

多段リンクにおける輻輳制御アルゴリズムの公平性分析

原田 香子[†] 上山 憲昭[†]

[†]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: †is0686ph@ed.ritsumei.ac.jp, ††kamiaki@fc.ritsumei.ac.jp

あらまし 近年、動画配信サービスやクラウドサービスの普及により、インターネットにおけるトラフィック量は爆発的に増加している。限られたネットワーク資源を効率的かつ安定的に利用するためには、輻輳制御の存在が必要不可欠である。輻輳制御はTCPの主要な機能として実装され、伝送量を調整することでパケット損失や遅延を防ぐ役割を担う。長年、TCP-Reno (Reno) や CUBIC-TCP (Cubic) といったパケット損失を検知して制御を行う損失ベースアルゴリズムが標準的に利用されてきたが、近年、Googleによって提案されたBBR (bottleneck bandwidth and round-trip propagation time) のような遅延ベースのアルゴリズムが登場し、急速に普及している。しかし、動作原理が異なるアルゴリズムが同一リンク上に存在するとき、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。これまでの輻輳制御アルゴリズムの公平性に関して分析した多くの先行研究では、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分析は十分に評価されていない。そこで本稿では、ダンベルトポロジおよびマルチホップトポロジにおいて、BBRとReno, Cubicが競合した際のスループット公平性を分析する。そしてダンベルトポロジとマルチホップトポロジで異なる傾向があることを確認する。数値評価結果より、単一リンクでの評価だけでは、RTT差が帯域分配に与える影響を十分に捉えきれない。したがって、マルチホップ環境におけるフロー間の相互作用と遅延やバッファサイズの影響を考慮した、新たな公平性評価の必要性を明らかにする。

キーワード 輻輳制御アルゴリズム、スループット公平性、多段リンク

Fairness Analysis of Congestion Control Algorithm on Multiple Links

Kako HARADA[†] and Noriaki KAMIYAMA[†]

[†]College of Information Science and Engineering, Ritsumeikan University

2-150, Iwakura, Ibaraki, Osaka 567-8570

E-mail: †is0686ph@ed.ritsumei.ac.jp, ††kamiaki@fc.ritsumei.ac.jp

Abstract The rapid expansion of video streaming and cloud services has resulted in a substantial increase in internet traffic volume. Effective and stable utilization of limited network resources requires robust congestion control mechanisms. As a core function of TCP, congestion control mitigates packet loss and delay by dynamically adjusting transmission rates. Traditionally, loss-based algorithms such as TCP-Reno (Reno) and CUBIC-TCP (Cubic) have been widely adopted, relying on packet loss detection to regulate transmission. More recently, delay-based algorithms like BBR (bottleneck bandwidth and round-trip propagation time), developed by Google, have gained significant traction. When congestion control algorithms with differing operational principles coexist on the same link, allocation of transmission bandwidth can become skewed, favoring certain flows and resulting in fairness issues. Previous studies on congestion control fairness have predominantly utilized a Dumbbell topology, which models only a single network link, thereby limiting the evaluation of more complex, multi-link paths. This study assesses throughput fairness among BBR, Reno, and Cubic in both Dumbbell and multi-hop topologies. The findings reveal distinct trends between these topologies, demonstrating that single-link evaluations do not adequately capture the influence of round-trip time (RTT) differences on bandwidth distribution. These results underscore the necessity for a new fairness evaluation framework that incorporates flow interactions in multi-hop environments, as well as the effects of delay and buffer size.

Key words congestion control algorithm, throughput fairness, multiple links

1. はじめに

インターネット利用者の増大とアプリケーションの多様化に伴い、ネットワーク上のトラフィック特性は大きく変化している。特に近年では、高画質な動画ストリーミングや Zoom や Teams 等のリアルタイム Web 会議等、広帯域かつ低遅延を要求するアプリケーションが主流となりつつある。このような背景において、ネットワーク資源を公平かつ効率的に配分する輻輳制御技術の重要性は、かつてないほど高まっている。輻輳制御は、ウィンドウサイズ (cwnd) を適応的に制御し輻輳の発生を回避しながら、ネットワーク内の利用可能帯域を有効に活用できるようにする機能である。トランスポート層では、通信ホスト間において任意のアルゴリズムを用いた通信が可能であるため、輻輳制御アルゴリズム (CCA: congestion control algorithms) にもさまざまな種類がある。長年、インターネットの標準として広く利用されてきた Reno や Cubic といった損失ベースアルゴリズムに対し、近年では Google 社によって提案された Bottleneck Bandwidth and Round-trip propagation time (BBR) のような遅延ベースアルゴリズムが登場し、急速に普及している。しかし、動作原理が異なるアルゴリズムが同一リンク上で競合した場合、伝送帯域の配分が特定のフローに偏りが生じるという不公平性の問題が顕在化している。

