

# Android OSにおける協調的輻輳制御手法による 異種クライアント端末通信性能の向上

島田 歩実<sup>†</sup> 山口 実靖<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

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

あらまし 近年のロススペース TCP はより高いスループットを確保するためにアグレッシブな輻輳制御手法を用いているが、無線接続環境においてはその手法によって膨大なパケットが無線 LAN アクセスポイントに蓄積され、その結果輻輳が発生してしまうという問題が生じている。先行研究では、ロススペース TCP の一種である CUBIC-TCP を輻輳制御アルゴリズムとして使用している Android 端末を用いて、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協調的制御手法がスマートフォン端末向けに提案、実装された。本研究では、スマートフォン向けの既存手法をタブレット端末にナイーブに適用し、その性能向上の程度を示す。また、既存手法に改善を加えさらなる性能向上を実現する手法を提案する。そして性能評価により、通信速度と公平性が先行研究の手法よりも多数台同時通信においてスマートフォンとタブレット共に向上したことを示す。

キーワード 輻輳制御, TCP, Linux, ミドルウェア, 無線通信, Android

## A Study of Android Tables Performance

Ayumi SHIMADA<sup>†</sup>, Saneyasu YAMAGUCHI<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

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

### 1. はじめに

TCP プロトコルは、インターネットにおいて標準的に利用されているプロトコルの一つであるが、近年のインターネットは通信速度と安定性が求められているため、TCP プロトコルはより高いパフォーマンスを得るために改善がなされてきた。TCP の輻輳制御アルゴリズムには、ロススペース、遅延ベース、ハイブリッドアルゴリズムの3種類があり、パケットロスを検知した時に輻輳ウィンドウを減少させる方式をとるロススペースアルゴリズムは、Linux の標準 TCP アルゴリズムとしても採用されている。しかし、近年のロススペース TCP はより高いスループットを確保するためによりアグレッシブにパケット送信を行う輻輳制御手法を用いており、有線接続と比べて脆弱な無線接続環境においてはその手法によって膨大なパケットが蓄積され、特に、同一アクセスポイント (AP) に接続する端末台数が多い場合は AP の通信機会が減ってしまうため、AP に膨大な ACK パケットが蓄積されてしまう。その結果パケットロスや輻輳などの深刻なエラーが発生し、エンドホスト側の端末が

十分な通信性能を發揮することができないという問題がある。

そこで先行研究では、輻輳制御アルゴリズムとしてロススペース方式の一種である CUBIC-TCP を採用している AndroidOS を用いて、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協調的制御手法がスマートフォン端末向けに提案、実装された [1] [2] [3]。手法としては、同一 AP に接続された端末数とクライアントサーバ間の往復遅延時間 (RTT) の二つのパラメータをエンドホスト側でリアルタイムに観測することで、その通信環境における理想的な輻輳ウィンドウ値 (CWND) をシステムが自動で算出し、補正を行う。この AP 周りでの ACK パケットの蓄積を回避する手法により、端末間で通う帯域を公平に分け合うことで全体の通信速度と公平性の向上を実現している。

本研究では、本手法によるスマートフォンとタブレット端末の通信性能の向上及びその改善について考察する。具体的には、スマートフォン向けの既存手法をタブレット端末にもナイーブに適用し、その性能向上の程度を示す。また、既存手法に改善を加えスマートフォンとタブレット端末の両方においてさらな

る性能向上を実現する手法を提案する。

本論文の構成は以下の通りである。2.章で研究背景について述べ、3.章で先行研究について紹介をする。続いて4.章で基礎実験について示す。5.章で先行研究のシステム評価実験とその結果を示し、6.章で先行研究のシステムへの改変内容とその評価実験と結果を示す。最後に、7.章で本稿をまとめる。

## 2. Android OS

本研究では、Android プラットホーム上で動作するシステムを扱う。Android OS [5] は Google 社が開発した OS で、オープンソースであることから誰でも自由にカスタマイズ可能であるという特徴を持つ。また Linux カーネルがベースとなっており、ロススペース TCP の一種である CUBIC-TCP [7] を輻輳制御アルゴリズムとして採用している。CUBIC-TCP は BIC-TCP [6] の改良版であり、BIC のスケラビリティを維持しながらウィンドウサイズ制御が簡素化した高速 TCP である。

