

# Android 端末の無線 LAN 通信時の制御に向けたパケット解析

松野 瑛南<sup>†</sup>    山口 実靖<sup>††</sup>    神山 剛<sup>†††</sup>    小口 正人<sup>†</sup>

<sup>†</sup> お茶の水女子大学    〒 112-8610 東京都文京区大塚 2-1-1

<sup>††</sup> 工学院大学    〒 163-8677 東京都新宿区西新宿 1-24-2

<sup>†††</sup> 長崎大学    〒 852-8521 長崎県長崎市文教町 1-14

E-mail: <sup>†</sup>ena@ogl.is.ocha.ac.jp, <sup>††</sup>oguchi@ogl.is.ocha.ac.jp, <sup>†††</sup>sane@cc.kogakuin.ac.jp, <sup>†††</sup>kami@nagasaki-u.ac.jp

**あらまし** 近年、スマートフォンやパソコンの市場が急速に拡大している。特にスマートフォンは小型のパソコンのような性能の高さから幅広い世代の方々が使用している。その為、無線接続環境での使用が増加している。しかし、無線接続環境は有線接続環境より通信環境が脆弱なため、AP に膨大なパケットが蓄積しパケットロスやトラフィックの輻輳の問題が発生する。本研究では、Android 端末内で機械学習により得た情報に基づいて通信制御を行うことを最終目的としている。本稿では目標達成の為、Android 端末の無線 LAN 通信を行い、パケットのトラフィックの解析を行う。  
**キーワード** 深層学習, 機械学習, Android

## Packet Analysis for Control of Android Terminal in WLAN Communication based on Deep Learning Result

Ena MATSUNO<sup>†</sup>, Saneyasu YAMAGUCHI<sup>††</sup>, Takeshi KAMIYAMA<sup>†††</sup>, and Masato OGUCHI<sup>†</sup>

<sup>†</sup> Ochanomizu University, 2-1-1 Otsuka, Bunkyo-ku, Tokyo 112-8610 Japan

<sup>††</sup> Kogakuin University, 1-24-2 Nishi-shinjuku, Shinjuku-ku, Tokyo 163-8677 Japan

<sup>†††</sup> Nagasaki University, 1-14 Bunkyo-cho, Nagasaki-shi, Nagasaki 852-8521 Japan

E-mail: <sup>†</sup>ena@ogl.is.ocha.ac.jp, <sup>††</sup>oguchi@ogl.is.ocha.ac.jp, <sup>†††</sup>sane@cc.kogakuin.ac.jp, <sup>†††</sup>kami@nagasaki-u.ac.jp

### 1. はじめに

近年、スマートフォンやパソコンで手軽にインターネット接続が可能になり、無線通信環境で使用する端末の市場が急速に拡大している。特にスマートフォンは小型で高性能、高機能の端末であるため一人一台持つことが珍しくない。さらには複数台所持する傾向もある。全世界のスマートフォン市場は、2020年の段階で約 12 億台、2024 年には 14 億台越えと今後さらに増加していくと予想されている。[1]

従来の携帯電話はメールや電話をメインに使用していたが、スマートフォンは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で使用されている。このことから、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がリモートワーク、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。[2]

他にも、同時に大容量データの通信や、多数のユーザが同じサイトにアクセスすることによって、AP(アクセスポイント)に膨大なパケットが蓄積するという問題が発生し、いずれ輻輳を

引き起こしてしまう。この問題解決の為に、大容量通信や通信の高速化のさらなる向上を目指し、5G という第五世代のモバイルネットワークを構築した [3]。5G は IoT などの新しいサービスをサポートし、新しい種類のネットワークの通信を可能にした新しいグローバルワイヤレス規定である。さらに 4G のトラフィック容量とネットワーク効率の 100 倍の増加をサポートしている。しかしながら、今現在 5G の通信環境のエリアは限られており、まだ 5G を構築していない地域もある。[4]

現在、無線通信環境では、5G を代表とした公衆無線通信と、無線 LAN 通信の高速化が進んでいる。無線 LAN 通信は WiFi4, 6 など通信の高速化に向けて新しい規格が出ている。しかし、公衆無線通信では、5G の通信環境エリアの制限や、各基地局が収容する端末数の増加による輻輳の発生、通信に費用がかかるなどの問題が発生している。無線 LAN 通信では、LIVE 配信などアプリケーションが要求する帯域が増加していることなどの問題から輻輳が発生する。

