社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.

複数のルータが協調動作するアクティブキュー管理機構の設計と性能評価

江口　智也† 　大崎　博之† 　村田　正幸††

† 大阪大学 大学院情報科学研究科　〒 560-8531 大阪府豊中市待兼山町 1-3
†† 大阪大学 サイバーメディアセンター　〒 560-0043 大阪府豊中市待兼山町 1-30
E-mail: †{t-eguti,oosaki}@ist.osaka-u.ac.jp, ††murata@cmc.osaka-u.ac.jp

あらまし　近年、ルータにおいて積極的にパケットを廃棄し、ルータのバッファ内パケット数を制御する、アクティブキュー管理機構が注目されている。しかし、従来の研究では、単一のルータを対象として設計されたものがほとんどであり、ネットワーク中の複数のルータを対象としたアクティブキュー管理機構は十分検討されていない。ネットワーク中に複数のルータが存在する、一般的なネットワークトポロジの場合、TCP は F_A^h 公平性をみたすことが知られている。これは、TCP のスループットは、エンド–エンド間の伝搬遅延やホップ数が小さいコネクションほど有利になることを意味する。本稿では、ネットワーク中に複数のルータが存在する、一般的なネットワークを対象とする。そのようなネットワーク上で動作する、複数のルータが協調動作するアクティブキュー管理機構を設計し、TCP コネクション間の公平性を改善することを目標とする。我々が提案するアクティブキュー管理機構では、ECN (Explicit Congestion Notification) 機構を利用し、ルータに到着するパケットの CE (Congestion Experienced) ビットの値に応じて、RED のパケット棄却率を変化させる。簡単な定常状態解析により、我々が提案するアクティブキュー管理機構を用いることで、多段接続されたネットワークにおける、TCP コネクション間の公平性が改善されることを示す。
キーワード　アクティブキュー管理機構、RED (Random Early Detection)、ECN (Explicit Congestion Notification)、公平性

Design and Performance Evaluation of Distributed Active Queue Management Mechanism Cooperating among Multiple Routers

Tomoya EGUCHI†, Hiroyuki OHSAKI†, and Masayuki MURATA††

† Graduate School of Information Science and Technology, Osaka University, Japan
†† Cybermedia Center, Osaka University, Japan
E-mail: †{t-eguti,oosaki}@ist.osaka-u.ac.jp, ††murata@cmc.osaka-u.ac.jp

Abstract　In recent years, AQM (Active Queue Management) mechanisms, which control the number of packets in the router's buffer by discarding an arriving packet, have been actively studied. In past studies, most of AQM mechanisms have been designed by taking account of a single router; i.e., few AQM mechanisms have been designed by taking account of multiple routers in the network. In a general network with multiple routers and heterogeneous TCP connections, it is known that the bandwidth allocation to TCP (Transmission Control Protocol) connections satisfies F_A^h fairness criterion. This means that the throughput of TCP connection tends to be high when its propagation delay and/or the number of its traversing links is small. In this paper, we focus on such a general network, and design an AQM mechanism that cooperates with other routers for improving the fairness among heterogeneous TCP connections. In our AQM mechanism, ECN (Explicit Congestion Notification) mechanism is used for differentiating the packet marking probability of RED according to CE (Congestion Experienced) bit of an arriving packet. We analyze the steady state behavior of our proposed AQM mechanism, and show that the fairness among TCP connections is improved compared with the RED router.
Key words　Active Queue Management Mechanism, RED (Random Early Detection), ECN (Explicit Congestion Notification), Fairness

1. はじめに

近年、ルータにおいて積極的にパケットを廃棄し、ルータのバッファ内パケット数を制御する、アクティブキュー管理機構が注目されている [1]。アクティブキュー管理機構を用いることにより、従来の DropTail ルータが持つさまざまな問題点を解消できる。例えば、アクティブキュー管理機構の制御により、ルータの平均的なバッファ内パケット数が減少するため、ルータのバッファにおける遅延時間が減少する。これにより、TCP (Transmission Control Protocol) のエンド–エンド間遅延の減少が期待できる。また、ルータのバッファあふれに起因する、TCP コネクションの同期を防ぐことが可能となる。TCP コネクションが同期すると、その結果、ルータのバッファにおいて連続的なパケット棄却が大量に発生する。この時、TCP はタイムアウト機構により、転送レートを大幅に減少させるので、TCP コネクションの同期を防ぐことはスループット向上に有効である。

アクティブキュー管理機構は、TCP の輻輳制御機構が、ネットワークからのフィードバック情報として、ネットワーク中でのパケット棄却の有無を用いていることを利用している [2]。つまり、アクティブキュー管理機構は、ネットワーク中でパケット棄却が発生すると、それに反応して TCP がパケット送出レートを減少させることを利用している。現在、インターネットのトラヒックの大部分が TCP によって転送されていることを考えると、アクティブキュー管理機構による制御は非常に有効であると考えられる。TCP の輻輳制御機構を補助するアクティブキュー管理機構は、これまでにもさまざまな方式が提案されてい

る。最も代表的なアクティブキュー管理機構は、RED (Random Early Detection) である [3]。RED は、移動指数平均 (Exponential Weighted Moving Average) を用いることにより、現在キュー長 (ルータのバッファ内パケット数) から平均キュー長を計算し、平均キュー長の値に応じた確率で、ルータに到着するパケットをランダムに廃棄する。

