Vol.2017-MBL-84 No.13
Vol.2017-CDS-20 No.13
2017/8/30

無操作状態スマートフォン端末の通信デバイス制御による消費電力低減に関する一考察

村上翼[†1]　栗原駿[†1]　福田翔貴[†1]　小口正人[†2]　山口実靖[†1]

近年，スマートフォンやタブレット PC などのモバイル端末の普及率が高くなっている．スマートフォンにおける重要な課題としてバッテリの持ち時間があり，この低減が重要な課題となっている．スマートフォンでは，無操作状態にあってもアプリケーションがバックグラウンドで通信を行い，電力が大きく消費される．よって，通信デバイスを無効化することにより無操作状態における消費電力を低減できると考えられる．本稿では，通信デバイスの有効化と無効化をくり返すことにより無操作状態における消費電力を低減する方法に着目し，適切な無効状態時間の推定手法について考察する．通信デバイスを無効化し一定時間後に再度有効化を行うと，デバイスの無効化処理と有効化処理が行われ一時的に消費電力が増加してしまう．よって，消費電力の低減を目的にデバイスを一時停止するには，デバイスの無効化と有効化のために消費される電力を上回る電力を，デバイスの無効化により低減させなくてはならない．本稿では，一時的な無効化により消費電力が増加してしまった量と，無効化により低減できた電力が等しくなるデバイス停止時間 BET(Break Even Time)と呼ぶ．有効化と無効化を繰り返すことにより消費電力を低減させるには，この BET を知る必要がある．本稿では，BET を推定する手法を 2 つ提案し，それらの評価を行い有効性を示す．

Saving Power Consumption of Smartphones in the Screen-off State with Disabling the Wi-Fi Device

MURAKAMI TSUBASA[†1]　KURIHARA SHUN[†1]　FUKUDA SHOKI[†1]
OGUCHI MASATO[†2]　YAMAGUCHI SANEYASU[†1]

1. はじめに

近年，スマートフォンやタブレット PC などのモバイル端末の普及率が高くなっており，これらは重要な機器の一つとなっている．スマートフォンにおける最も重要な課題はバッテリの持ち時間であるとの報告があり[1]，これらの端末における消費電力の低減は非常に重要な課題であると言える．

スマートフォンは，無操作状態であってもアプリケーションが動作し，通信などが行われバッテリを消費することがある[2][3][4][5][6]．この無操作状態端末におけるアプリケーションの通信による消費電力を削減する方法として，Wi-Fi やセルラーなどの通信デバイスを無効状態にする方法が考えられる．多くのアプリケーションにおいて，無操作状態の期間に一時的に通信不能状態に陥ることは想定されており，一時的な通信デバイスの無効化は大きな問題にならないと予想される．ただし，アプリケーションの多くは何らかの趣旨を持って通信を行っており，通信不能の状態を長期間保つことはサービスの品質に影響を与えることも予想される．よって，無操作状態における通信による電力消費を低減させるには，多くの時間帯において通信デバイスを無効状態にしておき消費電力の低減を図り，定期的に通信デバイスを有効状態としてサービスの品質の低減を抑制する手法が効果的であると考えることができる．

デバイスを基本的に無効，停止状態にしておき，使用する時間帯のみ短期的に有効，稼働状態にする手法は，HDD などのデバイスでも用いられており，消費電力に効果があることが確認されている[7]．多くのデバイスにおいて，状態の移行(例えば無効状態から有効状態への移行)を行うと移行のための処理が必要となり，移行処理のために電力を消費することとなる．よって，消費電力の低減を目的にデバイスを一時的に無効状態にするには，デバイスの無効化のために必要となる移行処理により消費される電力を上回る電力を，デバイスの無効化により低減させなくてはならない．当然，無効時間が短いほど無効化により低減できる電力が少なく，長いほど低減できる電力が大きい．状態の移行により消費電力が増加してしまった量と，無効化により低減できた電力が等しくなるデバイス無効時間は BET (Break Even Time)などと呼ばれ[7]，本稿においてもこれを BET と呼ぶ．