図1はCUBIC-TCPの振舞で、縦軸がCWND、横軸が時間を示す。最後のパケット損失時からの経過時間を元に3次関数(1)によりウィンドウサイズを算出し、パケットロスが検出されるとウィンドウサイズを急減させる。3次関数のCはスケールファクタ、tは最後のパケット損失時からの経過時間、Wmaxは最後のパケット損失時のウィンドウサイズを表す。

このようにCUBIC-TCPは遅延ベースアルゴリズムと比べ経路の状況を把握せずに通信をするため、パケット損失が生じるまで各端末がアグレッシブに通信をし過ぎる傾向がある。よって無線接続環境においてはAP周りでACKパケットの蓄積が生じ輻輳の原因になってしまうといった問題がある。

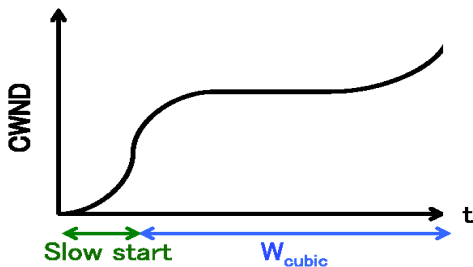


図1 CDUBIC-TCPの振舞

$$W_{cubic} = C(t - \sqrt[3]{\frac{W_{max}}{C}})^3 + W_{max} \quad (1)$$

## 3. 先行研究

### 3.1 カーネルモニタ

カーネル内部の処理は通常バックグラウンドで進められているため、通常ユーザ空間からその処理の様子を監視することはできない。そこで既存研究[1]によりカーネル内部の情報を見るためにカーネルモニタと呼ばれるツールが開発された。カーネルモニタは本研究のベースとして用いられており、Android端末のTCP通信時における輻輳ウィンドウサイズ(CWND)や往復遅延時間(RTT)などのカーネル内部の様々なパラメー

タをリアルタイムにモニタする汎用PC向けシステムツールである。図2に示したように、TCPのソースコードにモニタ関数を挿入し、カーネルを再構築することで、メモリからログを得ることが可能となる。既存研究[1]は、すでにAndroidが搭載されたスマートフォン端末へのカーネルモニタの組み込みに成功しており、本研究においては同様の方法でスマートフォン端末だけでなくタブレット端末への導入も行った。

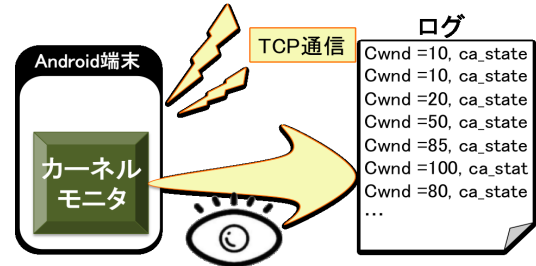


図2 カーネルモニタの概要

### 3.2 輻輳制御ミドルウェア

先行研究[2][3]で開発された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り、無線LAN-APにおけるACKパケットの蓄積を回避し、複数の端末が同一のAPに接続して通信するときの全体の通信速度と公平性の向上を可能にしている。

具体的な制御方法は以下の通りである。

- (1) APを共有する端末間でUDPパケットをブロードキャストし合うことで、同じAPへの同時接続端末数を把握。
- (2) (1)で検知した接続台数と式2から、適切なCWNDサイズの上限值候補1と下限値候補1をそれぞれ算出。
- (3) カーネルモニタから現在のRTT値と最小RTT値を取得。
- (4) (3)から式5を用いてRTT増減量を計算。
- (5) RTT増減量と表1から、適切なCWNDサイズの上限值候補2と下限値候補2をそれぞれ算出。
- (6) 上限値候補1と上限値候補2、下限値候補1と下限値候補2をそれぞれ比べ、小さいほうを最適なCWNDの上限值及び下限値としてプロセスインタフェースに0.5秒ごとに書き込み、補正。

