DarkReader: Bridging the Gap Between Perception and Reality of Power Consumption in Smartphones for Blind Users

Jian Xu[*]
Stony Brook University
Stony Brook, USA
jianxu1@cs.stonybrook.edu

Syed Masum Billah[*]
Stony Brook University
Stony Brook, USA
sbillah@cs.stonybrook.edu

Roy Shilkrot
Stony Brook University
Stony Brook, USA
roys@cs.stonybrook.edu

Aruna Balasubramanian
Stony Brook University
Stony Brook, USA
arunab@cs.stonybrook.edu

ABSTRACT

This paper presents a user study with 10 blind participants to understand their perception of power consumption in smartphones. We found that a widely used power saving mechanism for smartphones–pressing the power button to put the smartphone to sleep–has a serious usability issue for blind screen reader users. Among other findings, our study also unearthed several usage patterns and misconceptions of blind users that contribute to excessive battery drainage. Informed by the first user study, this paper proposes DarkReader, a screen reader developed in Android that bridges users' perception of power consumption to reality. DarkReader darkens the screen by *truly* turning it off, but allows users to interact with their smartphones. A second user study with 10 blind participants shows that participants perceived no difference in completion times in performing routine tasks using DarkReader and default screen reader. Yet DarkReader saves 24% to 52% power depending on tasks and screen brightness.

Author Keywords

Vision impairment, blind; screen readers, TalkBack, Android, iOS; low-power, battery, screen, brightness; privacy, curtain mode, shoulder-surfing.

CCS Concepts

•**Human-centered computing** → **Accessibility technologies; Accessibility systems and tools;** *Touch screens;* •**Security and privacy** → *Privacy protections;*

[*]These authors contributed equally

ASSETS '19, October 28–30, 2019, Pittsburgh, PA, USA.

ACM ISBN 978-1-4503-6676-2/19/10 ...$15.00.
http://dx.doi.org/10.1145/3308561.3353806

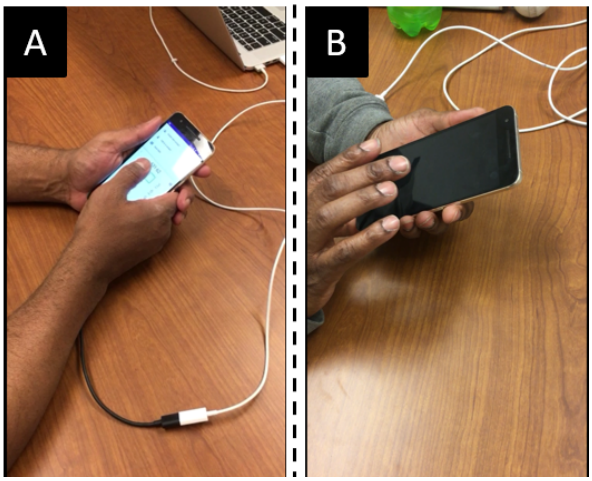


Figure 1. A blind user is interacting with his smartphone using a (A) standard screen reader (e.g., TalkBack in Android), and (B) DarkReader. Unlike TalkBack, DarkReader keeps the screen off to save power.

INTRODUCTION

Smartphone devices embody the convergence of personal computing, communication, and mobility. They have woven themselves inextricably into people's daily routines as a hub for information and communication. Thus, interacting with smartphones has also become an absolute necessity for people with vision impairments, just like the vast majority of people around the world.

People with vision impairments rely on using screen readers to interact with smartphones. Screen readers, such as TalkBack [35] in Android or VoiceOver [36] in iOS, are special-purpose assistive technology that read aloud the textual representation of the graphical user interface (UI) of an application. Screen readers also offer a number of special gestures (e.g., swipe left or right, double-tap) to navigate through the application UIs (e.g., buttons, menus), as well as to perform user actions in an accessible manner.

However, visually impaired users encounter two key problems when interacting with smartphones via screen readers. **First** is an issue that sighted users also face – the rapidly draining battery. Smartphones are battery-operated devices with limited charge capacity. As a result, power is one of the most constrained resources on smartphones. The battery problem is exacerbated for visually impaired users because they typically take longer time to complete a given task [26], requiring the phone to be switched on for an extended period of time, resulting in more power drain. For instance, to read news from a website, users with visual impairment must listen to the news sequentially, rather than to read them in saccade like sighted users, requiring more time to complete the task, and consequently more power consumption. The **second** smartphone problem encountered by visually impaired users is loss of privacy, where bystanders can eavesdrop on the content of the phone visually [8, 33].

One way to solve both of these problems, at least for visually impaired users who are blind, is to switch the smartphone's screen completely off. It makes sense because the screen is one of the most energy-intensive components on a mobile device [19], and these screen reader users do not benefit from the screen being on. Further, bystanders cannot eavesdrop or shoulder surf when the screen is off.

