その他の場合には、指定のマイクロ秒数（デフォルト=0）でnanosleep() をコールする。40

#### 【0100】

NICE\_PCAP\_TIMEOUT : 収集されたパケットのリストが取得されるまで50

オペレーティング・システムが待機する時間を示す整数。Linuxシステムは待機時間に関わらず1つのパケットが使用可能になるまで待機するだけであるため、この変数はLinuxシステムでは効果がない。この変数が有効となるSolarisのようなシステムでは、タイムアウトが低く設定されたときは正確なRTTが得られるが、ネットワークがアイドル状態にあるときはCPU時間が浪費されるというトレード・オフが存在する。デフォルトは「10ミリ秒」(Solarisでサポートされる最小値)である。

## 【0101】

NICE\_QUEUE\_LIMIT: configスレッドから読み出されたスニッフィング・スレッドによって書き込まれるパケット・キューの最大長を指定する整数である。キューが長くなるほど当該キュー内のパケットの移動時間が長くなり、それによってconfigスレッドの処理対象となる情報の到着に遅延が生じるため、Niceの応答性は低くなる。デフォルトは「10」である。10

## 【0102】

NICE\_RTT\_MAX\_CHANGE: ベースRTT及び最大RTTについて、それぞれの事前値に対して変更を行うことが可能な最大量を示す整数。このパラメータが設定されると、適応型帯域幅制御処理におけるスプリアス性の極端なRTT値に対する耐性が向上することになる。デフォルトは「1.5」である。

## 【0103】

NICE\_RTT\_MIN\_STDS: ベースRTTを上回る最小標準偏差数を示す整数。RTTは、ベースRTTを上回るものと見なされる必要がある。デフォルトは「0」である。20

## 【0104】

NICE\_RTT\_SMOOTHING: RTT(ラウンド・トリップ時間)測定を指數関数的減衰によってどの程度平滑化すべきかを示す整数。指數関数的減衰は、事前測定値の平均をとることによって行われる。このパラメータを設定すると、RTTの各変動の過補償によってNiceが不規則な挙動を示す場合に役立つ可能性がある。デフォルトは「50」である。

## 【0105】

NICE\_RTT\_STD\_SMOOTH: NICE\_RTT\_SMOOTHINGと同様であるが、標準偏差に関する整数である。標準偏差は、最近の(recent)RTT測定値の加重平均値に基づくものとなる。デフォルトは「0」である。30

## 【0106】

NICE\_SEND\_RTX\_Q: このパラメータが設定されると、送信スレッドは、各発信パケットの推定値を追加し、又は当該推定値をRTTの計算に使用されるrtx\_qに更新する。送信スレッドにはパケットがドロップされないという利点があるが、当該スレッドによる発信時間の推定値が不正確となる可能性がある。デフォルトは設定済みである。

## 【0107】

NICE\_THROUGHPUT\_AVG: 最新の(most recent)パケットをスループット計算にいくつ含めるべきかを示す整数。この値を大きくするほどスループット計算の正確さが向上するが、その代償として応答性が低下することになる。デフォルトは「20」である。40

## 【0108】

NICE\_THROUGHPUT\_START: スループット計算が開始される前に送信しなければならない最小パケット数を示す整数。このパラメータでは当初空状態にあった低速リンク・キューが加味される故に、ソケットの開始部分におけるスループットは当該ソケットの典型的なスループットとならない可能性があり、そのようなスループットは無視されることになる。デフォルトは「100」である。

## 【0109】

NICE\_WAIT\_BASE: ある接続で送信を断念してsnd\_nxt及びsnd

50

\_un aをリセットするまでの送信待機時間が、v\_baseRTT（グローバル最小RTT）の何倍に相当するかを示す整数。このように計算されたタイムアウトがNICE\_WAIT\_MAXのタイムアウトに加算される。デフォルトは「0」である。

