

Android 端末におけるバッテリー消費に関する考察と制御手法について

理学専攻 情報科学コース 小柳 文乃 (指導教員: 小口 正人)

1 はじめに

近年スマートフォン端末は爆発的に普及しており、我々の生活に無くてはならないものとなりつつある。そのため、スマートフォンのバッテリーの持ち時間の改善は非常に重大な課題となっている [1]。スマートフォンのバッテリー消費の原因は主に「ディスプレイ」「通信機能」「便利機能」の三点があげられる。ディスプレイの明るさや画面の回転などは、ユーザの好みで自由にカスタマイズするものであり、バッテリーの持ちのためにスマートフォンが使いにくくなってしまふことは好ましくない。しかし、「通信機能」については、ユーザ個人で通信によるバッテリー消費を改善することは難しい。そこで本研究では通信、特に多数台の端末が同時に通信する劣悪な環境に着目をする。

2 実験概要

本章では本研究の実験環境について説明する。

2.1 実験環境

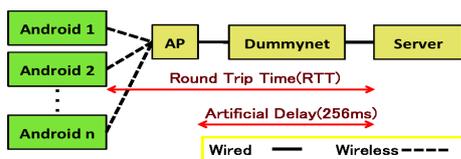


図 1: 実験環境

実験は図 1 で示す様にサーバ、ダミーネット、AP は有線で繋ぎ、Android 端末を AP に接続させる。バッテリー残量、データ送信量を記録する事ができるように、カーネルにコードを加える改変を行った Android 端末を用いる。スリープ状態になることを防ぐため、端末がスリープすることを防ぐ自作のアプリケーションを起動させ、iperf を用いて Android 端末を AP に 30 分間接続し通信させる。

2.2 評価方法

評価式は式 1 で表す。本研究では、バッテリー消費の大小やデータ送信量の大小をただ単純に比較するのではなく、少ないバッテリー消費でいかにデータ送信量を大きくできるか評価する。この評価方法によって、例えばデータ送信量が同じ場合でもバッテリー消費が小さい方が評価は高くなる。

$$\text{バッテリー消費 1\%あたりの送信量} = \frac{30 \text{ 分間のデータ送信量 (MBytes)}}{30 \text{ 分間のバッテリー消費量 (\%)}} \quad (1)$$

3 関連研究

3.1 通信制御ミドルウェアの導入

基本性能調査として、AP に同時接続する端末台数と、AP と端末の距離がバッテリー消費やデータ送信量にどのような影響を与えるのか測定を行った。その結果 AP に同時接続する端末台数が多い程、また、AP と端末の距離が遠い程、バッテリー消費 1%あたりの送信量

が悪化することを確認できた。よって本研究では、そのような劣悪な環境でいかに通信効率を上げられるかを考える。

先行研究で開発されたカーネルモニタリングツールと輻輳制御ミドルウェア (MW) により、カーネル内部の処理をリアルタイムでユーザ空間から監視し、輻輳ウィンドウ値やエラーイベントの発生タイミングなどを元に制御可能となった。具体的には同一 AP に接続した端末間でお互いの接続状況を把握し、その組み合わせを予測し、輻輳ウィンドウの上限値を設定する。可用帯域を公平に分け合うことで、全体の通信速度と公平性の向上に成功している [2]。

この通信制御を導入し、バッテリー消費とデータ送信量への影響を調査した。その結果を図 2 に示す。

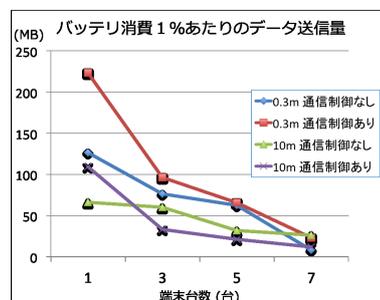


図 2: 通信制御 MW の効果

図 2 より、通信制御手法は距離が短い場合、端末台数に関わらず通信効率の向上に効果があることが確認できた。しかし AP と端末の距離が 10m の場合、端末台数が増加した際に通信制御手法を用いていない方がバッテリー消費の効率は良くなっている。

以上のことから、通信制御手法は距離に応じた柔軟な制御が必要であるといえる。

4 提案手法

4.1 提案手法の概要

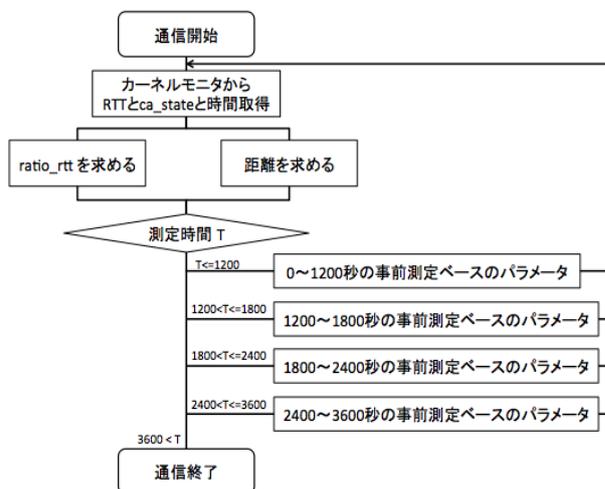


図 3: フローチャート

前章より、輻輳制御 MW は距離を考慮しなくては、かえってバッテリー消費と送信量の比を悪化させてしまう場合もある。そこで本研究では、距離に応じた柔軟な制御手法を提案する。なお、端末台数は5台とする。

