Battery State-of-Health Estimation for Mobile Devices

Liang He
University of Michigan
Ann Arbor, MI
lianghe@umich.edu

Eugene Kim
University of Michigan
Ann Arbor, MI
kimsun@umich.edu

Kang G. Shin
University of Michigan
Ann Arbor, MI
kgshin@umich.edu

Guozhu Meng
Nanyang Technological University
Singapore
gzmeng@ntu.edu.sg

Tian He
University of Minnesota
Minneapolis, MN
tianhe@umn.edu

ABSTRACT

Insufficient support of electric current sensing on commodity mobile devices leads to inaccurate estimation of their battery's state-of-health (SoH), which, in turn, shuts them off unexpectedly and accelerates their battery fading. In this paper, we design `V-BASH`, a new battery SoH estimation method based only on their voltages and is compatible to commodity mobile devices. `V-BASH` is inspired by the physical phenomenon that the relaxing battery voltages correlate to battery SoH. Moreover, it is enabled on mobile devices with a common usage pattern of most users frequently taking a long time to charge their devices. The design of `V-BASH` is guided by 2,781 empirically collected relaxing voltage traces with 19 mobile device batteries. We evaluate `V-BASH` using both laboratory experiments and field tests on mobile devices, showing a <6% error in SoH estimation.

CCS CONCEPTS

•Computer systems organization →Embedded software;

KEYWORDS

Battery management, mobile device, state of health, relaxation

ACM Reference format:
Liang He, Eugene Kim, Kang G. Shin, Guozhu Meng, and Tian He. 2017. Battery State-of-Health Estimation for Mobile Devices. In *Proceedings of The 8th ACM/IEEE International Conference on Cyber-Physical Systems, Pittsburgh, PA USA, April 18-21, 2017 (ICCPS 2017),* 10 pages.
DOI: http://dx.doi.org/10.1145/3055004.3055018

1 INTRODUCTION

Batteries have been widely used to power mobile devices, such as phones, tablets, and smartwatches, whose capacity fades over usage and time [18, 19, 27, 40], shortening device operation time [11]. The state-of-health (SoH) of batteries is a metric quantifying their capacity fading, commonly defined as the ratio of batteries' full charge capacity to their originally rated levels [40, 45].

DOI: http://dx.doi.org/10.1145/3055004.3055018

However, battery SoH is missing on many commodity mobile devices. For example, Android only specifies battery health as `good` or `dead`, without any quantified information [3]. This is because most existing SoH estimation methods require complex battery parameters (e.g., impedance [22] and resistance [31]) and specific operating conditions (e.g., small current of 0.05C to fully charge and discharge the batteries [39, 42]) to be applied, preventing their implementation on mobile devices due to hardware limitation and dynamic usage pattern. Moreover, even Coulomb counting — the most widely deployed SoH estimation method in practice via current integration [32, 42] — is not supported well on mobile devices. This is because (i) not all power management ICs (PMICs) of mobile devices support electric *current sensing* [41], thus making Coulomb counting infeasible; (ii) the PMIC-provided current information, even when available, suffers from poor accuracy and lacks real-time capability.

The absence of battery SoH information degrades the estimation accuracy of their real-time state-of-charge (SoCs) [12, 30] and thus devices' remaining operation time [42], shutting off the device unexpectedly [11, 34]. For example, users have reported their devices shut off when the devices are still shown to have 10–30% remaining power [5, 7, 9]. Apple has announced a free battery-replacement program for iPhone 6S in Nov. 2016 due to such unexpected device shutoffs [2]. Moreover, inaccurate SoC easily leads to battery over-charging/discharging [45], accelerating SoH degradation and causing even more inaccurate SoC estimation — forming a positive feedback loop between SoH degradation and SoC error. Last but not the least, the absence of battery SoH information confuses users about whether the shortened device operation is due to system updates and APP installation (e.g., Android 6.0 Marshmallow is reported to reduce device operation when first launched [17]), or because of hardware module failure [4], or a result of battery fading.

In this paper, we design `V-BASH`, a new battery SoH estimation method based only on their voltages, and is thus compatible to commodity mobile devices. `V-BASH` is inspired by a physical battery characteristic and enabled on mobile devices by a common usage pattern. Specifically, `V-BASH` is based on an empirical observation that the relaxing voltages of batteries — a time series of voltages when resting the batteries after either charging or discharging — correlate to their SoH. We construct a mathematical model to describe such relaxing voltages, based on which a voltage-based SoH fingerprint is proposed. `V-BASH` is guided by 2,781 relaxing voltage traces collected during 26 cycling tests with 19 batteries, each consisting of 48–298 discharging/charging/resting cycles. In

Table 1: Needed battery information for SoH estimation.

Ref.	Vol.	Curr.	OCV	SoC	Resis.	Imped.
[30]		✓	✓	✓		
[31]	✓	✓	✓		✓	
[39]			✓			
[44]		✓				
[22]						✓
[16]			✓	✓		
[12, 29, 45]	✓	✓		✓	✓	
[11, 21, 33, 35]	✓	✓				
[14, 20, 25]	✓					

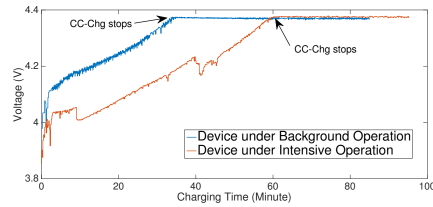


Figure 1: Voltage curve during CC-Chg is affected by device operation.

total, about 1.1G data samples are collected during the accumulated 22-month tests.

Collecting the relaxing voltages on mobile devices, however, is challenging because their continuous operation and dynamic usage patterns prevent their batteries to rest. V-BASH mitigates this using the fact that most users charge their devices for a long time and frequently (e.g., over-night charging) [23, 38], during which the charger is connected even after the device is fully charged. This, in turn, relaxes the battery because of charger's separate power paths — commodity chargers use two power flows to charge the battery and power the device, respectively, thus resting the battery once fully charged. We collect 976 charging cases from real-life device usage of 7 users over 1–3 months, showing 34% of them last over 6 hours and are long enough to allow the battery to relax after fully charged. These collected relaxing voltage traces are then used to extract the SoH fingerprints, based on which the SoH is estimated. We evaluate V-BASH using both laboratory experiments and field-tests on multiple mobile devices, showing a less than 6% error in SoH estimation.

This paper makes the following major contributions:

- Revealing the correlation between relaxing battery voltages and their SoH (Sec. 4);
- Designing V-BASH, an SoH estimation method for mobile devices based only on their voltages (Sec. 5);
- Evaluating V-BASH using both laboratory experiments and field tests on mobile devices, showing a <6% error in SoH estimation (Secs. 6 and 7);

2 RELATED WORK

SoH estimation is the core of battery management. Researchers have been using various techniques, such as Kalman filter and its variations [45], support vector machine [12], fuzzy logic systems [44], non-linear observers [30], Coulomb counting [29], ultrasonic inspection [36], etc., to estimate battery SoH based on a wide range of mathematic/circuit/empirical models [12, 25, 45]. These existing methods require battery voltage, current, SoC, open-circuit-voltage (OCV), internal resistance and even impedance, as summarized in Table 1.

However, none of these SoH estimation methods can be applied to mobile devices due to limited sensing hardware support and dynamic operating conditions.

Mobile devices offer limited hardware sensing support, making some of the needed battery information unavailable. For example, the battery impedance needed in [22] requires a specialized impedance meter to collect, costing as much as $5,000 apiece. Actually, even the relatively easy-to-measure electric current is not always available on mobile devices [41], and suffer from low accuracy and lacks timeliness when available. We will elaborate more on the insufficient current sensing on mobile devices in Sec. 3.

