(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第3953819号
(P3953819)
(45)発行日　平成19年8月8日(2007.8.8)　(24)登録日　平成19年5月11日(2007.5.11)

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

請求項の数 10　(全 20 頁)

(21)出願番号	特願2002-601(P2002-601)
(22)出願日	平成14年1月7日(2002.1.7)
(65)公開番号	特開2002-232469(P2002-232469A)
(43)公開日	平成14年8月16日(2002.8.16)
審査請求日	平成16年11月2日(2004.11.2)
(31)優先権主張番号	60/260807
(32)優先日	平成13年1月10日(2001.1.10)
(33)優先権主張国	米国(US)
(31)優先権主張番号	10/008425
(32)優先日	平成13年11月13日(2001.11.13)
(33)優先権主張国	米国(US)

(73)特許権者　596092698
ルーセント　テクノロジーズ　インコーポレーテッド
アメリカ合衆国．０７９７４－０６３６　ニュージャーシィ，マレイ　ヒル，マウンテン　アヴェニュー　６００
(74)代理人　100064447
弁理士　岡部　正夫
(74)代理人　100085176
弁理士　加藤　伸晃
(74)代理人　100106703
弁理士　産形　和央
(74)代理人　100096943
弁理士　臼井　伸一

最終頁に続く

(54)【発明の名称】スケジューリング装置およびスケジューリング方法

(57)【特許請求の範囲】
【請求項１】
複数のデータパケットフローについてデータパケットの送信をスケジューリングするスケジューリング装置において、
前記データパケットフローには、通信リンクの送信容量ｒのうちの所定シェアが割り当てられ、前記データパケットフローは複数のバンドルにまとめられ、
前記バンドルには、前記通信リンクの処理容量のうちのサービスシェアが割り当てられ、
通信リンクを通じての送信はサービスフレームに分けられ、１サービスフレームは、あらゆるバックログデータパケットフローに少なくとも１つの送信機会を提供し、バックログデータパケットフローは、複数のパケットキューのうちの対応するパケットキューに格納された少なくとも１つのデータパケットを有するデータパケットフローであり、
前記スケジューリング装置は、
サービスフレームの期間を決定する手段と、
各データパケットフローが、常に少なくとも該データパケットフローに割り当てられたサービスシェアを受けるとともに、各バンドルが、１サービスフレームの全期間中連続してバックログのあるバンドル内に少なくとも１つのデータパケットフローがある場合に、少なくとも該バンドルに割り当てられたサービスシェアを受けることを保証する手段とを有し、
前記保証する手段は、

各バンドルＩについて、同じバンドルＩにまとめられた前記データパケットフローのそれぞれに割り当てられる前記サービスシェアの総和に関係する累積シェアΦ_Iを管理する手段と、

各バンドルＩについて、該バンドルＩに割り当てられるサービスシェアR_Iと、該バンドルの前記累積シェアΦ_Iとの間のサービス比を計算する手段と、

前記複数のバンドルのそれぞれについて計算されたサービス比を用いて、前記複数のデータパケットフローのそれぞれに割り当てられる前記サービスシェアを修正する手段とを有することを特徴とするスケジューリング装置。

【請求項２】

重み付けラウンドロビン（ＷＲＲ）アルゴリズム、不足ラウンドロビン（ＤＲＲ）アルゴリズム、および余剰ラウンドロビン（ＳＲＲ）アルゴリズムを含む群から選択されるアルゴリズムが、データパケットの送信をスケジューリングするために用いられることを特徴とする請求項１記載のスケジューリング装置。

【請求項３】

前記サービスフレームの期間を決定する手段は、

グローバルフレームカウンタＦＲＭＣＮＴと、

前記複数のバンドルの各バンドルＩに対する開始フラグσ_Iと、

前記複数のデータパケットフローの各データパケットフローｉに対するフレームフラグFF_iとを有し、

全てのデータパケットについて、前記開始フラグと前記グローバルフレームカウンタとを比較することにより、データパケットが現サービスフレーム内でサービスを受ける権利がまだあるか否かが決定されることを特徴とする請求項１記載のスケジューリング装置。

【請求項４】

前記グローバルフレームカウンタＦＲＭＣＮＴは、全てのサービスフレームの終わりにおいて値がトグルされるブールフラグであり、バンドルＩの開始フラグσ_Iは、該バンドル内の最初のデータパケットフローにバックログが生じたときに、グローバルフレームカウンタＦＲＭＣＮＴに等しくセットされることを特徴とする請求項３記載のスケジューリング装置。

【請求項５】

データパケットフローｉのフレームフラグFF_iは、該フローにバックログが生じるか、または、該フローが現サービスフレーム内で最後に処理されたフローであるときに、グローバルフレームカウンタＦＲＭＣＮＴとは異なる値にセットされることを特徴とする請求項３記載のスケジューリング装置。

【請求項６】

サービスフレームの終了と次のサービスフレームの開始は、処理されるべき次のデータパケットフローｉのフレームフラグFF_iがグローバルフレームカウンタＦＲＭＣＮＴとは異なる値を有するときに、同時に検出されることを特徴とする請求項３記載のスケジューリング装置。

【請求項７】

バンドルＩの累積シェアΦ_Iの値は、バックログを生じているバンドルＩのデータパケットフローのサービスシェアの総和に等しいことを特徴とする請求項１記載のスケジューリング装置。

【請求項８】

バンドルＩの累積シェアΦ_Iの値は、該バンドルの最初のデータパケットフローがサービスフレーム内で最初にサービスされるときにセットされ、たとえバンドルＩの１個または複数のデータパケットフローのバックログ状態が該サービスフレームの期間中に変化しても、同じサービスフレームの期間全体で不変のまま保持されることを特徴とする請求項１記載のスケジューリング装置。

【請求項９】

現在シェアϕ_Iは、バンドルＩ内に蓄積されたデータパケットフローのサービスシェア

の合計を維持し、そしてバンドル内の１または複数のデータパケットフローのバックログ状態が変化するときに変化し、前記現在シェアの値は全ての新たなサービスフレームの開始時に前記累積シェアΦ_Iの値を設定するのに使用されることを特徴とする請求項８記載のスケジューリング装置。

【請求項１０】

複数のデータパケットフローについてデータパケットの送信をスケジューリングするスケジューリング方法において、

前記データパケットフローには、出通信リンクの送信容量のうちの所定シェアが割り当てられ、前記データパケットフローは複数のバンドルにまとめられ、

前記バンドルには、前記出通信リンクの送信容量ｒのうちのサービスシェアが割り当てられ、

通信リンクを通じての送信はサービスフレームに分けられ、１サービスフレームは、あらゆるバックログデータパケットフローに少なくとも１つの送信機会を提供し、バックログデータパケットフローは、複数のパケットキューのうちの対応するパケットキューに格納された少なくとも１つのデータパケットを有するデータパケットフローであり、

前記方法は、

サービスフレームの期間を決定するステップと、

各データパケットフローが、常に少なくとも該データパケットフローに割り当てられたサービスシェアを受けるとともに、各バンドルが、１サービスフレームの全期間中連続してバックログのあるバンドル内に少なくとも１つのデータパケットフローがある場合に、少なくとも該バンドルに割り当てられたサービスシェアを受けることを保証するステップと、

各バンドルＩについて、同じバンドルＩにまとめられた前記データパケットフローのそれぞれに割り当てられる前記サービスシェアの総和に関係する累積シェアΦ_Iを管理するステップと、

各バンドルＩについて、該バンドルＩに割り当てられるサービスシェアR_Iと、該バンドルの前記累積シェアΦ_Iとの間のサービス比を計算するステップと、

前記複数のバンドルのそれぞれについて計算されたサービス比を用いて、前記複数のデータパケットフローのそれぞれに割り当てられる前記サービスシェアを修正するステップとを有することを特徴とするスケジューリング方法。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

本発明は、パケットスケジューラに関し、特に、個々のデータソースへ、および、それらのデータソースの集合(aggregate)へのデータ転送レートを保証するパケットスケジューリング装置および方法に関する。

