流体近似法および待ち行列理論を組み合わせた TCP のフィードバック型輻輳制御機構のモデル化

高垣 景一† 大崎 博之‡ 村田 正幸‡

†大阪大学 大学院基礎工学研究科
〒 560-8531 大阪府豊中市待兼山町 1-3
Phone: 06-6850-6616, Fax: 06-6850-6589
E-mail: takagaki@ics.es.osaka-u.ac.jp

‡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
Phone: 06-6879-8793, Fax: 06-6879-8794
E-mail: {oosaki,murata}@cmc.osaka-u.ac.jp

あらまし インターネットでは、TCP (Transmission Control Protocol) においてウィンドウ型のフロー制御方式が用いられている。これまで、さまざまな研究者らによって TCP の解析が行われている。従来の研究では、ネットワークにおけるパケット棄却率を一定と仮定し、この時の TCP の平均的な特性を解析したものがほとんどである。しかし現実のネットワークでは、TCP のウィンドウサイズが変化すれば、それによってネットワークにおけるパケット棄却率は変化する。そこで本稿では、TCP の輻輳制御機構とネットワークをフィードバックシステムとしてモデル化し、TCP の過渡特性を解析する。つまり、TCP はネットワークでのパケット棄却率を入力とし、ウィンドウサイズを出力とするシステムとしてモデル化する。一方、ネットワークは TCP のウィンドウサイズを入力とし、パケット棄却率を出力とする一つのシステムとしてモデル化する。なお、ネットワークは、TCP 以外のバックグラウンドトラヒックをも考慮した、待ち行列としてモデル化する。得られたモデルに対して過渡特性解析を行い、バックグラウンドトラヒックの量や、TCP のコネクション数などによって、TCP の過渡特性がどのように変化するかを定量的に明らかにする。
和文キーワード TCP、流体近似法、待ち行列理論、フィードバックシステム、過渡特性

Modeling Feedback Congestion Control Mechanism of TCP using Fluid Flow Approximation and Queueing Theory

Keiichi Takagaki† Hiroyuki Ohsaki‡ Masayuki Murata‡

†Graduate School of Engineering Science, Osaka University
1-3 Machikaneyama, Toyonaka, Osaka 560-8531, Japan
Phone: +81-6-6850-6616, Fax: +81-6-6850-6589
E-mail: m-morita@ics.es.osaka-u.ac.jp

‡Cybermedia Center, Osaka University
1-30 Machikaneyama, Toyonaka, Osaka 567-0043, Japan
Phone: +81-6-6879-8793, Fax: +81-6-6879-8794
E-mail: {oosaki, murata}@cmc.osaka-u.ac.jp

Abstract The Internet uses a window-based flow control mechanism in TCP (Transmission Control Protocol). In the literature, there have been a significant number of analytical studies on TCP. Most of those studies have focused on the statistical behavior of TCP by assuming a constant packet loss probability in the network. However, the packet loss probability, in reality, changes according to packet transmission rates from TCP connections. In this paper, we explicitly model the interaction between the congestion control mechanism of TCP and the network as a feedback system. In other words, we model the congestion control mechanism of TCP as a system having the input (packet loss probability) and the output (window size). We also model the network as a system having the input (window size) and the output (packet loss probability). The network is modeled as a queueing system by assuming a single bottleneck link. Using our analytic model, the transient behavior of TCP connections is quantitatively investigated with several numerical examples.
key words TCP, Fluid Flow Approximation, Queueing Theory, Feedback System, Transient Behavior

1 研究の背景

パケット交換ネットワークにおいて、データ系のサービスを効率的に収容するためには、フィードバック型の輻輳制御が不可欠である。現在のインターネットでは、フィードバック型の輻輳制御機構として、ウィンドウ型のフロー制御方式が TCP (Transmission Control Protocol) において使用されている。TCP は、ネットワーク内でパケットが棄却された場合に、棄却されたパケットを再び受信側ホストに送出するパケット再送機構と、ネットワークの輻輳状況に応じてウィンドウサイズを変更し、送出するパケット数を調整する輻輳制御機構を持っている。

現在広く利用されている TCP Reno では、ネットワークからのフィードバック情報として、ネットワーク内部でのパケット棄却の有無を利用している [1, 2] 。これは、パケット棄却の発生が、ネットワークが輻輳状況にあることを意味するためである。TCP Reno では、パケット棄却が発生しない限り、送信側ホストは連続的にウィンドウサイズを増加させる。やがてウィンドウサイズが帯域遅延積 (利用可能帯域 × コネクションの伝搬遅延時間) を超えると、余分なパケットはルータのバッファに蓄えられる。さらにウィンドウサイズが増加すると、ルータにおいてバッファあふれが発生し、その結果パケットが棄却される。送信側ホストでは、同じシーケンス番号を持つ ACK パケットを複数受信することなどにより、ネットワーク内でのパケット棄却を検出し、ウィンドウサイズを減少させる。これによりパケット棄却が発生しなくなれば、再びウィンドウサイズを連続的に増加させる。TCP Reno はこのような制御を繰り返し行うことにより、利用可能帯域を有効に利用する。