このような背景から、無線 LAN 通信環境の技術は進化しているが、世界中で普及するには時間がかかり、現在使用可能な帯域を分け合うという環境が続くと考えられる。

これに対し、無線環境よりも有線環境でボトルネックが発生する可能性が出てきた。従来は、有線通信環境は帯域が広く、エラーの発生はほとんどが無線通信環境で起こっていた。その為、無線通信環境での制御に関する研究がされてきた。しかし、Android 端末や AP の無線通信環境で使用する機器の通信性能が向上したことにより、無線通信環境下のエラー発生が減少し、無線通信環境の通信速度の高速化が進められている。それに対し、有線通信環境下では、無線に比べ有線機器を最新に交換することや、機器の取り替えが困難であるという問題が出てくる場合がある。その為、有線通信環境が無線通信環境よりも帯域が狭い環境で通信するという事例が発生した。

これらのことに注目し、本研究では、有線接続部分がボトルネックになる場合とそうでない異なる環境を構築し、実験を行う。そして、実験結果からネットワークの状態を観察し、輻輳制御のパラメータを制御するミドルウェアの構築を最終目的とする。本稿では、有線接続部分である AP とサーバ間に 100Mbps(以降、100M と呼ぶ) のスイッチングハブを挿入し、1Gbps(以降、1G と呼ぶ) と 100M 対応の回線が混合している環境下を構築した。そして、全て 1Gbps 対応の回線と、1G と 100M の対応の回線の異なる環境での iperf 通信を行い、カーネルモニタで TCP パラメータの振る舞いを解析する。

## 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 (図 1)。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、TCP パラメータの値の変化を記録が可能となる。

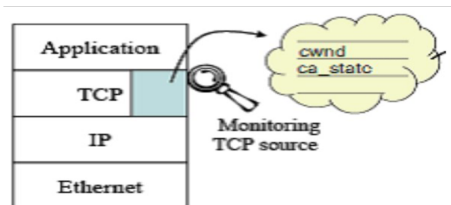


図 1 カーネルモニタ

### 2.2 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 TCPBBR [7] が提案され今後の普及が期待されている。ただし、CUBICTCP

と TCPBBR 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。

### 2.3 輻輳制御ミドルウェア

先行研究で開発された輻輳制御ミドルウェア [12] [13] は、カーネルモニタをベースとしたシステムである (図 2)。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。

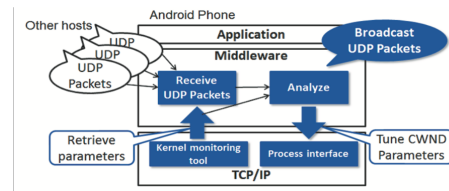


図 2 輻輳制御ミドルウェア

### 2.4 Android

Android 端末は、Google 社が開発した携帯端末向けのソフトウェアプラットフォームで、OS が Linux カーネルがベースである。Android は、世界シェアの約 8 割を占めている。Android 端末の魅力はオープンソースであり、ソースがインターネット上で無料公開されている為、誰でも OS のカスタマイズ、改良が可能であることだ。また、個人が Java 言語を使用してアプリケーションを開発し、Android マーケットで自由に配布することや、ブラウザのカスタマイズも可能である。

### 2.5 深層学習

機械学習は、人間が特徴を定義し、入力された情報に対し出力する処理を行う手法である。しかし、複雑な画像識別など特徴の抽出が難しい場合に機械学習は使用できない。そこで深層学習という、人工知能が学習データから特徴量を抽出する手法が構築された。具体的には、ニューラルネットワークの中間層を人間の脳神経回路をモデルにした多層構造アルゴリズムを用いて特徴量を抽出する。現在、文字認識、自動運転や医療機器に使用されている。次に、深層学習で使用するアルゴリズムを説明する。

#### 2.5.1 深層学習を用いたトラフィック予測

モバイルトラフィックを深層学習を用いて予測する研究として次の様なものがある。Trinh らは、LTE トラフィックを LSTM を用いて予測する手法についての考察をし、時間的相関を取得できることを示している [14]。