これまで、RED 以外にもさまざまなアクティブキュー管理機構が提案されている [3-7]。しかし、従来の研究では、単一のルータを対象として設計されたものがほとんどであり、ネットワーク中の複数のルータを対象としたアクティブキュー管理機構は十分検討されていない。ネットワーク中に単一のルータのみが存在する場合は、これまでに提案されているアクティブキュー管理機構を用いることにより、TCP のエンド–エンド間遅延を短く抑え、TCP コネクションの同期によって生じるスループットの低下を防ぐことができる。また、FRED (Fair RED) [7] や SRED (Stabilized RED) [5] のように、ルータにおいて TCP コネクションを識別し、TCP コネクションごとに異なるパケット棄却方法を用いることにより、ルータ内に収容されている TCP コネクション間の公平性を向上することも可能である。

一方、ネットワーク中に複数のルータが存在するような、一般的なネットワークトポロジの場合、TCP は F_A^h 公平性をみたすことが知られている [8]。これは、TCP のスループットは、エンド–エンド間の伝搬遅延やホップ数が小さいコネクションほど有利になることを意味する。この問題は、(1) TCP のウィンドウ型フロー制御は、ラウンドトリップ時間ごとにウィンドウサイズを増減すること、(2) TCP はネットワーク中でのパケット棄却をもとにウィンドウサイズを変更すること、に起因している。つまり、伝搬遅延の小さな TCP コネクションほどウィンドウサイズが速く増加するので、ホップ数の小さな TCP コネクションほどパケット棄却率が小さくなるためである。

本稿では、ネットワーク中に複数のルータが存在する、一般的なネットワークを対象とする。そのようなネットワーク上で動作する、複数のルータが協調動作するアクティブキュー管理機構を設計し、TCP コネクション間の公平性を改善することを目標とする。我々が提案するアクティブキュー管理機構では、文献 [9] で提案されている ECN (Explicit Congestion Notification) 機構を利用する。具体的には、ルータに到着する IP ヘッダの CE (Congestion Experienced) ビットの値に応じて、RED のパケット棄却率を変化させるという方式である。これにより、ホップ数の多い (IP ヘッダの CE ビットが 1 である確率が高い) TCP コネクションに対する輻輳通知を抑える。簡単な定常状態解析により、我々が提案するアクティブキュー管理機構を用いることで、多段接続されたネットワークにおける、TCP コネクション間の公平性が改善できることを示す。

本稿の構成は以下の通りである。まず、2. 章では、TCP コネクション間の公平性に関する関連研究を紹介する。3. 章では、アクティブキュー管理機構に要求される、一般的な設計目標を議論する。4. 章では、本稿で提案するアクティブキュー管理機構のアルゴリズムを述べる。5. 章では、簡単な定常状態解析により、提案するアクティブキュー管理機構の有効性を示す。最後に 6. 章において、本稿のまとめと今後の課題について述べる。

2. 関 連 研 究

これまで、TCP コネクション間の公平性に関して、さまざまな研究が行なわれてきた。代表的なものとして、AIMD (Additive Increase Multiplicative Decrease) 型のフロー制御方式の、コネクション間の公平性を解析した研究がある [10-13] 。AIMD 型のウィンドウフロー制御方式では、ネットワーク中でパケット棄却がなければ、ウィンドウサイズ (ラウンドトリップ時間内に送出するパケット数) を α だけ線型に増加させる。一方、ネットワーク中でパケット棄却が発生した場合、AIMD 型のウィンドウフロー制御方式はパケット棄却を検知し、ウィンドウサイズを $(1-\beta)$ 倍に減少させる。このため、TCP の輻輳回避フェーズは、AIMD 型の輻輳制御方式の一種ととらえることができる。つまり、TCP の輻輳回避フェーズは、$\alpha = 1$ および $\beta = 0.5$ の場合に相当する。

文献 [10] は、ベクトル表記法を用いて、AIMD 型のフロー制御方式の公平性を解析した先駆的な論文である。一方、文献 [11] では、AIMD 型フロー制御方式のスループットおよびパケット棄却率を解析している。この論文では、AIMD 型フロー制御方式のスループット T が、近似的に次式で与えられることが示されている。

$$T = \frac{\sqrt{2-\beta}\sqrt{\alpha}}{(2R\sqrt{2\beta}\sqrt{p})} \tag{1}$$

ここで、p はパケット棄却率、R はラウンドトリップ時間である。この式からも、ラウンドトリップ時間 R の小さなコネクションや、パケット棄却率 p の小さなコネクションほどスループットが大きくなり、パケット転送を有利に行えることがわかる。

文献 [12] では、AIMD を含んだ Increase-Decrease 型のウィンドウフロー制御である、Binomial アルゴリズムの特性を解析している。Binomial アルゴリズムとは、ウィンドウサイズの増加量が α/w^k であり、ウィンドウサイズの減少量が β/w^l となる方式である。例えば、AIMD 型のフロー制御方式は、$k = 0$ かつ $l = 1$ の場合に相当する。文献 [12] では、文献 [10] のベクトル表記法を用いて、Binomial アルゴリズムの定常状態におけるコネクション間の公平性を解析している。その結果、(1) スループットは $1/p^{1/(k+l+1)}$ (p はパケット棄却率) に比例すること、(2) $k + l = 1$ かつ $l \leqq 1$ ならば、TCP-friendly (TCP とリンク帯域を公平に分けあう状態) になることなどが示されている。さらに、文献 [13] では、Binomial アルゴリズムをベクトル表記法を用いて解析し、定常状態におけるリンク利用率およびパケット棄却率を導出している。その結果、コネクション数 N が増加すれば、パケット棄却率が N^2 のオーダで増加することが示されている。