【０００２】

【従来の技術】

Integrated Service (Intserv)［後記の［付録］に掲げた文献１参照］やDifferentiated Service (Diffserv)［２］のような精緻なサービス品質（ＱｏＳ）フレームワークがますます広まるとともに、フレキシブルな帯域管理の可能なパケットスケジューラが重要視されている。従来のいくつかのパケットスケジューラは、すぐれた最悪の場合の遅延性能を提供するとともに、正確な帯域保証を提供する［３，４，５，６，７，８］が、それらのコストはかなり高い［６，７，８］。

【０００３】

この理由のため、産業界は、重み付けラウンドロビン（ＷＲＲ：Weighted Round Robin）スケジューラ［９，１０，１１］に相当の関心を示している。ＷＲＲスケジューラは、ぎりぎりまでの（タイトな）遅延限界を必ずしも達成するものではないが、最小限の複雑さで頑強（ロバスト）な帯域保証を提供する。このようなスケジューラのさまざまな実例が文献に現れている。よく知られた例には、不足ラウンドロビン（ＤＲＲ：Deficit Round

Robin）アルゴリズム［１０］および余剰ラウンドロビン（ＳＲＲ：Surplus Round Robin）［１１］アルゴリズムがある。

【０００４】

【発明が解決しようとする課題】

ＷＲＲスケジューラの上記の実例は、異種のデータパケットフローの帯域保証を差別化することには成功している。しかし、それらが現在規定されている形態に関しては、新しいサービス品質フレームワークのすべての帯域要求を満たすには十分でない。一般に、ネットワークノードにおけるフレキシブルな帯域管理は、帯域が、個々のフローに割り当てられることが可能であるだけでなく、それらのフローの集合にも割り当てられることが可能であるような、階層的スケジューリング構造の展開を必要とする。従来のＷＲＲスケジューラでは、帯域分離を達成するための階層構造の重ね合わせは、基本のスケジューラの単純さを損なう。

【０００５】

データパケットネットワークのためのスケジューリングの技術分野においては、基本スケジューラの単純さを損なわずに階層的帯域分離を達成する改良されたスケジューラ装置が必要とされている。

【０００６】

【課題を解決するための手段】

本発明の目的は、階層的帯域分離のためのスケジューラ装置を規定することである。この新規な技術の目標は、完全に透過的な方法で、すなわち、追加のスケジューリング構造を使用せずに、帯域保証を、個々のデータパケットフローに提供するとともに、それらのフローの集合（「バンドル」という）にも提供することである。

【０００７】

各バンドルについて、スケジューラは、そのバンドルに名目上割り当てられる帯域と、そのバンドルで現在積滞している（バックログのある）フローの帯域割当ての総和との間の比を決定する。スケジューラは、その比を用いて、個々のフローへの帯域の分配を規制するタイムスタンプ増分を修正する。このようにして、そのバンドルに対する比が大きいほど、そのバンドル内のそれぞれのバックログフローが受ける帯域は大きくなる。スケジューラは、常に、フローが積滞している限り、与えられたデータパケットフローの帯域要求を満たし、バンドル内に少なくとも１つのバックログフローがある限り、与えられたバンドルの帯域要求を満たす。

【０００８】

具体的には、本発明のスケジューリング装置は、複数のデータパケットフローについてデータパケットの送信をスケジューリングする。前記データパケットフローには、通信リンクの送信容量ｒのうちの所定シェアが割り当てられ、前記データパケットフローは複数のバンドルにまとめられる。前記バンドルには、前記通信リンクの処理容量のうちのサービスシェアが割り当てられる。通信リンクを通じての送信はサービスフレームに分けられる。１サービスフレームは、あらゆるバックログデータパケットフローに少なくとも１つの送信機会を提供する。バックログデータパケットフローは、複数のパケットキューのうちの対応するパケットキューに格納された少なくとも１つのデータパケットを有するデータパケットフローである。本発明のスケジューリング装置は、（１）サービスフレームの期間を決定する手段と、（２）各データパケットフローが、十分な数の連続するサービスフレームにわたり連続してそのデータパケットフローにバックログがある場合に、常に少なくともその割り当てられたサービスシェアを受けるとともに、各バンドルが、十分な数の連続するサービスフレームにわたり１サービスフレームの全期間中連続してバックログのあるバンドル内に少なくとも１つのデータパケットフローがある場合に、少なくともその割り当てられたサービスシェアを受けることを保証する手段とを有する。前記保証する手段は、（Ａ）各バンドルＩについて、同じバンドルＩにまとめられた前記データパケットフローのそれぞれに割り当てられる前記サービスシェアの総和に関係する累積シェアΦ_Iを管理する手段と、（Ｂ）各バンドルＩについて、該バンドルＩに割り当てられるサービ

スシェアR_Iと、該バンドルの前記累積シェアΦ_Iとの間のサービス比を計算する手段と、（Ｃ）前記複数のバンドルのそれぞれについて計算されたサービス比を用いて、前記複数のデータパケットフローのそれぞれに割り当てられる前記サービスシェアを修正する手段とを有する。

【０００９】

本発明はまた、複数のデータパケットフローについてデータパケットの送信をスケジューリングする方法に関する。前記データパケットフローには、出通信リンクの送信容量のうちの所定シェアが割り当てられ、前記データパケットフローは複数のバンドルにまとめられる。前記バンドルには、前記出通信リンクの送信容量ｒのうちのサービスシェアが割り当てられる。通信リンクを通じての送信はサービスフレームに分けられる。１サービスフレームは、あらゆるバックログデータパケットフローに少なくとも１つの送信機会を提供する。バックログデータパケットフローは、複数のパケットキューのうちの対応するパケットキューに格納された少なくとも１つのデータパケットを有するデータパケットフローである。この方法は、（１）サービスフレームの期間を決定するステップと、（２）各データパケットフローが、十分な数の連続するサービスフレームにわたり連続してそのデータパケットフローにバックログがある場合に、常に少なくともその割り当てられたサービスシェアを受けるとともに、各バンドルが、十分な数の連続するサービスフレームにわたり１サービスフレームの全期間中連続してバックログのあるバンドル内に少なくとも１つのデータパケットフローがある場合に、少なくともその割り当てられたサービスシェアを受けることを保証するステップと、（３）各バンドルＩについて、同じバンドルＩにまとめられた前記データパケットフローのそれぞれに割り当てられる前記サービスシェアの総和に関係する累積シェアΦ_Iを管理するステップと、（４）各バンドルＩについて、該バンドルＩに割り当てられるサービスシェアR_Iと、該バンドルの前記累積シェアΦ_Iとの間のサービス比を計算するステップと、（５）前記複数のバンドルについて計算されたサービス比を用いて、前記複数のデータパケットフローのそれぞれに割り当てられる前記サービスシェアを修正するステップとを有する。

【００１０】

【発明の実施の形態】

図１に、複数の通信スイッチ１０１－１～１０１－ｐが通信リンクにより相互に接続された例示的なパケットネットワークを示す。いくつかのデータソース１０２－１～１０２－ｑが通信スイッチに接続される。ネットワークコネクションが、それぞれのデータソースから、対応する宛先１０３－１～１０３－ｇへと確立され、データパケットは、各データソースから対応する宛先へ送信される。

【００１１】

図２は、パケットネットワークの通信スイッチ１０１－１の例示的なブロック図である。図示のように、通信スイッチは、スイッチファブリック２５０と、複数の通信リンクインタフェース２００－１～２００－ｓを有する。各通信リンクインタフェースは、複数の入力リンクを１つの出力リンクに接続し、データパケットを入力リンクから出力リンクへ転送する。通信スイッチ１０１－１は、このような通信リンクインタフェース２００を１個だけ含むことも、複数含むことも可能である。例えば、入力通信リンクインタフェース２００－１はスイッチファブリック２５０の前に配置され、その場合、入力通信リンクインタフェース２００－１の入力リンク２０１－１～２０１－ｒは通信スイッチ１０１－１の入力リンクであり、入力通信リンクインタフェース２００－１の出力リンク２０３はスイッチファブリック２５０に接続される。