そこで [1] [2] [3] など、これまでに輻輳制御アルゴリズムの公平性に関して分析した多くの研究が見られるが、单一のリンクのみを考慮したダンベルトポロジが評価トポロジとして用いられることが多い。しかし実際にはフローは複数のリンクを経由する場合が多く、このような実際の複雑な経路における輻輳制御アルゴリズムの公平性分析は行われていない。そこで本稿では、ダンベルトポロジおよびマルチホップトポロジにおいて、BBR と Reno, Cubic が競合した際のスループット公平性を分析する。

2. 関連研究

2.1 BBR の攻撃性と公平性評価

Hock らは、BBR が Cubic 等の損失ベースアルゴリズムに対して著しい不公平性を引き起こすことを指摘した [1]。この実験では、小規模なバッファを持つボトルネックリンクにおいて、BBR が帯域の大部分を占有し、競合する Cubic で輻輳制御を行っているフローの伝送帯域を圧迫する現象が確認されている。これは、BBR がパケットロスを無視して送信レートを維持しようとする一方で、Cubic がロス検知により cwnd を縮小させるという、両者の動作原理の根本的な不整合に起因するものである。また、BBR はその設計上、ボトルネック帯域幅を過大評価し、深刻なキューの蓄積を引き起こす脆弱性があることが解析されており、これが不公平性を助長する主な要因となっている [7]。

2.2 不公平性の発生メカニズムとモデル化

Ware らは、BBR と Cubic が競合した際のバッファ内の挙動を解析し、不公平性が生じるメカニズムを理論的に解明した [2]。まず定性的なメカニズムとして、BBR はボトルネック帯域を推定するために定期的に過剰なデータを注入する性質があり、これが浅いバッファ環境においてパケットロスを誘発させる。損失ベースのフローはこのロスを検知して伝送レートの減少処理を行うが、パケットロスを輻輳シグナルとして扱わない BBR は送信レートを維持するため、空いた帯域を即座に埋

めることで定常的な帯域独占状態が形成される。さらに、この挙動を定量的に裏付けるために、BBR の帯域占有率を予測する数学的モデルを提案している。近年では、このような多様な CCA が混在する環境において、フローレベルの性能を機械学習によって高精度に予測する手法も提案されており、複雑なネットワーク条件下での動的な相互作用の理解が進んでいる [6]。このモデルは、輻輳時において BBR が自身の In-flight cap (網内滞留パケット数の上限) によって実質的に制限を受けるという特性に基づき、以下の式で定義される。

$$p = \frac{1}{2} - \frac{1}{2X} - \frac{4N}{q} \quad (1)$$

$$BBR_{\text{frac}} = (1 - p) \times \left(\frac{d - \text{Probe}_{\text{time}}}{d} \right) \quad (2)$$

ここで、 BBR_{frac} は BBR フロー群の帯域占有率、 p は損失ベースフロー群のパケット損失率、 X は BDP (bandwidth-delay product) に対するバッファサイズ倍率、 q はパケット数、 d は RTT、 $\text{Probe}_{\text{time}}$ は往復伝搬遅延の再計測状態の時間を表す。また、 N はボトルネックを通過する BBR フローの数である。

2.3 インターネット上の実態調査

シミュレーションやラボ環境だけでなく、実際のインターネット環境における公平性の実態も明らかになりつつある。Philip らは、インターネット上の数千規模のサーバに対してアクティブ測定を行う監視システム「Prudentia」を開発した [3]。さらに、「Nebby」と呼ばれる受動的測定手法を用いた最新の調査では、Alexa Top 20k の約 36% 以上のサーバが依然として Cubic を採用しており、Cubic がインターネット上で最も支配的なアルゴリズムであることが示されている [4]。一方で、BBR の採用率は 2019 年以降やや縮小傾向にあり、特定のクラウドプロバイダに依存した分布となっているなど、ネットワーク全体で CCA の異質性が拡大している実態が報告されている [5]。

3. 評価トポロジ

本稿では、複数のリンクを経由するようなインターネット構造における、Reno や Cubic, BBR の挙動を評価する。加えて、単一のボトルネックリンクを持つダンベルトポロジにおいて各 CCA の挙動との比較を行う。具体的には、従来の評価で用いられてきた単純なダンベルトポロジに加え、より実環境に近い 2 ホップおよび 10 ホップのトポロジモデルを構築し、ホップ数の増加や競合環境の複雑化が公平性に与える影響を明らかにする。