これまで、さまざまな研究者らによって TCP の解析がおこなわれてきた [3, 4, 5]。文献 [3, 4] では、ネットワークにおけるパケット棄却率を一定と仮定し、この時の TCP Reno の平均ウィンドウサイズやスループットなどを導出している。しかし現実のネットワークでは、TCP のウィンドウサイズが変化すれば、それによってネットワークにおけるパケット棄却率は変化する。そこで本稿では、TCP の輻輳制御機構およびネットワークをあわせてフィードバックシステムととらえることにより、TCP の過渡特性を解析する。つまり、TCP の輻輳制御機構を、ネットワークでのパケット棄却率を入力とし、TCP のウィンドウサイズを出力とするシステムと考える。一方、ネットワークを、TCP のウィンドウサイズを入力とし、ネットワークでのパケット棄却率を出力とするシステムと考える。なお、TCP の解析モデルとして、文献 [3, 4, 5] で提案されている 4 種類のモデルを用いる。ネットワークは、TCP 以外のトラヒックをも考慮した、待ち行列としてモデル化する。

文献 [6] では、ネットワークを M/D/1/m 待ち行列としてモデル化し、TCP の性能を解析している。しかし、ここでは TCP Tahoe のみを対象としており、TCP Reno の高速再送機能などはモデル化されていない。また、定常状態のみに着目しており、TCP の過渡特性はまったく明らかにされていない。また、文献 [5, 7] では、TCP Reno と RED (Random Early Detection) ゲートウェイを対象として、フィードバックシステムとして TCP の性能を解析している。しかし文献 [7] では、定常状態のみに着目しており、過渡特性については定性的な評価しか行なわれていない。文献 [5] では、制御理論を適用することによって TCP の安定性および過渡特性が解析されている。ここでは、RED ゲートウェイを離散時間システムとしてモデル化しているが、本稿では Drop-Tail ゲートウェイを待ち行列としてモデル化する。

本稿の構成は以下の通りである。2 章では、TCP およびネットワークを、フィードバックシステムとしてどのようにモデル化するかを説明する。また、TCP の解析モデルを 4 種類説明する。3 章では、シミュレーション結果と比較することにより、どの解析モデルが過渡特性解析に適切であるかを検討する。さらに、4 章において TCP の過渡特性解析を行い、バックグラウンドトラヒックの量や、TCP のコネクション数などによって、TCP の過渡特性がどのように変化するかを定量的に示す。最後に、 5 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 解析モデル

2.1 ネットワーク全体のモデル化

本稿では、図 1 に示すようなネットワークを対象とする。複数の TCP コネクションがボトルネックリンクを共有しており、ボトルネックリンクを流れる TCP 以外のトラヒック (バックグラウンドトラヒック) をも考慮した解析を行う。本稿ではネットワーク全体をフィードバックシステムととらえ、送信側ホストにおける TCP の輻輳制御と、送信側ホストから見たネットワークを、それぞれ個別のシステムとしてモデル化する。TCP の輻輳制御機構は、ネットワーク内でのパケット棄却の有無をもとにウィンドウサイズを変更する、ウィンドウ型のフロー制御方式である。そのため、ネットワークでのパケット棄却率が低い場合にはウィンドウサイズが大きくなり、ネットワークでのパケット棄却率が高い場合には、ウィンドウサイズが小さくなるといった傾向がある。そこで本稿では、送信側ホストにおける TCP の輻輳制御機構を、ネットワークでのパケット棄却率を入力とし、ウィンドウサイズを出力とするシステムと考える。一方、送

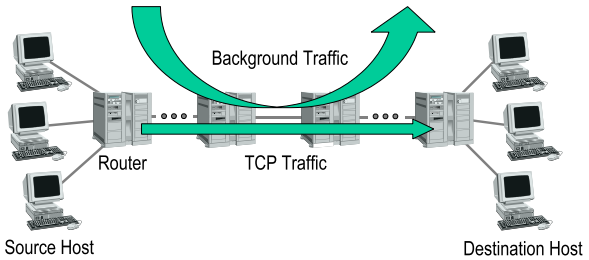

図 1: 解析するネットワークモデル

信側ホストから見たネットワークでは、流入するパケッ

トの量が増えると、ボトルネックリンクへ向かうルータのバッファ内パケット数が増加する。その結果、バッファあふれによるパケット棄却が発生しやすくなり、パケット棄却率が高くなる。TCP のウィンドウサイズが大きいと、ネットワークに流入するパケットの量が増える。このため、送信側ホストから見たネットワークを、TCP のウィンドウサイズを入力とし、パケット棄却率を出力とするシステムと考える。つまり、TCP の輻輳制御機構を含むネットワーク全体を、これらの 2 つのシステムが相互に作用しながら動作する 1 つのフィードバックシステムととらえる (図 2)。以下では、TCP の輻輳制御機構および送信側ホストからみたネットワークを、それぞれをどのようにモデル化するかを説明する。