Also, existing SoH estimation methods require battery information measured under specific conditions. However, the operation of mobile devices is dynamic due to human interactions and background activities, making it difficult to control battery condition. For example, information such as OCV and SOC is measurable only when the battery has been charged/discharged with small current (e.g., less than 0.05C) for a long period (e.g., 30 minutes) [39, 42], which does not always hold due to devices' dynamic usage pattern and thus suffers from low accuracy, e.g., a SoC estimation error of ±25% is specified in the datasheet of Qualcomm's PM8916. Similarly, the voltage curve during CC-Chg is used in [25] to estimate SoH, which is not reliable on mobile devices due to their dynamic operation. Fig. 1 plots the voltage curves during two consecutive charging of a Galaxy S6 Edge phone — the phone is left idle during the first charging and operates intensively during the second, showing clear dependency of voltage curve on device operation. The authors of [14] estimate SoH based on the voltage after resting a battery for 30 minutes, which is not feasible on mobile devices because of the trickle charging, as we will explain in Sec. 7. Moreover, the proposed method therein requires beginning-of-life battery information for calibration, which is usually not available on mobile devices. The SoH estimation in [44] is not applicable on mobile devices either as it requires to rest battery after discharging it to a fixed SoC.

In summary, existing SoH estimation methods are not applicable on mobile devices because (i) the needed battery information is usually unavailable, and (ii) the needed measuring conditions may not be satisfied. To overcome this deficiency, we propose V-BASH which estimates SoC based only on voltage information and is enabled on mobile devices with a common usage pattern.

3 MOTIVATION

Discussed below is our motivation behind V-BASH.

3.1 Battery State-of-Health

Batteries fade over time and usage, shortening device operation. The state-of-health (SoH) of batteries quantifies their fading, commonly defined as the ratio of batteries' full charge capacity to their originally rated levels, i.e.,

$$SoH = C_{\text{full}}/C_{\text{rated}} \times 100\%. \quad (1)$$

The full charge capacity is the foundation of SoH estimation, which is traditionally estimated via Coulomb counting [42, 43],

Table 2: Availability of current information on mobile devices and observations when implementing Coulomb counting.

Devices	Android V.	PMIC	Curr. Sens.	Observations when Implementing Coulomb Counting
Nexus 6P	6.0	PM8994	✓	concluding a 3, 769mAh full charge capacity; reasonable for the equipped 3, 450mAh battery
Nexus 6	5.0	PMA8084	✓	low updating rate of about once per 20 seconds
Nexus 5X	6.0	PM8994	✓	concluding a 6, 150mAh full charge capacity, which is 2.8x of the equipped 2, 700mAh battery
Nexus 5	5.0.1	PM8941	✓	current information is not always available
Nexus S	4.2.1	MAX8998	X	N.A.
Galaxy S6 Edge	5.0.2	MAX77843	✓	current information is not always available, and low updating rate of about twice per minute when available
Galaxy S5	6.0.1	QFE1100	✓	provided current fixes at 450mA when discharging
Galaxy S4	4.4.2	S2MPS11	X	N.A.
Galaxy W	4.1.2	MAX17043	X	N.A.
Note 3	5.0	PM8941	✓	provided current fixes at 450mA when discharging
Note 2	4.4.2	PM8941	✓	provided current fixes at 0mA when discharging
Note 8.0	4.4.2	MAX77686	✓	provided current fixes at 0mA when discharging
Xperia Z	4.4.4	PM8921	✓	concluding a 2, 287mAh full charge capacity; reasonable for the equipped 2, 330mAh battery
Zenfone Selfie	5.0.2	PS63020	✓	low updating rate of about once per 20 seconds
P8	6.0	Hi6561	X	N.A.
HONOR 7i	5.1.1	PM8916	✓	low updating rate of about once per minute
Mi 2S	5.0.2	PM8018	✓	low updating rate of about once per 10 seconds

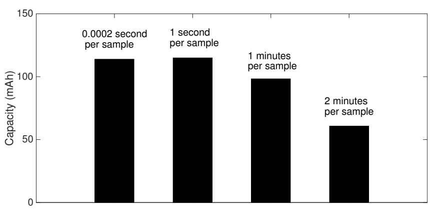


Figure 2: Insufficient sampling rate causes up to 47% errors in Coulomb counting.

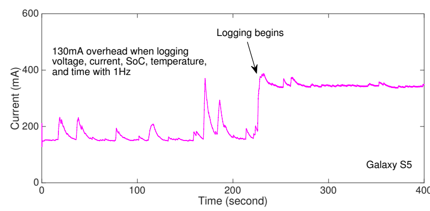


Figure 3: Read/write of battery information incurs non-negligible amounts of energy consumption.

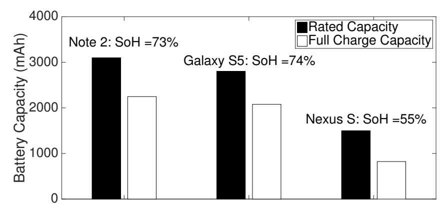


Figure 4: Android tells all these batteries are in *good* health even if up to 45% fading is observed.

i.e., integrating the current when discharging/charging the battery between two SoCs (state-of-charge) to calculate the discharged/charged capacity as

$$\Delta C = \int_{t(SoC_1)}^{t(SoC_2)} i(t)dt,$$

where $i(t)$ is the current at time t, and estimating the full charge capacity as

$$C_{\text{full}} = \Delta C / |SoC_1 - SoC_2|.$$

3.2 Deficiency in Coulomb Counting Support

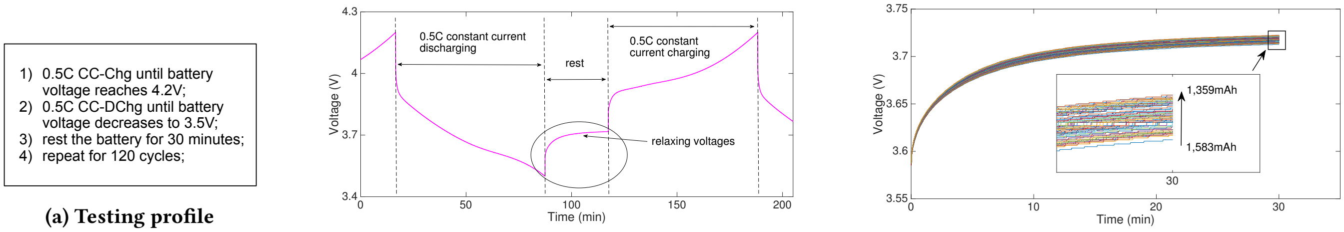
Commodity mobile devices do not support Coulomb counting well, thus making it difficult to estimate their battery SoH. First of all, not all the power management ICs (PMICs) of mobile devices support current sensing. Table 2 lists the PMICs of several devices, showing their lack of current sensing support and thus making it infeasible to use Coulomb counting.

Moreover, the PMIC-provided current information, even when available, is not accurate. Commodity mobile devices estimate their current with a series-connected resistor r, measure the voltage v across the resistor and estimate the current as $i = v/r$ [6].[1] The resistor incurs the heating overhead (i.e., $i^2 r$), which must be low and thus require a small r. For example, Maxim requires <0.5mW heating overhead, indicating $r < 50m\Omega$ for devices operating with 100mA current. Such a small resistance, however, reduces the voltage across it and thus degrades current sensing accuracy. Also, resistance is temperature-dependent and the temperature of device batteries varies a lot, easily causing 5–10% resistance variations [24].

Last but not the least, the current information may lack of timeliness. For example, Android's `BatteryManager` supports only two sampling rates of 1 and 10 minutes per sample, which are too slow for Coulomb counting, especially when devices' currents are known to be highly dynamic, i.e., varying from tens to thousands of milliamps in milliseconds. To show the impact of sampling rate on Coulomb counting, we collected a 12-minute current trace using the Monsoon power monitor running at 5, 000Hz, during which a 113.9mAh capacity is discharged. Then, based on this trace, we implement Coulomb counting by emulating different sampling rates of 1Hz, 1/60Hz, and 1/120Hz, achieving a discharged capacity of 114.9mAh, 98.3mAh, and 60.6mAh, respectively — an insufficient sampling frequency causes up to 47% counting error (Fig. 2). An alternative facilitating fine-grained current sensing is to directly read the system file containing the current information at a customized frequency. However, the high-frequency read/write of files incurs non-negligible power consumption overhead, thus preventing its implementation on mobile devices, as shown in Fig. 3 for a Galaxy S5 phone.

