

# 改造 TCP がネットワークに与える影響に関する一検討

丸山 純一<sup>†</sup> 長谷川 剛<sup>†</sup> 村田 正幸<sup>††</sup>

<sup>††</sup> 大阪大学大学院情報科学研究科 〒 565-0871 吹田市山田丘 1-5

<sup>†</sup> 大阪大学サイバーメディアセンター 〒 560-0043 豊中市待兼山町 1-32

E-mail: <sup>†</sup>{j-maruyama,murata}@ist.osaka-u.ac.jp, <sup>††</sup>hasegawa@cmc.osaka-u.ac.jp

あらまし 本稿では、悪意あるユーザによって不正に改造され、利己的な挙動をする TCP (tampered-TCP と呼ぶ) がネットワークに与える影響を評価する。具体的には、tampered-TCP の中でもウィンドウサイズの上げ幅と下げ幅を変更したものを対象とし、数学的解析手法を用いて、TCP Reno コネクションと tampered-TCP コネクションが共存する環境における、tampered-TCP コネクションの平均スループットを導出する。さらに、シミュレーション評価によって数学的解析手法の妥当性を検証するとともに、以下の 3 点を明らかにする。(1) 上げ幅が 3 以上の場合、再送タイムアウトが増加することによってスループットが低下すること、(2) 下げ幅を小さくすることでスループットが増大すること、(3) 下げ幅を小さくすることの効果より、上げ幅の増加によるスループットの低下の影響の方が強いこと。これらの結果を通じて、tampered-TCP の有効範囲がごく狭い領域に限られることを示す。

キーワード tampered-TCP, ウィンドウサイズ, 上げ幅, 下げ幅, スループット

## Evaluation of the effectiveness of tampered-TCP

Junichi MARUYAMA<sup>†</sup>, Go HASEGAWA<sup>†</sup>, and Masayuki MURATA<sup>††</sup>

<sup>††</sup> Graduate School of Information Science and Technology, Osaka University Yamadaoka 1-5, Suita-shi, Osaka 565-0871 Japan

<sup>†</sup> Cyber Media Center, Osaka University Machikaneyama 1-32, Toyonaka-shi, Osaka 560-0043 Japan

E-mail: <sup>†</sup>{j-maruyama,murata}@ist.osaka-u.ac.jp, <sup>††</sup>hasegawa@cmc.osaka-u.ac.jp

**Abstract** In this paper, we investigate the effectiveness of tampered-TCP, in which the congestion control mechanism is modified by malicious users for higher throughput than the normal TCP. We focus on a tampered-TCP which changes the increase and decrease rate of congestion window size. We discuss the performance of the tampered-TCP through the mathematical analysis and simulation, and show its characteristics as: (1) when the increase rate is larger than 2 packets per RTT, the throughput degrades significantly due to many timeouts, (2) decreasing the decrease rate is effective for the throughput improvement, (3) the effectiveness of decreasing the decrease rate is smaller than the throughput degradation due to increasing the increase rate. We conclude that there is little region where the tampered-TCP obtains much higher throughput.

**Key words** tampered-TCP, window size, increase rate, decrease rate, throughput

### 1. はじめに

現在のインターネットには、[1-3] に示されるように悪意あるユーザによって改造された TCP が存在する。これは、カーネル内のソースコードやパラメータを少し変えるだけで TCP を改造できることに起因する。例えば、輻輳ウィンドウサイズの増減のレートを変更するには、カーネル内のソースコードのうち、輻輳ウィンドウサイズの増加幅と減少幅を決定しているパラメータを変更するだけでよく、わずかに数行の改変で改造が可能である。このようにして改造された TCP を、本稿では tampered-TCP と呼ぶ。

改造者が増えてネットワーク内に tampered-TCP コネクションが増加すると、公平性の観点から他のユーザに悪影響を与え、という問題が発生することが考えられる。上述の輻輳ウィンドウサイズの上げ幅と下げ幅を変更した tampered-TCP においては、[4, 5] のように公平性を考慮に入れた改造を行った場合

には、競合する他のコネクションへの影響は小さいと考えられる。だが、改造者がこのように公平性を考慮に入れて改造することは期待できない。そのため、tampered-TCP コネクションが多く帯域を占領し、公平性が保たれないことが予想される。

しかし、上記のような安易な改造はうまく機能しない可能性もある。例えば、輻輳ウィンドウサイズの上げ幅を大きくすることで輻輳ウィンドウサイズが急激に大きくなるため、輻輳時に発生するパケット廃棄数が増加し、通常のコネクションよりタイムアウトが発生しやすくなることが考えられる。これは、tampered-TCP が通常のコネクションよりも積極的にデータ転送を行うことで、逆に自身のスループットを低下させることを意味する。

本稿では、この仮説を明らかにすることにより、TCP の安直な改造では改造者が想定するような効果は得られないことを明らかにする。本稿では、輻輳ウィンドウサイズの上げ幅と下げ幅を変更した tampered-TCP に着目して議論を行う。こ

