

Android 端末における無線環境と有線環境を考慮した TCP 実装による通信制御

松野 瑛南[†] 山口 実靖^{††} 神山 剛^{†††} 小口 正人[†]

[†] お茶の水女子大学 〒 112-8610 東京都文京区大塚 2-1-1

^{††} 工学院大学 〒 163-8677 東京都新宿区西新宿 1-24-2

^{†††} 長崎大学 〒 852-8521 長崎県長崎市文教町 1-14

E-mail: [†]ena@ogl.is.ocha.ac.jp, oguchi@is.ocha.ac.jp, ^{††}sane@cc.kogakuin.ac.jp, ^{†††}kami@nagasaki-u.ac.jp

あらまし スマートフォンやパソコンの市場拡大や、無線通信の高速化により、以前では不可能であったライブ配信の視聴や、大容量データの送受信が日常的に使用されている。しかし、輻輳が発生する事例は今現在多く存在している。原因の一部に無線環境側では、多数のユーザが同じサイトにアクセスすると、AP に膨大なパケットが蓄積し輻輳が発生する事例、有線環境側では、ボトルネックとなり輻輳が発生する事例がある。そこで本研究では、先行研究で開発された輻輳回避のミドルウェアを改良し、無線側と有線側で発生する輻輳を回避するミドルウェアの構築を目標としている。これを実現するために本論文では、通信性能が異なる複数の端末を同時通信した場合に、各端末に最適な通信を考慮した TCP チューニングを行い、評価する。

キーワード 無線規格、通信制御、Android

1 はじめに

近年、スマートフォンやパソコンの市場の拡大により、手軽にインターネットに接続することが可能になった。さらに、無線規格の進化などにより、より高性能になったスマートフォンが高頻度で更新され、端末の種類も豊富である。

従来の携帯電話の使用方法は、メールや電話のみだったが、先述したようにスマートフォンは高性能になり、端末上でライブ配信の視聴、オンライン授業や会議の参加、データ保存など大容量のデータ通信など様々な方法の使用が可能になった。これらの通信は全て無線環境で行われるため、ネットワークトラフィック量が急激に増加している。特に、コロナウイルスの影響で学校や企業リモートワークを導入した為、トラフィック量が前年比の 2 倍になっている [1]。

現在、先に述べたように、無線通信の高速化により、無線環境、有線環境ともに帯域が広がり通信速度が向上している。しかし、図 1 からスマートフォンなど日常的に使用する無線端末の通信量は年々増加していることがわかる [2]。以上のことから、無線環境側で輻輳が発生し、これを回避するための制御が必要になる。このような状況に対する制御手法として、先行研究の輻輳制御ミドルウェアがある [3] [4]。輻輳制御ミドルウェアは、TCP 通信を行っている同一アクセスポイント (以下、AP) 上の全ての端末がブロードキャスト通信によって情報を通知し合い、少なくとも一台の端末の RTT 値が大幅に上昇すると、全端末においてミドルウェアによる制御を開始するツールである。また、AP 周りでの ACK パケットの蓄積を回避する手法により、全体の通信速度と公平性の向上を実現している。

また、有線環境側では無線側に比べ、機器の交換頻度が少な

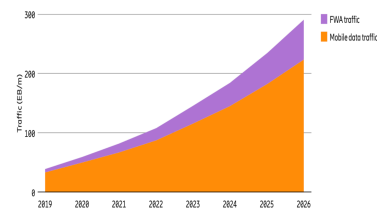


図 1 ネットワークトラフィック量の推移

く、取り替えが困難である事例が発生している。一方で、無線通信側は、交換する機器としては AP やスマートフォンなどが挙げられるが、AP は前に使用していた線をそのまま差し直すだけでとても簡単である。また、スマートフォンは毎年のように新しい種類のスマートフォンが発売されており、機種変更すると安く買える購入パッケージもあるため、取り替えは比較的簡単である。このような有線環境の事例に加えて、無線通信の高速化や機種の交換が進むと、無線環境側では、通信性能の高い AP が設置されるのに対して従来の通信速度の環境が引き続き使用される可能性が高い。結果として、有線環境よりも無線通信の方が通信速度が速くなり、有線側のボトルネックの影響で輻輳が発生する。先行研究の輻輳制御ミドルウェアは、有線側のボトルネックによる輻輳発生時の制御を考慮していない。また、先行研究の実験の際に使用していた Android 端末の OS も更新されており、最新の実験環境や、最新の端末に輻輳制御ミドルウェアを導入した場合正常に動作しているか確認する必要がある。

