(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第4474192号
(P4474192)
(45)発行日　平成22年6月2日(2010.6.2)　(24)登録日　平成22年3月12日(2010.3.12)

(51)Int.Cl.　F I
H04L 12/56 (2006.01)　H04L 12/56　200C

請求項の数 37　外国語出願　(全 24 頁)

(21)出願番号　特願2004-130090(P2004-130090)
(22)出願日　平成16年4月26日(2004.4.26)
(65)公開番号　特開2004-328753(P2004-328753A)
(43)公開日　平成16年11月18日(2004.11.18)
審査請求日　平成19年3月30日(2007.3.30)
(31)優先権主張番号　0305069
(32)優先日　平成15年4月24日(2003.4.24)
(33)優先権主張国　フランス(FR)

(73)特許権者　591034154
フランス・テレコム
フランス・７５０１５・パリ・プラス・ダルレ・６
(74)代理人　100089118
弁理士　酒井　宏明
(72)発明者　サラ　ウエスラティ
フランス国，９２１３０　イシー　レ　ムーランノー，アレ　オッシュ　２８
(72)発明者　ジェームズ　ロバーツ
フランス国，７８９６０　ヴワザン　ル　ブルトヌー，リュ　エメロード　３

審査官　石田　紀之

最終頁に続く

(54)【発明の名称】ネットワークにおけるサービス品質の暗黙的弁別のための方法及び装置

(57)【特許請求の範囲】
【請求項１】
ネットワークリンク上のフローのパケット（２０）を処理するための装置であって、
フェアビットレートに対する前記フローの着信ビットレートの解析に基づいて、且つ優先順位アルゴリズムを用いるフェアキューイングに従って、優先順位の関数としてキュー中のパケットをスケジューリングするためのスケジューリング手段（２８）を備える、装置。
【請求項２】
受け入れ基準に従って前記パケットの前記装置への受け入れを制御するための受け入れ制御手段（２４）を更に備える、請求項１に記載の装置。
【請求項３】
前記スケジューリング手段（２８）は、前記受け入れ制御手段（２４）に受け入れ条件データ（３６）を送信する、請求項２に記載の装置。
【請求項４】
前記受け入れ制御手段は、着信パケットごとに保護フローのリスト（３０）を問い合わせるための手段を備える、請求項２または３に記載の装置。
【請求項５】
最後のパケットが受信されてから経過した時間が閾値を超えるフローを、前記保護フローのリストから消去するための手段を更に備える、請求項４に記載の装置。
【請求項６】

前記受け入れ制御手段は、前記保護フローのリストの中にないフローにパケットが属する場合に、前記受け入れ基準が満たされるか否かを判定するための手段を備える、請求項４または５に記載の装置。

【請求項７】

前記受け入れ基準が満たされる場合に前記リスト（３０）に新たなフローを登録するための手段を備える、請求項４～６のいずれか一つに記載の装置。

【請求項８】

前記受け入れ条件データは、

送信すべきパケットを常に有するデータフローによって達成されるビットレートを表すフェアビットレート値と、

ある期間に送信される優先順位パケットの長さの和をその期間の持続時間で割って求めた値に相当する優先順位負荷値と、

を含む、請求項２～７のいずれか一つに記載の装置。

【請求項９】

前記スケジューリング手段は、アクティブフローのリスト内にない前記キュー内のフローのパケットを優先順位パケットとして、且つ前記リスト内にすでにあるフローのパケットを非優先パケットとしてスケジューリングする、請求項１～８のいずれか一つに記載の装置。

【請求項１０】

前記スケジューリング手段は、ＰＩＦＯキュー内のパケットをスケジューリングする、請求項１～９のいずれか一つに記載の装置。

【請求項１１】

ポインタＰが、前記キューの先頭で最後の優先順位パケットを識別する、請求項１０に記載の装置。

【請求項１２】

アクティブフローの識別子を含むアクティブフローのリストを使用するのに更に適し、パケットをスケジューリングするためにタイムスタンプを用いる、請求項１１に記載の装置。

【請求項１３】

フローのパケットの到着と出発の関数として、フローをアクティブフローのリストに書き込み、且つフローを該リストから消去するための手段を更に備える、請求項１１に記載の装置。

【請求項１４】

輻輳測定手段を更に備える、請求項１２または１３に記載の装置。

【請求項１５】

現地時間と、現在の測定期間中に送信される優先順位パケットのバイト数と、この現在の測定期間中にダミーフローが送ることができるバイト数との関数として輻輳の測定を実行する、請求項１４に記載の装置。

【請求項１６】

前記ＰＩＦＯキューが空か否かを判定するための手段を備える、請求項１０～１５のいずれか一つに記載の装置。

【請求項１７】

受け入れ制御レベルでサービスの等級を弁別するための弁別手段を更に備える、請求項１～１６のいずれか一つに記載の装置。

【請求項１８】

前記フローは、アドレス属性に適用されるハッシュ関数によって識別される、請求項１～１７のいずれか一つに記載の装置。

【請求項１９】

ネットワークリンク上のフローのパケット（２０）を処理する方法であって、

フェアビットレートに対する前記フローの着信ビットレートの解析に基づいて、且つ優

先順位アルゴリズムを用いるフェアキューイングに従って、優先順位の関数としてキュー中のパケットをスケジューリングするためのスケジューリングステップを含む、方法。
【請求項２０】
前記パケット（２０）を処理する装置への前記パケットの受け入れを受け入れ基準に従って制御する受け入れ制御ステップを更に含む、請求項１９に記載の方法。
【請求項２１】
前記データ（３６）の受け入れを制御するための手段（２４）に向けて、受け入れ条件を送出するためのステップを更に含む、請求項２０に記載の方法。
【請求項２２】
前記受け入れ制御ステップは、着信パケットごとに保護フローのリスト（３０）を問い合わせることを含む、請求項２１に記載の方法。
【請求項２３】
最後のパケットが受信されてから経過した時間が閾値を超えるフローが保護フローのリスト（３０）から消去される、請求項２２に記載の方法。
【請求項２４】
前記保護フローのリストの中にないフローにパケットが属する場合に前記受け入れ基準が満たされるか否かを判定するためのステップを含む、請求項２２または２３に記載の方法。
【請求項２５】
前記受け入れ基準が満たされる場合に前記リスト（３０）に新たなフローを登録するステップを備える、請求項２２～２４のいずれか一つに記載の方法。
【請求項２６】
前記受け入れ条件データは、
送信すべきパケットを常に有するデータフローによって達成されるビットレートを表すフェアビットレート値と、
ある期間に送信される優先順位パケットの長さの和をその期間の持続時間で割って求めた値に相当する優先順位負荷値と、
を含む、請求項２１～２５のいずれか一つに記載の方法。
【請求項２７】
前記スケジューリングステップは、アクティブフローのリスト内にない前記キュー内のパケットを優先順位パケットとして、且つ前記リスト内にすでにあるフローのパケットを非優先パケットとしてスケジューリングする、請求項２０～２６のいずれか一つに記載の方法。
【請求項２８】
前記スケジューリング手段は、ＰＩＦＯキュー内のパケットをスケジューリングする、請求項２０～２６のいずれか一つに記載の方法。
【請求項２９】
ポインタＰが、前記キューの先頭で最後の優先順位パケットを識別する、請求項２８に記載の方法。
【請求項３０】
前記フローの識別子を含むアクティブフローのリストを更に使用し、パケットをスケジューリングするためにタイムスタンプを用いる、請求項２９に記載の方法。
【請求項３１】
フローのパケットの到着と出発の関数として、フローをアクティブフローのリストに書き込み、且つフローを該リストから消去するためのステップを更に含む、請求項３０に記載の方法。
【請求項３２】
輻輳測定を更に備える、請求項３０または３１に記載の方法。
【請求項３３】
現地時間と、現在の測定期間中に送信される優先順位パケットのバイト数と、この現在

の測定期間中にダミーフローが送ることができるバイト数との関数として、輻輳の測定を実行する、請求項３２に記載の方法。
【請求項３４】
前記ＰＩＦＯキューが空か否かを判定するためのステップを備える、請求項２８～３３のいずれか一つに記載の方法。
【請求項３５】
パケットの損失に関する信号をユーザに向けて送信する、請求項１９～３４のいずれか一つに記載の方法。
【請求項３６】
受け入れ制御レベルでサービスの等級を弁別することを更に備える、請求項１９～３４のいずれか一つに記載の方法。
【請求項３７】
複数のリンクにわたるフローの負荷分散が、前記フロー識別子の自由部分を含むアドレス属性の作用を利用して行われる、請求項１９～３６のいずれか一つに記載の方法。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、パケット交換ネットワークのアーキテクチャとそこで行われるサービス品質管理に関する。
【背景技術】
【０００２】
インターネットは、サービスとアプリケーションを幅広く支援するのに適したマルチサービス機構である。インターネット上のトラフィックは、一般的には音声や映像アプリケーションによって生成されるリアルタイム（ストリーミング）トラフィックと、デジタル文書の転送に相当するデータ（伸縮性）トラフィックの２つに大別される。リアルタイムトラフィックには信号を保存するというサービス品質の要求がある。つまり、発信元で生成される信号を特徴づけるビットレートの変化を、その信号がネットワークを通過する過程で保たれなければならない。データトラフィックのサービス品質は文書の転送時間によって測定する。この時間、すなわち転送中に達成される平均ビットレートは、発信元から目的地に至る通信チャネルの総体に左右される。インターネットのネットワークのためのサービス品質の目的は、それがデータトラフィックに対して透過的に現れることである。透過的であるならば、別の場所（サーバ、アクセスネットワーク、ユーザ装置）で持ち込まれる制限要因をおいてほかにビットレートを減少させるものはなくなり、この意味で、ネットワークはデータフローのビットレートを保存するものといえる。
【０００３】
インターネットは、営利的な面においては、ユーザクライアントに移送サービスを提供する公共的ネットワークである。したがって、対価の請求が重要な課題となる。ネットワークアーキテクチャは運営者に投資利潤をもたらし、なおかつユーザが求める良質サービスについて、競争に有利な価格を設定できるものでなければならない。
【０００４】
この品質と投資利潤の要求については、ネットワークアーキテクチャに関する種々の提案がさまざまな形で応えている。アーキテクチャには、Ｉｎｔｓｅｒｖ、Ｄｉｆｆｓｅｒｖ、ＭＰＬＳ－ＴＥのように標準化されたものもあれば、独自の仕様を持つものもある。
【０００５】
既存のＩＰネットワークアーキテクチャはベストエフォートアーキテクチャであり、サービス品質を一切保証しない。性能は過剰仕様によって満足なものとなり、ユーザの協力に依存する。つまりユーザは原則として、（ＴＣＰやＴＣＰフレンドリプロトコルを使用する場合は特に）ネットワークの輻輳状態に応じて自らのアプリケーションのビットレートを調整しなければならない。リアルタイムトラフィックとデータトラフィックパケットは、両者を区別することなく処理される。