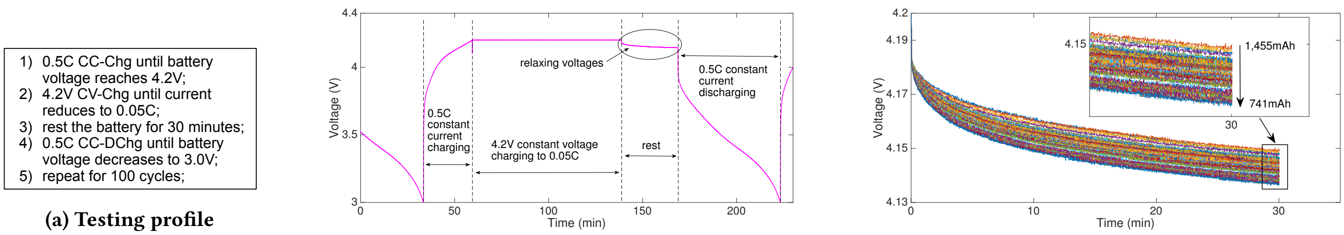
These together yield unreliable current information on mobile devices. The last column of Table 2 summarizes the observations when implementing Coulomb counting based on the PMIC-provided current information on these 17 mobile devices, showing the unreliability of current readings in both value and timeliness. Note the listed update frequency is for the system file containing the current information, i.e., the highest achievable sampling rate supported by the PMIC. The unreliable current information on Galaxy S3, Nexus

[1] An alternative current sensing approach is via Hall sensors, which requires a large PCB real estate and suffers from slow response to current change. As a result, it is not suitable for mobile devices with strict sizing-requirements and dynamic currents.



(a) Testing profile **(b) Exemplary voltage traces during one cycle** **(c) Relaxing voltages and capacity delivery**

Figure 5: After-discharging relaxing voltages indicate battery SoH: relaxing voltages become higher over the measurement while battery SoH declines.



(a) Testing profile **(b) Exemplary voltage traces during one cycle** **(c) Relaxing voltages and capacity delivery**

Figure 6: After-charging relaxing voltages indicate battery SoH: relaxing voltages lower over the measurement while battery SoH declines.

4, and Nexus 7 is also reported by Ampere, a current sensing APP with millions of downloads [1].

3.3 Lack of Quantified Battery SoH

The deficiency of Coulomb counting support on mobile devices leads to limited health information on their batteries, e.g., Android only specifies battery health as good or dead. We use 3 Android phones whose batteries are all tagged to be good to demonstrate this limitation. We measure the batteries' full charge capacity by fully charging and then discharging them, based on which their SoH is estimated. The measurements are made with the NEWARE battery tester, with which the charging/discharging processes can be controlled with error less than 0.5% and logged at up to 10Hz. Fig. 4 plots the thus-obtained battery SoH, showing up to 45% capacity fading although all were specified to be good.

Mobile devices' deficiency in supporting Coulomb counting and their limited SoH information are the main motivations for us to explore the possibility of current-free SoH estimation, developing V-BASH as presented next.

4 DESIGN PRINCIPLE

V-BASH is inspired by a key physical battery characteristic (Sec. 4.1) and enabled by a common usage pattern of mobile devices (Sec. 4.2).

4.1 Relaxing Voltages Indicate SoH

V-BASH is based on the fact that the relaxing battery voltages indicate their SoH, as shown in the following two measurements.

In the first measurement, we charge and discharge a Li-ion battery for 120 cycles according to the profile specified in Fig. 5(a),

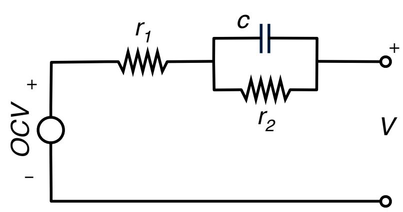


Figure 7: Thevenin's battery circuit model, consisting of a series resistor and a parallel resistor-capacitor.

in which a 30-minute rest period is inserted *after each discharging*. Fig. 5(b) plots the battery voltage during one of such cycles, and highlights the relaxing voltages during the after-discharging rest — the battery voltage increases instantly to a certain degree when resting and then rises gradually until it converges. Fig. 5(c) compares the thus-collected 120 relaxing voltage traces, where the curves get higher as the measurement continues. Also, the battery weakens during the measurement, observed as its reduced capacity delivery in each discharging, i.e., from 1,583mAh in the first cycle to 1,359mAh in the last one. All of these together reveal a monotonic relationship between the battery's relaxing voltages and its capacity delivery (or SoH).

In the second measurement, we charge and discharge another battery for 100 cycles according to Fig. 6(a). Unlike the first measurement, a 30-minute rest period is inserted *after each battery charging*. Fig. 6(b) plots the battery voltage during one of such cycles, and Fig. 6(c) shows the collected 100 relaxing voltage traces, each of which drops instantly to a certain degree upon resting, and then decreases further until it converges. Again, certain monotonicity

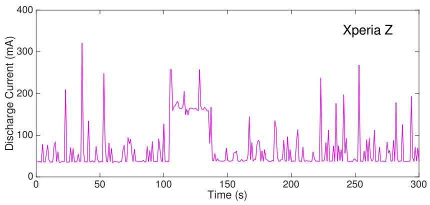


Figure 8: Battery is continuously discharged even for idle devices.

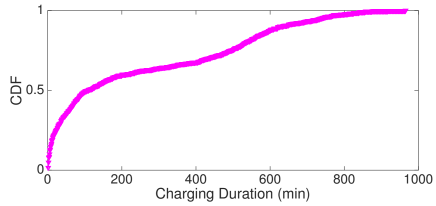


Figure 9: 34% **of the** 976 **collected charging cases last over** 6 **hours.**

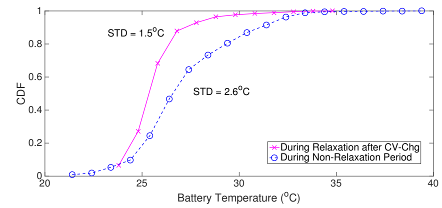


Figure 10: More stable battery temperature during after-charging relaxation.

between the relaxing voltages and SoH is observed — the relaxing voltage lowers over the measurements while the capacity delivery degrades.

This relationship between battery's relaxing voltages and its SoH can be explained by Thevenin's battery circuit model [25, 26, 28], which describes the battery with an ideal voltage source, a series resistor r_1, and a parallel capacitor c, as shown in Fig. 7, and both r_1 and c increase as battery fades [13, 29, 30]. The battery voltage is modeled as

$$v(t) = OCV(t) - i(t)r_1 - v_c(t),$$

where $OCV(t)$ is the battery's open-circuit-voltage at time t. Resting the battery at time t to start its relaxation, the battery OCV will not change at time $t + 1$ as it is not charged/discharged. However, an instantaneous voltage response of $i(t)r_1$ occurs because the voltage across r_1 vanishes instantly, which is followed by a gradual change due to the parallel capacitor (i.e., $v_c(t)$ changes gradually after resting). These lead to an instant change in $v(t)$ upon resting and then followed by a graduate change until converged, as observed in Figs. 5 and 6. This way, we expect larger voltage change after resting the battery as its SoH degrades (and thus r and c become larger) — the relaxing voltage curve rises/lowers more as the measurement continues.

4.2 Long-Time Charging Allows Relaxation

Battery's relaxing voltages, albeit reflecting its SoH, are not always obtainable on mobile devices for the following reasons. First, relaxing voltages require idle batteries, e.g., the 30-minute resting period in the above measurements.[2] Mobile devices, however, have a continuous and dynamic discharge current of 40–300mA due to device monitoring and background activities even left idle, as illustrated in Fig. 8 with a Xperia Z phone. Also, battery voltage is temperature-dependent (e.g., Trojan Battery uses a 0.28V voltage compensation for every 10^oF change in battery temperature [10]), requiring a stable thermal environment to collect the relaxing voltages. This is challenging because of the notorious heating issue of mobile devices [37].

