

# 複数リンク経由時の輻輳制御アルゴリズムの公平性分析

原田 香子<sup>†</sup> 上山 憲昭<sup>†</sup>

<sup>†</sup> 立命館大学 情報理工学部

〒 567-8570 大阪府茨木市岩倉町 2-150

E-mail: [jis0686ph@ed.ritsumei.ac.jp](mailto:jis0686ph@ed.ritsumei.ac.jp), [†kamiaki@fc.ritsumei.ac.jp](mailto:†kamiaki@fc.ritsumei.ac.jp)

あらまし 近年、インターネット上では異なる輻輳制御アルゴリズム (CCA: Congestion Control Algorithm) が混在しており、CCA の種類によって経由リンクの伝送帯域の配分が特定のフローに偏るという不公平性の問題が顕在化している。これまでの CCA の公平性に関して分析した多くの研究では、ネットワーク構成に単一のリンクのみを考慮したダンベルトポロジが用いられることが多く、複数リンクを経由するような複雑な経路における CCA 間の競合については十分に評価されていない。通常、インターネット上でフローは多数のリンクを経由し、経路途中で競合相手は変化し、異なるフローの集合が各リンクの伝送帯域をシェアする。また、経由リンクの重畳するフローの CCA の組み合わせによって、エンドツーエンドのスループットが決まることが考えられる。そのため、CCA の公平性を分析するには、異なるフロー集合が各リンクを共有する複数のリンクを考慮した評価が重要である。そこで本稿では、フローが複数のリンクを経由し、各リンクに異なる CCA を用いるフローが多重した場合を想定し、単一リンク経由時の公平性評価結果とは異なる公平性の結果が生じるかを分析する。実験にはネットワークエミュレータ Mininet を用いて仮想的に実験用トポロジを構築し、Visual Studio Code の SSH リモート接続機能を用いて、ホスト端末からスクリプトの編集・実行・結果の確認を一元的に行える開発環境を準備する。これにより、Mininet 上で動作する仮想ホストの設定・監視を効率的に行うことができる。仮想ホストには、`sysctl` コマンドを用いて任意の CCA を選択・適用する。実験には 3 つの代表的な CCA を用いる。1 つ目は、初期に標準的に使用されていた損失ベースの TCP Reno である。2 つ目は、現在標準的に使用されている損失ベースの TCP Cubic である。3 つ目は、Google 社が開発し、YouTube や Google Drive などで使用されている遅延ベースの BBR である。これら 3 つの CCA を各ホストから生成されるフローに適用し、スループットを観測する。

謝辞 本研究は JSPS 科研費 (JP25K03113, JP23K28078) の助成を受けたものである。

キーワード 輻輳制御アルゴリズム, スループット公平性, 複数リンク

## Fairness Investigation of Congestion Control Algorithms Through Multiple Links

Kako HARADA<sup>†</sup> and Noriaki KAMIYAMA<sup>†</sup>

<sup>†</sup> College of Information Science and Engineering, Ritsumeikan University

2-150, Iwakuracho, Ibaraki, Osaka 525-8570

E-mail: [jis0686ph@ed.ritsumei.ac.jp](mailto:jis0686ph@ed.ritsumei.ac.jp), [†kamiaki@fc.ritsumei.ac.jp](mailto:†kamiaki@fc.ritsumei.ac.jp)

**Abstract** In recent years, the coexistence of different congestion control algorithms (CCA) on the Internet has led to the emergence of unfairness issues, where the allocation of transmission bandwidth across links becomes biased toward specific flows depending on the type of CCA. Many previous studies analyzing CCA fairness have often used a dumbbell topology, considering only a single link in the network configuration. Consequently, competition between CCAs on complex paths traversing multiple links has not been sufficiently evaluated. Typically, flows on the Internet traverse numerous links, encountering different competitors along the path, with sets of different flows sharing the transmission bandwidth of each link. Furthermore, the combination of CCAs for flows overlapping on intermediate links likely determines the end-to-end throughput. Therefore, evaluating fairness across multiple links where different flow sets share each link is crucial for analyzing the fairness of CCA. This paper aims to analyze whether fairness evaluation results differ when assuming different CCA multiplexing patterns on each link within a multi-link scenario compared to a single-link scenario. For experiments, a virtual experimental topology is constructed using the network emulator Mininet. A development environment is set up for Mininet using Visual Studio Code's SSH remote connection feature, allowing for centralized script editing, execution, and result verification from the host terminal. This allows efficient configuration and monitoring of virtual hosts running on Mininet. Arbitrary CCAs were selected and applied to the virtual hosts using the `sysctl` command. Three representative CCAs are used in the experiments. The first is TCP Reno, a loss-based CCA initially used as the standard. The second is TCP Cubic, the loss-based CCA currently used as standard. The third is BBR, a delay-based CCA developed by Google and used in services like YouTube and Google Drive. We apply these three CCAs to flows generated from each host and observed throughput.

**Key words** congestion control algorithm, throughput fairness, multiple links