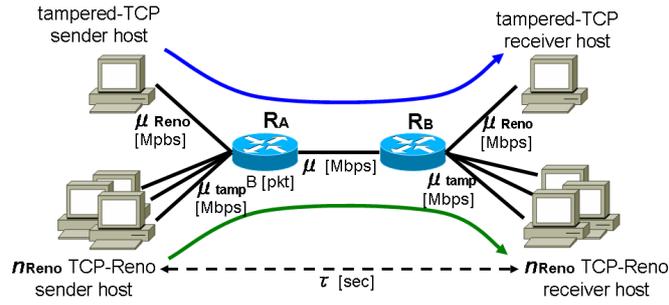


図 1 評価モデル

の tampered-TCP を対象にする理由は次の 2 点である．まず，改造者の視点から見て，改造が容易な上に効果が期待できるため，改造者が採用する可能性がもっとも高いと考えられることである．次に，輻輳ウィンドウサイズの上げ幅および下げ幅を変更した tampered-TCP は数学的解析手法により性能解析が可能 [6, 7] で，その挙動を説明することが比較的容易であることが挙げられる．以降では，輻輳ウィンドウサイズの上げ幅および下げ幅を変更した TCP を単に tampered-TCP と呼び，その有効性に関する議論を行う．

本稿では数学的解析およびシミュレーションを用いて，tampered-TCP の性能評価を行う．まず解析に関しては [7] で行われている数学的解析手法を拡張し，tampered-TCP が通常の TCP Reno と共存する環境でのそれぞれのスループットを導出する．また，シミュレーション結果から，数学的解析手法の妥当性を示し，その結果から，tampered-TCP の性質として，(1) 上げ幅が 3 以上の場合，再送タイムアウトが増加することによってスループットが低下すること，(2) 下げ幅を小さくすることでスループットが増大すること，(3) 下げ幅を小さくすることの効果より，上げ幅の増加によるスループットの低下の影響の方が強いこと，などを明らかにする．これらの検討を通じて，tampered-TCP の有効範囲がごく狭い領域に限られることを示す．

以下，2 章では tampered-TCP の評価環境と評価指標についての説明を行う．3 章では，tampered-TCP の挙動について [7] をもとにした数学的解析を行い，その有効性に関して議論を行う．4 章では，シミュレーション評価によって数学的解析の妥当性を示す．5 章では，TCP Reno コネクションが複数本存在する環境での，tampered-TCP の性能評価を行う．最後に 6 章で本稿のまとめと今後の課題を述べる．

## 2. 評価方法

本稿では，tampered-TCP の性能に関して，現在最もよく利用されている TCP Reno と共存する環境で評価を行う．本章では，具体的な評価環境および評価指標について述べる．

### 2.1 評価モデル

図 1 に，性能評価に用いるネットワークモデルを示す．ネットワークモデルは，tampered-TCP コネクションの送信側/受信側ホスト，TCP Reno コネクションの送信側/受信側ホスト，2 つの Droptail パッファを持つルータ，およびそれらをつなぐリンクから構成されている．ルータ  $R_A$  とルータ  $R_B$  をつなぐリンクの帯域を  $\mu$  Mbps，ルータ  $R_A$  のパッファサイズを  $B$  packets，送信側/受信側ホスト間の伝搬遅延時間を  $\tau$  sec，tampered-TCP コネクションの送信側/受信側ホストとルータ間をつなぐリンクの帯域を  $\mu_{\text{tamp}}$  Mbps，TCP Reno コネクションの送信側/受信側ホストとルータ間をつなぐリンク帯域を  $\mu_{\text{Reno}}$  Mbps とする．コネクション数は，tampered-TCP を 1 本とし，TCP Reno は  $n_{\text{Reno}}$  本とする．各コネクションの送信側ホストには常に送信するデータがあるものとし，ウィンドウサイズが許す限りデータを送信し続けると仮定する．

### 2.2 評価指標

本稿では tampered-TCP の効果を評価するために，tampered-TCP コネクションと TCP Reno コネクションが共存する環境におけるそれぞれのコネクションのスループット値を用いて，次式で定義されるスループット比を評価指標とし

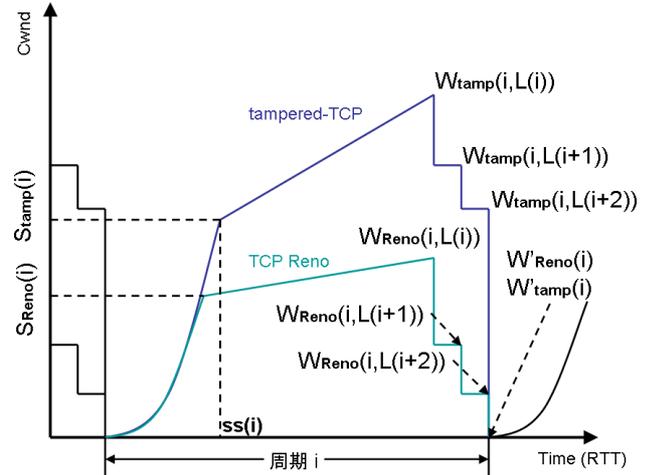


図 2 周期  $i$  におけるウィンドウサイズの時間的変動

て用いる．

$$\text{スループット比} = \frac{(\text{tampered-TCP のスループット})}{(\text{TCP Reno のスループット})}$$