### 3. 実験概要

図3は実験環境を表している。Android 端末と AP 間は無線接続環境, AP とサーバ間は有線接続環境である。表1に本実験で使用した機器の性能, 100M のスイッチングハブの性能を記した。本実験環境の有線接続部分である AP とサーバ間に転送速度 100M 対応のスイッチングハブを設置し, 1G 環境と 1G と 100M の環境の異なる環境下を構築した。Android 端末と PC にクライアント, Linux サーバにサーバに設定し, ネットワーク測定ツール iperf [15] を用いて 50 秒間の通信を行い, スループットを測定した。さらに, 端末 5 台の iperf 通信時の TCP パラメータの振る舞いをカーネルモニタで記録した。iperf 通信は 10 回行い, 最大値最小値を除く 8 回の平均を求めた。

カーネルモニタは, スマートフォンとタブレットに搭載している。使用する機器を 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, タブレット 2, Surface は PC とする。

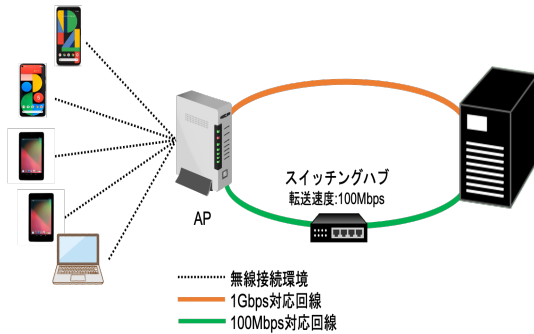


図3 実験環境

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
PC(Surface)	OS	Windows 10 pro		
	CPU	i7-6650U		
	Main Memory	16GB		
	WLAN	Wi-Fi 802.11 a/b/g/n/ac		
server	OS	Ubuntu 20.04 LTS		
	CPU	Intel(R)Core(TM) i7-10400		
	Main Memory	32GB		
AP	Model	WSR-5400AX6-MB		
	Support Format	IEEE 802.11a/b/g/n/ac/ax		
	Frequency Band	5 GHz		
スイッチングハブ	Model	FX-08ES		
	転送速度	100BASE-TX		

### 4. 実験結果

#### 4.1 各端末の通信性能比較

スマートフォン, タブレット, PC の各端末 1 台ずつの通信性能を測定した。図4は全て 1G 対応の回線での結果である。スマートフォンは 356.41Mbps, タブレットは 41.94Mbps, PC は

359.84Mbps となった。また 1G と 100M 対応の回線では, スマートフォンは 93.83Mbps, タブレットは 36.88Mbps, PC は 94.54Mbps となった(図5)。スマートフォンと PC は 802.11ac, タブレットは 802.11n と無線通信方式が異なるため, 平均スループット値に差が出ている可能性がある。さらに図4,5のグラフを比較すると, 各端末 3 台とも全て 1G 対応の回線の方が平均スループットの値が大きいことがわかる。これは, 1G と 100M 対応の回線の環境下では 100M 対応の回線と同等の振る舞いになり, 使用可能帯域が狭まることから, 全て 1G 対応の回線に比べ平均スループット値が下がっている。

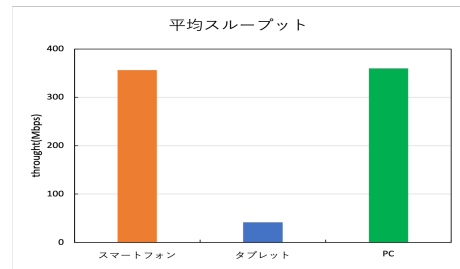


図4 全て 1Gbps 対応の回線

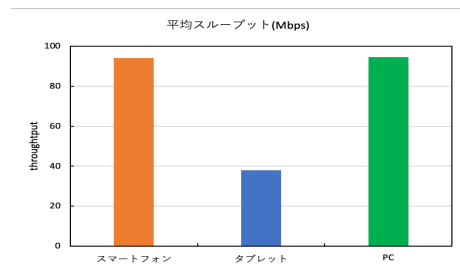


図5 1Gbps と 100Mbps 対応の回線

#### 4.2 各端末における TCP パラメータの比較

##### 4.2.1 全て 1G 対応の回線

図4の実験結果であるスマートフォンとタブレットの端末1台でのカーネルモニタで取得可能な輻輳ウィンドウサイズとエラー状態の振る舞いを記録した。スマートフォンはエラーが発生せず, 通信中の輻輳ウィンドウサイズは一定の値であることから, 最大の輻輳ウィンドウサイズがおおよそ 2200 あたりであると予測できる。

