

実機を用いた高速 TCP の公平性の評価

An Experimental Analysis of Fairness between Modern TCP Congestion Avoidance Algorithms

大浦 亮†

山口実靖‡

Ryo Oura

Saneyasu Yamaguchi

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。従来の TCP の輻輳制御アルゴリズムである TCP-Reno では広帯域・高遅延なネットワーク環境においてネットワーク帯域を十分に使いきることができない問題が指摘されており[1], TCP-Renoに代わる高速 TCP が数多く提案されている。本稿では、これらを”高速 TCP”と呼ぶ。これらの高速 TCP に関して、多くのシミュレーションによる評価が行われているが、実機環境における評価は十分にはされていない。

本稿では、高速 TCP の中の代表的なロスベースの輻輳制御アルゴリズムとハイブリッド型輻輳制御アルゴリズムを対象に実機のネットワークを用いて性能測定、性能解析、公平性の解析を行う。そして、既存の公平性改善手法の効果について考察する。

2. 高速 TCP の輻輳制御と公平性

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。TCP 輻輳制御アルゴリズムは主に、ロスベースの手法、遅延ベースの手法、ハイブリッド型手法に分類される。

ロスベースのアルゴリズムはパケットロスを元に輻輳ウィンドウを制御する手法である。通常時は輻輳ウィンドウを増加させ、パケットロス検出時に輻輳ウィンドウを減少させる。従来の TCP である TCP-Reno がロスベースの手法であり、代表的なロスベースの高速 TCP に CUBIC-TCP[2]がある。

遅延ベースの輻輳制御アルゴリズムは、RTT の増減にあわせて輻輳ウィンドウを変化させる手法である。ロスベース手法の様に輻輳が発生してから速度を減少させるのではなく、RTT の増加からネットワークの混雑状態を推定し、輻輳が発生する前に速度を減少させるため安定した通信速度が期待できる。

欠点としてロスベース手法とネットワークを共有したときに得られる性能が低くなってしまふということが指摘されている[3]。代表的な遅延ベースの手法に TCP-Vegas[4]がある。

また、両者を組み合わせた手法ハイブリッド型手法がある。代表的なハイブリッド型手法の TCP に Compound-TCP(CTCP)[5]がある。

2.1 CUBIC-TCP

CUBIC-TCP は、BIC-TCP のスケラビリティを維持しながら、TCP Fairness(既存の TCP アルゴリズムと公平であること)、RTT Fairness(RTT の異なる通信間で公平であること)、制御手法の複雑さを改善した高速 TCP である[6]。

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

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

$$K = \sqrt[3]{\frac{W_{\max} \beta}{C}}$$

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(実時間)、 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。

上記のようなパケットロス検出時からの経過時間を用いて、ウィンドウサイズの増加から RTT の影響を排することで RTT Fairness を向上するとともに、BIC-TCP の低遅延環境で輻輳ウィンドウサイズを急速に成長させすぎる問題を解決している。また、TCP-Reno を用いた場合に得られるウィンドウサイズを次式により計算し、現在のウィンドウサイズが計算値よりも小さい場合はその計算値をウィンドウサイズとして採用することとしている。

$$cwnd = W_{\max} (1 - \beta) + 3 \frac{\beta}{2 - \beta} \frac{t}{RTT}$$

さらに、新規にネットワークに参加したフローにも公平に帯域を与えるために以下の Fast Convergence が採用されている。前回のパケットロス時のウィンドウサイズより今回のパケットロス時のウィンドウサイズが小さい場合は、 W_{\max} を次式により決定する。これにより既存のフローの W_{\max} が減少し、新規に参加したフローがより多くの帯域を得られる仕組みとなっている[2]。

$$W_{\max} = cwnd \times (2 - \beta) / 2$$

2.2 Compound TCP

CTCP は損失ベースの輻輳制御で動作する損失ウィンドウと、遅延ベースの輻輳制御で動作する遅延ウィンドウの両方を用いて、ネットワークに送出するパケット数を調節する。

損失ウィンドウは、スロースタートフェーズと輻輳回避フェーズの2つのフェーズから構成されている。それぞれのフェーズによって損失ウィンドウの増加量が異なり、各フェーズでのウィンドウサイズは、次式で与えられる。これは TCP-Reno と同等である。

$$cwnd(t+1) \leftarrow \begin{cases} cwnd(t)+1 & (\text{スロースタートフェーズ時}) \\ cwnd(t) + \frac{1}{swnd(t)} & (\text{輻輳回避フェーズ時}) \end{cases}$$

† 工学院大学大学院工学研究科電気・電子専攻
Graduate School of Electrical and Electronics
Engineering, Kogakuin University

‡ 工学院大学情報通信工学科 Department of Information
of Communications Engineering, Kogakuin University

ただし, $swnd$ は現在の損失ウィンドウと遅延ウィンドウの和であり, t は ACK を受信するごとに増加する値であり実時間とは比例しない. スロースタートフェーズでは, 損失ウィンドウを指数関数的に増加させ, 輻輳回避フェーズでは線形的に損失ウィンドウを増加させることになる.