以上のことから本研究では、無線側、有線側の両方がボトルネックとなって発生する輻輳を制御するツールの構築を目標としている。本稿では、複数の Android 端末の同時通信を行い、

通信速度が異なる 2 つの有線環境を構築し、実験を行った。有線環境の通信速度の違いによって、輻輳制御ミドルウェアの制御に違いが出るのか、正しく制御されているかを合計通信速度と公平性の観点から評価を行った。さらに、公平性の指標である Fairness Index の問題点を見出し、その改善を目指すために、通信性能によって重みをつけた改良版 Fairness Index を提案する。提案した指標と Fairness Index を比較するために評価を行った。

2 関連研究

2.1 カーネルモニタ

通常、カーネル内部の処理は、バックグラウンドで処理されるため監視することができない。そこで先行研究 [5] でカーネルモニタというツールが開発された。これは、TCP ソースコードにモニタ関数を挿入しカーネルで再コンパイルすることで、TCP パラメータをモニタ可能にするツールである。モニタできる値は、輻輳ウィンドウ、ソケットバッファのキュー長、重複 ACK、SACK 受信、タイムアウト検出など各種エラーイベントの発生タイミング等がある。

このツールを組み込み機器である Android 端末上で実装することで、TCP 通信におけるカーネル内部の処理をユーザ空間からリアルタイムにモニタや解析し、TCP パラメータの値の変化を記録することが可能となる。

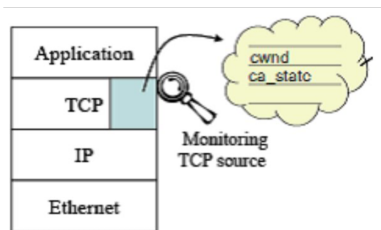


図 2 カーネルモニタ

2.2 輻輳制御ミドルウェア

先行研究で開発された [3] [4] 輻輳制御ミドルウェアは、カーネルモニタをベースとしたシステムである。同一のアクセスポイントに接続した Android 端末間を連携し、各端末の接続状況を把握することで、通信速度の高速化と公平性の向上を可能にする。具体的には、各端末の通信状況や RTT を計測する。その情報から利用可能な帯域幅を計算し、帯域幅を上回るパケットを送信するのを防ぐために輻輳ウィンドウの上限値を自動で算出、補正する。これにより、端末間で可用帯域を公平に分け合い、無線 LAN アクセスポイントにおける ACK パケットの蓄積を回避する。

2.3 スマートフォンにおける輻輳制御

Android OS はカーネルに Linux カーネルを用いており、輻輳制御アルゴリズムに CUBIC TCP [6] を用いている。また、近年は新しい TCP 輻輳制御アルゴリズムとして TCP BBR [7] が提案され今後の普及が期待されている。ただし、CUBIC TCP

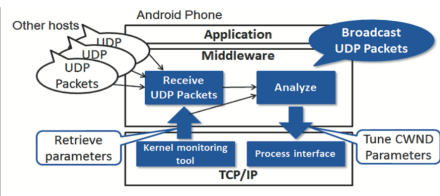


図 3 輻輳制御ミドルウェア

と TCP BBR が共存して通信を行った場合は両 TCP のスループットの公平性が低くなることが示されており [8] [9] [10]、今後は複数種類の輻輳制御手法や端末が混在する環境における性能の公正性の考察が重要であると考えられる。特にモバイル環境における不公平性 [11] が示されており、スマートフォン端末などにおける性能公平性の改善が重要であると考えられる。

3 実験

3.1 実験環境

表 1 に実験に使用した機器の性能を、図 4 に本研究で構築した実験環境を示す。Android 端末と AP 間は無線接続環境、AP とサーバ間是有線接続環境である。また、AP とサーバ間に転送速度 100Mbps のスイッチングハブを設置し、1Gbps 環境と 100Mbps 環境の通信速度が異なる環境を構築した。本研究では、クライアントとしてスマートフォン 2 台、タブレット 2 台、サーバとして Ubuntu Linux を OS とするデスクトップ機を実験に用いた。本稿では、Pixel4 はスマートフォン 1、Pixel5 はスマートフォン 2、Nexus7 の 2 台のタブレット 1、2 と記す。