また、先行研究[4]において帯域幅は接続台数に伴い変化することが確認されているため、帯域幅に関しては、式3から計算した値を用いている。ただし、BWmaxはその通信環境における最高通信速度、f(端末数)は端末数の増加による合計通信速度の低下を表す単調減少の関数であり、式4を用いている。接続台数が増えるほど、CWNDの補正值は小さくなる。

$$Ideal\_cwnd = \frac{Bandwidth[Mbps] \times RTT[sec]}{Segmentsize(1.5Kbyte) \times NumberOfTerminals} \quad (2)$$

$$Bandwidth = BWmax \times f(NumberOfTerminals) \quad (3)$$

$$f(\text{NumberOfTerminals}) = \text{NumberOfTerminals}^{-0.15} \quad (4)$$

さらに、通信中はカーネルモニタを 0.5 秒単位で常時監視し、その時点の RTT(current\_rtt) とその最小値 (min\_rtt) を取得する。min\_rtt は、通信中で最も小さい RTT を常に上書きしていくことで値を更新する。取得した値をもとに式 5 を用いて RTT の増減量 (ratio\_rtt) を求める。そして表 1 のように ratio\_rtt の大きさに従って、CWND の上限値候補と下限値候補を定める。特に ratio\_rtt が 331.0 以上になるとトラフィックが混んできたとみなし、CWND を 1 にして一時的に通信最小モードに移行し、その後起こり得るさらなる遅延を回避する。RTT 増減量が大きくなるほど、CWND の補正値は小さくなる。

$$\text{ratio\_rtt} = \text{current\_rtt} - \text{min\_rtt} \quad (5)$$

表 1 RTT 値による制御パラメータ

ratio_rtt	0-	51-	106-	161-	331-
max_cwnd2	555	555	555	555	555
min_cwnd2	200	50	10	2	1

これらの制御によって、通信中において同じ AP を共有する他端末が通信を始めたことやそれに伴って急に RTT の値が増加したことを本システムが検知すると、リアルタイムに CWND を理想的な値に補正することでトラフィック発生量を制限し、途中から通信を始めた端末にも均等に帯域を分け合えるよう制御することができる。

ただし本システムの制御手法は、AP 周りの通信状況に基づき、問題が発生した場合にのみ迅速に CWND の最大値と最小値を定めるだけであり、その範囲の中でデフォルトの TCP アルゴリズムによる制御が行われる。このように、基本的な TCP の輻輳制御アルゴリズムは変更せずエンドホスト側にのみ変更を加えるだけであるため、本システムの導入は容易である。

本研究では、近年スマートフォン端末と共に普及しているタブレット端末にもこのシステムを組み込み、従来のスマートフォンに加え、より性能の高い端末が同一 AP に接続して通信を行うときの全体の通信性能の検証実験を行う。

## 4. 基礎実験

### 4.1 実験概要

まず基礎実験として、本研究で用いる Android 端末である、スマートフォン端末 (Nexus 5) とタブレット端末 (Nexus 7) の評価を行った。

図 3 と表 2 に実験環境を示す。サーバ機と AP の間に人工遅延装置 dummynet [8] を挟み、有線部の往復遅延時間を特に輻輳が生じやすい高遅延環境を模擬するために 512ms に設定している。この環境において一つの AP に接続した Android 端末とサーバ間において iperf [9] を用いて通信を行い、ネットワーク全体の通信スループットを測定した。この環境において、Android 端末の台数を 1 台から 6 台まで 1 台ずつ増加させた。

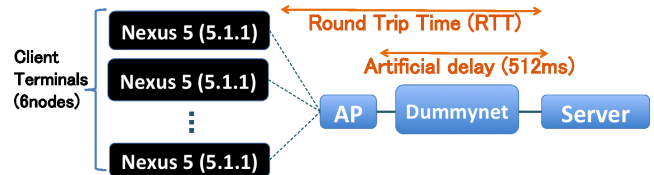


図 3 実験トポロジ

表 2 実験環境

Android	Model number	Nexus 5	Nexus 7
	Firmware version	5.1.1	6.0.0
	Kernel version	3.4.0-geaa8415-dirty	3.4.0-g272b14b-dirty
	Build number	LMY49H	MRA58K
server	OS	Ubuntu 14.04 (64bit) / Linux 3.13.0	
	CPU	Intel(R) Core(TM)2 Quad CPU Q8400	
	Main Memory	8.1GiB	
AP	Model	MZK-MF300N(Planex)	
	Communication system	IEEE 802.11g	

### 4.2 実験結果

スマートフォン端末、タブレット端末それぞれの端末数増加時における合計通信速度の評価結果を図 4 に示す。スマートフォン端末は 3 台同時通信、タブレット端末は 2 台同時通信をピークに、台数が増加するほど合計通信速度も減少することが確認できた。