##### 4.2.2 1G と 100M 対応の回線

図5の実験結果であるスマートフォンとタブレットの端末1台でのカーネルモニタから取得可能な TCP パラメータの振る舞いを記録した。スマートフォンはエラーが一定の間隔で発生し, エラーの発生に伴い輻輳ウィンドウサイズが減少していることがわかる。タブレットは, 40 秒あたりからエラーの発生もなく輻輳ウィンドウサイズも一定になっていることから, 輻輳ウィンドウサイズはおおよそ 470 であると予測できる。

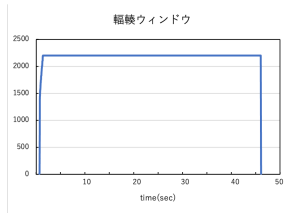


図 6 スマートフォンの CWND

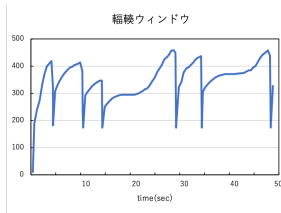


図 7 タブレットの CWND

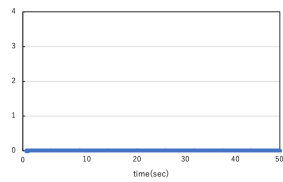


図 8 スマートフォンのエラー状態

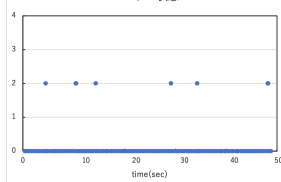


図 9 タブレットのエラー状態

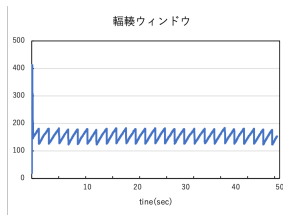


図 10 スマートフォンの CWND

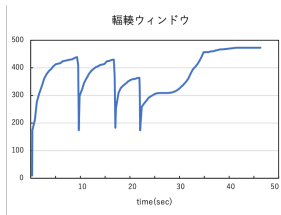


図 11 タブレットの CWND

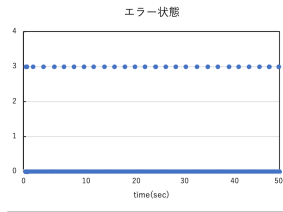


図 12 スマートフォンのエラー状態

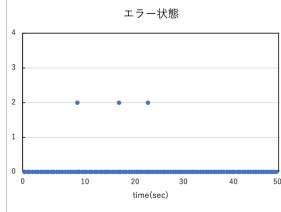


図 13 タブレットのエラー状態

### 4.3 端末 5 台の通信性能比較

図 14,15 は端末 5 台でスループット測定を行なった結果である。全て 1G 対応の回線ではスマートフォン 2 台が他の 3 台に比べ平均スループットが高く、1G と 100M 対応の回線ではタブレット 2 台が一番高いという結果が得られた。このことから、有線接続部分の環境の違いによって優先的に帯域を確保する端末が異なっていることがわかる。この原因を調べるために、カーネルモニタでカーネル内部の TCP パラメータの振る舞いを調べた。