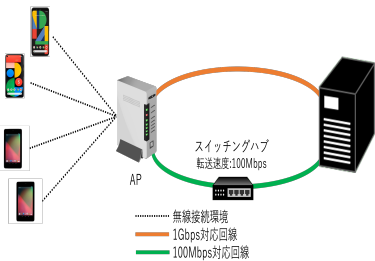


図 4 有線環境 100Mbps

表 1 実験機器の性能

Android	Model number	Pixel4	Pixel5	Nexus 7(2013)
	OS	Android 10	Android 11	Android6
	CPU	Snapdragon 855	Snapdragon 765G	Snapdragon S4 Pro
	Memory	6 GB	8 GB	2 GB
	WLAN	Wi-Fi 802.11 a/b/g/n/ac	Wi-Fi 802.11 a/b/g/n/ac	Wi-Fi 802.11 a/b/g/n
Server	OS	Ubuntu 20.04 LTS		
	CPU	Intel(R)Core(TM) i7-10400		
	Main Memory	32GB		
AP	Model	WSR-2533DHP2		
	Support Format	IEEE 802.11a/b/g/n/ac		
	Frequency Band	5 GHz		

3.2 予備実験

予備実験として、本実験で使用する端末の通信性能を計測した。図5は有線環境 100Mbps の通信性能、図6は有線環境 1Gbps の通信性能を示す。スマートフォン2台とタブレットは、無線規格、AndroidOS が異なるため、有線環境 100Mbps、1Gbps で平均スループットの値が大幅に異なった。

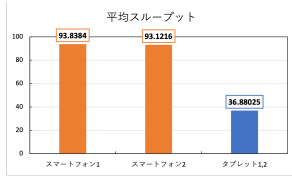


図5 有線環境 100Mbps

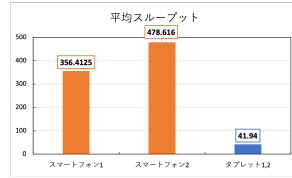


図6 有線環境 1Gbps

3.3 実験概要

本稿では、先行研究の輻輳制御ミドルウェアが本実験環境で適切に動作するか評価を行う。具体的には、4台のAndroid端末とサーバ間でiperf[12]を用いた同時通信を行う。そして、合計通信速度と公平性を測定し輻輳制御ミドルウェアの制御がある場合とない場合の比較を行う。

公平性を評価する評価指標として、Jainのfairness index[13]が主に使用されている。次式で定義されており、 x は取得したスループット、 n は標本数である。

$$F_j(X_n) = \frac{(\sum_{i=1}^n x_i)^2}{n \sum_{i=1}^n x_i^2} (1 \leq k \leq 1)$$

さらに、輻輳制御ミドルウェアの制御に効果がほぼない1Gbps環境では、RTTとカーネルモニタから取得したTCPパラメータの輻輳ウィンドウと各種エラーイベントを取得し各端末の動作を解析、比較を行った。実験で使ったカーネルモニタのエラーコードを表2に示した。

表2 カーネルモニタのエラーコード

番号	発生条件
1番	パケット順序の入れ替わり検出
2番	送信側のバッファが溢れることによるエラーの検出
3番	重複ACK, SACKなどの受信
4番	タイムアウトによる検出

4 実験結果

4.1 輻輳制御ミドルウェアの性能評価実験

先行研究によって開発された輻輳制御ミドルウェアの性能を評価する。

4.1.1 100Mbps 環境

実験結果として、有線環境が 100Mbps での平均スループットを図7に示す。図8は、通信速度と公平性を散布図で評価した。縦軸が公平性 (Fairness Index)、横軸が合計通信速度 (Mbps) を示す。また、色によって、デフォルト通信、先行研究のミドルウェアを区別する。

実験結果から、デフォルト通信では、図5の通信性能の高い順にスループットの値が高くなる結果となった。一方で、先行研究の輻輳制御ミドルウェアの場合は、全端末の平均スループットの値がおおよそ20あたりであった。図8からデフォルト通信に比べ、先行研究のミドルウェアがグラフの右上に分布している。これは、先行研究の輻輳制御ミドルウェアが合計通信速度、Fairness Index による公平性ともに向上していること示す。先行研究の輻輳制御ミドルウェアは、通信速度と公平性の両方の向上を目指しているため、100Mbps 環境では輻輳制御ミドルウェアの制御が適切に動作していることが確認できた。