There are two modes that users can use to switch the screen off: **(i) Sleep mode:** where the user can press the power button, which puts the phone to sleep. One problem is that when the phone is in the sleep mode, the users cannot interact with the app; and **(ii) Curtain mode:** in this mode, popular in iOS [15], the screen is darkened in response to a pre-defined gesture (such as a three-finger triple taps). The advantage of the curtain mode is that users can still interact with the app, but the screen is darkened. On some Android phones, this functionality is called *dark-screen* mode.

Preliminary User Study. To understand blind users' interaction experience with these two modes—sleep mode and curtain/dark-screen mode—and their perception of the power consumption in smartphone in general, we conducted a user study with 10 blind participants. The study uncovered several issues with these modes ranging from the usability of sleep mode to the perception of power consumption in curtain mode. The study also revealed that (i) unlike sighted users, blind users are reluctant to use auto-lock, a feature that automatically puts a phone to sleep mode after being idle for some time; (ii) as soon as battery indicator goes under 20%, the sheer feeling of running out of power causes blind users great anxiety; (iii) curtain mode only protects privacy, and offers very little to no benefit in saving the battery because it does not truly turn off the screen; and (iv) blind users prefer battery status to be reported as an estimated remaining usage time, rather than a percentage of battery lifetime as shown in their phones.

Proposed Solution. Findings from the study led us to develop ***DarkReader***, an ultra-low-power screen reader on top of Android's TalkBack that keeps the screen off ("goes dark"), similar to the screen darkening techniques, such as curtain mode. But unlike curtain mode, DarkReader *truly* turns off the screen and saves *24%* to *52%* power, depending on tasks and the brightness of screen.A second study with the same participants indicated that the users perceived no difference in completion times in performing routine tasks while using DarkReader compared with using TalkBack.

In summary, our contribution is two-fold:

- A study that reveals blind users' perception and behaviors towards power consumption in smartphones.
- *DarkReader*, a power-saving, privacy-preserving screen reader for blind smartphone users.

RELATED WORK

Mobile Power Consumption Mitigation

In mobile research community, energy consumption has attracted a great deal of attention, as battery performance is a bottleneck for all stakeholders—hardware manufacturers, operating system designers, app developers, and end users. Chen et al. [19] show that phone screens consume over 20% of the total energy usage on a daily average, pointing to the foremost factor in power consumption.

In order to mitigate this over-spending of power from screen usage, researchers have proposed several systems. Dalton et al. [20] propose a system that turns the display either on or off based on the presence of a user. He et al. [23] dynamically scale the resolution of the smartphone display depending on the distance between the user and the screen for saving power without affecting user experience.

Mobile operating systems (OSes), such as Android and iOS, provide their own low-power modes for saving power. These power saving modes reduce energy consumption by disabling certain functionalities. For example, in Android, the low-power mode [13] saves battery by disabling power hungry apps or functionalities, such as map services, vibration sensors, and background services (e.g., fetching emails). Similarly, Samsung offers an ultra-power-saving mode [30] that saves battery by changing the color of the screen to grayscale and restricting other functionalities.

Xu et al. [39] propose an ultra-low-power mode (ULPM) in Android that takes extreme measures to save battery by switching the screen off, but keeping the application interactive. However, blind users cannot benefit from this approach because ULPM does not work with screen readers. Existing screen readers run as an accessibility service, and the ULPM technology does not apply to accessibility services.

Another line of work is to use machine learning techniques to estimate smartphones' remaining usage time at any instant [27, 16]. These techniques can greatly benefit blind users, as knowing how long their battery would last reduces their anxiety associated with smartphone battery running out of power.

Perception and Concerns in Non-Visual Interaction

Rahmati et al. [28] conducted a large-scale survey on the power consumption of mobile phone users for a month, and concluded that mobile phone users can be categorized into

two groups: those who have considerable knowledge about their smartphone power consumption, and others who have inadequate knowledge regarding phone power characteristics. We found that majority of blind users belong to the latter group.

Several works [17, 8, 32] identify a number of security and privacy concerns for people with vision impairments when using technology. For instance, (i) aural eavesdropping while entering passcodes to unlock phones (Azenkot et al. [17]); (ii) physical and aural eavesdropping, shoulder surfing (Ahmed et al. [8]); and (iii) social stigma and misperceptions surrounding the use of assistive technologies (Shinohara et al. [32]). This paper uncovers the misconceptions that blind users possess about their smartphone power consumption and bridges the gap between the misconception and reality.

