

# Android 端末における通信性能の可視化ツール

平井 弘実 (指導教員: 小口 正人)

## 1 はじめに

近年, 携帯電話に代わる端末としてスマートフォンが爆発的に普及している. 従来の携帯電話は通話や電子メールを主な機能としたが, スマートフォンはそれらに加え, 音楽や動画を再生したり, アプリケーションを起動するなどコンピュータとしての多機能を持つ. 同時にこのようなマルチメディアデータは, ユーザが日々新しいものを web からダウンロードして楽しむことが一般的になったため, 高度な無線 LAN ネットワーク性能も備えている. 日本でも近年様々なスマートフォンが続々と市場に登場したが, これらのスマートフォンの OS としては, 主に Android, iOS, Symbian OS, BlackBerry OS, Windows Mobie が搭載されている. これらの OS の中でもシェア率が高く, またオープンソースであるため開発が完全に自由であることから, Android を研究題材として取り上げた.

## 2 研究目的

これまでも被災地やイベント会場などで, 多くの人が集中して携帯電話を利用すると回線の混雑による通信性能が低下が引き起こされたが, スマートフォンは一台で大量のデータ通信を行うことができるモバイル端末であるため, ユーザの増加に伴い, より頻繁な回線の混雑が予想される. 本研究では, この混雑状況における送信側の自主的制限を容易に解析するため, 通信時の輻輳ウィンドウサイズとその時のスループットを可視化するツールを開発した. このツールの特徴として, モバイル端末上での可視化であることがあげられる. スマートフォンは移動する端末であるため, 端末のみで測定や解析を行えたら, 大変有効である. このツールを用いて, 混雑状況での輻輳ウィンドウサイズの動きを観察し, 今後の研究に活かしたい.

## 3 輻輳制御

Android のアーキテクチャは, Linux カーネルをベースとしている. この Linux カーネル内部のトランスポート層の輻輳制御が, 通信において大きな役割を果たしている. まず輻輳ウィンドウとは, 受信側からの確認応答無しで連続送信できる最大のセグメント数を表す. データの送信側が, 自主的に通信を制限するためのパラメータである. 輻輳制御とは, 正常な通信時には確認応答を受信するごとにこの輻輳ウィンドウを増加させ, エラーが起これば減少させることで, 送信側の動きを制限する仕組みのことである. しかし, 帯域が混雑した際には, パケットロスが頻繁に起こるので, この輻輳制御によって輻輳ウィンドウサイズが急減する. そして一度急減すると元のサイズまで増加するまで通信性能が低いままとなる.

## 4 ツールの処理内容

本研究ではサーバに対してソケット通信により, 1Mbyte のバッファ転送を繰り返し, その転送状況を観察する可視化ツールを構築した. 取得したデータは, android.graphics.drawable パッケージの View クラ

スを継承してグラフを描画した. 図 1 がその画面キャプチャである. 縦軸は, 左側が輻輳ウィンドウサイズを表し, 右側がスループットを表す. スループットの単位は Mbps とする. 横軸は経過時間であり, データを取得するごとに, 輻輳ウィンドウサイズを水色, スループットを桃色で動的にプロットする. この図からスループットと輻輳ウィンドウには相関がみられることが確認できる.

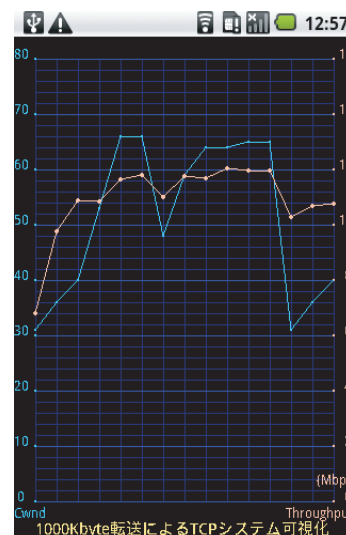


図 1: スループットと輻輳ウィンドウの可視化ツール

### 4.1 カーネルモニタ

カーネルモニタはオリジナルのツールであり, TCP の処理が行われるごとにモニタ関数によって記録を出力する. 輻輳ウィンドウサイズの他に TCP のパラメータであるスロースタート閾値, タイムスタンプ, ソケットバッファキュー長などの情報も取得することができる.

### 4.2 輻輳ウィンドウサイズ

混雑時の輻輳制御を観察するため, 輻輳ウィンドウサイズを可視化対象に選んだ. 輻輳ウィンドウは TCP のパラメータであり, ユーザからは直接見ることができないカーネル内部で制御している. そこで本研究では, カーネルモニタを利用し, 輻輳ウィンドウサイズを取得した. 1Mbyte のバッファ転送を行う際に, TCP のアルゴリズムは高速に極めて多数回実行されており, 輻輳ウィンドウサイズの変化を全て可視化することは Android 端末の限られたリソースでは, 非常に困難であるため, 一度のバッファ転送で最後のパケットを送り出した際の輻輳ウィンドウサイズを受け取ってプロットするように実装した. このようにして 1Mbyte の連続バッファ転送の輻輳ウィンドウの値の遷移を可視化した.