#### 【0110】

NICE\_WAIT\_MAX：ある接続で送信を断念してリセットを行うまでの送信待機時間が、v\_maxRTT（グローバル最小RTT）の何倍に相当するかを示す整数。デフォルトは優先度に基づく。

#### 【0111】

次に図14を参照すると、本発明の例示的な一実施形態に従って配信物の送信をカスタマイズする処理のフローチャートが示されている。図14に示される処理は、各自のネットワーク内でカスタマイズされた配信システムを所望する顧客に解決策を提供するのに使用される処理である。10

#### 【0112】

図14の処理は、顧客要求を受信することから開始する（ステップ1400）。当該要求は、上記の解決策を実現するのに必要な情報を含む。例えば、当該要求には、顧客ネットワーク内のゲートウェイIDが含まれる。また、ゲートウェイを介してアクセスされる様々なクライアントの特性を含めることもできる。更に、図示の例にはクライアントによって行われる様々なタイプの配信物の送信要求も含まれる。当該要求が受信されたことに応じて、配信物を各エンドポイントに送信するためにクライアント・パラメータの分析が行われる（ステップ1402）。20

#### 【0113】

この分析では、様々な優先度で使用される各パラメータの設定が特定される（ステップ1404）。これらの実例において、優先度は、閾値、、及び最大待ち時間である。このように特定される各パラメータは、様々な優先度レベルとすることができます。例えば、図示の例では、「高」、「中」、及び「低」の3つの優先度レベルが使用されている。顧客要求に応じて図示の例とは異なる数の優先度レベル、例えば2つあるいは4つの優先度レベルを使用することもできる。特定の優先度レベルで生成されるパラメータは、クライアントの要求とゲートウェイが所在するネットワーク特性とに応じて、各ゲートウェイ毎に異なる可能性がある。

#### 【0114】

その後、適応型帯域幅制御用のコードが各ゲートウェイ毎に生成される（ステップ1406）。当該コードは、それ自体を特定のゲートウェイにインストールすることができるようパッケージ化される。当該コードは、特定のゲートウェイを対象とする適応型帯域幅制御処理及び各種パラメータを含む。当該コードは、ゲートウェイ上の適応型帯域幅制御処理のインストール及びセット・アップに必要な実行可能ファイルも含むことができる。その後、当該コードは配信対象の顧客に送信される（ステップ1408）。このようにして、顧客は、カスタマイズされた配信システムを要求に応じて企業又は他のプロバイダから受け取ることができる。30

#### 【0115】

かくして、本発明の諸態様は、データ転送速度を調整するコンピュータ実装方法、装置、及びコンピュータ使用可能なプログラム・コードを提供する。本発明の諸態様は、ユーザ入力を受信してアプリケーション・レベルでパラメータの各値を選択する。これらのユーザ定義パラメータは、適応型帯域幅制御処理においてネットワークを介して宛先に送信可能なパケット数を特定するのに利用される。本発明の諸態様によれば、ユーザは、様々なネットワーク状態及びネットワーク特性に基づいて各パラメータを定義することが可能となる。このように、本発明の諸態様によれば、様々なタイプのネットワーク及びネットワーク状態に基づいてデータを適応的に送信することが可能となる。40

#### 【0116】

本発明の諸実施形態は全体としてハードウェアの実施形態の形をとることも、全体としてソフトウェアの実施形態の形をとることもでき、ハードウェア要素とソフトウェア要素

1020304050

の両方を含む一実施形態とすることもできる。好ましい一実施形態では、本発明は、必ずしもそれだけに限定されるわけではないが、ファームウェア、常駐ソフトウェア、マイクロコード等を含むソフトウェアの形で実施される。

