多段接続されたネットワークにおける TCP Vegas にもとづくウィンドウ型フロー制御方式の解析

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情報数理系専攻
〒 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
〒 567-0047 茨木市美穂ヶ丘 5-1

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

あらまし 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 が提案されている。我々はこれまで、ネットワーク中に単一のボトルネックルータが存在する場合を対象とし、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てきた。本稿では、これらの解析手法を拡張することにより、多段接続されたネットワークにおいて、ネットワーク中に複数のボトルネックルータが存在する場合を対象とした解析を行う。まず、定常状態における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を導出する。この結果を用いて、TCP コネクション間の公平性を実現するために、ウィンドウ型フロー制御方式の制御パラメータを、どのように決定すべきかを明らかにする。さらに、現代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制御パラメータと、ネットワークの安定性および過渡特性の関係を定量的に明らかにする。
和文キーワード ウィンドウ型フロー制御方式、TCP Vegas、公平性、安定性、過渡特性

Analysis of a Window-Based Flow Control Mechanism based on TCP Vegas in Multi-Hop Networks

Keiichi Takagaki† Hiroyuki Ohsaki‡ Masayuki Murata‡

†Department of Informatics and Mathematical Science
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

‡Cybermedia Center, Osaka University
5-1 Mihogaoka, Ibaraki, Osaka 567-0047, Japan

Phone: +81-6-6879-8793, Fax: +81-6-6879-8794
E-mail: {oosaki, murata}@cmc.osaka-u.ac.jp

Abstract Another version of TCP called TCP Vegas, which can achieve better performance than the current TCP Reno, has been proposed and studied. We have analyzed a window-based flow control mechanism based on TCP Vegas for a simple network model having a single bottleneck. In this paper, by extending our previous analysis, we analyze the behavior of the window-based flow control mechanism for a generic network configuration having multiple bottlenecks. We first derive equilibrium values in steady state: the window size of each TCP connection and the number of packets queued in the router's buffer. We then investigate how control parameters of the window-based flow control mechanism should be configured for achieving desirable fairness among TCP connections. We finally quantitatively show the effect of control parameters of the window-based flow control mechanism on stability and transient behavior of the network.
key words Window-Based Flow Control Mechanism, TCP Vegas, Fairness, Stability, Transient Behavior

1 研究の背景

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 が提案されている [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, 5] 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。これらの研究では、ネットワーク中に単一のボトルネックルータが存在する場合を対象とし、TCP のコネクション数や、各 TCP コネクションの伝搬遅延時間、ウィンドウ型フロー制御方式の制御パラメータなどが、ネットワーク全体の性能にどのような影響を与えるかを明らかにした。しかし、解析モデルが単一のボトルネックルータのみが存在する場合に限定されていたため、解析結果を現実のネットワークにそのまま適用するのは困難であった。そこで本稿では、文献 [3, 4, 5] における解析手法を拡張することにより、多段接続されたネットワークにおいて、ネットワーク中に複数のボトルネックルータが存在する場合を対象とした解析を行う。

TCP のウィンドウ型フロー制御方式は、各 TCP コネクションがそれぞれ独立にフロー制御を行う、分散型の制御である。このため、TCP コネクション間の公平性を実現することは一般に容易ではない。そこで本稿では、まず、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について検討を行う。TCP Vegas にもとづくウィンドウ型フロー制御方式では、制御パラメータを適切に選択すれば、定常状態においてネットワークの動作が安定する。本稿では、定常状態における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を導出する。この結果を用いて、TCP コネクション間の公平性を実現するために、ウィンドウ型フロー制御方式の制御パラメータを、どのように決定すべきかを明らかにする。文献 [6] では、定常状態における各 TCP Vegas のコネクションのスループットは、ある目的関数を最大化するように分配されることを示している。しかし文献 [6] では、複数のボトルネックリンクを経由する場合に、具体的にスループットにどのような影響があるかについては書かれていない。本稿では、ネットワークトポロジなどをある程度限定することによって複数のボトルネックリンクを経由する場合に、TCP Vegas のスループットにどのような影響があるかについて定量的に明らかにする。

TCP のウィンドウ型フロー制御方式は、ネットワークの輻輳状況に応じて送信側ホストのウィンドウサイズを調整する、フィードバック型の制御である。このため、制御パラメータの値が不適切であれば、ネットワークの動作が不安定となりうる。また、一般にネットワークの安定性と過渡特性にはトレードオフの関係があるため、これらの両方を考慮して制御パラメータの値を決定する必要がある。そこで本稿では、現代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制御パラメータと、ネットワークの安定性および過渡特性の関係を定量的に明らかにする。

以下 2 章において、本稿で対象とする解析モデル、安定性解析の手法、公平性の考察などを行う。次に 3 章において具体的な数値例を用いて安定性解析を行った結果や、公平性についてより詳細に述べる。最後に、4 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 定常状態解析

