伝搬遅延時間の異なる複数の TCP コネクションを考慮したウィンドウ型フロー制御方式の特性解析

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) +81-6-6850-6616
(Fax) +81-6-6850-6589
(E-mail) takagaki@ics.es.osaka-u.ac.jp

あらまし　パケット交換網において、ベストエフォート型のトラヒックを効率的に網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。現在広く普及しているインターネットでは、ウィンドウ型のフロー制御方式が TCP (Transmission Control Protocol) で用いられている。近年、一般に広く用いられている TCP Reno よりも、高い性能を示す TCP Vegas が注目されている。我々はこれまでの研究で、TCP Vegas にもとづくウィンドウ型フロー制御方式の安定性解析を行なった。本稿では、ここでの解析結果を利用し、各 TCP コネクションの伝搬遅延時間がさまざまに変化した場合に、ウィンドウ型フロー制御方式の特性がどのように変化するかを明らかにする。また、さまざまなネットワークパラメータが、ウィンドウ型フロー制御方式の安定性だけでなく、過渡特性にもどのような影響を与えるかを明らかにする。
和文キーワード　インターネット、ウィンドウ型フロー制御、TCP Vegas、制御理論、特性解析

Analysis of a Window-based Flow Control Mechanism for TCP Connections with Different Propagation Delays

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) takagaki@ics.es.osaka-u.ac.jp

Abstract　A feedback-based congestion control mechanism is essential to realize an efficient best-effort service in high-speed networks. A window-based flow control mechanism called TCP (Transmission Control Protocol), a sort of feedback-based congestion control mechanism, has been widely used in the current Internet. Recently-proposed TCP Vegas is another version of TCP mechanism, and achieves much better performance than the current TCP Reno. In our previous works, we have analyzed the stability of the window-based flow control mechanism based on the TCP Vegas. In this paper, using our analytic results, we invesitigate how dynamics of the window-based flow control mechanism is affected by the difference in propagation delays of TCP connections. We also investigate the effect of various system parameters on transient performance of the window-based flow control mechanism as well as the effect on its stability.
英文 key words　Internet, Window-Based Flow Control Mechanism, TCP Vegas, Control Theory, Characteristic Analysis

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) において用いられている。TCP は、ネットワーク内でパケットが失われた場合に、失わ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能と、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能を持っている。

最近、現在広く使用されている TCP Reno よりも良い性能を示す TCP の実装例として、TCP Vegas が提案された [1] 。TCP Vegas では、TCP Reno に比べて以下の点が改良されている。(1) 新しいタイムアウト機構、(2) ネットワーク内でのバッファの占有量を制御する輻輳回避機構、(3) 改良されたスロースタート機構の 3 点である。特に、(2) の輻輳回避機構では、パケットを送出してからそのパケットに対応した ACK (Acknowledgment) パケットを受信するまでの時間、すなわち RTT (ラウンドトリップ時間) を測定し、これをネットワークからのフィードバック情報としてウィンドウサイズの調整を行う。このため、TCP Reno のように、ネットワーク内で意図的にパケット棄却を発生させる必要がなく、TCP Reno に比べてスループットが向上することが報告されている [2]。

我々はこれまで、文献 [3, 4] において、TCP Vegas の輻輳制御機構にもとづくウィンドウ型フロー制御方式の特性解析を行った。ただし、ここでは RTT ごとのウィンドウサイズの変化量を、TCP Vegas のように 1 パケットと固定するのではなく、制御パラメータとして自由に指定できるようなモデルを扱った。まず、文献 [3] において、すべての TCP コネクションの伝搬遅延時間が等しい場合の、ウィンドウ型フロー制御方式の安定性解析および過渡特性解析を行った。さらに、文献 [4] では、すべての TCP コネクションの伝搬遅延時間が等しいという仮定を取り除き、伝搬遅延時間が異なる場合について安定性解析を行った。また数値例により、TCP コネクションの伝搬遅延時間の比が 1:2 の場合に、制御パラメータがシステムの安定性にどのような影響を与えるかを明らかにした。本稿では、文献 [4] と同じく、TCP コネクションの伝搬遅延時間が異なる場合を解析する。ただし本稿では、文献 [4] で用いたものよりもモデル化の誤差が小さい解析モデルを用いる。また、TCP コネクションの伝搬遅延時間の比が 1:2 以外の場合について検討を行う。さらに、ルータの処理能力や伝搬遅延時間といったさまざまなシステムパラメータが、ウィンドウ型フロー制御方式の安定性だけでなく、過渡特性にもどのような影響を与えるかを明らかにする。