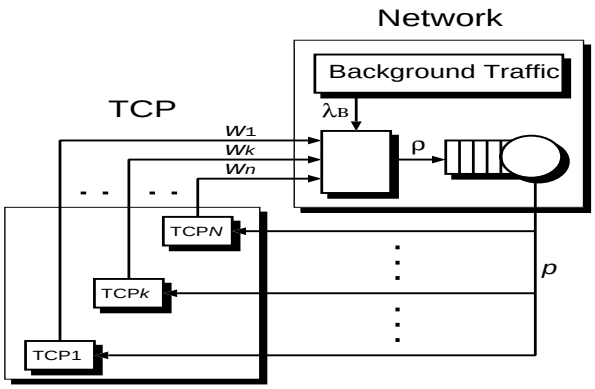

図 2: 解析モデル

2.2 TCP から見たネットワークのモデル化

ネットワーク中に存在するボトルネックリンクは単一であると仮定する。以下では、ボトルネックリンクの直前のルータを、ボトルネックルータと呼ぶ。ボトルネックルータとして、Drop-Tail ルータを考える。ネットワークが定常であれば、ボトルネックリンクへ向かうルータのバッファを単一の待ち行列でモデル化することができる。つまり、ボトルネックルータへのパケット到着率およびボトルネックリンク容量 (サービス率) から、待ち行列理論を適用することにより、ボトルネックルータでのパケット棄却率や、ボトルネックルータに到着したパケットの平均待ち時間などを求めることができる。実際には、TCP の輻輳制御によって TCP が送出するパケット量は振動的に変化するため、ネットワークは定常ではない。しかし、3 章で示すように、比較的大きな時間間隔 (例えば TCP のラウンドトリップ時間) で考えれば、待ち行列によってうまくモデル化することが可能である。以下では、送信側ホストから見たネットワークを、待ち行列を用いてどのようにモデル化するかを説明する。

TCP のコネクション数を N、i $(1 \leq i \leq N)$ 番目の TCP コネクションのウィンドウサイズおよびラウンドトリップ時間をそれぞれ w_i および r_i とする。TCP コネクションが連続的にパケットを送出すると仮定すれば、TCP のパケット送出レートは w_i/r_i によって近似できる。従って、ボトルネックルータへの平均パケット到着率 λ は以下の式で与えられる。

$$\lambda \simeq \sum_{i=1}^{N} \frac{w_i}{r_i} + \lambda_B \quad (1)$$

ここで λ_B は、ボトルネックルータに到着するバックグラウンドトラヒックの平均到着率である。ボトルネックリンクの容量を μ とすれば、ボトルネックルータの利用率 ρ は以下の式で与えられる。

$$\rho = \frac{\lambda}{\mu} \quad (2)$$

ボトルネックルータに到着するパケットの到着過程や、ボトルネックルータのパケット処理時間の分布、バッファサイズなどによって、適用できる待ち行列モデルが異なる。本稿では、簡単のため $M/M/1/m$ を用いてネットワークをモデル化する。

2.3 TCP の輻輳制御機構のモデル化

TCP の輻輳制御機構では、タイムアウトによるパケット棄却の検出や、高速再送機能などさまざまな制御が行われており、そのアルゴリズムは複雑である。このため、これらのすべての機能を数学的にモデル化することは困難である。本稿では、TCP の輻輳制御のうち、TCP の輻輳回避 (Congestion Avoidance) フェーズにおける基本的な動作 (ウィンドウ型フロー制御、高速再送制御、タイムアウト) をモデル化する。定常状態においても、TCP のウィンドウサイズは振動的に変化し、一定値に収束することはない。送信側ホストでは、パケット棄却が発生するまで、ACK パケットを受信するごとにウィンドウサイズを $1/w$ だけ増加させる。やがてウィンドウサイズが大きくなりすぎると、ネットワーク内でパケット棄却が発生する。TCP は、同じシーケンス番号を持つ ACK パケットを複数受信することにより、パケット棄却の発生を検出し、ウィンドウサイズを 1/2 に減少させる。ネットワーク内でパケット棄却が連続的に発生すると、同じシーケンス番号を持つ複数の ACK パケットを受信する前にタイムアウトが起こることがある。この場合にはウィンドウサイズを 1 にする。TCP の輻輳回避フェーズでは、このような制御が繰り返し行われる。文献 [4, 8, 5] では、輻輳回避フェーズにおける TCP の動作をモデル化し、パケット棄却率とウィンドウサイズとの関係を導出している。本稿では、これらの解析結果をもとに、以下のような 4 種類の解析モデルを考える。

- **モデル A**
  文献 [4] では、ネットワークでのパケット棄却率 p が一定であると仮定して、ある TCP コネクションの定常状態でのウィンドウサイズの変化をモデル化している。これにより、定常状態における、ある TCP コネクションの平均スループットが導出されている。解析においては、(1) 輻輳回避フェーズの最初のウィンドウサイズと、次の輻輳回避フェーズの最初のウィンドウサイズが等しく、(2) 各輻輳回

避フェーズ中に $1/p$ 個のパケットを送出する、と仮定している。その結果、定常状態における TCP の平均スループット λ_T が、以下のように求められている。

$$\lambda_T = \frac{\frac{1-p}{p} + E[W] + \hat{Q}(E[W])\frac{1}{1-p}}{r\left(\frac{b}{2}E[W]+1\right) + \hat{Q}(E[W])T_o\frac{f(p)}{1-p}}$$

ただし、

$$E[W] = \frac{2+b}{3b} + \sqrt{\frac{8(1-p)}{3bp} + \left(\frac{2+b}{3b}\right)^2}$$

$$\hat{Q}(w) = \frac{(1-(1-p)^3)(1+(1-p)^3(1-(1-p)^{w-3}))}{(1-(1-p)^w)}$$

$$f(p) = 1+p+2p^2+4p^3+8p^4+16p^5+32p^6$$

ここで r は平均ラウンドトリップ時間、b は遅延 ACK のパラメータであり、受信側ホストが b 個のパケットにつき 1 個の ACK パケットを送信側ホストに返送することを意味する。また T_O は TCP のタイムアウト時間である。なお、$\hat{Q}(w)$ は、ウィンドウサイズが w の時に発生したパケット棄却をタイムアウトを待つまで検出できない確率である。これより、パケット棄却率 p が与えられた時の、定常状態における TCP のウィンドウサイズ w は以下の式で与えられる。