この値が 1 より大きい場合は tampered-TCP が効果を発揮しており，1 より小さい場合は効果を発揮していないと考えることができる．

## 3. スループット解析

### 3.1 TCP Reno および tampered-TCP の挙動

TCP Reno のウィンドウサイズは，パケット廃棄をきっかけして周期的に変化する．図 2 に，TCP Reno コネクション 1 本と tampered-TCP コネクション 1 本がネットワーク内に共存した場合の，両コネクションの輻輳ウィンドウサイズの変化の典型例を示す．ここで， $(i-1)$  番目のパケット廃棄から  $i$  番目のパケット廃棄までの期間を周期  $i$  とする．また，TCP Reno によるデータ転送を RTT ごとに区切り，周期  $i$  の開始時から数えて  $j$  番目の RTT における TCP Reno コネクションのウィンドウサイズを  $W_{\text{Reno}}(i, j)$  と定義する．

TCP Reno の輻輳制御方式は，スロースタートフェーズおよび輻輳回避フェーズと呼ばれる 2 つのフェーズから構成され，それぞれにおいてウィンドウサイズの増加速度が異なる．スロースタートフェーズにおいては，1 つの ACK を受信する度にウィンドウサイズ  $W_{\text{Reno}}(i, j)$  を 1 パケット増加させる．一方輻輳回避フェーズにおいては，1 つの ACK パケットを受信する度にウィンドウサイズ  $W_{\text{Reno}}(i, j)$  を  $1/W_{\text{Reno}}(i, j)$  パケットだけ増加させる．この様子は，以下の式で表すことができる．

$$W_{\text{Reno}}(i, j) = \begin{cases} 2 \cdot W_{\text{Reno}}(i, j-1) & (\text{if } W_{\text{Reno}}(i, j-1) < S_{\text{Reno}}(i)) \\ W_{\text{Reno}}(i, j-1) + 1 & (\text{if } W_{\text{Reno}}(i, j-1) \geq S_{\text{Reno}}(i)) \end{cases} \quad (1)$$

ここで， $S_{\text{Reno}}(i)$  は周期  $i$  における TCP Reno がスロースタートフェーズから輻輳回避フェーズに移行する際の閾値  $ssthresh$  である．ネットワーク内でパケット廃棄が発生すると，TCP Reno はタイムアウトの発生，あるいは重複 ACK の受信によりそれを検出し，廃棄パケットの再送を行う．タイムアウトによってパケット廃棄を検出すると，ウィンドウサイズを 1 に設定してスロースタートフェーズへ移行する．一方，重複 ACK によってパケット廃棄を検出すると，ウィンドウサイズを現在の値の  $1/2$  に設定して輻輳回避フェーズに移行する．また，いずれの場合も  $S_{\text{Reno}}(i)$  の値を現在のウィンドウサイズの値の  $1/2$  に設定する．

tampered-TCP に関して，基本的な動作は TCP Reno と同じである．しかし，輻輳回避フェーズにおいて，ウィンドウサイズの増加速度が TCP Reno の場合と異なる．具体的には，

周期  $i$  の開始時から数えて  $j$  番目の RTT における tampered-TCP コネクションのウィンドウサイズを  $W_{\text{tamp}}(i, j)$  とすると、1 つの ACK を受信する度に  $W_{\text{tamp}}(i, j)$  を  $\alpha \cdot 1/W_{\text{tamp}}(i, j)$  ずつ、つまり TCP Reno の  $\alpha$  倍の速度でウィンドウサイズを増加させる。この様子は、以下の式で表すことができる。

$$W_{\text{tamp}}(i, j) = \begin{cases} 2 \cdot W_{\text{tamp}}(i, j-1) \\ \text{(if } W_{\text{tamp}}(i, j-1) < S_{\text{stamp}}(i)) \\ W_{\text{tamp}}(i, j-1) + \alpha \\ \text{(if } W_{\text{tamp}}(i, j-1) \geq S_{\text{stamp}}(i)) \end{cases} \quad (2)$$

ここで、 $S_{\text{stamp}}(i)$  は周期  $i$  における tampered-TCP の  $ssthresh$  の値である。tampered-TCP がパケット廃棄を検出した場合も、TCP Reno と同様にタイムアウト、または重複 ACK によってそれを検出し、廃棄パケットの再送を行う。タイムアウトによってパケット廃棄を検出した場合は TCP Reno と同じ挙動をする。一方、重複 ACK によってパケット廃棄を検出した場合、tampered-TCP は TCP Reno と違い、ウィンドウサイズを現在の値の  $\beta$  ( $> 1$ ) 倍に設定する。また、いずれの場合も、 $S_{\text{stamp}}(i)$  の値は現在のウィンドウサイズの値の  $\beta$  倍に設定する。

### 3.2 解析

解析では、tampered-TCP および TCP Reno の、パケット廃棄をきっかけとした周期的なウィンドウサイズの変化をモデル化 (図 2) することによって、それぞれのコネクションの平均スループットを導出する。

