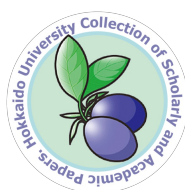




Title	FQ-Code1を用いた超低遅延ネットワークL4Sの性能評価
Author(s)	小倉, 拓海; 三橋, 力麻; 砂原, 悟 他
Citation	電子情報通信学会技術研究報告, 125(385), 351-356
Issue Date	2026-02-25
Doc URL	https://hdl.handle.net/2115/100412
Rights	Copyright ©2026 IEICE
Type	conference paper
File Information	NS2025-280.pdf



FQ-CoDel を用いた超低遅延ネットワーク L4S の性能評価

小倉 拓海[†] 三橋 力麻^{††} 砂原 悟^{†††} 飯田 勝吉^{†††} 高井 昌彰^{†††}

[†] 北海道大学 大学院情報科学院 情報理工学コース 先端ネットワーク研究室

〒060-0811 札幌市北区北 11 条西 5 丁目

^{††}, ^{†††} 北海道大学 情報基盤センター 〒060-0811 札幌市北区北 11 条西 5 丁目

E-mail: [†]ogura.takumi.m8@elms.hokudai.ac.jp, ^{††}mitsuhashi@tanega.iic.hokudai.ac.jp,

^{†††}{suna,iida,ytakai}@iic.hokudai.ac.jp

あらまし 近年、RFC9330 で定義された L4S は超低遅延ネットワークを実現する手段として注目を集めている。L4S を扱った先行研究は多くあるが、そのほとんどは Classic 通信と L4S 通信をそれぞれ一フローずつ流したシンプルな条件での性能比較に止まっており、実ネットワークのように多数の競合フローが存在する環境における L4S の性能は十分に検証されていない。そこで本研究では、AQM に FQ-CoDel を用いた L4S を実装し、基礎的性能評価を行う。具体的には、Classic 通信と L4S 通信を用意し、それぞれのフロー数を変動させることで、ボトルネックのキューイング遅延に与える影響を評価する。本実験では、異なる RTT のネットワーク環境において、「非常に低いキューイングレイテンシ」の条件を満たすフロー本数について調査を行った。

キーワード L4S, 輻輳制御, 低遅延, AQM, FQ-CoDel

Performance Evaluation of Ultra-low Latency Network L4S using FQ-CoDel

Takumi OGURA[†], Rikima MITSUHASHI^{††}, Satoru SUNAHARA^{†††}, Katsuyoshi IIDA^{†††}, and

Yoshiaki TAKAI^{†††}

[†] Laboratory of Advanced Information Network,

Course of Computer Science and Information Technology, Graduate School of Information Science & Engineering,
Hokkaido University, Kita 11 Nishi 5, Kita-ku, Sapporo, 060-0811, Japan.

^{††}, ^{†††} Information Initiative Center, Hokkaido University, Kita 11 Nishi 5, Kita-ku, Sapporo, 060-0811, Japan.

E-mail: [†]ogura.takumi.m8@elms.hokudai.ac.jp, ^{††}mitsuhashi@tanega.iic.hokudai.ac.jp,

^{†††}{suna,iida,ytakai}@iic.hokudai.ac.jp

Abstract In recent years, L4S defined in RFC 9330, has gained attention as a means to realize ultra-low-latency networks. While there are many previous studies on L4S, most are limited to performance evaluations of existing TCP communication with L4S communication. Consequently, the performance of L4S in environments with numerous competing flows, such as real networks, has not been sufficiently verified. Therefore, this study implements L4S using FQ-CoDel as the AQM and conducts a fundamental performance evaluation. Specifically, it varies Classic and L4S communications to assess their impact on bottleneck queuing delay. In this experiment, we investigated the number of flows that satisfy the condition of “very low queuing latency” in network environments with different RTTs.

Key words L4S, Congestion control, Low latency, AQM, and FQ-CoDel

1. はじめに

近年、RFC9330 で定義された Low Latency, Low Loss, Scalable Throughput (L4S) [1] は超低遅延ネットワークを実現する手段として注目を集めている。L4S は、送信者に対する早期の輻輳通知とボトルネックルータのキュー管理を組み合わせること

で、キューイング遅延の増大を抑えつつ高いスループットを目指す仕組みである。L4S が注目を集めている背景には、クラウドゲーミング、XR (AR/VR)、遠隔自動車運転、リアルタイムビデオ会議など、ネットワーク性能が重要な役割を果たすアプリケーションが増加している状況がある [2]。これらのアプリケーションは、操作性能や応答性能を確保するために高品質な