3.1 ダンベルトポロジ

単一のボトルネックリンクを持つダンベルトポロジは、輻輳制御アルゴリズムの基礎特性を評価するために用いられる最も一般的なネットワーク構成である。本稿でも、まず、この簡易なトポロジを用いて、BBR と損失ベースアルゴリズム (Reno/Cubic) 同士や損失ベースアルゴリズム同士が競合した際の帯域占有率を測定する。実験に用いたトポロジを図 1 に示す。

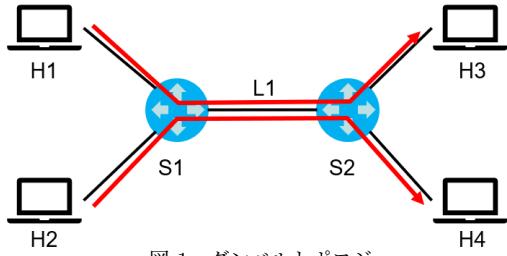


図 1 ダンベルトポロジ

3.2 2 ホップトポロジ

図 2 に示すように、2 ホップトポロジは、ダンベルトポロジを拡張し、ボトルネックとなるリンク L1 と L2 が直列に 2 箇所存在するモデルである。赤色の矢印で示す、 $H1 \rightarrow \text{Router}$ $1 \rightarrow \text{Router}$ $2 \rightarrow \text{Router}$ $3 \rightarrow H6$ を経由する、L1 と L2 の両方を経由する 2 ホップフロー（メインフロー）と、オレンジ色の矢印で示す、 $H2$ から $H3$ と、 $H4$ から $H5$ の、L1 か L2 の一方のみを経由するフロー（競合フロー）の 3 つのフローを考える。そして Link 1 および Link 2 の帯域やパケット数を制限することで、多段のボトルネックを形成する。ホップ数の増加による遅延の蓄積が公平性に与える影響を分離して評価するため、リンクあたりの帯域や遅延パラメタおよびパケット数は均一に設定する。そして 1 対 1 の状況下で BBR が損失ベースフローと競合した際や損失ベースフロー同士が競合した際に、ダンベルトポロジとの結果と比較し、その違いを明らかにする。

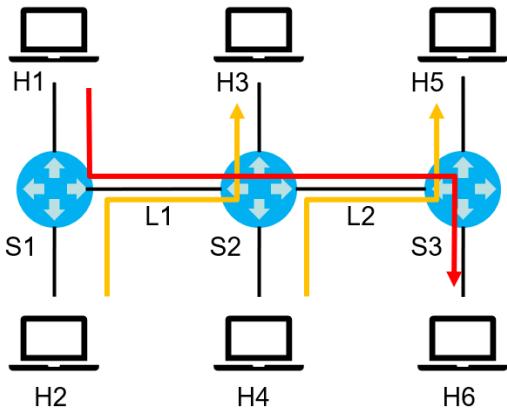


図 2 2 ホップトポロジ

3.3 10 ホップトポロジ

図 3 に示すように、10 ホップトポロジは、ボトルネックとなるリンクが直列に 10 箇所存在するモデルである。赤色の矢印で示す、 $H1 \rightarrow \text{Router}$ $1 \rightarrow \text{Router}$ $2 \rightarrow \dots \rightarrow \text{Router}$ $10 \rightarrow H22$ を経由する、L1~L10 の 10 本の全てのリンクを経由する 10 ホップフロー（メインフロー）と、オレンジ色の矢印で示す、 $H2$ から $H3$, $H4$ から $H5$, ..., $H20$ から $H21$ の、10 本のボトルネックリンクのいずれか一つのみを経由するフロー（競合フロー）の 11 のフローを考える。そして L1~L10 に対し帯域幅制限やバッファ制限を設けることで、多段のボトルネックを形成する。Main Flow は、すべての経由リンクにおいて自身よりも RTT が短いフローと競合する。Main Flow にとって RTT 不公平性の影響が強く現れる環境において、各 CCA の競合挙動がダンベルトポロジでの結果と異なるかを明らかにする。