前述の通り，スマートフォンやタブレット PC においては消費電力の低減が重要であると考えられる．これらの機器にて通信デバイスの有効化と無効化を繰り返すことにより消費電力を低減させる手法を適用する場合，これらの端末における BET の調査が必要となる．本稿では，Android 端末を搭載したタブレット端末に着目し，アプリケーショ

†1 工学院大学
Kogakuin University
†2 お茶の水女子大学
Ochanomizu University

ⓒ2017 Information Processing Society of Japan

ンがインストールされているタブレット端末における BET の推定方法についての考察を行う．具体的には，観測電流値の積分により推定する手法と，平均電流より求める手法を提案し，性能評価によりそれらの有効性に示す．

Android は，スマートフォンやタブレット PC 向けに開発されたモバイル端末用の OS であり，2016 年第 3 四半期において世界のモバイル OS において 86.8%の高いシェアを持っており[8]，非常に重要なスマートフォンプラットフォームの一つとなっている．また，電流の値などをソフトウェア的に取得可能である．よって，本稿では本 OS を用いて実装と評価実験を行った．

本稿の構成は以下の通りである．2 章にて，無操作状態スマートフォンの消費電力の調査結果を示す．3 章にて，通信デバイス断続的無効化による消費電力の低減手法について説明する．4 章にて，BET 算出手法の提案を行う．5 章にて評価を行う．6 章にて考察を述べ，7 章にて本稿をまとめる．

2. 無操作状態スマートフォンの消費電力調査

Android 搭載のスマートフォン端末では，通信デバイスが有効化状態にある場合は無操作状態においてもアプリケーションが通信を行い，これにより消費電力が増加する．このことから，ユーザがスマートフォン端末を使用していない場合は通信デバイスを無効化することにより消費電力を低減できると期待できる．ただし，通信デバイスの有効化や無効化の処理に電力を消費するため，有効化や無効化直後に一時的に消費電力が増加することも予想される．本章にて，通信デバイス有効状態と無効状態の単位時間当たりのバッテリ減少量と平均電流，有効化や無効化直後の一時的な電流の増加量の調査結果を示す．なお本稿では，電圧は一定であり，電流と単位時間あたりの消費電力は比例すると仮定し，電流の測定を持って単位時間あたりの消費電力の調査とした．

2.1 通信デバイス有効，無効状態の消費電力

Android 搭載のスマートフォンに 20 個のアプリケーションをインストールし，無操作状態で放置してバッテリ残量が 100[%]から 50[%]に至るまでの時間を測定した．インストールしたアプリケーションは，2016 年 11 月 6 日の Google Play Store ランキング上位 20 個のアプリケーションである．測定端末は Nexus7 (2013)を使用し，通信デバイスは無線 LAN(Wi-Fi)を使用した．使用端末の仕様は表 1 の通りである．測定するにあたり，バッテリ残量は kernel 内の `/sys/class/power_supply/battery/capacity` の値から，電流値は`/sys/class/power_supply/battery/current_now` の値より得た．一部のアプリケーションにおいて，通信を行ったときにスクリーンを OFF 状態から ON 状態に変更することが確認された．本稿では，スクリーンの点灯や消灯の影響を排除して通信デバイスによる消費電力に着目して考察を行うために，スクリーンを WakeLock し続けるアプリケーションを自作しこれを動作させながら測定を行った．WakeLock は `SCREEN_DIM_WAKE_LOCK` により行い，非常に暗い状態で点灯している状態である．

測定結果を図 1，2，3 に示す．図 1 は，通信デバイス有効，無効状態でバッテリ残量が 100[%]から 50[%]に至るまでのバッテリ残量の推移を示している．図 2 は，通信デバイス有効状態でバッテリ残量が 100[%]から 50[%]に至るまでの電流の変化を示している．図 3 は，通信デバイス無効状態でバッテリ残量が 100[%]から 50[%]に至るまでの電流の変化を示している．

