

通信デバイス断続的無効化による省電力手法の BET 推定とその正確度の評価

村上 翼^{†1} 神山 剛^{†2} 福田 晃^{†3} 小口 正人^{†4} 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おり、スマートフォンにおける重要な課題として消費電力の低減があげられている。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、電力を大きく消費してしまう。我々は過去に、通信デバイスの有効化と無効化を繰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手法に着目し、平均電流を用いた BET 推定手法を提案した。そして、ランキング上位アプリケーションと単一の端末を用いた推定正確度評価により、その手法の有効性を示した。しかし、同手法が実際のユーザの使用環境のアプリケーションセットにおいても有効であるかや、他の端末においても有効であるかなどの検証は行われていない。本稿では同手法の推定正確度を、実際のユーザの利用実態を考慮した環境や、他端末での環境において評価し、同手法が実際のユーザの環境や複数の端末においても有効であることを示す。

1. はじめに

スマートフォンやタブレット PC は無操作状態においてもバッテリーを非常に消費するため、消費電力を減らすことは重要な課題となっている[1][2][3]。スマートフォンアプリケーションは、スクリーン OFF 状態においてユーザの操作なしでもバックグラウンドで実行されている。これらは通信デバイスを使用し通信しており、これによりバッテリーを大量に消費する[4][5]。この消費電力を低減する手法として、Wi-Fi デバイスやセルラーデバイスなどの通信デバイスを一時的に無効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、端末が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重大な問題を起こさないと予想される。また通信デバイスの長期間の無効化は、ユーザーエクスペリエンスの低下を招く。したがって、バッテリーを節約するためには、多くの時間帯において通信デバイスを無効化しておき、ユーザーエクスペリエンスの低下を防ぐために定期的に短期間通信デバイスを有効にする手法が効果的であると考えられることができる。

デバイスを基本的に無効にしておき、使用する時間帯のみにおいて短期的に有効にする手法は、ハードディスクドライブ(HDD)などのデバイスでも用いられており、消費電力の低減に効果的であることが確認されている[6][7][8][9]。多くのデバイスにおいて、デバイスの有効化、または無効化にするための状態移行処理は消費電力を一時的に増加させる。したがって、頻繁なデバイスの状態移行は逆に消費電力を増加させてしまう。よって、消費電力低減を目的に通信デバイスを一時的に無効状態にするには、デバイスの

無効化と有効化のために必要となる移行処理により消費す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。当然ながら無効時間が短いほど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。状態の移行により増加してしまう消費電力量と、無効にしたことにより低減された消費電力量を比較、これらが等しくなるデバイス無効時間は BET(break even time)と呼ばれ[11][12][13]、本稿においてもこれを BET と呼ぶ。

スマートフォンやタブレット PC に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 の情報が必要となる。しかし、これらの端末において BET 情報は提供されていない。我々は過去の緩急にて、Android を搭載した端末に着目し、複数のアプリケーションがインストールされている端末における BET の推定方法を二つ提案した。具体的には、観測した電流値の積分により推定する手法と、観測した電流値の平均電流より推定する手法を提案した。そして、タブレット端末とアプリケーション配布サイトランキング上位のアプリケーションを用いた推定正確度の評価を行い、同評価環境においては平均電流手法が高い正確度で推定できることを示した。ただし、同手法が実際のユーザの使用環境のアプリケーションセットにおいても有効であるかや、他の端末においても有効であるかなどの検証は行われていない。本稿では、平均電流より推定する手法に着目し、複数の環境における BET 推定の正確度を評価し、考察する。具体的には、他端末である Nexus5X を用いた BET 推定正確度の評価、実際のアプリケーション利用実態を考慮した環境下での BET 推定正確度の評価を行う。