#### 【0117】

更に、本発明は、コンピュータ又は任意の命令実行システムによって使用され又はそれらと共に使用されるプログラム・コードを備えるコンピュータ使用可能な又はコンピュータに読み込み可能な媒体からアクセス可能なコンピュータ・プログラムの形をとることもできる。本明細書では、コンピュータ使用可能な又はコンピュータに読み込み可能な媒体は、上記命令実行システム、装置、又はデバイスによって使用され又はそれらと共に使用される上記プログラムを収容し、記憶し、通信し、伝搬し、又は移送することが可能な任意の装置であってもよい。

#### 【0118】

上記媒体は、電子系、磁気系、光学系、電磁気系、赤外線系、又は半導体系（すなわち装置又はデバイス）であっても、伝搬媒体であってもよい。コンピュータに読み込み可能な媒体の例としては、半導体又は固体メモリ、磁気テープ、取り外し可能コンピュータ・ディスクケット、ランダム・アクセス・メモリ（RAM）、読み取り専用メモリ（ROM）、固定磁気ディスク、及び光ディスクが挙げられる。光ディスクの現行例としては、コンパクト・ディスク・読み取り専用メモリ（CD-ROM）、コンパクト・ディスク・読み出し／書き込み（CD-R/W）、及びDVDが挙げられる。

#### 【0119】

プログラム・コードを記憶し又は実行しあるいはその両方を行うのに適したデータ処理システムは、システム・バスを介してメモリ要素に直接又は間接的に連結される少なくとも1つのプロセッサを含む。上記メモリ要素は、上記プログラム・コードの実際の実行中に利用されるローカル・メモリ、バルク記憶装置、及び実行中にバルク記憶装置からコードを検索しなければならない回数を少なくするために、少なくともいくつかのプログラム・コードを一時的に記憶するキャッシュ・メモリを含むことができる。

#### 【0120】

入力／出力すなわちI/Oデバイス（必ずしもそれだけに限定されるわけではないが、キーボード、ディスプレイ、ポインティング・デバイス等を含む）は、上記システムに直接又はI/Oコントローラを介して、連結することもできる。

#### 【0121】

介在する私設ネットワーク又は公共ネットワークを通して他のデータ処理システムあるいは遠隔プリンタ又は記憶装置に、上記データ処理システムが連結できるようにするために、上記システムにネットワーク・アダプタを連結することもできる。モデム、ケーブル・モデム及びイーサネット・カードは、現時点で使用可能なタイプのネットワーク・アダプタのほんのいくつかに過ぎない。

#### 【0122】

本発明の記載は、例示及び説明のために提示されるものであり、本発明を余すところのないものとし、又は開示の形態に限定することは、本出願人の意図するところではない。当業者には多くの修正形態及び変更形態が明らかとなるであろう。上記実施形態は、本発明の諸原理及び実際の適用形態が最良の形で説明されるように、また、想定される特定の使用に適するように様々な修正が施された様々な実施形態について当業者が本発明を理解することが可能となるように選択され説明されている。

