

実ネットワークにおける CUBIC TCP の RTT 公平性

神津 智樹[†] 秋山 友理愛[†] 山口 実靖[‡][†]工学院大学大学院工学研究科電気・電子工学専攻 [‡]工学院大学工学部情報通信工学科

1. はじめに

TCP は現在のインターネットにおいて標準的に用いられているトランスポート層プロトコルである。高遅延環境下での通信性能には、OS の TCP 実装の動作が大きな影響を与える。多くの OS に実装されてきた古典的な TCP である TCP Reno ではネットワーク帯域を十分に使いきれないという問題が指摘されており [1], Linux OS には高遅延環境でも高い性能を提供できる CUBIC TCP [2] が実装されている。

我々はこれまでに CUBIC TCP の RTT 公平性向上手法を提案し [3], CUBIC TCP 間における RTT 公平性の向上を実現してきた。しかし、実ネットワークを用いての評価は十分に行われていない。本稿では、実ネットワークにおける、CUBIC TCP の RTT 公平性や提案手法の有効性についての評価を行う。

2. 関連研究

2.1 輻輳制御アルゴリズム

TCP はネットワークの輻輳状況に応じて輻輳ウィンドウサイズを動的に増減させ、ネットワークへ送出するデータ量を調節することにより輻輳制御を行う。輻輳制御手法には、複数のアルゴリズムがあり、多くのサーバ OS で利用されている Linux OS では CUBIC TCP [2] と呼ばれる手法が用いられている。

2.2 CUBIC TCP

CUBIC TCP は、BIC TCP のスケーラビリティを維持しながら、TCP Fairness と RTT Fairness, 制御手法の複雑さを改善した高速 TCP である。

CUBIC TCP では、BIC TCP のバイナリサーチを用いて、利用可能帯域を検索するアルゴリズムを次の式のような 3 次関数を用いた制御によって実現している。

$$cwnd = c(t - K)^3 + W_{max} \quad (1)$$

$$K = \sqrt[3]{\frac{W_{max}\beta}{C}} \quad (2)$$

ここで、 $cwnd$ は輻輳ウィンドウサイズ、 t はパケットロス検出時からの経過時間、 W_{max} はパケットロス検出時の輻輳ウィンドウサイズ、 C は増加幅を決めるパラメータ、 β はパケットロス検出時のウィンドウサイズ減少幅を表している。一般に C は 0.4、 β には 0.2 が用いられる。 K は、式 (1) における 3 次関数の変曲点までの時間であり、 $t=K$ において、輻輳ウィンドウサイズが前回のパケットロス検出時の値と等しくなる。よって K が小さいほど短い時間で輻輳ウィンドウサイズが回復することとなり、高い性能が得られやすくなる。一般に RTT が大きい通信ほど、輻輳ウィンドウサイズの値が性能に影響を与え、より大きな輻輳ウィンドウサイズが必要となる。しかし、式 (2) より、前回のパケットロス検出時の輻輳ウィンドウサイズが大きいほど K の値が大きくなり、高い性能が得られづらいたことが分かる。よって、RTT が大きい通信ほど輻輳ウィンドウサイズが大きくなり、結果 K の値が大きくなり通信性能が低くなりやすいと予想される。これは、RTT 公平性を低下させる原因になると考えられる。以下、この手法を標準 CUBIC と呼ぶ。

2.3 CUBIC TCP の RTT 公平性改善手法

文献 [3] では、異なる RTT 同士の CUBIC TCP が競合すると、RTT の大きいコネクションのスループットが低くなり、RTT 公平性が低くなること示されている。そのため、文献 [3] では CUBIC TCP の輻輳ウィンドウ回復時間式である K を調整することにより RTT 公平性を改善する手法を提案し評価により有効性を示している。当該手法では、 K を次式のように調整し、RTT が大きいコネクションの輻輳ウィンドウが短い時間で回復できるように制御している。

$$K = \sqrt[3]{\frac{W_{max}\beta}{C}} \times \frac{1}{\sqrt[3]{RTT}} \quad (3)$$

以下、この手法を RTT 公平性改善手法と呼ぶ。

3. 実ネットワークによる RTT 公平性の評価

3.1 実験環境

図 1 のネットワークを構築し下記の測定を行い、RTT 公平性を評価した。Client PC, Server1, Server2 では Linux OS を用い、TCP は CUBIC TCP が使用されている。ネットワークエミュレータ

“RTT Fairness on CUBIC TCP in Practical Networks”†Tomoki Kozu, Yuria Akiyama, Electrical Engineering and Electronics, Kogakuin University Graduate School ‡Saneyasu Yamaguchi, Department of Information and Communications Engineering, Kogakuin University

は人工的にネットワーク遅延を発生させる装置である。

通信性能は以下の方法で計測した。2 台の PC サーバ(Server 1, Server 2)にて http サーバを稼働させ、Client PC にて wget コマンドを用いて両 PC サーバからファイルをダウンロードし、そのときのスループットを測定した。通信は Client PC-Server 1 間と Client PC-Server 2 間で同時に行っている。両接続はネットワークエミュレータから Client PC までを共有しており、実ネットワークを介して接続されている。