4.1.2 1Gbps 環境

有線環境が 1Gbps での平均スループットを図9に示す。本実験結果では、通信速度と公平性の両方に着目するために、散布図で実験結果を評価する(図10)。縦軸が公平性 (Fairness Index)、横軸が合計通信速度 (Mbps) を示す。また、色によって、デフォルト通信、先行研究のミドルウェアを区別する。図10から、デフォルト通信、先行研究の輻輳制御ミドルウェアともにほぼ同じ位置に分布している。これは、先行研究の輻輳制御ミドルウェアとデフォルト通信が合計通信速度、Fairness Index による公平性がほぼ変わらない結果であること示す。先行研究の輻輳制御ミドルウェアは、通信速度と公平性の両方の向上を目指し制御を行っているが、1Gbps 環境下では輻輳制御の効果がなく正常に動作してない結果となった。

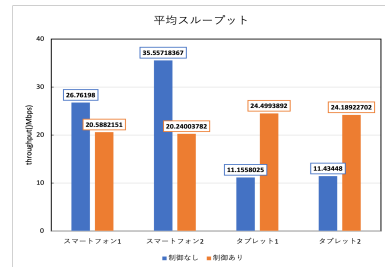


図7 有線環境 100Mbps の平均スループット

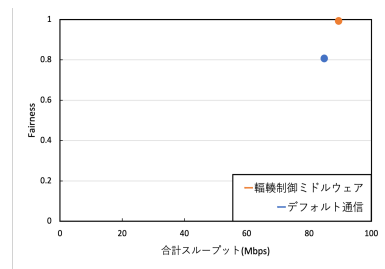


図8 有線環境 100Mbps の公平性と合計通信速度の評価

4.2 1Gbps 環境でのパラメータ

1Gbps 環境下でも先行研究の輻輳制御ミドルウェアによる制御の効果が発揮されるよう改良を行うため、各端末で TCP パラメータと RTT を取得し、評価を行った。

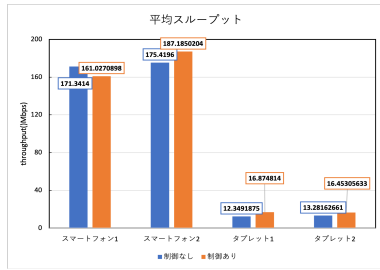


図 9 有線環境 1Gbps の平均スループット

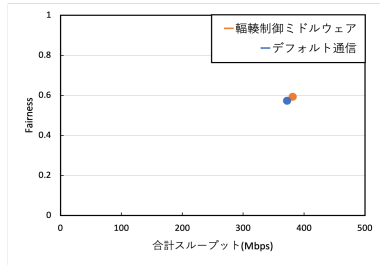


図 10 有線環境 1Gbps の公平性と合計通信速度の評価

4.2.1 TCP パラメータ

図 11,12 は、スマートフォン 2 台の TCP パラメータを、図 13,14 はタブレット 2 台の TCP パラメータの振る舞いを示している。本実験で比較した TCP パラメータは、エラー状態 (ca-state) と輻輳ウィンドウ (cwnd) である。

タブレット 2 台で一番多く発生している 2 番のエラーは、送信側のバッファが溢れることによるエラーの検出である。これは、通信経路にある AP でもバッファ溢れが発生する為、スマートフォン 2 台にも影響を与えると推測する。しかし、スマートフォン 2 台は一度もエラーが発生していない結果となった。この結果と、無線規格や Android OS も異なることから、スマートフォン 2 台とタブレット 2 台は端末の制御に違いがあると推測される。