本稿の構成は以下の通りである。2 章にて、無操作状態

†1 工学院大学
Kogakuin University

†2 九州大学 スマートモビリティ研究開発センター
R&D Center for Smart Mobility, Kyushu University

†3 九州大学
Kyushu University

†4 お茶の水女子大学
Ochanomizu University

端末における消費電力の調査結果を示す。3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やそのためのBETの推定手法などの、既存の研究を紹介する。4章にて、BET推定手法の性能評価をする。5章にて考察を行う。6章にて関連研究を紹介し7章にて本稿をまとめる。

2. 無操作状態端末における消費電力調査

前述のようにAndroid搭載のスマートフォン端末では通信デバイスが有効状態の場合は無操作状態であってもバックグラウンドでアプリケーションが通信を行い、これにより電力を消費する。よって、通信デバイスを無効状態にすることで消費電力を低減できると期待できる。

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ー減少量の調査結果を示す。前述同様に、電圧は一定であり、電流と単位時間当たりの消費電力は比例すると仮定し、電流の測定をもって消費電力の測定としている。

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ー残量が100%から75%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Nexus5X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、バッテリー残量は`/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電力へ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測定を行った。

測定結果を図3に示す。図3は通信デバイス有効、無効状態でバッテリーが100%から75%に至るまでのバッテリー残量の推移を示している。図3より、通信デバイス有効状態では100%から75%に至るまでに310分を要し、無効状態では347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ーを消費していることがわかる。平均バッテリー減少速度は、有効状態にて4.83 [%/h]、無効状態にて4.32 [%/h]であった。

表1, 測定端末の仕様

Device Name	Nexus5X
OS	Android 7.1.2
CPU	Qualcomm Snapdragon 808 1.8GHz (2) + 1.4GHz (4)
Memory	2GB

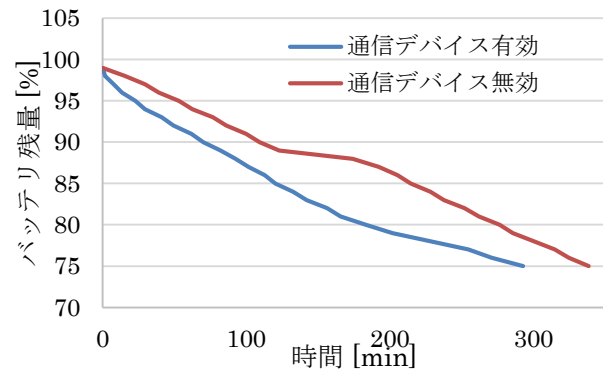


図3, 通信デバイスの状態(有効/無効)とバッテリー減少速度

3. 通信デバイスの断続的無効化とそのBETの推定

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推定手法について述べる。

3.1 通信デバイスの断続的無効化による消費電力の低減

通信デバイスが有効状態のときと無効状態のときの消費電力を比較すると無効状態の方が低消費電力である。また、通信デバイスの有効化や無効化の直後に、移行処理によって一時的に消費電力が増加する[12][13]。これより、通信デバイスを通常は無効状態にしておき、定期的に短時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1に示す。A₁の面積は通信デバイス無効化の処理に伴う消費電力増加量、A₂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に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1より消費電力の低減を目的に通信デバイスを一時的に無効化する場合は

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

である必要がある。また、

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

となるときは無効状態時間は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。

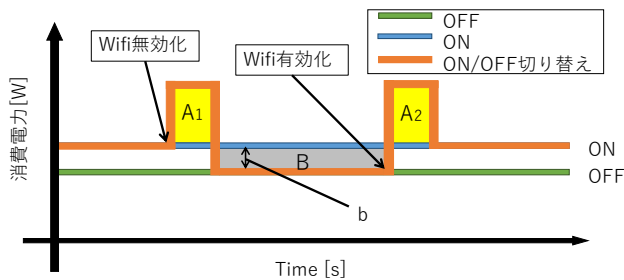


図1 通信デバイス有効化/無効化時の消費電力変化

3.2 BET の推定(積分手法)

まず、積分手法[12][13]を紹介する。前節で述べたように、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。本稿では、電圧は一定であり、電流と単位時間当たりの消費電力は比例すると仮定し、電流の観察をもって単位時間当たりの消費電力の観察とする。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[6]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。