以下、2 章において、本稿で扱うウィンドウ型フロー制御方式の解析モデルを説明する。3 章では、制御理論を適用することにより、ウィンドウ型フロー制御方式の安定性解析を行う。なお、ここでの結果は、文献 [4] とほぼ同じであるため、解析の概要を説明するにとどめる。4 章では、システムを安定化できる制御パラメータの範囲が、ルータの処理能力や伝搬遅延時間などのシステムパラメータによってどのように変化するかを数値例を用いて示す。また、3 章での結果を利用して、最適な過渡特性を得るためには、どのような制御パラメータを選べばよいかを明らかにする。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

本稿で用いる解析モデルを図 1 に示す。複数の送信側ホストが、単一のルータを介してそれぞれ対応する受信側ホストに接続されている。ここでは、伝搬遅延時間が異なる M 種類の TCP コネクションを考え、往復伝搬遅延時間が τ_m $(1 \leq m \leq M)$ であるコネクションの集合をグループ m と呼ぶ。また、グループ m に属する TCP のコネクション数を N_m とする。Δ_m $(1 \leq m \leq M)$ を、往復伝搬遅延時間 τ_m の比をあらわす既約な正整数とする。この時、

$$\frac{\tau_1}{\Delta_1} = \frac{\tau_2}{\Delta_2} = \cdots \frac{\tau_{M-1}}{\Delta_{M-1}} = \frac{\tau_M}{\Delta_M}$$

という関係が成立する。ここでバッファにおけるパケットの待ち時間が無視できるとすれば、RTT の比もまた Δ_m で与えられる。TCP Vegas では、RTT を単位としてウィンドウサイズを変化させるため、$\tau(\equiv \tau_m/\Delta_m)$ を単位として時間をスロットに分割し、ネットワーク全体を離散時間モデルとして考えることができる。この場合、グループ m の TCP コネクションは、Δ_m スロットごとにウィンドウサイズを変更することになる。

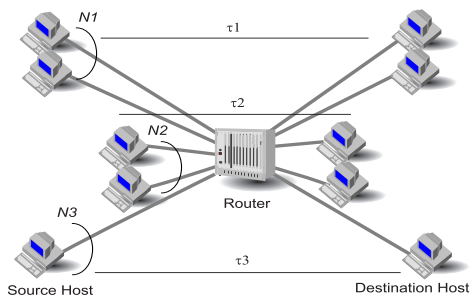


図 1: 解析モデル ($M = 3$ の場合)

スロット k における、グループ m に属する n $(1 \leq n \leq N_m)$ 番目の送信側ホストのウィンドウサイズを $w_{m,n}(k)$ とする。さらに、スロット k においてルータのバッファ内に存在するパケット数を $q(k)$ 、ルータのバッファ容量を L とする。また、ルータの処理能力を B とする。この時、$k + \Delta_m$ 番目のスロットにおける送信側ホストのウィンドウサイズおよび $k + 1$ 番目のスロットにおけるルータのバッファ内パケット数は、それぞれ以下の式で与えられる [4] 。

$$w_{m,n}(k + \Delta_m) = \max(w_{m,n}(k) + \delta_{m,n}(\gamma_{m,n} - d_{m,n}(k)), 0) \quad (1)$$

$$q(k+1) = \min\left[\max\left\{\sum_{m=1}^{M}\sum_{n=1}^{N_m}(w_{m,n}(k)\right.\right.$$

$$-\frac{w_{m,n}(k)B\Delta_m\tau}{\sum_{m=1}^{M}\sum_{n=1}^{N_m}w_{m,n}(k)}\right),0\Bigg\},L\Bigg] \quad (2)$$