【０００６】
ＩＰルータは現在、パケットの配信にあたって２つ以上のパスを利用できる場合に、負荷分散の原理を用いる。所与のパケットに対する選択は普通、パケットの元のＩＰアドレスにハッシュ関数を適用することによって決定される。この関数を適用した結果として、該当するインターフェースを指定する整数が得られる。こうして利用可能なパスにわたって負荷が分散され、同じフローのパケットはどれも同じパスを取り、それゆえ正確な順序で到着する。
【０００７】
通常、小口ユーザの料金は固定されている。一方大口ユーザは、送受されるトラフィックの量に応じた額を運営者に支払うことができる。この場合は普通、パケット損失率やネットワーク横断遅延、信頼性要素といった商品のサービス品質を規定する取り決めが存在する。
【０００８】
上記のアーキテクチャに対する改良として提案された高水準アーキテクチャのほとんどは、リソース予約の原理を採用している。ネットワークとユーザは、予定されたコールについて次のとおりにトラフィックの取り決めを設定する。
【０００９】
・ユーザは、トラフィックの性質を記述することによって予定されたコールを告知する。
・ネットワークは、可能であれば必要なリソースを割り当てる（受け入れ制御）。
・送信されたトラフィックを進入時に監視し、それがあらかじめ提供された記述に一致することを確認するか、割り当てられたリソースをＷＦＱタイプのスケジューリングで監視する。
【００１０】
提案されたアーキテクチャによって異なるが、この取り決めは、マイクロフローかフローの集合、ポイントツーポイントコールかマルチポイントコールに関係する。
【００１１】
料金は取り決めに盛り込まれる。料金は一般に、ユーザが提供するトラフィックの記述に応じて異なり、その記述によってリソースの割り当てが左右される。
【００１２】
サービスの等級に基づいて弁別がなされる場合、ネットワークは同じ等級の全パケットを特定の方法で処理する。リアルタイムトラフィックとデータトラフィックのそれぞれに求められる２タイプの品質を弁別し、さらに同じタイプの中で種々の品質レベルを弁別する。通常ならば、トラフィック集団についてユーザと運営者との間で交わされる取り決めの条件に従って、各種サービス等級へのパケットの割り当てを綿密に監視することが必要となる。
【００１３】
料金は取り決めの一部であり、通常はそれぞれの等級で送信されるトラフィックの量によって決まる。当然、品質水準に応じた課金差が生じる。そうでない場合、ユーザは、すべてのトラフィックを最上級レベルのトラフィックで送信することになる。
【００１４】
非特許文献１に記載された自己管理インターネットでは、サービスの予約と等級を避けている。このアーキテクチャで輻輳が発生すると、ネットワークは“単価”を設けた明確な輻輳通知（ＥＣＮ）を送信する。ユーザは、受け取った通知にどう応答するかによって“勘定書”を調整でき、それ故、自らのコールについて余分に支払うことを了解するユーザは、受け取った通知に応答しないことによって輻輳を無視できる。
【００１５】
リアルタイムトラフィックとデータトラフィックとの弁別は、原理上必要でなくなる。ＥＣＮマーキングの決定にあたって仮想キューの占有を用いることでパケットの遅延を無視できる程度にまで抑えることができる（リアルタイムフローなら十分に低く抑えること

ができる）と主張する。

【００１６】

自己管理ネットワークなら専用のシグナリングプロトコルを用意する必要はなく、ベストエフォートネットワークの場合と同じくらい簡素なリソース管理が行える。

【００１７】

例えば、非特許文献２に記載されたフロー認識ネットワーキングアーキテクチャでは、それぞれのユーザフローが進行中に識別され、それらのフローに基づいてトラフィック制御が行われる。フローとは、ヘッダの不変フィールドの中で同じ値を持つパケットの集合であり、その不変フィールドとは一般に発信元ＩＰアドレスと目的地ＩＰアドレス、トランスポートプロトコルポート番号（ＩＰｖ４）、フローレベルフィールド（ＩＰｖ６）などである。フローには、無活動期間にからむタイムアウト値を指定することができる。その無活動期間中にパケットが観察されないとフローは終了する。図１は、そのアーキテクチャの原理とコンポーネントを示す。手段４はリアルタイムフローのピークビットレートを監視する。パケットは配信決定モジュール６に送信され、配信決定モジュール６は保護フローのリスト１０を調べる。このモジュール６は、受け入れを許可されたパケットをモジュール８に送信し、モジュール８は、リアルタイムフローのパケットに優先順位を与える優先順位キューに従ってそれらのパケットを順序付けする。モジュール８は、モジュール６に向けて受け入れ条件データ１６を送信する。このデータは原則的に、リアルタイムフローの現在の累積ビットレートの推定値と、データフローに利用できる帯域幅の測定値とを含む。

【００１８】

リンクやパスが瞬間的に輻輳した場合に新たなフローが立ち上がるのを防ぐため、暗黙的受け入れ制御を適用する。この制御は、すでに進行中のフローのサービス品質を保護する。非特許文献３などの文献では、暗黙的受け入れ制御を実施するための提案がなされている。

【００１９】

図１のアーキテクチャでは、リアルタイムトラフィックとデータトラフィックとが、それぞれのサービス等級で区別される。ただしその受け入れ条件は、フローのタイプにも、フローのトラフィック特性にも依存しない。このため、フローのトラフィック特性を伝える必要はない。リアルタイムフローのピークビットレートが事前に定められた制限を超えないことを進入時にチェックする必要は依然としてある。受け入れ条件を決定するには、その制限を知ることが不可欠である（非特許文献４参照）。

【００２０】

そのアーキテクチャで料金を左右するのは、クライアントによって送受されるトラフィックの量だけである。すでに開始したフローはどれも保護され、それ故課金できるので、有効である。リアルタイムとデータとの間の区別は、リアルタイムフローが無視できる損失と遅延を受け且つデータフローがビットレートで制限されないので、置き換えの動機がなく、差別的な課金を必要としない。

【００２１】

米国企業Ｃａｓｐｉａｎ　Ｎｅｔｗｏｒｋｓは、タイプの異なるフロー認識アーキテクチャを提案している（米国特許出願ＵＳ－２００２／５７６９９、ＵＳ－２００２／５７６５１及びＵＳ－２００２／８０７８６などを参照）。そのフローは、上述のとおり進行中に識別される。受け入れを許可されたそれぞれのフローは、サービス品質のパラメータとルートに関連付けられる。サービス品質のパラメータは、パケットのヘッダやユーザのＳＬＡから導き出される仕様表など、種々の情報源から導き出す。フローの性能要求に応じるため、サービス品質のパラメータとネットワークリソースの使用状況に応じてルートを算出する。サービス品質のパラメータは、数ある中でも、例えば保証レート（ＧＲ）を指定する。フローには、そのルートの輻輳状況に応じて利用可能レート（ＡＲ）が一定の期間にわたって割り当てられる。このアーキテクチャでは、フローのビットレートが割り当てられたレート（ＧＲ＋ＡＲ）に一致することを保証するため、進入時にスケジューリ

ングを行う。退出時のスケジューリングは、性能要求（特にパケットあたりの遅延）への応諾を保証する。進入時のスケジューリングでは余剰パケットの拒絶をもって間隔を設け、退出時のスケジューリングは、ＷＦＱアルゴリズムを各フローに使用する。新たに着信するフローのために適切なルートを見つけることができない場合は、フロー受け入れ制御を使用しなければならない。

【００２２】

【非特許文献１】F.P.Kelly,「Model for a self-managed Internet（自己管理インターネットのためのモデル）」, the Royal Society of Philosophical Transactions, Ａ３５８巻, ｐ.２３３５－２３４８, ２０００年

【非特許文献２】J.Roberts他,「Quality of service by flow-aware networking（フロー認識ネットワーキングによるサービス品質）」, the Royal Society of Philosophical Transactions, Ａ３５８巻, ｐ.２１９７－２２０７, ２０００年