$$w = \lambda_T\, r \qquad (3)$$

- **モデル A'**

  パケット棄却率が非常に小さい ($p << 1$) 場合、式 (3) は以下のように近似できる [4]。

$$w \simeq \sqrt{\frac{3}{2bp}} \qquad (4)$$

- **モデル B**

  文献 [8] では、ECN (Explicit Congestion Notification) を用いた輻輳制御機構の解析が行われている。ECN とは、ルータが送信側ホストに対して明示的に輻輳の発生を通知する機構である。ルータにおいて輻輳が発生した場合に、ルータに到着するパケットの ECN ビットを設定し、送信側ホストに輻輳の発生を通知する。文献 [8] では、ある送信側ホストが受信する ACK パケットは、確率 p_E で ECN ビットが設定されていると仮定し、この時のウィンドウサイズの変化式を導出している。ただし、実際の TCP とは異なり、ACK の ECN ビットが設定されていればウィンドウサイズを $I(w)$ だけ増加し、輻輳通知ビットが設定されていなければウィンドウサイズを $D(w)$ だけ減少するというモデルを解析している。この時、ACK パケット受信ごとのウィンドウサイズの平均的な変化を考えれば、これは次式で与えられる。

$$w \leftarrow w + (1-p_E)\, I(w) - p_E\, D(w) \qquad (5)$$

  これは TCP の輻輳回避フェーズに以下のように当てはめることができる。ECN ビットが設定されていない ACK パケットは、シーケンス番号の異なる ACK パケット (パケット棄却が発生していない) に相当する。また、ECN ビットが設定された ACK パケットは、シーケンス番号の同じ ACK パケット (パケット棄却が発生した) に相当する。また、タイムアウトを考慮した式を導出すると、以下の関係が成立する。

$$w \leftarrow w + (1-p)\frac{1}{w} - p\,(1-\hat{Q}(w))\frac{w}{2} - p\,\hat{Q}(w)\,(w-1) \qquad (6)$$

- **モデル C**

  文献 [5] では、輻輳回避フェーズにおける、あるパケット棄却の発生から次のパケット棄却の発生までを一つのスロットと考え、スロット間でのウィンドウサイズの変化式を導出している。ただし、Drop-Tail ルータではなく、パケットを確率的に棄却する RED ルータを対象としている。Drop-Tail ルータの場合には、文献 [5] で導出されている式を、以下のように変更すればよい。

  文献 [5] では、k 番目の離散スロットにおいて RED ルータを棄却されずに連続して通過するパケット数の期待値 $\overline{X}(k)$ を導出している。

$$\overline{X}(k) = \frac{1/p_b(k)+1}{2}$$

  ここで $p_b(k)$ は、k 番目の離散スロットにおいて、RED ルータが平均バッファ内パケット数から計算する、RED のパケット棄却率である。Drop-Tail ルータのパケット棄却率を p とすれば、$\overline{X}(k)$ は以下の式で与えられる。

$$\overline{X}(k) = \sum_{n=1}^{\infty} n(1-p)^{n-1}p = \frac{1}{p}$$

  これにより、ネットワークにおけるパケット棄却率 p が与えられた時の、パケット棄却発生直後のウィンドウサイズ w_M は以下の式で与えられる。

$$w_M \leftarrow \frac{1}{4}\left\{-1+\sqrt{(1-2w_M)^2+\frac{8}{p}}\right\} \qquad (7)$$

上記の 4 つのモデルのうち、モデル A およびモデル A' は、定常状態におけるウィンドウサイズに着目している。このため、TCP の過渡特性解析には適さないと考えられる。一方、モデル B およびモデル C は、輻輳回避フェーズにおけるウィンドウサイズの動的な変化をモデル化している。このため、モデル A よりも、モデル B またはモデル C のほうが、TCP の過渡特性解析には適していると考えられる。本稿では、式 (7) を導出する際に、あるパケット棄却の発生から次のパケット棄却の発生ま

で、パケット棄却率が一定であると仮定している。このため、モデル C では、ウィンドウサイズの増加にともなう、パケット棄却率の上昇がモデル化できていない。

3 シミュレーションとの比較

以下では、シミュレーション結果と比較することにより、2 章で説明した解析モデルのうち、どのモデルが TCP の過渡特性解析に適しているかを検討する。なお、本稿でのシミュレーションはすべて ns2 [9] を用いて行った。

3.1 シミュレーションモデル

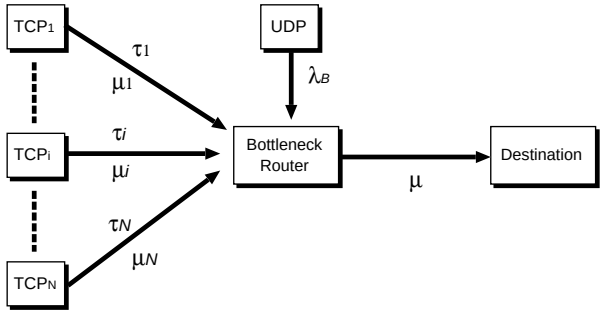

図 3: シミュレーションモデル

シミュレーションモデルを図 3 に示す。ここでは 10 本の TCP コネクションがボトルネックリンクを共有している。i 番目の送信側ホストからルータまでの伝播遅延時間を $5+i$ [ms]、リンク容量を $5+0.5i$ [パケット/ms] とした。また、ボトルネックリンク上のバックグラウンドトラヒックを、平均到着率が $\lambda_B = 2$ [パケット/ms] のポアソン過程に従う UDP パケットによってモデル化した。特に断りのない限り、シミュレーションでは以下のようなパラメータを使用した。TCP のコネクション数 $N = 10$ [本]、TCP のパケット長 1,000 [バイト]、UDP のパケット長 1,000 [バイト]、ルータのバッファサイズ $m = 50$ [パケット]、ボトルネックリンクの容量 $\mu = 5$ [パケット/ms]、ボトルネックリンクの伝播遅延時間 $\tau = 5$ [ms]。なお、1 [パケット/ms] は約 8 Mbit/s に相当する。シミュレーション時間は 30 秒である。