2.1 TCP Vegas の輻輳回避機構と解析モデル

本稿では、定常状態におけるネットワークのふるまいを解析する。このため、すべての送信側ホストは TCP Vegas の輻輳回避機構に従ってウィンドウサイズを変更すると仮定する。また、すべてのルータのバッファは十分大きく、ネットワーク内部でパケット棄却は発生しないものと仮定する。TCP Vegas の輻輳回避機構は、ラウンドトリップ時間ごとに、送信側ホストのウィンドウサイズ w_c を以下のように変更する。

$$w_c \leftarrow \begin{cases} w_c + 1 & \text{if } d_c < \alpha_c \\ w_c - 1 & \text{if } d_c > \beta_c \\ w_c & \text{otherwise} \end{cases} \tag{1}$$

ここで d_c は TCP Vegas の輻輳回避機構によって計算される、ネットワークの輻輳状況を示す値である。これは送信側ホストにおいて以下のように計算される。

$$d_c = \left(\frac{w_c}{\tau_c} - \frac{w_c}{r_c} \right) \tau_c \tag{2}$$

ここで、τ_c は往復伝搬遅延時間であり、r_c は実際に観測されたラウンドトリップ時間である。つまり、期待されるスループット (w_c/τ_c) と実際のスループット (w_c/r_c) が等しい場合に d_c は 0 となり、スループットが低下するにつれ d_c の値が大きくなる。α_c および β_c は、ネットワーク内にどれだけ余分なパケットを送出するかを決定する、TCP Vegas の制御パラメータである。

本解析では、TCP Vegas の輻輳回避機構を一部変更し、送信側ホストのウィンドウサイズ w_c を以下のように変更する。

$$w_c \quad \leftarrow \quad (w_c + \delta_c(\gamma_c - d_c))^+ \tag{3}$$

ここで、$(x)^+ \equiv \max(x, 0)$ と定義する。また、δ_c は、

表 1: 記号の定義 (送信側ホスト)

w_c	:	コネクション c のウィンドウサイズ
γ_c	:	コネクション c の制御パラメータ (バッファ占有量)
δ_c	:	コネクション c の制御パラメータ (ウィンドウサイズの変化量)
τ_c	:	コネクション c の往復伝搬遅延時間
r_c	:	コネクション c のラウンドトリップ時間

ネットワーク内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。TCP Vegas の輻輳回避機構では、式 (1) のように、d_c の値が $[\alpha_c,\beta_c]$ の時はウィンドウサイズを変更しないが、この結果コネクション間の公平性が低下することが知られている [7] 。そこで本解析では、TCP Vegas の輻輳回避機構におけるコネクション間の不公平性を防ぐために、2 つの制御パラメータ α_c および β_c の代わりに、単一の制御パラメータ γ_c を用いている。本節で定義した記号を表 1 に示す。

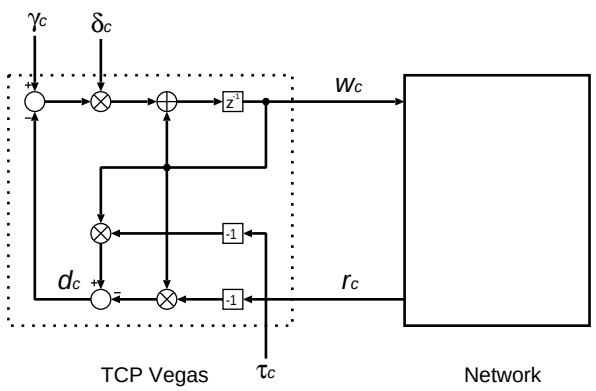

図 1: ウィンドウ型フロー制御方式のブロック線図

TCP Vegas の輻輳回避機構にもとづく、ウィンドウ型フロー制御方式のブロック線図は図 1 のようになる (ここでは w_c の境界条件を考慮していない)。図中の「z^{-1}」はラウンドトリップ時間ぶんの遅延要素を、「-1」は-1 乗をあらわしている。左側の点線で囲まれた部分が、TCP Vegas にもとづくウィンドウ型フロー制御方式をあらわしており、右側の実線で囲まれた部分が、送信側ホストから見たネットワークをあらわしている。この図より、TCP Vegas にもとづくウィンドウ型フロー制御方式は、ネットワークから観測されるラウンドトリップ時間を入力とし、ウィンドウサイズを出力とする、1 入力 1 出力の非線型フィードバック制御ととらえることができる。

本稿では、任意のネットワークトポロジにおいて、すべての送信側ホストが、TCP Vegas にもとづくウィンドウ型フロー制御方式によってフロー制御された場合を解析する。本解析では、複数の TCP コネクションを取り扱うが、送信側ホストおよび受信側ホストが接続されているルータがそれぞれ同じである TCP コネクションは、すべて同じ経路を通る (つまり、同じルータを同じ順序で経由する) と仮定する。また、TCP コネクションのルーティングは固定されていると仮定する。ネットワークのモデル化に用いる記号を、表 2 のように定義する。

表 2: 記号の定義 (ネットワーク)

$\mathcal{N}$	:	ノード (ホストもしくはルータ) の集合
$\mathcal{L}$	:	リンクの集合
μ_l	:	リンク l の伝送容量
τ_l	:	リンク l の伝搬遅延時間
q_l	:	リンク l へ向かう出力バッファ内パケット数
$\mathcal{C}$	:	コネクションの集合
$C_i \subset \mathcal{C}$	:	ルータ間で同じ経路を通るコネクションの集合 (送受信側ホストは異なる)
$\mathcal{L}(c) \subset \mathcal{L}$	:	コネクション c が経由するリンクの集合
$\mathcal{C}(l) \subset \mathcal{C}$	:	リンク l を通過するコネクションの集合
$\mathcal{L}(c,l)$	:	コネクション c がリンク l 以降に経由するリンクの集合