この結果より、同時接続端末数の増加と共に減少するスループットの向上が求められると考察することができる。

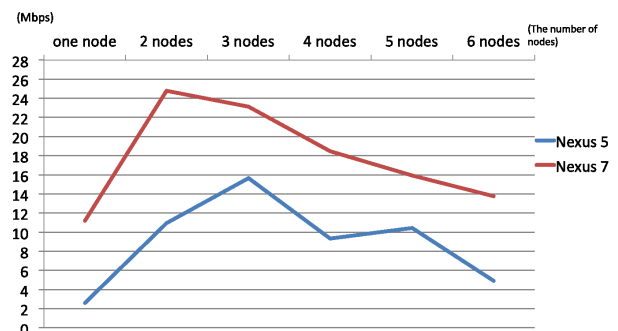


図 4 端末数変化時における合計通信速度

## 5. 性能評価実験

### 5.1 実験概要

続いて、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べく、スマートフォン端末及びタブレット端末にミドルウェアをナイーブに適用した。

4. 章と同様の実験環境下において、スループットやカーネルモニタによって観測されたパラメータ値から、それぞれの端末の振舞いを検証、評価した。

### 5.2 実験結果

合計の通信速度の評価結果を図 5 に示す。青のグラフはスマートフォン端末 (Nexus 5) の結果、赤のグラフはタブレット端末 (Nexus 7) の結果であり、スマートフォン端末、タブレッ

ト端末それぞれにおける、補正を行わないデフォルトのミドルウェアオフの状態、輻輳制御手法による補正を行ったミドルウェアオン状態の結果を示す。グラフより、スマートフォン端末においては3台同時通信、6台同時通信共に、スループットの向上が見られるが、やはり台数増加と共に通信性能が落ちる減少に変わりはないことがわかる。一方、タブレット端末においては、6台同時通信においては多少のスループットの向上が見られるが、3台同時通信においては大きく低下してしまうといった結果が得られた。

以上の結果より、台数増加と共に急激に通信速度が低下する減少や、ミドルウェアによる性能の低下を改善すべく、輻輳制御手法に改良を加えた。その改良の詳細は6.節に示す。

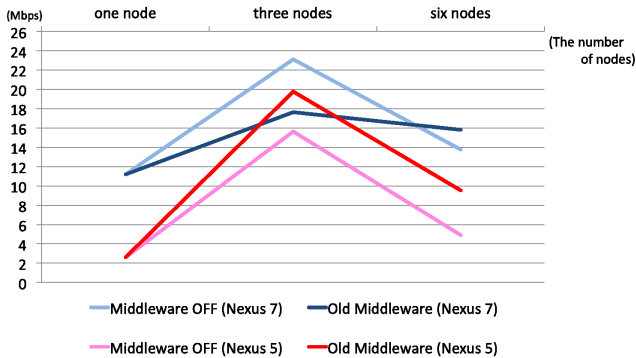


図5 合計通信速度 (改良前)

## 6. 既存研究の改良

### 6.1 改良点詳細

5.節の実験から確認した問題点を解決すべく、先行研究によって提案された輻輳制御手法を改良した。図6と図7は先行研究のミドルウェアによって制御されたCWNDの6台同時通信における推移を示す。3.2節に記述したように、本システムは通信中において同じAPを共有する他端末が通信を始めたことに伴って急激にRTT値が増加したことを感知すると、輻輳制御手法を発動する。図6は全端末のRTT値が急激に増え、本制御手法が発動し、CWNDが制御されている場合であるが、全ての端末においてCWNDの抑制が厳しすぎる様子がわかる。また図7は、6台中2台のみのRTT値が急激に増加し、本制御手法が発動している場合であるが、本システムによってCWNDが制御されている2台の端末のCWNDが厳しく抑えられているのに対し、その他4台の端末は本手法が発動していないことによって帯域を大きく占領している。このように、本システムが発動している端末とそうでない端末が混在する場合には、それぞれの端末に与えられる帯域に大きな差が出てしまい、全体の公平性が落ちていることが確認できた。そこでこれらの点に着目し、CWNDの抑制を緩くすることで、全端末において本システムが発動している場合に、帯域を無駄なく使用するようにした。また、本システムが発動している端末とそうでない端末が混在する場合には、帯域の使用量の差が縮まるようにした。その変更点を表3に示す。

