

# 無線 LAN 通信時のアプリケーションに依存する Android 通信特性に関する考察

熊谷 菜津美<sup>†1</sup> 平井 弘 実<sup>†1</sup> 三木 香央理<sup>†1</sup>  
山口 実 靖<sup>†2</sup> 小口 正 人<sup>†1</sup>

近年、スマートフォンが急速に普及しており、スマートフォンの中でも Google 社が開発した Android はシェア率が高く注目を集めている。そこで、本研究ではユーザの関心度が高い Android を研究対象とし、その通信特性について解析を行う。Android はオープンソースであり自由にアプリケーション開発できるため、数多くのアプリケーションが登場している。アプリケーションを用いることで、様々な場面でネットワークに接続し、モバイル環境においても新しい情報をリアルタイムに取得、または、発信できるようになった。Android 端末自身の通信性能向上のための研究は既になされているが、実際のアプリケーションについても考慮した通信性能の評価はまだ十分に行われていない。

そこで、本研究では、利用するアプリケーションによって、通信の振舞いどのような特性が表れるかについて解析を行う。アプリケーションは様々な通信方式のものが存在するが、本稿では、アップロード方向のアプリケーションを用いて、様々な無線 LAN 通信環境において通信する際の、Android 通信特性を調査する。

## A Study of Android Communication Characteristic Depending on Application in a wireless LAN

NATSUMI KUMATANI,<sup>†1</sup> HIROMI HIRAI,<sup>†1</sup> KAORI MIKI,<sup>†1</sup>  
SANEYASU YAMAGUCHI<sup>†2</sup> and MASATO OGUCHI<sup>†1</sup>

In recent years, the smart phone has become quickly popular, also in the smart phone, the share rate of Android developed by Google is high and attracts attention. So, in this research, we make a study of Android and analyzed about Android communication characteristics. Because Android is OpenSource and we can develop freely, many Application appear. With using Application, it connects to network in various scenes, and it became possible to get or send new information in real time in a mobile environment. Research to improve the communication performance of the Android terminal already been made, but evaluation of communication performance for the actual application has not been done yet.

In this study, by the Application to use, we analyze about what kind of characteristic appears in communicative behavior. In this paper, we investigate Android communication characteristics when one or more Android terminal uses application of the upload direction and communicates in various wireless LAN communication environment.

### 1. はじめに

近年、従来の携帯電話からスマートフォンへのシフトが急速に進み、今ではなくてはならないモバイル端末になりつつある。各キャリアによる新製品の多くがスマートフォンになり、スマートフォンに対するユーザの関心も高まっている。代表的なスマートフォンの OS には Android, iOS, Symbian OS, BlackBerry OS, Windows Mobile が存在するが、中でもトップシェアを占めているものが Android<sup>1)</sup> である。Android は Google 社によって開発されたソフトウェアプラットフォームであり、主にスマートフォンやタブレット PC などのモバイル端末に搭載されている。Android が持つ特徴としては、ソースコードをオープンソースとして無償で提供している点が挙げられる。さらに一般ユーザでも自由にアプリケーション開発を進められるため、開発コストの削減に繋がる。本研究では、スマートフォン OS の中で注目を集めている Android を研究対象とし、特にその無線 LAN 通信性能に着目した。

スマートフォンは、従来の携帯電話で主に使われてきた音声通話やメール機能に加え、パソコンとしての機能を有しているため、いつでもどこでもインターネット機能を利用できるようになった。従来の携帯電話におけるネットワークアプリケーションは、通信回線に 3G などの広域無線アクセスのみを利用していたため、通信帯域が狭く、大きなデータをやり取りするアプリケーションは実行するのが難しかった。しかし、スマートフォンでは一般に無線 LAN を使うことができるため、広帯域な通信を必要とするアプリケーションも実行可能となり、アプリケーションのバリエーションが大幅に広がった。これにより、スマートフォンに搭載されているアプリケーションを利用することで、ニュースサイトやソーシャルネットワークサービスを通じて、手軽に日々の情報を得ること、またユーザ自身からリアルタイムに情報を発信することができるようになった。