本解析では、すべてのルータは、出力段にそれぞれのリンクごとに異なるバッファを持つと仮定する。各出力バッファは、パケットを到着順に処理する、FIFO (First-In First-Out) キューによりモデル化する。さらに、出力バッファが空であれば、到着したパケットはすぐさま出力リンクに送られる (つまり、ルータにおける処理遅延は無視できる) と仮定する。

受信側ホストは送信側ホストからのパケットを受信した場合、即座に ACK パケットを送信側ホストに送り返す。受信側ホストから送信側ホストへの逆方向の経路では輻輳が発生しないと仮定すれば、コネクション c の往復伝搬遅延時間 τ_c は、以下の式で与えられる。

$$\tau_c = \sum_{l \in \mathcal{L}(c)} \tau_l$$

ここで、各リンクの往復伝搬遅延時間の比をあらわす、既約な正整数を Δ_l ($l \in \mathcal{L}$) とおけば、

$$\forall l \quad \frac{\tau_l}{\Delta_l} = \tau$$

となる定数 τ が存在する。本稿では、ネットワーク全体を、τ を単位として時間をスロットに分割した、離散時間モデルで考える。

2.2 状態遷移方程式の導出

前節で述べた解析モデルにおいて、ネットワークの状態は、TCP コネクション c のウィンドウサイズ w_c 、ボトルネックリンク l へ向かう出力バッファ内パケット数 q_l で一意に定められる。本稿でボトルネックリンクとは、定常状態においてリンク容量とリンクを通過するパケットのレートが等しいリンクのことをいう。いいかえれば、定常状態においてリンクへ向かうルータのバッファ内パケット数が 0 でないリンクのことである。本節では、各スロット間で w_c や q_l がどのように変化するかを記述する、状態遷移方程式を導出する。

まず、k 番目のスロットにおける w_c や q_l の値を、$w_c(k)$ および $q_l(k)$ のように表記する。この時、式 (2) および (3) より、ウィンドウサイズ $w_c(k)$ は以下の式で与えら

れる。

$$w_c(k) = \begin{cases} \left(w_c(k-\frac{\tau_c}{\tau}) + \delta_c(\gamma_c - d_c(k))\right)^+ \\ \qquad \text{if} \quad k \equiv 0 \pmod{\frac{\tau_c}{\tau}} \\ w_c(k-1) \quad \text{otherwise} \end{cases} \tag{4}$$

ただし、

$$d_c(k) = \left(\frac{w_c(k-\frac{\tau_c}{\tau})}{\tau_c} - \frac{w_c(k-\frac{\tau_c}{\tau})}{r_c(k)}\right)\tau_c \tag{5}$$

$$r_c(k) = \tau_c + \sum_{l\in\mathcal{L}(c)} \frac{q_l(k-\sum_{m\in\mathcal{L}(c,l)}\Delta_m)}{\mu_l} \tag{6}$$

となる。また、リンク l へのルータのバッファ内パケット数は以下のように変化する。

$$q_l(k) = \left\{ q_l(k-1) + \left(\sum_{c\in\mathcal{C}(l)} R_{l,c}(k-1) - \mu_l\right)\tau \right\}^+ \tag{7}$$

ここで、$R_{l,c}(k)$ はスロット k にリンク l を通過するコネクション c のパケットのレートであり、以下のように再帰的に定義される。

$$R_{l,c}(k) = \begin{cases} \frac{w_c(k)}{\hat{r}_c(k)} & \text{if } l \in L \\ \mu_l \frac{R_{b(l,c),c}(k)}{\sum_{c'\in\mathcal{C}(l)} R_{b(l,c'),c'}(k)} & \text{if } l \notin L \wedge q_l(k) > 0 \\ R_{b(l,c),c}(k) & \text{if } l \notin L \wedge q_l(k) = 0 \end{cases}$$

但し、

$$\hat{r}_c(k) = \tau_c + \sum_{l\in\mathcal{L}(c)} \frac{q_l(k)}{\mu_l}$$

ここで、$b(l,c)$ はコネクション c がリンク l の直前に経由するリンクを表す。また、$L(\subset \mathcal{L})$ は送信側ホストから、その送信側ホストが直接接続されているルータへのリンクの集合である。

2.3 スループットおよび公平性に関する考察

TCP Vegas にもとづくウィンドウ型フロー制御方式では、ネットワークの状況に応じて、制御パラメータを適切に設定すれば、定常状態において TCP コネクション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する [3]。本節では、ネットワークの動作が安定となる (次節で導出する安定条件をみたす) 場合に、TCP コネクションのウィンドウサイズ $w_c(k)$ や、バッファ内パケット数 $q_l(k)$ が収束する値 (平衡点と呼び、w_c^* や q_l^* のように表記する) を導出する。