ネットワーク性能を要求しており、特にネットワーク遅延が影響を与えることが知られている [3]。

L4S を扱った先行研究は多くあるが [4]~[7]、そのほとんどは Classic 通信と L4S 通信をそれぞれ一フローずつ流したシンプルな条件での性能比較に止まっており、実ネットワークのように多数の競合フローが存在する環境における L4S の性能は十分に検証されていない。多数の競合フローとボトルネックを共有した場合、L4S の性能が十分に発揮されない可能性がある。

そこで本研究では、ns-3 ネットワークシミュレータ [8], [9] の環境において、Active Queue Management (AQM) に FQ-CoDel [10]、を用いた L4S を実装し、多数の競合フローが発生する条件下で基礎的性能評価を行う。具体的には、Classic 通信の輻輳制御アルゴリズムに CUBIC [11], [12]、L4S 通信の輻輳制御アルゴリズムに DCTCP [13], [14] を設定し、それぞれのフロー数を変動させることで、ボトルネックのキューイング遅延に与える影響を評価する。

2. 関連研究

前章で述べたように、近年 L4S は超低遅延通信を実現する手段として注目を集めている。本章では、L4S 通信と Classic 通信の性能比較に関する研究と、L4S を組み込んだシステムの提案に関する研究を紹介する。

先行研究では、L4S 通信を利用することによって Classic 通信よりも低遅延な通信を実現できることが報告されている。例えば、D. BoruOljira ら [4] は、本論文では、ボトルネックルータに DualPI2 AQM を実装した場合における、DCTCP や TCP Prague など L4S 向けの輻輳制御と、TCP Cubic などの従来型輻輳制御の共存性を評価した。S. Nádas ら [5] は、コア・ステートレス AQM (CSAQM) の概念を拡張した仮想デュアルキュー CSAQM を提案し、この手法を用いた L4S によって異なる輻輳制御を利用するフロー間で生じる不公平なリソース共有の問題を解決できることを示した。F. B. Sarpkaya ら [6] は、現状の TCP (CUBIC や BBR) と L4S 向けの Prague の性能を比較し、現状の TCP から Prague へ切り替えることの有効性を示した。L. Hagen ら [7] は、L4S で用いられる TCP Prague のフォールバックアルゴリズムに関するバグを修正した。また、バグ修正後、CoDel および DualPI2 AQM 上で Prague と Cubic/BBRv2 の共有挙動を分析した。

一方近年、高いネットワーク性能を要求するアプリケーションに L4S を組み込んだシステムを提案する研究が多数提案されている。例えば、D. Brunello ら [15] は、輻輳通知のための L4S マーキング戦略を提案し、独自開発のネットワークシミュレータを用いて 5G 環境で実装した。具体的には、L4S によって拡張現実 (AR) ビデオゲームトラフィックをサポートし、輻輳制御には SReAM を使用した。

J. Son ら [16] は、WebRTC を用いたリアルタイム通信のためのクラウドベースストリーミングシステムに組み込んだ適応型 L4S 輻輳制御 (aL4S-CC) を提案した。

これまで示した通り、L4S に関する研究は多数報告されているが、そのほとんどは L4S 通信と Classic 通信をそれぞれ一

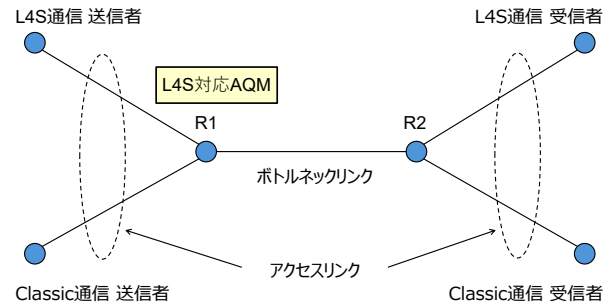


図 1: 多数の競合フローが発生する条件下での L4S の基礎的性能評価に用いるネットワークトポロジー

フローずつ流したシンプルな条件での性能比較に止まっており、実ネットワークのように多数の競合フローが存在する環境における L4S の性能は十分に検証されていない。多数の競合フローとボトルネックを共有した場合、L4S の性能が十分に発揮されない可能性がある。そこで本研究では、AQM に FQ-CoDel [10]、を用いた L4S を実装し、多数の競合フローが発生する条件下で、L4S の基礎的性能評価を行う。

3. 多数の競合フローが発生する条件下での L4S の基礎的性能評価手法

前章で説明した通り、L4S に関する研究は多数報告されているが、そのほとんどは Classic 通信と L4S 通信をそれぞれ一フローずつ流したシンプルな条件での性能比較に止まっており、実ネットワークのように多数の競合フローが存在する環境における L4S の性能は十分に検証されていない。そこで本章では、多数の競合フローが発生する条件下での L4S の基礎的性能評価手法を提案する。以下では、提案手法を構成するネットワーク環境、L4S アーキテクチャ、性能評価指標について説明する。