#### 【図面の簡単な説明】

#### 【0123】

【図1】本発明の諸態様を実施することが可能な複数のデータ処理システムから成るネットワークを示す図である。

【図2】本発明の諸態様を実施することが可能なデータ処理システムのブロック図である。

【図3】本発明の例示的な一実施形態に従って、様々なプロパティを使用して配信を行う

10

20

30

40

50

ことが可能な例示的なネットワーク・データ処理システムを示す図である。

【図4】本発明の例示的な一実施形態に係る適応型帯域幅制御で使用される各コンポーネントを示す図である。

【図5】本発明の例示的な一実施形態に係る伝送制御プロトコル／インターネット・プロトコル(TCP/IP)及び同様のプロトコルを示す図である。

【図6】本発明の例示的な一実施形態に係る適応型帯域幅制御を実現する際に使用される各ソフトウェア処理及びコンポーネントを示す図である。

【図7】本発明の例示的な一実施形態に係る優先度を使用して、配信物を適応型送信機から受信機に送信するネットワークを示す図である。

【図8】本発明の例示的な一実施形態に係るネットワーク内の帯域幅使用率を示す図である。  
10

【図9】本発明の例示的な一実施形態に従ってパラメータに関するユーザ設定を構成する処理を示すフローチャートである。

【図10】本発明の例示的な一実施形態に係る送信スレッドに関する処理を示すフローチャートである。

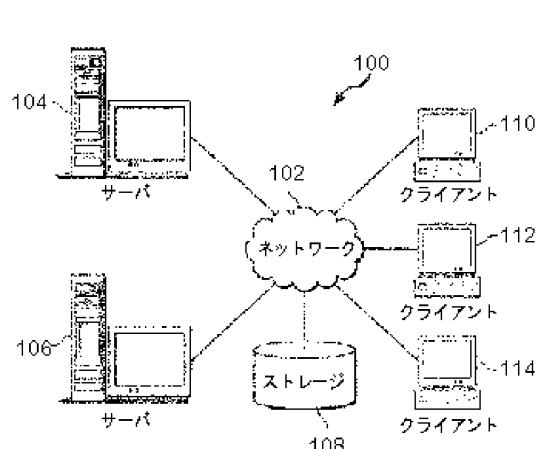
【図11】本発明の例示的な一実施形態に係るパケット・スニッフィング・スレッドの処理を示すフローチャートである。

【図12】本発明の例示的な一実施形態に係る輻輳制御スレッドの処理を示すフローチャートである。

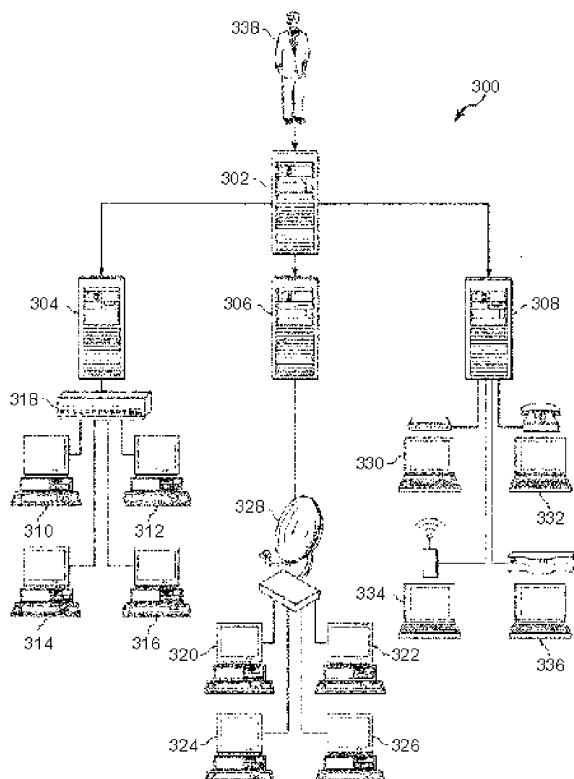
【図13】本発明の例示的な一実施形態に係る適応型帯域幅制御を実行する処理を示すフローチャートである。