ここで、

$$d_{m,n}(k) = \left(\frac{w_{m,n}(k)}{\tau_m}-\frac{w_{m,n}(k)}{r_m(k)}\right)\times\tau_m \quad (3)$$

$$r_m(k) = \tau_m+\frac{q(k)}{B} \quad (4)$$

である。なお、γ_m および $\delta_{m,n}$ は、ともに送信側ホストの制御パラメータである。γ_m はグループ m に属する送信側ホストが、ネットワーク中にどれだけ余分なパケットを送出するかを決定する制御パラメータである。$\delta_{m,n}$ はグループ m の n 番目の送信側ホストが、RTT ごとにどれだけウィンドウサイズを変化させるかを決定する制御パラメータである。式 (4) は、グループ m の送信側ホストがスロット k において観測する RTT を表している。なお、文献 [4] で用いた解析モデルでは、Δ_m の値が大きい時にルータのバッファ内パケット数 $q(k)$ がずれるという問題があったため、式 (2) は誤差が小さくなるよう変更している。

3 特性解析

本章では、式 (1)–(4) で与えられるウィンドウ型フロー制御方式について、定常状態における平衡点の近傍での安定性および過渡特性を解析する。ただし、ここでの結果は文献 [4] と基本的に同じであるため、解析の概要を説明するにとどめる。解析の詳細については、文献 [4] を参照されたい。

以下では、伝搬遅延時間が等しい送信側ホストの、ウィンドウサイズの初期値や制御パラメータはすべて等しいと仮定する。この場合、送信側ホストの制御パラメータを $\delta_m \equiv \delta_{m,n}(1 \le m \le M, 1 \le n \le N_m)$ などと表記する。また、全ての送信側ホストは式 (1) に従って RTT ごとにウィンドウサイズを変更する。この時、スロット $k+1$ におけるバッファ内パケット数 $q(k+1)$ は以下の式で与えられる。

$$q(k+1) = \min\left[\max\left\{\sum_{m=1}^{M}N_m\left(w_m(k)-\frac{w_m(k)\,B\,\Delta_m\,\tau}{\sum_{m=1}^{M}N_m w_m(k)}\right),0\right\},L\right] \quad (5)$$

ただし、$w_m(k) \equiv w_{m,n}(k)\ (1 \le n \le N_m)$ とする。

$w_m(k)$、$q(k)$、$d_m(k)$ の平衡点をそれぞれ w_m^*、q^*、d_m^* とする。式 (1)–(4) において、$w_m(k+\Delta_m)=w_m(k)$ などとおくことにより、w_m^*、q^*、d_m^* は容易に求めることができる。ここで、スロット k におけるシステムの状態と平衡点の差を

$$\mathbf{x}(k) \equiv \begin{bmatrix} w_1(k)-w_1^* \\ \vdots \\ w_M(k)-w_M^* \\ q(k)-q^* \end{bmatrix}$$

と定義する。

式 (1) で与えられる $w_m(k)$ は非線形であるため、これを平衡点の近傍において線形化し、さらに Δ_L を $\Delta_1\cdots\Delta_M$ の最小公倍数とすれば、$\mathbf{x}(k+\Delta_L)$ は以下のように書くことができる [5]。

$$\mathbf{x}(k+\Delta_L)=\mathbf{A}\,\mathbf{x}(k) \quad (6)$$

ここで $\mathbf{A}$ はシステムの状態遷移行列であり、この行列の固有値の大小によってシステムの安定性や過渡特性が決定される。つまり、式 (6) で与えられる離散時間システムにおいて、平衡点が局所漸近安定となるためには、$D(s)\equiv|s\mathbf{I}-\mathbf{A}|=0$ によって得られる、特性方程式のすべての解 $s_i(1\le i\le M+1)$ に関して $|s_i|<1$ であればよい [6]。また、システムの過渡特性は $|s_i|$ の大小によって決定される。つまり、$|s_i|$ が 0 に近いほど過渡状態における収束の速度が速くなる。一般に、$|s_i|$ の最大値、すなわち

