

# Androidの無線LAN通信性能に関する一検討

三木 香央理†

小口 正人†

†お茶の水女子大学

## 1 はじめに

近年, スマートフォン市場の成長に伴い, 携帯端末で動作する組み込み機器のソフトウェアプラットフォームとして Google 社開発の Android が注目されている. アプリケーション開発や柔軟な拡張性において注目度の高い Android 携帯に対し, 本研究ではそのサービス提供を可能にしたシステムプラットフォームとしての Android に興味を持ち, 特にそのネットワークおよびネットワークコンピューティング能力について評価する. Android 携帯の無線ネットワークにおける通信能力について解析し, その設定に手を加えることで, より高性能な通信を目指す.

## 2 研究背景

### 2.1 Android

Android のアーキテクチャを図 1 に示す. Android は Linux2.6 カーネルを用いて構成されており, この OS に各種コンポーネントを追加し Android というプラットフォームを構築している. 市場に複数ある Linux パッケージでも Android が構成できるようになっている.

また, Linux カーネルの上に Android 独自のアプリケーション実行環境である Android Runtime を実装し, Dalvik と呼ばれる独自の仮想マシンを搭載している. これは Java の仮想マシンに相当する.

その上にアプリケーション・フレームワーク, アプリケーションが乗る形態であるため, アプリケーションは Dalvic にあわせて開発すればよく, ポータビリティが高い.

このように, これまでの携帯端末用開発ツールとは異なり, オープンソースでありキャリア間の制約がないためユーザのカスタマイズ自由度が高く, アプリケーション開発の負荷が軽減され, 他キャリア, 他機種への柔軟な拡張性があるといえる.

一方, 通信は Linux カーネルの中のプロトコルスタックを用いて実行されているため, この TCP 実装部分などで性能が決まってくると考えられる. そのため, 我々はカーネル中のトランスポート層実装に焦点を当て研究を進めていく.



図 1: Android のアーキテクチャ

## 3 研究概要

本研究では, まず AP の近くにいる人同士の通信を想定し, Wi-Fi 環境における同一無線 LAN 内の通信を評価した. 次に Wi-Fi を用いず, Bluetooth を用いたアドホック接続の通信を評価し, 最後に遠隔地にあるモバイルクラウドを提供するサーバへ携帯端末からアクセスする通信の 3 パターンの無線通信のケースを想定しその性能を評価した. スループット測定は iperf-2.0.4[2] をクロスコンパイルし, Android 実機に送り込んで実行した.

### 3.1 実験手順

表 1, 2 および図 2 ~ 4 に本研究の実験環境の一部を示す

表 1: 実験環境

Ubuntu	OS	Ubuntu8.10(Linux2.6.27-7)
	CPU	Intel(R) Pentium(R) M processor 1.73GHz
	Main Memory	512MB
WindowsXP	OS	Microsoft Windows XP
	CPU	Intel(R) Pentium(R) 4 CPU 3.00GHz
	Main Memory	512MB
	Bluetooth	Bluetooth Ver2.1+EDR
mobilegw	OS	Fedora Core release 3
	CPU	Intel(R) Pentium(R) 4 CPU 3.00GHz
	Main Memory	1GB

表 2: 実験環境 (Android 側)

Android1	Model number	HT-03A
	Firmware version	1.5
	Baseband version	6.2.50S.20.17H.2.22.19.26I
	Mod version	docomo standard
	Build number	CDB72
	CPU	Qualcomm MSM7201A 528MHz
Android2	Model number	T-Mobile myTouch 3G
	Firmware version	1.6
	Baseband version	2.6.29.6-cm42 shade@toyxygene
	Mod version	CyanogenMod-4.2.3.1
	Build number	DRC92
	CPU	Qualcomm MSM7201A 528MHz
	Bluetooth	Bluetooth Ver2.0+EDR

### 3.2 測定環境



図 2: Wi-Fi を用いた AP 経由の通信

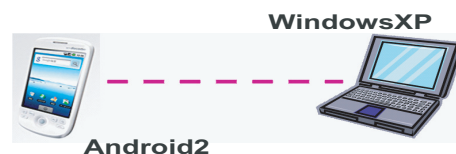


図 3: Bluetooth を用いた AP を経由しない通信

A Study about performance of communication on Android terminals in a wireless LAN

† Kaori Miki, Masato Oguchi  
Ochanomizu University (†)

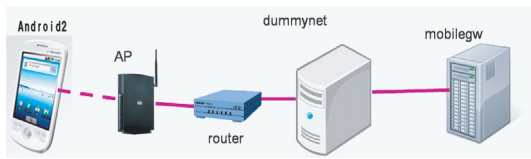


図 4: 遅延環境におけるサーバと Android の通信

#### 4 性能測定

本研究では iperf を用いてソケット通信の性能を測定した。モバイルクラウドを利用する通信の測定結果を以下に示す。

Wi-Fi 環境における同一無線 LAN 内の通信においては,Ubuntu から Android への通信の平均スループットは 9.83(Mbps),Android から Ubuntu への通信の平均スループットは 7.22(Mbps) と,Android の方が送信性能が低いことがわかった。UDP における通信は双方間共に 1.0(Mbps) 前後と TCP 通信に比べ明らかに低い結果となった。この結果は buffer size にスループットが依存しなかった。また Android から Android への通信の場合,TCP 通信の平均スループットは 4.62(Mbps),UDP 通信は 0.584(Mbps) という結果になった。

