1J-2

カーネルモニタを用いたAndroid端末の無線LAN通信の解析

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, スマートフォン市場の成長に伴い, 携帯端末で動作する組込み機器のソフトウェアプラットホームとしてGoogle 社開発の Android が注目されている [1]. 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 が動作する組込み機器は, 汎用の PC などとはアーキテクチャが異なるため, インタフェースやリソースの問題から, 通信などの動作時に, 組込み機器の中でどのような事が起こっているのか, 正確に把握することは難しく, その通信動作を解析する事は興味深い. 本研究では, 組込み機器において, カーネルの中の振舞を把握することができるカーネルモニタツールを開発し,Android のトランスポート層においてこれを動作させた. これを用いて, 組込み機器の通信時の内部動作を解析することが可能である事を示し, システムプラットホームとしての Android に焦点を当て, 特にそのネットワーク能力およびネットワークコンピューティング能力について掘り下げてる.

2. 実験システム

2.1 Android のアーキテクチャ

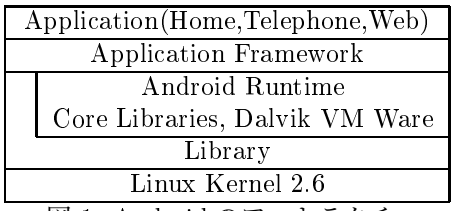

図 1: Android のアーキテクチャ

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築されており, 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成している. 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（JVM）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 Dalvik にあわせて開発すればよく, ポータビリティが高い. 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, カスタマイズ自由度が高く, 他キャリアおよび他機種への柔軟な拡張性があるといえる.

一方, 通信については Linux カーネルの中のプロトコルスタックを用いて行われているため, 主にその TCP 実装部分などで性能が決まってくると考えられる. そこで, 本研究ではカーネル中のトランスポート層実装に焦点を当て評価を行う.

2.2 カーネルモニタ

カーネルモニタは通信時に, どの時間にカーネルのコードのどの部分が実行されて, その結果カーネル内部のパラメータの値がどのように変化したかを記録する事ができるツールである [4].

図 2 に示すように, カーネル内部の TCP ソースにモニタ関数を挿入しカーネルを再コンパイルすることで TCP パラメータをモニタ可能にしている. これによりモニタできるようになった値には, 輻輳ウィンドウ, ソケットバッファのキュー長の他, 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) の発生タイミングなどがある. カーネルモニタよって, 正常動作時のカーネルの振舞を知る事ができ, さらには通信において問題が生じている場合に, その問題を特定し何が起こっているのか調べる事も可能となる.

本研究ではこのカーネルモニタを組み込み機器である Android に応用する.

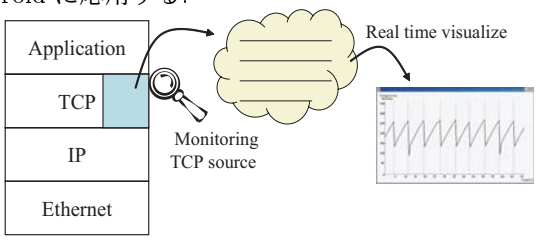

図 2: Kernel Monitor

2.3 Android 端末のためのカーネルモニタ構築

Android は Linux カーネルをベースとしているが, 組込み機器であり汎用の PC と異なる点も多数ある. またシステムもアプリケーションも開発はクロスコンパイルを用いる必要があり, OS のビルトは特殊な方法を用いて行い, さらにコンパイルした OS を Android 携帯の実機で立ち上げるためにも特別な手順が必要となる.

本研究では, Android 携帯の実機向けの汎用 PC にお

An Analysis of performance of communication on Android terminals in a wireless LAN environment with Kernel Monitor
† Kaori Miki ‡Saneyasu Yamaguchi, and †Masato Oguchi
Ochanomizu University (†)
Kogakuin University(‡)

けるカーネルモニタと同様のツールをを開発し，これをクロスコンパイルによりOSのコードに埋め込み，Android携帯の実機に送り込んで立ち上げ，カーネルモニタが動作する事を確認した．そしてカーネルモニタを動作させて,Androidの通信時の内部動作を解析することが可能である事を示す．今回はその一例として,Androidの通信における輻輳ウィンドウサイズとスループットの関係について解析する．

3. 実験システムと基本性能測定

表1に本研究の実験環境を示す．本研究では．スループット測定にはiperf-2.0.4[2]をクロスコンパイルし,Androidに送り込んでソケット通信の性能を測定した．クロスコンパイラとしてはarm-2008q3[3]を使用した．