$$|s| \equiv \max_{1\le i\le M+1}(|s_i|) \quad (7)$$

によってシステム全体の過渡特性が大きく左右されることが知られている [6]。

4 数値例

本章では、$M=2$ の場合 、すなわち伝搬遅延時間の異なる 2 種類の TCP コネクションが存在する場合を考える。この時、各 TCP コネクションの伝搬遅延時間の違いによって、システム全体の安定性や過渡特性がどのように変化するかを示す。

4.1 安定性に関する考察

本稿で対象としているウィンドウ型フロー制御方式では、制御パラメータの設定によっては、ネットワーク全体の動作が不安定になる [4]。ネットワークが不安定になると、定常状態において送信側ホストのウィンドウサイズやルータのバッファ内パケット数が一定値に収束しなくなる。このため、ネットワークが安定となるように、送信側ホストの制御パラメータを適切に選ぶことが重要となる。文献 [4] において、2 種類の TCP コネクションの伝搬遅延時間の比が 1:2 の場合、つまり $\Delta_1=1$ および $\Delta_2=2$ の場合に、制御パラメータをどのように選べばネットワークを安定化できるかを示した。その結果、ウィンドウサイズの変化量を決定する制御パラメータ δ_m を、伝搬遅延時間に比例して設定することにより、ネットワークを安定化できることがわかった。以下では、TCP コネクションの伝搬遅延時間の比が 1:2 以外の場合に、制御パラメータ δ_m をどのように選べばネットワークを安定化できるかを明らかにする。

ルータの処理能力 B、TCP のコネクション数 N_m、送信側ホストの制御パラメータ γ_m、TCP コネクションの伝搬遅延時間 τ_m をさまざまに変化させて、(δ_1,δ_2) に関する安定領域を調べた結果、

$$F_m \equiv \frac{N_m\,\gamma_m}{B\,\tau} \quad (8)$$

の値が同じ場合、まったく同じ安定領域が得られた。そこで以下の数値例では、この F_m の値に着目して議論を行う。

対象とするウィンドウ型フロー制御方式では、F_m の値は以下のような意味を持っている。まず、式 (1) からわかるように、送信側ホストは RTT 中に γ_m 個の余分なパケットをネットワーク内に送出するようウィンドウサイズを調整する。このため、リンク上に存在するパケットを無視できるとすれば、定常状態におけるルータのバッファ内パケット数は、近似的に以下の式で与えられる [4]。

$$q^* \simeq \sum_{m=1}^{M} \sum_{n=1}^{N_m} \gamma_{m,n} \tag{9}$$

ルータのバッファでのパケット待ち時間は、q^*/B で与えられることから、F_m は各 TCP コネクションあたりのバッファでのパケット待ち時間を、伝搬遅延時間で割った値と考えることができる。つまり F_m の値は、送信側ホストで観測される RTT 中に、ルータのバッファでのパケット待ち時間がどの程度含まれているかを示す指標ととらえることができる。例えば、F_m の値が小さい場合は、ルータのバッファでのパケット待ち時間が、TCP コネクションの伝搬遅延時間に比べて小さいということを意味する。逆に、F_m の値が大きい場合には、TCP コネクションの伝搬遅延時間よりも、ルータのバッファでのパケット待ち時間のほうが大きいことを意味する。

まず、制御パラメータ (δ_1,δ_2) の安定領域を、図 2(伝搬遅延時間の比が 1:4 の時) および図 3(伝搬遅延時間の比が 2:3 の時) に示す。これらの図では、(δ_1,δ_2) が境界線の内側 (原点を含む側) にあればシステムが安定となることを意味している。これらの図では、ともに $F_1 = F_2$ という関係を保ちながら、F_m の値を 0.015 ~ 150 と変化させている。