3.3 BET の推定(平均電流ベース手法)

次に、平均電流ベース手法[12][13]を紹介する。前節同様に、電流の観察をもって単位時間当たりの消費電力の観察とする。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2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} は通信デバイスが有効である時間、 t_{off} は無効である時間、 A は通信デバイス有効化と無効化により増加する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2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。これを A について解くと、

$$A = (t_{com} + t_{on})(i_{loop} - i_{on}) + t_{off}(i_{loop} - i_{off}) \quad (5)$$

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となる。

また、無効時間である t_{off} を x とすると BET は、

$$i_{loop} = i_{on} \quad (6)$$

をみたす x が BET となり、

$$x = \frac{A}{i_{on} - i_{off}} \quad (7)$$

となる。

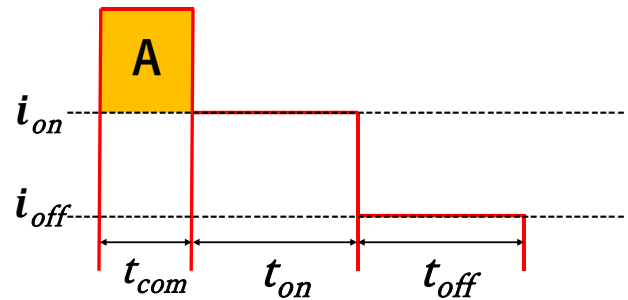


図2, 有効化/無効化ループのイテレーション1回の消費電力モデル

以上より、ある無効時間における測定値を式(5)に代入し A を求め、求められた A を式(7)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。

3.4 積分手法と平均電流ベース手法の評価

文献[12][13]にて、積分手法と平均電流ベース手法の推定精度の評価が行われ、平均電流ベース手法が高い精度で BET を推定できることが確認できている。ただし、評価は単一端末(Nexus7)においてのみ行われ、評価対象アプリケーションはランキングトップのアプリケーションに限定されており、現実のアプリケーション利用実態を考慮した評価は行われていない。

4. 性能評価

本章にて、平均電流ベース手法[12][13]の性能評価を行う。

4.1 Nexus5X による評価

まず、既存研究で用いた Nexus7(2013)とは別端末である Nexus5X を用いた性能評価について述べる。対象アプリケーションを2016年11月6日の Google Play Store ランキング上位20件とし、これらをインストールし平均電流ベース手法を用いて BET を推定し、その正確度を評価した。BET 推定に必要な電流値を測定するにあたり、3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り、

Wi-Fi デバイスの有効時間を 30 秒で固定し、無効時間を 25 秒, 50 秒, 100 秒, 150 秒, 200 秒の 5 通りに変化させ、バッテリー残量が 100%から 75%に至るまでの時間からバッテリー減少速度の調査を行った。

推定結果、測定結果を図 4, 5 に示す。図 4 は Nexus5X における平均電流ベース手法を用いた BET の推定結果であり、図 5 は無効時間ごとのバッテリー減少速度の測定結果を示している。

まず図 4 より、式(4)より計算した値である無効時間ごとの平均電流と、測定値である有効状態の平均電流のグラフが重なっている場所が BET となる。よって推定 BET は 52.1 秒という結果になった。また、図 5 の測定結果から、実際の BET は 53.9 秒となった(50 秒と 100 秒の結果より一時近似)。両図を比較し高い正確度で推定ができたことがわかる。