We mitigate these difficulties with the fact that users often charge their devices for a long time — the charging duration is so long (e.g., during over-night charging) that the charger is kept connected even after the battery is fully charged [23, 38]. Fig. 9 plots the charging time (i.e., the time since the charger is connected to its disconnection) distribution of 976 charging cases collected from 7

Figure 11: Laboratory settings for cycling tests.

users over 1–3 months,[3] showing 34% of them lasted over 6 hours and are long enough to keep the charger connected after fully charging the device. This, in turn, relaxes the battery and thus facilitates collection of its relaxing voltages.

First, such long-time charging of devices rests their batteries. Commodity chargers use separate power paths to charge the battery and power the device [15]. This allows the battery to rest after being fully charged if the charger stays connected to the device. Also, the long-time charging offers a relatively stable thermal environment to collect its relaxing voltages. This is because most mobile devices adopt the 2-phase Constant-Current Constant-Voltage (CCCV) charging method, described by triple $< I_{cc}, V_{max}, I_{cutoff} >_{cccv}$ — charging the battery with constant current I_{cc} until its voltage reaches V_{max} (i.e., CC-Chg), and then charging it further with constant voltage until the charging current decreases to I_{cutoff} (i.e., CV-Chg). The completion of CV-Chg concludes the device's full charging and starts the battery relaxation if the charger is kept connected. As CV-Chg is long and has a small charging rate, it heats the battery little and allows for its thermal equilibration, making a stable thermal environment for the relaxing period afterwards. To verify this, we monitor the battery temperature of a Galaxy S6 Edge phone during 8-day real-life usage. Fig. 10 compares the temperature distributions during the relaxing and non-relaxing periods, showing clearly reduced thermal diversity, e.g., the temperature range is narrowed from 21.4–39.7^{o}C (and with a STD of 2.6^oC) for non-relaxing periods to 25.8–35.4^{o}C (and with a STD of 1.5^oC) when relaxing.

5 VOLTAGE-BASED SOH ESTIMATION

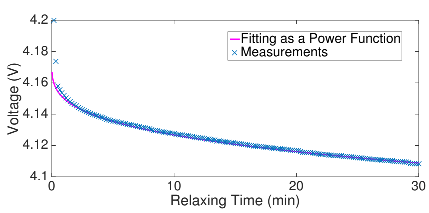
This section details the design of V-BASH, guided by a set of empirically collected relaxing voltage traces.

[2] Relaxation occurs so long as battery current decreases and keeps low, but non-zero current affects the relaxing voltage curves, thus introducing noises in SoH estimation.

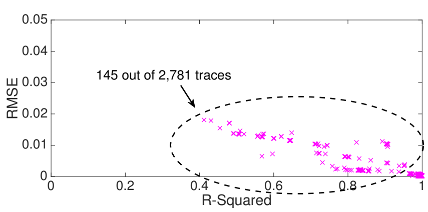
[3] One of the user-traces was collected from our data-collection campaign and the other six traces were obtained from the device analyzer dataset of Cambridge University [38].

Table 3: V-BASH is guided by 2, 781 empirically collected relaxing voltage traces via 26 cycling tests with 19 phone batteries.

Cycle Test	Battery	Rated Capacity	Initial SoH	# of Cycles	Per-Cycle Profile
#1	Nexus 6P (1)	3,450mAh	79%	48	$< 0.5C, 4.35V, 0.05C >_{\text{cccv}}$; 30min rest; 0.5C CC-DChg to $V_{\text{cutoff}} = 3.3$V;
#2	Nexus 6P (2)	3,450mAh	84%	48	$< 0.5C, 4.35V, 0.05C >_{\text{cccv}}$; 30min rest; 0.5C CC-DChg to $V_{\text{cutoff}} = 3.3$V;
#3	Nexus 5X (1)	2,700mAh	84%	217	$< 0.5C, 4.35V, 0.05C >_{\text{cccv}}$; 30min rest; 0.5C CC-DChg to $V_{\text{cutoff}} = 3.3$V;
#4	Nexus 5X (2)	2,700mAh	71%	136	$< 0.5C, 4.35V, 0.05C >_{\text{cccv}}$; 30min rest; 0.5C CC-DChg to $V_{\text{cutoff}} = 3.3$V;
#5	Nexus S (1)	1,500mAh	53%	48	$< 0.5C, 4.20V, 0.05C >_{\text{cccv}}$; 30min rest; 0.5C CC-DChg to $V_{\text{cutoff}} = 3.2$V;
#6	Nexus S (1)	1,500mAh	52%	98	$< 0.5C, 4.20V, 0.05C >_{\text{cccv}}$; 30min rest; 0.5C CC-DChg to $V_{\text{cutoff}} = 3.2$V;
#7	Xperia Z5 (1)	2,900mAh	64%	98	$< 0.5C, 4.20V, 0.05C >_{\text{cccv}}$; 30min rest; 0.5C CC-DChg to $V_{\text{cutoff}} = 3.2$V;
#8	Xperia Z5 (2)	2,900mAh	62%	99	$< 0.5C, 4.20V, 0.05C >_{\text{cccv}}$; 30min rest; 0.5C CC-DChg to $V_{\text{cutoff}} = 3.2$V;
#9	Galaxy S5 (1)	2,800mAh	86%	49	$< 0.5C, 4.35V, 0.05C >_{\text{cccv}}$; 30min rest; 0.5C CC-DChg to $V_{\text{cutoff}} = 3.3$V;
#10	Galaxy S5 (1)	2,800mAh	84%	49	$< 0.5C, 4.35V, 0.05C >_{\text{cccv}}$; 30min rest; 0.5C CC-DChg to $V_{\text{cutoff}} = 3.3$V;
#11	Galaxy S4 (1)	2,600mAh	92%	53	$< 0.5C, 4.20V, 0.05C >_{\text{cccv}}$; 30min rest; 0.5C CC-DChg to $V_{\text{cutoff}} = 3.0$V;
#12	Galaxy S4 (1)	2,600mAh	91%	48	$< 0.5C, 4.20V, 0.05C >_{\text{cccv}}$; 30min rest; 0.5C CC-DChg to $V_{\text{cutoff}} = 3.0$V;
#13	Galaxy S4 (2)	2,600mAh	93%	53	$< 0.5C, 4.20V, 0.05C >_{\text{cccv}}$; 30min rest; 0.5C CC-DChg to $V_{\text{cutoff}} = 3.0$V;
#14	Galaxy S4 (2)	2,600mAh	91%	48	$< 0.5C, 4.20V, 0.05C >_{\text{cccv}}$; 30min rest; 0.5C CC-DChg to $V_{\text{cutoff}} = 3.0$V;
#15	Galaxy S4 (3)	2,600mAh	92%	52	$< 0.5C, 4.20V, 0.05C >_{\text{cccv}}$; 30min rest; 0.5C CC-DChg to $V_{\text{cutoff}} = 3.0$V;
#16	Galaxy S4 (3)	2,600mAh	92%	48	$< 0.5C, 4.20V, 0.05C >_{\text{cccv}}$; 30min rest; 0.5C CC-DChg to $V_{\text{cutoff}} = 3.0$V;
#17	Galaxy S4 (4)	2,600mAh	90%	54	$< 0.5C, 4.20V, 0.05C >_{\text{cccv}}$; 30min rest; 0.5C CC-DChg to $V_{\text{cutoff}} = 3.0$V;
#18	Galaxy S4 (4)	2,600mAh	88%	48	$< 0.5C, 4.20V, 0.05C >_{\text{cccv}}$; 30min rest; 0.5C CC-DChg to $V_{\text{cutoff}} = 3.0$V;
#19	Galaxy S3 (1)	2,200mAh	92%	298	$< 0.5C, 4.20V, 0.05C >_{\text{cccv}}$; 30min rest; 0.5C CC-DChg to $V_{\text{cutoff}} = 3.3$V;
#20	Galaxy S3 (2)	2,200mAh	92%	298	$< 0.5C, 4.20V, 0.05C >_{\text{cccv}}$; 30min rest; 0.5C CC-DChg to $V_{\text{cutoff}} = 3.3$V;
#21	Galaxy S3 (3)	2,200mAh	93%	298	$< 0.5C, 4.20V, 0.05C >_{\text{cccv}}$; 30min rest; 0.5C CC-DChg to $V_{\text{cutoff}} = 3.3$V;
#22	Galaxy S3 (4)	2,200mAh	93%	298	$< 0.5C, 4.20V, 0.05C >_{\text{cccv}}$; 30min rest; 0.5C CC-DChg to $V_{\text{cutoff}} = 3.3$V;
#23	Note 2 (1)	3,100mAh	31%	48	$< 0.5C, 4.20V, 0.05C >_{\text{cccv}}$; 30min rest; 0.5C CC-DChg to $V_{\text{cutoff}} = 3.2$V;
#24	Note 2 (1)	3,100mAh	27%	48	$< 0.5C, 4.20V, 0.05C >_{\text{cccv}}$; 30min rest; 0.5C CC-DChg to $V_{\text{cutoff}} = 3.2$V;
#25	iPhone 6 Plus (1)	2,900mAh	79%	50	$< 0.5C, 4.35V, 0.05C >_{\text{cccv}}$; 30min rest; 0.5C CC-DChg to $V_{\text{cutoff}} = 3.3$V;
#26	iPhone 6 Plus (2)	2,900mAh	62%	50	$< 0.5C, 4.35V, 0.05C >_{\text{cccv}}$; 30min rest; 0.5C CC-DChg to $V_{\text{cutoff}} = 3.3$V;