図 1 より，通信デバイス有効状態では 100[%]から 50[%]に至るまでに 655 分を要し，無効状態では 731 分を要していることがわかる．これより，有効状態では無効状態より高い速度でバッテリを消費していることがわかる．また 1 分あたりのバッテリ減少量は，有効状態で 0.075[%/分]，無効状態で 0.067[%/分]であった．図 2，3 の電流値の平均を求めると，有効状態の平均値が-148[mA]，無効状態の平均値が-135[mA]となっており，有効状態の方が無効状態よりも電流の絶対値が大きくなっていることがわかる．

表 1，使用端末の仕様

Device Name	Nexus7(2013)
OS	Android 6.0
CPU	Snapdragon S4 Pro(APQ8064) 1.5GHz (Quad Core)
Memory	2GB

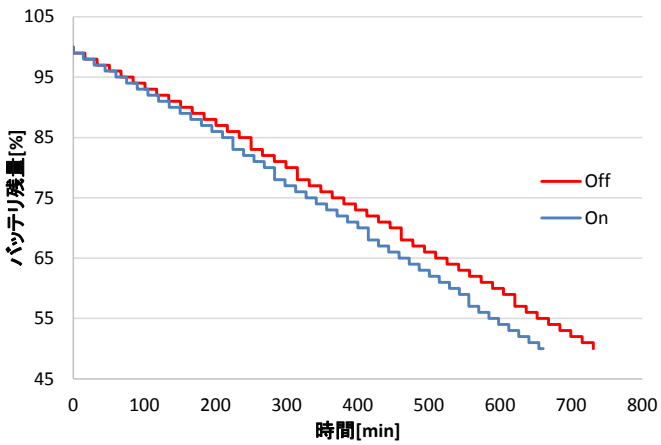

図 1，Wi-Fi 有効，無効状態でのバッテリ残量推移

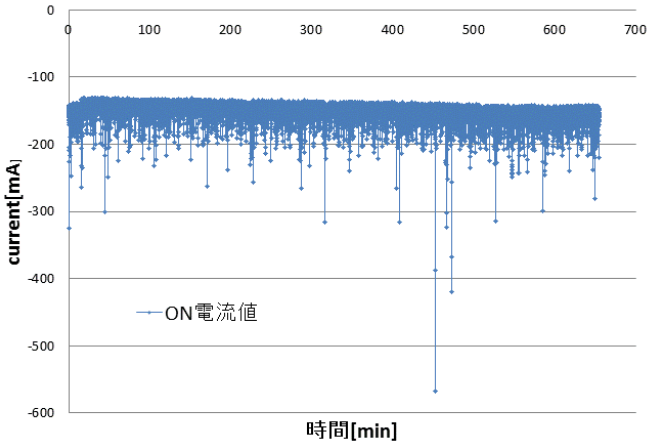

図 2，Wi-fi 有効状態の電流推移

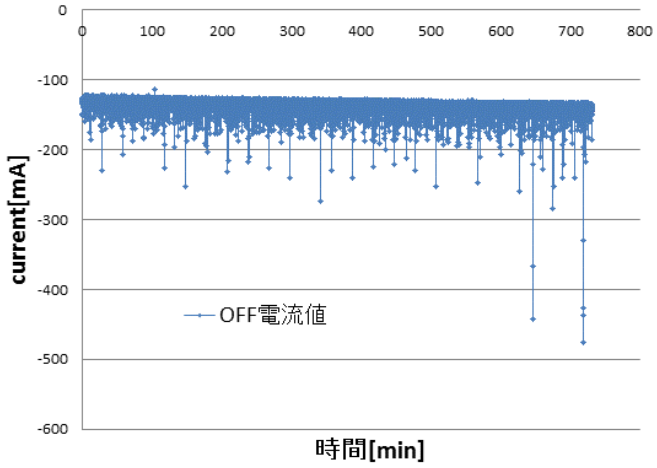

図 3，Wi-fi 無効状態の電流推移

2.2 通信デバイス有効化，無効化直後の一時的電流増加量

通信デバイスの有効化や無効化の直後の一時的な電流増加を観察した．観察結果を図 4，5，6 に示す．図 4 は，通信デバイスを 20 秒の時点で無効化し，120 秒の時点で有効化したときの電流変化を示している．図 5 は無効化時，図 6 は有効化時の観察結果を拡大したものとなっている．