式 (4)–(6) より、定常状態では以下の関係が成立する。

$$\gamma_c = d_c^* \tag{8}$$

$$d_c^* = \left(\frac{w_c^*}{\tau_c} - \frac{w_c^*}{r_c^*}\right)\tau_c \tag{9}$$

$$r_c^* = \tau_c + \theta_c^* \tag{10}$$

ただし、θ_c^* はコネクション c に属するパケットが、ルータのバッファで待たされる時間の合計であり、次式のように定義される。

$$\theta_c^* \equiv \sum_{l\in\mathcal{L}(c)} \frac{q_l^*}{\mu_l}$$

式 (8)–(10) より、定常状態においては

$$\gamma_c = \frac{\theta_c^*}{\tau_c + \theta_c^*} w_c^* \tag{11}$$

という関係が成り立つことがわかる。定常状態における、TCP コネクション c のスループットを $\rho_c^*(=w_c^*/r_c^*)$ とすれば、式 (11) より、

$$\gamma_c = \rho_c^* \, \theta_c^* \tag{12}$$

が成立する。これより、定常状態における TCP コネクションのスループットは、制御パラメータ γ_c の値と、ルータのバッファにおける待ち時間の合計 θ_c^* で決まることがわかる。なお、ρ_c^* をコネクション c からのパケット到着率、θ_c^* をルータのバッファでの待ち時間ととらえれば、式 (12) は Little の公式とみなすことができる。つまり、コネクション c に属するパケットのうち、ルータのバッファで待たされているパケットの合計が γ_c で与えられることがわかる。

さらに、式 (11) および (12) より、

$$w_c^* = \rho_c^* \tau_c + \gamma_c$$

という関係が成り立つことがわかる。つまり、TCP Vegas にもとづくウィンドウ型フロー制御方式では、各コネクションのウィンドウサイズが、

$$(\text{帯域}) \times (\text{遅延時間}) + (\text{制御パラメータ}\gamma_c)$$

の値に収束することがわかる。

制御パラメータ γ_c の値は、以下に示すように、TCP コネクション間の公平性にも大きな影響を与える。今、同じ経路を通る 2 種類の TCP コネクション c および c' を考える。経路が同じであれば、ルータのバッファにおける待ち時間の合計 θ_c^* も等しくなる。従って、式 (12) より、これらの TCP コネクションのスループットの比は、以下の式で与えられる。

$$\frac{\rho_c^*}{\rho_{c'}^*} = \frac{\gamma_c}{\gamma_{c'}}$$

つまり、同一の経路を通るコネクションのスループットの比は、制御パラメータ γ_c の値 のみ に依存することがわかる。

2 種類の TCP コネクション c および c' が異なる経路を通る場合、スループットの比は次式で与えられる。

$$\frac{\rho_c^*}{\rho_{c'}^*} = \frac{\gamma_c \, \theta_{c'}^*}{\gamma_{c'} \, \theta_c^*}$$

つまり、θ_c^* の値が小さいほど、他のコネクションに比べて相対的にスループットが向上することがわかる。式 (11) より、一般的にコネクションが経由するルータの数 (ホップ数) が少ないほど、もしくはリンク容量の大きな

リンクを経由するほど、相対的にスループットが向上することがわかる。

なお、定常状態における、ウィンドウサイズ w_c^* や q_l^* の値を求めるためには、すべてのコネクションおよびボトルネックリンクについて、式 (11) および (13) を連立に解けばよい。

$$\sum_{c \in \mathcal{C}(l)} \rho_c^* = \mu_l \tag{13}$$

式 (13) は、平衡点においては、ボトルネックリンク l を経由するコネクションのスループットの合計がリンク容量に等しくなることを意味している。ただし、ネットワークのトポロジが複雑な場合は、数値計算によって解を求める必要がある。なお、*scaled gradient projection algorithm* によって解を求める方法が、文献 [6] で紹介されている。

2.4 安定性及び過渡特性

ウィンドウ型フロー制御方式はフィードバック型の制御であるため、制御パラメータを適切に設定しなければ、ネットワークの動作が不安定になる場合がある。そこで本節では、制御理論を適用することにより、ネットワークの動作が安定となるための条件を導出する。

まず、k 番目のスロットにおけるシステムの状態と、それぞれの平衡点との差を以下のように定義する。本解析では $C_i(1 \leq i \leq N)$ に属する全てのコネクションは、コネクション開始時のウィンドウサイズが等しく、その後も同期してウィンドウサイズを変化させると仮定する。

$$\mathbf{x}(k) \equiv \begin{bmatrix} w_{c_1}(k) - w_{c_1}^* \\ \vdots \\ w_{c_N}(k) - w_{c_N}^* \\ q_{l_1}(k) - q_{l_1}^* \\ \vdots \\ q_{l_{|\mathcal{L}|}}(k) - q_{l_{|\mathcal{L}|}}^* \end{bmatrix} \tag{14}$$