Bluetooth による通信も,WindowsXP から Android への通信の平均スループットは 1.564(Mbps),Android から WindowsXP への通信の平均スループットは 1.444(Mbps) と Android の方が送信性能が低いことがわかった。しかし,Wi-Fi を用いた AP 経由による通信の結果よりも,スループットの差が小さいことがわかった。また TCP 通信においてスループットは window size に依存しないことがわかった。Bluetooth を用いた UDP 通信のスループットの値は TCP 通信とほぼ同じ結果になった。さらに UDP 通信では Android から WindowsXP に送る場合 window size に結果は依存しないが Windows から Android に送る場合,受け手である Android の buffer size が大きい方がスループットが高くなることがわかった。

##### 4.1 モバイルクラウドを利用する通信

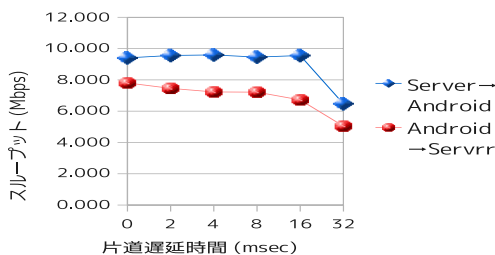


図 5: 遅延環境におけるサーバと Android の TCP 通信

遅延装置を使って,片道遅延時間 0-32ms の遅延環境を作り,Android と mobilegw 間のスループットを測定した。結果は図 5 からわかる通り,片道遅延時間が 16ms までは安定したスループットが得られるが,それ以上になると大幅にスループットが低下してしまうことがわかった。UDP における通信ではスループットが 1.0(Mbps) と TCP に比べ激減することから,UDP 通信はうまく機能していないことがわかった。

#### 4.2 TCP チューニング

次にモバイルクラウドを利用するケースにおいて TCP の輻輳制御アルゴリズムを変えてみることでその性能の違いを測定した。本研究で用いた輻輳制御アルゴリズムは,bic,cubic,htcp,reno,westwood の 5 種類である。TCP チューニングを行うためにサーバの OS を Fedora Core release 9 にバージョンアップした。

この OS のデフォルト輻輳制御アルゴリズムは cubic である。bic は RTT 公平性に着目し積極的に window size を増加させるアルゴリズムである。cubic は bic の改良版で window size の増加が緩やかである。htcp はロスペースの高速 TCP であり,htcp は RTT や帯域幅が極端に大きい広域・高速ネットワーク上では,その帯域幅に合わせて効率的な振る舞いをする。reno は輻輳が発生したときの輻輳 window size の 2 分の 1 の値を ssthresh に設定し,後に輻輳が発生した場合は輻輳 window size を ssthresh から輻輳回避段階に入るアルゴリズムである。これにより過剰な輻輳 window size の減少を避けられるため高速な転送が可能としている。westwood は reno に比べ,太い帯域で且つ漏れやパケットロスに強いアルゴリズムである。

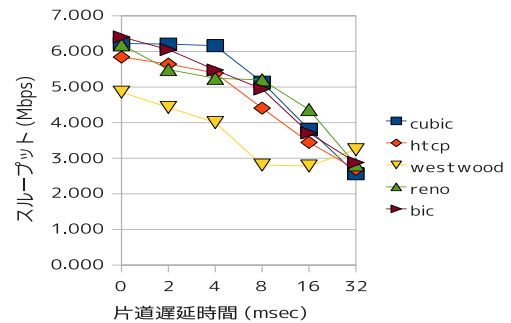


図 6: 遅延環境におけるサーバと Android の TCP 通信

図 6 にパケットロスのない環境での測定結果を示す。結果として westwood アルゴリズムにおいて性能が低下することがわかった。現在,パケットロスを与えた場合の性能を測定中である。

#### 5 まとめと今後の課題

Android における様々な通信ケースを想定し,それらのスループット測定を行った。結果からそれぞれの通信ケースにおけるスループットが window size に依存するときとしないときがあることなど,それぞれの通信ケースの特徴がわかった。また UDP 通信の場合,本実験で用いた Android 端末は正常に機能していないらしいということもわかった。今後は TCP の輻輳制御アルゴリズムを適応させた場合のスループットの変化について検討していきたい。

#### 参考文献

- [1] Android:<http://www.google.co.jp/mobile/android>
- [2] Iperf:<http://downloads.sourceforge.net/project/iperf/iperf/2.0.4>