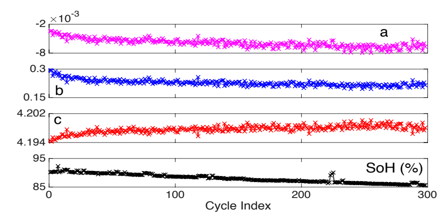


(a) Fitting as $v(t) = a \cdot t^b + c$

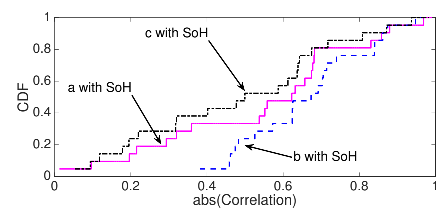


(b) Goodness-of-fit

Figure 12: Relaxing voltages conform to a 2-term power function $v(t) = a \cdot t^b + c$ $(t \geq 0)$: (a) power fit of a particular relaxing voltage trace; (b) goodness-of-fit when fitting the 2, 781 traces.

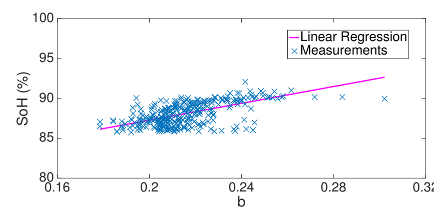


(a) Series of a, b, c, and SoH over test

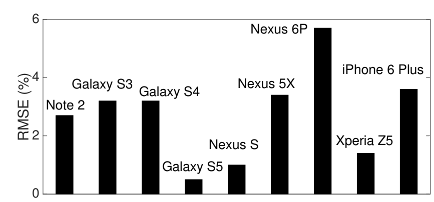


(b) b has stronger correlation with SoH

Figure 13: Describing relaxing battery voltages as $v(t) = at^b + c$, the power factor b is a better SoH fingerprint because of their stronger and more reliable correlation.



(a) b is linear to SoH for a given battery



(b) Regression error on same-model batteries

Figure 14: b is linear to battery SoH: (a) the linear regression based on Test #19 in Table 3 as an example; (b) errors after applying linear regression to same-model batteries.

5.1 Trace Collection

We empirically identify the relationship between relaxing battery voltages and their SoH. Specifically, we conduct 26 cycling tests with 19 batteries used to power devices such as Nexus 6P, Xperia Z5, iPhone 6 Plus etc, each consisting of 48–298 discharging/charging/resting cycles. Battery information such as voltage and current are logged at 1Hz, and about 1.1G data points are collected in total. Fig. 11 shows our laboratory settings for these tests. This way, a total number of 2, 781 relaxing voltage traces are collected, each lasting 30 minutes. The discharged capacity during each cycle also allows to collect the SoH ground truth based on Eq. (1). Table 3 summarizes the details of these cycling tests. The settings of $<0.5C, 4.2V, 0.05C>_{\text{cccv}}$ and $V_{\text{cutoff}} = 3.0$V for discharging are commonly used to specify battery properties in industry data sheets and in the literature of battery testing [25], and $V_{\max} = 4.35$V and a V_{cutoff} of 3.2–3.3V capture more device characteristics: mobile devices are normally charged to a maximum voltage of 4.3–4.4V and shut off when their battery voltage reduces to 3.2–3.3V.

5.2 Relaxing Voltages Fit as Power Function

Directly inferring battery SoH with the time series of relaxing voltages would be computationally expensive (and unnecessary as we will see), demanding a simple and accurate relaxing voltage descriptor. Unlike the traditional wisdom that the relaxing voltages change exponentially [20, 29], our examination of the collected relaxing traces reveals their conformance to a 2-term power function:

$$v(t) = a \cdot t^{b} + c \;(t \geq 0), \tag{2}$$

where t is the time since the relaxation began. This power fitting is visually illustrated in Fig. 12(a) by fitting one particular relaxing voltage trace to such a power function.

We apply the 2-term power fitting to the collected 2,781 traces to statistically verify the accuracy. Fig. 12(b) summarizes the goodness-of-fit in terms of root-mean-square error (RMSE) and R-Squared, where each point represents the goodness-of-fit when fitting a particular relaxing trace — with RMSE as y-value and R-Squared as x-value. As an RMSE close to 0 and an R-Square approaching 1 indicates accurate fitting, the fact that most fitting results are clustered at the right-bottom corner of Fig. 12(b) verifies that the relaxing voltage traces can be accurately described by a 2-term power function. Also, although the goodness-of-fit of a few traces have relatively large RMSE (i.e., between 0.01 to 0.02) and small R-Squared (i.e., between 0.4 to 0.6), counting these outliers reveals that 145 samples have RMSE larger than 0.0018 and R-Squared smaller than 0.95. In other words, about $1 - 145/2,781 \approx 95\%$ of the collected traces fit as power function with RMSE less than 0.0018 and R-Square larger than 0.95, showing excellent fitting accuracy.

Guided by these statistics, V-BASH uses a 2-term power function (or a, b, and c in Eq. (2) more specifically) to describe the relaxing battery voltages.

5.3 Power Factor b as SoH Fingerprint

The next step is how to map the relaxing voltages to battery SoH — i.e., defining a voltage-based SoH fingerprint.

V-BASH uses the power factor b as the SoH fingerprint, among other choices such as a, c or any combinations thereof. Mathematically, for power functions in the form of Eq. (2), b is the most descriptive parameter among a, b, and c, which determines the function's overall shape and behavior, such as the growth/decay rate [8]. Also, using b as the SoH fingerprint is supported further by its stronger and more reliable correlation with battery SoH. For each cycling test in Table 3, we obtain (i) a series of relaxing voltage traces as shown in Fig. 12(a), and thus a series of a, b, and c after applying the 2-term power fit to them, and (ii) a series of capacity delivery during each full discharge and thus SoH. Fig. 13(a) plots an example of these series during one 298-cycle test (Test #19 in Table 3). We can then calculate the correlations of a, b, and c with the SoH based on the series for each cycling test, whose distributions are shown in Fig. 13(b). Note that we use the absolute value of correlations in Fig. 13(b) because c is negatively correlated to SoH, as observed in Fig. 13(a). Clearly, b's correlation with SoH is larger (i.e., with larger mean value) and more reliable (i.e., with a larger minimum value) than a and c.