図 4 より，20 秒時点と 120 秒時点で電流の絶対値が増加していることがわかる．これより，通信デバイスの有効化や無効化の時に一時的な電流の絶対値の増加があることが確認できる．無効化により一時的に増加する量は，図 5 の 20 秒から 23 秒の電流値と有効状態における平均電流の差をこの区間で積分することにより求まり，約-357[mA 秒]であった．同様に有効化により一時的に増加する電流の量も図 6 を積分することにより求められ，-482[mA 秒]であった．この積分値については次章で詳しく述べる．

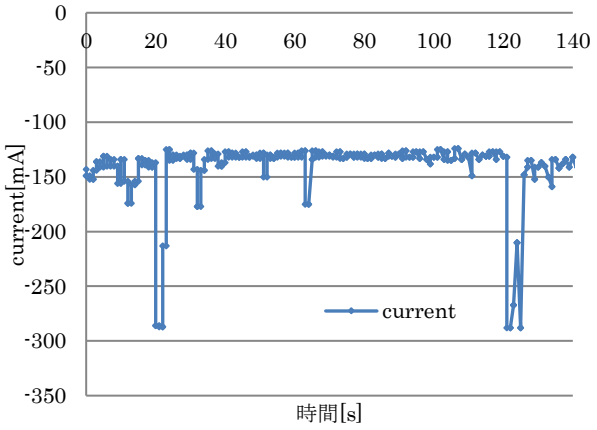

図 4，通信デバイス有効化，無効化時の電流推移

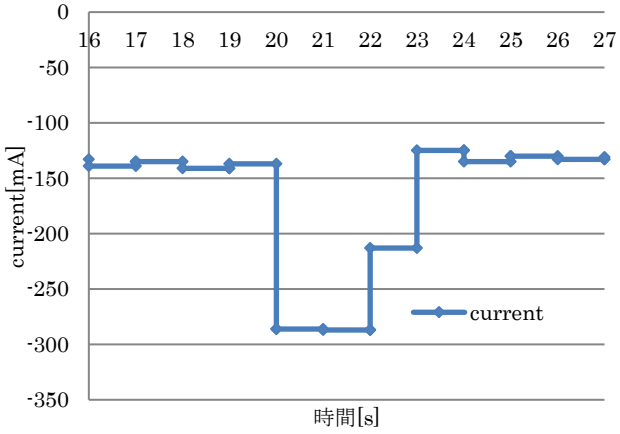

図 5，通信デバイス無効化直後の電流の推移

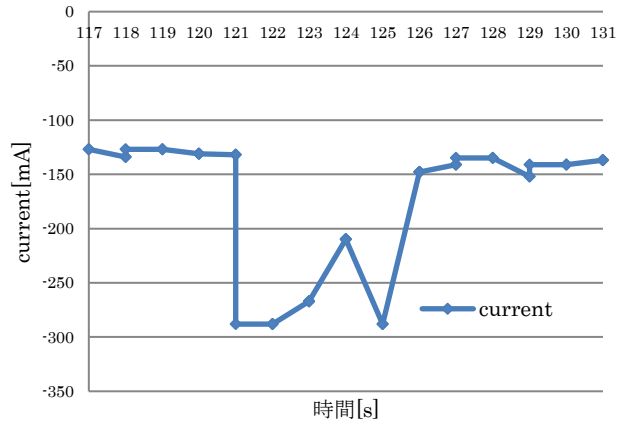

図 6，通信デバイス有効化直後の電流推移

3. 通信デバイスの断続的無効化による消費電力の低減

本章にて，通信デバイスの断続的無効化による省電力手法と，その BET について述べる

2 章の測定結果より，通信デバイスが有効状態と無効状態を比較すると無効状態の方が低消費電力であること，通信デバイスの有効化や無効化の直後に一時的に消費電力が増加することが確認された．これらより，通信デバイスを

