L-015

遅延時間制御手法の改変による TCP 公平性改善手法における破棄率の動的制御
Improving TCP Fairness by Modified CoDel with Dynamic Dropping Rate

花井 雅人† 山口 実靖† 小林 亜樹†
Masato Hanai Saneyasu Yamaguchi Aki Kobayashi

1. はじめに

複数の高速 TCP アルゴリズムの提案により，TCP アルゴリズム間の性能公平性の確保という課題が生じている[1][2]．本研究では，著名な TCP アルゴリズムである CUBIC TCP と Compound TCP に着目し，アルゴリズム間の性能公平性を評価し通信速度に不公平が生じることを示す．そして，遅延時間制御に破棄率を設け TCP 公平性を改善する手法およびその破棄率を動的に制御し公平性のさらなる改善を行う手法を提案する．また，性能評価によりこれらの手法の有効性を示す．

2. 基礎性能調査

2.1 Taildrop と CoDel

TailDrop と CoDel のスループットの比較を行う．図 1 のネットワークを構築し，CUBIC TCP と Compound TCP コネクションをそれぞれ確立し，通信速度を iperf[3]を用いて測定した．Linux 1，Linux 2，Windows の 3 台の計算機の仕様は表 1 の通り，CoDel を搭載した CoDel 機およびネットワークの遅延をエミュレートする Delay 機の仕様は表 2 の通りである．Delay 機は人工的にネットワーク遅延時間を発生させる装置であり，Linux Netem を用いて構築した．

通信はそれぞれ Linux1-Linux2 間と，Windows-Linux2 間で行い，Linux1 と Windows を送信端末，Linux2 を受信端末とした．Linux1-Linux2 間通信と Windows-Linux2 間の通信は個別に行った． Delay 機における人工的な遅延時間は 1ms から 64ms 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 32MB とした．

測定結果を図 2，図 3，図 4 に示す．図より，TailDrop，CoDel ともものコネクション数や送信端末の TCP アルゴリズムによらず最大(ワイヤレート)に近いスループットを出せていることが分かる．つまり、CoDel は TailDrop と比較してより積極的にパケットを破棄しているが、それによるスループットの低下は小さくなっていることが分かる．

2.2 TCP 公平性

前節の評価と同様に，図 1 の実験環境にて Linux1-Linux2 間と，Windows-Linux2 間で iperf のコネクションを同時に確立し，CUBIC TCP コネクションと Compound TCP コネクションが混在する環境における通信速度を測定した．

Linux1-Linux2 間の通信と Windows-Linux2 間の通信は CoDel 機から Delay 機，Linux2 までのネットワークを共有している．

測定結果を図 5，図 6 に示す．横軸の値は Delay 機により付加したネットワーク内における送信者-受信者間の往復遅延時間である．縦軸は各通信が得た通信速度で 10 コネクションの平均値である．図より，CUBIC TCP の通信速度と Compound TCP の通信速度には大きな差があり，TCP 公平性が非常に低いことが確認できる．特に付加遅延時間=64[ms]において公平性が低くなっている．

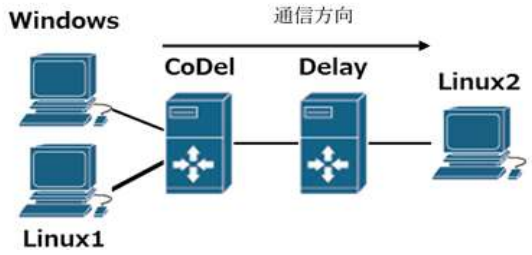

図 1 ネットワーク構成

表 1 計算機仕様

CPU	AMD Turion, 2.20[GHz]
メモリ	4[GB]
OS	(Linux1) Linux 4.2.3 (Linux2) Linux 3.3.4 (Windows) Windows7 Enterprise

表 2 計算機仕様

CPU	Intel CelronG540, 2.4[GHz]
メモリ	2[GB]
OS	(Codel, Delay) Linux 3.17.4

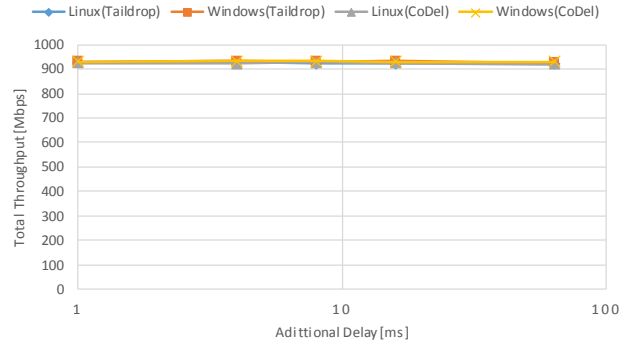

図 2 スループット(1 コネクション)

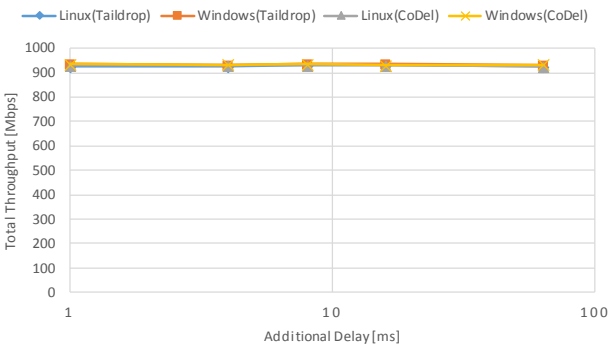

図 3 スループット(5 コネクション)