【非特許文献３】A.Kumar他,「Non-intrusive TCP connection admission control for bandwidth management of an Internet access link（インターネットアクセスリンクの帯域幅管理のための非侵入性ＴＣＰ接続受け入れ制御）」, IEEE Comm. Mag., ３８巻, 第５号, ｐ.１６０－１６７, ２０００年

【非特許文献４】T.Bonald他,「IP traffic and QoS control:towards a flow-aware architecture（フロー認識アーキテクチャに向けて：ＩＰトラフィックとＱｏＳ制御）」, Proc. of World Telecom. Conf., パリ, ２００２年

【発明の開示】

【発明が解決しようとする課題】

【００２３】

上述のアーキテクチャにはどれも、品質保証の性質や実施の複雑さ、ネットワークの投資利潤などにさまざまな形で関係する欠点がある。

【００２４】

ベストエフォートアーキテクチャの欠点は以下のように公知である。

・輻輳が発生した場合にすべてのコールで性能が低下する。

・データトラフィックと競合するリアルタイムコールでパケットの損失や遅延を制御できない。

・とりわけユーザに悪意がある場合に生じる、不公正な帯域幅分配。

・過剰仕様でサービス品質を保証するにはコストがかかる。

・固定料金では、ユーザトラフィックの変動が激しい場合に、十分な投資利潤を保証しつつ公正な価格を設定することは困難である。

・トラフィックの量に応じた課金には、その輻輳に対する感受性ゆえ（失われたパケットの再生）、そして最低レベルの品質に応じることができず進行中のコールが中断される可能性があるため、疑問が残る。

【００２５】

予約の原理には以下の欠点がある。

・進入時にチェックできるパラメータを使用して（典型的には高い可変ビットレートを有する）コールのトラフィックを簡潔に記述することが実際には不可能であることが判明している。

・予約は複雑であり、シグナリングプロトコルを必要とし、さらに各コールのパラメータ（“状態”）を含むデータベースを維持すること、そしてスケジューリング機構を使用することが必要になる。

・呼が取るべきパスが指定されない場合、予約の概念は不明確になる（特にＤｉｆｆｓｅｒｖアーキテクチャの場合の異なる目的地に向かうトラフィックフロー集合の予約、ＩＰ配信の不安定さ、等）。

・厳格なサービス品質の保証にはリソースの点でコストがかかる（例えばＩｎｔｓｅｒｖの保証サービス等で）。

・ユーザが自らのトラフィックを過剰に見積もるという事実を補償するために一般的に

用いられる過剰予約は、現実の保証の不在を招く。
課金は問題をはらむ。過剰予約の場合は、（トラフィック取り決めの）トラフィックパラメータに基づく料金が、実際に送信されるトラフィックの量に左右されるコールのコストに直結しない。厳格保証の場合には、予約されるリソース（帯域幅、メモリ）に関係する料金が法外に高くなるおそれがある。
【００２６】
サービス等級による区別は別の問題を招く。
・データトラフィックについて、定量可能で検証可能なサービス品質保証がない。
・コールのパスが固定されない場合、リアルタイムトラフィックのサービス品質は保証できない。
・トラフィックの単位を等級とユーザによって管理するのは煩雑なままである（ＳＬＡ管理、ポリシーサーバ、トラフィックパラメータの伝達、その他）。
・低優先順位等級のトラフィックは、ベストエフォートアーキテクチャにおけるトラフィックと同じ欠点を被る（輻輳が発生した場合の性能破綻、悪意のあるユーザに対する保護の不在）。
・種々のサービス等級に対する課金は問題をはらむ。価格差は必要であるが、品質に明白な差がなければユーザは納得しないであろう。
・料金とコストと間に直接的な関係がなければ、投資利潤をもたらす課金構造を定義するのは困難である。
【００２７】
自己管理インターネットは上述の問題のかなり多くを回避するが、その課金原理そのものが問題をはらんでいる。
・ＥＣＮマークに基づく課金は複雑であり、なおかつ争いを起こし得る。
・運営者Ａのネットワークに加入するユーザに対し、別の運営者Ｂに属する下流ネットワークから来る輻輳マークについて、どのように支払を請求すればいいか？
・ＥＣＮ課金収入は、適切に指定されたネットワークの投資利潤を提供しない。別の課金原理でもってその利潤を提供しなければならないならば、ＥＣＮ課金は、不公正な割増金としてユーザに認識される。
【００２８】
上記で引用した公開文献で開示されたフロー認識アーキテクチャには、なおも以下の欠点がある。
・リアルタイムトラフィックとデータトラフィックとを明示的に区別するには、リアルタイムフローのピークビットレートを制御する必要がある。
・たとえ輻輳がなく、性能を低下することなくより高いビットレートのフローを受け入れることが可能であったとしても、リアルタイムフローの最大ピークビットレートを固定し、それを厳格に監視する必要がある。
・依然として、悪意あるユーザよって性能に支障をきたすおそれがある。
・（擬似ＴＣＰ接続を使って使用可能帯域幅を推定するか、又は観察された損失率から）輻輳状態を測定するために提案された方法は、実施に困難をともなう。
【００２９】
Ｃａｓｐｉａｎのフロー認識アーキテクチャについては、以下の問題に留意されたい。
・サービス品質のパラメータとフローあたりのトラフィックのパラメータを考慮に入れると実施が複雑になり、拡張の問題を招く。
・このアーキテクチャに関する文献では、フローのサービス品質への準拠を保証するための方法が明記されてないが、そのような方法や、とりわけ受け入れ制御の原理の定義は、提案されたアーキテクチャの効率性を決定づけるものである。
・言及したスケジューリングの形態（進入時の離間、退出時のフローあたりのＷＦＱ）でサービス品質要求に応じるためには、それぞれのフローに特有のパラメータ（特にフローがリアルタイムフローならばＧＲ、ＭＲ、データフローならばＡＲ）を把握する必要がある。

・課金の問題が未解決である。
【課題を解決するための手段】
【００３０】
本発明の第１の目的は、優先順位アルゴリズムを用いるフェアキューイングに従ってキューの中のパケットをスケジューリングするためのスケジューリング手段（又はスケジューリングステップ）を備える、ネットワークリンクのフローのパケットを処理するための装置と方法を提供することである。
【００３１】
優先順位スケジューリングを用いるフェアキューイングは、ビットレートがダイナミック閾値よりも低いフローのパケットに優先順位を与えるものであり、この閾値は、例えばフェアキューイングアルゴリズムに従って処理されることを前提に、複数の待機パケットを有するフローに現在用いられているビットレートに相当する。
【００３２】
この種の装置や方法は、コアネットワークの場合よりも輻輳のリスクを御しやすいアクセスネットワークにおいては特に、受け入れ制御手段なしで作用することができる。
【００３３】
本発明によれば、フローごとの受け入れ制御には、フェアキューイングスケジューリングにからめることができる。
【００３４】
このように、本発明のさらなる目的は、ネットワークリンクのフローのパケットを処理するための装置と方法の提供であって、受け入れ基準に従ってこの装置への前記フローの受け入れを制御するための受け入れ制御手段（又は受け入れ制御ステップ）と、優先順位アルゴリズムを用いるフェアシェアリングに従ってキューの中のパケットをスケジューリングするためのスケジューリング手段（又はスケジューリングステップ）と、を備える。
【００３５】
スケジューリング手段は、受け入れ制御手段に向けて受け入れ条件データを送ることができる。
【００３６】
この装置と方法は、リアルタイムフローとデータフローとを明示的に判別することなくサービス品質を保証する。
【００３７】
これにより、ビットレートが、受け入れ条件によって決まる閾値を下回るリアルタイムフローのパケットで遅延を最小限に抑えることができる。受け入れ制御が作動しない場合は、受け入れ条件の代わりに、トラフィックの状態によってこの閾値が決まる。
【００３８】
受け入れ条件は、データフローによって達成されるフェアビットレートと優先パケットによる負荷を測定するスケジューリング手段によって直接に決定される。
【００３９】
受け入れ制御とフェアキューイングスケジューリングの組合せは、ある程度の“相互保護”を提供する。すなわち、データフローのビットレートはフェアキューイングによって保護され、リアルタイムフローのパケットに優先順位を与えることによってそのパケットの遅延を防ぐことができ、受け入れ制御を用いて考慮すべきフローの数の制限することでフェアキューイングスケジューリングの複雑さが回避され、受け入れ制御は、スケジューリング手段に一体化された測定によって容易になる。
【００４０】
（暗黙的）受け入れ制御技術とフェアキューイングスケジューリングを組合せることによって、本発明は、トラフィック特性を宣言したり、サービス等級を定義したり、伝達したり、リソースを明示的に予約することなく、“十分な”品質を保証する。
【００４１】
受け入れ制御が作動しない場合はクロス保護の利点がいくつか失われるが、リンクが輻

輳しないならば、サービス品質の暗黙的区別は保たれる。ビットレートが低いリアルタイムフローのパケットの待ち時間はごくわずかであり、データフローは、フェアキューイングによって保証されるビットレートを達成できる。処理すべきユーザ数が比較的少なく、輻輳のリスクが低いアクセスネットワークのリンクの場合は、この構成で十分である。

【００４２】

選択されたスケジューリング形態なら、ユーザの協力がない場合でもアーキテクチャが被害を被ることはない。このアーキテクチャは、送受するトラフィックの量に基づく単純な課金に対応するものである。自己管理ネットワークにおけるユーザと運営者との関係の単純さは保たれ、ＥＣＮマークに応じた課金にからむ欠点もない。この単純さは、フロー認識アーキテクチャの堅牢さと効率性によって達成される。