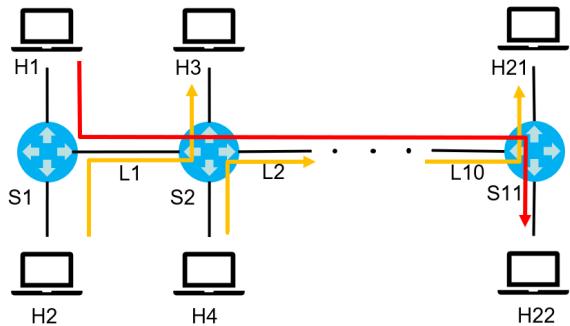


図 3 10 ホップトポロジ

4. 性能評価

4.1 性能評価条件

前節で示したダンベルトポロジ、2 ホップトポロジ、10 ホップトポロジを用いて数値評価を行う。本実験におけるシミュレーション環境および基本パラメタ設定値を表 1 に示す。実験環境の構築にはネットワークエミュレータ CORE (common open research emulator) を使用し、実環境に近い挙動を再現した。ネットワーク設定として、アクセスリンクの帯域幅は 1,000Mbps と十分に広く設定し、ボトルネックリンクの帯域幅を 50Mbps に制限することで、意図的にボトルネック化させる。また、ボトルネックリンクの遅延時間は、近距離通信から長距離通信までを想定し、2 ms (低遅延), 18 ms (中遅延), 100 ms (高遅延) の 3 段階に変動させる。

バッファサイズはボトルネックリンクにおけるパケット数の上限と定義する。このバッファサイズ B (packets) は、帯域幅遅延積 (BDP: bandwidth-delay product) を基準とし、係数パラメタ α を用いて、 $\alpha \times BDP$ と設定する。ここで α の値は、 BDP に対するバッファサイズの倍率を示し、 $\alpha \in \{0.5, 1, 2, 4, 8, 16\}$ の 6 通りの値を設定する。この算出値を適用することで、浅いバッファから深いバッファ環境までを網羅的に評価する。各評価パターンにおいて、2 回の測定を行い、その平均で評価を行う。トラフィックの生成およびスループットの測定には iPerf を用い、定常状態観測のため 60 秒間測定を行う。送信端末の CCA として、BBR, Cubic および Reno を用いる。

表 1 実験パラメタおよび設定条件

項目	設定値
基本環境設定	
ネットワークエミュレータ	CORE [10] [11]
使用する CCA	Reno, Cubic, BBR
アクセスリンク帯域幅	1,000 Mbps
アクセスリンク遅延	1 ms
ボトルネック帯域幅	50 Mbps
実験時間	60 秒
変動パラメータ	
ボトルネックリンク遅延	2 ms, 18 ms, 100 ms
バッファサイズ	$\alpha \times BDP$ ($\alpha \in \{0.5, 1, 2, 4, 8, 16\}$)

上記のシナリオにおいて、平均スループットを用いて各 CCA の性能および公平性を定量的に評価する。

4.2 ダンベルトポロジ

図 1 に示したトポロジにおいて、Reno と BBR, Cubic と BBR, Reno と Cubic を競合させた際の各フローのスループットを図 4 に示す。図では、高いスループットを獲得した場合ほ

ど、濃い青色で表示している。図4より、BBRと損失ベースCCAが競合する場合、小・中バッファ ($0.5 \leq \alpha \leq 2$)においては、BBRが帯域の約9割を取得しており、損失ベースCCAと比較して高いスループットを維持する傾向が見られる。これはBBRがパケットロスを直接的な輻輳信号として利用せず、推定された帯域とRTTに基づいて送信率を決定するためであり、既存研究においてもBBRが損失ベースCCAの帯域を圧倒する傾向が報告されている[1][2]。しかし、 $\alpha = 16$ の大バッファ領域においては、パケットロスが抑制されることでCubic等のウィンドウサイズが拡大し、BBRの推定帯域を圧迫する現象が確認された。このバッファサイズの増大に伴い、損失ベースCCAのスループットが回復しBBRのスループットを抑圧する挙動は、先行研究における実験結果とも一致する[1][3]。

一方、RenoとCubicが競合する場合、リンクあたりの遅延が2msのように極めて小さい環境において、RenoとCubicのスループットは拮抗した。これはCubicがRenoと同等のスループットを取得するTCP公平性の設計が機能しているためであると考えられる[8][9]。

高遅延環境(100ms)では、小バッファ領域 ($0.5 \leq \alpha \leq 2$)においてCubicが明確な優位性を示した。これは、高遅延環境での頻繁なパケットロスに対し、Renoのウィンドウサイズ回復がRTTに依存するのに対し、時間に基づいて制御を行うCubicがより迅速な帯域回復を実現したためと考えられる[4][5]。

中遅延環境(18ms)は中間的な挙動を示し、中規模バッファサイズ ($2 \leq \alpha \leq 4$)において、Cubicの時間に依存した制御による回復速度の優位性が局所的に現れた。