式 (4) および (7) で与えられる離散時間システムは非線形であるため、平衡点の近傍で 1 次近似することにより、これを線形化する [8] 。

また、Δ_{LCM} をすべての TCP コネクションに関する $\sum_{l \in \mathcal{L}(c)} \Delta_l$ の最小公倍数とする。k 番目のスロットにおいて、すべての TCP コネクションが同期して転送を開始したと仮定すれば、$k + \Delta_{LCM}$ スロットにおいて、再びすべての TCP コネクションが同期する。従って、$\mathbf{x}(k)$ から $\mathbf{x}(k + \Delta_{LCM})$ への状態遷移は、行列 $\mathbf{A}$ を用いて以下のように書くことができる。

$$\mathbf{x}(k + \Delta_{LCM}) = \mathbf{A}\,\mathbf{x}(k) \tag{15}$$

これは、$N + |\mathcal{L}|$ 次の自由系システムであり、ネットワーク全体の安定性は、行列 $\mathbf{A}$ の固有値によって決定される [9]。つまり、すべての固有値の絶対値が 1 より小さい (つまり、複素平面上で単位円内に存在する) 場合は、ネットワークの動作が安定となる。なお、固有値の絶対値が小さいほど、平衡点の近傍での過渡特性が良くなることがわかっている [10]。

3 数値例による考察

本章では、いくつかの数値例を用いて、多段接続されたネットワークにおける、TCP Vegas にもとづくウィンドウ型フロー制御方式の、公平性や安定性、過渡特性について検討を行う。

3.1 ネットワークモデル

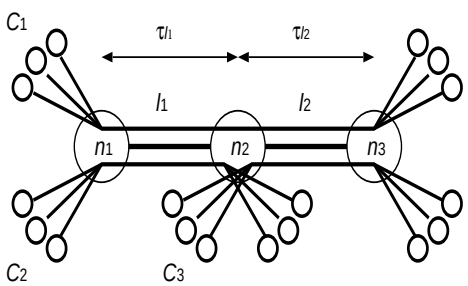

図 2: ネットワークモデル

以下では、図 2 に示すような、比較的単純なネットワークモデルを対象とする。このモデルでは、ルータ $n_i(1 \leq n \leq 3)$ が 3 段直列に接続されている。図中に示すように、すべての TCP コネクションは、どのリンクを経由するかによって 3 種類 $C_i(1 \leq i \leq 3)$ に分類される。以下では、説明の簡単化のため、C_i に属する各 TCP コネクションを $c_i(1 \leq i \leq 3)$ と呼ぶ。TCP コネクション c_1 は、送信側ホストから受信側ホストまで、ルータ $n_1 \rightarrow n_2 \rightarrow n_3$ を経由する。一方、TCP コネクション c_2 および c_3 は、ルータ $n_1 \rightarrow n_2$ もしくは $n_2 \rightarrow n_3$ を経由する。以下では、ルータ–ルータ間のリンク (l_1 および l_2) がボトルネックになると仮定する。つまり、送信側ホスト–ルータ間およびルータ–受信側ホスト間のリンク容量は、l_1 および l_2 に比べて十分大きいものと仮定する。つまり、TCP コネクション c_1 はボトルネックリンクを 2 本、コネクション c_2 および c_3 はボトルネックリンクを 1 本経由することになる。

文献 [5] において、すべての TCP コネクションが経由するルータの数が等しい場合でも、各 TCP コネクションの伝搬遅延時間の違いによって、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ている。そこで、TCP コネクションが経由するルータの数の違いによる影響と、TCP コネクションの伝搬遅延時間の違いによる影響を区別するために、以下に示すような 2 種類のモデルを対象とする。

- モデル 1
  モデル 1 は、TCP コネクションの伝搬遅延時間が、経由するボトルネックリンクの数に比例する場合である。この時、リンクの伝搬遅延時間 Δ_l は以下の式で与えられる。

$$\Delta_l = \begin{cases} 1 & \text{if } l \in (l_1, l_2) \\ 0 & \text{otherwise} \end{cases}$$

- モデル 2
  モデル 2 は、TCP コネクションの伝搬遅延時間が、経由するボトルネックリンクの数によらず一定の場合である。この時、リンクの伝搬遅延時間 Δ_l は以下の式で与えられる。

$$\Delta_l = \begin{cases} 1 & \text{if } l \in (l_1, l_2, \mathcal{L}(c_2, l_1), \mathcal{L}(c_3) - \mathcal{L}(c_3, l_2)) \\ 0 & \text{otherwise} \end{cases}$$

3.2 公平性に関する考察

モデル 1 での各コネクションの定常状態におけるスループット ρ_c^* は、図 3 のようになる。ここでは、$|C_1|$=$|C_2|$=$|C_3|$=10[本]、τ=1 [ms]、γ_{c_1}=γ_{c_2}=γ_{c_3}=3 [パケット] である。

