

# 動的パケット破棄による TCP 公平性向上と 多種コネクション環境における評価

秋山友理愛<sup>†</sup> 神津智樹<sup>†</sup> 山口実靖<sup>†</sup>

工学院大学大学院工学研究科電気・電子工学専攻<sup>†</sup>

## 1. はじめに

高遅延ネットワークで高い通信性能を提供できる TCP 輻輳制御アルゴリズムとして CUBIC TCP[1]などの複数の高速 TCP が提案され、活用されている。これらの複数の高速 TCP の提案により、TCP 輻輳制御アルゴリズム間の公平性という新しい課題が生じ、公平性に関する研究が行われてきている。ネットワーク上で動的にパケットを優先破棄し公平性を向上させる手法として、動的パケット優先破棄[2]がある。その有効性は[2][3]で示されているがこれらはいずれもすべてのコネクションの RTT が同一の環境で行われたものであり、RTT が異なるコネクションが混在する環境における有効性が確認されていない。

本稿では、代表的な高速 TCP として CUBIC TCP と Compound TCP[4]を対象に、動的パケット優先破棄を RTT の異なるコネクションが混在する環境で用いた場合の TCP 公平性について評価する。

## 2. RED

RED[5]は待ち行列長に応じた確率でパケットの廃棄を行う方法である。ネットワーク上のルータにおいて RED を用いることにより通信性能やネットワーク負荷の安定化、コネクション間の公平性の改善が実現されると期待されている。

RED では、図 1 の様に平均待ち行列長によりパケット破棄率を決定し、その確率によりパケットが廃棄される。 $min_{th}$ ,  $max_{th}$ ,  $max_p$  は制御用パラメータである。

平均待ち行列長は  $w_q$  を用いて以下に式により決定される。

$$\bar{q} = (1 - w_q)\bar{q} + w_q q$$

## 3. パケット優先破棄手法

本章で、ネットワーク利用率の高いコネクションのパケットを優先的に破棄し、既存の RED より高い精度の公平性を目指す手法として動的優先破棄手法を紹介する。

動的優先破棄手法[2]では、ルータが最も通信帯域を消費しているコネクションの推測を行い、そのコネクションのパケットの破棄率を  $n$  倍にして優先的に破棄する。

最も通信帯域を消費しているコネクションの推定は以下の様に行う。

ルータに到着するパケットの数が最も多いコネクションを、最も通信帯域の消費が大きいコネクションと仮定する。ルータにて記録周期  $rec\_int$  毎に到着 TCP パケットのコネクション情報を記録し、最新の  $hist\_len$  個のコネクション情報を履歴としてルータ内のメモリに保持する。

TCP Fairness Improvement by Active Packet Dropping  
Yuria AKIYAMA<sup>†</sup>, Tomoki KOZU<sup>†</sup>, Saneyasu YAMAGUCHI<sup>†</sup>

<sup>†</sup>Electrical Engineering and Electronics, Kogakuin University Graduate School

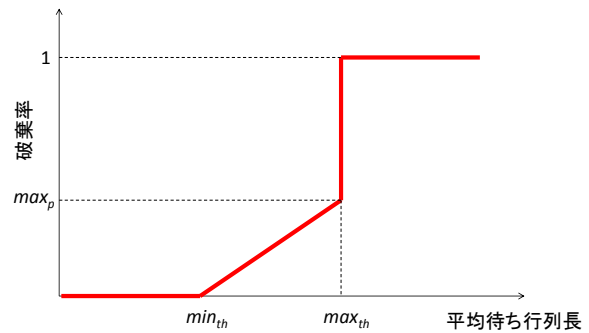


図 1 RED

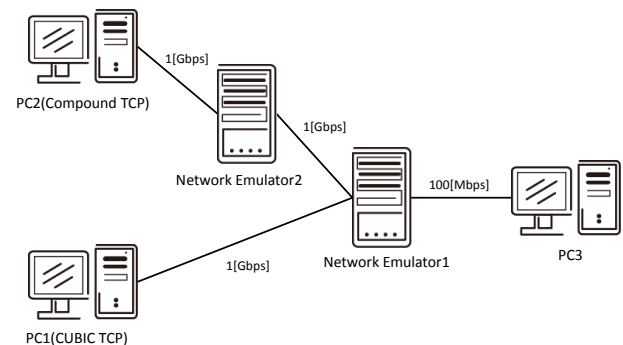


図 2 ネットワーク構成

そして、集計周期  $sta\_int$  毎に履歴中に最も多く登場するコネクションを抽出し、それを最も通信帯域を消費しているコネクションとする。

コネクション情報としては、送信元 IP アドレス、送信先 IP アドレス、送信元ポート番号、送信先ポート番号を保持する。

## 4. 評価

図 2 のネットワークを構築し、CUBIC TCP と Compound TCP が混在する環境における通信速度を  $netperf$  を用いて測定した。ネットワーク機器は全て 1Gigabit Ethernet 対応のものであり、Network Emulator1 と PC3 の間の通信速度はエミュレータにより 100Mbps に設定されている。PC1 では Linux 2.6.35.6(CUBIC-TCP)が動作し、PC2 では Windows7(Compound TCP)が動作している。PC1-PC3 の間と PC2-PC3 の間で  $netperf$  の接続を確立し、同時通信時の通信速度を測定した。両接続は Network Emulator1 から PC3 までを共有している。Network Emulator1 において片道遅延時間を 2[ms]、Network Emulator2 において片道遅延時間を 30[ms]設定し、PC1-PC3 間の RTT を 4[ms]、PC2-PC3 間の RTT を 64[ms]