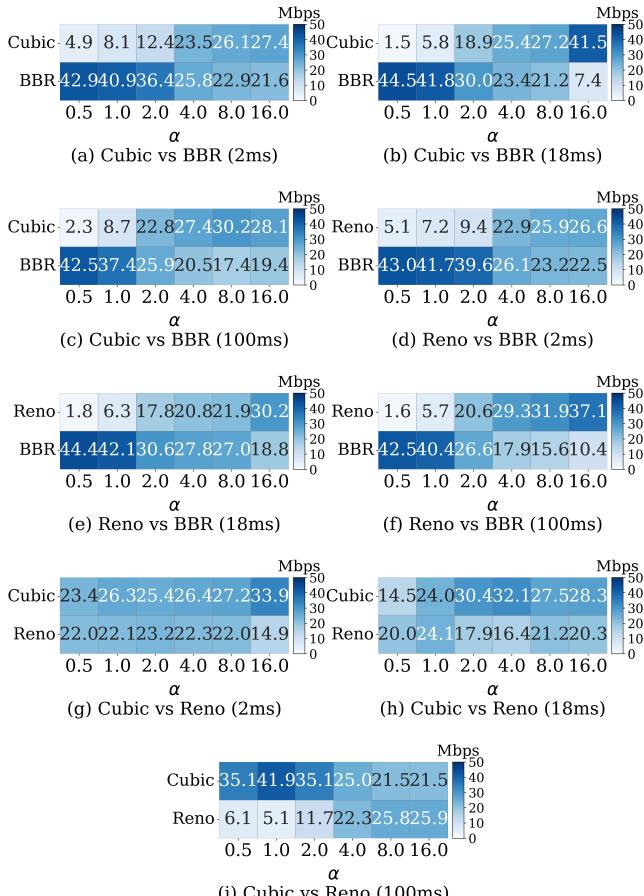


図4 Throughput Comparison in dumbbell topology

4.3 2 ホップトポロジ

図2に示したトポロジにおいて、各CCAを競合させた際の各フローのスループットを図5に示す。

図5より、小・中バッファ領域では、ホップ数の小さいフロー (BBR/Cubic) に2ホップのRenoフローは帯域を圧迫される傾向がある。しかし、遅延18msかつ $\alpha = 16$ の条件下では逆転現象が発生し、2ホップするRenoが最も高いスループット (28.7Mbps) を獲得した。大バッファ環境によりパケットロスが抑制されたことで、RenoやCubicはウィンドウサイズを十分に拡大できる。しかし、Cubicは飽和点に近づくとウィンドウ増加率を低減させる特性がある。この間もRenoが一定のペースでウィンドウサイズを線形増加させ続けるため、結果的にRenoの方が多くの帯域を取得したと考えられる。また、BBRについては、Renoのバッファ占有によって生じたRTTの増大をネットワーク輻輳と判定し、送信レートを抑制したことでスループットが低下したと推察される。

図5より、小バッファ領域 ($0.5 \leq \alpha \leq 1$)においては、2ホップするBBRフローが帯域の約8割を占有し、1ホップする損失ベースフローを圧迫する。しかし、中・大バッファ領域 ($\alpha \geq 2$)では、パケットロスが抑制されたことで1ホップするフローがウィンドウサイズを拡大し、2ホップするBBRのスループットを圧倒した。

図5より、2ホップのCubicフローは小バッファ領域では1ホップフローに圧迫されるが、中バッファ領域 ($2 \leq \alpha \leq 4$)において一時的にスループットが向上し、1ホップするBBRやRenoを上回る結果を示した。しかし、大バッファ領域 ($\alpha = 16$)では、RTTの短い1ホップフローがウィンドウサイズを拡大するため、2ホップするCubicはスループットが低下する傾向が見られた。

さらに、本実験結果を前節(ダンベルトポロジ)の結果と比較すると、トポロジの違いによる顕著な挙動の変化が確認された。前節では、大バッファ領域においてBBRが損失ベースアルゴリズムのフローに帯域を圧迫される傾向にあったが、2ホップトポロジにおいては、1ホップに割り当てられたBBRが、2ホップの損失ベースフローを逆に圧迫する結果となった。同様に、ダンベルトポロジの高遅延かつ大バッファ領域ではRenoがCubicに対して優位性を示したが、2ホップトポロジにおいては、1ホップのCubicが2ホップのRenoを圧迫する結果となった。

以上の結果より、单一リンクでのCCAの特性と比較して、マルチホップ環境ではフロー間のホップ数差(RTT差)が公平性に支配的な影響を与え、競合の勝敗を逆転させうることが明らかになった。