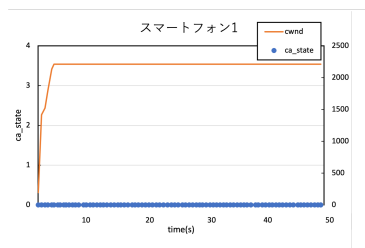


図 11 スマートフォン 1

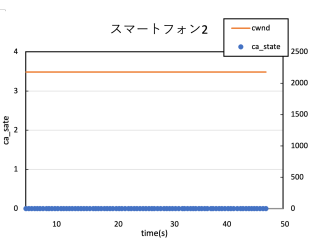


図 12 スマートフォン 2

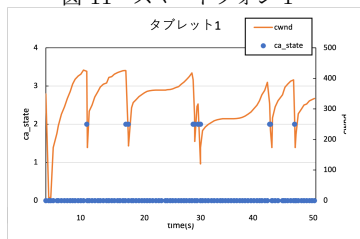


図 13 タブレット 1

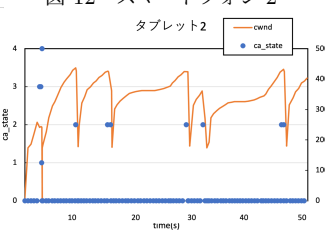


図 14 タブレット 2

4.2.2 RTT

図 15 に、同時通信実験時の各端末の RTT を示した。タブレット 2 台はスマートフォン 2 台と比較し、RTT が倍近く高い値となった。端末同士の物理的な距離の違いもなく、AP から先の有線経路は全端末同じ経路を通っていることから、スマートフォンとタブレットの RTT の値の差は端末の制御に違いがあると推測される。先に述べたエラーの振る舞いや通信規格などの端末の通信性能の違い、さらに RTT の値が異なる点から、スマートフォン 2 台とタブレット 2 台の AP の待ち行列が別々であることが分かった。

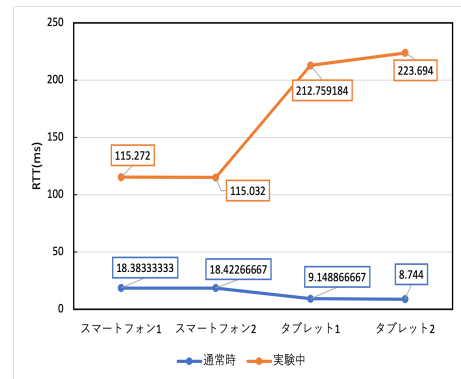


図 15 各端末の RTT 値の比較

5 改良版 Fairness index の提案

5.1 既存の Fairness index における問題点

既存の評価指標 Fairness index の問題点は、端末数と取得したスループットから公平性を求めるという点である。この指標は、性能の違いに関わらず複数端末を同時通信した際に各端末が取得したスループットが同等のスループットを取得しているかどうかを評価する指標となっている。しかし本実験環境では、前章に記述したように、通信性能が異なる端末 4 台をクライアントとし、実験を行った。図 6 からスマートフォン 1 とスマートフォン 2 の差も大きい、スマートフォン 2 とタブレットの平均スループットの差はおよそ 10 倍である。この環境で同時通信をした場合、スマートフォン 2 とタブレットが同じスループット値を取れないのは明白である。つまり、既存の評価指標は、通信性能が異なる端末同士で通信を行うと、輻輳制御ミドルウェアを導入しても公平性が向上することはなく、合計通信速度の値の増加もほぼ変わらないといった問題が発生することがわかった。

5.2 評価指標について

本実験で使用している二つの指標は、一つの評価指標が向上すると、もう一つの評価指標の値が下がってしまうトレードオフの関係である。通信性能が異なる端末で実験を行う場合、二つの評価指標が高くなるにはどのように制御すべきか、また制御した結果を解析した。

5.2.1 合計通信速度重視

通信をする際に重要になるのは、通信速度の速さである。複数端末の同時通信も同様に重要である。本研究の実験環境で合計通信速度を高めるためには、スマートフォン2台のスループットを高めるよう制御を行う。これは、取得したスループット値(図9)と最大スループット値(図6)の差が、タブレットよりスマートフォンの方が大きいことから、制御を行うと合計通信速度が高くなると予測できるためである。

TCP チューニングを行い、タブレット2台の輻輳ウィンドウの値を抑える制御を行うことにより、前章と同様の実験を行った。具体的には、タブレット2台の輻輳ウィンドウ値を10, 50, 100と三段階で実験を行った。本実験の有線環境は、1Gbps環境である。図16は、TCPチューニングを行った際の実験結果の各端末の平均スループットの値を示した。図17は、図8と同様に通信速度と公平性を散布図で評価した。縦軸が公平性(Fairness Index)、横軸が合計通信速度(Mbps)を示す。また、青い点はミドルウェアを用いずにデフォルト通信を行った結果、オレンジの点は先行研究のミドルウェアを用いた結果、緑の点はTCPチューニング変更を用いた結果を示す。またマークによって、輻輳ウィンドウ変更1, 2, 3を区別する。図17から、合計通信速度を重視しTCPチューニングを行った結果、デフォルト通信と先行研究の輻輳制御ミドルウェアに比べ、合計通信速度は向上しているが、公平性が著しく下がっていた。このことから、タブレット2台の輻輳ウィンドウ値を下げ、平均スループットを減少させることで、合計通信速度が向上することが確認できた。しかし、公平性の面でFairness indexの値が低く、あまり良い制御とはいえない。輻輳ウィンドウの値を下げる程Fairness index 0.5の値に近づいている。そこまで悪くない値に見えるが、これはスマートフォン2台とタブレット2台の値が極端に異なることを示す。

