

## 3X-04 Android 端末を用いた無線 LAN アクセスポイントにおける TCP ACK パケット蓄積回避手法の提案と実装

早川 愛<sup>†</sup>山口 実靖<sup>‡</sup>小口 正人<sup>†</sup><sup>†</sup>お茶の水女子大学<sup>‡</sup>工学院大学

## 1. はじめに

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、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たり、その結果ロスしたりしてしまうという問題が生じている。本研究では、そのロススペース TCP の一つである TCP CUBIC 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 (AP) における ACK パケットの蓄積を回避する手法の提案と実装を行う。

本手法では、同一 AP に接続された端末数とそのネットワークの混み具合を示す往復遅延時間 (RTT) の二つのパラメータにより、各端末の最大輻輳ウィンドウ値 (CWND) を補正する。それにより、AP での ACK パケットの蓄積を防ぎ、かつ端末間で可用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可能にしている。

## 2. 研究背景

## 2.1 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の開発を行う。Android は Google 社が開発したモバイル端末用プラットフォームであり、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できるという特徴がある。

Android は linux カーネルがベースとなっており、TCP CUBIC を輻輳制御アルゴリズムとして採用している。TCP CUBIC は、最後のパケット廃棄が発生してから経過時間を基にして、リアルタイムに CWND を制御する。またロススペース方式であるため、遅延ベース方式のものに比べて各端末がアグレッシブに通信しすぎる傾向にある。それにより、同時に通信する端末数が多いときには AP で ACK パケットが蓄積しやすく、タイムアウトやパケットロスが生じてしまうという問題がある。

## 2.2 輻輳制御ミドルウェア

先行研究 [1] で開発された輻輳制御ミドルウェアは、同一 AP に接続した端末間でお互いの接続状況を把握し、その接続台数によって混み具合を予測し、CWND の上限値を設定する組み込みシステムである。可用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に成功している。しかしながら、この先行研究における CWND の補正値は経験則により決め打ちの値が用いられていた

ため、システム環境が変わると、補正値を決め直さなければならなかった。

そこで本研究では、この補正値をシステムに自動で算出させることで、より最適な制御を行うとともに、各端末の通信状況をより詳細に把握するために、接続台数だけでなく各端末の RTT をリアルタイムに観測し、制御パラメータに適応することで、本システムのさらなる性能と利便性の向上を目指す。

## 2.3 カーネルモニタリングツール

本システムのベースとして用いられているカーネルモニタ [2] は、Android 端末の通信時における CWND や RTT などの様々なパラメータをモニタできるシステムツールである。このツールを用いることで、本システムにおいてもカーネル内のパラメータをリアルタイムに解析することができる。

## 3. 提案手法

本研究における提案手法の概要を図 1 に示す。2.2 節において紹介した変更前のミドルウェアでは、各端末においてカーネルモニタで読み込んだ情報を UDP ブロードキャストする発信部と、その情報を受信し、解析を行い最適化チューニングをする受信部に分かれて制御を行っていた。これに対し本研究では、実際に最適化チューニングを行う受信部においてもカーネルモニタの読み込みが必要となったため、この受信部と発信部を一括にまとめることで、導入や制御の簡単化を実現した。また、CWND の補正値を本システムが自動で算出するように CWND の理想値を示す式 1 を用いた。先行研究 [3] 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ので、帯域幅に関しては、式 2 のように計算した値を用いている。ただし、 $BW_{max}$  はその通信環境における最高通信速度である。 $f(\text{端末数})$  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、先行研究 [3] を元に式 3 とした。

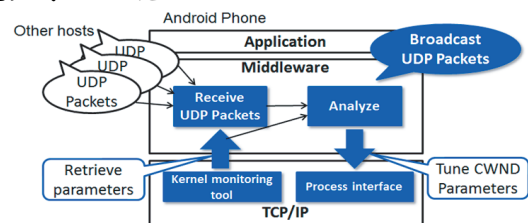


図 1: 提案手法の概要

$$Ideal\_cwnd = \frac{Bandwidth[Mbps] \times RTT[sec]}{Segment\ size(1.5K\ byte) \times \text{端末数}} \quad (1)$$

$$Bandwidth = BW_{max} \times f(\text{端末数}) \quad (2)$$

$$f(\text{端末数}) = \text{端末数}^{-0.15} \quad (3)$$

Reducing the TCP ACK Packet Backlog at the WLAN Access Point using the Android Terminal

<sup>†</sup>Ai Hayakawa, <sup>‡</sup>Saneyasu Yamaguchi, and <sup>†</sup>Masato Oguchi  
Ochanomizu University (<sup>†</sup>)  
Kogakuin University (<sup>‡</sup>)

さらに、通信中はカーネルモニタを常時監視し、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 4 を用いて増減の比率 (ratio\_rtt) を求める。

$$ratio\_rtt = \frac{RTT}{min\_rtt} \quad (4)$$

この ratio\_rtt が 6.0 以上になると、トラフィックが混んできたとみなし、CWND を補正するフェーズに切り替わる。また、式 1 を用いた端末数による補正をした上でも RTT が増加してしまう場合においては、ratio\_rtt が 5.0 以上になると CWND を 1 にして一時的に通信停止モードに移行する。