【００４３】

キューの中に少なくとも１個のパケットが常に存在する程度のビットレートで着信するフローは、退出時にほぼ同じフェアビットレートを達成する。本発明において、（例えば上流のキューでの待機のため）フェアビットレートよりも瞬間的に高いビットレートを持つリアルタイムフローのパケットが殺到すると、スケジューリング装置は、最初のパケットに優先順位を与えるが、フェアビットレートに整合するスペーシングを課すことによって後続のパケットを遅らせる。フローの元のビットレートがこのフェアビットレートよりも低い場合、目的地で受信メモリによって課せられる遅延に対して補足の遅延を導入しないことによって、このスペーシングがフローを“平滑化する”。なお、この受信メモリは、最後のユーザにパケットの元のタイミングレートでそれらのパケットを送信しなければならない。この意味で、遅延は取るに足りない。反対に、ビットレートがフェアビットレートよりも大きいリアルタイムフローは降格され、発信元は通常ならばパケットの損失を防ぐため、ビットレートの削減を適用することになる。

【００４４】

したがって、使用する機構は、当該ネットワークの外的要因によって決まるピークビットレートが特定の閾値を下回るフローに対して、微々たる遅延と微々たるパケット損失を保証する。その閾値はダイナミックであり、選択されたスケジューリング形態によって達成される公平な（ｍａｘ－ｍｉｎの意味において）ビットレートに相当する。そして、受け入れ制御がデータフローのビットレートを保存しリアルタイムフローの信号を保存する、という二重の目的を達成する。

【００４５】

受け入れ制御は、考慮に入れるべきフローの数を制限することによってフェアキューイングスケジューリングを拡張できる。リンクの輻輳状態の測定はフェアキューイングによって促進され、測定された状態は受け入れ条件を決定するために使われる。

【００４６】

本発明は、優先順位スケジューリング装置と方法を有するフェアキューイングを使用してリアルタイムフローとデータフローとを暗黙的に弁別し、リアルタイムフローが少ないパケットあたりの遅延を有することを保証する。

【００４７】

フローごとの暗黙的受け入れ制御と優先順位スケジューリングを用いるフェアキューイングとを組み合わせることによって、リアルタイムフローとデータフローとを明示的に区別することなく、サービス品質が保証される。

【００４８】

受け入れ制御は、輻輳した時に受信する最初のパケットを拒絶することによって到着するフローを拒否するので、負荷分散機構を用いれば、フロー識別子に基づいて順応的配信を行うことができる。ユーザが成功するまで同じパケットを再送することによって再試行みれば、トラフィックは失われない。それぞれの試みでパケットを目的地に配信できる別のリンクにパケットが提示されるならば、成功の確率はより大きくなる。

【００４９】

それには、ルータで使われる負荷分散機構がフロー識別子に基づいて動作し、そのフロ

ー識別子は、ＩＰｖ４のポート番号やＩＰｖ６のフローラベルなど、ユーザによって自由に満たされるフィールドを含む。負荷分散を実行するハッシュ関数は引数としてこの自由フィールトを含み、その結果ユーザは、更新のたびに問題の該フィールドの値を変更することによって一種の順応的配信を実行する。

【００５０】

本発明は、フローの知識を有するネットワークの堅牢さと自己管理インターネットの単純さを組み合わせることを可能にする。前者の暗黙的受け入れ制御はそのまま利用できるので、ユーザがリアルタイムトラフィックとデータトラフィックとを区別する必要はない。この区別は、ネットワークがそれぞれのトラフィック特性に基づいて自動的に行う。

【発明を実施するための最良の形態】

【００５１】

図２は、第１の実施の形態を示す。この図の参照番号２４は、着信フローのパケット２０に関して受け入れ制御を実行する配信モジュール又は配信手段を示す。

【００５２】

フローの定義は、例えばT.Bonaldらの論文「IP traffic and QoS control：the need for a flow-aware architecture（ＩＰトラフィックとＱｏＳ制御：フロー認識アーキテクチャの必要性）」，World Telecom Conference（世界電気通信会議），パリ，２００２年に開示されたフロー認識アーキテクチャの定義である。

【００５３】

このモジュールに提示されるパケット２０は、フロー識別子フィールドの部分集合にハッシュ関数を適用することによって達成される負荷分散を含む、従来の配信機能によって決定される。ユーザによって自由に満たされるヘッダのフィールド（トランスポートプロトコルポート番号、ＩＰｖ６フローラベル）をこの部分集合に含めることで、順応的配信の一形態が得られる。

【００５４】

モジュール２４は保護フローのリスト３０を調べ、これを最新に保ち、保護フローとはようするに、手段２４によって許可され、アクティブなフローである（つまり、一定の期間内にそのフローの新しいパケットが識別された）。

【００５５】

参照番号２８は、優先順位方法又はアルゴリズムを用いるフェアキューイングに従ってパケットのキューを管理するためのスケジューリングモジュール又はスケジューリング手段を示す。

【００５６】

そのアルゴリズム又は方法においては、ビットレートが、現在のフェアビットレートに相当する閾値を超えないフローのパケットに優先順位が割当てられる。この後特定の実施の脈絡の中で指摘するとおり、この条件は、使用するフェアキューイングアルゴリズムの一定のパラメータの値の中で具体化される。着信ビットレートがフェアビットレートを超えるフローのパケットは、非優先パケットとして分類される。

【００５７】

ビットレートが高過ぎる着信フロー２０は、スケジューリングモジュール２８によって非優先フローとして分類されるので、すなわち非優先フローに降格されるので、リアルタイムフローのピークビットレートを規制することにもなる。

【００５８】

この機構はこのように、図１に示すリアルタイムフローのピークビットレートを規制するモジュール４に取って代わるものであり、それゆえモジュール４は省略することができる。

【００５９】

また、モジュール又は手段２８は、受け入れ条件データ又はパラメータ３６を配信モジュール２４に供給する。

【００６０】

最後に、参照番号３２はパケットの出発を示し、参照番号３４及び３８はそれぞれ、スケジューリングモジュールによって行われるパケットの拒絶と受け入れ制御モジュールによって行われるパケットの拒絶とを指す。

【００６１】

なお、本発明の一使用様態においては受け入れ制御モジュールは無くてもよく、その場合フロー２０は、モジュール２８に直接提示されることに留意されたい。

【００６２】

この機構では、パケットやフローの種類を、すなわちリアルタイムの部類であるとか伸縮性の部類であるとかを先験的に識別する必要はない。

【００６３】

図２のアーキテクチャは普通、ルータの中で実施されるかそこに設けられ、あるいはルータのためのプロセッサやマイクロプロセッサの中で実施されるかそこに設けられ、要求された機能を実施する形にプログラムされる。

【００６４】

以下、配信決定モジュール２４の一実施の形態をさらに詳しく述べる。従来のＩＰネットワークでは、輻輳の状態に関わりなく目的地ＩＰアドレスによって決まる単一のパス上でフローが配信される。本発明のアーキテクチャでは、配信決定モジュール２４が１つ又は複数の保護フローのリスト３０に含まれる情報に基づいて、所与のフローのパケットを配信するか否かを決定する。

【００６５】

保護フローのリスト３０はフロー識別子のリストであり、最後のパケットの到着時間をフローごとに指摘する。それぞれのリストには識別子空間の区画が付随する。この区画によって各リストの容量が制限され、ゆえに拡張性が保証される。

【００６６】

フローの最後のパケットが受信されてから経過した時間が閾値もしくはタイムアウトを超えると、そのフローはリスト３０から消去される。タイムアウトはシステムのパラメータであり、例えば２～３秒程度である。表のサイズは、飽和が起こる可能性、すなわちフローをリストに入れるべきであるがリストが満杯である状態が起こる可能性を抑えられるようなサイズに設定するのが好ましい。たとえ飽和が起きても、フローが保護フローの地位を獲得するのが遅れるに過ぎない。輻輳の状態が許されるなら、そのパケットは正確に配信される。十分なサイズに設定すれば、飽和が起こる確率を十分に低く抑えることができる。

【００６７】

配信の決定はパケットの到着時に下される。モジュール２４は、パケットのヘッダからの情報に基づいて出力インターフェースと適切なリストを共同で決定する。保護されたフローに属するパケットはそのまま配信される。リストの中では最後のパケットの到着時間が更新される。フローがまだ保護されていない場合は、配信決定を行わなければならない。

【００６８】

受け入れ条件が満たされないとパケットは拒絶される。以下、これらの条件の性質を説明する。適用される条件は、トラフィックの等級フィールド（ＩＰｖ６）やＴｏＳフィールド（ＩＰｖ４）の値、発信元ＩＰアドレスや目的地ＩＰアドレスなど、パケットの属性に応じて異なる。

【００６９】

受け入れ制御は、受け入れを許可されたフローの透過性を保証する。つまり、リアルタイムフローの信号の保存と、データフローのビットレートの保存が保証される。実際には、（外的制限によって決まる）ピークビットレートが一定の閾値を常に下回るフローに限り、この透過性が提供される。この許容ビットレート閾値は、データフローのそれよりもリアルタイムフローのそれのほうが低い。なぜなら、データフローとリアルタイムフローの場合とで、それぞれの透過性保証（すなわち、ビットレートの保存と信号の保存）は異

なる時間尺度を暗示するからである。リアルタイムフローにとっての目的は、微々たるパケット遅延（数ミリ秒）に匹敵する短い時間尺度で利用可能ビットレートがピークビットレートを上回ることを保証することである。データフローにとっての目的は、（数秒程度の）比較的長い期間にわたって一定の平均ビットレートに接近させることである。