しかし、これらの研究では、(1) ネットワーク中には単一のボトルネックルータのみが存在すること、(2) すべてのコネクションの伝搬遅延が等しいこと、を仮定している。つまり、単一のルータ内に収容されているコネクション間の公平性や定常特性のみに着目している。従って、本稿で対象とする、ルータが多段接続されたネットワークにはそのまま適用することができない。

多段ネットワークにおける、TCP コネクション間の公平性を解析した研究としては、文献 [14, 8] などが挙げられる。文献 [14] では、すべてのコネクションのラウンドトリップ時間が等しいと仮定し、AIMD 型のフロー制御方式のコネクション間の公平性を解析している。その結果、AIMD 型フロー制御方式のスループットは、以下の関数 $F_A(x)$ を最大化するように割り当てられる ($F_A(x)$ 公平性) ことを示している。

$$F_A(x) = \sum_{i=1}^{N} \log \frac{x_i}{r_0 + \nu x_i} \tag{2}$$

ここで、N はコネクション数、x_i はコネクション i の転送レート、r_o および ν は AIMD 型フロー制御方式のゲイン (α と β に相当する) である。TCP の場合は、$r_o = 1/R$ および $\nu = 1/2$ となるため、式 (2) は次式のようになる。

$$F_A(x) = \sum_{i=1}^{N} \log \frac{x_i}{\frac{1}{R} + \frac{x_i}{2}} \tag{3}$$

ここで、R は TCP コネクションのラウンドトリップ時間である。

さらに、文献 [8] では、文献 [14] の結果を、ラウンドトリップ時間が異なる場合に拡張している。その結果、AIMD 型のフロー制御方式のスループットは、以下の $F_A^h(x)$ を最大化するように割り当てられる ($F_A^h(x)$ 公平性) ことを示している。

$$F_A^h(x) = \sum_{i=1}^{N} \frac{1}{R_i} \log \frac{x_i}{r_i + \nu_i x_i} \tag{4}$$

ここで、S はコネクションの集合であり、R_i はコネクション i のラウンドトリップ時間、r_i および ν_i はコネクション i のゲインである (α と β に相当する)。TCP の場合は、$r_i = 1/R_i$ および $\nu_i = 1/2$ となるため、式 (4) は次式のようになる。

$$F_A^h(x) = \sum_{i=1}^{N} \frac{1}{R_i} \log \frac{x_i}{\frac{1}{R_i} + \frac{x_i}{2}} \tag{5}$$

このように、多段接続されたネットワークでは、TCP コネクション間の公平性は Max-Min 公平性とならない。さらに、ここで紹介した研究では、すべて AIMD 型のフロー制御方式の、公平性を解析している。つまり、TCP のウィンドウ型フロー制御機構が、輻輳回避フェーズで動作している場合のみを解析している。実際には、TCP コネクションのパケット棄却率が大きい場合、複数のパケット棄却率が連続して発生し、タイムアウトが発生して、TCP はスロースタートフェーズで動作することも考えられる。このため、ホップ数の大きな TCP コネクション (パケット棄却率の大きな TCP コネクション) のスループットが著しく低下し、$F_A^h(x)$ 公平性よりも、さらに公平性が低下することが予想される。

なお、多段接続されたネットワークにおいて、TCP コネクション間の公平性を改善する方式もいくつか提案されている [15, 16] が、これらの方式はすべて送信側ホストの TCP 自体を変更する必要がある。しかし、送信側ホストの TCP をすべて変更するというのは、TCP がこれだけ普及した現在では極めて困難である。そこで本稿では、TCP を変更するのではなく、アクティブキュー管理機構を用いて、多段接続されたネットワークにおける TCP コネクション間の公平性を改善することを目的とする。まず、次章では、一般的なアクティブキュー管理機構に要求される設計目標を議論する。

3. 設 計 目 標

本章では、一般的なアクティブキュー管理機構が、満たすべき設計目標を議論する。なお、提案方式と設計目標との適合性については、4. 章で詳しく述べる。

(1) TCP の輻輳制御のタイムスケールの考慮

TCP は、エンド–エンド間で ACK (確認応答パケット) をやり取りし、その情報をもとに輻輳制御を行う。このため、TCP の輻輳制御は、TCP コネクションのラウンドトリップ時間のタイムスケールで動作する。アクティブキュー管理機構を設計する際には、このような TCP の輻輳制御のタイムスケールを考慮する必要がある。つまり、TCP がラウンドトリップ時間程度のタイムスケールで輻輳制御を行っているため、アクティブキュー管理機構の輻輳制御は、これと干渉しないように注意深く設計する必要がある。

例として、ネットワーク中の、他の全てのルータと制御情報をやり取りするような、アクティブキュー管理機構を考える。この時、制御情報の伝搬遅延は、TCP コネクションのラウンドトリップ時間とほぼ同じタイムスケールとなる。このように、TCP の輻輳制御の場合と同じタイムスケールの制御となるため、アクティブキュー管理機構の制御は、TCP の輻輳制御と独立して動作するのではなく、協調して動作させることが望ましいと考えられる。ラウンドトリップ時間と同じタイムスケールの制御の例として、ECN [9] や ICMP Source Quench [17] が挙げられる。一方、ネットワーク中の隣接ルータとのみ制御情報をやり取りするような、アクティブキュー管理機構であれば、TCP コネクションのラウンドトリップ時間よりも、短いタイムスケールでの輻輳制御は可能である。ラウンドトリップ時間よりも短かいタイムスケールの輻輳制御の例として、ルータ間のバックプレッシャ [18] などが挙げられる。