まず、周期  $i$  終了時点の TCP Reno と tampered-TCP のウィンドウサイズを  $W'_{\text{Reno}}(i)$ 、 $W'_{\text{tamp}}(i)$  とすると、周期  $i$  開始時のウィンドウサイズは、周期  $(i-1)$  終了時のウィンドウサイズに等しいため、以下の式が成り立つ。

$$W_{\text{Reno}}(i, j) = W'_{\text{Reno}}(i-1)$$

$$W_{\text{tamp}}(i, j) = W'_{\text{tamp}}(i-1)$$

式 (1)、(2) にしたがって、各コネクションのウィンドウサイズが増加していくと、ルータ  $R_A$  のバッファにパケットが蓄積され始め、いずれパケット廃棄が発生する。このパケット廃棄が、周期  $i$  の  $L(i)$  番目の RTT で初めて発生するものとする、以下の式が成立する。

$$W_{\text{Reno}}(i, L(i)-1) + W_{\text{tamp}}(i, L(i)-1) \leq 2\tau\mu + B$$

$$W_{\text{Reno}}(i, L(i)) + W_{\text{tamp}}(i, L(i)) > 2\tau\mu + B \quad (3)$$

この時に廃棄されるパケット数を  $D(i)$  とすると、 $D(i)$  は次の式で表される。

$$D(i) = W_{\text{Reno}}(i, L(i)) + W_{\text{tamp}}(i, L(i)) - (2\tau\mu + B)$$

さらに、TCP Reno コネクションの廃棄パケット数を  $D_{\text{Reno}}(i)$ 、tampered-TCP コネクションの廃棄パケット数を  $D_{\text{tamp}}(i)$  とすると、 $D_{\text{Reno}}(i)$  と  $D_{\text{tamp}}(i)$  の比は、パケット廃棄が発生した瞬間の、両コネクションのウィンドウサイズの比に等しいと考えられるため、それぞれ次の式で表される。

$$D_{\text{Reno}}(i) = \frac{W_{\text{Reno}}(i, L(i))}{W_{\text{Reno}}(i, L(i)) + W_{\text{tamp}}(i, L(i))} D(i)$$

$$D_{\text{tamp}}(i) = \frac{W_{\text{tamp}}(i, L(i))}{W_{\text{Reno}}(i, L(i)) + W_{\text{tamp}}(i, L(i))} D(i)$$

次に、パケット廃棄が発生した後のウィンドウサイズを導出する。本解析では、ボトルネックルータ  $R_A$  に Droptail ルータを用いている。そのため、ルータ  $R_A$  のバッファがあふれてパケット廃棄が起きるとき、パースティックにパケット廃棄が発生するため、 $D(i) > 1$  であると仮定する。以下では、各コネクションのパケット廃棄数が 3 以上の場合のウィンドウサイズの導出方法について述べるが、パケット廃棄数が 2 以下の場合でも、以下で述べる方法によってパケット廃棄後のウィンドウサイズを求めることができる。同一ウィンドウ内で 3 個以上のパケットが廃棄されると、TCP Reno は Fast Retransmit アルゴリズムによる再送を 2 度行った後タイムアウトを起こす [8]。

tampered-TCP もパケット廃棄に対して TCP Reno と同様の処理をする。そのため、各コネクションの、1 度目の再送後のウィンドウサイズは次の式で表される。

$$W_{\text{Reno}}(i, L(i)+1) = \frac{W_{\text{Reno}}(i, L(i))}{2}$$

$$W_{\text{tamp}}(i, L(i)+1) = \beta \cdot W_{\text{tamp}}(i, L(i))$$

さらに、2 度目の再送後のウィンドウサイズは次の式で表される。

$$W_{\text{Reno}}(i, L(i)+2) = \frac{W_{\text{Reno}}(i, L(i)+1)}{2}$$

$$W_{\text{tamp}}(i, L(i)+2) = \beta \cdot W_{\text{tamp}}(i, L(i)+1)$$

2 度目の再送後にタイムアウトが発生すると、ウィンドウサイズは両コネクション共に 1 パケットに設定され、 $ssthresh$  は次のように更新される。

$$S_{\text{Reno}}(i+1) = \frac{W_{\text{Reno}}(i, L(i)+2)}{2}$$

$$S_{\text{stamp}}(i+1) = \beta \cdot W_{\text{tamp}}(i, L(i)+2)$$

となる。ここで、パケット廃棄発生時の tampered-TCP のウィンドウサイズに関してさらに詳しく見ていく。我々の調査から、周期  $i$  中の tampered-TCP のパケット廃棄数  $D_{\text{tamp}}(i)$  の平均値は、次の式で表されることがわかっている。

$$D_{\text{tamp}}(i) = \alpha$$

よって、 $\alpha < 3$  の場合にはタイムアウトが発生せず、 $\alpha \geq 3$  の場合にはタイムアウトが発生する。以下では、この 2 つの場合について、tampered-TCP のウィンドウサイズを導出する。

#### a) タイムアウトが発生しない場合

tampered-TCP コネクションにおいてタイムアウトが発生しない場合、先述のように  $\alpha < 3$  となる。この場合、周期  $i$  はスロースタートフェーズではなく、輻輳回避フェーズから始まる。周期  $(i-1)$  において、tampered-TCP コネクションでは  $D_{\text{tamp}}(i-1) = \alpha$  個のパケット廃棄が発生するため、 $\alpha$  回のパケット再送が行われる。よって、周期  $i$  開始時のウィンドウサイズ  $W_{\text{tamp}}(i, j)$  は、次の式で表すことができる。

$$W_{\text{tamp}}(i, 1) = \beta^\alpha W_{\text{tamp}}(i-1, L(i-1))$$

となる。式 (2) より、tampered-TCP は 1 RTT 毎に  $\alpha$  ずつウィンドウサイズを増加させるため、周期  $i$  の  $j$  番目の RTT におけるウィンドウサイズ  $W_{\text{tamp}}(i, j)$  は次のように表すことができる。

$$W_{\text{tamp}}(i, j) = \beta^\alpha W_{\text{tamp}}(i-1, L(i-1)) + \alpha \cdot j$$