ここでは μ_{l_1} を 20[パケット/ms] と固定し、μ_{l_2} を変化させたときの各コネクションのスループットの変化をプロットしている。

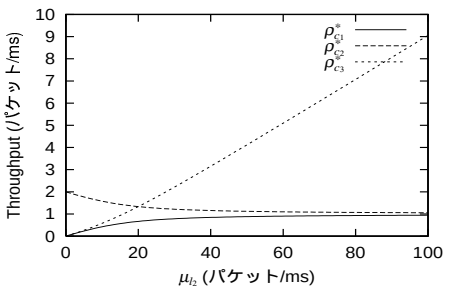

図 3: モデル 1 における各コネクションのスループット

図 3 から分かるように、μ_{l_2} が 20[パケット /ms] のとき、つまり $\mu_{l_1} = \mu_{l_2}$ のときは、$\rho^*_{c_2} = \rho^*_{c_3} = 2\rho^*_{c_1}$ となる。すなわち、2 本のボトルネックリンクを経由するコネクションのスループットが 1 本のボトルネックリンクを経由するコネクションのスループットの 2 倍になっている。また、μ_{l_2} が大きくなると、$\rho^*_{c_1}$ と $\rho^*_{c_2}$ が近付いていくのが分かる。また、平衡点での値を計算することにより、

$$\lim_{\mu_{l_2} \to \infty} \frac{\rho^*_{c_1}}{\rho^*_{c_2}} = 1 \tag{16}$$

となる。このときの ρ_c^* は、コネクション c_1 がリンク l_2 を通らないとした場合の ρ_c^* と等しくなる。このことから、ボトルネックとなるリンクが複数であっても、そのリンク容量に大きな差がある場合には、容量の大きなボトルネックリンクが存在しない場合に比べて、コネクションのスループットがあまり低下しないことが分かる。

2.3 節で述べたように、一般的に経由するボトルネックの数が増えるとコネクションのスループットが低下するが、送信側での制御パラメータ γ_c を大きくすることによって、スループットを向上することができる。つまり、各送信側における制御パラメータ γ_c を適当に定めることによって、ボトルネックリンクを複数経由するコネクションが存在する場合にも、適当に各コネクションにスループットを割り当てることができる。よく知られる公平性の一つとして *Max-Min* 公平性 [11] がある。*Max-Min* 公平性では、スループットの最も低いコネクションに、できるだけ大きなスループット割り当てるようにする。モデル 1 において *Max-Min* 公平性を実現するためには、$\mu_{l_1} > \mu_{l_2}$ のときには、以下を満たすように γ_c を決めれば良い。

$$\frac{\mu_{l_1}}{\mu_{l_2}} = \frac{|C_1|(\gamma_{c_1} - \gamma_{c_2}) + |C_3|\gamma_{c_3}}{(|C_1| + |C_2|)(\gamma_{c_1} - \gamma_{c_2})} \tag{17}$$

3.3 安定性及び過渡特性に関する考察

モデル 1 および モデル 2 では、式 (14) によって定義される、システムの状態ベクトルは次式で与えられる。

$$\mathbf{x}(k) \equiv \begin{bmatrix} w_{c_1}(k) - w^*_{c_1} \\ w_{c_2}(k) - w^*_{c_2} \\ w_{c_3}(k) - w^*_{c_3} \\ q_{l_1}(k) - q^*_{l_1} \\ q_{l_2}(k) - q^*_{l_2} \end{bmatrix}$$

このため、式 (15) における状態遷移行列 **A** は、5 次の正方行列となる。2.4 節で述べたように、ネットワーク全体の安定性および過渡特性は、状態遷移行列 **A** の固有値の大きさによって決定される。

まず、モデル 1 における安定条件 (5 つの固有値 $\lambda_i (1 \leq i \leq 5)$ が単位円内にあるための条件) を図 4 に示す。l_1 のリンク容量 μ_1 と l_2 のリンク容量 μ_2 を同じ値 μ_l とし、μ_l を 0.2 ~ 2,000[パケット /ms] と変化させた時の安定領域 (ネットワークの動作が安定となるパラメータ領域) を示している。この図は、$(\delta_{c_1}, \delta_{c_2})$ の点が、境界線の内側に存在するとき、ネットワークの動作が安定となることを示している。なお、コネクション c_2 と c_3 の制御パラメータ δ_c の値は同じ値 $(\delta_{c_2} = \delta_{c_3})$ に設定している。その他のパラメータについては、$|C_1|$=$|C_2|$=$|C_3|$=10 [本]、τ=1 [ms]、γ_1=γ_2=γ_3=3 [パケット] とする。

この図より、リンク容量 μ_l が小さくなるにつれ、安定領域がほぼ $0 \leq \delta_{c_1}, \delta_{c_2} \leq 2$ に収束することがわかる。これはつまり、リンク容量が非常に小さい場合には、TCP コネクションの制御パラメータ δ_c の値は、TCP コネクションが経由するボトルネックリンクの数に無関係となることを意味している。一方、リンク容量 μ_l が大きくなるにつれ、安定領域がほぼ $0 \leq \delta_{c_1} \leq 3.5, 0 \leq \delta_{c_2} \leq 1$ 内に収束することがわかる。これはつまり、リンク容量が非常に大きい場合には、経由するボトルネックリンクの数が多いコネクションほど、制御パラメータ δ_c の値を大きくできることを意味している。ただしこれは、TCP コネクション c_1 の伝搬遅延時間が、他の TCP コネクションの伝搬遅延時間に比べて大きいことも影響していると考えられる [5] 。