UNDERSTANDING BLIND USERS' PERCEPTION OF POWER CONSUMPTION IN SMARTPHONES

We recruited 10 legally blind participants (6 males, 4 females) through local mailing lists and word-of-mouth. All participants lived in urban or metro areas, were full-time screen reader users, with an average age of 40.8 (SD=12.8, Range=27∼60). Table 1 presents the anonymized participant demographics. Except for P5, all participants used iPhones that were more than 3 years old. P2 was still learning how to use an iPhone. P1 and P7 were familiar with both iPhone and Android, as they were professional IT instructors. The participants interacted with their phones for 1 to 8 hours daily. On average, they recharged their phones 1.8 times/day.

Study Procedure

The interviews were semi-structured, in-person, and conducted by two interviewers in an office environment. We started with asking general questions related to their smartphone usage, charging frequency, and their perception of power consumption by different apps, including screen readers. Later, we asked the following specific research questions: (1) What measures do they usually take to deal with the limited battery capacity of smartphones? (2) Are they aware of privacy-preserving curtain mode? (3) How do they react to battery indicators? (4) Are there any usability issues with the user interfaces for power-saving settings in smartphones? We also discussed potential solutions to problems that participants experienced. The interview process culminated with participants making suggestions and recommendations. Each interview lasted for an hour, and we compensated them by an hourly rate of $25 USD.

Analysis

All interviews were fully transcribed. Following the completion of the first five interviews, the interviewers analyzed the transcripts using an iterative coding process with initial coding and identified concepts [18], categorized them, framed new questions for subsequent interviews, and updated the concept list. Next, we report several key concepts that emerged from the study.

ID	Age/Sex	L.P	Personal Smartphone	Expertise
P1	35/M	Yes	iPhone, Android	Expert
P2	59/M	No	Feature phone, iPhone	Beginner
P3	53/M	No	iPhone	Inter.
P4	33/F	No	iPhone	Inter.
P5	29/F	No	iPhone X	Expert
P6	60/F	No	iPhone	Inter.
P7	30/F	Yes	iPhone, Android	Expert
P8	34/M	Yes	iPhone	Expert
P9	27/M	No	iPhone	Inter.
P10	48/M	No	iPhone	Inter.

Table 1. Participant demographics. L.P: Light Perception.

Findings

F1. Resetting Screen Reader Cursor in Sleep Mode

The participants stated a major usability issue with sleep mode—it resets the screen reader cursor to the beginning of an app or to the first UI element visible in the screen. For example, if the screen reader cursor is halfway through an app, and then the phone goes to sleep mode either by accidentally pressing the power button or by auto-lock, they have to start all over again, once the screen is back on. Therefore, unlike sighted users, blind users cannot pick up their cursor (i.e., screen reader cursor) where they left off.

F2. Disabling Timer-based Auto-Lock

Because of the aforementioned reason, all participants were frustrated with the timer-based auto-lock feature that automatically puts their phones to sleep mode after being idle for a certain amount of time. Six participants disabled auto-lock, whereas others applied a longer time-out (e.g, 5 minutes) using their phone's auto-lock settings.

F3. Awareness of Curtain Mode

With the exception of P2, all 9 participants were aware of the existence of curtain mode in iPhone. However, only 6 participants who disabled auto-lock, used curtain mode frequently. The remaining 3 participants, P3, P6, and P10, who did not use curtain mode frequently, stated the following reasons: (i) they forgot the gesture to enable/disable it, (ii) they often needed to show the screen to sighted persons, and (iii) they mostly stayed at home these days, hence were not concerned about screen privacy. Two participants (P1, P7) who were familiar with Android as well, mentioned that they were aware of dim-screen setting in TalkBack, but not the default dark-screen mode in Android. Henceforth, we call this dark-screen mode in Android as Android's curtain mode, to avoid confusion with our system DarkReader.

F4. Misconception Regarding Curtain Mode

All 9 participants (except for P2) believed that curtain mode saved power. P5 and P8 stated that they always kept their phones in curtain mode. However, they were puzzled by the fact that their phones were burning in the morning, even though they did not use them overnight.

F5. Anxiety Due to Low Battery

All participants reported that they experienced anxiety when hearing the warning message (pop-up) indicating that the bat-

tery availability is under 20%, particularly when they were at work or commuting. According to them, they relied on multiple smartphone apps these days including personal assistance (e.g., [3]), navigation and whereabouts (e.g., [4, 6]), and transportation (e.g., [2]). To avoid running out of power, they always carried external battery packs (e.g. JuiceBox [24], Morphie [25]) believing that those apps were power-hungry, but not realizing that some of their choices (e.g., not putting the phone to sleep mode) was also contributing to battery drain.

User Recommendation

Simpler gesture

The participants wished to have a simpler gesture to enable or disable the curtain mode. Ideally, they wanted to use power button, but realized that it was reserved for sleep mode.

Battery Notification

The participants stated that the reason why they got anxious when battery level dropped to 20% or below was because they were uncertain about how long the battery would last. They would prefer to know exactly when their phones would run out of charge at regular intervals, so that they could plan ahead.