【図14】本発明の例示的な一実施形態に従って顧客の受信機への配信物の送信をカスタマイズする処理を示すフローチャートである。  
20

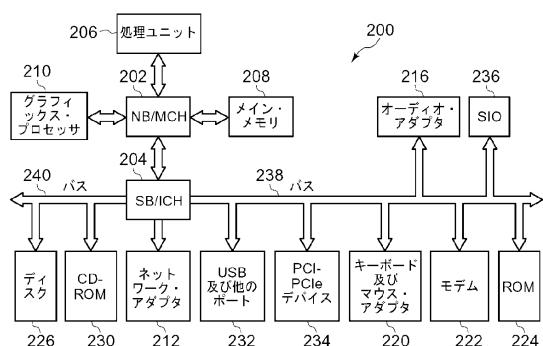
【図1】



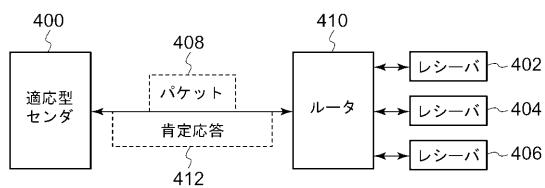
【図3】



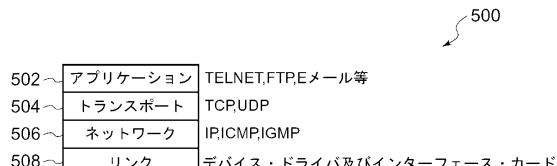
【図2】



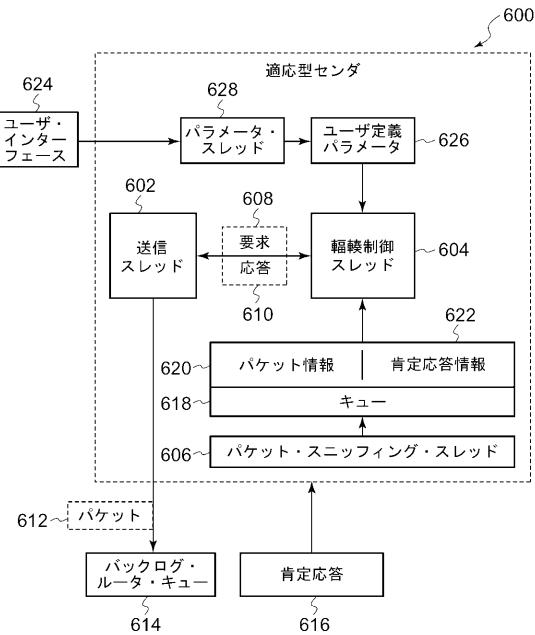
【図4】



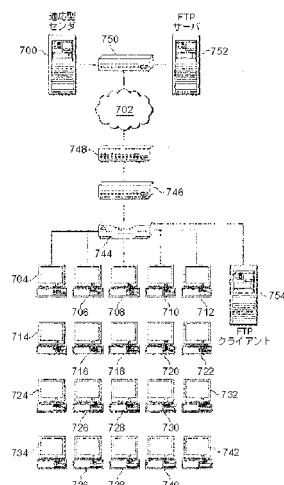
【図5】



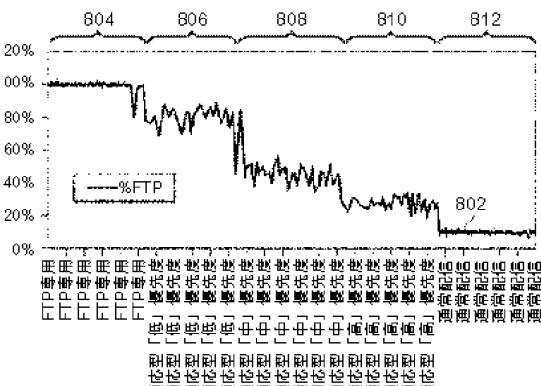
【図6】



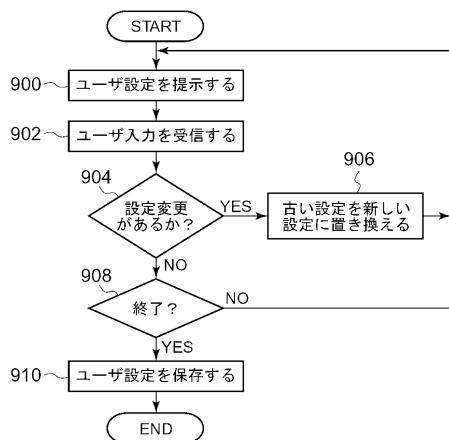
【図7】



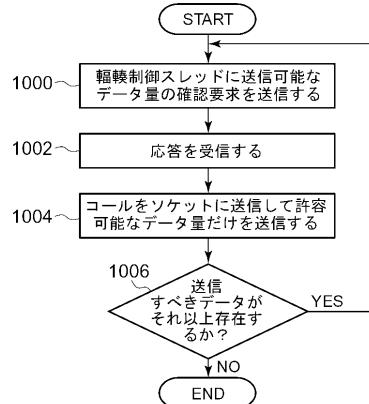
【図8】



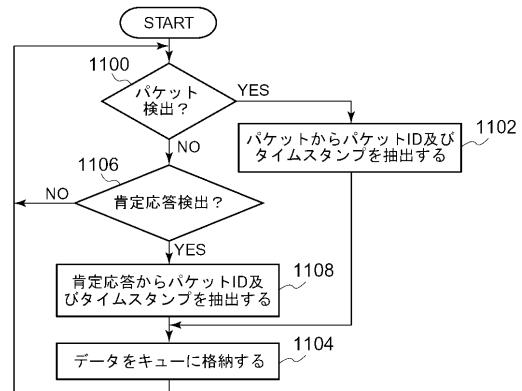