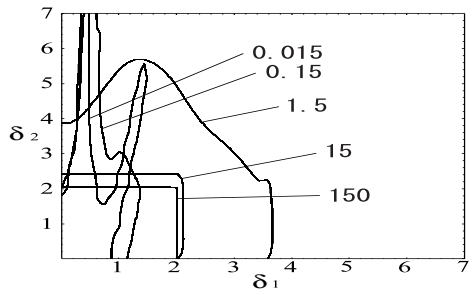


図 2: 伝搬遅延時間の比が 1:4 の時の安定領域 ($F_1 = F_2$ = 0.015 ~ 150, $\Delta_1 = 1$, $\Delta_2 = 4$)

これらの図より、F_m の値が小さい時、システムを安定化できる δ_2 の最大値は、δ_1 の最大値よりも大きくなっていることがわかる (図 2、図 3 ともに $\Delta_1 < \Delta_2$ である)。F_m の値が小さい場合には、送信側ホストで観測される RTT は、ほぼ伝搬遅延時間によって決定される。このため、伝搬遅延時間の比が 1:2 の場合、[4] と同様に、各 TCP コネクションの伝搬遅延時間の違いにより、システムを安定化できる δ_m の最大値が異なっていると考えられる。

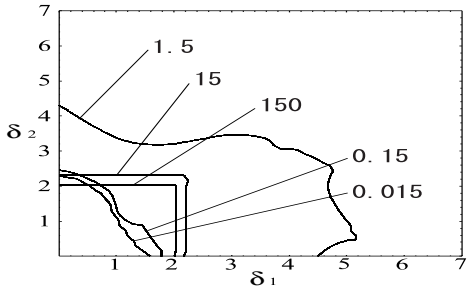


図 3: 伝搬遅延時間の比が 2:3 の時の安定領域 ($F_1 = F_2$ = 0.015 ~ 150, $\Delta_1 = 2$, $\Delta_2 = 3$)

一方、F_m の値が大きい時は、システムを安定化できる δ_m の最大値は、TCP コネクションの伝搬遅延時間の違いによらず、ほぼ同じ値となっている。図 2 および図 3 より、F_m の値が大きい場合の安定条件は、$0 < \delta_1, \delta_2 < 2$ であることがわかる。以上の考察から、F_m の値が大きい (すなわち、ルータでのパケット待ち時間が大きい) 場合には、例えば (δ_1, δ_2) = (1, 1) と設定することにより、TCP コネクションの伝搬遅延時間の違いによらず、常にネットワークを安定化できることがわかる。しかし、F_m の値が小さい (すなわち、ルータでのパケット待ち時間が小さい) 場合には、伝搬遅延時間の違いに応じて、δ_m を慎重に決定しなければならない。

4.2 過渡特性に関する考察

ウィンドウ型フロー制御方式が効率的に動作するためには、ネットワーク全体の安定性だけでなく、過渡特性をも考慮して制御パラメータを決定する必要がある。そこで以下では、ネットワーク全体の安定性を保ちながら過渡特性を最適化するために、制御パラメータをどのように設定すればよいかについて検討を行う。具体的には、送信側ホストのウィンドウサイズやルータのバッファ内パケット数が、平衡点に到達するまでの時間を最小化するために、制御パラメータ δ_m や γ_m をどのように選べばよいかを検討する。

3 章で説明したように、ネットワーク全体の過渡特性は式 (7) で与えられる $|s|$ の値によって大きく左右される。ただし、離散時間モデルにおける 1 スロットの長さは τ であり、式 (6) では Δ_L スロットごとの状態遷移を考えているため、実際の過渡特性は $|s|$ の値だけでなく、$\Delta_L\tau$ の大きさにも依存する。つまり、$|s|$ の値が同じであっても、$\Delta_L\tau$ の値により実際に平衡点に収束するまでの時間が変化することになる。