(2) TCP コネクション間の公平性の向上

一般に、パケット交換型のネットワークでは、コネクション間の帯域の割り当てが、Max-Min 公平性 [19] をみたすことが望ましい。Max-Min 公平とは、あるコネクションに割り当てられた帯域を増加させるためには、そのコネクションよりも割り当て帯域の小さなコネクションの帯域を減少させる必要がある状態をいう。一方、TCP では、TCP コネクション間の帯域割り当てが、F_A^h 公平性となることが知られている [8]。つまり、TCP コネクションに割り当てられる帯域は、(1) 伝搬遅延の短い TCP コネクションほど大きくなり、(2) ホップ数 (経由するルータ数) の小さなコネクションほど大きくなる。このため、アクティブキュー管理機構は、TCP コネクション間の公平性を、Max-Min 公平性に近づけることが望ましい。

(3) 実装の容易さ

アクティブキュー管理機構の設計に際しては、実際のルータ上への実装の容易さも考慮する必要がある。インターネットのコアルータ上にアクティブキュー管理機構を実装するためには、~ 10 Gbps 程度のスループットを実現し、少なくとも数千から数万程度の TCP コネクションを収容できる必要がある。このような高速な動作を実現するためには、ハードウェアでの実装が不可欠となるため、アクティブキュー管理機構のアルゴリズムは単純である必要がある。

なお、本稿で提案するアクティブキュー管理機構では、TCP コネクション間の公平性を実現するため、ルータにおいて TCP コネクションを識別し、TCP コネクションごとにパケットの処理方法を変えるという方式を用いる。そのため、提案方式の実装は複雑となり、この設計目標をみたさない。実装を容易にする手法は今後の課題である。

(4) 堅牢性の実現

アクティブキュー管理機構は一種の輻輳制御であるため、ネットワーク障害等に対する堅牢性を有する必要がある。つまり、ネットワーク障害が発生しても、アクティブキュー管理機構の制御が停止しないことが望ましい。一般に、アクティブキュー管理機構のような制御は、非集中型かつ分散型のアルゴリズムであることが望ましい。つまり、他のネットワーク機器が故障したり、アクティブキュー管理機構が用いる制御情報が何らかの原因で届かない場合であっても、アクティブキュー管理機構自体は正常に動作することが求められる。

(5) スケーラビリティの実現

アクティブキュー管理機構が、ネットワークの帯域や、収容する TCP コネクション数、ネットワーク中に存在するルータ数などに関する、スケーラビリティを持つことは重要である。つまり、さまざまなネットワークの帯域、TCP コネクション数、ネットワーク中に存在するルータ数の環境において、アクティブキュー管理機構の輻輳制御が良好に動作することが求められる。例えば、 ~ 10Gbps 程度のスループットを実現し、少なくとも数千から数万程度の TCP コネクションを収容できる必要がある。一方、アクティブキュー管理機構のようなフィードバック型 / フィードフォワード型制御では、その有効性は制御パラメータの設定に依存する。ここで、アクティブキュー管理機構の最適な制御パラメータ設定は、一般に、ネットワークの帯域、収容する TCP コネクション数などのシステムパラメータに依存する。このため、アクティブキュー管理機構の持つ制御パラメータを、システムパラメータの値に応じて自動的にチューニングする機構を持つことが望ましい。

なお、本稿で提案するアクティブキュー管理機構では、TCP フローごとにパケットの棄却方法を変えるため、TCP フローごとの情報を格納したテーブルを持つ必要がある。そのため、ルータに収容できる最大 TCP コネクション数は、このテーブルの大きさによって制限されてしまう。スケーラビリティを向上させる手法については、今後の課題である。

(6) 既存のネットワーク機器との互換性

新しく設計するすべてのアクティブキュー管理機構は、既存のネットワーク機器との互換性を保つことが望まれる。現実的には、既存のネットワーク中のすべてのルータを、新しいアクティブキュー管理機構対応のルータに置き換えることは不可能である。このため、従来の DropTail ルータや RED など他のアクティブキュー管理機構と混在した環境であっても良好に動作すべきである。また、新しいアクティブキュー管理機構を、既存のネットワークに段階的に導入できるようにすべきである。つまり、新しいアクティブキュー管理機構を、既存のネットワークに一部だけ導入した場合、ネットワーク全体の性能が劣化しないことが望まれる。さらに、ネットワーク中でのパケット棄却を元に輻輳制御を行う、さまざまなバージョンの TCP や TCP-friendly なレート制御方式をサポートすることが望ましい。

4. 提案方式のアルゴリズム

本章では、提案するアクティブキュー管理機構の、動作アルゴリズムを説明する。基本的なアイデアは、代表的なアクティブキュー管理機構である RED を ECN モード [9] (パケットを廃棄するのではなく、IP ヘッダの CE ビット [20] を 1 にする) で動作させ、ルータにおいて TCP コネクションを区別することにより、TCP コネクション間の公平性を改善するというものである。具体的には、提案するアクティブキュー管理機構は、ルータにおいて各 TCP フローを識別し、それぞれのフロー i ごとに IP ヘッダの CE ビットが 1 である確率 $p_c(i)$ を推定する。また、通常の RED とまったく同じアルゴリズムを用いて、ルータにパケットが到着するたびに、アクティブキュー管理機構の