【００１２】

第２の例として、出力通信リンクインタフェース２００－ｊはスイッチファブリック２５０の出力に配置され、出力通信リンクインタフェース２００－ｊの入力リンクはスイッチファブリック２５０の複数の出力リンク２０４であることが可能であり、出力通信リンクインタフェース２００－ｊの出力リンクは通信スイッチ１０１－１の出力リンク２０２－ｊである。注意すべき点であるが、１つの特定のリンクを通じて、または、複数の相異な

るリンクを通じて受信されるパケットは、同じ長さであることもそうでないこともある。例えば、スイッチファブリック２５０が非同期転送モード（ＡＴＭ）スイッチであり、図１のネットワークがＡＴＭネットワークである場合、すべてのパケットは同じ長さを有する。本発明の以下の説明では、１つの特定のリンクを通じて、または、複数の相異なるリンクを通じて受信されるパケットは、長さが必ずしも同じでないと仮定する。

【００１３】

図６に関連して後の段落で説明するように、図２の通信リンクインタフェース２００のそれぞれは一般に、少なくともパケット受信器、スケジューラ、およびパケット送信器を有する。前述のように、スケジューラは、重み付けラウンドロビン（ＷＲＲ）スケジューラ［９，１０，１１］であることが可能であり、さらにこれは、不足ラウンドロビン（ＤＲＲ：Deficit Round Robin）［１０］または余剰ラウンドロビン（ＳＲＲ：Surplus Round Robin）［１１］のアルゴリズムに従って実装されることが可能である。

【００１４】

不足ラウンドロビン（ＤＲＲ）アルゴリズムは、その最小限の実装複雑さと、割り当てられるサービスシェアに比例してフローをサービスする際のその効率とにより、可変サイズのパケットに対するＷＲＲスケジューラの最も普及している実例の１つである。ＷＲＲパラダイムに従って、ＤＲＲアルゴリズムは、サービスシェアρ_iを各設定フローｉに関連づける。サービスシェアは、すべての設定フローにわたるそれらの和がサーバの容量ｒを超えないときに、最小の保証サービスレートとなる。

【数１】

$$\sum_{i=1}^{V} \rho_i \qquad (1)$$

式（１）の限界（ただし、Ｖは設定フローの総数である）は、ρ_iより低くない長期レートでフローｉがサービスを受けることを保証する。

【００１５】

ＤＲＲアルゴリズムは、サーバのアクティビティをサービスフレームに分割する。本発明は、基準タイムスタンプ増分T_Qを用いて仮想時間領域でフレーム期間を表現するアルゴリズムの定式化を参照する。この定式化は、最初に［１０］で提示されたＤＲＲの定義と機能的に等価であるが、本発明の記述にはより適している。

【００１６】

１フレーム内で、各設定フローｉには、次式のような量Q_iの情報ユニットの送信の権利が与えられる。

$$Q_i = \rho_i \cdot T_Q \qquad (2)$$

【００１７】

スケジューラは、フレームあたり１回だけバックログフローを処理するため、一発でそのフレームに対するそれらのフローのサービス期待値を充足する。各フローｉは、パケットのキュー（フローキュー）と、長さl_i^kの新たなパケットp_i^kがフローキューの先頭に到着するごとに更新される次式のタイムスタンプF_iとを管理する。

$$F_i^k = F_i^{k-1} + l_i^k / \rho_i \qquad (3)$$

【００１８】

スケジューラは、フローへのサービスを、そのタイムスタンプがT_Qより小さくとどまる限り続ける。タイムスタンプF_iが基準タイムスタンプ増分T_Qを超えると、スケジューラは、フローｉの処理の終了を宣言し、T_Qをフローｉのタイムスタンプから減算し、サービスすべき別のバックログフローを探す。その結果、T_Qの減算後、F_iの値は、フローｉに対するサービスクレジット（貸し）を表現する。一般に、タイムスタンプは、バックログフローのサービスクレジットを後のフレームに持ち越すため、スケジューラは、長

期間では（すなわち、複数のフレームにわたって）、割り当てられたサービスシェアに比例してサービスを分配することができる。

【００１９】

フローｉがアイドルになると、スケジューラは直ちに別のフローに移り、それへのサービスの提供を開始する。フローｉが短時間で再びバックログになると、フローｉは、サーバからの次の処理を受けるためには、次のフレームの開始を待たなければならない。フローがアイドルになると、同じフローが将来のフレームで再びバックログになった場合のサービスの損失を避けるために、そのフローのタイムスタンプはゼロにリセットされる。構成により、アイドルのフローのタイムスタンプは常にT_Qより小さいため、タイムスタンプのリセットは、他のフローに不利になるような余分のサービスクレジットを生じることはない。

【００２０】

構成により、サービスフレームの始めには、フローｉのタイムスタンプF_iの値は０と$L_i／\rho_i$の間にある。ただし、L_iはフローｉのパケットの最大サイズである。タイムスタンプの初期値のゆらぎは、フローｉが１フレームで送信する情報ユニットの量のゆらぎを引き起こし、これは区間（$Q_i－L_i$，$Q_i＋L_i$）内にある。このため、すべての設定フローが永続的に積滞しているときでさえ、サーバが１フレームで送信する情報ユニットの総量は一定でない。

【００２１】

ＤＲＲスケジューラは、［１０］では、ＦＩＦＯ順に巡回されるバックログフローの単一の連結リストにより実装された。次式のように、基準タイムスタンプ増分T_Qが、最小サービスシェアのフローに対する最大サイズのパケットにより定まるタイムスタンプ増分より小さくない場合には、バックログフローを単一ＦＩＦＯキューに配置すると、Ｏ（１）の実装複雑さとなる。

$$T_Q \geqq L_{ma}／\rho_{mi} \qquad （４）$$

【００２２】

式（４）の条件が満たされない場合、スケジューラのアルゴリズム複雑さは、連続するパケット送信間に実行されるべき基本オペレーションの最悪の場合の数とともに爆発的に増大する（基本オペレーションには、連結リストにおけるフローの取り出しおよび挿入、タイムスタンプ更新、タイムスタンプと基準タイムスタンプ増分の比較、が含まれる）。実際、スケジューラは、基準タイムスタンプ増分の反復減算によりタイムスタンプが［０，T_Q）区間内に入るまで、いくつかの連続するフレームの間、与えられたフローへのサービスを拒否しなければならない可能性がある。図３Ａに、ＤＲＲにおいて、新たなパケットがフローｉのキューの先頭に到着したときに、そのフローｉを処理しそのタイムスタンプを更新するための規則を指定する例示的な擬似コードのリストを示す。

【００２３】

余剰ラウンドロビン（ＳＲＲ）の記述は［１１］に与えられている。このアルゴリズムは、ＤＲＲと同じパラメータおよび変数を有するが、異なるイベントがタイムスタンプの更新を引き起こす。すなわち、フローｉは、パケットp_i^kの送信が完了すると、その結果のフローのバックログ状態とは独立に、新たなタイムスタンプF_i^kを受け取る。フレームの終わりは、１パケットの送信後に常に検出され、その前には検出されない。タイムスタンプは、（ＤＲＲの場合のクレジット（貸し）の代わりに）現フレームの期間中にフローによって累積されたデビット（借り）を次フレームに持ち越す。

【００２４】

ＤＲＲに比べてＳＲＲの利点は、ＳＲＲは、フローのフレームの終わりを決定するためにキュー先頭パケットの長さを前もって知る必要がないことである。逆に、悪意のあるフローが競争相手から帯域を盗むことを防ぐために、このアルゴリズムは、アイドルになったフローのタイムスタンプをリセットすることができない。アイドルフローのヌルでないタイムスタンプは、そのフローがアイドルになったのと同じフレームの終了によって結局は不用(obsolete)となる。理想的には、タイムスタンプは、不用になったらすぐにリセット

されるべきである。しかし、数十万、さらには数百万のフローを処理するスケジューラでは、同時に不用になる可能性のあるすべてのタイムスタンプを即座にリセットすることは現実には不可能である。