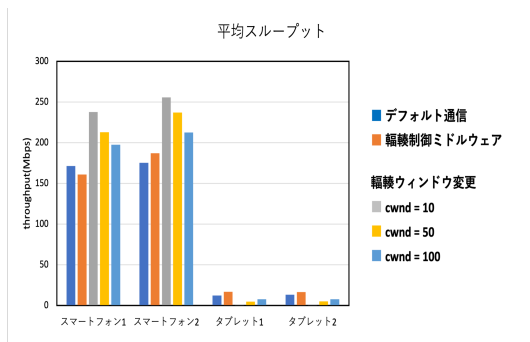


図 16 合計通信速度を重視した制御の平均スループット

5.2.2 公平性重視

近年日常で使用する端末数が増加している。輻輳の発生を回避するために、端末の制御を行う上で公平性は非常に重要となる。図9から、スマートフォン2台のスループット値を下げ、タブレット2台の値を上げるよう制御を行うと公平性は高くなると推測した。

実際に上記のような制御をTCPチューニングで環境を構築し、実験を行った。具体的には、スマートフォン2台の輻輳ウィ

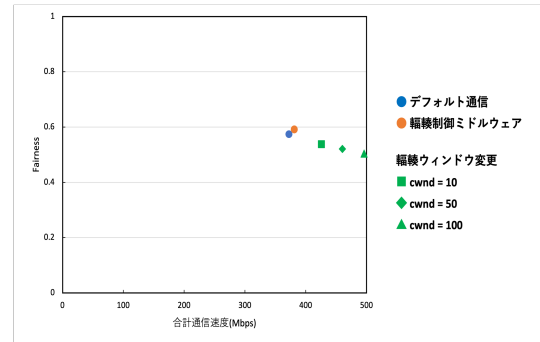


図 17 合計通信速度を重視した公平性と合計通信速度の散布図

ンドウの値を下げると、タブレット2台が取れる帯域が広がり、公平性が高くなる制御を行った。スマートフォン2台の輻輳ウィンドウ値を100, 500, 1000, 1500と四段階に設定して実験を行った。前節同様、本実験の有線環境は、1Gbps環境である。実験結果は、図18に示した。図19は、図8と同様に通信速度と公平性を散布図で評価した。色によって、デフォルト通信、先行研究のミドルウェア、輻輳ウィンドウ変更を区別する。またマークによって、輻輳ウィンドウ変更1, 2, 3, 4を区別する。

いずれの結果も、デフォルト通信や先行研究のミドルウェアはグラフの右下に分布しているのに対し、輻輳ウィンドウ値を変更した場合は、徐々に左上に分布し、公平性は向上したが合計通信速度が減少した結果となった。このことから、公平性を重視した制御は、スマートフォン2台の輻輳ウィンドウの値を下げていくとスマートフォン2台の平均スループット値が減少し結果として公平性の向上することが確認できた。しかし、使用可能な帯域をほぼ使わない制御手法となり、前節同様より良い制御を行うためには、合計通信速度も考慮する必要がある。

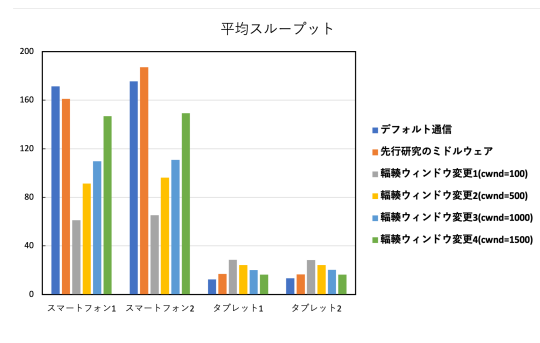


図 18 公平性を重視した制御の平均スループット

5.3 提 案

前節を踏まえて本実験のような環境にも適応する公平性の指標について、通信性能に重みをつけた改良版の fairness index を提案する。この指標は前述した fairness index の式に取得したスループット値をそのまま代入せず、取得したスループットを最大スループットの値で割った値を代入する手法である。具体的には、各端末の最大スループットに対して取得したスループットがどのくらいの割合なのかを算出し、その割合から公平