3.1 ネットワーク環境

L4S の基礎的性能評価は、図 1 に示すネットワークトポロジーで構成したネットワーク環境を用いる。このネットワークトポロジーでは、ボトルネックリンク (R1-R2) を配置し、ボトルネックルータ R1 上に L4S を適用する。

本ネットワーク環境の通信フローには、L4S を適用する L4S 通信と、競合フローであり L4S は適用しない Classic 通信を用意する。L4S 通信の輻輳制御アルゴリズムには DCTCP、Classic 通信の輻輳制御アルゴリズムには CUBIC を設定し、いずれの輻輳制御アルゴリズムに対しても ECN [17] を有効にする。

L4S 通信と Classic 通信はそれぞれ、ネットワークトポロジーの左側に置かれている送信者から、ネットワークトポロジーの右側に置かれている受信者に向けてデータが送信される。本ネットワーク環境では、すべての通信フローがボトルネックリンクを通過するため、多数の競合フローが L4S の性能に与える影響を調査できる。

3.2 L4S アーキテクチャ

L4S は、送信者に対する早期の輻輳通知とボトルネックルータのキュー管理を組み合わせることで、キューイング遅延の増大を抑えつつ高いスループットを目指す仕組みである。RFC9330

によると、L4S は、1) ボトルネックルータで L4S 通信とクラシック通信を分離する仕組み、2) ボトルネックルータで L4S 通信を識別する仕組み、3) 送信者に対する輻輳通知の制御の 3 つの要素からなる。以下ではこれらの 3 つの要素について説明する。

3.2.1 ボトルネックルータで L4S 通信とクラシック通信を分離する仕組み

L4S は、キューイング遅延の増大を抑えるため、L4S 通信のキューと Classic 通信のキューを分離する仕組みが必要である。本ネットワーク環境では、Active Queue Management (AQM) の一つである FQ-CoDel [10] をボトルネックルータ R1 に設定する。RFC9330 では、L4S の AQM として Dual-Queue Coupled AQM [18] を定めているが、L4S に対応した FQ-CoDel が ns-3 で標準機能として提供されており実装が容易なため、本ネットワーク環境ではこれを採用する。FQ-CoDel は、フローキューイングと CoDel (Controlled Delay) を組み合わせたパケットスケジューラ兼 AQM である [10]。FQ は到着パケットをフロー単位 (実装上はハッシュ等) で複数のサブキューに分離し、ラウンドロビン / DRR (Deficit Round Robin) 系のスケジューリングで送出することで、特定フローの占有や head-of-line blocking を緩和し、フロー間の公平性を高める。CoDel は各サブキューにおけるパケットのキュー滞留時間 (sojourn time) を監視し、遅延が目標値を継続的に超える場合にドロップまたは ECN マーキングを段階的に強めることで、バッファ肥大化 (bufferbloat) を抑制し、低遅延化を図る。本研究ではこの フロー分離+遅延制御により、競合フローが多数存在する条件でもキューイング遅延を抑えつつ、スループットの公平性を確保することを狙う。なお、CoDel の仕組みについては [19] が、FQ-CoDel の仕組みについては [20] と、[21] の Fig. 5 が参考になる。

3.2.2 ボトルネックルータで L4S 通信を識別する仕組み

ボトルネックルータは、L4S 通信と Classic 通信を異なるキューに振り分けるため、L4S 通信を識別する必要がある。本ネットワーク環境では、L4S 通信を識別するための仕組みとして、RFC9331 で定義されている DCTCP の ECT [22] を用いる。

3.2.3 送信者に対する輻輳通知の制御

L4S はキューイング遅延の増大を抑えるため、ECN を使って送信者に対して早期の輻輳通知を行う。

CE Threshold は、FQ-CoDel において ECN の輻輳通知 (CE マーク) を付与し始めるしきい値を表している。L4S では、低遅延を維持するために従来よりも小さい遅延で輻輳通知を発生させることが重要であり、CE Threshold は CE マークの開始点を制御する役割を持つ。一般に、CE Threshold を小さく設定するほど早期に CE マークが発生し、キューが短く保たれてキューイング遅延は小さくなる。ただし、CE Threshold を小さく設定した場合、比較的短いバーストでも CE マークが増加し、L4S 通信のスループットや公平性を低下させる可能性がある点には注意が必要である。