現在、様々なアプリケーションが登場しており、アプリケーションの通信方式は、実行するアプリケーションによって異なると考えられる。データの通信方向の視点からみると、

<sup>†1</sup> お茶の水女子大学

<sup>†2</sup> 工学院大学

Android 端末がクライアント端末となり、アプリケーションの種類としては、情報を発信するアップロード通信が主であるアプリケーション、反対に知りたい情報をクラウドサーバから取得するダウンロード通信が主であるアプリケーション、アップロードとダウンロードのどちらの通信も行う双方向通信のアプリケーションの3種類に分けられる。また扱うデータの視点からみると、音声や映像などのデータをリアルタイムに送受信するアプリケーション、写真や動画などの一度保存したファイルを送受信するアプリケーションに分けられる。そこで本研究では、通信方式の異なるアプリケーションを実行した際の通信特性について解析する。

本論文ではまず、2章で Android について説明する。3章では研究目的について述べる。4章では実験に利用する、我々が独自に開発したシステムツールについて説明する。5章では実験システムについて、及び、実験に使用するアプリケーションについて説明する。6章では1台の端末を用いた環境における、各アプリケーションの基本性能を評価した。7章では複数台の端末を用いた環境における、アプリケーションの通信特性について解析を行った。最後に8章でまとめと今後の課題を述べる。

## 2. Android のアーキテクチャ

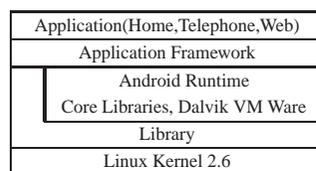


図1 Android のアーキテクチャ

図1に Android のアーキテクチャを示す。5つの層に分かれており、下がハードウェアに近い層で上にいくほどユーザーインターフェイスに近い層となっている。Android OS の中心部分は Linux であり、この部分も併せて Google から提供されている。基礎部分には Linux2.6 カーネルを採用しており、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。Linux カーネルはハードウェア上でソフトウェアの最下層に位置

し、ドライバによってデバイスの機能を Linux から利用できるようになる。Linux カーネルの上にはライブラリがあり、機能ごとにまとめられた汎用性の高いプログラム群をまとめている。このライブラリの機能は、アプリケーションフレームワークを経由して複数のアプリケーションから利用できるようになっている。Android ランタイムは Android 独自の仮想マシンである Dalvik と基本的な API を提供するコアライブラリで構成されている。Android のアプリケーションは全てこの Dalvik VM 上で動作している。その上にアプリケーションフレームワークがあり、アプリケーション動作に関係する、共通に利用される API を規定している。さらにこのアプリケーションフレームワーク上にアプリケーションがある。アプリケーションは Java で開発され、アプリケーションフレームワーク上で自由に構築することができる。

## 3. 研究目的

Android に関する研究において、無線環境での Android の通信性能について解析し、複数台の端末が通信する場合、効率良くネットワーク帯域を使用することができず、端末間で通信性能に差が生じることが確認されている。このとき、周囲の端末の通信状況を確認できた場合には、状況に合わせた通信制御を行うことで、同一アクセスポイントに繋がる全ての端末と協調して効率よく通信できることが示されている<sup>2)</sup>。周囲の状況確認には、端末同士が互いに通信状況を知らせ合うミドルウェアを開発することで、情報を共有することを目指している<sup>3)</sup>。

しかし、これらの研究において、実際のアプリケーションを用いた場合の Android 端末の通信性能に関しては考慮されていない。一般にユーザが Android 端末を利用する場合、直接操作する部分はアプリケーションである。したがって、利用するアプリケーションの通信特性にも応じて通信を制御することで、より効率の良い通信帯域の利用が可能になり、安定して通信できると考えられる。そこで本研究では、各アプリケーションの通信特性の違いに着目し、アプリケーションを実行している Android 端末の通信性能について解析する。通信方式の異なるアプリケーションの要求に応じて制御することで、より良い無線 LAN 通信環境の実現を目指す。

## 4. カーネルモニタ

本研究では、通信時のカーネル内部のパラメータの値の変化の様子を調査するため、カーネルモニタという我々が開発したオリジナルシステムツールを用いている。通常、Android