さて、TCP コネクションの制御パラメータ δ_c を増加させた場合、ネットワーク全体の動作がどのように不安定になるかは、状態遷移行列 **A** の固有値 λ_i の変化から説明できる。図 5 は、リンク容量 μ_l が 0.2 [パケット/ms] の場合に、$\delta_c (= \delta_{c_1} = \delta_{c_2} = \delta_{c_3})$ を 0 ~ 2.1 へと変化させた時の、固有値 $\lambda_i (1 \leq i \leq 5)$ の値を複素平面上に描いたものである。その他のパラメータは図 4 と同じである。この図より、制御パラメータ δ_c の値が大きくなる

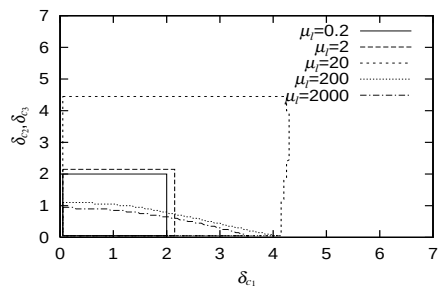

図 4: モデル 1 の安定領域 $|C_1|=|C_2|=|C_3|=10$[本]、τ=1 [ms]、$\gamma_{c_1}=\gamma_{c_2}=\gamma_{c_3}$=3 [パケット]

につれ、3 つの固有値 λ_1 、λ_2 、λ_3 が単位円外に移動していることがわかる。一方、2 つの固有値 λ_4 および λ_5 は、制御パラメータ δ_c の値にほとんど影響を受けていないこともわかる。

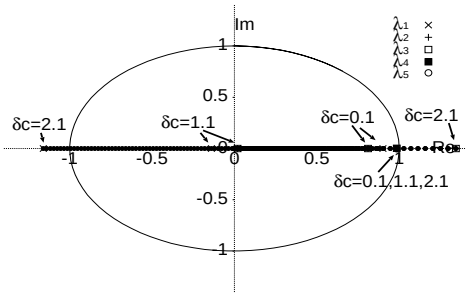

図 5: 複素平面上での固有値の変化 ($\mu_l = 0.2$ [パケット /ms])

それぞれの固有値 λ_i の値が、物理的にどのような意味を持っているかは、その固有値に対応する固有ベクトルを調べることによって調べることができる。例えば、リンク容量が非常に小さい場合として、$\mu_l \to \infty$ の場合を考えると、固有値 λ_i は、以下の値に収束する。$\lim_{\mu_l \to 0} \mathbf{A}$ の固有値は

$$\begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \\ \lambda_5 \end{bmatrix} \simeq \begin{bmatrix} 1-\delta_{c_1} \\ (1-\delta_{c_2})^2 \\ (1-\delta_{c_3})^2 \\ 1 \\ 1 \end{bmatrix} \tag{18}$$

のようになる。これより、$\mu_l \to 0$ の時の安定条件は、$0 < \lambda_i < 2 (1 \le i \le 3)$ となることがわかる。また簡単な計算により、固有値 $\lambda_i (1 \le i \le 3)$ に対応する固有ベクトルは、w_{c_i} 以外の要素が 0 となることがわかる。つまり、リンク容量が非常に小さい場合、ある TCP コネクション c の制御パラメータ δ_c の値は、そのコネクションのウィンドウサイズ w_c の安定性 のみ に影響を与えることがわかる。

次に、リンク容量 μ_l が 20[パケット /ms] の時の、固有値 λ_i の変化を図 6 に示す。ここでは、$\delta_c (= \delta_{c_1} = \delta_{c_2} = \delta_{c_3})$ の値を、0 ~ 4.6 まで変化させている。その他のパラメータについては、図 5 と同じ値を用いている。これより、制御パラメータ δ_c が変化するにつれ、すべての固有値 λ_i が複雑に変動していることがわかる。特に、制御パラメータ δ_c が大きくなるにつれ、λ_4 が実軸上を左に、λ_2、λ_3 が実軸付近を右に移動している。このため、δ_c 4.5 程度で λ_2、λ_3、λ_4 が単位円外に移動し、ネットワーク全体が不安定となることがわかる。また、固有値が複素平面上で原点に近づくにつれ、ネットワーク全体の過渡特性が向上する。従って、制御パラメータ δ_c を適切に設定することにより、ネットワーク全体の過渡特性が改善できることがわかる。

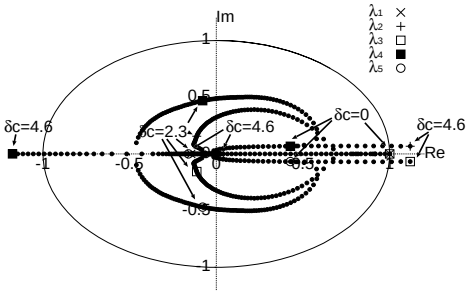

図 6: 複素平面上での固有値の変化 ($\mu_l = 20$ [パケット /ms])

しかし、リンク容量 μ_l が非常に大きい場合には、制御パラメータ δ_c をどのように選んでも過渡特性が改善されない。このことは、図 7 から知ることができる。この図は、$\mu_l = 2000$ [パケット /ms] とし、$\delta_c (= \delta_{c_1} = \delta_{c_2} = \delta_{c_3})$ の値を、0.1 ~ 1.0 まで変化させた時の、固有値 λ_i の変化を示している。この図より、制御パラメータ δ_c の値を変化させても、λ_1 が単位円の近傍から移動しないことがわかる。また、他の固有値についても、制御パラメータ δ_c の値をどのように選んでも、固有値が複素平面上で原点に近づくことはない。このように、リンク容量 μ_l が非常に大きい場合には、過渡特性が非常に悪くなることがわかる。