#### b) タイムアウトが発生する場合

tampered-TCP コネクションにおいてタイムアウトが発生する場合、先述のように  $\alpha \geq 3$  となる。この場合、周期  $(i-1)$  にてタイムアウトが発生しているため、周期  $i$  はスロースタートフェーズで始まる。周期  $(i-1)$  において、tampered-TCP コネクションでは  $D_{\text{tamp}}(i-1) = \alpha$  個のパケット廃棄が発生し、2 回の再送の後タイムアウトが発生する。よって、 $ssthresh$  の値は次の式で表される。

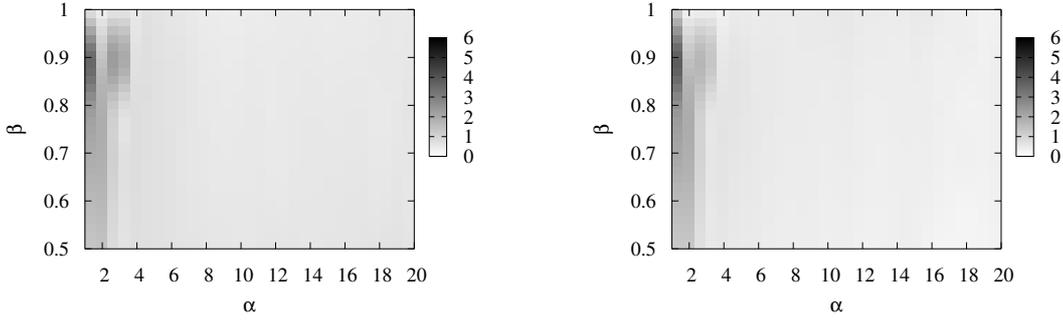
$$S_{\text{stamp}}(i+1) = \beta \cdot W_{\text{tamp}}(i, L(i)+2)$$

また、周期  $i$  開始時点のウィンドウサイズは 1 となる。

周期  $i$  のスロースタートフェーズが  $ss(i)$  番目の RTT で終了するとすると、周期  $i$  の  $j$  番目の RTT のウィンドウサイズ  $W_{\text{tamp}}(i, j)$  は、次の式で表すことができる。

$$W_{\text{tamp}}(i, j) = \begin{cases} 2^j & \text{(if } j < ss(i)) \\ 2^{ss(i)} + \alpha \cdot (j - ss(i)) & \text{(if } j \geq ss(i)) \end{cases} \quad (4)$$

ここで、スロースタートフェーズから輻輳回避フェーズに移行



(a) 解析結果

(b) シミュレーション結果

図 3 各 TCP コネクションが 1 本ずつの場合の解析結果とシミュレーション結果の比較

する瞬間である  $ss(i)$  番目の RTT に着目すると、次の関係が成り立つ。

$$2^{ss(i)} \geq \beta^2 \cdot W_{\text{tamp}}(i-1, L(i-1))$$

この式を変形すると、次式が得られる。

$$ss(i) \geq \log_2(\beta^2 \cdot W_{\text{tamp}}(i-1, L(i-1))) \quad (5)$$

ここで、式 (5) にて等号が成立すると仮定する。すると、tampered-TCP の輻輳回避フェーズに関して、式 (4) を用いて次の式を導くことができる。

$$W_{\text{tamp}}(i, j) = \beta^2 W_{\text{tamp}}(i-1, L(i-1)) + \alpha [j - \log_2\{\beta^2 \cdot W_{\text{tamp}}(i-1, L(i-1))\}]$$

ここから、tampered-TCP の輻輳回避フェーズにおけるウィンドウサイズは次のように表すことができる。

$$W_{\text{tamp}}(i, j) = \begin{cases} \beta^\alpha W_{\text{tamp}}(i-1, L(i-1)) + \alpha \cdot j & (\text{if } \alpha < 3) \\ \beta^2 W_{\text{tamp}}(i-1, L(i-1)) & \\ + \alpha [j - \log_2\{\beta^2 W_{\text{tamp}}(i-1, L(i-1))\}] & (\text{if } \alpha \geq 3) \end{cases} \quad (6)$$

次に、これまでの解析により求めたウィンドウサイズを用いて、TCP Reno コネクションと tampered-TCP コネクションの平均スループットを導出する。周期  $i$  の  $j$  番目の RTT における、ボトルネックルータ  $R_A$  のキュー長は  $\max((W_{\text{Reno}}(i, j) + W_{\text{tamp}}(i, j) - 2\tau\mu), 0)$  と表される。この時、ルータ  $R_A$  におけるパuffァのキューイング遅延を  $Q(i, j)$  とすると、 $Q(i, j)$  は次の式で表される。

$$Q(i, j) = \frac{\max((W_{\text{Reno}}(i, j) + W_{\text{tamp}}(i, j) - 2\tau\mu), 0)}{\mu}$$

よって、TCP Reno の平均スループット  $\rho_{\text{Reno}}$ 、tampered-TCP の平均スループット  $\rho_{\text{tamp}}$  は、それぞれ次の式で表される。

$$\rho_{\text{Reno}} = \frac{\sum_{i=1}^{\infty} \sum_{j=1}^{L(i)} W_{\text{Reno}}(i, j)}{\sum_{i=1}^{\infty} \sum_{j=1}^{L(i)} (Q(i, j) + 2\tau)}$$

$$\rho_{\text{tamp}} = \frac{\sum_{i=1}^{\infty} \sum_{j=1}^{L(i)} W_{\text{tamp}}(i, j)}{\sum_{i=1}^{\infty} \sum_{j=1}^{L(i)} (Q(i, j) + 2\tau)}$$