3.2 TCP から見たネットワークモデル

図 4 に、シミュレーションにおいて測定した、ボトルネックルータの利用率 ρ とパケット棄却率 p の関係を示している。この図では、シミュレーションにおいて測定した、10 [ms] ごとのボトルネックルータの瞬間的な利用率と、その時のパケット棄却率の関係もあわせて示している。これは、10 [ms] の間にボトルネックルータに到着したパケット数と、10 [ms] の間にボトルネックルータのバッファで棄却されたパケット数から計算している。図中には、$M/M/1/m$ 待ち行列を用いて計算したパケット棄却率を示している。参考のため、$M/M/1$ のパケット棄却率もあわせて示している。これより、$M/M/1/m$ を用いることにより、TCP から見たネットワークをうまくモデル化できていることがわかる。ただし、シミュレーション結果の点は、広範囲にわたって存在している。つまり、ボトルネックルータの利用率が一定の場合、パケット棄却率は実際には大きく変動することがわかる。

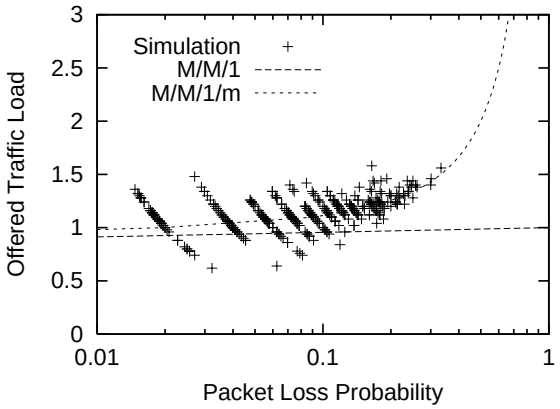

図 4: ボトルネックルータ利用率とパケット棄却率の関係

3.3 TCP の輻輳制御のモデル

以下では、TCP の輻輳制御のモデル A 、A' 、B 、C が、どの程度正確にウィンドウサイズとパケット棄却率の関係を示しているかを、シミュレーション結果と比較することにより検討する。図 5 は、モデル A 、A' 、B 、C それぞれについて、定常状態におけるウィンドウサイズとパケット棄却率の関係を示している。定常状態におけるウィンドウサイズとパケット棄却率の関係は、それぞれのモデルの式を w について解くことで求めている。ただし、モデル A ではタイムアウトが発生しない ($\hat{Q}(w) = 0$) と仮定した場合の結果を示している。また、モデル C におけるウィンドウサイズは、パケット棄却が発生した直後のウィンドウサイズである。他のモデルとの比較のため、ここでは文献 [5] で説明されている方法を用いて、ウィンドウサイズの平均値を計算した。図中には、シミュレーションにおける、1 [s] ごとに測定した各 TCP コネクションのウィンドウサイズの平均値と、ボトルネックルータのパケット棄却率に対応する点をあわせて示している。この図から、パケット棄却率がおよそ 0.02 以下の時には、シミュレーション結果の点が、ほぼモデル A 、A' 、B の曲線付近に集中している。一方、パケット棄却率がおよそ 0.03 以上のときには、シミュレーション結果の点が、ほぼモデル B とモデル C の曲線の間にある。

4 TCP の過渡特性解析

最後に、これまでに説明した解析モデルを用いて、TCP の輻輳回避フェーズにおける過渡特性 (ウィンドウサイズが初期状態から定常状態までどのように変化するか) を解析する。TCP のウィンドウサイズは、ネットワークでのパケット棄却の有無によって変化するが、パケット棄却は確率的に発生する。このため、ウィンドウサイズもまた確率的に変動することになる。以下では、確率的に変動するウィンドウサイズの平均を調べることにより、TCP の輻輳回避フェーズにおける過渡特性を解析する。