また，図 6 より Codel の適用により大きな通信速度の劣化なく通信遅延時間の減少が実現できていることが確認できる．

3. 提案手法

本章にて，CoDel に基づく TCP 公平性の改善手法を提案する．CoDel ではパケットの待ち時間が指定値 target を超えると必ずパケットの破棄を行うが，本稿では通信帯域の消費が最も大きいと予想されるコネクションのパケットは必ず破棄し，それ以外のパケットは確率 p で破棄する手法を提案する．p はチューニングパラメータである．

通信帯域の消費が最も大きいコネクションの推定は以下の通りに行う．通信機器にて，通過するパケット *stat_int* 個につき 1 個のパケットのコネクション情報を記録する．履歴は *hist_len* 個記録し，この数の履歴が得られる度に履歴を集計し履歴内における登場回数が最も多いコネクションを特定する．これをネットワーク帯域消費が最大のコネクションであるとし，これ以外のコネクションに対しては破棄率 p を適用する．

4. 性能評価

提案手法の有効性を検証するために，提案手法を実装し性能評価を行った．

2 章の評価と同様に，図 1 の実験環境にて Linux1-Linux2 間と，Windows-Linux2 間で iperf のコネクションを同時に確立し，通信速度を測定した．

提案手法における評価結果(FairnessIndex)を図 7 に示す．図の横軸は破棄率 p である．図の p=100%は既存の CoDel と同等である．図より提案手法が TCP 輻輳制御アルゴリズム間の公平性を大きく改善していることが確認できる．

図における p と通信速度の関係に着目すると，p が大き過ぎると優先制御が不十分であり，不公平性が十分に解消できていないことがわかる．今回の測定の例では破棄率 33%程度が適切な値となっている．また p を過度に小さくすると公平性が劣化することを確認できるが，p が小さすぎることによる負の影響は大きくなく，p は小さい値とすることが適切であることがわかる．

5. おわりに

本稿では，通信機器にて通信パケットの観察を行いネットワーク帯域消費が最大のコネクションの推定を行い，この推定に基づきパケット優先破棄を行い，TCP 公平性を向上させる手法を提案した．

今後は，新規パラメータ(破棄率)の導入を行わず，既存パラメータ(target)の制御のみにより公平性の改善を行う手法について考察していく予定である．

謝辞

本研究は JSPS 科研費 25280022，26730040，15H02696 の助成を受けたものである．
本研究は，JST，CREST の支援を受けたものである．

参考文献

[1] 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

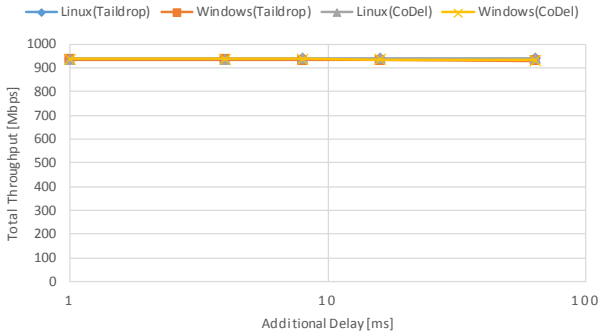

図 4 スループット(10 コネクション)

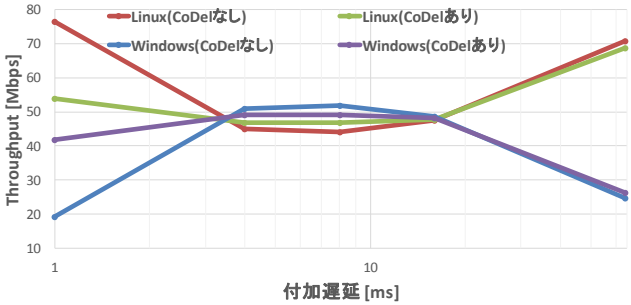

図 5 評価結果

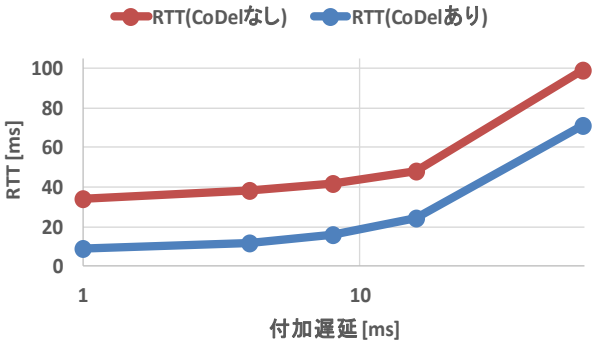

図 6 評価結果(送信者-受信者間の RTT)

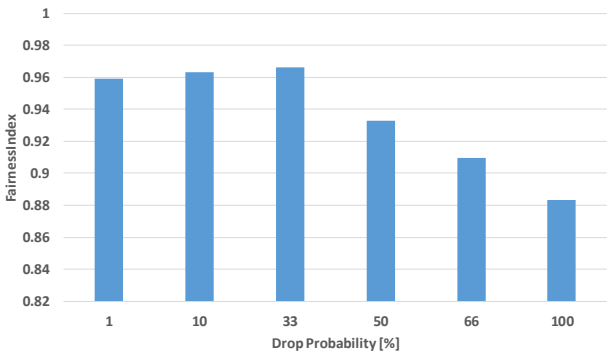

図 7 評価結果(FairnessIndex)

[2] Ryo Oura, Saneyasu Yamaguchi, “Fairness Analysis among Modern TCP Congestion Avoidance Algorithms Using Actual TCP Implementation and Actual Network Equipments,” ICNC 2011: 297-299

[3] iperf homepage, https://iperf.fr/