1X-10

輻輳ウィンドウ回復時間の制御によるCUBIC TCPのRTT公平性の改善

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 が実装されている．

本稿では，CUBIC TCP の輻輳ウィンドウの回復時間に着目し，RTT が異なる環境における CUBIC TCP のコネクション間の公平性について考察する．

2. 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 では CUBIC TCP[2]と呼ばれる手法が用いられている．

3. 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[2]．

CUBIC TCP では，BIC TCP のバイナリーサーチを用いて，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．

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

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

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．

4. RTTを考慮したKの決定

本章でRTTを考慮してKを調整し，RTT公平性を向上させる手法を提案する．K の決定式を式(3)に示すように改善する．3 章で述べたとおり，CUBIC TCP の性能はパケットロスから輻輳ウィンドウが回復するまでの時間 K に依存しており，t=K となると輻輳ウィンドウが以前(パケットロス検出時)の値まで回復し，更に使える帯域を探し輻輳ウィンドウサイズを急激に上昇させる．K が小さいと，CUBIC TCP の輻輳ウィンドウサイズは短い時間で回復する．一般に RTT が大きい通信ほど大きな輻輳ウィンドウサイズが必要であり，式(2)よりW_{max}が大きいほど Kが大きくなることがわかる．これは，RTT が大きい通信は大きな輻輳ウィンドウサイズを必要とするが，大きな輻輳ウィンドウサイズは大きな K を招き，結果として大きな輻輳ウィンドウサイズを維持できないことを意味しており，RTT 公平性を低下させる要因であると考えられる．この不公平性を改善するには，輻輳ウィンドウサイズが大きい

"Improvement of RTT Fairness on CUBIC TCP with Controlling Congestion　Window Size Recovery Time"
†Tomoki Kozu, Yuria Akiyama, Electrical Engineering and Electronics, Kogakuin University Graduate School
‡Saneyasu Yamaguchi, Department of information and Communications Engineering, Kogakuin University

Copyright ©2014 Information Processing Society of Japan.
All Rights Reserved.

通信ほど K を短くするべきであると考えられる．そのため本提案手法では RTT が大きな通信ほど K が小さくなるように改変しており，これにより公平性が向上すると期待される．

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

5. 性能測定

5.1 実験環境

図 1 のようなネットワーク環境で下記の実験 1 および実験 2 を行い，通信性能の測定を行った．Dummynet はネットワーク遅延をエミュレーションする装置である．ネットワーク機器は全て 1Gigabit Ethernet に対応している．PC1，PC2，PC3 では Linux OS を用い，TCP は CUBIC TCP を実装している．通信は PC1-PC3 間と PC2-PC3 間で同時に行い，PC1 と PC2 を送信者，PC3 を受信者とした．PC1-PC3 間と PC2-PC3 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．両接続はネットワークエミュレータから PC3 までを共有している．ネットワークエミュレータにより PC1-PC3 間の RTT を 2ms から 64ms に，PC2-PC3 間の RTT を 2ms から 128ms に変動させて測定を行った．それぞれの受信ウィンドウサイズは 16MB に設定した．

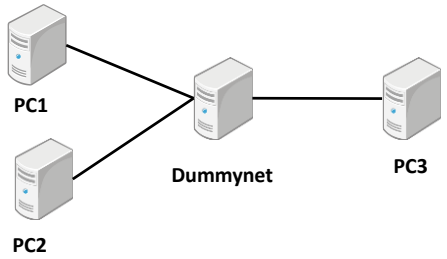

図 1　実験のネットワーク図

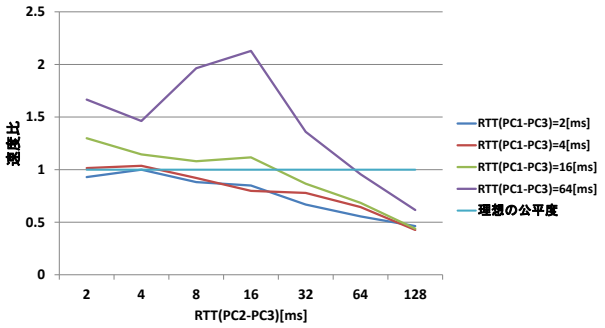

図 2　公平度

5.2 初期状態の K による性能測定

実験 1 では，式(2)の初期状態の K を用いて測定を行った．

実験結果を図 2 に示す．横軸の値はネットワークエミュレータにより設定した PC2-PC3 間の往復遅延時間である．縦軸は速度比(PC2-PC3 間の性能/PC1-PC3 間の性能)であり，1 に近いほど RTT 公平性が高いことを示している．図 2 の実験結果より，RTT 比が大きくなるほど速度比が低くなることがわかり，RTT 公平性が低いことが確認できる．

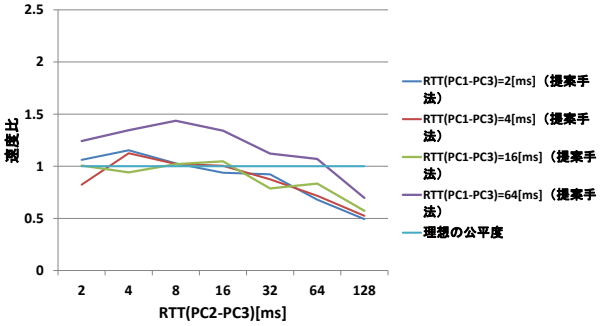

図 3　公平度（提案手法）

5.3 RTT を考慮した K による性能測定

実験 2 では，4 章の RTT を考慮した K を用いて測定を行った．

実験結果を図 3 に示す．初期状態の K より提案手法の方が速度比が 1 に近いことが分り，提案手法が有効であることが確認できる．

6. おわりに

本研究では，CUBIC TCP の RTT 公平性改善手法として，RTT を考慮した K の決定法を提案し，性能測定を行った．結果，RTT を考慮した K が有効であることを確認した．

今後は多数の TCP コネクションが確立されている環境における評価，他の TCP アルゴリズム間の公平性の評価を行っていく予定である．

謝辞

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

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]大浦亮，山口実靖，“実機を用いた高速 TCP の公平性の評価” 第 10 回情報科学技術フォーラム(FIT2011) RL-003.
[4] 鈴川 竜司，安達 直世 “CUBIC-TCP におけるフロー間帯域の公平性に対する改善手法の提案” 電子情報通信学会信学技報 NS2008-108

Copyright ©2014 Information Processing Society of Japan.
All Rights Reserved.