TCP Reno、tampered-TCP の平均スループットは、 $\rho_{\text{Reno}}$  と  $\rho_{\text{tamp}}$  がある値に収束する  $i$  まで、上式を計算することによって求められる。

#### 4. シミュレーション評価

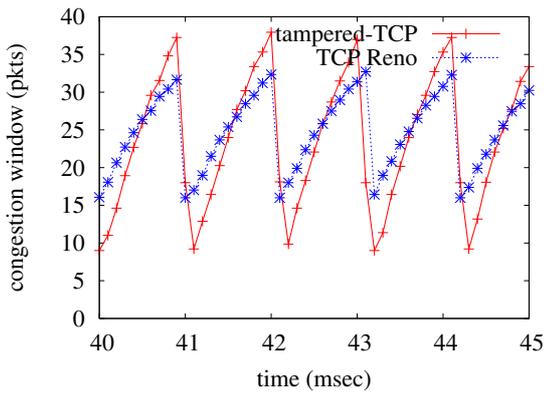
本章では、前章で述べた解析の妥当性を、解析結果とシミュレーション結果を比較することによって評価し、tampered-TCP の有効性に関する議論を行う。シミュレーションには図 1 に示すネットワークモデルを用い、 $\mu_{\text{Reno}} = \mu_{\text{tamp}} = 100$  Mbps、 $\mu = 10$  Mbps、送信側/受信側ホストとルータを結ぶリンクの伝搬遅延時間を 5 msec、ルータ間を結ぶ伝搬遅延時間を 10 msec、パケット長を 1500 byte、バッファサイズ  $B$  を 33 packets とした。また、tampered-TCP の輻輳回避ウィンドウサイズの上げ幅  $\alpha$  を 1~20 の間で、同時に下げ幅  $\beta$  を 0.5~1 の間で変化させ、60 秒間のシミュレーションを行った。図 3 は、 $\alpha$  および  $\beta$  の値に対する、スループット比の変化を示したグラフである。図には、3.2 節で示した解析結果、およびシミュレーション結果を示している。

この図から、前章で述べた解析が、tampered-TCP と TCP Reno のスループット比をほぼ正確に導出していることがわかる。また、ほとんどの  $\alpha, \beta$  に対してスループット比が 1 以下となっており、tampered-TCP が効果を発揮していないことがわかる。このような結果が得られる理由として、以下の 3 点が挙げられる。

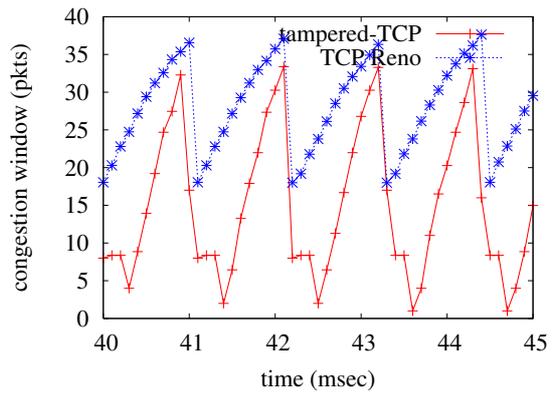
まず [8] で述べられている、1 ウィンドウ中に 3 個以上のパケット廃棄が発生すると TCP はタイムアウトを起こすという性質が挙げられる。この性質は、シミュレーション結果においても確認できる。図 4(a) は、 $\alpha = 2, \beta = 0.5$  の場合の、各 TCP コネクションのウィンドウサイズの変動を示したシミュレーション結果である。この場合、tampered-TCP コネクションでは、1 ウィンドウ中に 2 個のパケット廃棄が発生するが、タイムアウトは全く発生していないことがわかる。一方、図 4(b) は、 $\alpha = 3, \beta = 0.5$  の場合の、各 TCP コネクションのウィンドウサイズの変動を示したシミュレーション結果である。この場合、tampered-TCP コネクションでは、1 ウィンドウ中に 3 個のパケット廃棄が発生するため、tampered-TCP コネクションでタイムアウトが頻発していることがわかる。

また、図 5, 6 は、それぞれ  $\beta$  を 0.5, 0.9 に設定した時の、 $\alpha$  の値に対する、各 TCP コネクションのスループットの変動を示している。これらの図から、 $\alpha \geq 3$  の場合では、 $\alpha$  の増加にしたがって tampered-TCP コネクションのスループットが小さくなることがわかる。これは、前述のように  $\alpha \geq 3$  の場合にタイムアウトが発生することが原因である。また、 $\alpha$  が大きくなると、tampered-TCP の輻輳回避ウィンドウサイズの増加速度が早くなるため、パケット廃棄の発生頻度が高くなり、タイムアウトも増加してスループットが低下することも原因として挙げられる。

さらに、図 5, 6 の  $\alpha < 3$  の場合に注目すると、 $\beta$  が大きくなって tampered-TCP コネクションの輻輳回避ウィンドウサイズの下げ幅が小さくなることで、スループットが大きくなっ