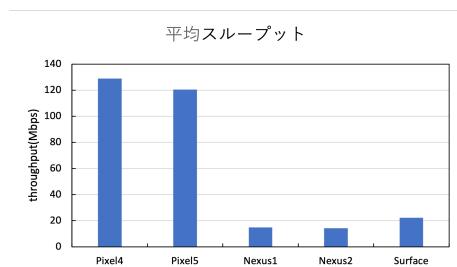


図 14 全て 1Gbps 対応の回線

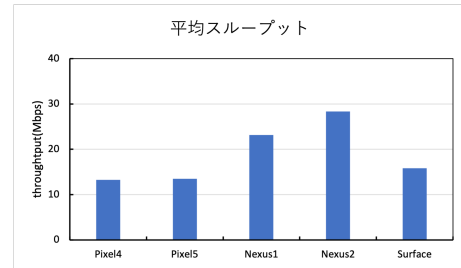


図 15 1Gbps と 100Mbps 対応の回線の混合

### 4.4 端末 5 台での TCP パラメータの比較

#### 4.4.1 全て 1G 対応の回線の振る舞い

図 16,17,18,19 は 5.2 節で示した実験で使用したスマートフォンとタブレットに搭載されたカーネルモニタから、TCP パラメータの振る舞いを示している。図 16,17 は、TCP の輻輳ウィンドウサイズ (CWND) の振る舞い、図 18,19 はカーネル内部の TCP 制御のエラー状態を示す。スマートフォンはエラーが発生していないため、輻輳ウィンドウが急速に上昇し、通信終了まで減少することなく最大の値を保持していることがわかる。対して、タブレットはエラー状態のグラフより 2 番の輻輳ウィンドウ減少の通知の発生後に輻輳ウィンドウサイズが減少し、スマートフォンに比べ輻輳ウィンドウサイズが伸びてないことがわかる。

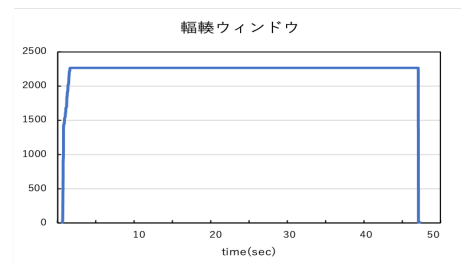


図 16 スマートフォンの CWND

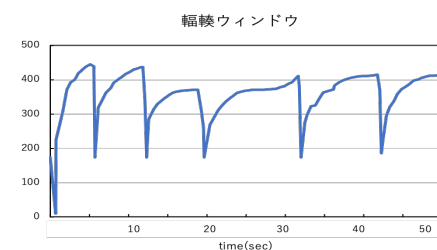


図 17 タブレットの CWND

#### 4.4.2 100M と 1G 対応の回線の振る舞い

図 20,21,22,23 は 5.2 節で示した実験で使用したスマートフォンとタブレットに搭載されたカーネルモニタから、TCP パラメータの振る舞いを示している。図 20,21 は、TCP の輻輳ウィンドウサイズ (CWND) の振る舞い、図 22,23 はカーネル内部の TCP 制御のエラー状態を示す。エラーは 1 番の packets 順序の入れ替わり検出,3 番の高速再転送等が発生した。発生した総

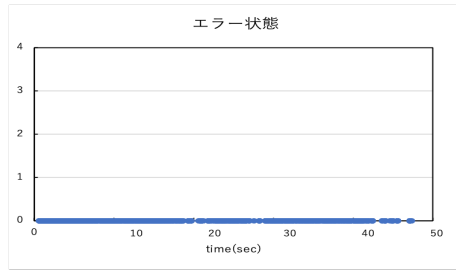


図 18 スマートフォンのエラー状態

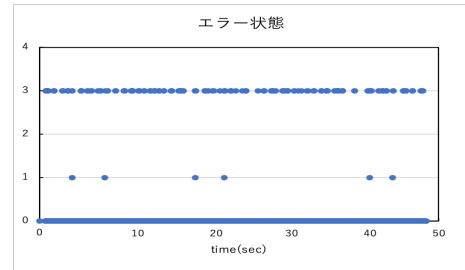


図 22 スマートフォンのエラー状態

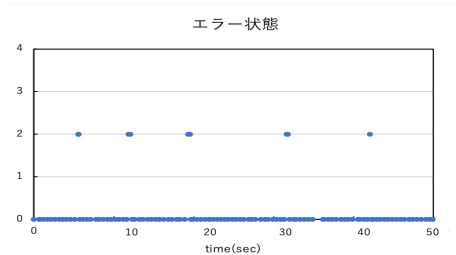


図 19 タブレットのエラー状態

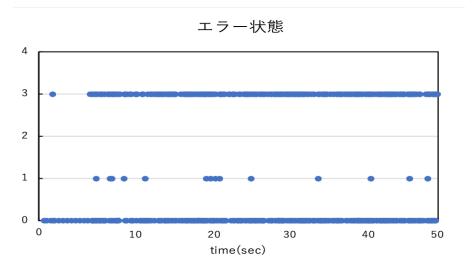


図 23 タブレットのエラー状態

エラー数を 5.2 節の結果と比較すると、格段に増えていることがわかる。また、スマートフォンとタブレットのエラー数はほぼ同等であるが、輻輳ウィンドウの値が図 20 では 20-40Mbps、図 21 では 120-170Mbps とタブレットの方が値が高いことが観察された。