【００７０】

問題の２つのピークビットレートは選択する受け入れ基準によって決まる。換言すれば、一定のビットレートを持つフローの透過性を保証するためには、フェアビットレートと優先順位負荷輻輳の測定値に基づいて適切な受け入れ条件を設定する。すべてのフローが伸縮性である場合、フェアビットレートのための適切な閾値の選択は、例えばBen Fredjらが説明するとおりである（Measurement based admission Control for Elastic Traffic, Teletraffic Engineering in the Internet Era（ＩＴＣ１７のインターネット時代の通信トラフィックエンジニアリングの伸縮性トラフィックのための測定ベースの受け入れ制御））。ｔｈｒｅｓｈｏｌ＿１（Ｔ１）は問題の閾値を指す。すべてのフローが限られたピークビットレートを持つリアルタイムの部類であるなら、Gibbensらのアプローチ「A decision theoretic approach to Call Admission in ATM Networks（ＡＴＭネットワークにおけるコール受け入れに対する決定理論アプローチ）」，IEEE Ｊ. on Selected Areas in Communication, １３巻、第６号、ｐ．１１０１－１１１４，１９９５年）を用いて優先負荷受け入れ閾値を導き出すことができる。ｔｈｒｅｓｈｏｌｄ＿２（Ｔ２）は第２の閾値を指す。

【００７１】

図３は、例示として、これら２つの閾値によって定義される受け入れ領域を示す。フェアビットレートがＴ１（グラフの領域Ｉ）を下回る場合、又は優先負荷がＴ２を上回り且つフェアビットレートがＴ２（領域ＩＩ）を下回る場合、新たなフローは拒絶される。なぜなら、優先負荷が閾値Ｔ２を超えるにもかかわらずフェアビットレートがその閾値よりも高い場合、それは、優先パケットとしてカウントされるパケットが、ピークビットレートが制限されたリアルタイムフローからのパケットばかりでないことを示唆するからである。この受け入れ可能領域の定義は、試行錯誤を通じて洗練されるであろう。

【００７２】

論理関数Ａｄｍｉｔは（フェアビットレート優先負荷）、Ａｄｍｉｔが値１を持つ場合には新たなフローを受け入れることができ、Ａｄｍｉｔが値０を持つ場合には新たなフローを拒絶する形に定義する。フローの性質を先験的に推定することはできないので、受け入れ方針の簡単な例を与えるため、すべてのフローは同じ条件の下で受け入れられるか拒絶される。したがって最悪の場合の仮定を適用し、新たなフローが最も要求的で最も妨害的であるとみなす。この方針は、ブロッキングの観点から、すべてのフローが公平に処理されることを保証する。

【００７３】

受け入れ条件に従ってパケットを拒絶してはならない場合、そのパケットは対応するリンクを介して配信される。そのフローの識別情報はリスト３０に登録すべき候補となる。最終的にそのフローの登録を許可するための補足条件を定めることができる。具体的には、その決定は確率的である。フローが追加される確率はｐであり、パケットが配信されてもフローは保護されない確率はｐ－１である。トラフィックの等級フィールドやＴｏＳフィールドの値、ＩＰアドレスなど、確率ｐはパケットの属性によって左右される。

【００７４】

ｐが低い（例えば０．１）場合、ほとんどの小さなフローは考慮されず、大きなフローは、最初の２０から３０パケットが送信された時点で直ちに保護される。ｐ＜１を選択するにあたってのひとつ問題は、進行しているフローの拒絶の可能性である。つまり、初期のパケットが送信されても、受け入れ制御の決定によってフローが食い止められて保護フローの地位に達することができない状態である。

【００７５】

上述の受け入れ条件は、スケジューリング装置又は手段２８の中で行われる輻輳測定に

依存する。例えば２つの指標、具体的にはフェアビットレートと優先負荷を測定する。すなわち、フェアビットレートは、送信すべきパケットを常に有するデータフローによって達成されるビットレートの測定値であり、優先負荷は、一定の期間内に送信される優先パケットの長さの和をその期間の持続時間で割って求めた値である。

【００７６】

フェアビットレートを推定するために、スケジューリングアルゴリズムは、ダミーパケットのフローと、フローが常に送信するパケットを有すると仮定して測定される、フローが割当てられたビットレートを含むことができる。勿論、ダミーパケットは実際には送信されない。ここに２つの状況が存在する。すなわち、複数の待機パケットを持つフローが存在する場合、ダミーフローはそのパケットを、それらのフローの送信間に周期的に挿入し、優先パケットの通過を許し、キューが空の場合（待機する実パケットがない）、ダミーフローは理論上リンクのビットレートに従属する。測定期間の終了時に行われるフェアビットレートの計算では、これらの無活動期間を考慮に入れる。

【００７７】

監視されるリンクの負荷状態は、周期的な測定を実行することによって連続的にカウントされる。この測定期間は普通、（例えば１００ミリ秒程度の）フェアビットレートと（例えば１０ミリ秒程度の）優先負荷とで異なる。

【００７８】

例えば、受け入れ制御のために用いる推定値は、１サイクルあたりの測定値の指数的に平滑化された平均値である（つまり、０＜(＜１であれば、推定量((×推定量＋（１－()×新しい測定値）。

【００７９】

次に、スケジューリングモジュール２８の一実施の形態をさらに詳述する。使用されるスケジューリングはフェアキューイングタイプのものが好ましい。

【００８０】

ＳＦＱと呼ばれるこの種のスケジューリングは、例えばGoyalらによる論文「Start-time Fair Queuing：A scheduling algorithm for integrated services packet switching network（スタートタイムフェアキューイング：統合サービスパケットスイッチングネットワークのためのスケジューリングアルゴリズム）」，IEEE/ACM Trans. Networking，第５巻，第５号，ｐ．６９０－７０４、１９９７年において説明されている。

【００８１】

このタイプのスケジューリングは、ユーザの協働的挙動を省みることなく、現下のフロー間でリンクの帯域幅を公平に分配する。これに受け入れ制御が組み合わさることで、受け入れを許可されたフローについてはビットレートの保証が追加的に提供される。

【００８２】

本発明のアーキテクチャは、受け入れ制御によってフローの数が必然的に制限されることを前提に、キューがアクティブである間に限りキューの状態の保守することによって、各フローのフェアキューイングに関連する拡張性の問題を回避する。

【００８３】

フローのリストに登録されていないという意味でフローが“アクティブ”でないときに到着するパケットには優先順位が与えられる。これは、そのフローのパケットがキューの中になく、なおかつそのフローのパケットが（ＳＦＱアルゴリズムの現在のパラメータによって決定する）一定の期間にわたってキューの中になかったことを意味する。原則として、ピークビットレートがフェアビットレートを超えないフローは、そのフローのパケットが到着した時点でアクティブでなくなる。

【００８４】

この順応と受け入れ制御とが組み合わさることで、ビットレートが一定の閾値を超えないリアルタイムフローについては、無視できるほど微々たる劣化が保証される。

【００８５】

このアーキテクチャはリアルタイムフローとデータフローとを区別しなので、スケジュ

ーリング装置はデータフローのパケット（例えばビットレートがフェアビットレートを常に下回るフローの全パケット）にも優先順位を与える。

【００８６】

優先パケットについては、無視できるほど微々たる遅延が受け入れ制御によって保証されるので、この曖昧さは問題にならない。

【００８７】

出力キューイング装備ルータとして知られる、キューイングが出力のみで作動するルータの例を参照しながら、スケジューリング装置の動作を説明する。

【００８８】

タイムスタンプの値によって決まる任意の位置でパケットを受け入れるプッシュイン、ファーストアウト（ＰＩＦＯ）キューイングシステムを使用する。そのキューは常にその先頭から空になる、つまりキューの先頭にあるパケットが送信される。

【００８９】

この例の装置は以下のデータや要素を使用する。

・ＰＩＦＯキュー：このキューの中ではパケットがタイムスタンプの昇順で格納される。ＰＩＦＯの要素は｛ｐａｃｋｅｔ、ｔｉｍｅ＿ｓｔａｍｐ｝セットであり、ｐａｃｋｅｔはパケットに関する情報（フロー識別子、サイズ、格納アドレス）を指定し、ｔｉｍｅ＿ｓｔａｍｐは（後述するＳＦＱアルゴリズムによって決定される）タイムスタンプである。

・ＰＩＦＯキューの先頭で最後の優先パケットを識別するポインタＰ。ファイルの中に優先パケットがない場合はＰ＝０（ゼロ）である。

・アクティブフローの識別子を含むｆｌｏｗ＿ｌｉｓｔと、最後のパケットのｔｉｍｅ＿ｓｔａｍｐ値にその長さを加えた値に相当するｆｌｏｗ＿ｔｉｍｅ＿ｓｔａｍｐ（これはＳＦＱにおけるパケットの“終了タグ”）。