パケット棄却を検出した場合には損失ウィンドウが減少するが, 減少量はパケットの棄却の検出方法によって異なる. 減少後の損失ウィンドウは次式で与えられる.

$$cwnd(t+1) \leftarrow \begin{cases} \frac{cwnd(t)}{2} & (\text{重複ACKによる検出}) \\ 1 & (\text{タイムアウトによる検出}) \end{cases}$$

重複 ACK を受信した場合のパケットロス, ネットワークに軽度の輻輳が発生したと判断して損失ウィンドウを現在の値の半分に減少させる. 一方, タイムアウトによるパケット棄却を検出した場合は, ネットワークに重度の輻輳が発生したと判断して損失ウィンドウを 1 に減少させる. もう一方のウィンドウである遅延ウィンドウもスロースタートフェーズと輻輳回避フェーズにより動作が異なる. 遅延ウィンドウは, スロースタートフェーズ時においては動作せず, 輻輳回避フェーズ時のみ動作する.

$$Expected = \frac{swnd(t)}{baseRTT}$$

$$Actual = \frac{swnd(t)}{RTT}$$

$$Diff = (Expected - Actual) \cdot baseRTT$$

ここで, $baseRTT$ は観測された往復遅延時間の最小値, RTT は現在の往復遅延時間である.

遅延ウィンドウサイズは, ネットワーク中に滞留しているパケット数 $Diff$ をもとに次式で計算される.

$$dwnd(t+1) = \begin{cases} dwnd(t) + (\alpha \cdot swnd(t)^k - 1) & (Diff < \gamma) \\ (dwnd(t) - \zeta \cdot Diff) & (Diff \geq \gamma) \end{cases}$$

推測値 $Diff$ が閾値 γ よりも小さい場合は, ネットワークに未使用帯域があると判断し, 遅延ウィンドウを増加させる. 逆に, 推測値 $Diff$ よりも閾値 γ が大きい場合は, ネットワークに輻輳が発生していると判断して, 遅延ウィンドウを減少させる.

CTCP は損失ウィンドウおよび遅延ウィンドウを用いて送出ウィンドウを次式に従って決定する.

$$swnd(t+1) = cwnd(t+1) + dwnd(t+1)$$

2.3 輻輳制御手法間の公平性に関する研究

これまでに複数の TCP アルゴリズムが混在する環境における TCP アルゴリズム間の公平性の評価が行われている.

TCP Reno と TCP Vegas の混在環境における公平性の評価として[3]がある. 本研究では TCP Reno が通信速度を上昇させるに伴い TCP Vegas が通信速度を減少させ, 両 TCP 間の公平性が損なわれることが示されている.

また, 逸身らにより CTCP と CUBIC-TCP の混在環境における両 TCP の公平性の評価, 公平性改善手法の提案と評価が行われている[7]. 両 TCP が混在する環境では CUBIC-TCP が CTCP の性能を大きく上回ってしまい公平性が低いことが示され, これを解決する手法としてネッ

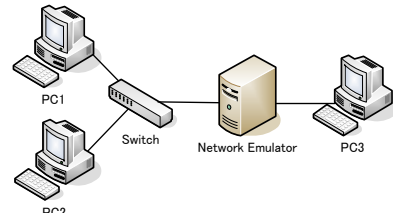


図 1. 実験ネットワーク (dummynet)

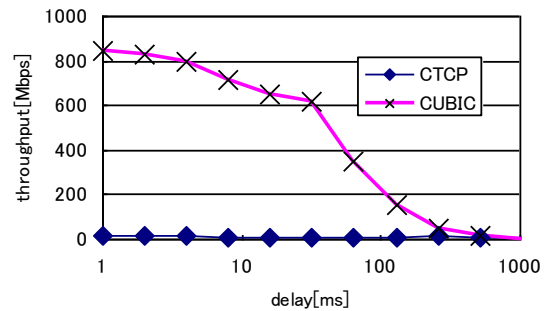


図 2. 混在環境での性能 (dummynet)

トワークルータにて RED (Random Early Detection) [8]を用いる手法が提案されている. そして, 性能評価により提案手法を用いることにより公平性の改善がなされることが示されている.