【００２５】

本発明についてのこの記述は、アイドルフローのタイムスタンプの不用性を何らチェックしないＳＲＲアルゴリズムの実装に焦点をしぼる。この実装では、新たに積滞したフローは常に、そのタイムスタンプの値がいかに古くとも、そのタイムスタンプの最近の値からアクティビティを再開する。この仮定の効果は、新たに積滞したフローが、前に長時間の間に累積したデビットの結果として、サーバが最初にやってくるときにその受けるべきサービスの一部をあきらめなければならないことがあることである。図３Ｂに、ＳＲＲにおいて、サーバがパケットp_i^kの送信を完了したときにフローｉを処理しそのタイムスタンプを更新するための規則を指定する例示的な擬似コードのリストを示す。

【００２６】

説明を簡単にするため、本明細書の残りの部分では、重み付けラウンドロビン（ＷＲＲ）の名称は一般に、ＤＲＲまたはＳＲＲを総称的に（それらを区別する特徴への明示的な言及がなければ）指すために用いるものとする。

【００２７】

ＷＲＲスケジューラは、個々のパケットフローへの帯域の分配だけを制御することができるという点で、本質的に「１層」である。複数のスケジューリング層を重ね合わせることにより、個々のフローに帯域を割り当てることができるだけでなく、フローの集合を生成しそれに従って帯域を分離することができる、階層的でフレキシブルな構造を実装すると、これらのスケジューラの単純さが損なわれる。

【００２８】

帯域分離の参照モデルを図４に示す。割り当てられるフローのセット（１，１）～（Ｋ，V_K）は、バンドルと呼ばれるＫ個のサブセット４０１－１～４０１－Ｋに分割される。各バンドルＩは、V_I個のフローを集め、割り当てられたサービスレートR_Iを有する。スケジューラの論理編成は、２層化された階層を反映する。すなわち、まず、バンドルの総割当てに従って帯域をバンドルに分配した後、フローの帯域割当てと、それぞれのバンドル内の他のフローのバックログ状態とに基づいてフローをサービスする。スケジューラは、少なくとも１つのフローが積滞している限り、各バンドルを、対応するフローのバックログ状態とは独立に扱う。

【００２９】

スケジューラは、いかなる種類の遅延保証もサポートしようとせずに、フロー集合と、集合内の個々のフローの両方に対して、厳格な帯域保証を強制することを目標とする（スケジューリング階層において厳格な遅延保証を提供するためのいくつかのフレームワークがすでに利用可能である［７，１２］が、それらはすべて、スケジューラの複雑さを相当に増大させる高度なアルゴリズムによるものである）。フロー集合の帯域要求をサポートするためには、バンドルのレート割当てに対して次の条件が常に成り立たなければならない。

【数２】

$$\sum_{I=1}^{K} R_I \leq r \qquad (5)$$

【００３０】

同様に、各バンドルＩ内で、関連するフローの帯域要求を満たすためには、次の限界が満たされなければならない。

【数３】

$$\sum_{i \in I} \rho_i \leq R_I \qquad (6)$$

【００３１】
［７，１２］で提示されたフレームワークによって示唆されるスケジューリング解では、フローとリンクサーバの間に、バンドルを処理する本格的な（そして高価な）スケジューリング層を導入することになる。一般に、本格的な階層スケジューラの実装コストは、バンドルの数について線形に増大する。その理由は、各バンドルがそれぞれ基本のフロー別スケジューラを必要とするからである。これに対して、本発明では、スケジューリング階層におけるバンドル要求を強制する層は純粋に仮想的であり、単一の基本スケジューラ上に重ね合わされる。したがって、個々のフローを処理する構造のコストは、設定されるバンドルの数とは独立であるため、スケジューリング階層の実装における相当の節約になる。
【００３２】
本発明によれば、改良されたＷＲＲスケジューラは、ＷＲＲスケジューラの基本構造の実質的な変更を必要とせずに、帯域を階層的に分離する。この新規な技術によれば、スケジューラは、完全に透過的な方法で、すなわち、追加のスケジューリング構造を使用せずに、帯域保証を、個々のフローに提供するとともに、フローの集合（バンドル）にも提供する。本発明は、この目標を、単に、スケジューラがタイムスタンプを操作する方法を改良することによって達成する。その結果得られる「ソフト」なスケジューリング階層は、無視できるほどの複雑さしか有しない。にもかかわらず、帯域保証を、個々のフローに提供するとともに、バンドルにも提供するのに有効である。
【００３３】
図５に、階層的帯域分離のソフト強制のための改良されたＷＲＲスケジューラによって利用されるキュー、状態テーブル、レジスタ、およびパラメータの機能図である。図４および図５をあわせて参照すると、スケジューラは、複数のバンドル４０１－１～４０１－Ｋにまとめられた、複数のデータパケットフローi_1～j_N（５０１）を処理する。フローキュー５０２は、それぞれのデータパケットフロー５０１ついてデータパケットを格納する。フローキュー５０２は、先入れ先出し（ＦＩＦＯ）キューとして実装可能である。各フローキュー５０２は、対応するフロー別状態テーブル５０３を有し、これは、対応するフロー５０１に対するいくつかの変数を格納する。
【００３４】
例えば、フローi_1の場合、フロー別状態テーブル５０３は、タイムスタンプF_{i1}、最小保証サービスレートρ_{i1}、フレームフラグFF_{i1}、そのフローを含むバンドルＩへのポインタ、ならびに、対応するフローキュー５０２の先頭および末尾のポインタ（図示せず）を含む。バンドルの状態は、バンドル別状態テーブル５０４で管理される。例えば、バンドルＩの場合、バンドル別状態テーブル５０４は、総帯域割当てR_I、現在シェアϕ_I、累積シェアΦ_I、および開始フラグσ_Iを格納する。フローポインタのＦＩＦＯキュー５０５は、スケジューラがフロー５０１のパケットを送信するために巡回する順序を決定する。レジスタ５０６は、フローポインタのＦＩＦＯキューの先頭および末尾のポインタを格納する。スケジューラは、テーブル５０７でグローバル状態情報を管理する。このテーブルは、フレームカウンタＦＲＭＣＮＴおよび基準タイムスタンプ増分T_Qを管理する。
【００３５】
図６は、スケジューラが利用される入力通信リンクインタフェース２００の例示的なブロック図である。通信リンクインタフェース２００は、データパケット受信器６０１、スケジューラ６０２、およびパケット送信器６０９を有する。例として、スケジューラは、コントローラ６０３、バンドル別状態ＲＡＭ６０４、およびレジスタ６０５を、すべて同一

のチップ６０６上に有するように示されている。パケットＲＡＭ６０７およびフロー別状態ＲＡＭ６０８は、別々のチップ上に位置するように示されている。明らかに、動作容量およびその他の特性に依存して、スケジューラ６０２は、他の構成でも実装可能である。

【００３６】

コントローラ６０３は、本発明の方法を実装したプログラムを格納し実行する。通信リンクインタフェース２００の動作を制御するプログラムの実施例を図７～図８にフローチャート形式で示す。図５および図６をあわせて参照すると、フローキュー５０２内のパケットはパケットＲＡＭ６０７に格納される。フロー別状態テーブル５０３はフロー別状態ＲＡＭ６０８に格納される。フローポインタのＦＩＦＯキュー５０５の先頭および末尾ポインタ５０６ならびにグローバル状態テーブル５０７はレジスタ６０５に格納される。

【００３７】

スケジューラ６０２の動作の概略は以下の通りである。パケット受信器６０１は、入力リンク２０１－１～２０１－ｒから、データパケットフロー５０１のデータパケットを受信する。パケット受信器６０１は、各パケット（図示せず）のヘッダに含まれるフロー識別フィールドの内容を用いて、そのパケットのデータパケットフロー５０１を識別する。データパケットフロー５０１の識別により、関連するフローキュー５０２およびバンドル４０１が識別される。スケジューラ６０２は、バンドル別状態テーブル５０４のエントリを用いて、各バンドルＩに対するサービス比を計算する。バンドルＩのサービス比は、公称帯域割当てR_Iと、そのバンドルの累積シェアΦ_Iとの間の比として定義される（累積シェアΦ_Iは、バンドルＩのバックログフローの帯域割当てを累積したものである）。構成により、バンドルＩのサービス比は決して１より小さくならない。