以上のことから、1G と 100M 対応の回線で、タブレット 2 台の平均スループットが高くなる原因として、使用可能帯域が狭まることで大量のエラーが発生し端末 5 台とも輻輳ウィンドウサイズが伸びないこと、そして、タブレットとスマートフォンの OS の違いから、エラー発生後の対処として挙げられる、輻輳ウィンドウサイズの減少の仕方や、制御のコントロールが異なるためであると推測される。

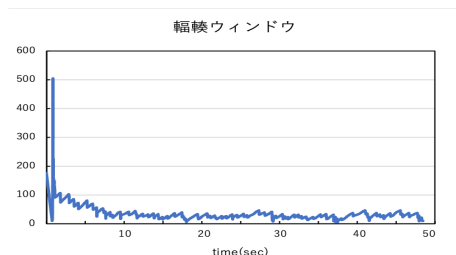


図 20 スマートフォンの CWND

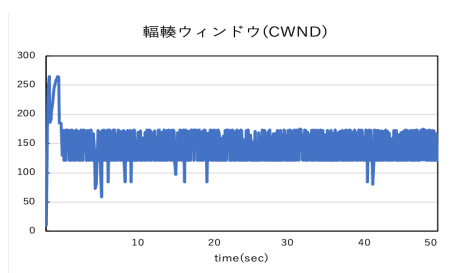


図 21 タブレットの CWND

## 5. まとめと今後の課題

本研究では、Linux サーバと端末 5 台の無線通信環境を構築し、iperf 通信を行いスループットを測定した。通信環境のうち、有線接続部分において 100Mbps のスイッチングハブを設置し、全て 1Gbps 対応の回線と 1Gbps と 100Mbps 対応の混合している回線の異なる通信環境での通信を比較した。また、スマートフォンとタブレットにカーネルモニタを搭載し、iperf 通信時の TCP パラメータである輻輳ウィンドウとエラー状態の振る舞いを記録し、観察した。

端末 5 台の通信では、全て 1G 対応の回線の際はスマートフォン 2 台が、そして、1G と 100M 対応の回線ではタブレット 2 台が帯域を優先的に確保することから、有線の回線の違いにより端末が異なることがわかった。この原因をカーネルモニタで記録できる TCP パラメータの振る舞いから推測した。全て 1G 対応の回線では、エラー数が少なく輻輳ウィンドウの値が大きいことがわかった。しかし、一度エラーが発生すると、輻輳ウィンドウサイズはおよそ 200 程大幅に減少してしまう。1G と 100M 対応の混合の回線では、エラーが頻繁に起こり輻輳ウィンドウサイズが伸びず、ある一定の値で増減が繰り返されていることがわかった。

本実験の結果から有線接続部分の違いによって、カーネルモニタで監視可能な TCP パラメータの振る舞いの結果に影響があることがわかった。その為、今後の課題としては、異なる有線接続部分の環境下で得られた情報を基に、無線通信の輻輳制御を行うミドルウェアの構築を行なっていきたい。さらに、得られた情報を機械学習で学習し、学習データから輻輳の発生を事前に予測することで輻輳の発生を回避するシステムの構築も行なっていきたい。

## 文 献

- [1] Worldwide smartphone shipment. <https://www.idc.com/promo/smartphonesmartphone-market-share/os>. Accessed: 2021-1.

- [2] 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.
- [3] What is 5g. <https://www.qualcomm.com/invention/5g/what-is-5g>. Accessed: 2021-1.
- [4] 5g market snapshot august 2020. <https://gsacom.com/paper/5g-market-snapshot-august-2020/>. Accessed: 2021-1.
- [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)*, pp. 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)*, pp. 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)*, pp. 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*, pp. 269–280, Cham, 2018. Springer International Publishing.
- [12] 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.
- [13] 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.
- [14] H. D. Trinh, L. Giupponi, and P. Dini. Mobile traffic prediction from raw data using lstm networks, 2018.
- [15] iperf. <https://iperf.fr/iperf-download.php>. Accessed: 2021-1.