3. 性能評価

本章で, 実機ネットワークを用いて異なる高速 TCP が混在する環境における各 TCP の通信性能と公平性の評価を行う.

3.1 ソケット通信性能の評価

実験 1 では, 図 1 の様なネットワークを構築し, CUBIC-TCP, CTCP の両者が混在する環境における通信速度を測定した. 基本的なソケット通信の性能評価として netperf を用いた. 図 1 のすべての通信機器は 1Gigabit Ethernet 対応の機器である. ネットワークエミュレータは FreeBSD Dummynet を使用した.

図 1 の PC1-PC3 の間と PC2-PC3 の間で同時に netperf 接続を確立し, それぞれの通信性能を測定した. PC1 では CTCP が動作し, PC2 では CUBIC-TCP が動作している. 両通信は Switch から PC3 の間のネットワークを共有している. それぞれにて得られた通信速度を図 2 に示す. 横軸が片道遅延時間を表す. 図より, 混在環境では CTCP の遅延ベースの輻輳制御が CUBIC-TCP の輻輳制御に圧迫され, 通信を控えることにより公平性が低くなり, CUBIC-TCP の性能が CTCP の性能を大きく上回る結果となった.

実験 2 では, 図 3 の様に Dummynet を 8 台の Cisco1841 Router に置き換えたネットワークを構築し, 実験 1 と同様の測定をした. 本ルータの NIC の最大通信速度は 100Mbps である. PC1, PC2 から PC3 の間の往復遅延時間は 0.98ms である. それぞれにて得られた通信速度を図 4 に示す. 図より, ネットワークエミュレータ環境でなく, 実ルータ環境においても実験 1 と同様の結果が得られ, 実験 1 の結果がネットワークエミュレータに依存するものでないことがわかる.

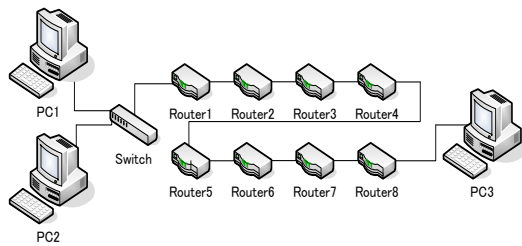


図3. 実験ネットワーク (Router)

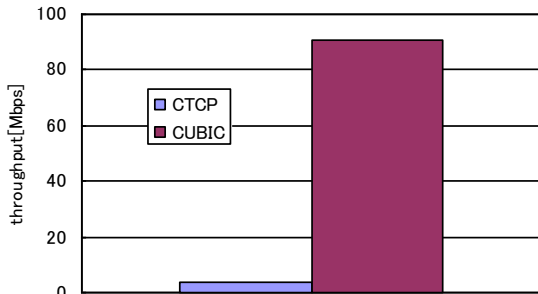


図4. 混在環境での性能 (Router)

3.2 HTTP を用いた評価

実験3では、図1のネットワークにおいて PC1-PC3 の間と PC2-PC3 の間で同時に HTTP 接続を確立し、それぞれの通信性能を測定した。得られた性能を図5に示す。実験1と同様に、CUBIC-TCP の性能が CTCP の性能を大きく上回る結果となった。

3.3 複数コネクションを用いた評価

実験4では、図1のネットワークにおいて PC1-PC3 の間に CTCP の HTTP コネクションを16本確立し、PC2-PC3 の間に CUBIC-TCP による HTTP のコネクションを1本確立し、それぞれの性能を測定した。測定結果を図6に示す。縦軸は全コネクションで得られた通信速度の合計である。図より、複数のコネクションを確立しても CUBIC-TCP の性能が CTCP の性能を大きく上回る結果となった。

3.4 パケットロス環境における評価

文献[7]において、ネットワーク上のルータにおいて RED を用いパケットを棄却させることにより公平性の改善が達成されることが示されている。これより、パケット棄却が公平性向上に寄与できると予想される。我々は簡略化されたパケット棄却環境としてキューの長さ依存せずランダムにパケットを棄却させる環境を構築し、この環境における公平性を評価した。パケット棄却は Dummynet で行い、パケット棄却率は 10^{-5} である。

本環境における両 TCP の性能を図7に示す。図1と比較を行うと、パケットを棄却させることにより、CUBIC-TCP の性能が減少するとともに CTCP の性能が向上することが確認され、両 TCP の公平性の改善がなされたことが分かった。