3.3 性能評価指標

L4S は、送信者に対する早期の輻輳通知とボトルネックルータのキュー管理を組み合わせることで、キューイング遅延の増

大を抑えつつ高いスループットを目指す仕組みである。本ネットワーク環境では、ボトルネックルータ R1 におけるキューイング遅延とスループットが性能評価の対象となる。

L4S のキューイング遅延を評価する指標として、RFC9330 の記述に基づき、「非常に低いキューイングレイテンシ」を用いる。具体的には、平均して 1 ms 未満かつ 99 パーセンタイルで約 2 ms 未満と定義される。そのため、本研究では、この遅延制約を満たすパラメータの範囲を探索する。

また、L4S 通信と Classic 通信のスループットの公平性を定量的に評価するために、Jain's Fairness Index (JFI) の指標を用いる。JFI は、各フローのスループットを x_i ($i = 1, \dots, n$) としたとき、次式で定義されている。本研究ではそれを満たすフロー数などの条件を探索する。

$$J(\mathbf{x}) = \frac{(\sum_{i=1}^n x_i)^2}{n \sum_{i=1}^n x_i^2}$$

ここで n は競合するフロー数である。JFI は $\frac{1}{n} \leq J \leq 1$ をとり、 J が 1 に近いほどフロー間でスループットが均等であることを意味する。

4. 評価実験

前章では多数の競合フローが発生する条件下での L4S の基礎的性能評価手法を説明した。本章では、前章で述べた提案手法に基づいて実装した実験環境と、キューイング遅延およびスループットの性能評価結果について述べる。

4.1 実験環境

本研究では、ネットワークシミュレータ ns-3 (version 3.45) を用いて性能評価実験を行った。すべての実験は Ubuntu 24.04.3 LTS を導入した計算機上で実行し、CPU は Intel(R) Xeon(R) E-2288G CPU @ 3.70GHz を使用した。以降に示す結果は、同一の実行環境において取得したものである。

シミュレーションは図 1 に示すダンベル型トポロジを用い、ボトルネックリンク (R1-R2) に 100Mb/s の帯域を設定し、Classic/L4S の競合区間とした。アクセスリンクは十分に大きい帯域 (1Gb/s) とし、ボトルネック以外で輻輳が発生しないように設定した。RTT はボトルネックリンクでの遅延を調整することで、送信者と受信者の間の往復遅延時間を 5ms、10ms、15ms と段階的に変化させる。また、Classic と L4S 双方のフロー数を変化させる。以下ではそれぞれの種別のフロー本数を 1-1、1-5、5-5 などと記載する。先に記載するフロー本数が Classic で、後に記載するフロー本数が L4S を指す。以上のパラメータが、2 つの性能指標であるボトルネックのキューイング遅延およびスループットに与える影響を調査する。また、本研究では、過渡状態 (スロースタート等) の影響を避けるため、スループットがおおむね収束した区間 (40-50s) を用いて性能を評価する。なお、シミュレーション時間は 55s とした。AQM の設定として、FQ-CoDel のキュー長 (最大蓄積パケット数) を 10240 パケットに設定した。また、FQ-CoDel は 1024 個のサブキューを持つ。本ネットワーク環境では FQ-CoDel のパラメータである CE Threshold に 1ms を設定する。

4.2 基礎的評価

本節では、Classic フローと L4S フローが競合する環境下におけるスループット性能と公平性について評価する。本研究におけるスループットは、各フローの受信側における受信バイト数の区間平均として算出する。図 2 に、各シナリオにおける L4S 機能のオフとオン時のスループット推移を示す。

また、実験の結果、表 1 に示す通り、L4S オンはいずれのフロー数においても JFI が 1 に近く、フロー間スループットがより均等に分配されていることが確認できる。

図 2(a) の L4S Off (1-1 フロー) では、CUBIC (赤) と DCTCP (青) が競合し、スループットが激しく振動している様子が確認できる。これは、L4S 機能が無効な状態では、DCTCP が標準的な CoDel の制御目標 (ターゲット遅延) に従わざるを得ず、DCTCP 本来の微細な輻輳反応が機能しきれていないためと考えられる。対照的に、図 2(b) の L4S On 環境では、両フローともに極めて平坦で安定したスループットを示している。FQ-CoDel のフロー分離に加え、L4S フローに対して適切な ECN マーキングが行われることで、DCTCP がウィンドウサイズを細かく制御し、安定した通信を実現していることがわかる。

フロー数が増加した図 2(g) (5-5 Off) や図 2(q) (10-10 Off) においても、L4S Off 時は初期の過渡応答後に振動が残る傾向が見られる。一方、図 2(h) (5-5 On) や図 2(r) (10-10 On) では、DCTCP フローが CUBIC フローと帯域を公平に分け合いながら、収束後速やかに安定状態へ移行している。これにより、L4S の導入が Classic フローを阻害することなく、ネットワーク全体の安定性を向上させることが確認された。