本提案手法は図3で示す流れで制御を行う。前章で紹介したカーネルモニタから、通信中 RTT、エラーイベント、時間を取得し、その値を元に通信の混雑具合、AP と端末の距離、経過時間を算出する。それらから適切なパラメータを用いて制御を行う。

【混雑具合の算出方法】

リアルタイムで取得した RTT と min_rtt(通信中で最も小さい RTT) の差分を取ることで、現在のトラフィックの混雑具合を判断する。

【距離の算出方法】

エラーイベントから成功率を求める。通信成功率と距離は相関関係であると明らかになっているため距離が算出できる。

【パラメータの設定方法】

AP と端末の距離が 0.3m と 10m の地点で事前測定を行い、その事前測定で 10 分毎の評価が最も高かったパラメータを「事前測定ベースのパラメータ」と設定した。この値を元に、3m, 5m, 7m の最適なパラメータを式 2, 3 から求めた。

$$X_{m_max} \text{ 値} = \frac{10m_max \text{ 値} - 0.3m_max \text{ 値}}{10 - 0.3} \times (X - 0.3) + 0.3m_max \text{ 値} \quad (2)$$

$$X_{m_min} \text{ 値} = \frac{10m_min \text{ 値} - 0.3m_min \text{ 値}}{10 - 0.3} \times (X - 0.3) + 0.3m_min \text{ 値} \quad (3)$$

算出した距離が 0~2m の場合 0.3m の、2~4m の場合 3m の、4~6m の場合 5m の、6~8m の場合 7m の、8m~ の場合 10m のパラメータ値を設定をする。

4.2 性能評価

図4は測定開始 0~30 分の、図5は測定開始から 30~60 分の測定結果である。

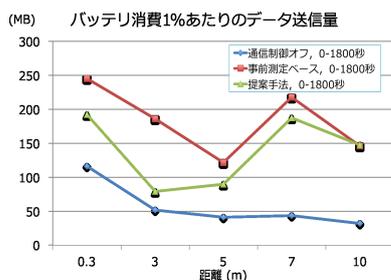


図 4: 実験結果

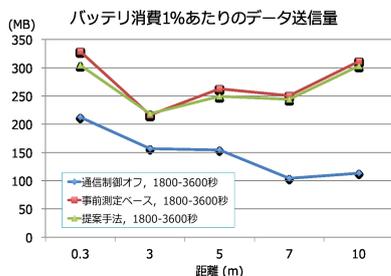


図 5: 実験結果

図より、提案手法は距離・時間に関係なく通信制御を行わない場合より性能が良くなることが確認できた。

しかし、測定開始 0~30 分では事前測定ベースのパラメータを設定した結果より性能が大幅に悪化していることから、距離を算出する過程で誤差が生じていると考えられる。一方で、測定開始 30~60 分では事前測

定ベースのパラメータを設定した結果と同等の性能を出すことが確認できた。よって本提案手法は、時間経過とともに性能が良くなり、事前測定ベースパラメータを設定した場合と同程度の性能を表すことから距離の算出結果も正確になるといえる。

4.3 同時接続する端末の距離が異なる場合の評価

4.2 節より、提案手法は距離の自動判別を行うことから柔軟な通信制御を実現が実現できた。しかし、実世界では全ての端末が同じ距離で通信するとは考えにくい。そこで、AP までの距離を 0.3m, 3m, 5m, 7m, 10m と全て異なるものとし、提案手法で正しく距離が算出できるのか、性能にどのような影響を及ぼすのか調査を行った。

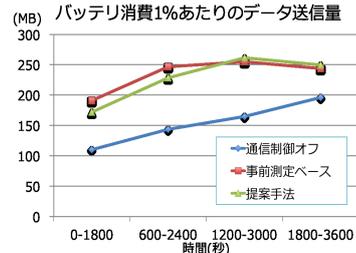


図 6: AP に同時接続する端末距離が異なる通信時の結果

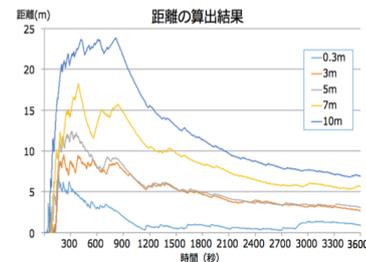


図 7: 時間経過における距離の算出結果

結果より、算出された距離は時間経過とともに正しい値に収束していくことが確認できた。また、本手法は事前測定ベースと同程度の性能を示す。以上のことから、事前測定ベースは無制限通りあるパターンの中であくまで有限回の測定を元に設定している。しかし想定しきれない未知の状況下でも十分な性能を示した。

5 おわりに

本研究では、スマートフォンのバッテリー消費の原因として、通信による消費に着目をした。先行研究で提案された通信制御手法を用いると、台数が多い場合の通信効率を上げることができているが、距離が遠い場合はうまく働いていない。そこで距離に応じた柔軟な制御手法を提案した。提案手法により、距離の算出やバッテリー消費あたりのデータ送信量の向上に成功した。

今後は距離測定の誤差を小さくするために、距離を算出するための手法を改良し、より性能のよい制御手法を用いて更なるバッテリー性能の向上を目指す。

参考文献

- [1] IDC: <http://www.idc.com/getdoc.jsp?containerId=prUS25037214>
- [2] 早川愛, 山口実靖, 小口正人: 無線 LAN-AP における TCP ACK パケット蓄積回避のための協調的輻輳制御手法の提案と実装, DEIM2015, C2-2, 2015 年 3 月.