棄却を発生させた場合とさせなかった場合の公平さと合計通信速度の比較を図8に示す。公平さは図8の”棄却あり 通信速度比”と”棄却なし 通信速度比”に表されており、ともに CUBIC-TCP と CTCP で得られた通信速度の比であ

る。1に近いほど公平性が高いこととなる。公平性を比較すると、棄却なしの場合は50倍から110倍程度の差であったものが、棄却を発生させることにより1倍から20倍程度の差にまで改善されているのがわかる。大きな改善がみられたが、依然として差が大きい。また、棄却ありと棄却なしにおける両通信の合計速度を比較すると、棄却により合計通信速度が1/3程度に減少していることがわかる。よって、CUBIC-TCP の性能が大きく削減され CTCP の性能が向上しているが、CTCP の性能向上量は CUBIC-TCP の性能減少量よりも小さいことがわかる。

3.5 コネクション連結

CTCP と CUBIC-TCP の公平性を保ちつつ、CTCP のスループットを向上させるために図9のようなコネクション連結ネットワークを構築した。PC1 と Switch の間に Proxy Server を配置して、これを介して PC1 と PC3 は通信を行う。

Proxy Server 機では Linux 上で squid が動作しており、TCP アルゴリズムとしては CUBIC-TCP が動作している。実験6では、図9のネットワークで実験1と同様の測定を行った。実験結果を図10に示す。図より CTCP が CUBIC-TCP と高い公平性を維持しつつ、CTCP のスループットが向上したことを確認できた。

実験7では、実験6と同様に PC1-Switch 間に ProxyServer を配置し、実験2と同様にルータを用いて測定を行った。この時得られた通信速度を図11に示す。図より、Router を用いた実機のネットワークにおいても CTCP のスループットと公平性が向上することを確認できた。これらより、ネットワーク上に TCP Ack を返す高速化装置を配置するなどの手法にも効果があると考えられる。

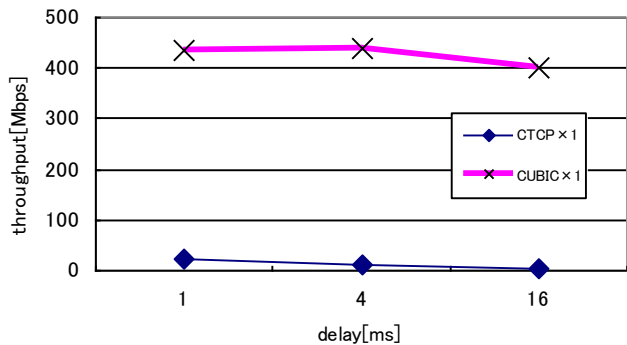


図5. 混在環境での性能 (HTTP, 単一コネクション)

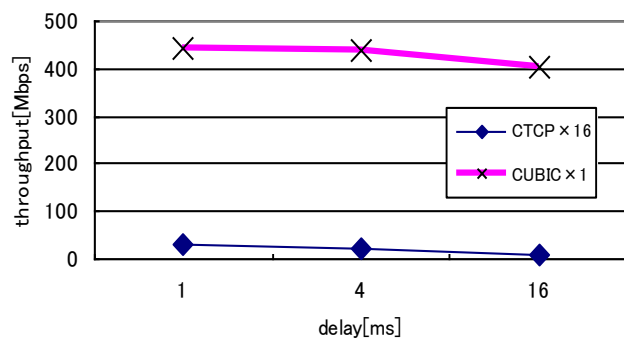


図6. 混在環境での性能 (HTTP, 複数コネクション)

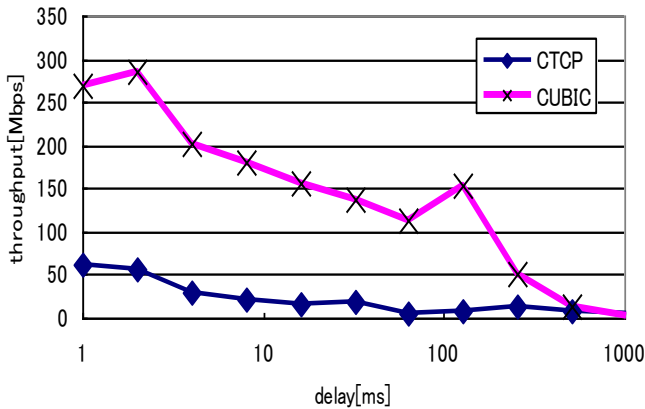


図7. 混在環境での性能(dummynet, packet loss)