【００３８】

サービス比には、個々のフローに対するタイムスタンプ増分の計算が関係する。これは、バンドルＩの各バックログフローが、同じバンドルの他のフローがフレームの最初にアイドルであると検出されたときのそのフレームの期間中に受けるサービスの量の増大分を決定する。サービス比により引き起こされるタイムスタンプ増分の修正は、スケジューラが、バンドルの総帯域割当てと、個々のフローの帯域割当てとの両方を依然として保証することができるようになされる。

【００３９】

［動作の詳細］

本発明の改良されたＷＲＲスケジューラは、従来技術のＷＲＲスケジューラ上に仮想スケジューリング層を重ね合わせることによって、サービス帯域の階層的分離を達成する。本発明のスケジューラは、複数のバンドルをサポートし、各バンドルは、データパケットフローの集合である。基礎となるＷＲＲスケジューラのタイムスタンプに基づく定式化において（説明のこの点では、ＤＲＲとＳＲＲの区別はまだ重要ではない）、本発明の技術は、タイムスタンプ更新の簡単な修正により、個々のフローの帯域保証を保存すると同時に、バンドルの総要求を満たす。

【００４０】

それぞれの設定されるバンドルＩについて、スケジューラは、公称帯域割当てR_Iおよび累積シェアΦ_Iを管理する。公称帯域割当てR_Iは、そのバンドル内のすべてのフローの帯域割当ての総和より小さくなることはない。累積シェアΦ_Iは、バンドルＩのバックログフローの帯域割当ての総和を追跡し、したがって、決してR_Iより大きくならない。

【数４】

$$\Phi_I = \sum_{i \in B_I} \rho_i \qquad (7)$$

式（７）において、B_Iは、フレームの最初に積滞している、バンドルＩのフローの集合である。

【００４１】
スケジューラは、バンドルＩの累積レート割当てR_Iと累積シェアΦ_Iの間のサービス比を用いて、そのバンドルのフローに付与される実際のサービス帯域を修正する。具体的には、サービス比は、バンドルのフローに対するタイムスタンプ増分の定義に寄与する。フローｉのパケットp_i^kに関連するタイムスタンプ更新は次式の通りである。
【数５】

$$F_i^k = F_i^{k-1} + \frac{l_i^k}{\rho_i} \cdot \frac{\Phi_I}{R_I} \qquad (8)$$

【００４２】
式（８）のタイムスタンプ割当て規則が実際にバンドルの帯域保証を強制することを確かめるには、バンドルＩ内の個々のフローが１フレームの期間中に受けることが期待されるサービスの量を計算することが必要である。原理的には、サービス比R_I/Φ_IがバンドルＩについて１より大きいとき、式（８）のタイムスタンプ割当ての結果、バンドルＩの個々のフローは、それらの公称帯域割当てのR_I/Φ_I倍のサービス帯域を得る。
【００４３】
この計算は、次の２つの仮定に基づく。
（１）バンドルの累積シェアΦ_Iは、バンドル内のフローのバックログダイナミクスとは無関係に、１フレームの期間中は不変である。
（２）そのフレームの期間中にサーバにアクセス可能なフローのセットは、そのフレームの最初に積滞していたフローのみを含む（もしバンドル内の一部のフローが、フレームの開始後に積滞した場合、それらのフローは、サーバにアクセス可能になる前に、新たなフレームの開始まで待たなければならない）。
【００４４】
基準タイムスタンプ増分T_Qは、式（８）と組み合わせて、バンドルＩのフローｉが１フレーム期間中に受けることが期待される基準サービス量Q_iを設定する。
【数６】

$$T_Q = \frac{Q_i}{\rho_i} \cdot \frac{\Phi_I}{R_I} \qquad (9)$$

【００４５】
すると、バンドルＩ内のすべてのフローのサービス量の総計は、バンドルのサービス量Q_Iとなる。
【数７】

$$Q_I = \sum_{i \in B_I} Q_i = \frac{\sum_{i \in B_I} \rho_i}{\Phi_I} \cdot R_I \cdot T_Q = R_I \cdot T_Q \qquad (10)$$

【００４６】
式（１０）におけるQ_Iの表式は、式（２）におけるフロー量Q_iの表式と同一であるため、フレームの最初にバンドル内で積滞しているフローのセットの組成とは無関係に、式（８）のタイムスタンプ更新がバンドルＩの帯域保証を保存することが証明される。
【００４７】
バンドルＩの累積シェアがフレーム期間中に変化しないという仮定を保持すると、次式のように、式（８）のタイムスタンプ更新規則は、フレーム期間中に決してアイドルにならないバンドルＩの任意の２個のフローｉ、ｊについて、サービス割合を保存することも示

すことができる。

【数８】

$$\frac{Q_i}{Q_j} = \frac{\rho_i \cdot \frac{R_I}{\Phi_I} \cdot T_Q}{\rho_j \cdot \frac{R_I}{\Phi_I} \cdot T_Q} = \frac{\rho_i}{\rho_j} \qquad (11)$$

【００４８】

帯域分離を有するＷＲＲアルゴリズムの詳細を指定するには、式（１０）および（１１）の結果を生成する仮定についての議論と、それら仮定のアルゴリズム的な意味の評価とを必要とする。スケジューラが１フレーム期間中に計算するすべてのタイムスタンプ増分において一定値の累積シェアΦ_Iを使用することは、バンドルＩのフローにサービスを一貫して分配するための共通の基準を提供する。フレームの開始後になってはじめて積滞したフローをサービスフレームから排除することも、同一の目的を有する。

【００４９】

タイムスタンプ増分は、システムがフローに対して、関連するパケットの送信について課す料金とみなすことができる。送信のコストは、それが実行されるときのバンドル内で利用可能な帯域に依存する。タイムスタンプ増分を、バンドル内の帯域資源のコストと整合させるためには、タイムスタンプ増分は、資源が使用されるときに、すなわち、対応するパケットの送信時に、計算されなければならない。もしスケジューラが、増分を前もって計算するならば、バンドルの状態は、パケットの送信が起こる前に急激な変化を受ける可能性があるため、課金メカニズムが帯域の分配と整合しなくなる。

【００５０】

考えている２つのＷＲＲアルゴリズムのうち、ＳＲＲは、送信とタイムスタンプ増分の間の整合性に対する要求に最もよく適合するものである。その理由は、ＳＲＲは、ちょうど送信したばかりのパケットの長さを用いてタイムスタンプを更新し、対応するフローのフレーム内状態を決定するからである。これに対して、ＤＲＲでは、スケジューラは、新たなキュー先頭パケットの長さを用いて、タイムスタンプ更新およびフレーム内状態チェックを実行するが、これは、そのパケットが実際に送信されるずっと前である可能性がある。ＤＲＲサーバが最終的にパケットを配信するとき、バンドルの累積シェア、したがってそのバンドル内の帯域のコストは、最後のタイムスタンプ更新以来相当に変化してしまっている可能性がある。

【００５１】

ＳＲＲにおける帯域分離のためのメカニズムの導入は簡単である。最小帯域保証R_Iおよび累積シェアΦ_Iに加えて、各バンドルＩは、現在シェアϕ_Iおよび開始フラグσ_Iを管理する。現在シェアは、次式のように、バンドル内のバックログフローのサービスシェアの総和を瞬間的に追跡する。

【数９】

$$\phi_I(t) = \sum_{i \in B_I(t)} \rho_i \qquad \forall t \qquad (12)$$

【００５２】

現在シェアは、バンドルの１つのフローがそのバックログ状態を変えるたびに更新される。一般に、現在シェアϕ_Iの更新は、累積シェアΦ_Iの即時更新を引き起こさない。実際