4.3 遅延に関する詳細評価

次に、本研究の主眼であるキューイング遅延について評価する。図 3 に各シナリオにおけるキューイング遅延の時系列変化を示し、図 4 に、RTT を変化させた際の遅延統計量 (平均値および 99 パーセンタイル値) を示す。

まず、フロー数が少ない条件 (例: 1-1) では、L4S On によりキューイング遅延のテールが抑制され、RTT を増加させても一定範囲では低遅延性を維持できることが確認された。一方、フロー数が増加する条件 (例: 5-5、10-10) では、平均値は比較的 low に保たれる場合があるものの、99 パーセンタイルが上昇しやすく、遅延のスパイク (瞬間的な混雑) を完全には抑制できない条件が存在した。

表 1: Jain's Fairness Index の評価結果

条件	L4S Off	L4S On
1-1	0.9725	0.9999
1-5	0.9926	0.9946
5-1	0.9811	0.9983
5-5	0.8759	0.9978
1-10	0.9639	0.9977
10-1	0.8456	0.9998
5-10	0.8893	0.9994
10-5	0.7725	0.9998
10-10	0.8910	0.9961

この傾向は、(i) フロー数増加に伴いボトルネックでの競合が激化すること、(ii) RTT が増大すると輻輳通知の往復が遅延し、送信レート制御が遅れて一時的に送信量が過大となりやすいこと、(iii) AQM と輻輳制御の組合せによっては、平均遅延よりもテール遅延が悪化し得ること、に起因すると考えられる。

5. 考 察

L4S On において遅延が低い水準に抑制された要因として、ECN による早期輻輳通知と、DCTCP のような ECN 応答型輻輳制御が組み合わさることで、キューが大きく膨らむ前に送信レートが調整される点が挙げられる。特に RTT が小さい条件では、フィードバックが速く、輻輳の兆候に対する追従性が高いため、キューイング遅延の平均値・テールの双方が抑制されやすい。

一方で、フロー数が多い条件や RTT が大きい条件では、99 パーセンタイルが増加し、RFC9330 の目標値を満たせないケースが見られた。図 4 をもとに、RFC9330 が示す「平均 1 ms 未満かつ 99 パーセンタイルで約 2 ms 未満」という条件を厳密に適用して達成可能なフロー本数を整理する。達成可能な上限は RTT=5 ms では 1-1 まで、RTT=10 ms では 2-2 まで、RTT=15 ms では 1-2 および 2-1 までが限界であった。一方で、条件を緩やかに捉え、平均 1 ms 未満は維持したままテール側を 99 パーセンタイルで 2.5 ms 以下に緩和すると、到達可能なフロー本数の上限は拡大する。緩和した場合の解釈では、RTT=5 ms において 3-2、4-1 まで、RTT=10 ms において 3-2 まで、RTT=15 ms において 3-2 までが限界であった。以上より、本実験では「平均値」よりも「テール」が許容フロー数を規定しており、テール要件を 0.5 ms 緩めるだけで RTT=5 ms では最大同時フロー数が顕著に増えることが確認できる。また、特筆すべき点として、競合するフロー数が増加した際に、かえって L4S のテール遅延が減少する逆転現象が一部の条件で確認された。通常、競合フロー数の増加はボトルネックの混雑度を高め、遅延を増大させると考えられる。しかし、本実験結果は、FQ-CoDel によるフロー分離とスケジューリング、および CUBIC と DCTCP という異なる輻輳制御アルゴリズムが混在する環境において、Classic フローの増加が必ずしも L4S の遅延悪化に直結しない、あるいは特定の負荷条件下では DCTCP のウィンドウ制御がより安定的に機能する可能性を示唆している。この逆転現象が発生する詳細なメカニズムについては、FQ-CoDel のサブキュー間のスケジューリング干渉や、バッファ占有率の変化に伴う ECN マーキング頻度の推移などを解析する必要があるが、現状では複雑な相互作用が推測され、本研究の範囲を超える。したがって、この現象の解明および条件の特定は、今後の課題とする。

FQ-CoDel はフロー間のキュー分離を行うため、原理的には他フローの影響を受けにくい。しかし、フロー数の増加はボトルネックリンク帯域の 1 フローあたりの割当量を減少させるため、パケット送出完了までの時間が相対的に増加する。加えて、ラウンドロビンスケジューリングにおいて、多数のフローが同時にバックログを持つ場合、自身のキューがサービスされるまでの巡回待ち時間 (Scheduling Latency) が増加し、これがテー