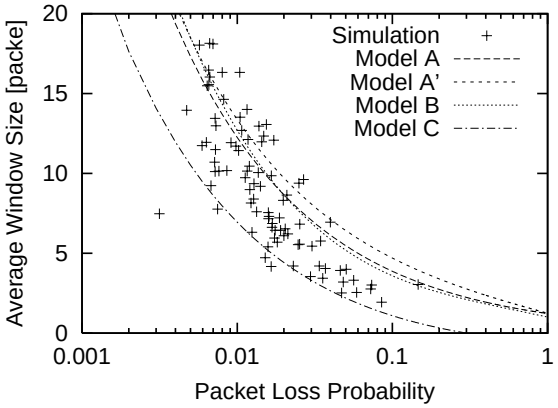

図 5: ウィンドウサイズとパケット棄却率の関係

まず、TCP の輻輳制御モデルとネットワークの待ち行列モデルが、それぞれ単位時間ごとに状態が変化する、離散時間モデルで考える。この時、ある時刻 (スロット) におけるネットワークの状態は、TCP のウィンドウサイズ $w(k)$ およびネットワークにおけるパケット棄却率 $p(k)$ によってあらわすことができる。つまり、ウィンドウサイズおよびパケット棄却率の初期値を与えれば、各スロットごとにウィンドウサイズがどのように変化するかを調べることができる。以下では、TCP の輻輳制御機構としてモデル B、ネットワークのモデルとして $M/M/1/m$ を用いた場合の解析結果を示す。文献 [5] の解析手法を用いれば、より厳密な安定性や過渡特性の解析が可能であるが、ここでは数値計算を使った比較的簡単な解析手法を用いる。2.3 節で述べたように、モデル B は、ACK パケットを受信するごとのウィンドウサイズの変化を表しているため、以下では送信側ホストへの ACK の到着間隔を 1 スロットと考える。簡単のためすべての TCP コネクションの伝播遅延時間は等しく、なおかつウィンドウサイズは同期して変化すると仮定する。

送信側ホストおよびボトルネックルータ間の伝搬遅延時間を 0 と仮定し、ボトルネックリンクの伝搬遅延時間を τ とする。送信側ホストは、1 ラウンドトリップ時間中に $w(k)$ 個のパケットを送出するため、1 ラウンドトリップ時間は $w(k)$ スロットに相当する。一方、ボトルネックルータにおけるパケット棄却率の変化が、送信側ホストに影響を与えるまで 1 ラウンドトリップ時間を要する。従って、k 番目のスロットにおけるウィンドウサイズ $w(k)$ は、直前のウィンドウサイズ $w(k-1)$ およびラウンドトリップ時間前のパケット棄却率 $p(k-w(k-1))$ によって決まる。このことと、2 章のモデル B および $M/M/1/m$ 待ち行列モデルから、以下のような状態遷移方程式が得られる。

$$
\begin{aligned}
w(k) &= w(k-1) + \frac{1-p(k-w(k-1))}{w(k-1)} \\
&\quad - \frac{p(k-w(k-1))\,w(k-1)}{2} \\
p(k) &= \frac{(1-\rho(k))\,\rho(k)^m}{1-\rho(k)^{m+1}}
\end{aligned}
$$

ただし、$\rho(k)$ および $r(k)$ は、それぞれ k 番目のスロットにおけるボトルネックルータの利用率および TCP のラウンドトリップ時間であり、以下のように定義される。

$$
\begin{aligned}
\rho(k) &= \frac{1}{\mu}\left(\frac{N\,w(k)}{r(k)} + \lambda_B\right) \\
r(k) &= 2\tau + \frac{1-\rho^m}{\mu(1-\rho^{m+1})}\left(\frac{1}{1-\rho} + \frac{m\rho^m}{1-\rho^m}\right) \\
&\simeq 2\tau + \frac{m}{\mu}
\end{aligned}
$$

ここでは簡単のため、タイムアウトによる影響は無視できると仮定している (式 (6) の右辺第 3 項を省いている)。また、TCP は利用率が 1.0 前後で動作することから、ラウンドトリップ時間 $r(k)$ を (伝搬遅延時間 + バッファサイズ / ボトルネックルータのパケット処理率) で近似している。なお、$w(k)$ は実際のウィンドウサイズの値そのものではなく、ウィンドウサイズの平均値である。これらの式を用いて、$w(k)$ および $p(k)$ を逐次的に計算することにより、ウィンドウサイズおよびパケット棄却率の、過渡的な変化を解析することができる。

次に、いくつかの数値例を用いて、バックグラウンドトラヒックのパケット到着率 λ_B やボトルネックリンクの伝搬遅延時間 τ が、TCP の過渡特性にどのような影響を与えるかを明らかにする。以下の数値例では、特に断りのない限り、ウィンドウサイズの初期値 1 [パケット]、パケット棄却率の初期値 0、TCP のコネクション数 $N = 10$、ボトルネックリンクの容量 $\mu = 5$ [パケット/ms]、ボトルネックリンクの伝搬遅延時間 $\tau = 15$ [ms]、ルータのバッファサイズ $m = 50$ [パケット] としている。

図 6 に、バックグラウンドトラヒックのパケット到着率を λ_B = 0、2.0、4.5 [パケット/ms] と変化させた時の、TCP の輻輳回避フェーズにおけるウィンドウサイズの時間的変動を示す。この図から、バックグラウンドトラヒック量が増加するにつれ、定常状態におけるウィンドウサイズが小さくなっていることがわかる (つまり、TCP のスループットが低下している) ことがわかる。ただし、バックグラウンドトラヒックの到着率にかかわらず、ウィンドウサイズが増加する速度はほぼ一定である。これは、TCP の輻輳回避フェーズでは、TCP のスループットにかかわらず 1 ラウンドトリップ時間に 1 パケットぶんのウィンドウサイズを増加させるためである。

図 7 に、ボトルネックリンクの伝搬遅延時間を τ = 5、15、25 [ms] と変化させた時の、TCP の輻輳回避フェーズにおけるウィンドウサイズの時間的変動を示す。ボトルネックリンクの伝搬遅延時間が増加するにつれ、ウィンドウサイズが大きくなっていることがわかる。これは TCP コネクションあたりの帯域遅延積が大きくなっているためである。さらに、ボトルネックリンクの伝搬遅延時間が大きくなるにつれ、(1) ウィンドウサイズの増加速度が遅くなっている、(2) ウィンドウサイズが収束するまでの時間が短かくなっていることがわかる。一般に、フィードバックシステムにおいてフィードバック遅延が大きくなると、それにともないシステムの過渡特性