・タイムスタンプを計算するためのｖｉｒｔｕａｌ＿ｔｉｍｅカウンタ。

【００９０】

この仮想時間は、例えばすでに言及したＳＦＱアルゴリズムの場合なら、送信されようとする最後のパケットのタイムスタンプに等しい（“開始タグ”）。

【００９１】

輻輳の推定量には以下のデータを使用する。

・現地時間に従うｌｏｃａｌ＿ｔｉｍｅ。

・現在の測定期間中に送信される優先パケットのバイト数、すなわちｐｒｉｏｒｉｔｙ＿ｂｙｔｅｓ。

・キューが空であるか否かを指摘する論理変数ｓｉｌｅｎｃｅ＿ｆｌａｇ。

・サイレント期間の累積持続時間ｓｉｌｅｎｃｅ＿ｔｉｍｅ。

・現在のサイレント期間の開始時間ｓｉｌｅｎｃｅ＿ｓｔａｒｔ。

【００９２】

図４はＰＩＦＯキュー５０を図式的に示す。パケット５２は、そのタイムスタンプに応じてその中に挿入される。優先パケット５６はキューの先頭にあるパケットの中に挿入され、パケット５４は次に送信されるべきパケットである。パケット５６は最後の優先パケットとして挿入され、キューへの挿入後にポインタＰが指すパケットである。なお、優先パケットはどれも、ｖｉｒｔｕａｌ＿ｔｉｍｅの現在の値に等しい、同じタイムスタンプを受け継ぐ。

【００９３】

優先負荷を推定するには、ＰＩＦＯキューに優先パケットが挿入されるたびにｐｒｉｏｒｉｔｙ＿ｂｙｔｅｓカウンタを更新する必要がある。次に、このカウンタをパケットのサイズをバイト数で表す値Ｌで増分する。このカウンタを規則的な間隔をおいてサンプリングすることにより、測定期間の開始時におけるｐｒｉｏｒｉｔｙ＿ｂｙｔｅｓ値と終了時におけるｐｒｉｏｒｉｔｙ＿ｂｙｔｅｓ値との差を、その測定期間の持続時間で割ることで求められる値として、優先負荷の推定値を導き出す。ＰＢ（Ｔ）は時間Ｔでのｐｒｉ

ｏｒｉｔｙ＿ｂｙｔｅｓの値を指し、（Ｔ１，Ｔ２）は測定期間（秒数）を指し、Ｃはリンクのビットレート（１秒あたりのビット数）を指す。その間隔での優先負荷推定量は次のとおりである。

【００９４】

優先負荷＝（ＰＢ（Ｔ２）＝ＰＢ（Ｔ１））＊８／（Ｔ２－Ｔ１）／Ｃ。

【００９５】

上記で引用した論文のBen Fredjらによれば、この測定値が極端に高い精度で評価されないことを前提に、フェアビットレートを推定するための種々の方法がある。ひとつの方法は、アクティブフローの数（ｆｌｏｗ＿ｌｉｓｔ中のフローの数）をカウントし、リンクのビットレートをその数で割って求めた値を測定フェアビットレートとして取ることである。後述するアルゴリズムでは、上で言及したダミーフローの概念に基づき、別のアプローチを使用する。

【００９６】

ダミーフローは１ビット長のパケットを送信し、そのパケットは、ＳＦＱアルゴリズムによって指示される順序で実パケットの間に挿入されると仮定する。キューが常に占有される期間にダミーフローから送信できたバイトの数は、ｖｉｒｔｕａｌ＿ｔｉｍｅカウンタの漸進的変化から導き出す。キューが空のときには、リンクのビットレートで送信できる。連続する占有期間とサイレント期間を結合することにより、以下の推定量を導き出す。ＶＴ（Ｔ）は時間Ｔでのｖｉｒｔｕａｌ＿ｔｉｍｅ値を指し、（Ｔ１，Ｔ２）は測定期間を指し、Ｓはその期間中のサイレントの総持続時間を指す。選択される推定量は次のとおりである。

【００９７】

フェアビットレート＝ｍａｘ（Ｓ＊Ｃ／（Ｔ２－Ｔ１）、（ＶＴ（Ｔ２）－ＶＴ（Ｔ１））＊８／（Ｔ２－Ｔ１）

【００９８】

リンクの負荷が低いときにはダミーフローがリンクの残りの利用可能容量のすべてを使用するため、通常ならば最初の項が優勢となる。第２項は占有期間に優勢となり、キューの中に少なくとも１個のパケットを常に有する実フローによって達成されるビットレートを近似的に見積もる。

【００９９】

図５と図６に示すとおり、アルゴリズムの動作はパケットの到着時かパケット送出の終了時に実行される。同図において、鎖線ボックス内の動作は輻輳測定に関係し、それについては後述する。

【０１００】

図５において、アルゴリズムの始まりはパケットの到着に一致する（ステップ１００）。まずはＰＩＦＯキューが輻輳しているか否かを判定する（ステップ１０２）。輻輳している場合は、拒絶すべきパケットがあればそれを選択する（ステップ１０４）。

【０１０１】

パケットは拒絶される場合（ステップ１０６及び１０８）と拒絶されない場合とがあり、拒絶される場合は（このとき別のフローのパケットが拒絶される）、ステップ１０２に対する応答が否の場合と同様に、ステップ１１０のテストを実行する。ステップ１１０は、フローのリストの中にパケットの識別子があるか否かを検査する。

【０１０２】

パケットのフロー識別子がフローのリストの中になければ、そのパケットが属するフローはアクティブではない。この種のパケットは、フローのリスト（ｆｌｏｗ＿ｌｉｓｔ）が飽和していなければ優先順位を得る。リストの飽和が判明する場合は（ステップ１２０）、パケットが拒絶される（ステップ１０８）。

【０１０３】

飽和していなければ、ｖｉｒｔｕａｌ＿ｔｉｍｅにパケットの長さＬを加えた値に等しいｆｌｏｗ＿ｔｉｍｅ＿ｓｔａｍｐを持つ新しいフローの識別子を加えてｆｌｏｗ＿ｌｉ

ｓｔを更新する（ステップ１２４）。パケットはＰＩＦＯキューの中のポインタＰによって指示される位置に挿入され、ポインタＰの値は更新される（ステップ１２４）。

【０１０４】

アクティブフローのリストにフローがすでに登録されている場合は（ステップ１１０で肯定応答）、ｆｌｏｗ＿ｔｉｍｅ＿ｓｔａｍｐの現在値に等しいタイムスタンプを持つＰＩＦＯキューにパケットが挿入される（ステップ１２２）。次に、ｆｌｏｗ＿ｌｉｓｔの中のこの値をパケットの長さＬで増分する。

【０１０５】

次に、アルゴリズムは終了する（ステップ１３４）。新たなパケットが到着すると同じアルゴリズムが再開する。パケットの受け入れが終わると次のアルゴリズムがスタートする（図６）。

【０１０６】

鎖線ボックス内の動作は、輻輳フェアビットレートと優先負荷のパラメータを連続的に測定する。ＰＩＦＯキューが空なら（ステップ２１０）、サイレント期間の終了が観察される。ｓｉｌｅｎｃｅ＿ｔｉｍｅカウンタが更新され、論理指標ｓｉｌｅｎｃｅ＿ｆｌａｇは値ＦＡＬＳＥに設定される（ステップ２１２）。さらに、新たなフローがｆｌｏｗ＿ｌｉｓｔに追加されると、ｐｒｉｏｒｉｔｙ＿ｂｙｔｅの値が問題のパケットの長さで増分される（ステップ２０４）。

【０１０７】

図６のアルゴリズムは、他の事象を契機に、すなわちパケット送信の終了を契機にスタートする（ステップ１５０）。最初のステップ（ステップ１５４）ではＰＩＯキューが空か否かを判定する。

【０１０８】

空ならば、まだアクティブであるすべてのフローがｆｌｏｗ＿ｌｉｓｔから消去される（ステップ１５６）。ポインタＰがリセットされる。

【０１０９】

空でなければ次のパケットが送信され、そのタイムスタンプｎｅｘｔ＿ｔｉｍｅ＿ｓｔａｍｐの値に注目する（ステップ１６０）。ｎｅｘｔ＿ｔｉｍｅ＿ｓｔａｍｐがｖｉｒｔｕａｌ＿ｔｉｍｅよりも大きくなければ（これらの変数が同じ値を持つなら）、アルゴリズムは終了する。そうでない場合はｖｉｒｔｕａｌ＿ｔｉｍｅを変更し、非アクティブになったフローをｆｌｏｗ＿ｌｉｓｔから消去する必要があり、それらのｆｌｏｗ＿ｔｉｍｅスタンプはｖｉｒｔｕａｌ＿ｔｉｍｅを下回る（ステップ１６４）。次に、アルゴリズムは終了する。

【０１１０】

輻輳測定に関連する動作は鎖線ボックスの中に囲まれている。キューが空になると、サイレント期間の始まりが記録されるときにｓｉｌｅｎｃｅ＿ｆｌａｇ指標が値ＴＲＵＥに設定される（ステップ２２０）。

【０１１１】

新たなパケットの到着を契機に前述のアルゴリズム（図５）がスタートし、パケット送信の終了を契機に図６のアルゴリズムが再びスタートする。

【０１１２】

図７及び８は、輻輳カウンタのサンプリング時に実行する動作を示す。図７の動作は、（現地時間に従って）ｔｉｍｅ＿ｉｎｔｅｒｖａｌ＿ＣＰ秒ごとに実行される。優先負荷が計算され（ステップ２３２）、次の実行のためにｐｒｉｏｒｉｔｙ＿ｂｙｔｅｓカウンタの現在値が変数ｐｒｉｏｒｉｔｙ＿ｂｙｔｅｓ＿ｏｌｄとして格納される。

【０１１３】

図８の動作は、（現地時間に従って）ｔｉｍｅ＿ｉｎｔｅｒｖａｌ＿ＤＥ秒ごとに実行される。計算は、リンクがサイレント（キューが空）か否かで異なる。サイレントならば、この計算の要件としてサイレント状態を中断する（ステップ２４８）。いずれにせよ、測定期間中の使用可能ビットレートに相当するフェアビットレートの推定値ｒａｔｅ＿１