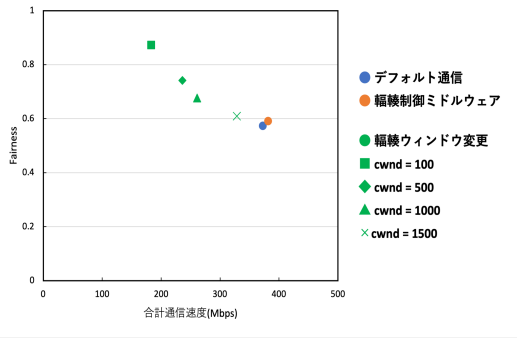


図 19 公平性を重視した公平性と合計通信速度の散布図

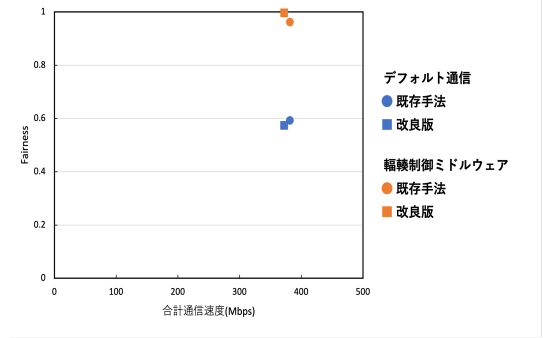


図 20 改良版 Fairness index の公平性と合計通信速度の散布図

性を導いていく，Max を各端末の最大スループット値とした場合，改良版の Fairness index は下記の式のようになる．

$$F_j(X_n) = \frac{\left(\sum_{i=1}^n \frac{x_i}{Max_i}\right)^2}{n \sum_{i=1}^n \frac{x_i}{Max_i}} (1 \leq k \leq 1)$$

この改良版 Fairness index は，通信性能が異なる端末の同時通信した時も通信性能ごとに比較するため，前節で述べた二つの指標のうち片方重視した制御と比較し，公平性合計通信速度共により良い値になる．本研究では，この改変によって，有線環境の通信速度が 1Gbps の場合の公平性の問題を解決することを目指した．

6 改良版 Fairness index の評価

1Gbps 環境でも輻輳制御ミドルウェアによる制御の効果が発揮されるよう改良を行うため，各端末で TCP パラメータと RTT を取得し，評価を行った．前節と同様の実験結果を用いて改良版 Fairness index の評価を行った．本実験結果のうち，各端末の平均スループットは，図 9 と同様である．通信速度と公平性の両方に着目するために，散布図で実験結果を評価する (図 20)．縦軸が公平性 (Fairness Index)，横軸が合計通信速度 (Mbps) を示す．また，青い点はミドルウェアを用いずにデフォルト通信を行った結果，オレンジの点は先行研究のミドルウェアを用いた結果を示す．また，マークによって，Jain の Fairness index，改良版 Fairness index を区別する．

まず，既存手法の Jain の Fairness index のマークはグラフの中央に分布しており，デフォルト通信，先行研究の輻輳制御ミドルウェアともに公平性はほぼ変わらずあまり高くない結果となった．一方で，改良版 Fairness index は，デフォルト通信，先行研究の輻輳制御ミドルウェアともにグラフの右上に分布している．これは，改良版 Fairness index で通信性能を考慮した影響により，公平性が向上した．

7 まとめと今後の予定

本稿では，先行研究の輻輳制御ミドルウェアが様々な環境に適応するか確かめる為に，通信性能が異なる複数の端末で iperf 通信の同時実験を行った．

100Mbps 環境下では，高い公平性が得られ，合計通信速度

も向上したことから輻輳制御ミドルウェアの効果があつた．しかし，1Gbps 環境では，Fairness index，合計通信速度ともに制御していない場合とほぼ変わりなく，制御に効果がなかった．その原因解明のために，カーネルモニタ TCP パラメータのエラー状態と輻輳ウィンドウ，RTT を取得し，各端末の振る舞いの比較を行った．結果は，タブレット 2 台のエラー発生と，スマートフォン 2 台のエラー発生の振る舞いが異なっていた．また，スマートフォン 2 台とタブレット 2 台で RTT の値も異なる結果となった．このことから，スマートフォン 2 台とタブレット 2 台の AP の待ち行列が別々であり，制御の仕方も異なることが確認できた．