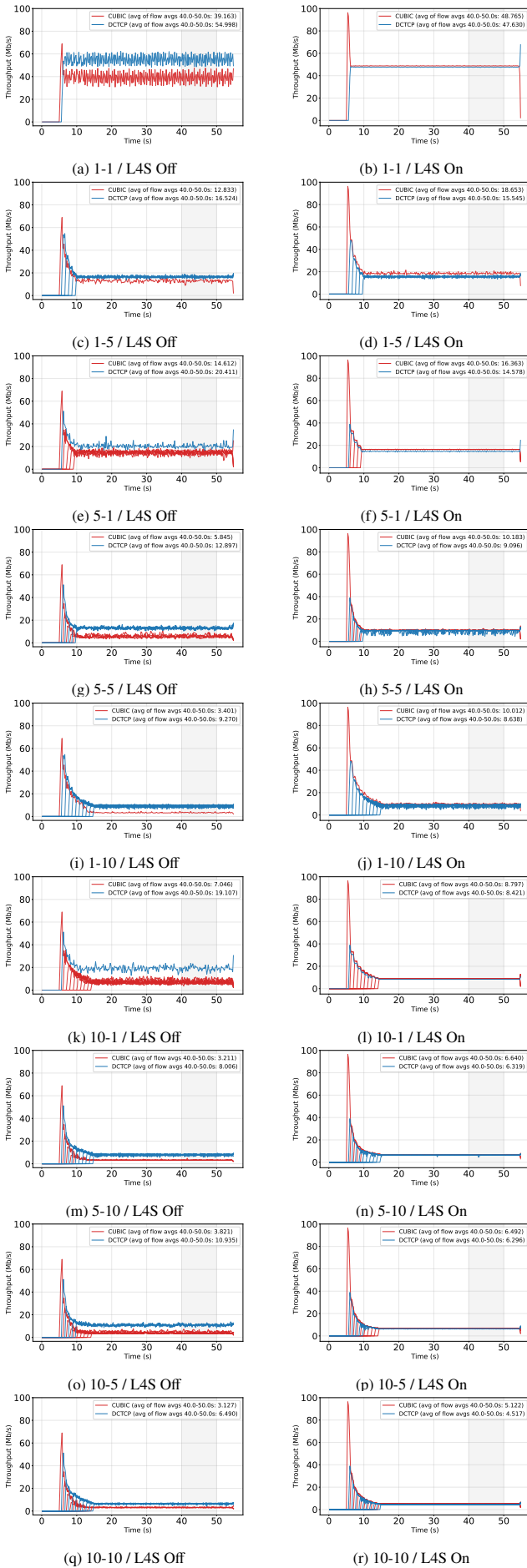


図 2: スループットの時間変化の比較 (L4S Off / On)