図 4 は、TCP コネクションの伝搬遅延時間の比が 1:2 ($\Delta_1 = 1$、$\Delta_2 = 2$) の場合に、δ_1–δ_2 平面上で $|s|$ の値 (0.2 ~ 1.0) に応じて曲線を描いたものである。ここでは、以下ようなパラメータを使用している。ルータの処理能力 $B = 2$ [パケット/ms]、TCP のコネクション数 $N_1 = N_2 = 3$、伝搬遅延時間 $\tau = 1$ [ms]、送信側ホストの制御パラメータ $\gamma_1 = \gamma_2 = 3$ [パケット]。なお、この時の F_m の値は、$F_1 = F_2 = 4.5$ である。この図では、

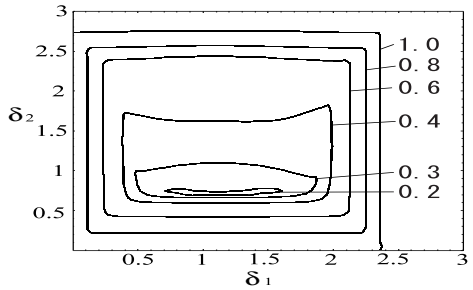
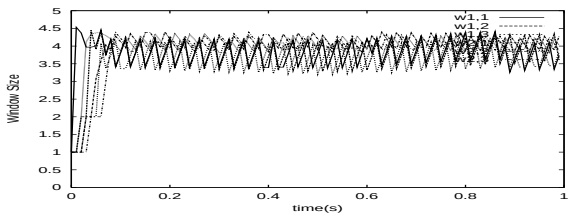
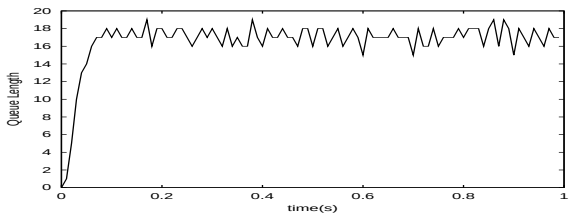


図 4: δ_1–δ_2 平面における $|s|$ の値 ($F_1 = F_2 = 4.5, \Delta_1 = 1, \Delta_2 = 2$)

(δ_1, δ_2) が $|s| = 1.0$ の曲線より外側の場合はシステムが不安定となり、$|s| = 0.2$ の曲線の内側においてシステムの過渡特性が最適となることを意味している。



(a) ウィンドウサイズの変動



(b) バッファ内パケット数の変動

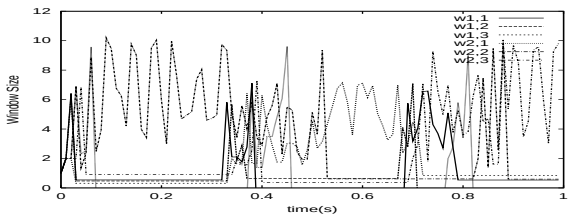
図 5: シミュレーション結果 ($(\delta_1,\ \delta_2)$=(1.3,\ 0.7),\ $F_1 = F_2 = 4.5, \Delta_1 = 1, \Delta_2 = 2$)

$|s|$ の値に応じてネットワーク全体の過渡特性がどのように変化するかを知るために、*ns2* (network simulator 2)[7] を用いてシミュレーションを行った。なお、本稿で解析しているウィンドウ型フロー制御方式では、RTT ごとのウィンドウサイズの変化量を、TCP Vegas のように 1 パケットと固定するのではなく、制御パラメータとして指定できる。このため、*ns2* に含まれる TCP Vegas のコードを一部変更して使用した。シミュレーションで使用した (δ_1, δ_2) の値、その時の $|s|$ の値、および対応するシミュレーション結果を表 1 に示す。