、スケジューラは、開始フラグσ_Iと、スケジューラがフレーム境界ごとにトグルする１ビットのグローバルなフレームカウンタＦＲＭＣＮＴとにおける整合しない値を検出した場合にのみ、バンドルの累積シェアΦ_Iを更新する（スケジューラは、バンドルＩの１つのフローのサービスを開始するごとに、σ_IとＦＲＭＣＮＴを比較する）。

【００５３】

これらの２つのビットが相違することは、将来のタイムスタンプ計算で用いられる累積シェアの更新を引き起こす（$\Phi_I \leftarrow \varphi_I$）とともに、バンドルの開始フラグをトグルする（$\sigma_I \leftarrow$ＦＲＭＣＮＴ）。他方、これらの２つのビットがすでに等しい場合、ちょうど完了したサービスは、現フレームの期間中にバンドルが受けた最初のサービスではないことが確実であり、バンドルパラメータに対して何らのアクションもとられてはならない。バンドルの最初のフローが積滞すると、開始フラグはＦＲＭＣＮＴに等しくセットされる。

$\sigma_I \leftarrow$ＦＲＭＣＮＴ　　（１３）

【００５４】

フレームの終わりを識別するため、各フローｉはフレームフラグFF_iを管理する。フローｉのフレームフラグは、そのフローがバックログフローのリスト５０５の末尾にキューイングされるときには必ず、ＦＲＭＣＮＴの補数に等しくセットされる。スケジューラは、フレームカウンタに一致しないフレームフラグを見つけると、新たなフレームの開始を宣言し、フレームカウンタをトグルする。１パケットの送信を完了した後に実行される一連のオペレーションは、図９の擬似コードにまとめられている。

【００５５】

図７～図８に、本発明に従ってパケット送信のスケジューリングを制御するために図６のスケジューリング装置を動作させる方法をフローチャート形式で示す。図７～図８のフローチャートおよび図９の擬似コードは、ＳＲＲが、基礎となるスケジューリングアルゴリズムであるという仮定に基づいている。機能性に関する限り、ＳＲＲの代わりにＤＲＲを用いても問題はない。同様に、図５の装置は、バックログフローの単一のＦＩＦＯキューを用いてソフトスケジューリング階層を実装している。フレーム内とフレーム外のフローの明確な分離が可能な他の任意のキューイング構造も同様に使用可能である。

【００５６】

以下の記述では、図４、図５、図６、および図７～図８を参照する。図４（図５、図６）で最初に現れる要素の参照符号は４（５、６）から始まるが、図７～図８のステップは、ステップ番号の前にＳを付けることによって（例えば、Ｓ５１０）示される。

【００５７】

図７において、Ｓ５１０で、コントローラ６０３は、新たに受信したデータパケットがあるかどうかをチェックする。Ｓ５１０で、新たに受信したデータパケットがなく、Ｓ５２０で、バックログフローがある場合、制御はＳ６８０に移る。他方、Ｓ５１０で、新たに受信したデータパケットがなく、Ｓ５２０で、バックログフローがない場合、コントローラ６０３は、新たなパケットが受信されるまで、ステップＳ５１０とＳ５２０の間を循環する。Ｓ５１０において、新たに受信したパケットの存在が受信器６０１で検出されると、Ｓ５５０で、コントローラ６０３は、それらのパケットのうちの１つを選択する。その後、Ｓ５６０で、コントローラ６０３は、データパケットのフローを識別し、最終的に（Ｓ５７０で）、パケットを適当なフローキュー５０２に格納する。Ｓ５８０で、識別されたフローのキューの長さがゼロでない場合、Ｓ９７０で、そのフローのキュー長をインクリメントし、制御はＳ６８０に移る。

【００５８】

他方、Ｓ５８０で、識別されたフローのキューの長さがゼロである場合、Ｓ５８５で、そのフローのフレームフラグ（フロー別状態テーブル５０３内）がＦＲＭＣＮＴ（グローバル状態テーブル５０７内）の補数に等しくセットされる。その後、コントローラは、Ｓ５９０で、バックログフローの総数を増大させ、Ｓ６００で、識別されたフローのキュー長を増大させる。Ｓ６１０で、識別されたフローのバンドルが、フロー別状態テーブル５０３のバンドルポインタを用いて識別される。識別されたバンドルの現在シェアが０である

場合、Ｓ６３０で、コントローラ６０３は、バンドルの開始フラグ（バンドル別状態テーブル５０４に格納される）を、グローバルフレームカウンタＦＲＭＣＮＴの値に等しくセットする。その後、制御はＳ６３５に移る。他方、Ｓ６２０で、バンドルの現在シェアが０でない場合、制御は直接Ｓ６２０からＳ６３５に移る。Ｓ６３５で、識別されたバンドルの現在シェアが、新たに積滞したフローの帯域割当てだけ増大させられる。そして、Ｓ６４０で、フローはフローポインタのＦＩＦＯキュー５０５の末尾に付加され、その後、制御はＳ６８０に移る。

【００５９】

Ｓ６８０で、送信器６０９が古いパケットの送信でビジーであり、したがって新たなパケットの送信のために利用可能でない場合、制御はＳ５１０に戻る。そうでない場合、Ｓ７００で、サービス後処理のために待機しているサービスされたばかりのフローの利用可能性がチェックされる。Ｓ７００で、サービスされたばかりのフローが利用可能でない場合、Ｓ７１０で、バックログフローがあるかどうかが判定される。バックログフローが存在しない場合、制御はＳ５１０に戻り、存在する場合、Ｓ７２０で、フローポインタのＦＩＦＯキュー５０５の先頭５０６にあるフロー５０１がサービスのために選択され、Ｓ７３０で、選択されたフローのフローキュー５０２内の最初のデータパケットが送信器６０９に送られる。Ｓ７４０で、サービスのために選択されたフロー５０１のフレームフラグがグローバルフレームカウンタＦＲＭＣＮＴに等しい場合、制御はＳ５１０に戻り、そうでない場合、Ｓ７５０で、グローバルフレームカウンタＦＲＭＣＮＴはトグルされ、制御はＳ５１０に戻る。

【００６０】

Ｓ７００で、サービス後処理のために待機しているサービスされたばかりのフロー５０１が利用可能である場合、コントローラはＳ７６０に進み、対応するフローキュー５０２の長さをデクリメントする。Ｓ７７０で、サービスされたばかりのフロー５０１のバンドル４０１が、フロー別状態テーブル５０３のバンドルポインタを用いて識別される。Ｓ７８０で、グローバルフレームカウンタＦＲＭＣＮＴが、サービスされたばかりのフロー５０１のバンドル４０１の開始フラグに等しいかどうかが判定される。開始フラグがグローバルフレームカウンタに等しくない場合、コントローラは、（Ｓ７９０で）バンドルの累積シェアをバンドルの現在シェアに等しいとセットし、（Ｓ８００で）バンドルの開始フラグをグローバルフレームカウンタに等しいとセットする。その後、制御は、フロータイムスタンプの更新のためにＳ８１０に進む。Ｓ７８０で、サービスされたばかりのフローのバンドルの開始フラグがグローバルフレームカウンタに等しい場合、サービスされたばかりのフローのタイムスタンプの更新のために、制御は直接Ｓ８１０に移る。

【００６１】

Ｓ８２０で、コントローラ６０３は、サービスされたばかりのフローのキュー長がゼロに等しいかどうかを判定する。Ｓ８２０で、サービスされたフロー５０１のフローキュー５０２が空であると判定された場合、Ｓ８８０で、コントローラは、サービスされたばかりのフローのタイムスタンプがグローバル状態テーブル５０７の基準タイムスタンプ増分以上であるかどうかをチェックする。タイムスタンプが基準タイムスタンプ以上である場合、制御はＳ８９０に移り、タイムスタンプは正当な範囲（０，T_Q）内にリセットされる。その後、制御はＳ９００に移る。Ｓ８８０で、サービスされたばかりのフローのタイムスタンプが基準タイムスタンプ増分より小さいと判定された場合、制御は直接Ｓ９００に移る。Ｓ９００で、サービスされたばかりのフローへのポインタが、フローポインタのＦＩＦＯキュー５０５の先頭５０６から取り出される。その後、Ｓ９１０で、サービスされたばかりのフローのバンドルの現在シェアが、アイドルになった、サービスされたばかりのフローの帯域割当てだけ減少させられる。Ｓ９２０で、コントローラ６０３は、バックログフローの総数をデクリメントした後、Ｓ７１０に進む。