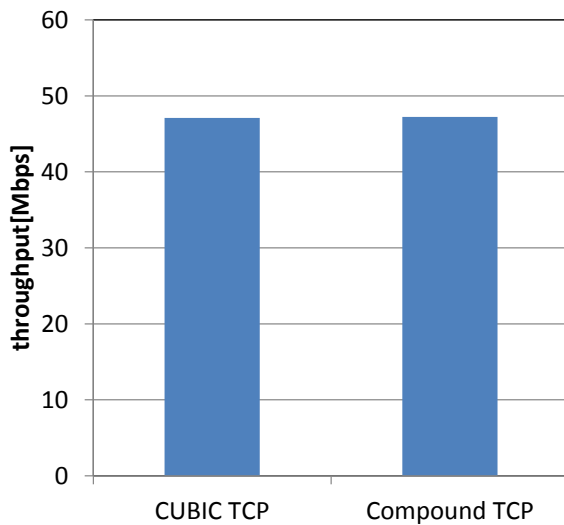


図3 単独通信での通信速度

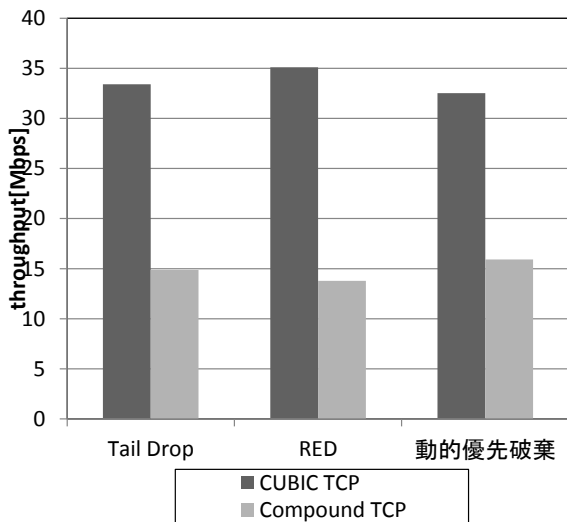


図4 競合通信での通信速度

とした。

パケット廃棄は Network Emulator1 にて行われ、通常のパケット廃棄(Tail Drop)、RED、動的優先廃棄手法により行った。全ての手法において、最大待ち行列長は 10,000 とし、RED の各パラメータを  $min_{th}=100$ 、 $max_{th}=9999$ 、 $max_p=0.5$ 、 $w_q=0.002$  としてパケット破棄を行った。

図3は両TCPが競合していない状態における平均通信速度である。測定は2本の netperf コネクションを確立し、Tail Dropを用いて行った。

CUBIC TCP の例では、PC1-PC3 間に CUBIC TCP コネクションを2本確立し、それぞれの速度を測定している。

図より CUBIC TCP と Compound TCP が競合しない状態では、各 TCP で得られる平均速度に大きな差がないことが分かる。

また、2本のコネクションの合計通信速度は CUBIC TCP が 94.22[Mbps]、Compound TCP が 94.47[Mbps]であり、通信帯域の 100[Mbps]をほぼ使い切っていることも分かる。

図4は CUBIC TCP と Compound TCP のコネクションをそれぞれ2本ずつ合計4本を同時に確立し、競合で通信を行った時の各 TCP の平均通信速度である。Tail Drop 手法における CUBIC TCP と Compound TCP の通信速度をみると、競合環境で通信を行った場合、CUBIC TCP が Compound TCP の帯域を圧迫し両 TCP の通信速度に大きな不公平が生じることが分かる。また、RED を用いても公平度が改善されないことが分かる。これに対して動的優先破棄手法では不公平を軽減できていることが分かる。ただし、改善後も小さくない不公平が生じており、さらなる改善も望まれる。

## 5. おわりに

本研究では、高速 TCP アルゴリズム間の帯域公平性に着目し、RTT が異なるコネクションが混在する環境における TCP 公平性の改善を評価した。評価の結果、動的優先破棄手法を用いることにより公平性の改善を行えることが確認された。今後は動的優先破棄手法のさらなる改良、1000本以上の TCP コネクションが確立されたより複雑な環境における評価などを行っていく予定である。

## 謝辞

本研究は JSPS 科研費 24300034, 25280022, 26730040 の助成を受けたものである。

## 参考文献

- [1] Injong Rhee and Lisong Xu "CUBIC: A New TCP-Friendly High-Speed TCP Variant," Proc. Workshop on Protocols for Fast Long Distance Networks, 2005, 2005.
- [2] 秋山友理愛, 大浦亮, 神津智樹, 山口実靖 “実機と実 TCP 実装を用いた TCP 公平性の評価”, 電子情報通信学会ネットワークシステム研究会, Sep. 2012
- [3] 秋山友理愛, 神津智樹, 山口実靖 “多コネクション環境における動的パケット優先破棄による TCP 公平性の改善”, 情報処理学会 第76回全国大会, March 2014
- [4] Kun Tan, Jingmin Song, Qian Zhang, and Murari Sridharan, "A Compound TCP Approach for High-speed and Long Distance Networks" Proc. IEEE Info COM 2005, July 2005.
- [5] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," IEEE/ACM Transactions on Networking, vol. 1, pp. 397-413, Aug. 1993.