

3Z-4 Android 端末を用いた無線通信時の制御ミドルウェア導入

平井 弘実[†]三木 香央理[†]山口 実靖[‡]小口 正人[†][†]お茶の水女子大学[‡]工学院大学

1. はじめに

近年スマートフォンにおいては無線通信でクラウドにアクセスし、データ通信を行うサービスが増えている。クラウドを構成するデータセンタは遠隔地に存在することが多く、そのような場合は高遅延通信を行うこととなる。高遅延通信の場合、特に輻輳ウィンドウの値が通信スループットに大きな影響を与える。またクライアント・サーバ間通信においては、モバイル端末から身近なアクセスポイントまでは無線通信、アクセスポイントからサーバまでは有線通信で行われる。すなわちパケットロスが発生するのは大部分が端末からアクセスポイントまでの無線空間と考えることができる。

本研究は、同一アクセスポイントを共有する無線空間内で互いの通信状況を知らせ合うことにより、各々の端末の輻輳制御を行おうとする試みである。その制御としてカーネル内に組み込んだ独自の TCP との切り替えを行うミドルウェアを開発した。無線通信を行うモバイル端末間で連携して通信の最適化を行うという手法はこれまでになく、本研究が初の試みと言える。

2. Android OS

Android 携帯とは、Google 社を中心に開発されている Android OS を搭載したスマートフォンを指す。Android は、無償で提供される開発環境において構築することができ、オープンソースであり開発が自由であるという点で大変注目されている [1]。また Android はキャリア間の制約がないため、アプリケーション開発においても自由度及び汎用性が高いだけでなく、一度マーケットに登録すると、世界中の Android ユーザがインストール可能となる。現在 Android マーケットでは、このような大きなビジネスチャンスを提供されているため、数多くのアプリケーションが登録されており、アプリケーション市場は賑わっている。

本研究はこれらのサービスを提供するシステムプラットフォームとしての Android 携帯に焦点を当て、通信システムの高速化を目指しているが、このように Android 端末においてアプリケーションの存在を無視することはできない。そこで本研究では Android アプリケーションからの無線通信利用を前提として、通信スループットの高速化を目指す。

3. 既存研究

本ミドルウェアでは切り替え用の TCP として [3] を利用している。この TCP は、Android のデフォルトで機

能している TCP-CUBIC が無線通信において不必要に輻輳ウィンドウを低下させている場合があるという点に着目し、輻輳ウィンドウを下がりやすく上がりやすい、強気な通信をするように改良したものである。この TCP は RTT=32ms 以上の高遅延環境において、性能が良いことが確認されている。また、輻輳ウィンドウの最大値を変えられるため、今後は環境に応じたフレキシブルな適応制御を実装する事が可能である。

4. 通信制御ミドルウェア

4.1 開発目的

本研究では未使用帯域を使い切れる通信システムの確立を目指している。独自の TCP は輻輳ウィンドウを上がりやすく、下がりにくい制御アルゴリズムを実現したが、このような強気な TCP を常に利用することは他端末の通信を阻害する可能性がある。そこで本研究では適切なタイミングで TCP を環境に適したものに切り替えるミドルウェアを開発する。

本研究は、Android 端末が広帯域有線ネットワーク接続されたクラウドサーバと通信する場合を想定し、輻輳が懸念されるアクセスポイント—Android 端末間の無線帯域を共有している他端末の通信状況を考慮した制御を目指している。そこで本ミドルウェアは、同一アクセスポイントを共有する無線 LAN 空間内において、互いの端末の通信状況、すなわち輻輳ウィンドウを通知し合い、周囲の他端末の通信状況に応じて、輻輳制御アルゴリズムを切り替え適応させる。カーネルモニタを組込んだ Android 端末を利用することにより、実機の通信中の輻輳ウィンドウの値を取得することを可能にした。

4.2 実装

本稿では、既存研究の TCP とデフォルトの TCP を切り替えるミドルウェアを紹介する。本ミドルウェアは汎用性を高めるため、アプリケーションから独立し、CPU 負荷の低いネイティブコードで実装した。ネイティブコードのクロスコンパイラとして、arm-2008q3[2] を使用している。

ミドルウェアは発信部と受信部に分けることができる。発信部は常駐でカーネルモニタのログを読み出し続け、ログがあれば解析し、輻輳ウィンドウの値を UDP パケットでブロードキャストする。受信部はこの UDP パケットを受信し、自分の IP アドレスと異なる送信元から一定値を超える輻輳ウィンドウの通知を受けた際にデフォルトの TCP-CUBIC に切り替える。また、他端末からの UDP パケットを一定時間受信しないときに帯域を独占したと判断し、デフォルトの TCP-CUBIC を強気な TCP に切り替える。TCP の切り替えは、proc インタフェースを利用している。

A Study on Transmission-Control Middleware on Android Terminal in a Wireless LAN Environment

[†] Hiromi Hirai, [‡] Kaori Miki, [‡] Saneyasu Yamaguchi, and [†] Masato Oguchi

Ochanomizu University ([†])

Kogakuin University ([‡])

5. 通信制御ミドルウェアを導入した通信実験

5.1 実験概要

本実験では図 1 に示すように、1 台のアクセスポイントを 2 台の Android 端末で共有する。Android 端末を送信側とし、アクセスポイントと人工遅延装置を経由してサーバに対してパケット転送を行う。端末 1 が 50 秒間の通信を行っている間に、端末 2 が途中から加わり 20 秒間のパケット転送を行う。パケット転送は今回は Iperf を利用したが、今後は Dalvik 上の通信テストを行い、アプリケーションから高速通信ができることを確認する方針である。