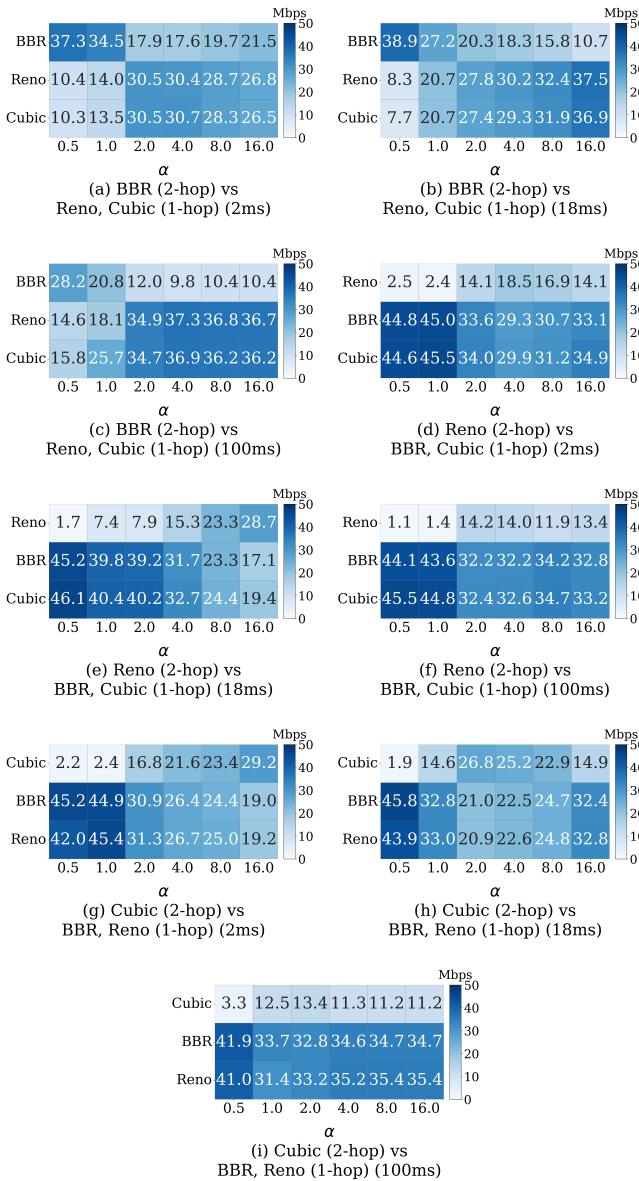


図 5 Throughput Comparison in 2-hops topology

4.4 10 ホップトポロジ

Reno, Cubic, BBR を 10 ホップフローに各々、割り当てたときの、各フローのスループットを、図 6, 7, 8 に示す。

図 6, 図 7 より、損失ベース CCA フロー同士が競合した場合、ホップ数の少ないフローが多くのスループットを取得した。ホップ数が少ないと ACK が頻繁に返るためウィンドウサイズの回復・増加が速やかに行われるため、長ホップフローに対して圧倒的な優位性を示した。

図 6, 図 7 より、1 ホップフローに BBR が割り当てられた場合、遅延 2ms かつ小バッファ領域 ($0.5 \leq \alpha \leq 2$) では BBR のフローが多くのスループットを取得した一方、大バッファ領域 ($\alpha \geq 8$) では損失ベース CCA フローが多くのスループットを取得した。この傾向は、ダンベルトポロジにおける実験結果と同様であり、バッファサイズに対する各 CCA の特性が再現されたものといえる。しかし遅延 18ms においては、競合するフロー間のスループットがある程度公平に分配される傾向を確認した。これは短ホップフローが持つ低遅延による物理的な有利さと、バッファサイズの増大に伴うパケットロス減少によって損失ベース CCA 享受する有利さが拮抗し、均衡状態が生じた

ためであると考えられる。また、10 ホップフローとして Reno を用いた場合の方が、Cubic を用いた場合と比較して、1 ホップの BBR フローに対して多くのスループットを取得する結果となった。この要因として、両者のウィンドウ制御アルゴリズムの差異が挙げられる。Cubic はパケットロス直前のウィンドウサイズに近づくと、増加率を意図的に下げる特性を持つ。長ホップ環境下では RTT が長いため、増加率が低い期間が引き延ばされ、ウィンドウサイズが停滞する期間が長く生じる。一方、Reno は停滞期を設けることなくウィンドウサイズを線形に増加する特性を持つ。このため、Reno の方が継続的にウィンドウサイズを増加することが可能になり、結果として Cubic よりも多くの帯域を確保できたと考えられる。