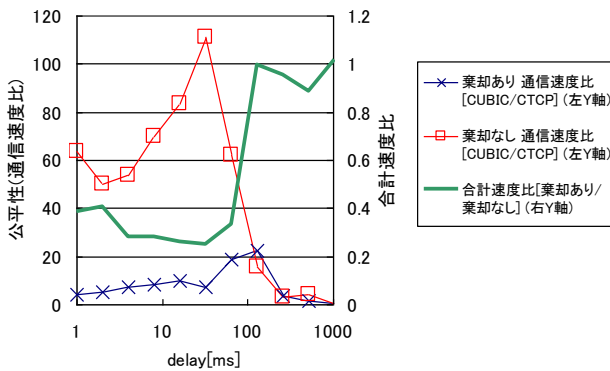


図8. 公平性(通信速度比)と合計速度

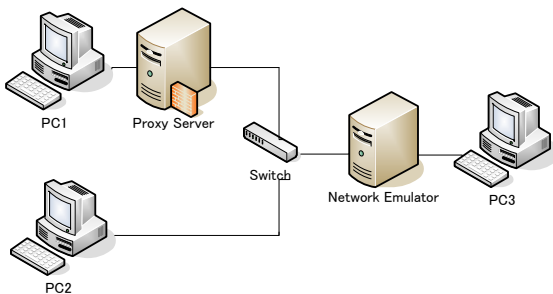


図9. proxy serverを用いた実験ネットワーク

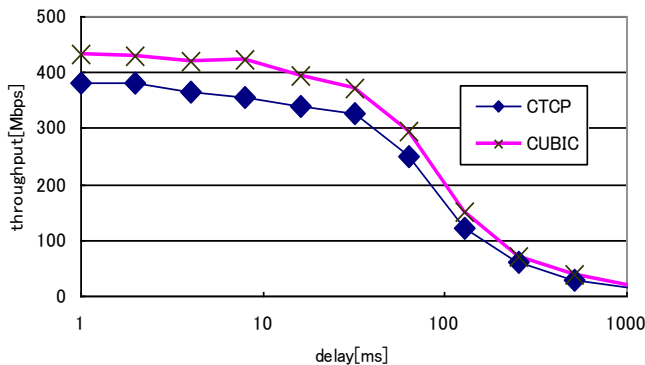


図10. 混在環境での測定(proxy server-Dummynet)

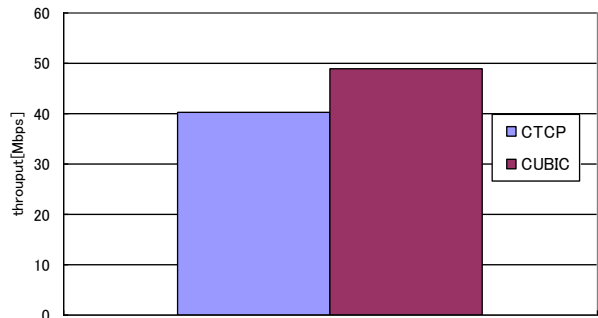


図11. 混在環境での通信速度(proxy server-Router)

4. おわりに

本稿では、CTCP と CUBIC-TCP を対象とし両者が共存する実機ネットワーク環境における通信性能の測定を行い、公平性に関する考察を行った。そして、その改善手法の効果の評価を行った。評価の結果、実機ネットワークにおいても公平性が低いことと、パケット棄却やコネクションの連結などの手法により性能と公平性の改善が可能であることが確認された。今後は、RED を用いた場合の評価などを行っていく予定である。

謝辞

本研究は科研費(22700039)の助成を受けたものである。

参考文献

- [1] D.Katabi, M.Handley, and C.Rohrs, "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] Jeonghoon Mo, Richard J. La, Venkat Anantharam, and Jean Walrand, "Analysis and comparison of TCP reno and vegas", in Proceedings of IEEE INFOCOM'99, March 1999.
- [4] L. S. Brakmo and L. L. Peterson, "TCP Vegas: End to End Congestion Avoidance on a Global Internet", IEEE Journal on Selected Areas in Communication, Vol.13, No.8, pp.1465-1480, October 1995.
- [5] Kun Tan, Jingmin Song, Qian Zhang, and Murari Sridharan, "A Compound TCP Approach for High-speed and Long Distance Networks" Proc.of IEEE INFOCOM'05, July 2005.
- [6] 鈴木 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
- [7] 逸身 勇人, 山本 幹, "CUBIC と Compound TCP 間の公平性改善手法の提案" 電子情報通信学会 信学技報 vol. 110, no. 372, NS2010-160, pp. 103-108.
- [8] Sally Floyd and Van Jacobson, "Random Early Detection Gateways for Congestion Avoidance" August 1993 IEEE/ACM Transactions on Networking.