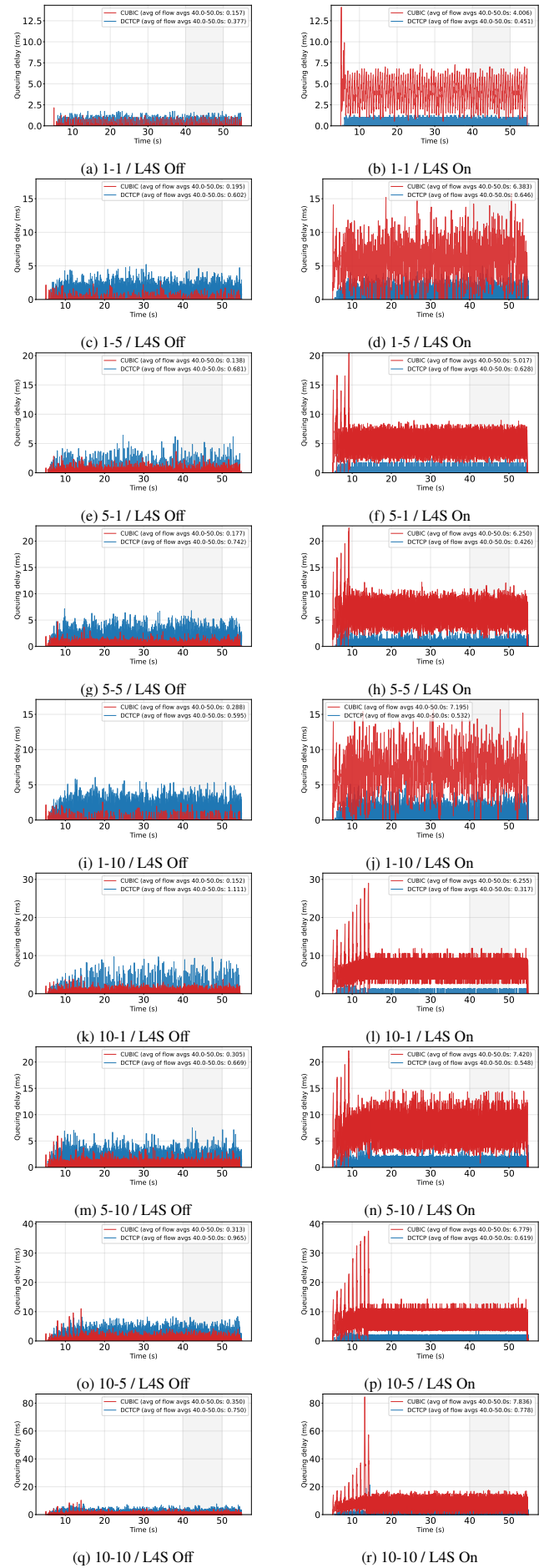


図 3: キューイング遅延の時間変化の比較 (L4S Off / On)

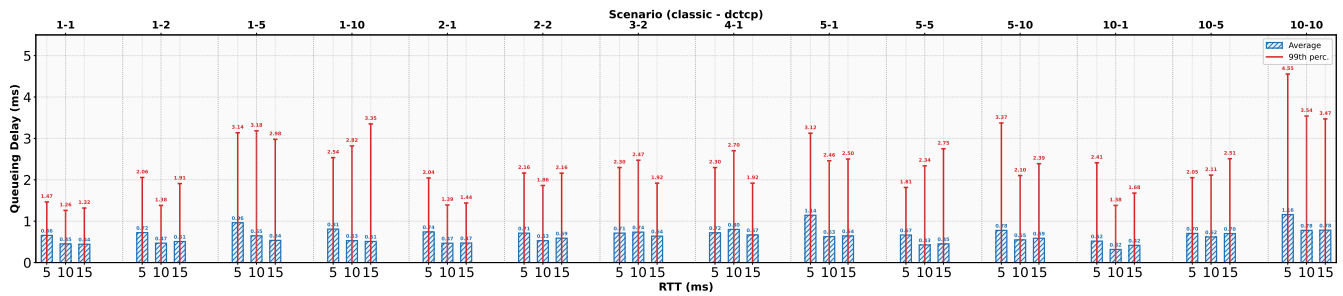


図 4: フロー数がキューイング遅延の平均値と 99 パーセンタイル値に与える影響

ル遅延として現れたと考えられる。また、RTT の増加はフィードバック遅延を引き起こし、過渡的なバーストに対するウィンドウ制御の遅れを生じさせる点も要因である。

6. おわりに

本研究では、FQ-CoDel を用いた L4S について、多数フロー競合環境下での性能評価を行った。実験の結果、L4S は高いスループット公平性を実現する一方で、フロー数や RTT の条件によってはテール遅延の抑制が課題となることが明らかになった。

既存研究では「Classic 1 フロー + L4S 1 フロー」の単純条件に基づく評価が多いのに対し、本研究はフロー数と RTT を系統的に変化させ、実ネットワークに近い競合条件での挙動を確認した点に意義がある。特に、低遅延性が発揮される条件には制約がある（フロー数や RTT に依存する）ことを明らかにした点は、L4S の適用領域の特定や、実運用に向けた設計（対象アプリの選定、アクセス網 RTT など）において重要な知見となる。

本研究における課題として、(i) 評価トポロジが単一ボトルネックであること、(ii) トラフィックモデルが TCP によるバルク転送のみであり、リアルタイム系（WebRTC 等）の混在を直接扱っていないこと、(iii) AQM や輻輳制御の選択肢（例：DualPI2、FQ-PIE、TCP Prague 等）を横断した比較が限定的であること、が挙げられる。

今後は、無線区間を含む環境（変動 RTT・変動帯域）、クロストラフィック（短フロー、バースト、アプリ混在）、および AQM/輻輳制御パラメータ感度を含めて評価し、L4S が「どの条件で確実に低遅延を提供できるか」をより実運用に近い形で明確化する必要がある。

文 献

- [1] B. Briscoe, K.D. Schepper, M. Bagnulo, and G. White, “Low latency, low loss, and scalable throughput (L4S) internet service: Architecture,” IETF RFC 9330, Jan. 2023.
- [2] Ericsson - Deutsche Telekom White Paper, “Enabling time-critical applications over 5G with rate adaptation,” <https://www.telekom.com/en/company/details/enabling-time-critical-applications-over-5g-with-rate-adaptation-628058>, May 2021.
- [3] M. Jarschel, D. Schlosser, S. Scheuring, and T. Hoßfeld, “Gaming in the clouds: QoE and the users’ perspective,” Mathematical and Computer Modelling, vol.57, no.11, pp.2883–2894, June 2013.
- [4] D. BoruOljira, K.-J. Grinnemo, A. Brunstrom, and J. Taheri, “Validating the sharing behavior and latency characteristics of the L4S architecture,” ACM SIGCOMM Computer Communication Review, vol.50, no.2, pp.37–44, May 2020.
- [5] S. Nádas, G. Gombos, F. Fejes, and S. Laki, “A congestion control in-

dependent L4S scheduler,” Proc. ACM Applied Networking Research Workshop (ANRW 2020), pp.45–51, July 2020.