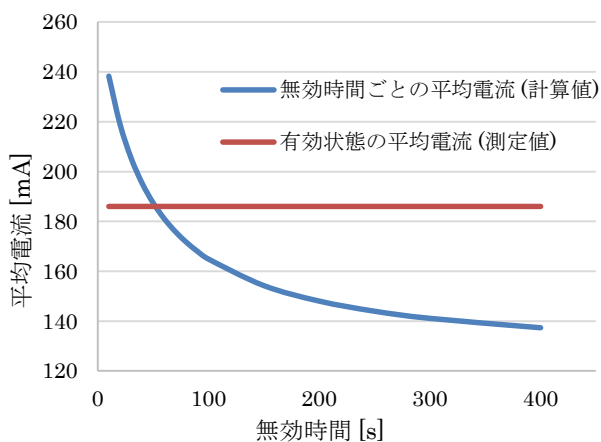


図 4, 平均電流ベース手法における無効時間と平均電流の関係の推定結果

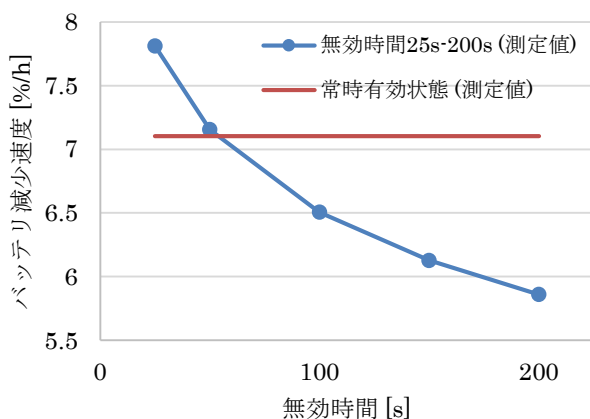


図 5, 無効時間とバッテリー減少速度(測定値, Nexus5X)

4.2 実アプリケーションモデルを用いた評価

次に現実のアプリケーションの使用モデルに基づいてアプリケーションをインストールし、BET 推定の評価を行う。文献[14]で紹介されているスマートフォン利用モデルを元に 11 アプリケーションの利用モデルを作成し、そのアプリケーションセットをインストールした。評価するにあたり、端末は Nexus7(2013)を使用し、前節と同様の条件で

測定を行った。BET 推定の評価をするにあたり、Wi-Fi デバイスの有効時間を 30 秒で固定し、無効時間を 25 秒, 50 秒, 100 秒, 150 秒, 200 秒, 250 秒, 300 秒の 7 通りに変化させ、バッテリー残量が 100%から 75%に至る時間からバッテリー減少速度を求めた。

図 6 に、平均電流ベース手法を用いた BET の推定結果を、図 7 に実際の BET の測定結果を示す。

まず図 6 にて、式(4)より計算した値である無効時間ごとの平均電流と、測定値である有効状態の平均電流のグラフが重なっている場所が BET となる。よって推定 BET は 55.9 秒という結果になった。また、図 7 の測定結果から、実際の BET は 48.0 秒となった(同様に一時近似を使用)。両結果を比較すると、低くない正確度で推定ができていくことがわかる。

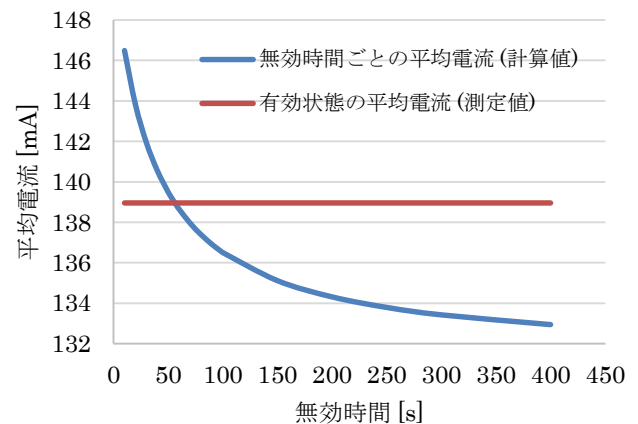


図 6, 平均電流ベース手法における無効時間と平均電流の関係の推定結果(実アプリケーション)