## 4. 性能評価

### 4.1 実験概要

図 2 と表 1 に示す環境において、本提案手法の評価実験を行った。本実験環境における BWmax は、 $1.9 \times 10^3$  である。AP とサーバ機の間には、人工遅延装置 Dummynet 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 256ms に設定している。この環境において、Iperf を用いて通信スループットを測定した。

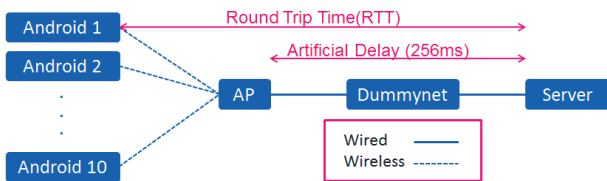


図 2: 実験トポロジ

表 1: Specifications of devices

Android	Model number	Nexus S
	Firmware version	4.1.1
	Baseband version	I9023XXKD1
	Kernel version	3.0.31-ai
	Build number	JRO03L
server	OS	Ubuntu 12.04 (64bit) / Linux 3.0.1
	CPU	Intel(R) Core 2Quad CPU Q8400
	Main Memory	7.8GiB
AP	Model	MZK-MF300N(Planet)
	Sommunication system	IEEE 802.11g

### 4.2 実験結果

合計の通信速度の評価結果を図 3 に示す。青のグラフは、補正を行わないデフォルトの状態、赤のグラフは本提案手法による補正を行った結果である。グラフより、RTT の増加が観測されたと考えられる 4 台以降において本提案手法による性能向上が確認され、最大で 10 台の端末が同時に通信するときの通信速度は 3.35 倍向上した。

次に、FairnessIndex を用いて通信時における公平性の評価を行った (図 4)。Fairness Index とは、公平性を示す指標であり、式 5 で算出された値が 1 に近いほど高い公平性を示す。

$$FairnessIndex : f_i = \frac{(\sum_{i=1}^k x_i)^2}{k \sum_{i=1}^k x_i^2} \quad (1 \leq i \leq k) \quad (5)$$

青のグラフに示す本提案手法を用いない場合には、同時通信台数が多くなるほど公平性が損なわれているが、それに対し、本提案手法を適応した赤のグラフでは台数が 10 台まで増加してもほぼ 1 の値を維持している、つまり公平性が保たれていることを確認した。

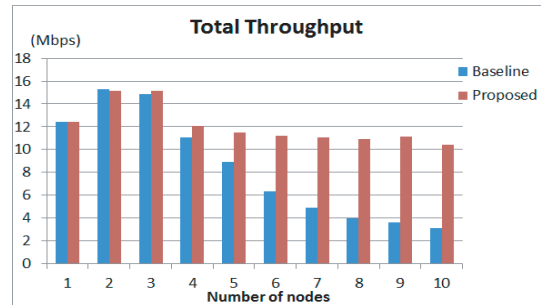


図 3: 合計通信速度

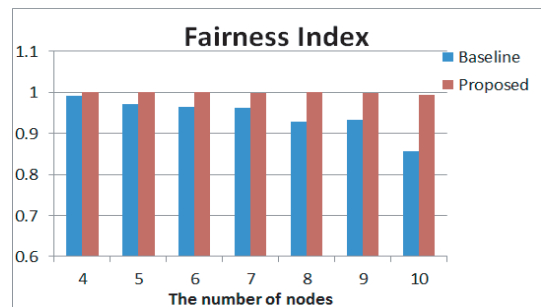


図 4: 公平性の評価

## 5. まとめと今後の課題

本研究では、ロスベース TCP 使用端末が多数台同一 AP に接続する時の ACK パケット蓄積回避手法として、端末間で混み具合を把握し協調的に可用帯域を分け合う手法を提案、実装を行った。評価結果より、本提案手法を用いることで 10 台で通信時の通信速度が最大で 3.35 倍、公平性が 1.16 倍向上することを確認した。

今後は、本提案手法による補正を行わない端末が混んでいる場合と、通信台数が動的に変化する場合の評価を行う。

### 参考文献

- [1] Hiromi Hirai, Saneyasu Yamaguchi, and Masato Oguchi: "A Proposal on Cooperative Transmission Control Middleware on a Smartphone in a WLAN Environment," In Proc. the 9th IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob2013), pp.710-717, October 2013.
- [2] Kaori Miki, Saneyasu Yamaguchi, and Masato Oguchi: "Kernel Monitor of Transport Layer Developed for Android Working on Mobile Phone Terminals," Proceedings of The Tenth International Conference on Networks (ICN), pp. 297-302. 2011.
- [3] 松本真紀子, Onur Altintas, 西堀 満洋, 小口 正人: 「環境情報を利用したマルチプルアクセス手法の一検討」情報処理学会第 73 回全国大会, 1V-6, 2011 年 3 月。