パケット廃棄確率 p_a を計算する。この時、$p_c(i) \geqq p_a$ であれば、アクティブキュー管理機構は IP ヘッダの CE ビットを変更しない。一方、$p_c(i) < p_a$ であれば、通常の RED と同様に確率 p_a で IP ヘッダの CE ビットを 1 にする。通常の RED では、ホップ数の大きな TCP コネクションほど、IP ヘッダの CE ビットが 1 である確率が大きくなる。よって、提案方式を採用することにより、ホップ数の多い TCP コネクションに対する不公平性を改善できると考えられる。

以下では、提案方式するアクティブキュー管理機構のアルゴリズムの詳細を述べる。提案するアルゴリズムの仮想コードを図 2 に示す。提案するアルゴリズムは、基本的には ECN モードで動作する RED と同じアルゴリズムであるが、確率的に IP ヘッダの CE ビットを 1 にする方法が異なっている。

(1) パケットが到着するごとに、IP ヘッダの ECT (ECN-Capable Transport) ビットおよび CE ビットを調べる (23 行目)。

(2) ECT ビットが 0 であれば、その TCP コネクションは ECN をサポートしていないことを意味する。この場合、ルータは通常の RED として動作する (34–36 行目)。一方、ECT ビットが 1 の場合は、ルータは以下のようなアルゴリズムで動作する (24–32 行目)。

(3) 送信側および受信側の IP アドレスと、送信側および受信側のポート番号を使用して、TCP フローを識別する。以下では、TCP フローの識別子を i とする (24 行目)。

(4) TCP フロー i から到着するパケットについて、CE ビットが 1 である確率 $p_c(i)$ を、ローパスフィルタの一種である、指数移動平均を用いて推定する。具体的には、以下の式を計算して推定する (25 行目)。ここで、ν は EWMA の重みである。

(5) RED と同じアルゴリズムを用いて、平均キュー長からパケット棄却率 p_a を計算する (07–21 行目)。

(6) $p_c(i) < p_a$ なら、確率 p_a で IP ヘッダの CE ビットを 1 にする (28–30 行目)。しかし、$p_c(i) \geqq p_a$ なら、IP ヘッダの CE ビットを変更せず、そのまま下流のルータへ転送する (32 行目)。

次に、3. 章で議論した、6 個の設計目標を、提案するアクティブキュー管理機構がどの程度満たしているかを説明する。

(1) TCP の輻輳制御のタイムスケールを考慮

提案方式では ECN を利用しているため、TCP のラウンドトリップ時間と同じタイムスケールの制御となっている。

(2) TCP コネクション間の公平性を向上

提案方式は、多段接続されたネットワークにおいて、TCP コネクション間の公平性を向上させている。通常の RED では、ホップ数の多い TCP コネクションほど、IP ヘッダの CE ビットが 1 である確率が高くなり、その結果、TCP コネクション間の公平性が低下していた。提案方式では、すでに CE ビットが 1 である TCP コネクションに対しては、CE ビットを変更しないことにより、ホップ数の多い TCP コネクションのスループット向上をはかっている。ただし、式 (1) に示されているように、TCP コネクションのスループットが、ラウンドトリップ時間に依存するという問題は解決されていない。

この問題を解決するために、ルータにおいてラウンドトリップ時間を測定し、その測定結果に応じてパケット処理方法を変えるなどの方法も考えられる。しかし、一般に TCP コネクションのラウンドトリップ時間の変動は大きく、また、ルータにおけるラウンドトリップ時間の正確な推定も困難である。さらに、インターネットのルーティングは必ずしも対称ではないため、ルータにおいて TCP コネクションのラウンドトリップ時間が測定できない場合もありえる。従って、アクティブキュー管理機構が、TCP コネクションのラウンドトリップ時間を測定するという方法は困難であると考えられる。TCP のスループットがラウンドトリップ時間に依存するという問題は、TCP の制御頻度がラウンドトリップ時間ごとであるという点に起因している。これは TCP の本質的な問題であるため、アクティブキュー管理機構による解決は困難であると考え、本稿ではラウンドトリップ時間に関する不公平性は扱わない。

(3) 実装の容易さ

提案するアクティブキュー管理機構は、ルータにおいて TCP フローを識別し、TCP フローごとの内部変数を保持するため、RED 等のアクティブキュー管理機構と比較すると実装は複雑である。実装を容易にする手法については今後の課題である。ただし、RIO [21] と同様の方法を用いることにより、TCP コネクション単位の優先制御をサポートすることは可能である。

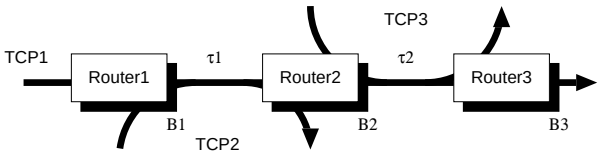

図 1　解析モデル
Fig. 1　Analytic model

(4) 堅牢性の実現

提案するアクティブキュー管理機構は、上流のルータがセットする、IP ヘッダの CE ビットの情報のみを利用する。また、IP ヘッダの ECT ビットが 0 であれば、通常の RED として動作する。このため、提案するアクティブキュー管理機構は、DropTail ルータや他のアクティブキュー管理機構を用いたルータなど、他のルータとの共存が可能である。また、提案するアクティブキュー管理機構を用いたルータを、ネットワーク中に数多く導入すればするほど、TCP コネクション間の公平性が改善されることが期待できる。このことは、ECN 導入へのインセンティブになると考えられる。