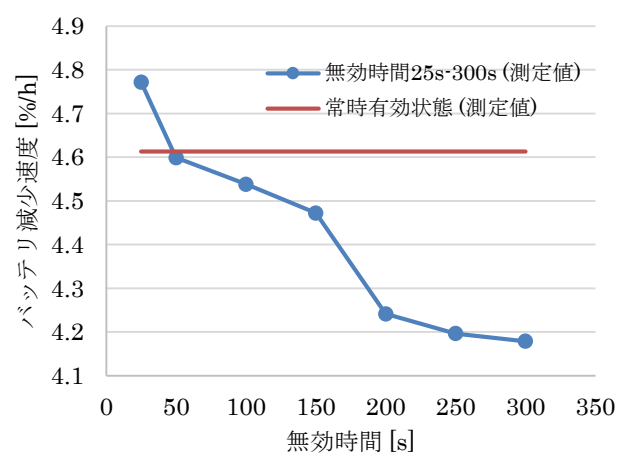


図 7, 無効時間とバッテリー減少速度(測定値, 実アプリケーション)

5. 考察

本稿では、通信デバイスの断続的無効化による消費電力低減手法について述べた。実際に通信デバイスを無効にすると、アプリケーションのサービス品質を低下させると予

想できる。しかし、一時的無効化と同様の状況は、エレベーターの中、地下、飛行機の中、トンネル内などで日常的に生じていると考えられる。したがって、アプリケーションの多くは一時的に通信ができない状況になっても重大な問題を引き起こさないように設計されていると期待でき、通信デバイスを頻繁に無効化する使用法は大きな問題なく使用できると考えられる。

通信デバイスを断続的に無効化する省電力手法が有効に活用できる状況の例として、ユーザが睡眠などの理由により端末を数時間触れずに放置する場合は考えられる。このような状況では、常に端末の状況を最新に保つ必要はなく、一定時間ごとに短期間の通信を行えばよいと考えられる。また、起床時などの再利用時に端末の状態が最新の状態と比較して著しく遅れている場合は、再利用時に端末の状態(メール受信の状態など)を最新の状況に更新するのに長い時間を要し、大きなユーザーエクスペリエンスの低下を招くと予想される。

6. 関連研究

この章では、関連する研究について説明する。

まず、Android のスマートフォンに標準で搭載されている省電力方法として Doze モードというものがある。スマートフォン端末がバッテリー駆動、スクリーン OFF かつ静止状態の場合に Doze モードに移行する。これにより生じる制限として、ネットワークアクセスの停止、SyncAdapter や Job Scheduler の停止、WakeLock の取得を無視、AlarmManager の無視、GPS と WiFi スキャンの停止などがある。次に Android のスマートフォンに標準で搭載されている省電力方法としてバッテリーセーバーがある。このモードが ON になると、バイブレーション機能、位置情報サービス、ほとんどのバックグラウンドでのデータ通信が制限され、メールやメッセージなどの同期を行うアプリケーションが起動しないと更新されないようになる。ただし、これらの省電力方法は通信デバイス自体を無効化しているわけではない。

次に PC の消費電力に関する研究として、以下のものがある。Mahesri らはラップトップシステムにおける消費電力の分析を行っている[15]。彼らは、CPU とディスプレイが同クラスのシステムの主要なエネルギー消費要因であり、他のコンポーネントは集中的に使用された場合にのみ主たる消費電力要因となることを示した。Sagahyroon は、ハンドヘルド PC の解析を行っている[16]。彼は、ディスプレイサブシステムにおいて大きな消費電力があること、特にバックライトにおいて大きな電力の消費があることを示している。また彼は、CPU とそのクロック周波数が消費電力の重要な要因であることも示唆した。