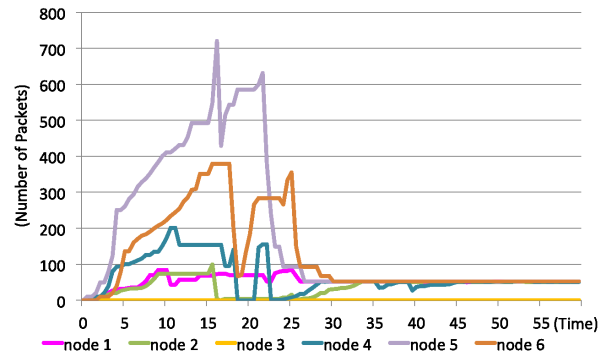


図6 CWNDの推移 (改良前1)

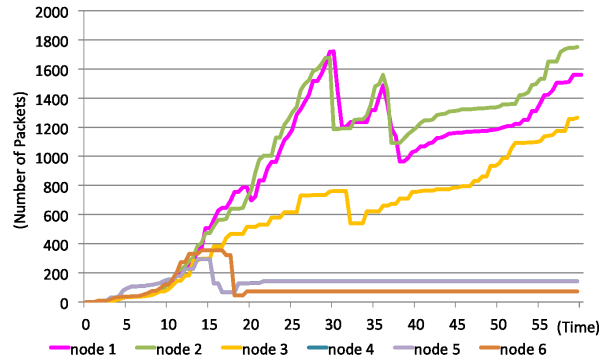


図7 CWNDの推移 (改良前2)

表3 RTT値による制御パラメータ

ratio_rtt	改善前		改善後	
	max	min	max	min
0-	555	200	1000	200
51-	200	50	700	150
106-	100	10	400	100
161-	10	2	200	10
331-	1	1	50	1

3.節に示したように、本システムは通信中において同じAPを共有する他端末が通信を始めたことやそれに伴って急にRTTの値が増加したことを本システムが検知すると、リアルタイムにCWNDを理想的な値に補正する。特に表3の改良前の制御パラメータ値を見てわかるように、RTT増減量が大きくなるほどCWNDの下限値が非常に小さく設定してあり、これが図6や図7に見られた結果の原因であると予測した。そこで、表3の改善後の制御パラメータ値のように変更を加え、CWNDの抑制を緩くした。以上のように改良された制御手法によって制御されたCWNDの振舞を図8と図9に示す。図8は全端末において本システムが発動している場合であるが、改良前よりも制御されているCWNDの振舞が大きくなっている。また、図9は、約50秒までは6台中3台のみに本システムが発動している場合であるが、発動していない他3台とのCWNDの大きさの差が明らかに縮まっていることが確認出来る。さらに、改良前と後の違いを容易に比較できるように、6台中1台のCWNDの推移を取り出し、図10にまとめた。改良前と比べると

とより明らかなように、CWND の抑制は穏やかになっている。この改良後の輻輳制御法を用い、5. 節と同様の環境において再実験を行った。

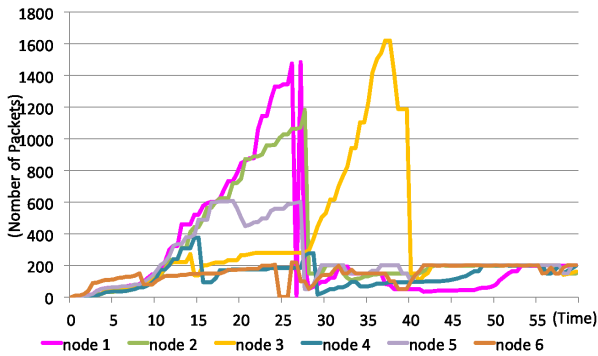


図 8 CWND の推移 (改良後 1)