(5) スケーラビリティの実現

提案するアクティブキュー管理機構は、TCP フローごとにパケットの棄却方法を変えるため、TCP フローごとの情報を格納したテーブルを持つ必要がある。そのため、ルータに収容できる最大 TCP コネクション数は、このテーブルの大きさによって制限されてしまう。スケーラビリティを向上させる手法については、今後の課題である。なお、提案するアクティブキュー管理機構は、基本的に RED と同じアルゴリズムを採用しているため、Adaptive RED [22] と同様の手法で、制御パラメータの自動チューニングは可能である。

(6) 既存のネットワーク機器との互換性

提案するアクティブキュー管理機構は、ECN (IP ヘッダの ECT ビットと CE ビット) を利用するだけであるため、ECN に対応したすべてのトランスポート層プロトコルに適用できる。また、送信側ホストが ECN に対応していない場合 (IP ヘッダの ECT ビットが 0 の場合)、RED とまったく同じように動作する。従って、ネットワーク中の一部のルータのみに、提案するアクティブキュー管理機構を導入することが可能である。さらに、ネットワーク中のすべてのルータに、提案するアクティブキュー管理機構を導入すれば、TCP コネクション間の公平性がより改善され、最も良好な結果が得られると考えられる。

これに加えて、提案するアクティブキュー管理機構は、基本的に RED と同じアルゴリズムを採用しているため、RED に関するさまざまな研究成果をそのまま適用することができる。例えば、SRED [5] や FRED [7] のような手法を用いることにより、同一のルータ内に収容された TCP コネクション間の公平性を、さらに向上することも可能である。

5. 性能評価

本章では、簡単な定常状態解析により、提案するアクティブキュー管理機構によって、TCP コネクション間の公平性がどの程度向上するかを評価する。

解析モデルを図 1 に示す。解析モデルは、3 つのルータ (ルータ 1 ~ 3) および 3 種類の TCP コネクション (TCP 1 ~ 3) から構成されている。TCP 1 は、ルータ 1 からルータ 3 までの 2 ホップの TCP コネクションである。TCP 2 および TCP 3 は、ルータ 1 からルータ 2 まで、もしくはルータ 2 からルータ 3 までの 1 ホップの TCP コネクションである。TCP i のコネクション数を N_i と表記する。

ルータ j ($1 \leq j \leq 3$) の処理能力を B_j、定常状態における平均キュー長を q_j^*、ルータ 1–ルータ 2 間、およびルータ 2–ルータ 3 間のリンクの伝搬遅延を、それぞれ τ_1 および τ_2 とする。TCP の送信側ホスト–ルータ間、および TCP の受信側ホスト–ルータ間のすべてのリンク帯域は、ルータの処理能力 B_j よりも十分大きく、かつリンクの伝搬遅延はすべて 0 と仮定する。また、すべての TCP コネクションは常に転送するデータを持っていると仮定する。

定常状態において、TCP のスループットは近似的に次式で与えられることが知られている [23]。

$$T(R,p) \simeq \frac{1}{R}\sqrt{\frac{3}{2p}} \tag{6}$$

ここで、R は TCP コネクションのラウンドトリップ時間、p はネットワーク中でのパケット棄却率、もしくは IP ヘッダの CE ビットが 1 である確率である。

定常状態における、ルータ j において IP ヘッダの CE ビットが 1 である確率を p_j とする。まず、ルータが RED の場合を考える。RED の場合、ルータ 1 およびルータ 2 は、それぞれ独立に IP ヘッダの CE ビットをセットするため、TCP 1 の IP ヘッダの CE ビットが 1 である確率 $p_{1,2}$ は次式で与えられる。

$$p_{1,2} = 1 - \Pi_{j=1}^{2}(1-p_j) \tag{7}$$

一方、本稿で提案するアクティブキュー管理機構では、定常状態において、TCP 1 の IP ヘッダの CE ビットが 1 である確率 $p_{1,2}$ は次式で与えられる。

$$p_{1,2} = \max(p_1, p_2) \tag{8}$$

また、TCP i の各コネクションのスループットを T_i と表記する。TCP i のラウンドトリップ時間 R_i を、リンクの伝搬遅延とルータにおけるキューイング遅延で近似すれば、以下のような関係が得られる。

$$T_1 = T(R_1, p_{1,2}) \simeq T(R_1, \sum_{j=1}^{2} p_j) \tag{9}$$

$$T_2 = T(R_2, p_1) \tag{10}$$

$$T_3 = T(R_3, p_2) \tag{11}$$

$$R_1 = \sum_{j=1}^{2}\left(2\tau_j + \frac{q_j^*}{B_j}\right) \tag{12}$$

$$R_2 = 2\tau_1 + \frac{q_1^*}{B_1} \tag{13}$$

$$R_3 = 2\tau_2 + \frac{q_2^*}{B_2} \tag{14}$$

図 1 のネットワークでは、定常状態においてルータの処理能力 B_j と TCP コネクションの実効スループットが等しくなる。このことから、以下のような関係が成立する。

$$N_1\,T_1 + N_2\,T_2 = \frac{B_1}{1-p_1} \tag{15}$$

$$N_1\,T_1 + N_3\,T_3 = \frac{B_2}{1-p_2} \tag{16}$$

上式を、p_1 および p_2 について解くことにより、定常状態における TCP コネクションのスループットを導出することができる。

最後に、いくつかの数値例により、提案するアクティブキュー管理機構を用いることにより、RED と比較して、TCP コネクション間の公平性がどの程度改善されるかを示す。数値例では、以下のようなパラメータを用いた。$N_1 = 1$、$N_2 = 1$、$N_3 = 1$、ルータの平均キュー長 $q_1^* = q_2^* = 10$ [packet]。