【００６２】

Ｓ８２０で、サービスされたフロー５０１のフローキュー５０２が空でないと判定された場合、Ｓ８３０で、コントローラは、サービスされたばかりのフローのタイムスタンプが

グローバル状態テーブル５０７の基準タイムスタンプ増分以上であるかどうかをチェックする。タイムスタンプが基準タイムスタンプより小さい場合、制御はＳ８４０に移り、サービスされたばかりのフローのフレームフラグがトグルされる。その後、Ｓ８５０で、サービスされたばかりのフローのタイムスタンプは正当な範囲（０，T_Q）内にリセットされる。Ｓ８６０で、コントローラ６０３は、サービスされたばかりのフローへのポインタを、フローポインタのＦＩＦＯキュー５０５の先頭５０６から取り出す。Ｓ８７０で、同じポインタが、ＦＩＦＯキュー５０５の末尾５０６に再びキューイングされる。その後、制御はＳ７１０に移る。Ｓ８３０で、サービスされたばかりのフローのタイムスタンプが基準タイムスタンプ増分以上であると判定された場合、制御は直接Ｓ７１０に移る。

【００６３】

図９のリストは、本発明に従ってバンドルＩのフローｉを選択しサービス後処理する方法を擬似コード形式で記述したものである。

【００６４】

上記の実施例は、本発明に従って、重み付けラウンドロビン上にソフトスケジューリング階層を重ね合わせるために使用可能な諸原理の単なる例示である。当業者であれば、ここに明示的に記載していなくとも、特許請求の範囲により規定される本発明の技術思想および技術的範囲から離れずに、それらの原理を実現するさまざまな構成を考えることが可能である。

【００６５】

［付録］

文献

［１］ R. Braden, D. Clark, and S. Shenker, "Integrated Services in the Internet Architecture: an Overview", Request for Comments (RFC) 1633, IETF, June 1994.

［２］ K. Nichols, V. Jacobson, and L. Zhang, "A Two-bit Differentiated Services Architecture for the Internet", Request for Comments (RFC) 2638, IETF, July 1999.

［３］ D. Stiliadis and A. Varma, "Design and Analysis of Frame-based Fair Queueing: A New Traffic Scheduling Algorithm for Packet-Switched Networks", Proceedings of ACM SIGMETRICS '96, pp.104-115, May 1996.

［４］ D. Stiliadis and A. Varma, "Efficient Fair Queueing Algorithms for Packet-Switched Networks",IEEE/ACM Transactions on Networking, Vol.6, No.2, pp.175-185, April 1998.

［５］ J. C. R. Bennett and H. Zhang, "WF2Q: Worst-Case-Fair Weighted Fair Queueing", Proceedings of IEEE INFOCOM '96, pp.120-128, March 1996.

［６］ D. Stiliadis and A. Varma, "A General Methodology for Designing Efficient Traffic Scheduling and Shaping Algorithms", Proceedings of IEEE INFOCOM '97, Kobe, Japan, April 1997.

［７］ I. Stoica, H. Zhang, and T. S. E. Ng, "A Hierarchical Fair Service Curve Algorithm for Link-Sharing, Real-Time, and Priority Services", Proceedings of ACM SIGCOMM '97, September 1997.

［８］ D. C. Stephens, J. C. R. Bennett, and H. Zhang, "Implementing Scheduling Algorithms in High-Speed Networks", IEEE Journal on Selected Areas in Communications, Vol.17, No.6, June 1999, pp.1145-1158.

［９］ M. Katevenis, S. Sidiropoulos, and C. Courcoubetis, "Weighted Round Robin Cell Multiplexing in a General-Purpose ATM Switch", IEEE Journal on Selected Areas in Communications, vol.9, pp.1265-79, October 1991.

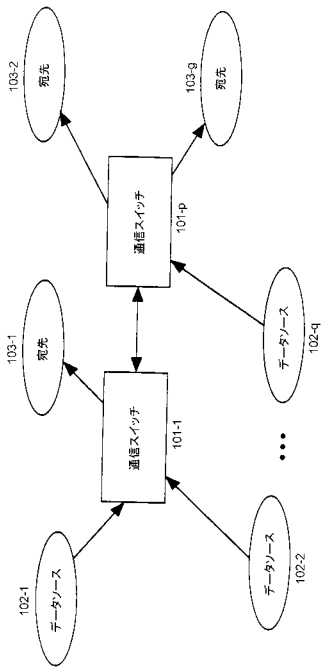
［１０］ M. Shreedhar and G. Varghese, "Efficient Fair Queueing Using Deficit Round Robin", IEEE/ACM Transactions on Networking, vol.4, no.3, pp.375-385, June 1996.

［１１］ H. Adiseshu, G. Parulkar, and G. Varghese, "A Reliable and Scalable Str

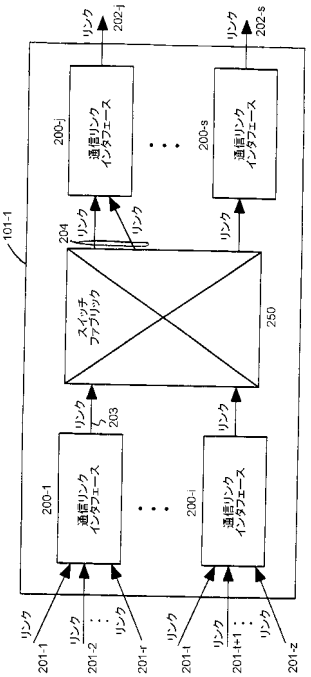
iping Protocol", Proceedings of ACM SIGCOMM '96, August 1996.
［１２］ J. C. R. Bennett and H. Zhang, "Hierarchical Packet Fair Queueing Algorithms", Proceedings of ACM SIGCOMM '96, pp.143-156, August 1996.
【００６６】
【発明の効果】
以上述べたごとく、本発明によれば、基本スケジューラの単純さを損なわずに階層的帯域分離が達成される。
【００６７】
特許請求の範囲の発明の要件の後に括弧で記載した番号がある場合は、本発明の一実施例の対応関係を示すものであって、本発明の範囲を限定するものと解釈すべきではない。
【図面の簡単な説明】
【図１】データソース、通信スイッチ、およびデータ宛先を含む例示的なパケットネットワークを示す図である。
【図２】図１のパケットネットワークで用いられる例示的な通信スイッチを示す図である。
【図３】図３Ａは、新たなパケットがフローキューの先頭に到着したときに不足ラウンドロビン（ＤＲＲ）アルゴリズムで用いられる擬似コードの例を示し、図３Ｂは、サーバが１パケットの送信を完了したときに余剰ラウンドロビン（ＳＲＲ）アルゴリズムで用いられる擬似コードの例を示す図である。
【図４】本発明による、スケジューラの２層論理編成を示す図である。
【図５】本発明のスケジューラによって利用されるキュー、状態テーブル、レジスタ、およびパラメータの機能図である。
【図６】図５の装置の具体的実装の例示的ブロック図である。
【図７】本発明に従ってパケットの送信をスケジューリングする方法を記述した例示的な流れ図である。
【図８】本発明に従ってパケットの送信をスケジューリングする方法を記述した例示的な流れ図である。
【図９】１パケットの送信を完了した後にスケジューラによって使用される擬似コードの例を示す図である。
【符号の説明】
１０１　通信スイッチ
１０２　データソース
１０３　宛先
２００　通信リンクインタフェース
２０１　入力リンク
２０２　出力リンク
２５０　スイッチファブリック
４０１　バンドル
５０１　データパケットフロー
５０２　フローキュー
５０３　フロー別状態テーブル
５０４　バンドル別状態テーブル
５０５　フローポインタのＦＩＦＯキュー
５０６　レジスタ
５０７　グローバル状態テーブル
６０１　パケット受信器
６０２　スケジューラ
６０３　コントローラ
６０４　バンドル別状態ＲＡＭ
６０５　レジスタ