通常は無効状態にしておき，定期的に短期間有効状態にすることにより，定期的に通信を行いながらも消費電力の低減を実現できると期待できる．通信デバイスの有効化と無効化を行った時の消費電力変化の模式図を図 6 に示す．A_1 の面積は通信デバイス無効化に伴う消費電力増加量，A_2 の面積は通信デバイス有効化に伴う消費電力増加量，B の面積は通信デバイスを無効状態にすることよって低減された消費電力量，b の長さは通信デバイス有効時と無効時の消費電力の差である．図 6 より消費電力の低減を目的に通信デバイスを無効化する場合は

$$A_1 + A_2 < B \quad (1)$$

である必要がある．また，

$$A_1 + A_2 = B \quad (2)$$

となる時間を本稿 BET と呼ぶ．

通信デバイスの断続的無効化により消費電力を低減させるには，通信デバイスを BET より長い時間無効状態にする必要がある．しかし，一般に BET は与えられていないため，本手法を用いるためには BET の推定を行う必要がある．

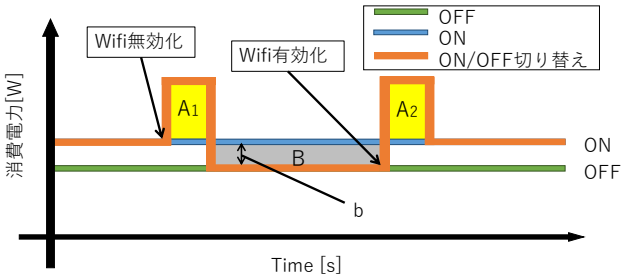

図 6，通信デバイス有効化／無効化時の消費電力変化

4. BET 推定手法の提案

本章にて，通信デバイスの断続的無効化による消費電力低減手法の BET を，電流の値を観察することにより推定する手法を 2 つ提案する．

4.1 積分手法

まず，積分手法を提案する．3 章で述べたように，式(2)が成立する無効状態時間が BET である．これを得るためには，A_1，A_2，b の値を得る必要がある．積分手法では，A_1，A_2 の値を観察された電流値を積分することにより求める．b の値は，無効状態電流と有効状態電流の平均値の差を求めることにより得る．BET は，A_1，A_2 の合計値を b で割ることによって求められ，

$$BET = (A_1 + A_2) / b \quad (3)$$

となる．

ただし，この積分手法では非常に時間の短い A_1，A_2 の値を正確に計測する必要があり，誤差が生じやすいと予想される．この手法は文献[7]にて用いられている BET 推定手法をナイーブに用いた手法であるといえる．

4.2 平均電流ベース手法

次に，平均電流ベース手法を提案する．本手法の有効化と無効化のループのイテレーション 1 回の消費電力のモデルを図 7 に示す．i_{on} は通信デバイス有効時の平均電流，i_{off} は無効時の平均電流，t_{com} は有効化と無効化の処理に要する時間，t_{on} は通信デバイスを有効にしている時間，t_{off} は無効にしている時間，A は通信デバイス有効化と無効化により増加した電流量の和となっている．また，通信デバイスの有効化と無効化を繰り返した場合の平均電流を i_{loop} と呼ぶ．図 7 より i_{loop} は，

$$i_{loop} = \frac{A + i_{on}(t_{com} + t_{on}) + i_{off} \cdot t_{off}}{t_{com} + t_{on} + t_{off}} \quad (4)$$