を得る（ステップ２４６又は２４８）。次に、ステップ２５０にと同じ公式を用いて第２のビットレート推定値ｒａｔｅ＿２を計算する。フェアビットレートは、２つの推定値ｒａｔｅ＿１及びｒａｔｅ＿２の内の大きい方である（ステップ２５４、２５６及び２５８）。最後に、ｓｉｌｅｎｃｅ＿ｔｉｍｅとｖｉｒｔｕａｌ＿ｔｉｍｅの現在値を次の期間のために格納する（ステップ２５２）。

【０１１４】

別のアプリケーションでは、各測定期間の終了時に読み出される値からスライディング平均値を計算する。選択された平滑化重みと測定期間は、システムとトラフィックのパラメータに応じて最適化する。このアプリケーションは、フェアビットレートと優先負荷の推定値を供給し、前に言及したＡｄｍｉｔ関数の値を導き出す。

【０１１５】

ステップ１０４（図５）では、アクティブフロー間での公平なビットレート分配が保証される形に拒絶すべきパケットを選択する。これは、拒絶にあたって総待機バイト数が最も高いフローのパケットを選択することによって確実なものとなる。拒絶の条件と拒絶すべきパケットを選択するための機構は、例えばB.Suterらの論文「Buffer Management schemes for supporting TCP in Gigabit Routers with Per-Flow Queuing（パーフローキューイングを有するギガビットルータにＴＣＰをサポートするためのバッファ管理スキーム）」, IEEE J. in Selected Areas in Communications, １９９９年８月の中で開示されている。なお、場合によっては、パケット拒絶をＩＰヘッダの明示的輻輳通知（ＥＣＮ）ビットを使用する単純なマーキングに置き換えることもできる。

【０１１６】

本発明の方法は拡張可能である。この方法は、例えば１メガビット／秒、１０メガビット／秒、１ギガビット／秒あるいはそれより高いビットレートのリンクで、負荷条件に関係なく動作する。

【０１１７】

スケジューリング機構の複雑さ、とりわけＳＦＱの複雑さは、アクティブフローの数に比例する。拡張性は、この数が受け入れ制御によって制限され、リンクＣ上のビットレートとは無関係に比較的低い数に保たれるという事実によって保証される。

【０１１８】

この数を推定するため、まずは受け入れ制御が完璧に作用していて、フェアビットレートが目標値θを決して下回らないと仮定する。（パス上の他の要素による制限を受けないため）ビットレートがθを超えるフローの数は当然、Ｃ／θ以下となる。θの値は、負荷が一定の制限を超えない場合に、Ｃ／θよりも多くのアクティブフローが出現する確率が非常に低くなるように選択する。負荷制限が９０％の場合、Ben Fredjらの分析によれば（「Statistical Bandwidth Sharing：a study of congestion at flow levels（統計的帯域幅シェアリング：フローレベルでの輻輳の考察）」, Proc. of ACM SIGCOMM 2001）、フローの数が１００を超える確率は１／１００００よりも低い。したがってθ＝Ｃ／１００と設定すれば、アクティブフローの数は常に１００未満となり、負荷が９０％を超えないなら、ブロッキングの確率は10^{-4}未満になる。

【０１１９】

ビットレートがθ未満のフローが多数進行するかもしれない。ただしそれらのフローの内、随時キューの中にパケットを持つことになるフローはごく少数である。簡潔にするため、パケットの長さはどのフローのパケットでも同じであると仮定する。これらのフローからキューの中に入るパケットは１つだけである（ビットレートがフェアビットレートを下回るため）。フローが独立している場合は、優先順位キューの中のパケットの数はＭ／Ｄ／１キューとして挙動する。同様に、（フェアキューイングアルゴリズムのフローのリストに登録される）アクティブフローの数は、同じキューの占有期間に参加しているクライアントの数に等しい。負荷が９０％を超えないなら、これが１００未満となる確率が高い。

【０１２０】

変化するパケットサイズと上流ネットワークで捕捉されるジッターの影響をもってしても、正常な負荷条件下で考慮すべきフローの数が常に２００から３００（上記の制限がすべて加わった場合は２００）未満であるという事実に変わりはない。過負荷の状況における受け入れ制御の役割はまさに、進行中のフローの数を制限することである。当然、アクティブフローの数はフェアビットレート制御によって制限される。ビットレートがフェアビットレートよりも低いフローに起因する局所的負荷が常に一定の制限（例えば９０％）を下回る確率は、優先負荷制御によって高くなる。

【０１２１】

受け入れ制御アルゴリズムの精度の欠如を見越した最大フロー数の最適値は、上述のアルゴリズムを用いて試行錯誤を通じて決定できるであろう。

【０１２２】

受け入れ制御なしで装置を使用する場合、考慮すべきフローの数は制限されない。これは、ユーザ数そのものに限りがあるアクセスネットワークの状況では問題にならないであろう。

【０１２３】

弁別は、サービス等級を受け入れ制御レベルで区別するのが望ましい。そうすれば、一定水準の輻輳で一定等級のフローを拒絶でき、優先順位が高いサービス等級のほかのフローのために容量を残しておくことができるからだ。例えば、優先順位はトラフィック等級（ＩＰｖ６）、ＴｏＳフィールド（ＩＰｖ４）、ＩＰアドレスなど、パケットのヘッダの種々のフィールドの値に依存する。

【０１２４】

そこで、ｍ個のサービス等級があり、ｉ＝１，．．，ｍに対して１組の論理関数Ａｄｍｉｔ＿ｉ（フェアビットレート、優先負荷）が存在すると仮定する。ｉ＜ｊなら、等級ｉは等級ｊよりも優先する。関数は、引数が同じならＡｄｍｉｔ＿１(Ａｄｍｉｔ＿２となる。したがって受け入れ方針は、Ａｄｍｉｔ＿１が値０を持つならインデックス等級ｉのフローを拒絶し、Ａｄｍｉｔ＿１が値１を持つならそれらを受け入れする。

【０１２５】

等級がない単純な方針におけるＡｄｍｉｔ関数のように、Ａｄｍｉｔ＿１は、ピークビットレートが目標ビットレートを下回るフローの透過性を維持するように選択する。ｉ＝２，．．，ｍに対してＡｄｍｉｔ＿１である他の関数は、優先順位等級のアクセス可能性に対して優先権を与える。

【０１２６】

ピークビットレートの制限は、トラフィック条件が許すならば、ユーザは難なくそれを超過できるという意味で、“頑な”でない。これは、仕様を決定し、正確な受け入れ条件を確立するための前提である。ピークビットレートが設定された制限に満たないなら、運営者によって通知される保証により、ユーザはリアルタイムサービスで非順応的コードを使用できる。外的な制限（アクセスネットワーク、サーバ容量、その他）が目標ピークビットレートを上回らない限り、データ転送のビットレートはネットワークによっては制限されない。

【０１２７】

本発明のアーキテクチャなら、従来の技術のアーキテクチャほどユーザ側の理不尽な挙動や悪意ある挙動による被害を被らずにすむ。ただし最適な性能のためには、パケットの損失よりなる暗黙的信号をアプリケーションが適切に解釈することが望ましい。

【０１２８】

新たなフローの最初のパケットの損失は、フローの拒絶として解釈しなければならない。基礎となるアプリケーションは、パケットの内の１つが受け入れられ、そのフローが保護フローのリストに書き込まれるまで、パケットを送り続けることができる。この再送は、ＴＣＰの損失パケットの再生ほどには問題にならない。前に言及した順応的配信の可能性は、より理知的な反応を可能にする。最初のパケットを損失すると、後続パケットのフロー識別子が変わることで別のパスをテストできる。

【０１２９】
リアルタイムアプリケーションは、パスの利用可能性を評価するためのテストパケットを送信できる。テストパケットの肯定応答により、フローが受け入れられ保護フローのリストに登録されることがわかる。テストパケットを処理するために追加の専用機構は必要ない。
【０１３０】
ほとんどのコールでは、それぞれの方法で１つずつ、全部で２つのフローが設定される。識別子の自由部分（ＩＰｖ６のラベルフローやＩＰｖ４のポート数）が両方向において同じであるとする規定を採用することは、ユーザにとって有利である。こうすれば、とりわけフラッディングによるルート指定の場合に、ユーザは特定のフローの肯定応答を認識できる。この場合ユーザは、異なる識別子を持つ複数のパケットを送出して複数のパスをテストする。そのコールは、最初に肯定応答されたフローで継続する。
【０１３１】
このアーキテクチャなら、サービス不能（ＤｏＳ）タイプの攻撃に新たなチャンスを与えずにすむ。以下の２種類の挙動が想定される。
・ユーザがパケットごとにフロー識別子を変更する。そうすると、保護フローのリストはすぐに飽和する。その結果、一部のフローは保護されなくなってしまうが、そうした事態が生じるのは、攻撃されたリンクで同時輻輳が起きる場合に限る。もっぱら低い確率ｐで、リスト３０でフローが書き込まれることで、そのような攻撃のインパクトを減少する。
・ユーザが複数のフローで同じ識別子を維持する。連続するフローは、２つのパケットの間の時間が常にタイムアウトに満たないのであれば、受け入れ制御の対象とならない。並列に送出されるフローの全体のビットレートは、フェアビットレートの現在値よりも大きくならない。いずれの場合でも、他のユーザにとっての不都合は微々たるものである。
【０１３２】
最後に、ユーザが同じアプリケーションを搬送するために複数のフローを設定することは勿論可能であり、その場合、ピークビックレートが許すならば、より高いビットレートが得られる。
【０１３３】
現在のルータのほとんどは、等しいコストのパスで負荷均衡を達成するため、発信元及び／又は目的地ＩＰアドレスにハッシュ関数を適用することで、所与のパケットの出力インターフェースを選択する。したがって、同じフローのパケットはどれも同じパスを取る。ハッシュ関数の引数は、フロー識別子の自由部分（すなわち、ＩＰｖ６におけるラベルフローやＩＰｖ４におけるポート数）も含む形に拡張できる。このように、ルートの選択はユーザが指定する値によって左右される。ルートで障害が発生したら、ユーザは自由に識別子を変更し、再び試みることができる。１度か複数回試みれば、十分な容量を持つルートが見つかるであろう。複数のフローが同時に始まることもあり、そのコールは、それらのフローの内の１フローのみで、例えば肯定応答が最初に受信されたフローで継続する。
【０１３４】
本発明のアーキテクチャを実施するネットワーク要素は、例えばハッシュ関数をアドレス属性に適用することによってフローを識別できる。このハッシュ関数は、実施の複雑さと、関数が同じ値を送り返す２つのフロー間で起きる混乱の可能性との間で妥協を達成することが好ましい。このような混乱の影響には限りがある。その場合、受け入れ制御によるブロッキングの回避、フェアキューイングによるキューに割り当てられるビットレートの減少が適用可能である。
【０１３５】
本発明においては、受け入れ制御がリンクの輻輳時に到着するフローを、その最初のパケットを拒否することによって拒絶するので、フロー識別子に基づく順応的配信のために負荷分散機構を使用する。ユーザが成功するまで同じパケットを再送することによって再