Next, we show how b is mapped to battery SoH, again, based on the collected traces. Fig. 14(a) plots b and battery SoH during each cycle of Test #19 in Table 3, indicating a linear relationship. As a further verification, we perform linear regression on the bs and the SoH during the cycling tests of same-model batteries, whose regression errors are summarized in Fig. 14(b) — the maximum RMSE of the linear regression is 5.7%, with an average of 2.7%. This shows that (i) a clear linearity exists between bs and SoH of a given battery, i.e.,

$$SoH = l_1 \cdot b + l_2 \tag{3}$$

for certain linear coefficients l_1 and l_2; (ii) second and more importantly, the linearity between b and SoH is similar for batteries of the same model, allowing V-BASH to train the linear model with a battery and estimate SoH for other batteries of the same model.

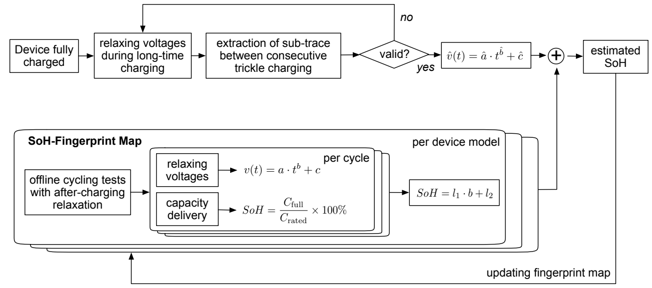


Figure 15: V-BASH overview: the device collects its relaxing voltages, which are compared with the offline constructed fingerprint map for SoH estimation. Thus-estimated SoH is then used to update the fingerprint map.

5.4 V-BASH Summary

Guided by the above statistic analyses, V-BASH estimates battery SoH according to Eq. (3) with offline learned coefficients l_1 and l_2. Fig. 15 provides an overview of V-BASH (the sub-trace extraction and validation will be explained in Sec. 7). Specifically, V-BASH starts by collecting relaxing battery voltages for various device models, deriving their corresponding bs, and constructing a per-device-model SoH fingerprint map accordingly. At the user side, the relaxing voltages of users' device battery are collected during a long-term charging, and the corresponding $\hat{b}$s are derived and compared with the fingerprint map for SoH estimation. The thus-estimated SoH and the corresponding relaxing voltages are then used to update the fingerprint map.

6 LABORATORY EXPERIMENTS

We first evaluate V-BASH based on our laboratory measurements, some of which has been summarized in Table 3. Specifically, we use the traces collected during one test to train V-BASH (i.e., construct the SoH fingerprint map based on bs derived from the traces), then use the traces during another test to validate its accuracy in SoH estimation (i.e., estimate the SoHs with V-BASH and compare them with the measured ground truth during each discharging of the test). For comparison, we also implement a baseline method, V-Drop, that uses the total voltage drop during the 30-minute relaxation as SoH fingerprint, which is an enhanced version of [14] as no beginning-of-life battery information is needed. These evaluations consist

Table 4: Additional cycling tests for the same-battery cross-profile validation.

Cycle Test	Battery	Rated Capacity	Initial SoH	Number of Cycles	Per-Cycle Profile
#27	Galaxy S4 (4)	2,600mAh	86%	48	$< 0.25C, 4.20V, 0.05C >_{cccv}$; 30min rest; 0.5C CC-DChg to 3.0V;
#28	Xperia Z5 (2)	2,900mAh	66%	48	$< 0.25C, 4.35V, 0.05C >_{cccv}$; 30min rest; 0.5C CC-DChg to 3.3V;
#29	Galaxy S4 (4)	2,600mAh	85%	48	$< 0.5C, 4.20V, 0.05C >_{cccv}$; 30min rest; 0.25C CC-DChg to 3.0V;
#30	Xperia Z5 (3)	2,900mAh	66%	48	$< 0.5C, 4.35V, 0.05C >_{cccv}$; 30min rest; 0.25C CC-DChg to 3.3V;

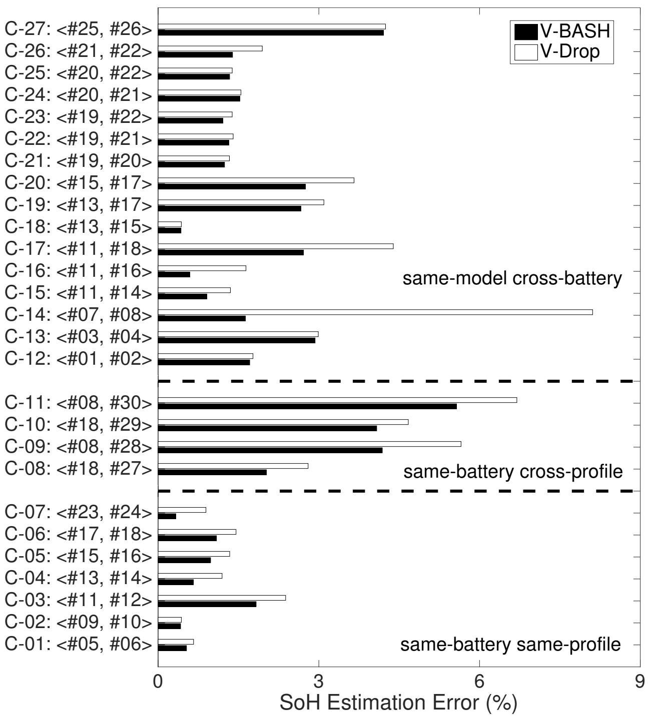


Figure 16: Evaluation of V-BASH with laboratory experiments: (bottom) V-BASH is trained and validated based on traces collected with the same battery during different tests with identical profiles; (middle) V-BASH is trained and validated based on traces collected with the same battery during tests with different profiles; (top) V-BASH is trained and validated based on traces collected with different same-model batteries.

of three parts: same-battery same-profile validation, same-battery cross-profile validation, and same-model cross-battery validation.

6.1 Same-Battery Same-Profile Validation

In this set of evaluations, the training and validation traces are collected with the same battery during different cycling tests with identical profiles. This way, 7 cases with different combinations of training/validation traces are selected from Table 3, as summarized in the bottom of Fig. 16, where the notation $< \#i, \#j >$ means using the traces collected in Test $\#i$ in Table 3 for training and those in Test $\#j$ for validation. As the battery SoHs differ between the training and validation traces due to degradation, these results verify whether the fingerprint map trained within a certain SoH range can be used to estimate SoHs out of its training range. V-BASH outperforms V-Drop in all the 7 cases explored, with a maximum error of 1.83% (Case #6) and mean error of 0.83% in SoH estimation.

6.2 Same-Battery Cross-Profile Validation

Next, we evaluate V-BASH based on training and validation traces collected with the same battery but during cycling tests with different profiles. These results reveal whether the voltage-based SoH fingerprint relies on particular charging/discharging profiles, in which case V-BASH's accuracy in SoH estimation may degrade in practice due to dynamic user behaviors. We perform additional cycling tests for these validations as summarized in Table 4, based on which 4 validation cases are explored as summarized in the middle of Fig. 16. Specifically, Cases #8 and #9 use the traces with different charging currents for training and validation (e.g., a real-life analogy when users use non-standard chargers to charge their devices), and Cases #10 and #11 use traces with different discharging currents (e.g., users with heavy or light usage patterns). Comparison between the bottom and middle parts of Fig. 16 shows the estimation errors for cross-profile validation, albeit larger than the same-battery same-profile case, are still within 5.6%, verifying V-BASH's tolerance to dynamic user's real-life behaviors.

6.3 Same-Model Cross-Battery Validation

Last but not the least, we evaluate V-BASH based on training and validation traces collected with different batteries of the same model. These evaluations are similar to V-BASH's real-life implementation — training it with batteries for particular device models offline, and then estimate the battery SoH of other same-model devices used by users in real-life. For each device model with multiple batteries in Table 3, we use the testing traces collected with one battery for training and use those collected with other batteries for validation. This way, a total number of 16 combinations of training/validation traces are explored, as summarized in the top of Fig. 16, showing a maximum error of 4.2% (in Case #27) and a mean of 1.78%.

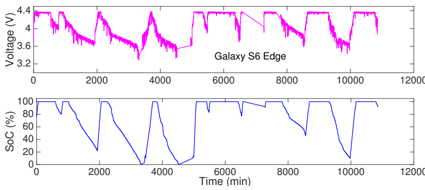
7 FIELD TEST ON ANDROID DEVICES

We also implemented V-BASH and verified its accuracy in SoH estimation on Android platform with multiple devices.