となる．この式を用いることにより，A 以外の値から A を算出することが可能である．

また，無効時間である t_{off} を x とすると BET は，

$$i_{loop} = i_{on} \quad (5)$$

をみたす x が BET となる．

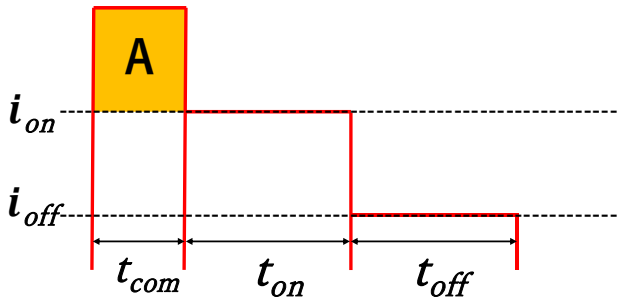

図 7，有効化/無効化ループのイテレーション 1 回の消費電力モデル

以上より，ある無効時間における測定値を式(4)に代入し A を求め，求められた A を式(5)に代入し x を求めることにより BET を推定できることとなる．この求め方を平均電流ベース手法と呼ぶ．

この手法は積分手法とは異なり長期間の計測値である平均電流から求めるため，より正確な推定ができると期待できる．

5. 性能評価

本章では，提案手法の性能評価を行う．性能評価の方法としては，アプリケーションが 20 個または 50 個インストールされている端末を用意し，積分手法と平均電流ベース手法により BET を推定し，その精度を評価する手法を用いる．アプリケーション 20 個としては 2016 年 11 月 6 日の Google Play Store ランキング上位 20 件を，アプリケーション 50 個としては 2017 年 2 月 18 日の Google Play Store ランキング上位 50 件を用いた．BET 推定に必要な電流値を測定するにあたり，端末は Nexus7(2013)を使用し，2 章と同様の条件にて測定を行った．BET 推定の評価をするにあた

ⓒ2017 Information Processing Society of Japan

り，Wi-Fi デバイスの有効時間を 30 秒で固定し，無効時間を 50 秒，100 秒，150 秒，200 秒，250 秒，300 秒，400 秒の 7 通りに変化させ，バッテリ残量が 100%から 75%に至るまでの時間の計測を行った．

推定結果，測定結果を表 3，4，図 8，9，10，11 に示す．表 3，図 8，図 9 はアプリケーションを 20 個インストールした状態の結果，表 4，図 10，図 11 はアプリケーションを 50 個インストールした状態の結果である．表 3，表 4 は無効時間ごとの積分手法による推定結果，図 8，図 10 は平均電力ベース手法による推定結果，図 9，図 11 は無効時間ごとのバッテリ残量 100%-75%に至るまでの時間の測定結果を示している．

まず 20 アプリケーションインストール状態での評価を行う．図 3 より，積分手法の推定 BET は OFF 時間によらずほぼ一定であることがわかる．ここでは，平均の 55 秒を推定結果とする．図 8 より，平均電力ベース手法の推定 BET は推定 i_{loop} と i_{on} の値が重なる 90 秒という結果になった．また図 9 の測定結果から，実際の BET は 90 秒付近となっている．よって，積分手法に比べ平均電力ベース手法の方がより高い精度で BET を推定しており，その推定値は実際の BET に近いことがわかる．

次に，50 アプリケーションインストール状態での評価を行う．積分手法の推定結果は，表 4 より無効時間によらずほぼ一定であることがわかる．ここでは平均の 53 秒を同手法の推定結果とする．平均電力ベース手法の推定結果は，図 10 より推定 i_{loop} と i_{on} の値が重なる 90 秒であるということがわかる．また図 11 の測定結果から，実際の BET は 100 秒付近となっていることがわかる．よって，50 アプリケーションの実験においても積分手法に比べ平均電力ベース手法の推定 BET は実際の BET に近いことがわかる．

以上より，平均電流ベース手法のほうが精度が高く，同手法は実 BET に近い BET を推定できたことが確認できた．

表 3，積分手法による推定結果(20 アプリケーション)