図 8 より、ホップ数の少ない損失ベース CCA フローが帯域の大部分を占有し、BBR のスループットは著しく低下する傾向がみられた。具体的には、遅延 100ms においては、BBR のスループットが 0.3Mbps まで低下する事例も確認された。この要因として、BBR の輻輳制御メカニズムと RTT の関係が挙げられる。BBR は RTT の増加をネットワークの輻輳と判断し、送信レートを抑制する特性を持つ。本実験のようなホップ数差が大きい環境では、10 ホップ側の RTT が物理的に長くなるため、BBR はこれを輻輳状態と認識する可能性が高い。さらに、1 ホップの損失ベース CCA がバッファを占有して遅延を増大させることで、RTT 比率がさらに拡大し、BBR の帯域抑制がより顕著になったと考察される。本来、BBR は広帯域高遅延環境においても高いスループット性能を発揮するよう設計された CCA である。しかし本実験の結果により、競合フローとの間に極端なホップ数差が存在する不均一な環境においては、その性能が極端に劣化する脆弱性が確認された。

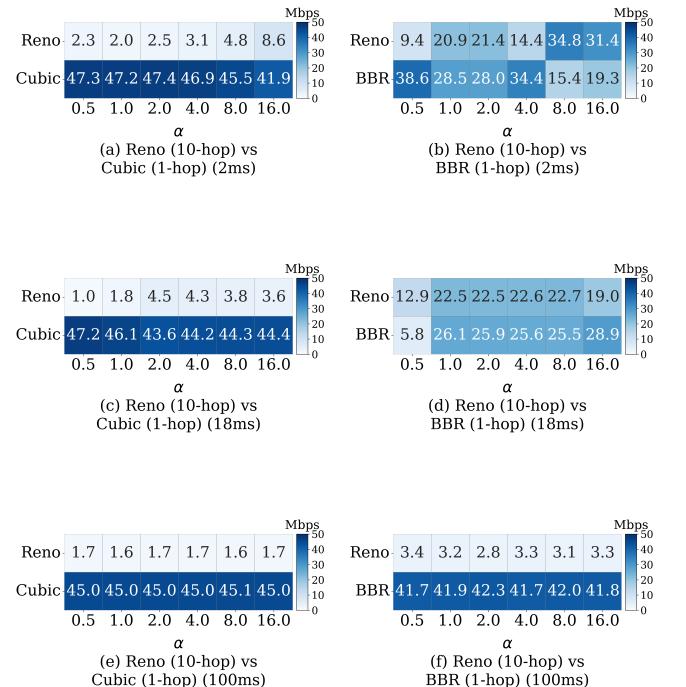


図 6 Throughput Comparison when Reno main-flow in 10-hops topology

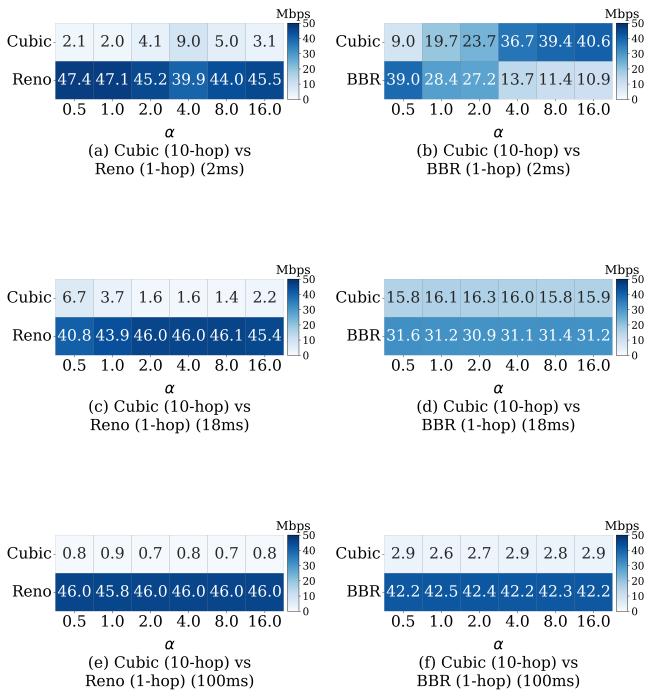


図 7 Throughput Comparison when Cubic main-flow in 10-hops topology

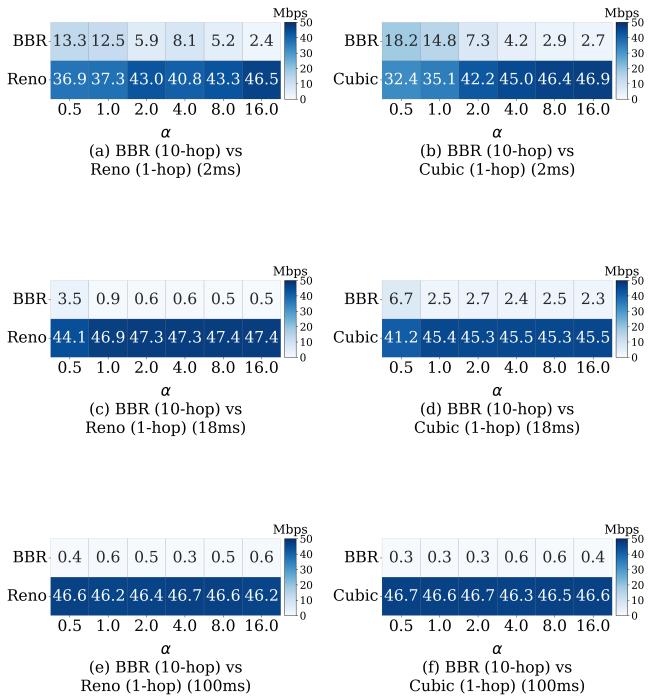


図 8 Throughput Comparison when BBR main-flow in 10-hops topology

5. まとめ

輻輳制御アルゴリズムの公平性分析に複数リンクを経由する場合のトポロジでの評価を提案した。従来の公平性分析ではダンベルトポロジのような单一のボトルネックリンクを経由するトポロジを用いての評価が行われているが、実際のインターネットにおいて、フローは複数のリンクを経由し、かつ、競合するフローの組み合わせもリンクごとに異なる点を考慮できて

いないことが課題である。そこで本稿では、複数のボトルネックリンクを経由した場合のCCAの挙動をスループットで評価し、単一のボトルネックリンクを経由する既存研究の評価方法との比較を行った。その結果、ダンベルトポロジでは単純な競合となるケースであっても、マルチホップ環境ではRTT差による不公平性の拡大などによってダンベルトポロジで得られた結果とは異なる、トポロジ構造に依存した複雑な分配挙動が確認された。したがって、輻輳制御アルゴリズムの公平性を評価するためには、単純なダンベルトポロジだけでなく、複数リンクを経由する場合のトポロジを使用した評価も不可欠であると結論付けられた。

謝辞 本研究はJSPS科研費(25K03113, 23K28078)の助成を受けたものである。ここに記して謝意を表す。

文 献

- [1] M. Hock, R. Bless, and M. Zitterbart, “Experimental evaluation of BBR congestion control”, the 25th IEEE International Conference on Network Protocols (ICNP 2017), Oct. 2017