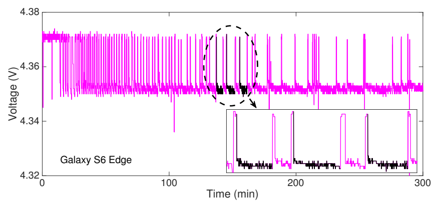
7.1 Implementation Details

We first explain a few challenges when implementing V-BASH and the corresponding remedies.

• **Ground-Truth Collection.** The first challenge in V-BASH's field test is the ground truth collection. The unreliable PMIC-provided current readings on mobile devices (as observed in Table 2) discourage us to estimate the true SoH based on them. A more reliable approach is to fully charge and then discharge the battery with the battery tester, and then derive its full charge capacity based on



(a) 8-day voltage and SoC traces



(b) Extract the valid relaxing patterns

Figure 17: Collecting the relaxing voltage traces of a Galaxy S6 Edge phone: (a) the collected 8-day traces; (b) extracting valid relaxing patterns from trickle-charging-polluted traces (zoom-in of the first 500 minutes voltage traces of (a), during which the phone has 100% SoC).

the logged process. This, however, requires removable device battery. As a result, we implement V-BASH on three Android devices, i.e., Galaxy S5, Galaxy Note 2, and Nexus S, all with removable batteries, and verify its SoH estimation accuracy by comparing with the measured battery SoH.[4] We have also implemented V-BASH on other devices with non-removable batteries (e.g., Galaxy S6 Edge and Nexus 6P) to verify its generality.

• **Relaxing Voltage Collection.** Device battery starts relaxation once fully charged, reflected by a 100% SoC. This way, V-BASH starts logging the battery voltage once observing a 100% battery SoC, and continuous until (i) the charger is disconnected or (ii) a long-enough relaxation period has been logged.

For Android devices, battery voltages can be collected via the `BatteryManager`: register a `BroadcastReceiver` to receive the broadcasted voltage information. However, `BatteryManager` broadcasts with a limited frequency to reduce its power consumption. For example, only 158 voltage readings are collected via the `BatteryManager` during a 790-minute relaxing period with a Galaxy S5 phone, which is not enough for V-BASH to collect the needed information.

As a remedy, instead of using `BatteryManager`, we implement a voltage collector with customizable sampling frequency, by reading the system file containing voltage information. For example, the battery voltage can be read from `/sys/class/power_supply/battery/voltage_now` for Nexus 6, Nexus 6P, etc. Also, we implement this voltage collector as a foreground service to avoid being terminated by Android when under memory pressure, achieved with the `startForeground` method. Although this foreground logging service increases power consumption when compared to its background counterpart, it is tolerable for V-BASH because of the connected charger and thus the existence of external power supply.

• **Trickle Charging Mitigation.** Mobile devices uses trickle charging — charging a fully charged battery under no-load at a rate equal to its self-discharge rate, to keep their batteries remain fully charged. This, however, invalids the battery relaxation and thus pollutes the collected relaxing voltages. Specifically, trickle charging on Android devices is triggered once the voltage of a fully-charged battery drops for a pre-defined value (e.g., 20mV for Galaxy S6 Edge), and stops once the battery voltage reaches the fully-charged level again. Fig. 17(a) plots the voltages and SoCs of a Galaxy S6 Edge phone during 1-week usage. The battery SoC reaches and stays at 100% from time to time, reflecting the long-time charging. However, zooming-in of Fig. 17(a) (as shown in FIg. 17(b)) shows the battery voltage after fully charging fluctuates between 4.37V and 4.35V, as a result of trickle charging. As a result, the collected relaxing voltages are not of the power-shape as shown in Fig. 12.

V-BASH mitigates these polluted relaxing voltages with the sub-traces between consecutively triggered trickle charging, whose duration increases over the relaxing period as observed in Fig. 17(b). Specifically, V-BASH examines the voltage sub-traces between consecutive trickle charging, smoothes them with a moving average filter, and concludes these sub-traces are valid for SoH estimation if (i) they last long enough (e.g., over 5 minutes), and (ii) power fitting of these traces returns good-enough goodness-of-fit (e.g., RMSE < 0.002 and R-Squared > 0.95). Fig. 17(b) also highlights the thus-extracted sub-traces, based on which the battery SoH is estimated. The feasibility of this pattern extraction technique is also verified on devices such as Nexus 5X, Galaxy S5, and Note 2.

7.2 Field Test Results

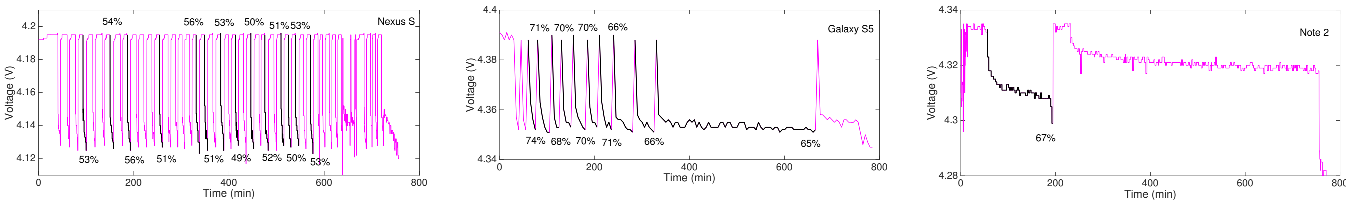
Fig. 18 shows the estimated SoH of the three devices based on relaxing voltages collected during an over-night charging. Extracting the valid relaxing patterns, a total number of 14, 10, and 1 valid sub-traces are obtained (as highlighted) for the three devices respectively, based on which their battery SoH is estimated. Taking Fig. 18(a) as an example, the battery SoH of the Nexus S phone is estimated as 49–56%, with a mean of 52%. Fully charging and discharging its battery with the battery tester shows a measured ground truth SoH of 54.9%, indicating a -2.9% error in SoH estimation. The SoH estimation errors for Galaxy S5 and Note 2 are $69 - 74.2 = -5.2\%$ and $67 - 73 = -6\%$, respectively.

8 CONCLUSION

In this paper, we have designed V-BASH, an SoH estimation method for mobile devices based on only their voltage information. V-BASH is based on an empirically revealed fact that the relaxing battery voltages indicate their SoH, and is enabled on mobile devices with a common usage pattern that most users charge their devices for a long time and frequently. The design of V-BASH is guided by 2,781 empirically connected voltage relaxing traces with 19 batteries. We have evaluated V-BASH using both laboratory experiments and field tests on multiple mobile devices.

Acknowledgment: The work reported in this paper was supported by NSF under Grants CNS-1329702 and CNS-1446117.

[4]The battery SoHs are measured with the same charging/discharge profiles as when collecting their training traces, eliminating the noises caused by battery's rate-capacity effect.



(a) Nexus S **(b) Galaxy S5** **(c) Note 2**

Figure 18: Field-tests results of V-BASH on Android devices: (a) 14 valid relaxing sub-traces are extracted on Nexus S, concluding a (estimated) mean SoH of 52%; the measured true SoH is 54.9%; (b) 10 valid relaxing sub-traces are extracted on Galaxy S5, concluding a (estimated) mean SoH of 69%; the measured true SoH is 74.2%; (c) 1 valid relaxing sub-trace is extracted on Note 2, concluding an estimated SoH of 67%; the measured true SoH is 73%.