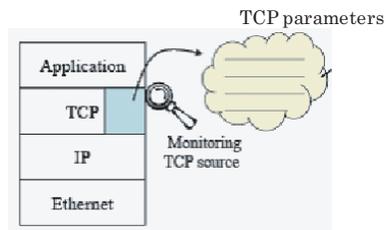


図 2 カーネルモニタ

を始めとするスマートフォンの端末では、カーネル内部の様子を知ることができない。しかし、図 2 に示すように、Android のカーネル内部の TCP ソースにモニタ関数を挿入し再コンパイルすることで、各パラメータの値をログとして記録し出力できるようになった。取得できるログには、輻輳ウィンドウサイズ、スロースタート閾値、タイムスタンプ、ソケットバッファのキュー長などの TCP パラメータが含まれており、さらに、各種エラーイベント (Local device congestion, 重複 ACK, SACK 受信, タイムアウト検出) がどのタイミングで発生したのかをモニタできる。本研究では、このカーネルモニタを Android 環境に移植したツールを用いて、アプリケーション実行時の輻輳ウィンドウの遷移の様子を解析した<sup>4)</sup>

## 5. 実験環境

### 5.1 実験システム

実験に使用した Android 端末およびサーバ機の仕様を表 1 に示す。Android 端末には、Samsung 社製の Nexus S を使用しており、前述のようにカーネルモニタを組込んで再コンパイルしたカーネルを用いている。

### 5.2 実験対象のアプリケーション

従来の携帯電話では、サイトから情報をダウンロードして楽しむ、という使い方が一般的である。一方スマートフォンでは、端末およびネットワークアクセス性能の向上により、大きなデータのアップロードを必要とするアプリケーションも動作できるようになったことから、アプリケーションもアップロードを行うものが増えている。同一アクセスポイント内の複数台の Android 端末がアップロード通信を行う場合、端末同士で通信帯域を取り合うため、この通信帯域をうまく分け合うことで通信性能が向上すると考えられる。またアップロードを行うデータは、音声や映像のような即時に通信することが求められるリアルタイム性の高いもの、単にファイルを転送するようなリアルタイム性の無いものに分けられる。

表 1 Experimental Environment

Android	Model number	Nexus S
	Firmware version	2.3.4
	Baseband version	I9023XXKB1
	Kernel version	2.6.35.7-kaori1198-ge382d80-dirty
	Build number	GRJ22
Server	OS	Linux2.6.38-10-generic
	CPU	Intel(R) Pentium(R) M processor 1.73GHz
	Main Memory	1.5GB

このことから、使用するアプリケーションとしては、アップリンク方向にリアルタイム通信、または、ファイル転送を行うタイプのものに注目し、以下の 3 つのアプリケーションを用いた。

- Ustream: ライブ動画の配信
- ES file explorer: 自分で設定した接続先のローカルサーバへ動画ファイルの転送
- Picasa Tool: Picasa ウェブサーバへ画像ファイルの転送

一般に、リアルタイム通信には UDP 通信を行う場合が多いが、ファイアウォールを超えやすいなどの理由から TCP 通信を行うものも存在する。本実験には TCP 通信を行うアプリケーションを使用した。

## 6. 基礎実験

Android 端末 1 台のみが通信する環境において、各アプリケーションを実行した時の輻輳ウィンドウの遷移の様子を比較する。実験環境を図 3 に示す。Android 端末 1 台を IEEE802.11n 無線 LAN 機能を用いてアクセスポイントに接続し、サーバに対して通信を行っている。この実験環境における輻輳ウィンドウの測定結果を図 4、図 5、図 6 に示す。図 4 は Ustream を用いた場合、図 5 は ES file explorer を用いた場合、図 6 は Picasa Tool を用いた場合の結果となっている。

### 6.1 Ustream 実行時の輻輳ウィンドウ解析

図 4 では、1 分間のストリーム配信を行っている。この結果から、比較的短い時間で輻輳ウィンドウは上がりきり、その後安定して輻輳ウィンドウサイズを保っていることが分かる。また、配信時間を延ばした場合にも遷移の様子は変わらず、輻輳ウィンドウサイズが安定し