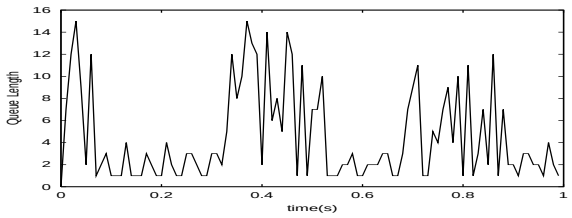
図 5 および図 6 では、図 1 と同じネットワークモデルに対してシミュレーションを行い、各送信側ホストのウィンドウサイズおよびルータのバッファ内パケット数の時間的な変動をプロットしている。$|s| = 0.18$ の場合 (図 5)、ウィンドウサイズおよびルータのバッファ内パケット数が、転送開始から 0.1 [s] 程度で平衡点付近で安定していることがわかる。ただし、定常状態においてもウィンドウサイズやルータのバッファ内パケット数がわずかに振動しているが、これは TCP のタイマの粒度の影響と考えられる [3]。一方、$|s| = 2.56$ の場合 (図 6) では、ウィンドウサイズおよびルータのバッファ内パケット数が激しく振動しており、非常に不安定な動作となっていることがわかる。

表 1: シミュレーションで使用したパラメータ

	(δ_1, δ_2)	$\|s\|$ の値	対応する図
最適な場合	(1.3, 0.7)	0.18	図 5
不安定となる場合	(3.0, 3.0)	2.56	図 6



(a) ウィンドウサイズの変動



(b) バッファ内パケット数の変動

図 6: シミュレーション結果 ($(\delta_1,\ \delta_2)$=(3.0,\ 3.0),\ $F_1 = F_2 = 4.5, \Delta_1 = 1, \Delta_2 = 2$)

次に、F_m の値が小さい場合に、δ_1–δ_2 平面上で $|s|$ の値がどのように変化するかを示す。図 7 では、図 4 と同じパラメータを用いているが、ルータの処理能力 B と伝搬遅延時間 τ を、$B = 200$ [パケット/ms] および $\tau_m = 10$ [ms] とそれぞれ変更している。この場合、F_m の値は、$F_1 = F_2 = 0.0045$ となる。図中では、$|s|$ の値 (0.992 ~ 1.0) に応じて曲線を描いている。この図より、(δ_1, δ_2) をどのように選んでも、$|s|$ の値が 0.99 程度にしか小さくならないことがわかる。

図 7 で、過渡特性を最適化する制御パラメータの値は、(δ_1,δ_2)=(0.6,1.7) 程度である。そこで、(δ_1,δ_2)=(0.6,1.7) と選んだ場合のシミュレーション結果を図 8 に示す。この図より、過渡特性を最適化するように (δ_1,δ_2) を選んだとしても、ネットワークが安定するまでに、非常に時間がかかる (この場合 10 [s] 程度) ことがわかる。つまり、過渡特性を向上させるためには、F_m の値を大きくするべきである。

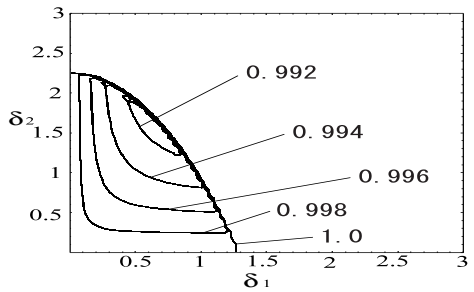
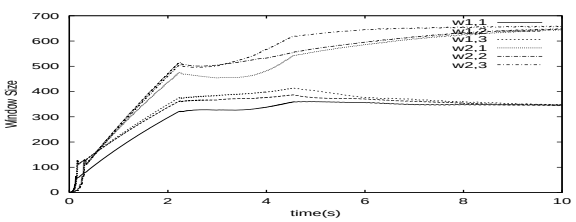


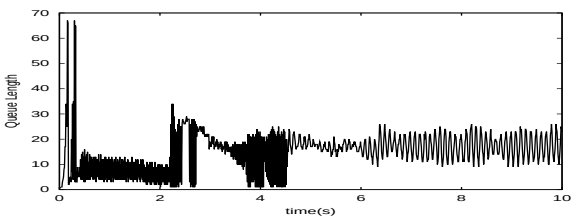
図 7: δ_1–δ_2 平面における $|s|$ の値 ($F_1 = F_2 = 0.0045$, $\Delta_1 = 1, \Delta_2 = 2$)