表 1: 実験システム

Android	Model number	AOSP on Sapphire(US)
	Firmware version	2.1-update1
	Baseband version	62.50S.20.17H_2.22.19.261
	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 CPU 3.00GHz
	Main Memory	1GB

4. 複数台のAndroid端末通信時の性能

この章では複数台のAndroid端末を1台のアクセスポイントを使って通信させた時の通信性能をカーネルモニタを用いて評価する．

4.1 2台通信

今回は図3に示すように2台のAndroid端末を使って実験を行った．この環境で実験を行った結果，2台で公

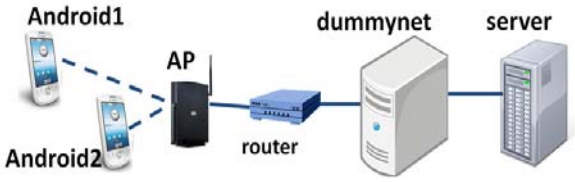

図 3: 2台のAndroid端末による通信時の実験環境

平に通信できる場合と，1台のみが帯域を安定して使用し，もう1台の通信は不安定になるという2パターンの結果が起こり得ることがわかった．その時のスループットと輻輳ウィンドウの関係をカーネルモニタを用いて解析する．

図4,5に1台が不安定な通信になる場合のスループットと輻輳ウィンドウサイズを示す．このとき，Android端末と通信相手のサーバ間のRTTは256(ms)でパケットロスは入れていない．Android2は輻輳ウィンドウが安定しており，スループットも安定した結果が得られたのに対し，Android1は輻輳ウィンドウが安定せず，ス

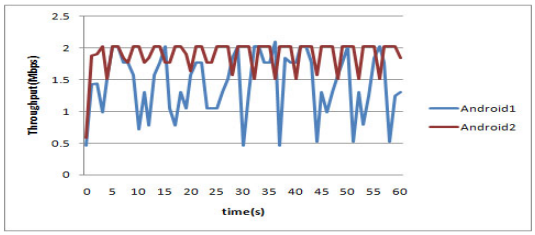

図 4: 2台通信時TCP通信スループット

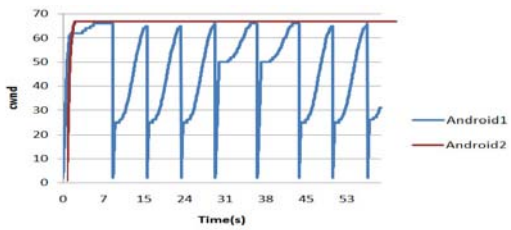

図 5: 2台通信時，輻輳ウィンドウサイズ

ループットも不安定な結果になることが確認された．この不安定な際の結果に対し，安定しているときは両者のスループットはほぼ等しく，輻輳ウィンドウサイズも2台とも安定しており，一度上昇すると下がらないことが確認された．基本的に2台とも同等の端末で，単独では安定した通信を行うことができるが，タイミングや周囲の状況次第で結果が分かれていることが推測される．

5. まとめと今後の課題

本論文ではAndroid実機にカーネルモニタツールを適用した．これを用い，Android端末の通信時における輻輳ウィンドウの値を解析した．その結果，汎用PCのカーネルモニタとほぼ同様な使い勝手でAndroidのカーネル内部の振舞を解析することが可能であることを示せた．またカーネルモニタを用いて輻輳ウィンドウとスループットの関係を解析することができた．今後はこのモニタツールを用いて輻輳ウィンドウ以外の様々なパラメータを解析し,Androidの特徴を見つけ，その詳細な振舞について研究を進めていきたい．また複数台のAndroid端末を通信させた際，2台とも安定した通信ができる場合とそうでない場合に分かれるためその原因を究明していきたい．

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
[3] Sourcery G++ Lite 2008q-3-72 for ARM GNU/Linux:http://www.codesourcery.com/, http://www.codesourcery.com/sgpp/lite/arm/portal/release644
[4] Reika Higa, Kosuke Matsubara, Takao Okamawari, Saneyasu Yamaguchi, and Masato Oguchi, "Analytical System Tools for iSCSI Remote Storage Access and Performance Improvement by Optimization with the Tools, "In the 3rd IEEE International Symposium on Advanced Networks and Telecommunication Systems (ANTS2009), December 2009.