表 1–表 3 に、ルータの処理能力 B_j およびリンクの伝搬遅延 τ_j を変化させた時の定常状態解析の結果を示す。これらの解析結果は、定常状態における CE ビットマーキング確率 p_j、および TCP のスループット T_i を示している。これらの解析結果より、まず CE ビットマーキング確率 p_j に着目すると、提案方式の場合の値が、RED の場合よりも値が大きくなっている。これは、提案方式を用いることで、TCP 1 のパケットが後段のルータ 2 でマーキングされにくくなり、TCP 1 のスループットが大きくなるためと考えられる。

次に、TCP コネクション間の公平性に着目する。 以下では、TCP コネクションのスループットの比 (T_2/T_1 および T_3/T_1) を、公平性の指標と考える。RED の場合、TCP コネクションのスループットの比は、 ルータの処理能力 B_j やリンクの伝搬遅延 τ_j によらず、 $T_2/T_1 = T_3/T_1 = 2.83$ となっている。 一方、提案方式を用いた場合は、 $T_2/T_1 = T_3/T_1 = 2.0$ となっている。 Max-Min 公平性では、 $T_2/T_1 = T_3/T_1 = 1.0$ となる。このことから、提案するアクティブキュー管理機構を用いることにより、RED の場合と比較して、 TCP コネクション間の公平性が約 30 %改善されていることがわかる。 RED の場合、ホップ数が大きい TCP 1 は、CE ビットマーキング確率が TCP 2 や TCP 3 よりも大きくなる。そのため、TCP 1 のスループットは、TCP 2 や TCP 3 よりも小さく抑えられる。一方、提案するアクティブキュー管理機構を用いることにより、 TCP 1 の CE ビットマーキング確率が減少し、 その結果、TCP 1 のスループットが向上し、 TCP コネクション間の公平性が向上している。

表 1　定常状態における CE ビットマーキング確率および TCP スループット ($B_1 = B_2 = 0.2$ [packet/ms]、$\tau_1 = \tau_2 = 10$ [ms])

	p_1	p_2	T_1	T_2	T_3
RED	0.0136413	0.0136413	0.264817	0.749015	0.749015
提案方式	0.0166507	0.0166507	0.338978	0.677955	0.677955

表 2　定常状態における CE ビットマーキング確率および TCP スループット ($B_1 = B_2 = 0.4$ [packet/ms]、$\tau_1 = \tau_2 = 10$ [ms])

	p_1	p_2	T_1	T_2	T_3
RED	0.00834107	0.00834107	0.263401	0.74501	0.74501
提案方式	0.0102051	0.0102051	0.33677	0.67354	0.67354

表 3　定常状態における CE ビットマーキング確率および TCP スループット ($B_1 = B_2 = 0.2$ [packet/ms]、$\tau_1 = \tau_2 = 20$ [ms])

	p_1	p_2	T_1	T_2	T_3
RED	0.00834107	0.00834107	0.263401	0.74501	0.74501
提案方式	0.0102051	0.0102051	0.336771	0.673542	0.673542

6. まとめと今後の課題

本稿では、ホップ数の異なる TCP コネクション間の公平性を向上させる、新しいアクティブキュー管理機構を設計した。まず、アクティブキュー管理機構が実現するべき一般的な設計目標を議論した。本稿では、これらの設計目標の中でも、特に、TCP コネクション間の公平性を Max-Min 公平性に近づけることを目標として、ECN を利用したアクティブキュー管理機構を設計した。さらに、簡単な定常状態解析により、提案するアクティブキュー管理機構の性能評価を行なった。その結果、提案方式を用いることにより、従来の RED に比べて、TCP コネクション間の公平性が 30 % 向上することを示した。

本稿では、ECN を利用することにより、ホップ数の異なる TCP コネクション間の公平性を向上させるアクティブキュー管理機構を設計した。しかし、4. 章で議論したように、我々が提案したアクティブキュー管理機構でも、TCP コネクションの伝搬遅延の違いによって生じる不公平性は改善できていない。また、ルータにおいて TCP コネクションごとのテーブルを保持する必要があるため、実装の容易さやスケーラビリティという点は不十分である。従って、今後は、本稿で提案したアクティブキュー管理機構を、これらの設計目標をみたすように改良する予定である。

謝　　辞

アクティブキュー管理機構を設計するにあたり、有意義な議論をしていただいた、大阪大学情報科学研究科教授の今瀬真氏に感謝いたします。

文　　献

[1] B. Braden et al., “Recommendations on queue management and congestion avoidance in the Internet,” *Request for Comments (RFC) 2309*, Apr. 1998.

[2] V. Jacobson and M. J. Karels, “Congestion avoidance and control,”

in *Proceedings of ACM SIGCOMM '88*, pp. 314–329, Nov. 1988.

[3] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Transactions on Networking*, vol. 1, pp. 397–413, Aug. 1993.