【図9】



【図10】



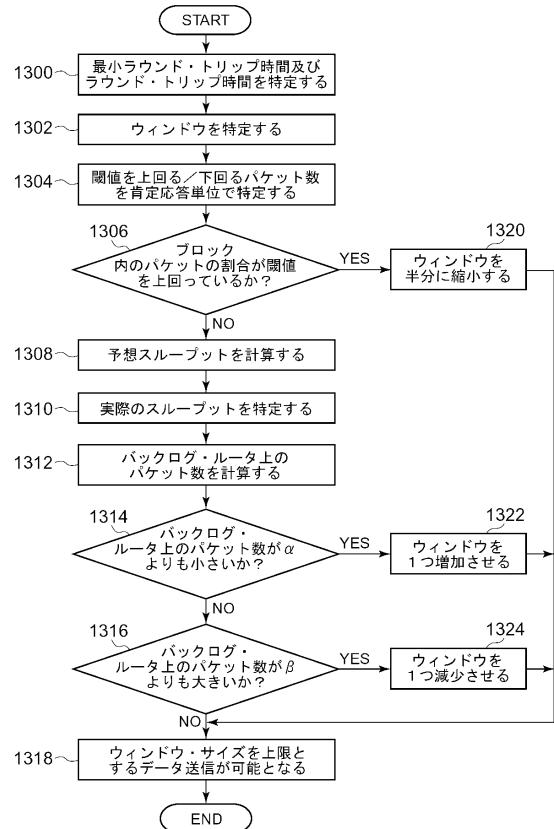
【図11】



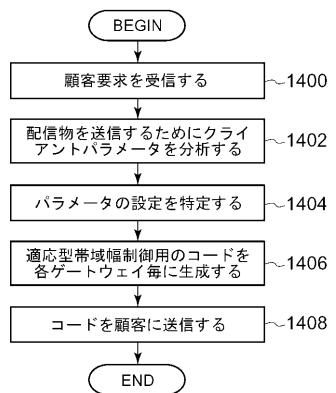
【図12】



【図13】



【図14】



---

フロントページの続き

(74)代理人 100086243

弁理士 坂口 博

(72)発明者 エリオット、スティーヴン

アメリカ合衆国78759 テキサス州オースティン リサーチ・ブルーヴァード11215

1133

(72)発明者 ラッザロ、クリストファー、ヴィクター

アメリカ合衆国78727 テキサス州オースティン カバナ・レーン12012

(72)発明者 トラン、タン

アメリカ合衆国78758 テキサス州オースティン ブローケン・フェザー・トレイル702

審査官 松崎 孝大

(56)参考文献 特開2002-44136 (JP, A)

特開2005-184494 (JP, A)

国際公開第02/052768 (WO, A1)

(58)調査した分野(Int.Cl., DB名)

H04L 12/56