1Gbps 環境の場合でも，公平性を保てるような制御を行う必要がある．そこで，図 6 に着目すると，通信性能の違いによって取得できる最大スループットが異なることが判明した．この環境で，Fairness index の値を高くしようと制御を行っても限界がある．そこで，通信性能の違いに着目し，Fairness index の改良を行う．具体的には，Fairness index の式の取得したスループットの値をそのまま代入するのではなく各端末の最大スループットの値で重み付けした値を代入する手法である．改良版 Fairness index は，通信性能を考慮した指標となり，1Gbps 環境下で評価を行った．実験結果は，通常の Fairness index と比較し，改良版 Fairness index の値が高くなっていた．これは，通信性能の違いによって，取得した平均スループットの値は異なるが，端末ごとに最大スループット値に対する割合はほぼ変わらないつまり，改良版 Fairness index で定義した公平性が高いということが分かった．

今後の課題としては，公平性も保ちながら合計通信速度の向上も行う制御の構築を目指している．本稿では，有線環境が 1Gbps の場合改良版 Fairness index の値が高くなったことは確認できたが，合計通信速度の影響はない．合計通信速度と公平性はトレードオフの関係なので，現在の制御が端末 4 台の限界点なのか，もしくは現在の公平性を保ちつつ，より高い合計通信速度を取得できる制御手法があるのか TCP チューニングなど実験を行っていきたい．

また，AP の制御は大変詳細な範囲まで設定が可能である．端末ではなく AP による制御ではどのような影響があるのかも実験を行っていきたい．具体的に一つの例として，バンドステアリング機能という AP が周波数帯を自動で決める設定がある．

AP のデフォルトは、5GHz か 2.4GHz のどちらかの周波数帯を端末からユーザが選択できるようになっているが、この機能を使うと SSID が 1 つになり、ユーザ側からは選択できない設定になる。このような環境下では、周波数帯の幅が広がることによって合計通信速度が向上するのか、もしくは減少するのかを解析、評価を行ってきたい。

文 献

- [1] Coronavirus is helping to drive twice as much traffic as last year to akamai's networks. <https://www.businessinsider.com/coronavirus-akamai-high-internet-traffic-2020-3>. Accessed: 2021-1.
- [2] Ericsson mobility report, november 2020, 2020. <https://www.ericsson.com/en/mobility-report/reports>. Accessed: 2021-5.
- [3] Hiromi Hirai, Saneyasu Yamaguchi, and Masato Oguchi. A Proposal on Cooperative Transmission Control Middleware on a Smartphone in a WLAN Environment. Proc. IEEE WiMob2013, pp. 710-717, October 2013.
- [4] Ai Hayakawa, Saneyasu Yamaguchi, and Masato Oguchi. Reducing the TCP ACK Packet Backlog at the WLAN Access Point. Proc. ACM IMCOM2015, 5-4, January 2015.
- [5] Kaori Miki, Saneyasu Yamaguchi, and Masato Oguchi. Kernel Monitor of Transport Layer Developed for Android Working on Mobile Phone Terminals. Proc. ICN2011, pp.297-302 January 2011.
- [6] Sangtae Ha, Injong Rhee, and Lisong Xu. Cubic: A new tcp-friendly high-speed tcp variant, July 2008.
- [7] Bbr: Congestion-based congestion control. ACM Queue, 14(5):50, 2016.
- [8] M. Hock, R. Bless, and M. Zitterbart. Experimental evaluation of bbr congestion control. In *2017 IEEE 25th International Conference on Network Protocols (ICNP)*, pages 1–10, 2017.
- [9] K. Miyazawa, K. Sasaki, N. Oda, and S. Yamaguchi. Cycle and divergence of performance on tcp bbr. In *2018 IEEE 7th International Conference on Cloud Networking (CloudNet)*, pages 1–6, 2018.
- [10] K. Sasaki, M. Hanai, K. Miyazawa, A. Kobayashi, N. Oda, and S. Yamaguchi. Tcp fairness among modern tcp congestion control algorithms including tcp bbr. In *2018 IEEE 7th International Conference on Cloud Networking (CloudNet)*, pages 1–4, 2018.
- [11] Jae Won Li, Fengand Chung, Xiaoxiao Jiang, and Mark Claypool. Tcp cubic versus bbr on the highway. In Robert Beverly, Georgios Smaragdakis, and Anja Feldmann, editors, *Passive and Active Measurement*, pages 269–280, Cham, 2018. Springer International Publishing.
- [12] iperf. <https://iperf.fr/iperf-download.php>. Accessed: 2021-1.
- [13] D.-M. Chiu and R. Jain. Analysis of the increase and decrease algorithms for congestion avoidance in computer networks. *Computer Networks and ISDN Systems*, vol. 17, pp. 1–14, 1989.