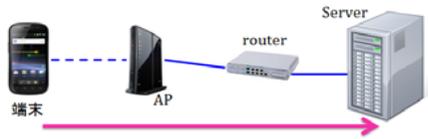


図3 実験環境 (1台通信時)

てからは一定の値を保つことを確認した。

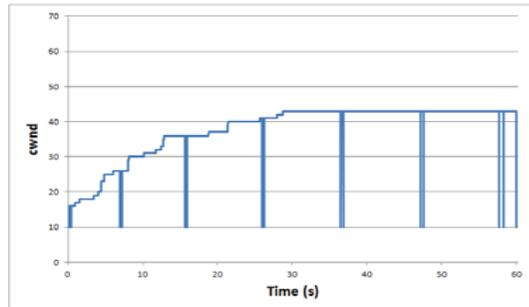


図4 Ustream 実行時の輻輳ウィンドウサイズ

### 6.2 ES file explorer 実行時の輻輳ウィンドウ解析

図5では、約67MBの動画ファイルを転送している。図4と同様、一端、輻輳ウィンドウが上がりきると、輻輳ウィンドウサイズを安定して保てるのが分かる。また、送信するファイルサイズを大きくすると、送信に掛かる時間は延びるが、遷移の様子や輻輳ウィンドウの最大値は変わらないことを確認した。

### 6.3 Picasa Tool 実行時の輻輳ウィンドウ解析

図6では、1MB程度の画像ファイルを計10枚まとめて転送している。図4、図5とは異なり、輻輳ウィンドウサイズの増減を繰り返しているが、これは1枚の画像ファイルの送信が完了するたびに、一度輻輳ウィンドウサイズを初期値まで下げているためであると考えられる。

以上の実験結果から、Android端末1台のみが通信を行う場合、アプリケーションがアップロード方向の通信を続けていると輻輳ウィンドウの値が大きくなり安定していることが分かる。

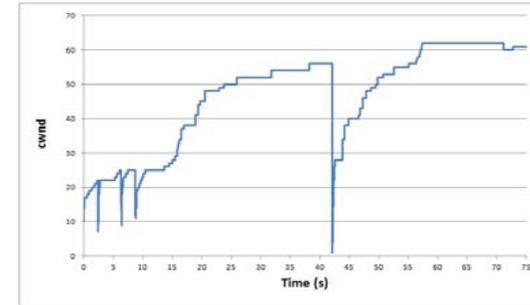


図5 ES file explorer 実行時の輻輳ウィンドウサイズ

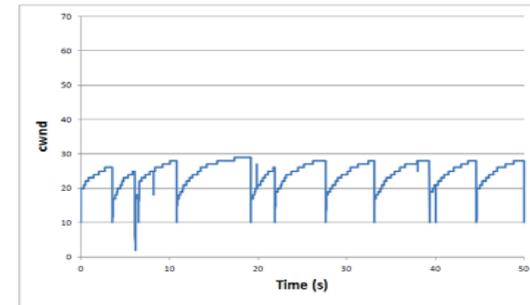


図6 Picasa Tool 実行時の輻輳ウィンドウサイズ

## 7. 複数台のAndroid端末による同時通信実験

次に、複数台のAndroid端末が通信する環境において、アプリケーションを実行した時の通信性能の測定を行う。今回は実行するアプリケーションにUstreamを用いる。Ustreamは、最適なストリーム品質を保つために必要な通信速度を最低300kbpsとしている。イベント等の高画質配信が求められる場合でも、500-700kbpsあれば良く、少ない帯域で高品質なストリーム配信を行えると予想される<sup>5)</sup>しかし、同一アクセスポイントに複数台のAndroid端末が繋がる環境においては、端末同士が通信帯域を取り合うため、確実に必要な帯域を確保できるとは限らない。そこで複数台の通信端末の内、1台のAndroid端末のみUstreamを実行した場合に、このUstreamを実行した端末の輻輳ウィンドウ、配信映像の見え方にはどのような変化が現れるかを検証する。