６０６　チップ
６０７　パケットＲＡＭ
６０８　フロー別状態ＲＡＭ
６０９　パケット送信器

【図１】



【図２】



【図３】

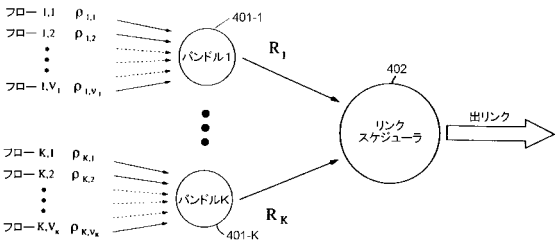
```
1  if（フローiが新たに積滞）
2        F_i^k ← l_i^k / ρ_i
3        iを連結リストの末尾に付加
4  else /* iのパケットが送信されたばかり */
5        F_i^k ← F_i^{k-1} + l_i^k / ρ_i
6        if（F_i^k ≥ T_U）
7              F_i^k ← F_i^k − T_U
8              フローiの処理を終了
9        else
10             フローiへのサービスを継続
```

A

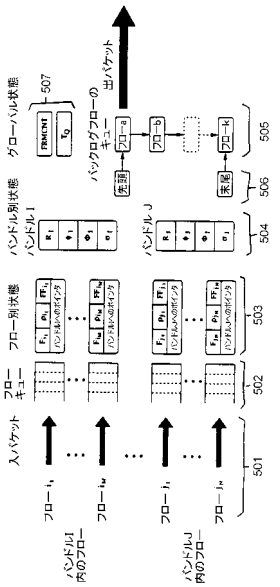
```
1  F_i^k ← F_i^{k-1} + l_i^k / ρ_i
2  if（F_i^k ≥ T_U）
3        F_i^k ← F_i^k − T_U
4        フローiの処理を終了
5  else if（フローiがまだ積滞）
6        フローiへのサービスを継続
```

B

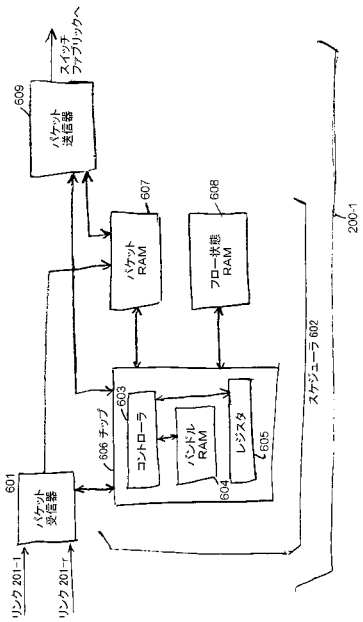
【図４】



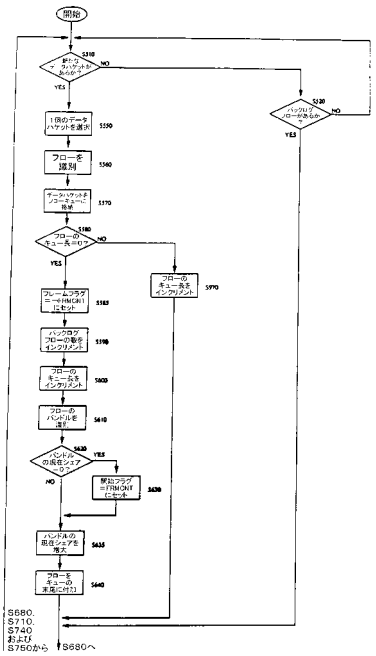
【図５】



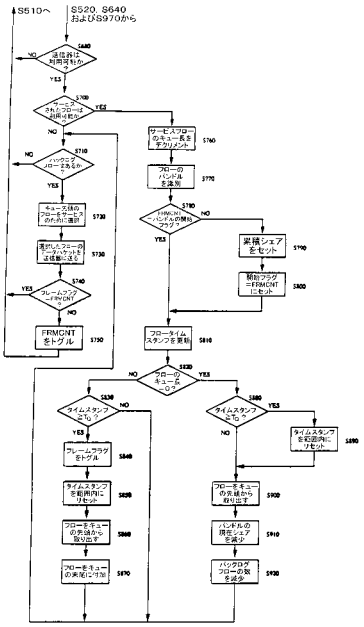
【図６】



【図７】



【図８】



【図９】

1 現在連結リストの先頭にあるフロー i を識別
2 フロー i のバンドル l を識別
3 **if**（$FF_i \neq FRMCNT$）
4 　　$FRMCNT \leftarrow \neg FRMCNT$
5 キュー先頭パケット p_i^k を送信のために用意
6 **if**（$\sigma_l \neq FRMCNT$）
7 　　$\Phi_l \leftarrow \phi_l$
8 $\sigma_l \leftarrow FRMCNT$
9 $F_i^k \leftarrow F_i^{k-1} + \frac{l_i^k}{R_l} \cdot \frac{\Phi_{l2}}{\rho_l}$
10 **if**（$F_i^k \geq T_{i2}$）/* フロー i のフレーム終了 */
11 　　$F_i^k \leftarrow F_i^k - T_{i2}$
12 　　$FF_i \leftarrow \neg FRMCNT$
13 　　フロー i を連結リストの先頭から取り出す
14 　　**if**（フロー i がまだ積滞）
15 　　　　フロー i を連結リストの末尾に付加
16 　　**else** /* フロー i はアイドルになる */
17 　　　　$\phi_l \leftarrow \phi_l - \rho_i$
18 **else if**（フロー i がアイドルになる）
19 　　フロー i を連結リストの先頭から取り出す
20 　　$\phi_l \leftarrow \phi_l - \rho_i$

フロントページの続き

(74)代理人 100091889
弁理士 藤野 育男
(74)代理人 100101498
弁理士 越智 隆夫
(74)代理人 100096688
弁理士 本宮 照久
(74)代理人 100102808
弁理士 高梨 憲通
(74)代理人 100104352
弁理士 朝日 伸光
(74)代理人 100107401
弁理士 高橋 誠一郎
(74)代理人 100106183
弁理士 吉澤 弘司
(74)代理人 100081053
弁理士 三俣 弘文
(72)発明者 ファビオ エム チウスイ
アメリカ合衆国、０７７４０ ニュージャージー州、ロング ブランチ、Ａｐｔ．４D、オーシャン アベニュー Ｎ ３００
(72)発明者 ロバート ティー クランシー
アメリカ合衆国、０６４３７ コネチカット州、ギルフォード、ホーソン ロード １１３
(72)発明者 ケヴィン ディー ドラッカー
アメリカ合衆国、０８８０１ ニュージャージー州、アナンデール、ウェストゲート ドライブ １８
(72)発明者 アンドレーエ フランシーン
アメリカ合衆国、０７７０１ ニュージャージー州、レッド バンク、ウェスト フロント ストリート ５３７
(72)発明者 ナサー イー イディ
アメリカ合衆国、０７７２４ ニュージャージー州、イートンタウン、ウェッジウッド サークル ６５

審査官 衣鳩 文彦

(56)参考文献 特開２０００－３１９９７（ＪＰ，Ａ）
F M Chiussi et al., A distributedscheduling architecture for scalable packet switches, IEEE Journal on Selected Areas, 米国, IEEE, ２０００年１２月, Volume:18 Issue:12, 第２６６５－２６８３頁
Kai-Yeung Siu et al., Virtual queueing techniques for UBR+ service in ATM with fair access and minimum bandwidth guarantee, Global Telecommunications Conference, 1997. GLOBECOM '97, 米国, IEEE, １９９７年１１月, VOL:2, 第１０８１－１０８５頁

(58)調査した分野(Int.Cl., ＤＢ名)
H04L 12/56