OFF time[s]	BET[s]
50	54.5
100	56.6
150	55.3
200	54.6
300	53.2
400	53.0

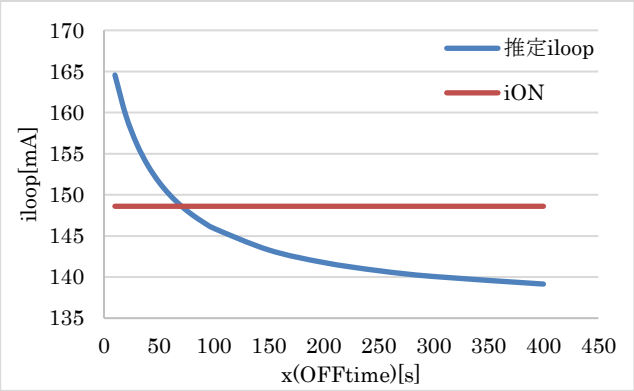

図 8，平均電力ベース手法による推定結果
(20 アプリケーション)

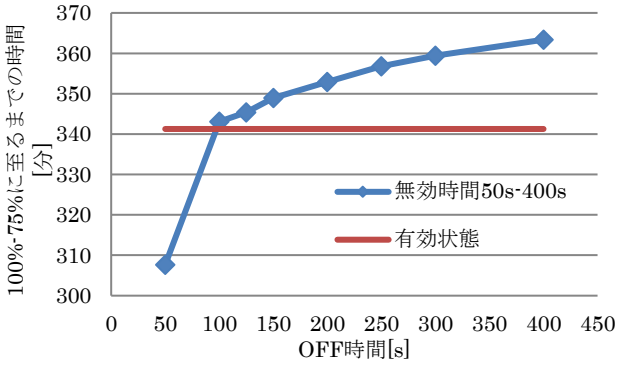

図 9，OFF 時間ごとのバッテリ持ち時間
(20 アプリケーション)

表 4，積分手法による推定結果(50 アプリケーション)

OFF time[s]	BET[s]
50	54.8
100	49.5
150	53.5
200	54.3
300	52.4
400	53.0

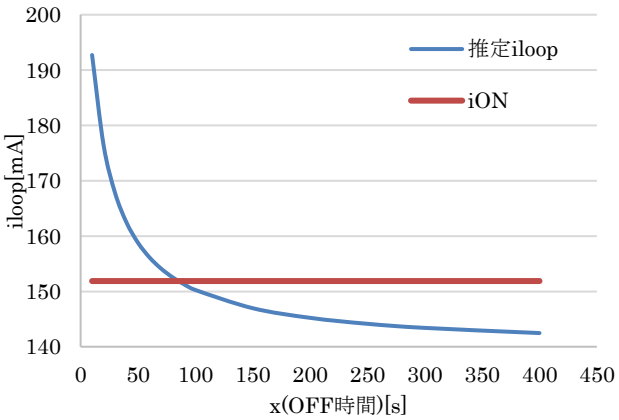

図 10，平均電力ベース手法による推定結果
(50 アプリケーション)

ⓒ2017 Information Processing Society of Japan

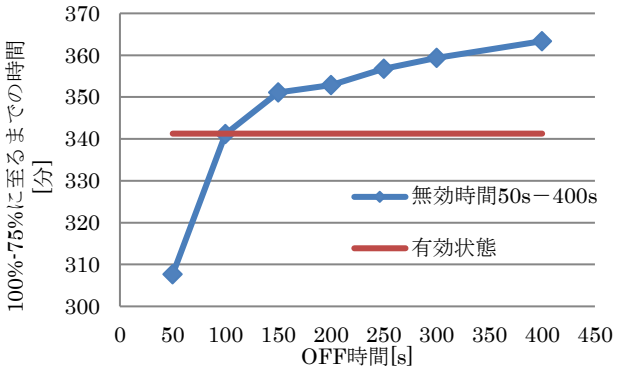

図 11，OFF 時間ごとのバッテリ持ち時間
(50 アプリケーション)