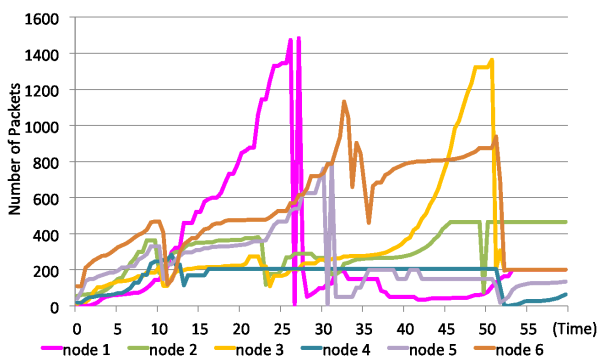


図 9 CWND の推移 (改良後 2)

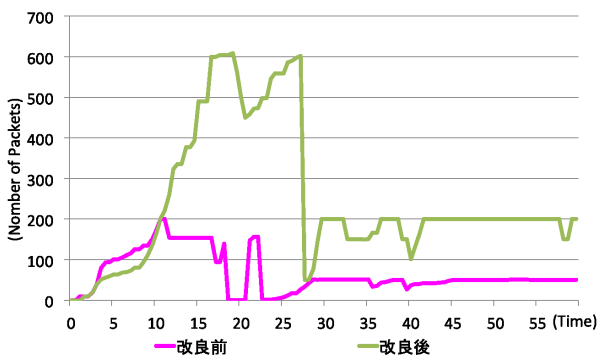


図 10 CWND の推移 (改良前, 改良後比較)

## 6.2 改良手法による実験結果

### 6.2.1 合計通信速度

まず、合計の通信速度の評価結果を図 11 に示す。青のグラフはスマートフォン端末 (Nexus 5) の結果、赤のグラフはタブレット端末 (Nexus 7) の結果である。また、それぞれ補正を行わないデフォルトの状態での結果、改良前の輻輳制御手法による補正を行った結果、改良後の輻輳制御手法による補正を行った結果である。グラフより、スマートフォン端末及びタブレット端末において、改良後の輻輳制御手法を使った場合、同時通信端末が 3 台から 6 台に増加した時のスループットの低下率が改

良前と比べ、大きく下がった。特に、スマートフォン端末においては、制御手法を用いないときと比べ、3 台同時通信の場合最大 32%、6 台同時通信の場合最大 317%スループットが向上した。これはより適した通信量を保つことで、帯域を有効に使うことができたためであると予測できる。一方、タブレット端末においては 6 台同時通信の場合最大 21%、スループットの向上が確認できたが、3 台同時通信の場合は約 22%低下してしまった。タブレット端末はスマートフォン端末に比べ、CPU やメモリの違いから高性能であり、制御手法を用いない場合も 3 台通信においては既に高い通信性能を發揮している。よって、同時接続台数が少ない場合は本システムを作動しないようにするなどといった対策が必要であると予測でき、今後の課題といえる。

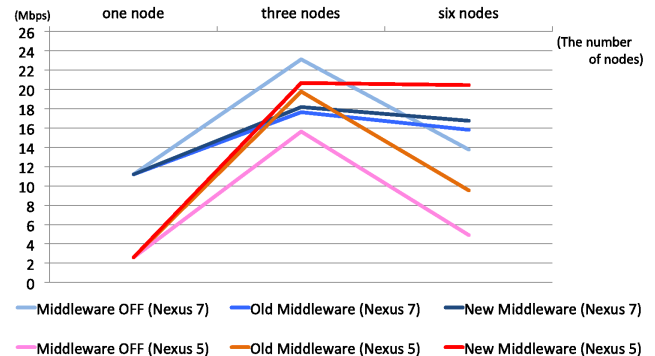


図 11 合計通信速度 (改良前, 改良後比較)

### 6.2.2 公平性

次に、Fairness Index [10] を用いて通信時における公平性の評価を行った。Fairness Index とは、公平性を示す指標であり、式 6 で算出された値が 1 に近いほど高い公平性を示す。

$$FairnessIndex : f_i = \frac{(\sum_{i=1}^k x_i)^2}{k \sum_{i=1}^k x_i^2} (1 \leq i \leq k) \quad (6)$$

図 12 に示すように、6 台同時通信の場合、公平性がスマートフォン端末において最大 18%、タブレット端末において最大 55%向上した。これは端末間で公平に帯域を分け合っているためであると考えられる。一方、3 台同時通信においては、スマートフォン端末とタブレット端末の両方において公平性が低下した。これは、輻輳制御モードに入らずに通信をしている端末が、輻輳制御モードに入っている端末よりも大きく帯域を占領してしまうためであると考えられ、改善を要すると思われる。

