

# Equation 拘束に基づく TCP Westwood の公平性の改善

## Fairness Improvement of TCP Westwood based on Equation Constraint

兼子 和巳  
Kazumi KANEKO

甲藤 二郎  
Jiro KATTO

早稲田大学 理工学部 電子・情報通信学科  
Department of Electronics, Information and Communication Engineering, Waseda University

### 1. はじめに

TCP として Reno ベースのバージョンが広く用いられているが、現在も TCP の特性を改善する様々な研究が行われている。それらの研究において、Reno との親和性は重要となる。近年提案された TCP Westwood(TCP W)でも同様の評価、改良が進められているが、ネットワークの状況によっては、その親和性は劣化する。本稿では、帯域推定手法 RCE[3]を用いてボトルネックルータのバッファサイズを推定し、Equation ベースの推定レート[1][2]が等しくなる条件下でパラメータを再設定することで、Reno との公平性を実現する TCP W の改良方式を提案する。

### 2. 従来手法とその問題点

TCP W では、より適切な End-to-End での輻輳制御を行うために、TCP の ACK パケットの到着間隔を利用し、推定した公平な利用可能帯域(FSE)をもとに、スロースタート閾値(ssthresh)を  $FSE \cdot RTT_{min}$  に設定する。RTTmin は最小の RTT である。しかし、TCP W では、ssthresh の設定に RTT の変動を考慮していない。このため、ボトルネックルータのバッファサイズによって性能が左右され、バッファサイズが小さい場合は Reno フローを追い出し、大きい場合は送信レートを下げてしまう現象が見られる。

### 3. 提案手法

#### バッファサイズ推定(BS\_estimate)

ボトルネックルータのバッファサイズを RTT と RTTmin の差分から推定する。この時、ボトルネックリンク帯域 B の推定には RCE[3]の手法を用いる。

$$BS\_estimate = \frac{(RTT - RTT_{min}) \times B}{Packet\_Size \times 8} \quad (1)$$

#### 公平となるバッファサイズ推定(BS\_fair)

Reno のレスポンス機能は、

$$S_{Reno} = \sqrt{\frac{2(1-p)}{p}} \quad [Packets/RTT] \quad (2)$$

で与えられる[1]。ただし、p はロス率である。同様に Westwood のレスポンス機能は、

$$S_{West} = \frac{\sqrt{RTT}}{\sqrt{T_q}} \sqrt{\frac{(1-p)}{p}} \quad [Packets/RTT] \quad (3)$$

で与えられる[2]。ただし、 $T_q = RTT - RTT_{min}$  であり、 $T_q$  はバッファ遅延を示す。ここで、 $T_q$  を  $BS\_fair$ (公平となるバッファサイズ)を用いて表す。

$$T_q = \frac{BS\_fair \times packet\_Size \times 8}{B} \quad (4)$$

(2)式と(3)式が等しい場合に、Reno と TCP W は公平ということになる。これは、

$$BS\_fair = \frac{B \times RTT_{min}}{Packet\_Size \times 8} \quad (5)$$

とした場合に対応する。

#### fair\_RTtmin の設定

$BS\_estimate$  と  $BS\_fair$  とのバッファ遅延差(diff\_delay)を算出し、(3)式の第 1 項に代入する。

$$fair\_response = \frac{\sqrt{(RTT - diff\_delay)}}{\sqrt{(RTT - diff\_delay) - RTT_{min}}} \quad (6)$$

よって TCP W のレスポンス機能の第 1 項を  $fair\_response$  に更新することにより、バッファサイズの大きさに関わらず、Reno と TCP W は公平となる。そこで、(3)式の第 1 項と  $fair\_response$  を等しくするために、(3)式の RTTmin に反映させることにより、 $fair\_RTTmin$  を設定する。

$$fair\_RTT_{min} = RTT - \frac{RTT}{fair\_response} \quad (7)$$

この値を利用して、ssthresh を  $FSE \cdot fair\_RTTmin$  とする。

### 4. シミュレーション結果

シミュレーションは ns2 を用いた。

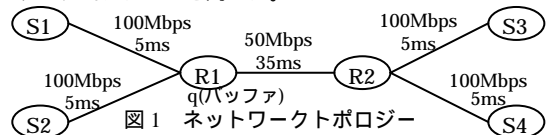


図 1 のネットワークポロジーを利用し、S1-S3 に Reno のフロー、S2-S4 に Westwood(従来 or 提案)のフローを流し、R1-R2 をボトルネックリンクとして、q のバッファサイズを 100 ~ 1000 まで変化させた。バッファでのパケット廃棄制御は Drop-Tail とする。以下に、Reno/Westwood の両方が定常状態にある場合の平均スループットの比率を示す。

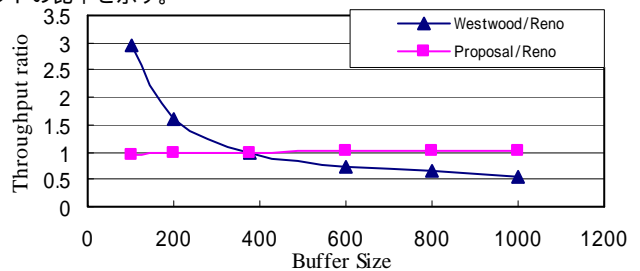


図 2 Throughput ratio

### 5. おわりに

本稿では、ボトルネックルータにおいて、TCP W と Reno が公平となるバッファサイズを定量的に示し、RTTmin に反映させることにより親和性を実現した。今後の展開として、提案手法の TFRC への応用や Vegas への適用も考える。

### 6. 参考文献

[1] Kelly, F. **Mathematical modeling of the Internet**, in Proceedings of the Fourth International Congress on Industrial and Applied Mathematics, (July 1999).  
 [2] L. A. Grieco and S. Mascolo, "TCP Westwood and Easy RED to Improve Fairness in High-Speed Networks", Seventh International Workshop on Protocols For High-Speed Networks (PHSN'2002), April 22 - 24, 2002 Berlin, Germany  
 [3] Claudio E. Palazzi "Residual Capacity Estimator for TCP on Wired/Wireless Links" WCC August 2004