6. 考察

通信デバイスを無効状態すると，アプリケーションが通信をできなくなる欠点が生じると予想できる．ただし，スマートフォンユーザの多くがエレベータの中や携帯電話の電波の届かない地下などに入り一時的に通信不可能な状態になる．よって，アプリケーションの多くは一時的に通信不可能な状態になっても大きな問題とならないように設計されていると予想でき，通信デバイスを頻繁に無効化する使用方法であっても効果的に使用できる例は多いと期待できる．

通信デバイスの無効化を繰り返すことが大きな弊害となってしまう例として，以下のような事例が考えられる．あるアプリケーションが巨大な通信を開始するが，一時的に通信デバイスが有効になっている期間内に通信を終えることができず，その通信は中断されてしまう．このアプリケーションは再度通信デバイスが有効になった時に転送を再開することができるが，アプリケーションが転送をレジュームせずに最初から転送を開始するように実装されていると，通信デバイスが有効になるたびに失敗の通信を行うことになってしまい，転送効率の側面においても消費電力の側面においても好ましくない状況となる．この課題は，アプリケーションが通信再開時に前回の通信の続きを行うように実装することにより回避が可能であると考えられる．

7. おわりに

本稿では，無操作状態の Android 端末の消費電力に着目し，通信デバイスの断続的無効化手法における BET の推定手法について積分手法と平均電流ベース手法を提案した．そして，実際に断続的無効化を無効時間を変えて測定し提案手法の評価を行い，平均電流ベース手法の方が積分手法より正確な推定が可能であるということを確認した．

今後は，他端末での検証，インストールされているアプリケーションを変えての評価を行う予定である．

謝辞

本研究は JSPS 科研費 26730040，15H02696，17K00109 の助成を受けたものである．

本研究は、JST、CREST JPMJCR1503 の支援を受けたものである．

参考文献

[1] 日本経済新聞 2013 年 4 月 1 日
http://www.nikkei.com/article/DGXNASFK2600W_W3A320C1000000/

[2] Shun Kurihara，Shoki Fukuda，Ayano Koyanagi，Ayumu Kubota，Akihiro Nakarai, Masato Oguchi, Saneyasu Yamaguchi，“A Study on Identifying Battery-Draining Android Applications in Screen-Off State”， 2015 IEEE 4th Global Conference on Consumer Electronics，2015-10

[3] Kurihara, S., Fukuda, S., Hamanaka, S., Oguchi, M., and Yamaguchi, S. 2016. Identifying Battery-Draining Applications by Monitoring Behavior in Screen-Off State in Android. In IEEE International Conference on Consumer Electronics-Taiwan (Nantou, Taiwan, May 27-29)

[4] 早川 愛，磯村 美友，竹森 敬祐，山口 実靖，小口 正人，"Android 端末省電力化のためのブロードキャストインテント情報の調査"，マルチメディア、分散、協調とモバイル DICOMO2014 シンポジウム, pp. 1461-1468, 2014.

[5] 小西 哲平，稲村 浩，川崎 仁嗣，神山 剛，大久保 信三，太田賢，“画面オフ状態におけるバックグラウンドタスク同時実行による Android 端末の省電力化”，情報処理学会論文誌, 55 巻, 2 号, pp. 587 － 597, 2014

[6] 中村 優太，早川 愛，半井 明大，竹森 敬祐，小口 正人，山口実靖，“Android 端末におけるインストールアプリケーションとブロードキャストインテント発行による電力消費に関する一考察”, DBS モバイル・地理情報システム, 2014/8

[7] Saneyasu Yamaguchi and Shunsuke Yagai, "Power Effective File Layout with Application Support in Virtualized Environment", 2015 IEEE International Conference on Control System, Computing and Engineering (ICCSCE2015)

[8] Smartphone OS Market Share,2016Q3
http://www.idc.com/prodserv/smartphone-os-market-share.jsp

ⓒ2017 Information Processing Society of Japan