## 7. まとめと今後の課題

本研究では、ロススペース TCP を使用する端末が多数台同時に通信する場合において、AP に ACK パケットが蓄積し輻輳を起こす問題を回避するために、先行研究によって開発された輻輳制御ミドルウェアの異種クライアントへの導入と変更を行った。具体的には、まず先行研究によってスマートフォン向けに開発された輻輳制御ミドルウェアをナイーブにタブレット端末へと導入し、評価実験を行った。その評価実験の結果から、

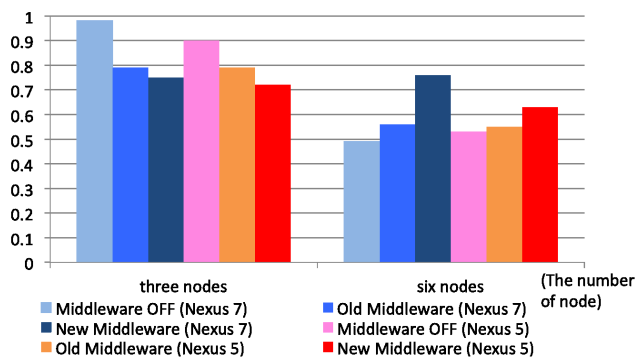


図 12 公平性

手法には CWND を厳しく抑制し過ぎており、帯域を有効に使用できていないという問題点を見出し、制御パラメータに改変を加え、再実験を行った。その結果、合計通信速度が、スマートフォン端末においては、3 台同時通信の場合最大 32%、6 台同時通信の場合最大 317% 向上した。一方、タブレット端末においては 6 台同時通信の場合最大 21%、スループットの向上が確認できたが、3 台同時通信の場合は約 22% 低下してしまったため、今後の課題といえる。さらに公平性を評価したところ、6 台同時通信の場合、スマートフォン端末において最大 18%、タブレット端末において最大 55% 向上した。この結果より、適切な CWND の上限値と下限値を設定することによって、多数の端末間で公平に帯域を分け合っていることが確認できた。一方、3 台同時通信においては、スマートフォン端末とタブレット端末の両方において公平性が低下した。

通信速度、公平性の両方の観点において、同時接続台数が少ない場合はデフォルトの状態ですでに高い性能を発揮しているため、必ずしも本システムを作動する必要はないと言える。よって、同一 AP への接続台数によってシステムの発動の有無をコントロールするといった改良が必要である。

さらなる今後の課題としては、スマートフォン端末とタブレット端末が複数混在している場合の制御手法の性能評価を予定している。さらに、WLAN 環境下だけでなく、LTE 環境下における輻輳制御ミドルウェアの作動も確認済みであるため、LTE 環境下における制御手法の効果も確認していく予定である。

## 謝 辞

本研究を進めるにあたり、NTT ドコモの早川愛さんに大変有用なアドバイスをいただきました。深く感謝いたします。

## 文 献

- [1] 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.
- [2] 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.
- [3] Ai Hayakawa, Saneyasu Yamaguchi, Masato Oguchi: "Reducing the TCP ACK Packet Backlog at the WLAN Access Point," Proc. ACM IMCOM2015, 5-4, January 2015.
- [4] Makiko Matsumoto and Masato Oguchi: "Multiple Access in MAC Layer Based on Surrounding Conditions of Wireless Stations" In Proc. the 1st International Workshop on Vehicular Communications and Applications (VCA2012) in conjunction with the 11th Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net2012), pp.133-140, June 2012.
- [5] Android open source project, <http://source.android.com>
- [6] L. Xu, K. Harfoush, and I. Rhee, "Binary Increase Congestion Control for Fast, Long Distance Networks," Proceedings of Tech. Report, Computer Science Department, NC State University, 2003.
- [7] Sangtae Ha, Injong Rhee and Lisong Xu "CUBIC: a new TCP-friendly high-speed TCP variant" ACM SIGOPS Operating Systems Review - Research and developments in the Linux kernel, vol.42, pp.64-74, July 2008.
- [8] The dummynet project: <http://info.iet.unipi.it/luigi/dummynet>
- [9] Iperf For Android Project in Distributed Systems, <http://www.cs.technion.ac.il/sakogan/DSL/2011/projects/iperf/index.htm>
- [10] 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.