- [2] R. Ware, M. K. Mukerjee, S. Seshan, and J. Sherry, “Modeling BBR’s Interactions with Loss-Based Congestion Control”, the ACM Internet Measurement Conference, (IMC 2019).
- [3] A. A. Philip, R. Athapathu, R. Ware, F. F. Mkocho, A. Schlomer, M. Shou, Z. Meng, S. Seshan, and J. Sherry, “Prudentia: Findings of an Internet fairness watchdog”, the ACM SIGCOMM 2024
- [4] Y. Cheng, G. Antichi, and M. Chiesa, “Keeping an Eye on Congestion Control in the Wild with Nebby”, the ACM SIGCOMM 2024
- [5] A. Mishra, “Understanding the Modern Internet’s Heterogeneous Congestion Control”, the ACM SIGCOMM 2024
- [6] C. Li, A. Nasr-Esfahany, K. Zhao, K. Noorbakhsh, P. Goyal, M. Alizadeh, and T. Anderson, “Predicting Flow-Level Performance with Machine Learning”, the ACM SIGCOMM 2024
- [7] R. Kumar, A. Koutsafitis, F. Fund, G. Naik, P. Liu, Y. Liu, and S. Panwar, “TCP BBR for Ultra-Low Latency Networking: Challenges, Analysis, and Solutions”, arXiv preprint arXiv:2311.08643, 2023
- [8] S. Ha, I. Rhee, and L. Xu, “CUBIC: a new TCP-friendly high-speed TCP variant”, ACM SIGOPS Operating Systems Review, 2008.
- [9] I. Rhee, L. Xu, S. Ha, A. Zimmermann, L. Eggert, and R. Scheffenegger, “CUBIC for Fast and High Speed Networks”, RFC 8312, Feb. 2018.
- [10] CORE Project Team, “Common Open Research Emulator (CORE) Source Code”, <https://github.com/coreemu/core>, Dec. 2025
- [11] DeepWiki, “CORE 3: Installation and Setup”, <https://deepwiki.com/coreemu/core/3-installation-and-setup>, Dec. 2025