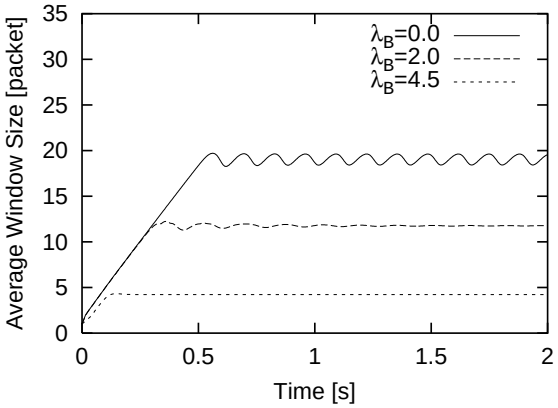

図 6: 輻輳回避フェーズにおける TCP の過渡特性 (バックグラウンドトラヒックの到着率 λ_B が変化した場合)

が低下し、より動作が不安定となる。しかし (2) は、これとはまったく逆の結果となっている。特に、伝搬遅延時間が $\tau = 5$ [ms」 の場合、ウィンドウサイズは 1.5 [s] 以上の間振動している。これは、TCP の輻輳回避フェーズでは、フィードバックゲインがラウンドトリップ時間に応じて変化するためである。つまり、TCP の輻輳回避フェーズでは、1 ラウンドトリップ時間ごとに 1 パケットだけウィンドウサイズを増加させる。このため、伝搬遅延時間を増加させることは、フィードバックゲインを小さくする (制御量を小さくする) ことを意味するからである。

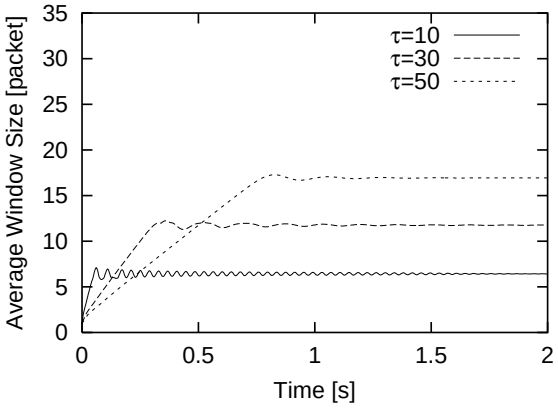

図 7: 輻輳回避フェーズにおける TCP の過渡特性 (ボトルネックリンクの伝搬遅延時間 τ が変化した場合)

ボトルネックルータのバッファサイズが変化した場合の数値例を図 8 に示す。ここでは、ボトルネックルータのバッファサイズを m = 10、50、100 [パケット] と変化させている。この図より、ボトルネックルータのバッファサイズを大きくすると、帯域遅延積の増加により、TCP のウィンドウサイズが増加することがわかる。これに加えて、バッファサイズが大きくなると、ウィンドウサイズが大きく変動する (振幅が大きくなる) ことがわかる。また、バッファサイズが小さくなると、ウィンドウサイズの増加速度はわずかに遅くなることがわかる。

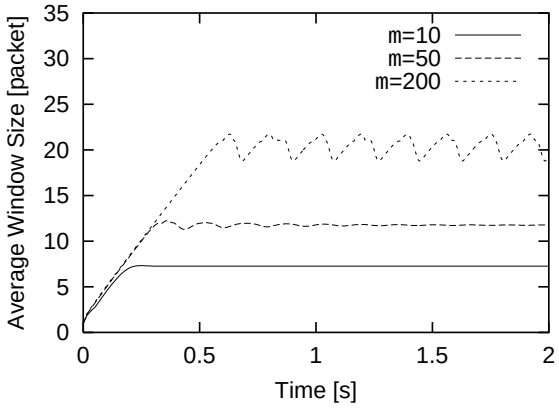

図 8: 輻輳回避フェーズにおける TCP の過渡特性 (ボトルネックルータのバッファサイズ τ が変化した場合)

最後に、解析結果の妥当性を検証するために、シミュレーション結果との比較を行う。図 9 および図 10 に、それぞれバックグラウンドトラヒックのパケット到着率 λ_B およびボトルネックルータのバッファサイズ m を変化させた場合のシミュレーション結果を示す。これらの図は、それぞれ解析結果の図 6 および図 8 と同じパラメータを用いている。それぞれ 20 回のシミュレーションを行い、合計 200 本の TCP コネクションのウィンドウサイズを測定し、その平均値の時間的な変化を示している。TCP の輻輳回避フェーズの過渡特性のみに着目するため、TCP が常に輻輳回避フェーズで動作するよう $ssthresh$ の値を 0 に設定した。これより、解析結果とシミュレーション結果がほぼ一致していることがわかる。ただし、解析結果とシミュレーション結果では、ウィンドウサイズが平均値まで増加した後の変化が異なっている。これは、シミュレーションではすべての TCP コネクションが同期しているため、ウィンドウサイズの変化がより大きくなっているためと思われる。