次に、スマートフォンの省電力に関する研究として以下のようなものがある。Carroll らは、携帯電話の消費電力を詳

細に分析している[17][17]。彼らは、システム全体の電力消費と、主要なハードウェアコンポーネントごとの電力消費内訳の両方を測定している。そして、電力消費モデルを開発し、多くの使用パターンにおけるエネルギー使用量とバッテリー寿命を分析している。また、プロセッサの動的な電圧と周波数スケーリングによるエネルギーへの影響も分析している。Pathak らは eprof と呼ばれるスマートフォンアプリケーション向けに詳細なエネルギープロファイラを提案している[18][18]。彼らは、無料のアプリケーションのエネルギーの 65~75%が第三者の広告モジュールに費やされたという調査結果を示している。Zhuang らは、ロケーションベースのアプリケーションに焦点を当て、位置検出デバイスにおける集中的な位置検出処理が多くのバッテリーを消費していることを指摘している[19]。そして彼らは、適応型位置検出フレームワークを提案している。文献[20][20]では、CPU クロック周波数を調整する方法が提案されている。この方法では性能をフレームレートとして定義し、フレームレートがユーザがあらかじめ定めた最大値と最小値の間の値になるように CPU クロック周波数を制御している。これらの研究はスマートフォンの消費電力に関する詳細な議論を提供しているが、ネットワークデバイスの中断による電力消費の減少に関する議論は提供していない。

以下の研究では、デバイスの使用状況の違いによる消費電力の削減について述べている。Nakamura らは、コンピュータシステムのコンポーネントを積極的にパワーオフし、通常時は動作させないようにする方法を提案している[21][21]。彼らはこれをノーマル・オフ・コンピューティングと呼んでいる。Juang らは、野生動物の追跡のためにモバイルセンサーネットワークで無線 P2P ネットワーキング技術を適用することについての議論を發表し、デバイスの使用状況による電力の違いについての議論を行っている[22][22]。文献[9][10]の研究では、デバイスを頻繁にパワーオフモードにすることによりストレージデバイスの消費電力を削減する方法が提案されている。本稿は、文献[9][10]の考えに基づいている。これらの研究においては、デバイスのオンとオフによる消費電力の違いや BET について言及されているが、BET を推定する手法については考察されていない。

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 の推定手法の性能評価を行った。そして、Nexus5X を用いた BET 推定と、実アプリケーションモデルに基づく BET 推定において、高い正確度で BET 推定が可能であることを確認した。