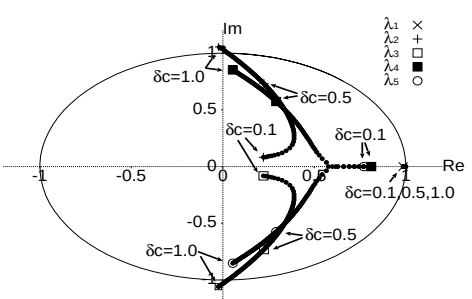

図 7: 複素平面上での固有値の変化 ($\mu_l = 2000$ [パケット/ms])

モデル 1 では、TCP コネクションの伝搬遅延時間は、経由するボトルネックリンクの数に比例していた。一方、モデル 2 では、TCP コネクションの伝搬遅延時間が、経由するボトルネックリンクの数によらず一定である。このため、モデル 2 では経由するボトルネックリンクの数が、ネットワーク全体の安定性や過渡特性に、どのよう

な影響を与えるかが、より顕著にあらわれると考えられる。図 8 に、モデル 2 における安定領域を示す。TCP の制御パラメータやリンクの伝搬遅延時間など、すべてのパラメータは図 4 と同じ値を用いている。

図 4 および図 8 を比較すれば、リンク容量 μ_l が小さい場合には、モデル 1 とモデル 2 の安定領域はほぼ同じであることがわかる。しかし、リンク容量 μ_l が大きくなるにつれ、モデル 2 の安定領域が、ほぼ $0 \leq \delta_{c_1} \leq 2.4$、$0 \leq \delta_{c_2} \leq 1.2$ 内に収束することがわかる。このことから、TCP コネクションの伝搬遅延時間が同じであっても、経由するボトルネックリンクの数が多いコネクションほど、制御パラメータ δ_c の値を大きく設定できることがわかる。つまり、複数のボトルネックリンクを経由する TCP コネクションは、単一のボトルネックリンクを経由する TCP コネクションに比べて、ネットワークの安定性に与える影響が小さいといえる。これは、TCP Vegas にもとづくウィンドウ型フロー制御方式では、2.3 節で述べたように、経由するボトルネックリンクの数が多くなるにつれ、スループットが低下するためと考えられる。つまり、経由するボトルネックリンクの少ないコネクションは、結果としてスループットが低く抑えられるため、ネットワークの安定性に与える影響も小さくなると考えられる。

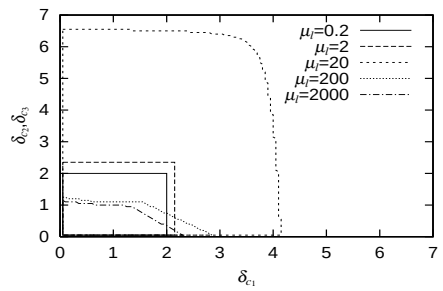

図 8: モデル 2 の安定領域 N_{C_1}=N_{C_2}=N_{C_3} = 10 [本], τ=1 [ms],γ_{c_1}=γ_{c_2}=γ_{c_3}=3 [パケット]

4 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を解析した。特に、多段接続されたネットワークを対象とし、経由するボトルネックの数がコネクション間の公平性や、ネットワーク全体の安定性及び過渡特性に与える影響を明らかにした。その結果、TCP Vegas は経由するボトルネックリンクの数が多くなると TCP コネクションのスループットが低下することが分かった。また、経由するボトルネックリンクの数が多くなるほど、そのコネクションのウィンドウサイズの変化がネットワーク全体の安定性に与える影響が小さくなることが分かった。今後の課題としては、あらゆるネットワークの状態に対応して、公平かつ安定な制御を行うための補償器の設計などを行う必要があると考えられる。

謝辞

日本学術振興会未来開拓学術研究事業「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) 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] 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, October 1998.

[4] 高垣 景一, 大崎 博之, 村田 正幸, "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安定性解析," 電子情報通信学会技術研究報告 (CQ99-60), pp. 25–30, October 1999.

[5] 高垣 景一, 大崎 博之, 村田 正幸, "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," 電子情報通信学会技術研究報告 (SSE2000-3), pp. 13–18, April 2000.

[6] L. P. Steven Low and L. Wang, "Understanding TCP Vegas: Theory and practice." available at `http://www.ee.mu.oz.au/staff/slow/research/`.

[7] G. Hasegawa, M. Murata, and H. Miyahara, "Fairness and stability of congestion control mechanism of TCP," in *Proceedings of 11th ITC Special Seminar*, pp. 255–262, October 1998.

[8] 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.

[9] G. F. Franklin and J. D. Powell, *Digital Control of Dyanamic Systems*. Addision-Wesley Publishing Company, 1980.

[10] 伊藤正美, 自動制御概論 *[上]*. 昭晃堂, 1983.

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