び試みるなら、対応するトラフィックは失われない。試みるたびに、パケットをその目的地まで配信できる別のリンクにパケットが提示されるなら、成功の確率はより大きくなる。ルータで採用される負荷分散機構がフロー識別子に基づいて動作し、そのフロー識別子に、ユーザによって自由に書き込まれるフィールド（例えばＩＰｖ４におけるポート番号、ＩＰｖ６におけるフローラベル）が含まれるとすれば、これを達成することができる。また、負荷分散を実施するハッシュ関数がこの自由フィールドを引数として含むとすれば、ユーザは更新のたびに問題のフィールドの値を変更することによって、順応的配信の一形態を果たすことになる。

【図面の簡単な説明】

【０１３６】

【図１】フロー認識ネットワーキングアーキテクチャの要素を示す図である。

【図２】本発明のアーキテクチャの要素を示す図である。

【図３】受け入れ領域図である。

【図４】ＰＩＦＯ（プッシュイン、ファーストアウト）キューの原理を示す図である。

【図５】パケットの到着時、又はパケットの送信の終了時に実行されるアルゴリズムを示す図である。

【図６】パケットの到着時、又はパケットの送信の終了時に実行されるアルゴリズムを示す図である。

【図７】輻輳、優先順位負荷及びフェアビットレート指標を測定する動作を示す図である。

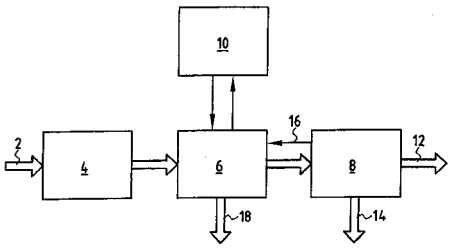
【図８】輻輳、優先順位負荷及びフェアビットレート指標を測定する動作を示す図である。

【符号の説明】

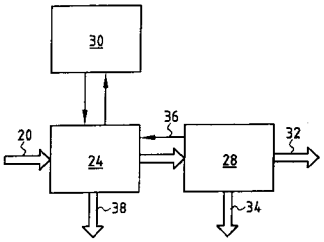
【０１３７】

２０　パケット
２４　配信モジュール
２８　スケジューリングモジュール
３０　保護フローリスト
３２　パケットの出発
３４、３６　パケットの拒絶

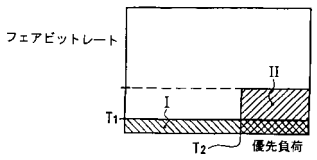
【図１】



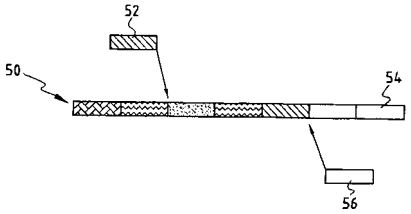
【図２】



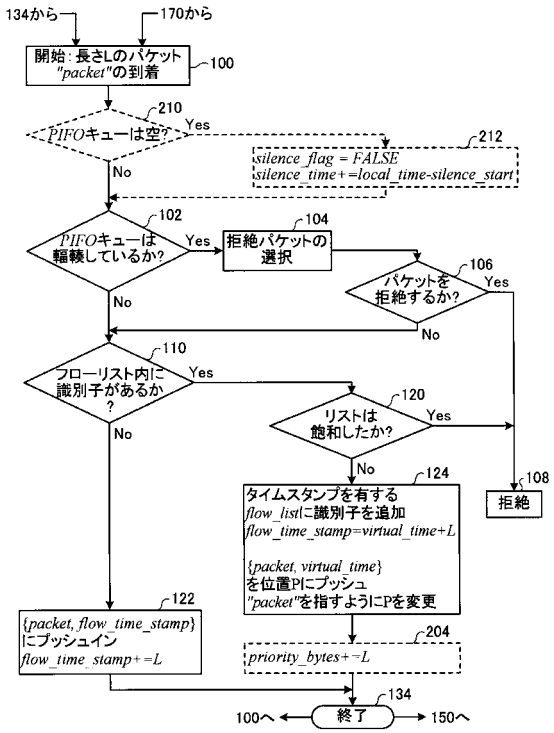
【図３】



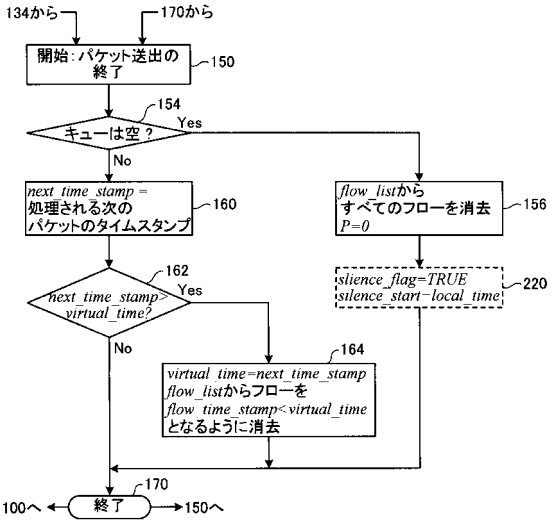
【図４】



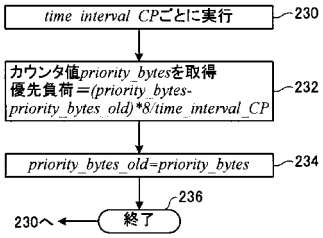
【図５】



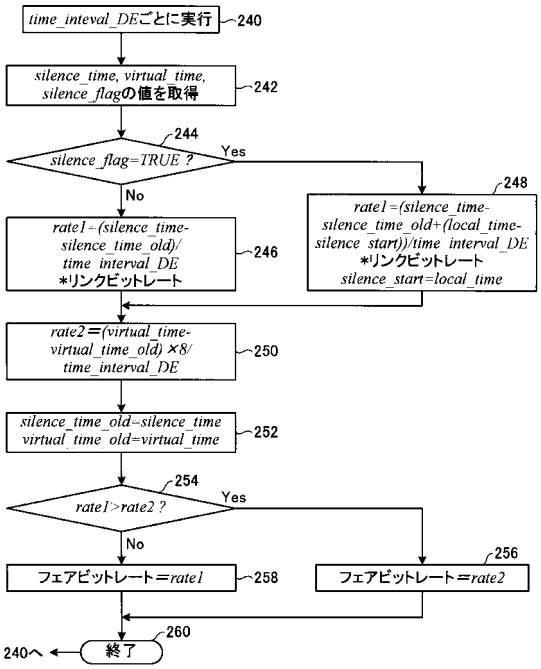
【図６】



【図７】


time_interval_CPごとに実行
230
カウンタ値priority_bytesを取得
優先負荷＝(priority_bytes-
priority_bytes_old)*8/time_interval_CP
232
priority_bytes_old=priority_bytes
234
236
230へ
終了

【図８】


time_inteval_DEごとに実行
240
silence_time, virtual_time,
silence_flagの値を取得
242
244
silence_flag=TRUE ?
Yes
No
248
rate1＝(silence_time-
silence_time_old)/
time_interval_DE
*リンクビットレート
246
rate1＝(silence_time-
silence_time_old+(local_time-
silence_start))/time_interval_DE
*リンクビットレート
silence_start=local_time
rate2＝(virtual_time-
virtual_time_old)×8/
time_interval_DE
250
silence_time_old=silence_time
virtual_time_old=virtual_time
252
254
rate1>rate2 ?
Yes
No
256
フェアビットレート＝rate1
258
フェアビットレート＝rate2
260
240へ
終了

フロントページの続き

(56)参考文献 特開２００１－１９７１１０（ＪＰ，Ａ）
特開２００１－２８５３５２（ＪＰ，Ａ）
特開２００２－１１１７１４（ＪＰ，Ａ）
特表２００２－５１８９３６（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｈ０４Ｌ　１２／５６