TCP の切替えを確認するために 2 つの実験を行った。実験 1 では Android 端末の TCP がデフォルトの状態であり切り替えは行わない。実験 2 では、30 以上の CWND 値を受信すると TCP を CUBIC に切り替え、30 以上の CWND 値を 5 秒以上受信しなかったら独自の TCP に切り替えるミドルウェアを導入した。

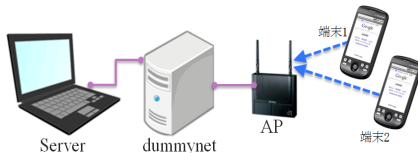


図 1: 実験システム

5.2 実験環境

本実験で使用した実験環境を表 1 に示す。人工遅延装置 FreeBSD の dummynet を利用し、経路上に人工パケットロス率 0.4%、往復遅延時間 64ms を加えた。

表 1: 実験環境

Android	Model number	AOSP on Sapphire(US)
	Firmware version	2.1-update1
	Baseband version	62.50S.20.17H_2.22.19.26I
	Kernel version	2.6.29-00481-ga8089eb-dirty
	Build number	aosp_sapphire_us-eng 2.1-update1 ERE27
server	OS	Fedora release 10 (Cambridge)
	CPU	CPU: Intel(R) Pentium(R)4 CPU3.00GHz
	Main Memory	1GB

5.3 実験結果と評価

これらの実験結果を図 2、図 3 に示す。これらの図は、ブロードキャストされた UDP パケットを受信し、輻輳ウィンドウの値を時系列に沿ってグラフを描画する自作の可視化ツールによる。グラフの縦軸は輻輳ウィンドウの値、横軸は時間 (s) を表す。実験 1 では、TCP の切り替えはなく、常にデフォルトの TCP を利用して通信が行われている。それに対し、実験 2 では初期状態を独自の TCP とし、端末 1 の通信開始から 10 秒後に端末 2 が通信を開始した。その結果、端末 2 の輻輳ウィンドウ値が 19 秒後に 30 を上回り、端末 1 はこの通知を受けて、デフォルトの TCP に切り替わった。また端末 1 の開始から 30 秒後に端末 2 の通信が完了し通知を終了した。端末 1 はその 5 秒後に帯域を独占したと判断し、独自の TCP に切り替わった。

6. まとめと今後の課題

本ミドルウェアは、TCP の通信処理中に周囲の環境に合わせて自動的に TCP を切り替えることに成功した。

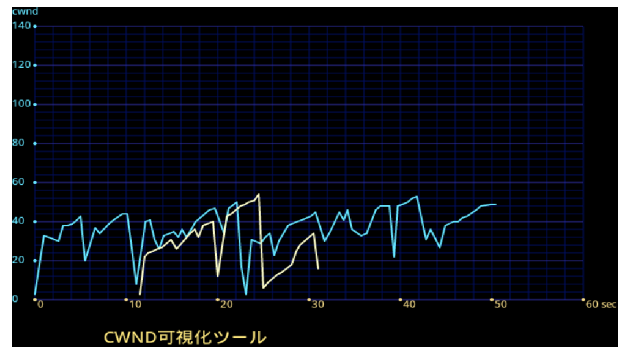


図 2: 実験 1: デフォルトの TCP による処理

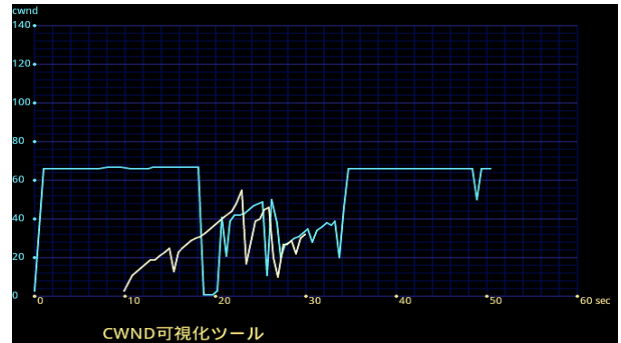


図 3: 実験 2: ミドルウェアを導入した処理

本稿では、帯域を余らせている時は、強気の TCP に切り替え、混み合った時に default に切り替えることで、複数の端末が効率良く共存することを可能にする通信システムを確立した。今後は他端末の輻輳ウィンドウ値に応じて TCP の上限値をフレキシブルに変動させる制御を開発する。独自の輻輳制御アルゴリズムは輻輳ウィンドウを増加させやすく、減少させにくい、一定値を上回らないことで、他端末の通信を阻害することなく、存在している帯域を限りなく使い切ることで、各端末の通信スループットとトータルスループットの向上を目指す。

謝辞

本研究は一部、独立行政法人情報通信研究機構の委託研究「新世代ネットワークを支えるネットワーク仮想化基盤技術の研究開発・課題ウ 新世代ネットワークアプリケーションの研究開発」によるものである。

また本研究を進めるにあたり、ご指導して下さった株式会社 KDDI 研究所の竹森敬祐さん、磯原隆将さん、株式会社 NEC Technologies の Iain Williams さんに深く感謝致します。

参考文献

- [1] android developers: <http://developer.android.com>
- [2] Sourcery G++ Lite 2008q-3-72 for ARM GNU/Linux: <http://www.codesourcery.com/>, <http://www.codesourcery.com/sgpp/lite/arm/portal/release644>
- [3] 三木香央理, 山口実靖, 小口正人: カーネルモニタを用いた Android 端末の無線 LAN 通信性能の解析と性能向上のための一検討, DICO2011, 7H-2, 2011 年 7 月。