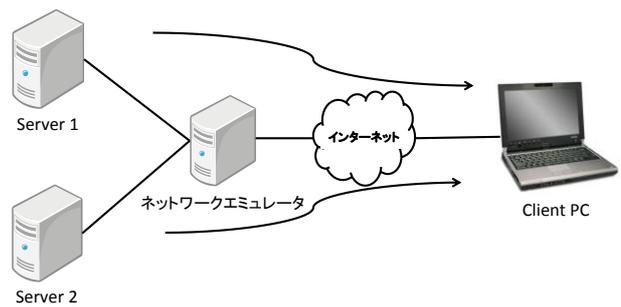


図 1 実験環境

3.2 測定結果

通信性能評価は、標準 CUBIC と RTT 公平性改善手法のそれぞれで行い、RTT 公平性改善手法は PC サーバに実装して評価を行った。ネットワークエミュレータにより、ネットワークエミュレータ-Server 1 間の RTT を 2ms に、ネットワークエミュレータ-Server 2 間の RTT を 2ms から 128ms に変動させてそれぞれで測定を行った。

測定結果を図 2 と図 3 に示す。図 2 は測定によって得られたスループットを、図 3 は Fairness Index[4]を示している。横軸はネットワークエミュレータによって設定したネットワークエミュレータ-PC サーバ間の RTT である。Fairness Index は 1 に近いほど公平性が高いことを示す。図 2 の破線は標準 CUBIC によって得られたスループット、実線が RTT 公平性改善手法によって得られたスループットである。標準 CUBIC に比べ RTT 公平性改善手法のほうが Client-Server2 間で得られたスループットが高い事がわかる。図 3 の破線は標準 CUBIC によって得られた Fairness Index, 実線が RTT 公平性改善手法によって得られた Fairness Index を示している。標準 CUBIC に比べ RTT 公平性改善手法のほうが RTT に差があるときの Fairness Index の値が 1 に近く、RTT 公平性改善手法が有効であることが分かった。

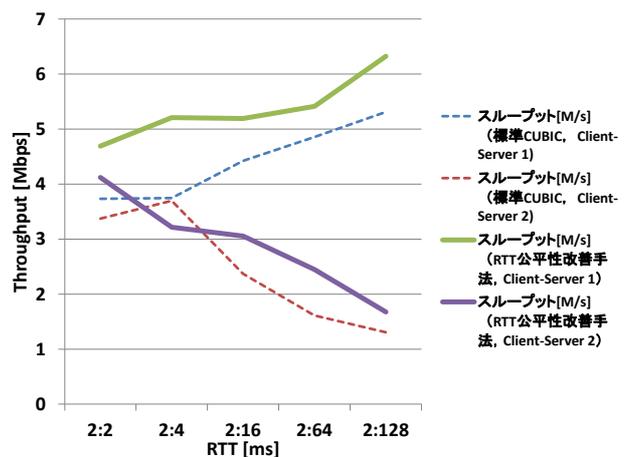


図 2 スループット

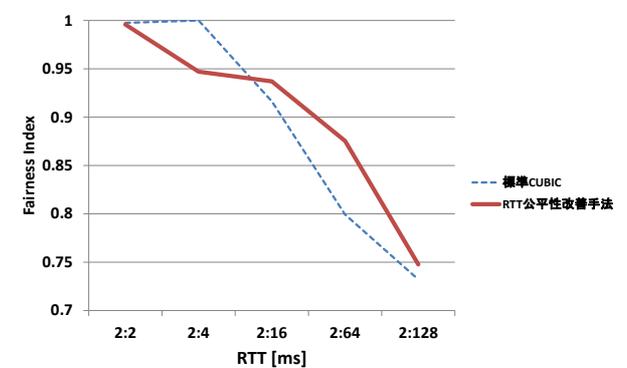


図 3 Fairness Index

4. おわりに

本研究では、先行研究である CUBIC TCP の RTT 公平性改善手法を実ネットワークを用いて評価した。その結果、RTT 公平性改善手法は実ネットワークにおいても有効であることを確認した。

今後は他の OS を用いた性能評価や、他の TCP アルゴリズム間との公平性について評価を行っていく予定である。

謝辞

本研究は一部、総務省戦略的情報通信研究開発推進事業 (SCOPE) 先進的通信アプリケーション開発推進型研究開発によるものである。

参考文献

[1] D.Katabi, M.Handley, and C.Rohr “Congestion control for high bandwidth-delay product networks”, in Proceedings of ACM SIG-COMM 2002,Aug.2002.
 [2] Injong Rhee, and Lisong Xu “CUBIC: A New TCP-Friendly High-Speed TCP Variant.” In Proc. Workshop on Protocols for Fast Long Distance Networks, 2005, 2005.
 [3]Tomoki Kozu, Yuria Akiyama, Saneyasu Yamaguchi “Improving RTT Fairness on CUBIC TCP,” International Journal of Networking and Computing, Vol 4, No 2(2014)
 [4] D. Chiu and R. Jain, “Analysis of the increase and decrease algorithms for congestion avoidance in computer networks,” Computer Networks and ISDN Systems, Volume 17, Issue 1, pp. 1-14, 1989.