- [6] F.B. Sarpkaya, A. Srivastava, F. Fund, and S. Panwar, “To switch or not to switch to TCP prague? incentives for adoption in a partial L4S deployment,” Proc. ACM Applied Networking Research Workshop (ANRW 2024), pp.45–52, July 2024.
- [7] L. Hagen and A. Brunstrom, “Reproducing and solving a fallback issue in TCP prague,” Proc. ACM Applied Networking Research Workshop (ANRW 2025), pp.92–98, July 2025.
- [8] “ns-3 network simulator,” <https://www.nsnam.org/>. Accessed: 2026/01/18.
- [9] G.F. Riley and T.R. Henderson, “The ns-3 network simulator,” Modeling and Tools for Network Simulation, pp.15–34, Feb. 2010.
- [10] T. Hoiland-Jorgensen, P. McKenney, D. Taht, J. Gettys, and E. Dumazet, “The flow queue CoDel packet scheduler and active queue management algorithm,” IETF RFC 8290, Jan. 2018.
- [11] S. Ha, I. Rhee, and L. Xu, “CUBIC: A new TCP-friendly high-speed TCP variant,” ACM SIGOPS Operating Systems Review, vol.42, no.5, pp.64–74, July 2008.
- [12] L. Xu, S. Ha, I. Rhee, V. Goel, and L. Eggert, “CUBIC for fast and long-distance networks,” IETF RFC 9438, Aug. 2023.
- [13] M. Alizadeh, A. Greenberg, D.A. Maltz, J. Padhye, P. Patel, B. Prabhakar, S. Sengupta, and M. Sridharan, “Data center TCP (DCTCP),” Proc. ACM SIGCOMM 2010 (SIGCOMM 2010), pp.63–74, Aug.-Sep. 2010.
- [14] S. Bensley, D. Thaler, P. Balasubramanian, L. Eggert, and G. Judd, “Data center tcp (DCTCP): TCP congestion control for data centers,” IETF RFC 8257, Oct. 2017.
- [15] D. Brunello, I. Johansson S, M. Ozger, and C. Cavdar, “Low latency low loss scalable throughput in 5G networks,” Proc. IEEE Vehicular Technology Conference (VTC2021-Spring), pp.1–7, April 2021.
- [16] J. Son, Y. Sanchez, C. Hellge, and T. Schierl, “Adaptable L4S congestion control for cloud-based real-time streaming over 5G,” IEEE Open Journal of Signal Processing, vol.5, pp.841–849, May 2024.
- [17] S. Floyd, D.K.K. Ramakrishnan, and D.L. Black, “The addition of explicit congestion notification (ECN) to IP,” IETF RFC 3168, Sept. 2001.
- [18] O. Albisser, K. De Schepper, B. Briscoe, O. Tilmans, and H. Steen, “DUALPI2: Low latency, low loss and scalable throughput (L4S) AQM,” Proc. Technical Conf. Linux Networking (Netdev 0x13), pp.1–8, March 2019.
- [19] 妹尾龍馬, 平澤崇佳, 森弘樹, 中務諭士, 高橋賢, “超低遅延サービスの提供を見据えた遅延抑制技術 CoDel の適用性評価,” 信学技報, vol.119, no.460, pp.129–134, March 2020.
- [20] A.A. Mahawish and H.J. Hassan, “Survey on: A variety of AQM algorithm schemas and intelligent techniques developed for congestion control,” Indonesian Journal of Electrical Engineering and Computer Science, vol.23, no.3, pp.1419–1431, Sept. 2021.
- [21] R. Al-Saadi, G. Armitage, J. But, and P. Branch, “A survey of delay-based and hybrid TCP congestion control algorithms,” IEEE Commun. Surveys & Tutorials, vol.21, no.4, pp.3609–3638, Nov. 2019.
- [22] K.D. Schepper and B. Briscoe, “The explicit congestion notification (ECN) protocol for low latency, low loss, and scalable throughput (L4S),” IETF RFC 9331, Jan. 2023.