## 7.1 基本性能測定

実際のアプリケーションを用いる前に、Android 端末自身のスループット測定を行う。スループット測定には iperf を用いた。図 7 に実験環境を示す。

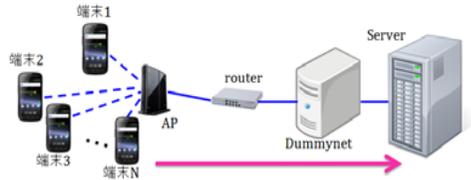


図 7 アプリケーションを実行しない Android 端末のスループット測定環境

まず、3 台の Android 端末が同一アクセスポイントに繋がって同時にサーバーに対して通信する環境において、最大で 8 台の Android 端末まで増やして実験を行った。Ustream でストリーム配信する際、遠隔地に存在するサーバに対してアクセスすると考えられるため、高遅延環境における通信性能を測定するために、dummynet を用いて人工的に遅延を発生させた。また Ustream は TCP 通信を行うことから 1 台の端末は TCP 通信に固定している。背景端末も含めた全ての端末が TCP 通信を行う場合と、1 台の端末は TCP 通信を行い、背景端末は UDP 通信を行う場合の 2 つの測定環境において 60 秒間通信を行い、TCP 通信に固定した端末のスループット及び輻輳ウィンドウサイズを比較する。図 8 に、往復遅延時間 (RTT) が 256(ms) の時のスループットを示す。また図 9、図 10 に 8 台の Android 端末で通信した時の輻輳ウィンドウサイズを示す。

図 8 より、背景端末が TCP 通信を行う場合は、Android 端末を増やしてもスループットの値に大きな差は見られないことが分かる。しかし、背景端末が UDP 通信を行う場合は、Android 端末の台数が多いほどスループットが低下していくことが分かる。これは UDP 通信を行う背景端末が、ネットワークが混んでいるかどうかにかかわらず大量の packets を送り続け、帯域を占領したことで、TCP 通信に固定した Android 端末が十分な帯域を確保できなかったため、スループットが上がらなかったと考えられる。また、8 台の Android 端末が同時に通信し、背景端末が UDP 通信を行う時、TCP 通信を実行する端末のスループットが、配信に必要な最低スループットに近い値まで低下することから、この環境において、TCP 通信に固定した Android 端末上で Ustream を用いた場合、配信映像の品質に劣化が生じることが予想される。図 9、図 10 から、どちらも頻繁に輻輳ウィンドウサイズを増減さ

せているが、図 10 では、輻輳ウィンドウサイズが高い値まで回復するのに多く時間が掛かることが分かる。

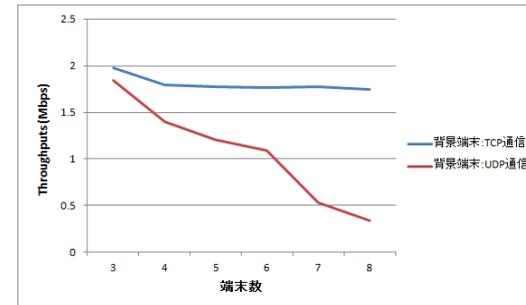


図 8 スループット比較

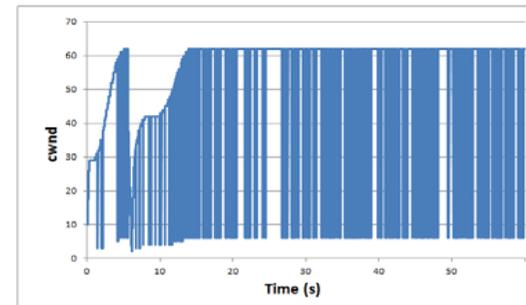


図 9 輻輳ウィンドウサイズ (背景端末:TCP 通信)

## 7.2 Ustream を用いる Android 端末の性能評価

次に、実際にアプリケーションを用いて実験を行う。図 11 に実験環境を示す。前節の結果を受け、8 台の Android 端末が同一アクセスポイントに繋がる環境において、1 台の端末は Ustream を使い、7 台の背景端末では UDP 通信を行う場合に、配信映像の品質にどのような影響が表れるかを調査し、さらに、輻輳ウィンドウの解析を行う。

配信映像には 2 パターンの品質劣化があり、映像が途切れてしばらく経過した後、配信されない場面がある場合、または、映像が途切れてしばらく経過した後、続きの映像が流