今後は、セルラー通信での通信デバイス断続的無効化による省電力手法に着目し、BET 推定を行う予定である。

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_W3A320C100000/)
- [2] I. Zahid, M. A. Ali, R. Nassr, "Android Smartphone: Battery saving service," *2011 International Conference on Research and Innovation in Information Systems*, Kuala Lumpur, 2011, pp. 1-4. doi: 10.1109/ICRIIS.2011.6125677
- [3] S. Doki, T. Ogishi, S. Ano, "Mobile interface control scheme can extend battery life," *2015 International Conference on Information Networking (ICOIN)*, Cambodia, 2015, pp. 116-121. doi: 10.1109/ICOIN.2015.7057867
- [4] 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
- [5] 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)
- [6] 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)*
- [7] Shunsuke Yagai and Saneyasu Yamaguchi, "Reducing Storage Power Consumption with File Layout Optimization in Virtualized Environment," *2015 Third International Symposium on Computing and Networking (CANDAR)*, Sapporo, 2015, pp. 590-594. doi: 10.1109/CANDAR.2015.101
- [8] Shunsuke Yagai and Saneyasu Yamaguchi, "Energy Efficient Storage Management Cooperated with Data Intensive Applications in Virtua IMachines," *2014 IEEE 33rd International Symposium on Reliable Distributed Systems Workshops*, Nara, 2014, pp. 119-124. doi:10.1109/SRDSW.2014.24
- [9] Shunsuke Yagai, Masato Oguchi, Miyuki Nakano, Saneyasu Yamaguchi, "Power-effective File Layout based on Large Scale Data-intensive Application in Virtualized Environment," *IEICE TRANSACTIONS on Information and Systems*, Volume and Number: Vol.E100-D, No.12, pp.-, Dec. 2017. DOI: 10.1587/transinf.2017PAP0003
- [10] Norifumi Nishikawa, Miyuki Nakano, Masaru Kitsuregawa, "Application Sensitive Energy Management Framework for Storage Systems," *IEEE Transactions on Knowledge and Data Engineering*, Vol.PP, pp.1-12, DOI=10.1109/TKDE.2015.2416737, 2015
- [11] Nakada T., Nakamura H, "Normally-Off Computing," Springer, Tokyo, 2017. DOI: 10.1007/978-4-431-56505-5
- [12] 村上 翼, 栗原 駿, 福田 翔貴, 小口 正人, 山口 実靖, "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", *情報処理学会 コンシューマ・デバイス&システム研究会(CDS) 2017*
- [13] Tsubasa Murakami, Shoki Fukuda, Shun Kurihara, Masato Oguchi and Saneyasu Yamaguchi. "Saving Power Consumption of Smartphones in the Screen-off State with Disabling the Wi-Fi", *2018 International Conference On Consumer Electronics (ICCE 2018)*, 2018
- [14] Takeshi Kamiyama, Kenji Hisazumi, Hiroshi Inamura, Teppei Konishi, Ken Ohta and Akira Fukuda. "Smartphone Usage Analysis Based on Actual-Use Survey", *2016 ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering) 2016*
- [15] Aqeel Mahesri and Vibhore Vardhan, "Power consumption breakdown on a modern laptop," In *Proceedings of the 4th international conference on Power-Aware Computer Systems (PACS'04)*, Babak Falsafi and T. N. VijayKumar (Eds.). Springer-Verlag, Berlin, Heidelberg, 165-180, 2004. DOI=http://dx.doi.org/10.1007/11574859_12
- [16] A. Sagahyroon, "Power Consumption in Handheld Computers," *APCCAS 2006 - 2006 IEEE Asia Pacific Conference on Circuits and Systems*, Singapore, 2006, pp. 1721-1724. doi: 10.1109/APCCAS.2006.342129
- [17] Aaron Carroll, Gernot Heiser, "An analysis of power consumption in a smartphone" *USENIXATC'10 Proceedings of the 2010 USENIX conference on USENIX annual technical conference*, 21-21, 2010.
- [18] Abhinav Pathak, Y. Charlie Hu, and Ming Zhang, "Where is the energy spent inside my app?: fine grained energy accounting on smartphones with Eprof," In *Proceedings of the 7th ACM european conference on Computer Systems (EuroSys '12)*. ACM, New York, NY, USA, 29-42, 2012. DOI: <http://dx.doi.org/10.1145/2168836.2168841>
- [19] Zhenyun Zhuang, Kyu-Han Kim, Jatinder Pal Singh, "Improving energy efficiency of location sensing on smartphones" *MobiSys '10 Proceedings of the 8th international conference on Mobile systems, applications, and services*, 315-330, 2010.
- [20] K. Nagata, S. Yamaguchi and H. Ogawa, "A Power Saving Method with Consideration of Performance in Android Terminals," *2012 9th International Conference on Ubiquitous Intelligence and Computing and 9th International Conference on Autonomic and Trusted Computing*, Fukuoka, 2012, pp. 578-585. doi: 10.1109/UIC-ATC.2012.133
- [21] H. Nakamura, T. Nakada and S. Miwa, "Normally-off computing project: Challenges and opportunities," *2014 19th Asia and South Pacific Design Automation Conference (ASP-DAC)*, Singapore, 2014, pp. 1-5. doi: 10.1109/ASPDAC.2014.6742850
- [22] Philo Juang, Hidekazu Oki, Yong Wang, Margaret Martonosi, Li Shiuian Peh, and Daniel Rubenstein, "Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with ZebraNet," *SIGARCH Comput. Archit. News* 30, 5, 96-107, 2002. DOI: <http://dx.doi.org/10.1145/635506.605408>