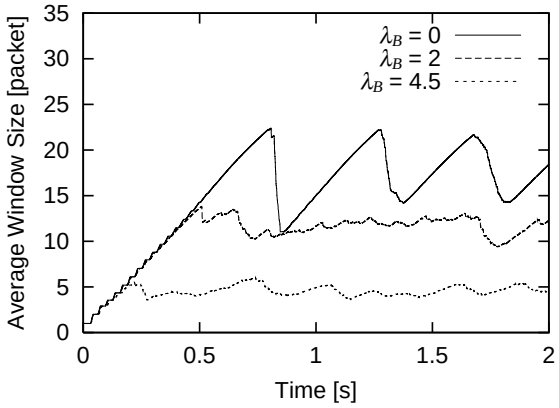

図 9: 輻輳回避フェーズにおける TCP の過渡特性 (シミュレーション結果、バックグラウンドトラヒックの到着率 λ_B が変化した場合)

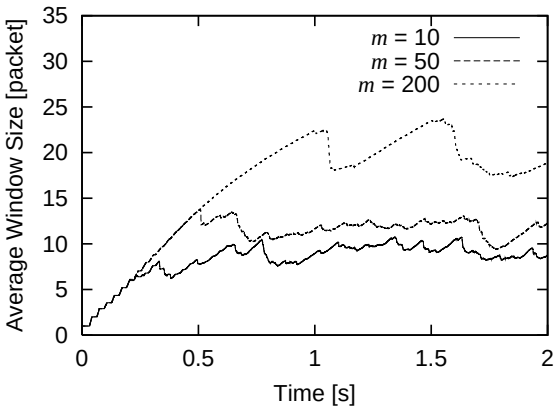

図 10: 輻輳回避フェーズにおける TCP の過渡特性 (シミュレーション結果、ボトルネックルータのバッファサイズ r が変化した場合)

5 まとめと今後の課題

本稿では、TCP の輻輳制御機構とネットワークをあわせて、フィードバックシステムとしてモデル化し、TCP の過渡特性を解析した。TCP をネットワークでのパケット棄却率を入力とし、ウィンドウサイズを出力とするシステムとしてモデル化した。一方、ネットワークを TCP のウィンドウサイズを入力とし、パケット棄却率を出力とするシステムとしてモデル化した。ネットワークは、バックグラウンドトラヒックの影響を考慮した、$M/M/1/m$ 待ち行列としてモデル化した。さらに過渡特性解析を行い、バックグラウンドトラヒックの量や、TCP のコネクション数などによって、TCP の過渡特性がどのように変化するかを定量的に明らかにした。

その結果、TCP の輻輳回避フェーズにおける過渡特性は、TCP コネクションの伝搬遅延時間に大きく依存するが、バックグラウンドトラヒックの量やボトルネックルータのバッファサイズには、ほとんど影響を受けないことが分かった。このことは、TCP の輻輳回避フェーズが、1 ラウンドトリップ時間ごとにウィンドウサイズを 1 パケットだけ増加させることから定性的にも説明できる。ただし本解析では、TCP の過渡状態におけるウィンドウサイズの変動を定量的に示した。さらに、バックグラウンドトラヒックの量が少なくなる、伝搬遅延時間が小さくなる、もしくはルータのバッファサイズが大きくなるについれて、TCP の輻輳回避フェーズの動作がより不安定になることが明らかになった。

本稿では、導出した状態遷移方程式を逐次的に計算することにより、TCP の過渡特性を解析した。しかし、文献 [5] で示されている方法を用いれば、より厳密な安定性や過渡特性の解析が可能となる。そこで今後は、本稿で提案した解析モデルを用いて、より詳細な性能解析を行ってゆく予定である。

謝辞

本研究の一部は、日本学術振興会未来開拓学術研究推進事業における研究プロジェクト「高度マルチメディア応用システム構築のための先進的ネットワークアーキテクチャの研究」 (JSPS-RFTF97R16301) によっている。ここに記して謝意を表す。

参考文献

[1] V. Jacobson, “Congestion avdoidance and control,” in *Proceedings of ACM SIGCOMM ’88*, pp. 314–329, August 1988.

[2] W. R. Stevens, *TCP/IP Illustrated, Volume 1: The Protocols*. New York: Addison-Wesley, 1994.

[3] J. Padhye, V. Firoiu, D. F. Towsley, and J. F. Kurose, “Modeling TCP Reno performance: A simple model and its empirical validation,” *IEEE/ACM Transactions on Networking*, vol. 8, pp. 133–145, April 2000.

[4] D. T. Jitendra Padhye, Victor Firoiu and J. Kurose, “Modeling TCP throughput: A simple model and its empirical validation,” in *Proceedings of IEEE SIGCOMM ’98*, September 1998.

[5] H. Ohsaki, Y. Mera, M. Murata, and H. Miyahara, “Steady state analysis of the RED gateway: stability, transient behavior, and parameter setting,” submitted to *Internet Performance and Control of Network Systems II (IT301)*, Feb. 2001.

[6] C. Casetti and M. Meo, “A new approach to model the stationary behavior of tcp connections,” in *Proceedings of IEEE INFOCOM 2000*, March 2000.

[7] V. Firoiu and M. Borden, “A study of active queue management for congestion control,” in *Proceedings of IEEE INFOCOM 2000*, 2000. available at `http://www.ieee-infocom.org/2000/papers/405.pdf`.

[8] T. J. Ott, “ECN protocols and the TCP paradigm,” in *Proceedings of IEEE INFOCOM 2000*, pp. 100–109, March 2000.

[9] “UCB/LBNL/VINT network simulator - ns (version 2).” available at `http://www-mash.cs.berkeley.edu/ns/`.