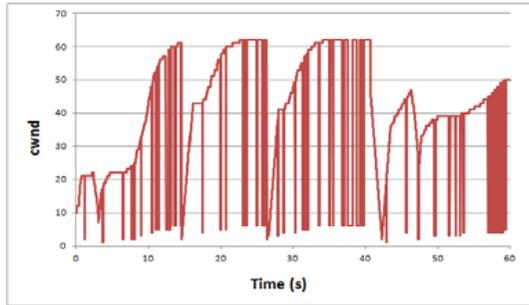


図 10 輻輳ウィンドウサイズ (背景端末:UDP 通信)



図 11 1 台の Android 端末のみ Ustream を実行する環境

れる場合が確認された。表 2、図 12 に 90 秒間のうち映像が途切れる回数と時間、輻輳ウィンドウサイズを示す。表 2 では 2 パターンの品質劣化が合わせて 5 回以上起こった部分を網掛けしており、半数を超えていることが分かる。また映像が途切れる回数が少なくても、長い時間、映像が途切れていることが確認できる。図 12 より、輻輳ウィンドウサイズも図 4 の場合と比べ安定しないことを確認した。

## 8. まとめと今後の課題

本稿では、様々な環境において、通信方式の異なるアプリケーションを実行した時の通信性能を測定した。アプリケーションの通信方式の違いに着目し、1 台の Android 端末上で、リアルタイム通信を行うアプリケーションとファイル転送を行うアプリケーションをそれぞれ実行した際の Android の通信特性を比較した。また複数台の Android 端末を通信させた際、TCP 通信と UDP 通信を行う端末が混在している場合、TCP 通信を行う端末のスループットが低下することを確認した。さらに、8 台通信において、1 台の端末のみリアルタイ

表 2 配信映像が途切れる回数と時間

試行	配信されない場面あり	続きから配信	合計	途切れる時間
1	2	4	6	1s,2s
2	1	5	6	2s
3	2	2	4	2s,3s
4	1	2	3	7s
5	1	2	3	2s
6	2	2	4	1s,2s
7	10	0	10	1s,2s(2 回),3s(2 回), 4s,9s,10s(2 回),13s
8	1	4	5	4
9	2	6	8	1s,7s
10	2	6	8	1s,2s
平均	2.4	3.7	5.7	4s

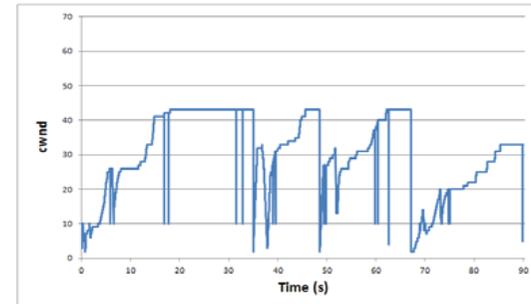


図 12 Ustream を実行する Android 端末の輻輳ウィンドウサイズ

ム通信アプリケーション (TCP 通信) を用い、背景端末では UDP 通信を行う環境では、背景端末にネットワーク帯域を奪われてしまいリアルタイム通信端末上で配信する映像の品質が低下することを確認した。

今後は、何台の Android 端末を同時通信させると品質が低下し始めるのが調査していく。また、品質劣化は段階的に生じるのか、ある台数までは品質を一定に保てるのかどうかについても検証していく。

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、磯原隆将さんに深く感謝致します。

#### 参 考 文 献

- 1) Android:<http://www.google.co.jp/mobile/android>
  - 2) 三木香央理, 山口実靖, 小口正人: 無線 LAN 通信環境におけるカーネルモニタを用いた TCP 解析による Android 端末の性能向上手法, DEIM2012, C6-5, 2012 年 3 月.
  - 3) 平井弘実, 三木香央理, 山口実靖, 小口正人: Android 端末を用いた無線通信時の制御ミドルウェアに関する考察, 電子情報通信学会 NS 研究会, NS2011-105, 2011 年 11 月
  - 4) 三木香央理, 山口実靖, 小口正人: Android 端末におけるカーネルモニタの導入, Comsys 2010, 2010 年 11 月
  - 5) Ustream:<http://helpcenter.ustream.tv/jp/content/>
-