### 4.3 スループット

輻輳制御が, 通信性能にどのような影響を与えるかを考察するために, 通信性能の評価基準であるスループットを測定した. スループットはデータ転送量 ÷ 転送時間

で示す。バッファの送信を始める時刻と、最後の確認応答を取得した時刻との差分を転送時間とした。今回は転送量は 1Mbyte なので、 $1\text{Mbyte} \div \text{時間 (msec)} = 1000 / \text{時間 (sec)} \text{Mbps}$  と計算できる。Android の大部分のアプリケーションは Android ランタイム上で起動するように実装されているため、本研究でも Android ランタイム上のアプリケーションレベルにおけるスループットを測定するようにプログラムを実装し、これを可視化した。

## 5 無線 LAN における通信実験

### 5.1 実験概要

混雑状況において、輻輳ウィンドウサイズとスループットがどのように遷移するか可視化して解析が行えることを示すために、このツールを導入した Android 端末を無線 LAN に接続し、サーバに対してソケット通信を行った。実験方法 1 では同一無線 LAN に 1 台の Android 端末のみが通信を行った場合、実験方法 2 では同一無線 LAN に 2 台の Android 端末を接続した場合において通信実験を行った。これらの実験方法を図 2、図 3 に示す。



図 2: 実験方法 1

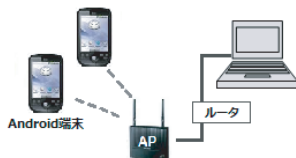


図 3: 実験方法 2

### 5.2 実験環境

無線 LAN のモードは、最大通信速度 54Mbps の IEEE802.11g を利用した。その他の実験環境を表 1 に示す。

表 1: 実験環境

Android端末	
ハードウェア	HT-03A
モデル番号	AOSP on Sapphire (US)
ファームウェアバージョン	2.1-update1
カーネルバージョン	2.6.29
アクセスポイント	
メーカー	BUFFALO
製品名	AirStation Wireless A&G HIGHPOWER
無線LAN	IEEE802.11g
サーバ	
OS	Ubuntu 10.4
CPU	Intel Pentium 4 CPU 3.00GHz

### 5.3 実験結果と考察

前節の実験環境を用いて行った実験の結果として、Android 端末が 1 台のみで通信を行った場合を図 4、2 台で同時に通信を行った場合を図 5 に示す。グラフの読み方は前章と同様である。実験方法 1 では、帯域を占有できるため輻輳ウィンドウはしばらくすると一定となり、11Mbps ~ 12Mbps の安定したスループットが得られた。一方実験方法 2 では、帯域内で競合が起きるため輻輳ウィンドウの変化が大きく、スループットも 8Mbyte ~ 10Mbyte の間で振れ幅が大きく不安定である。さらにスループットとの関係を見ていくと、輻輳ウィンドウサイズの昇降とほぼ同期して遷移していること

がわかる。

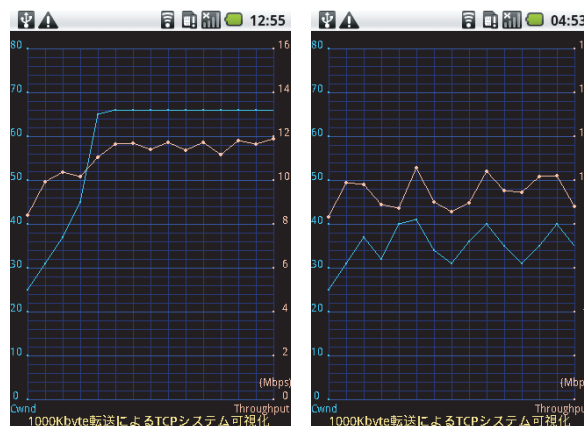


図 4: 同一無線 LAN 内に Android 端末 1 台のみの通信状況  
図 5: 同一無線 LAN 内に Android 端末 2 台の通信状況

## 6 まとめと今後の課題

### 6.1 まとめ

本研究では Android における通信のスループットとその間の輻輳ウィンドウサイズの遷移を Android 端末上で可視化できるツールを開発し、このツールを用いることにより、スループットを制御している輻輳ウィンドウサイズが見えて相関が容易に確認できるようになった。

### 6.2 今後の課題

今回の実験でわかったように、輻輳ウィンドウサイズの急減がスループットの低下に大きく影響を及ぼしている。そのため、輻輳ウィンドウサイズをできるだけ安定して大きく保ちたいのだが、TCP 輻輳制御アルゴリズムは通常パケットロスが発生するたびに輻輳ウィンドウサイズを大幅に下げている。また一度下げたまま元々のサイズに増加するまでには時間がかかってしまう。こうして現行の TCP 輻輳制御アルゴリズムでは必要以上にスループットを落としていることがあるので、大きなデータの転送を行う前に、接続している AP が実際にどの程度混み合っているかを知ることが有効である。この情報を用い、例えば複数台の Android 端末が 1 台の AP に接続した際に、同じ AP を共有する端末の通信状況を相互に知らせる機能を実装し、実際にどの程度 AP が混み合っているかを他の端末の通信状況から確認できるようにしたい。

### 参考文献

- [1] android developers: <http://developer.android.com>
- [2] W・リチャード・スティーヴンス著、橋康雄訳、井上尚司、詳解 TCP/IP Vol.1 プロトコル、ピアソン・エデュケーション
- [3] 小口正人、コンピュータネットワーク入門-TCP/IP プロトコル群とセキュリティ、サイエンス社、2007
- [4] 平井弘実、三木香央理、山口実靖、小口正人: Android 端末における通信性能の可視化ツール (情報処理学会全国大会 2011, 2011 年 3 月発表予定)