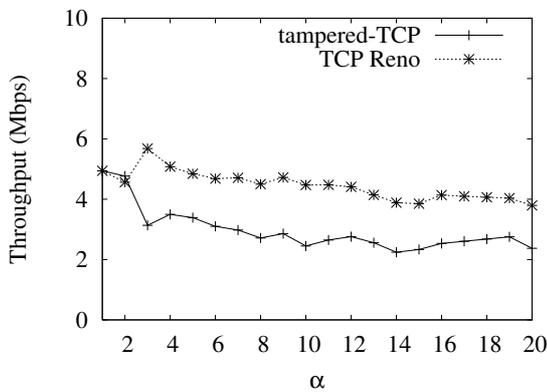


(a)  $\alpha = 2$  の場合

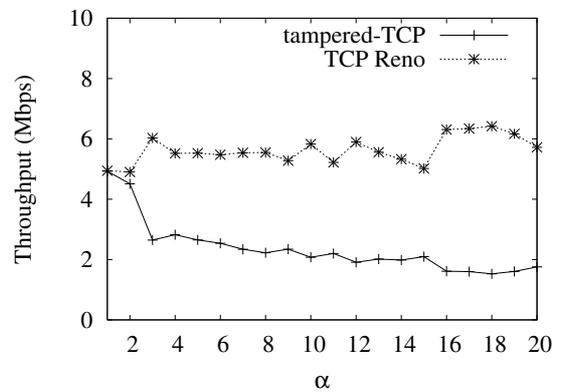


(b)  $\alpha = 3$  の場合

図 4 シミュレーションにおける各 TCP コネクションのウィンドウサイズの変動

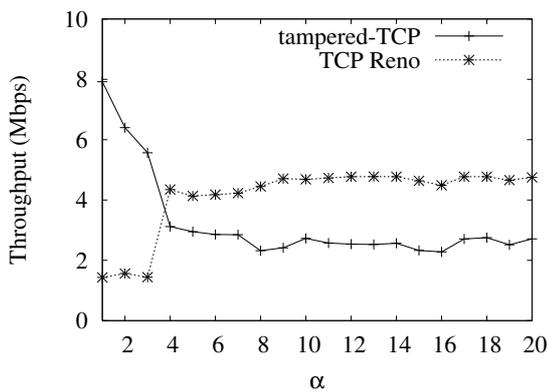


(a) 解析結果

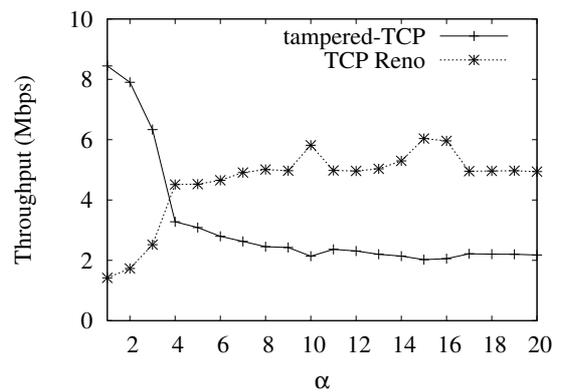


(b) シミュレーション結果

図 5  $\alpha$  を変化させた場合の解析とシミュレーションのスループットの变化 ( $\beta = 0.5$ )



(a) 解析結果

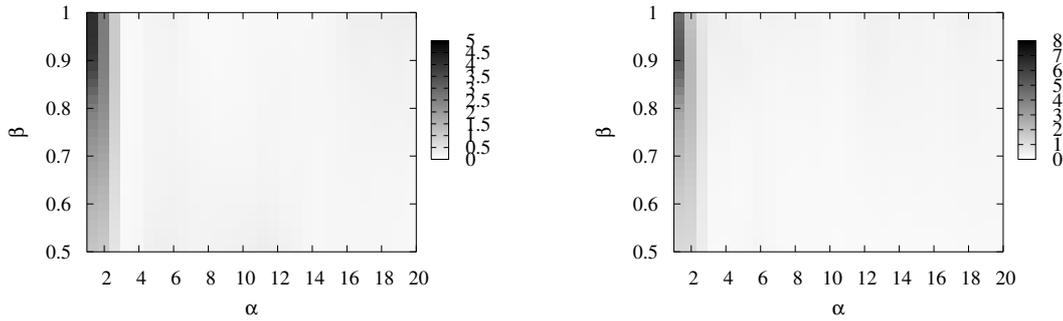


(b) シミュレーション結果

図 6  $\alpha$  を変化させた場合の解析とシミュレーションのスループットの变化 ( $\beta = 0.9$ )

ていることがわかる。これは、前述のように  $\alpha < 3$  の場合は tampered-TCP コネクションでタイムアウトが発生しないため、 $\beta$  が大きくなってウィンドウサイズの減少量が少なくなると、その分 tampered-TCP コネクションが獲得するスループットが大きくなるためである。しかし、 $\alpha \geq 3$  の場合になると、

$\beta$  が大きくなってもスループットは増加していないことがわかる。これは、 $\beta$  が増加して tampered-TCP コネクションの輻射ウィンドウサイズの減少幅が低下する効果よりも、 $\alpha \geq 3$  であるためにタイムアウトが発生し、急激にスループットが低下することの影響の方が強いためである。