[4] S. Floyd, "Recommendations on using the gentle variant of RED," May 2000. available at `http://www.aciri.org/floyd/red/gentle.html`.

[5] T. J. Ott, T. V. Lakshman, and L. Wong, "SRED: Stabilized RED," in *Proceedings of IEEE INFOCOM '99*, pp. 1346–1355, Mar. 1999.

[6] J. Aweya, M. Ouellette, and D. Y. Montuno, "A control theoretic approach to active queue management," *Computer Networks*, vol. 36, pp. 203–235, 2001.

[7] D. Lin and R. Morris, "Dynamics of random early detection," in *Proceedings of ACM SIGCOMM '97*, pp. 127–137, Oct. 1997.

[8] M. Vojnovid, J.-Y. L. Boudec, and C. Boutremans, "Global fairness of additive-increase and multiplicative-decrease with heterogeneous round-trip times," in *Proceedings of IEEE INFOCOM 2000*, pp. 1303–1312, Mar. 2000.

[9] K. Ramakrishnan and S. Floyd, "A proposal to add explicit congestion notification (ECN) to IP," *Request for Comments (RFC) 2481*, Jan. 1999.

[10] D.-M. Chiu and R. Jain, "Analysis of the increase and decrease algorithms for congestion avoidance in computer networks," *Computer Networks and ISDN Systems*, vol. 17, pp. 1–14, June 1989.

[11] S. Floyd, M. Handley, and J. Padhye, "A comparison of equation-based and AIMD congestion control," May 2000. available at `http://www.icir.org/tfrc/aimd.pdf`.

[12] D. Bansal and H. Balakrishnan, "Binomial congestion control algorithms," in *Proceedings of IEEE INFOCOM 2001*, pp. 631–640, Apr. 2001.

[13] D. Loguinov and H. Radha, "Increase-decrease control for real-time streaming: scalability," in *Proceedings of IEEE INFOCOM 2002*, pp. 525–534, June 2002.

[14] P. Hurley, J. L. Boudec, and P. Thiran, "A note on the fairness of additive increase and multiplicative decrease," in *Proceedings of ITC-16*, pp. 467–478, June 1999.

[15] S. Floyd, "Connections with multiple congested gateways in packet-switched networks part1: One-way traffic," *ACM Computer Communication Review*, vol. 21, pp. 30–47, Oct. 1991.

[16] T. H. Henderson, E. Sahouria, S. McCanne, and R. H. Katz, "On improving the fairness of TCP congestion avoidance," in *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM)*, vol. 1, pp. 539–544, Nov. 1998.

[17] F. Baker, "Requirements for IP version 4 routers," *Request for Comments (RFC) 1812*, June 1995.

[18] T. Blackwell, H. T. Kung, and D. Lin., "Credit-based flow control for ATM networks," *IEEE Network Magazine*, vol. 90, pp. 40–48, Mar. 1994.

[19] D. Bertsekas and R. Gallager, *Data Networks*. Englewood Cliffs, New Jersey: Prentice-Hall, 1987.

[20] D. B. K. Ramakrishnan, S. Floyd, "The addition of explicit congestion notification (ECN) to IP," *Request for Comments (RFC) 3168*, Sept. 2001.

[21] D. D. Clark and W. Fang, "Explicit allocation of best-effort packet delivery service," *IEEE/ACM Transactions on Networking*, vol. 6, pp. 362–373, Aug. 1998.

[22] S. Floyd, R. Gummadi, and S. Shenker, "Adaptive RED: an algorithm for increasing the robustness of RED," Aug. 2001. available at `http://www.icir.org/floyd/papers/adaptiveRed.pdf`.

[23] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP Reno performance: a simple model and its empirical validation," *IEEE/ACM Transactions on Networking*, vol. 8, pp. 133–145, Apr. 2000.

```
01: for all pCE
02:   pc(0) = 0
03: avg = 0
04: count = -1
05:
06: for each packet arrival
07:   if the queue is nonempty
08:     avg = (1 - wq) * avg + wq * q
09:   else
10:     m = f(time - q_time)
11:     avg = (1 - wq)^m
12:
13:   if minth <= avg < maxth
14:     increment count
15:     pb = maxp(avg - minth)/(maxth - minth)
16:     pa = pb/(1 - count * pb)
17:   else if maxth <= avg
18:     pa = 1.0
19:     count = 0
20:   else
21:     count = -1
22:
23:   if ECT == 1
24:     i = flow_id(src_ip, dst_ip, src_port, dst_port)
25:     pc(i) = nu * CE + (1 - nu) * pc(i)
26:
27:     if pc(i) < pa
28:       with probability pa:
29:         mark the arriving packet
30:       count = 0
31:     else
32:       do nothing
33:
34:   else
35:     with probability pa:
36:       mark arriving packet

Variables:
    avg: average queue size
    q_time: start of the queue idle time
    count: packets since last marked packet
    pa: current packet marking probability
    q: current queue size
    time: current time
    pc: CE bit marked probability for arriving packets
    i: flow id
    nu: CE bit weight

Functions:
    f(t): a linear function of the time t
    flow_id(): get flow id from IP addresses and port numbers
               of arriving packet

Fixed parameters:
    wq: queue weight
    minth: minimum threshold for queue
    maxth: maximum threshold for queue
    maxp: maximum value for pb
    nu: weight of marked CE bit
```

図 2　提案するアルゴリズムの仮想コード
Fig. 2　Pseudo Code of proposed algorithm