REFERENCES

[1] 2016. Ampere the charging meter. (2016). http://forum.xda-developers.com/android/apps-games/app-ampere-charging-meter-t3012890.
[2] 2016. Apple finally reveals the cause of its iPhone 6s "unexpected shutdown" bug. (2016). http://www.dailymail.co.uk/sciencetech/article-4000774/Apple-FINALLY-reveals-cause-iPhone-6s-unexpected-shutdown-bug-Chinese-website.html.
[3] 2016. BatteryManager. (2016). https://developer.android.com/reference/android/os/BatteryManager.html.
[4] 2016. Galaxy S4: 10 problems users have, and how to fix them. (2016). http://www.digitaltrends.com/mobile/samsung-galaxy-s4-problems/.
[5] 2016. iPhone 5 shuts down with plenty of battery life left. (2016). https://discussions.apple.com/thread/5451794?start=45&tstart=0.
[6] 2016. MAX17047/MAX17050 ModelGauge m3 Fuel Gauge. (2016).
[7] 2016. Nexus 6P goes from 15% to 0% almost straight away. (2016). https://productforums.google.com/forum/#!topic/nexus/SeB67voFk38.
[8] 2016. Power Functions. (2016). http://wmueller.com/precalculus/families/1_41.html.
[9] 2016. Samsung Galaxy S4 turns off but still has 30% battery life. (2016). http://forums.androidcentral.com/samsung-galaxy-s4/303065-samsung-galaxy-s4-turns-off-but-still-has-30-battery-life.html.
[10] 2016. Temperature on Battery Voltage. (2016). http://www.trojanbattery.com/Tech-Support/FAQ/Temperature.aspx.
[11] Hoque Mohammad A. and Tarkoma Sasu. 2015. Understanding Smartphone State of Charge Anomaly. In *HotPower'15*.
[12] Dave Andre, Christian Appel, Thomas Soczka-Guth, and Dirk Uwe Sauer. 2013. Advanced mathematical methods of SOC and SOH estimation for Lithium-ion batteries. *Journal of Power Sources* 224 (2013), 20 – 27.
[13] Davide Andrea. 2010. Battery Management Systems for Large Lithium-ion Battery Packs. *Artech House* (2010).
[14] Issam Baghdadi, Olivier Briat, Philippe Gyan, and Jean Michel Vinassa. 2016. State of health assessment for Lithium batteries based on voltage-time relaxation measure. *Electrochimica Acta* 194 (2016), 461 – 472.
[15] Yevgen Barsukov and Jinrong Qian. 2013. Battery power management for portable devices. *Artech House* (2013), 67.
[16] J. Bond, J. Dermott, and E. Listerud. 2016. Systems and methods for determining battery state-of-health. (2016).
[17] Marie Brewis. 2016. How to fix poor battery life on Android 5.0 Lollipop and Android 6.0 Marshmallow. *TechAdvisor* (2016).
[18] Isidor Buchmann. 2012. Checking Battery Health while Charging. *Cadex Electronics* (2012).
[19] John Cannarella and Craig B. Arnold. 2014. State of health and charge measurements in Lithium-ion batteries using mechanical stress. *Journal of Power Sources* 269 (2014), 7 – 14.
[20] L. Canals Casals, A. M. Schiffer Gonzlez, B. Amante Garca, and J. Llorca. 2016. PHEV Battery Aging Study Using Voltage Recovery and Internal Resistance From Onboard Data. *IEEE Transactions on Vehicular Technology* 65, 6 (June 2016), 4209–4216.
[21] Yi-Hsien Chiang and Wu-Yang Sean. 2011. Apparatus for estimating battery state of health. (2011).
[22] J. Christophersen, J. Morrison, W. Morrison, and C. Motloch. 2012. Rapid Impedance Spectrum Measurements for State-of-Health Assessment of Energy Storage Devices. *SAE Int. J. Passeng. Cars - Electron. Electr. Syst.* 5, 1 (2012), 246–256.
[23] Denzil Ferreira, Anind K. Dey, and Vassilis Kostakos. 2011. Understanding Human-smartphone Concerns: A Study of Battery Life. In *Pervasive'11*.
[24] Bill Giovino. 2015. Making sense of current sensing. *While Paper* (2015).
[25] Zhen Guo, Xinping Qiu, Guangdong Hou, Bor Yann Liaw, and Changshui Zhang. 2014. State of health estimation for Lithium ion batteries based on charging curves. *Journal of Power Sources* 249 (2014), 457 – 462.
[26] Hongwen He, Rui Xiong, and Jinxin Fan. 2011. Evaluation of Lithium-Ion Battery Equivalent Circuit Models for State of Charge Estimation by an Experimental Approach. *Energies* 4 (2011), 582–598.
[27] Liang He, Sunmin Kim, and Kang G. Shin. 2016. ∗-Aware Charging of Lithium-ion Battery Cells. In *ICCPS'16*.
[28] L. He, G. Meng, Y. Gu, C. Liu, J. Sun, T. Zhu, Y. Liu, and K. G. Shin. 2016. Battery-aware mobile data service. *IEEE Transactions on Mobile Computing* PP, 99 (2016), 1–1.
[29] T. Kim, W. Qiao, and L. Qu. 2013. Online SOC and SOH estimation for multicell Lithium-ion batteries based on an adaptive hybrid battery model and sliding-mode observer. In *ECCE'13*.
[30] M. El Lakkis, O. Sename, M. Corno, and D. Bresch Pietri. 2015. Combined battery SOC/SOH estimation using a nonlinear adaptive observer. In *ECC'15*.
[31] A. Marongiu and D. U. Sauer. 2016. On-board aging estimation using half-cell voltage curves for LiFePO4 cathode-based lithium-ion batteries for EV applications. *International Journal of Automotive Technology* 17, 3 (2016), 465–472.
[32] Kong Soon Ng, Chin-Sien Moo, Yi-Ping Chen, and Yao-Ching Hsieh. 2009. Enhanced coulomb counting method for estimating state-of-charge and state-of-health of lithium-ion batteries. *Applied Energy* 86, 9 (2009), 1506 – 1511.
[33] Anki Reddy Papana, Harmohan N. ingh, Hassan Ali Kojori, Subodh Keshri, and David Lazarovich. 2015. Method and apparatus for online determination of battery state of charge and state of health. (2015).
[34] A. Pathak, Y. C. Hu, and M. Zhang. 2011. Bootstrapping energy debugging on smartphones: A first look at energy bugs in mobile devices. In *HotNets'11*.
[35] Harmohan Singh, Thirumalai G. Palanisamy, Richard B. Huykman, and William C. Hovey. 2002. Systems and method for determining battery state-of-health. (2002).
[36] B. Sood, M. Osterman, and M. Pecht. 2013. Health monitoring of lithium-ion batteries. In *ISPCE'13*.
[37] Radu Tyrsina. 2016. What Causes Android Heating Issues and How to Avoid. (2016). http://techpp.com/2011/08/18/android-heating-issues-causes-and-how-to-avoid/.
[38] Daniel Wagner, Andrew Rice, and Alastair Beresford. 2013. Device Analyzer: Understanding smartphone usage. In *MOBIQUITOUS'13*.
[39] Caihao Weng, Jing Sun, and Huei Peng. 2014. A unified open-circuit-voltage model of Lithium-ion batteries for state-of-charge estimation and state-of-health monitoring. *Journal of Power Sources* 258 (2014), 228 – 237.
[40] Bolun Xu. 2013. Degradation-limiting Optimization of Battery Energy Storage Systems Operation. *Master Thesis, ETH* (2013).
[41] Fengyuan Xu, Yunxin Liu, Qun Li, and Yongguang Zhang. 2013. V-edge: Fast Self-constructive Power Modeling of Smartphones Based on Battery Voltage Dynamics. In *NSDI'13*.
[42] Ming Yu, Yevgen Barsukov, and Michael Vega. 2008. Theory and Implementation of Impedance Track Battery Fuel-Gauging Algorithm in bq2750x Family. *Application Report, SLUA450* (2008).
[43] Ming Yu and Michael Vega. 2008. Impedance Track Fuel Gauge Accuracy Test for GSM Phone Applications. *Application Report, SLUA455* (2008).
[44] A. Zenati, P. Desprez, H. Razik, and S. Rael. 2012. A methodology to assess the State of Health of Lithium-ion batteries based on the battery's parameters and a Fuzzy Logic System. In *IEVC'12*.
[45] Yuan Zou, Xiaosong Hu, Hongmin Ma, and Shengbo Eben Li. 2015. Combined State of Charge and State of Health estimation over Lithium-ion battery cell cycle lifespan for electric vehicles. *Journal of Power Sources* 273 (2015), 793 – 803.