(a) 解析結果

(b) シミュレーション結果

図 7 TCP Reno コネクションを複数本にした場合の解析結果とシミュレーション結果の比較

## 5. TCP Reno コネクションを増加させた場合の tampered-TCP の性能評価

前章までは、TCP Reno コネクションおよび tampered-TCP コネクションがそれぞれネットワーク中に 1 本ずつ存在する場合を考えた。しかし、実際のネットワークでは、多数の TCP Reno コネクションの中に、少数の tampered-TCP コネクションが存在する環境が一般的であると考えられる。そこで、本章では、TCP Reno コネクションが複数本、tampered-TCP コネクションが 1 本存在する場合の評価を行う。

### 5.1 スループット解析

3.2 節では、TCP Reno コネクション、tampered-TCP コネクションがそれぞれ 1 本ずつ存在する環境での数学的解析手法について述べた。ここでは、それを TCP Reno が  $n_{\text{Reno}}$  本存在する環境に拡張する。

基本的な解析は 3.2 節の解析とほぼ同じである。異なる点としては、TCP Reno コネクションが増加することで、パケット廃棄が発生するタイミングが変化することが挙げられる。具体的には、(3) 式を、以下のように書き換える。

$$n_{\text{Reno}}W'_{\text{Reno}}(i-1) + n_{\text{Reno}}L(i) + W'_{\text{tamp}}(i-1) + \alpha L(i) > 2\tau\mu + B$$

この式から、 $L(i)$  を下のように表すことができる。

$$L(i) = \frac{(2\tau\mu + B) - n_{\text{Reno}}W'_{\text{Reno}}(i-1) + W'_{\text{tamp}}(i-1)}{\alpha + n} \quad (7)$$

### 5.2 シミュレーション評価

シミュレーションに用いるネットワークモデル、および各パラメータの値は 3.2 節と同様とし、 $n_{\text{Reno}} = 5$  とした。図 7 は、tampered-TCP の  $\alpha$ 、 $\beta$  の値に対するスループット比の変化を示したグラフである。この図から、TCP Reno コネクションが複数本の場合においても、解析結果が tampered-TCP と TCP Reno のスループット比をほぼ正確に導出していることがわかる。また、TCP Reno コネクションが増加した環境においても、TCP Reno コネクションが 1 本の環境と同様、tampered-TCP が広いパラメータ領域において効果を発揮していないことがわかる。これは、前章での議論に加え、式 (7) からわかるように、コネクション数が増加することでパケット廃棄およびタイムアウトが発生するまでの RTT 数が減少し、その結果スループットが低下するためである。

## 6. おわりに

本稿では、TCP Reno のウィンドウサイズの上げ幅および下げ幅を変更した tampered-TCP の有効性に関する評価を行った。その結果、tampered-TCP は  $\alpha \geq 3$  になるとタイムアウトを起こしてスループットが激減すること、 $\alpha \geq 3$  の場合では

$\alpha$  の増加にしたがってパケット廃棄の発生頻度が高くなり、タイムアウトが原因となりスループットが低下すること、および  $\alpha < 3$  かつ、 $\beta$  が大きい場合にのみ tampered-TCP の効果が現れること、下げ幅を小さくすることの効果より、上げ幅の増加によるスループットの低下の影響の方が強いことが明らかとなった。また、TCP Reno コネクションが複数本存在する環境において、tampered-TCP コネクションの効果は変化しないことがわかった。

今後の課題としては、tampered-TCP の SACK オプションを ON にした場合の評価が挙げられる。また、インターネットを用いた実験を行い、本稿で得られた結果が実ネットワークにおいても成り立つことを確認したい。さらに、他の種類の tampered-TCP について評価を行うことも検討したい。

### 文 献

- [1] S. Savage, N. Cardwell, D. Wetherall, and T. Anderson, "TCP congestion control with a misbehaving receiver," *ACM Computer Communications Review*, 29(5):71-78, Oct. 1999.
- [2] S. Floyd and K. Fall, "Router mechanisms to support end-to-end congestion control," tech. rep., Network Research Group at LBNL, Feb. 1997.
- [3] M. Baldi, Y. Ofek, and M. Yung, "Idiosyncratic signatures for authenticated execution of management code," in *14th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management (DSOM 2003)*, Heidelberg, Germany, Oct. 2003.
- [4] Y. R. Yang and S. S. Lam, "General AIMD congestion control," in *Proceedings of the IEEE International Conference on Network Protocols*, Nov. 2000.
- [5] L. Mamatas and V. Tsaoussidis, "Protocol behavior: More effort, more gains?," *Personal, Indoor and Mobile Radio Communications, 2004. PIMRC 2004. 15th IEEE International Symposium*, 125-129 Vol.1, Sept. 2004.
- [6] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP throughput: A Simple Model and its Empirical Validation," in *Proceedings of ACM SIGCOMM '98, Vancouver, B.C.*, Sept. 1998.
- [7] K. Tokuda, G. Hasegawa, and M. Murata, "Performance analysis of HighSpeed TCP and its improvement for high throughput and fairness against TCP Reno connections," in *Proceedings of IEEE High Speed Network Workshop 2003 (HSN '03)*, (San Francisco), Mar. 2003.
- [8] K. Fall and S. Floyd, "Simulation-based Comparisons of Tahoe, Reno, and SACK TCP," *Computer Communication Review*, vol. 26, pp. 5-21, July 1996.