F_m を決定するパラメータのうち、TCP のコネクション数 N_m、ルータの処理能力 B、伝搬遅延時間 τ は送信側ホストが変更できないが、制御パラメータ γ_m は送信側ホストにおいて自由に設定できる。従って、良好な過渡特性を得るためには、送信側ホストにおいて γ_m をできるだけ大きな値に設定すればよい。しかし、式 (9) から分かるように、ルータのバッファ内パケット数は制御パラメータ γ_m に比例して大きくなる。このため、制御パラメータ γ_m の値は、ルータのバッファでのパケット待ち時間と、ネットワーク全体の過渡特性の間のトレードオフを考慮して決定する必要がある。

4.1 節で述べたように、制御パラメータ γ_m を十分大きな値に設定できれば、ウィンドウサイズの変化量を決める制御パラメータ δ_m を、$0 < \delta_m < 2$ の範囲で設定すれば、ネットワークを安定化できる。ただし、ルータのバッファサイズの制約などから、γ_m をあまり大きな値に設定できない場合には、各 TCP コネクションの伝搬遅延時間の違いに応じて δ_m を慎重に設定しなければならない。



(a) ウィンドウサイズの変動



(b) バッファ内パケット数の変動

図 8: シミュレーション結果 ($(\delta_1,\ \delta_2)$=(0.6, 1.7), $F_1 = F_2 = 0.0045$ $\Delta_1 = 1, \Delta_2 = 2$)

5 まとめと今後の課題

本稿では、TCP Vegas にもとづくウィンドウ型フロー制御方式の特性解析を行なった。特に、各 TCP コネクションの伝搬遅延時間が異なる場合に、ウィンドウ型フロー制御方式の安定性や過渡特性がどのように変化するかを明らかにした。その結果、ルータのバッファにおけるパケット待ち時間と、各 TCP コネクションの伝搬遅延時間の大小関係により、ネットワーク全体の安定性や過渡特性が大きく変化することがわかった。また、安定性および過渡特性を向上するために、送信側ホストの制御パラメータをどのように選べばよいかを明らかにした。具体的には、送信側ホストがネットワーク中にどれだけ余分なパケットを送出するかを決める制御パラメータをできるだけ大きくし、ウィンドウサイズの変化量を決める制御パラメータを 1 前後と設定すればよいことがわかった。今後の課題として、ネットワーク内に複数のルータが存在する場合の特性解析があげられる。

参考文献

[1] L. S. Brakmo, S. W. O'Malley, and L. L. Peterson, "TCP Vegas: New techniques for congestion detection and avoidance," in *Proceedings of ACM SIGCOMM '94*, pp. 24–35, October 1994.

[2] L. S. Brakmo and L. L. Peterson, "TCP Vegas: End to end congestion avoidance on a global Internet," *IEEE Journal on Selected Areas in Communications*, vol. 13, pp. 1465–1480, October 1995.

[3] 大崎 博之, 村田 正幸, 潮 俊光, 宮原 秀夫, "制御理論を用いた TCP Vegas にもとづくウィンドウ型 フロー制御方式の特性解析," 高品質インターネット研究会, Oct. 1998.

[4] K. Takagaki, H. Ohsaki, and M. Murata, "Stability analysis of a window-based flow control mechanism for TCP connections with different propagation delays," in *Proceedings of INET 2000: The Internet Global Summit*, July 2000.

[5] 平井 一正, 池田 雅夫, 非線形制御システムの解析. オーム社, 1986.

[6] R. Isermann, *Digital control systems, Volume 1: fundamentails, deterministic control*. Springer-Verlag Berlin Heidelberg, 1989.

[7] "The network simulator." available at `http://www.isi.edu/nsnam/ns/`.