Technical Analysis

In order to analyze our findings F1 and F4 from technical standpoints, we first gained a deeper understanding of how screen readers work in smartphones. To that end, we studied the source code of TalkBack [22], an open source screen reader in Android. Although the following explanations are based on our understanding of Android and TalkBack, we believe these explanations also stand for VoiceOver, a proprietary screen reader on iPhone.

Why Do Screen Readers Reset the Cursor in Sleep Mode?

In Android, TalkBack runs in the background as an Accessibility Service [9]. As shown in the Figure 2, it indirectly communicates with a user app via Accessibility Manager Service (AccMS) [10]. When a user starts interacting with an app, AccMS first extracts the meta information associated with each UI element (shown as *UI tree* in the figure) of the app, and sends this information to TalkBack. Based on this information, TalkBack then constructs an intermediate representation of the app to facilitate all screen reading functions, e.g., navigation, simulation of user action, screen reading cursor. If the UI tree changes due to a user activity, the app pushes those changes to AccMS, which relays the updated meta information to TalkBack by collaborating with Activity Manager Service [11], another Android service that manages the states and life-cycles of UI elements. TalkBack then updates its intermediate representation accordingly.

Since sleep mode is a power saving technique in smartphones by design, Android operating system (OS) suspends as many components as possible, including foreground apps and accessibility services (e.g., TalkBack), assuming that the user no longer wants to interact with her phone. When screen is back on, the OS brings back suspended apps to the foreground, triggering AccMS to extract the same meta information again. This is because, AccMS does not maintain any state information about the previous read position. Consequently, the screen reader starts from the beginning.

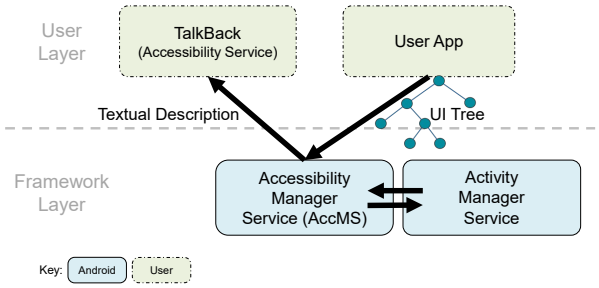


Figure 2. Underlying mechanism of how TalkBack communicates with a user app. Note that in Android, TalkBack runs as an Accessibility Service which is managed by the Accessibility Manager Service (AccMS).

Does Curtain Mode Save Power?

The anecdotes of P5 and P8 in *Finding F4* suggests that the role of curtain mode on power consumption needs further inspection. To this end, we conducted several lab experiments in both Android (Samsung Galaxy S9, Android 9 Pie) and iPhone (iPhone 5s, iOS v.11.4.1). We performed different tasks including making a phone call, reading a web article, and watching videos on YouTube using screen readers with their respective curtain modes switched off and switched on.

We found that curtain modes in both Android and iPhone provide little benefit in terms of power consumption. A description of the rigorous experiment along with its result is presented in the "Evaluation of DarkReader" section under "Power Consumption in Curtain Mode". Based on these experiments, we have reasons to believe that these modes only occlude the screen, but do not switch off resources for power saving. Simply occluding the screen does solve the privacy problem for blind users, as it keeps the application running, but at the expense of power.

DARKREADER

We propose DarkReader, an ultra-low-power screen reader that works as if it were in curtain mode, but saves power as if it were in sleep mode. The key challenge in DarkReader is as follows: how do we switch the screen off to save power, but ensure that the application and services (in this case, the screen reader) works correctly?

This is challenging because most mobile devices switch off all power-hungry components when the screen is off. Specifically for screen readers, when the screen is turned off (i) the user input is no longer sent to the screen reader, (ii) the current screen context is deleted (e.g., a green box in Android TalkBack that indicates users' current focus or cursor), (iii) and the application moves to the background and the read out of the screen content stops.

Dark Mode

As a first step, DarkReader introduces a new mode, ***dark mode*** that allows the screen reader to continue working correctly even when the screen is switched off. By default, DarkReader keeps the phone in *dark* mode. Users can temporarily disable dark mode if they need to show their screens to sighted users. To enable or disable dark mode, users can either press the power button (i.e., gesture for sleep mode), or tap 3 times with

3 fingers (i.e., gesture for curtain mode). Therefore, dark mode unifies both sleep and curtain modes.

DarkReader also supports auto-lock, which automatically enables dark mode by switching off the screen in 5 minutes, if the user forgets to do so.

Battery Status Notification in Dark Mode

Estimating the remaining battery time is beyond the scope of this work. However, in our study, we simulated this feature by announcing the message "Your phone will last for X hour(s)" at 3 different timestamps, where X was either 2, 1, or 0.5.

Techniques to Enable Dark Mode

DarkReader uses four techniques to overcome the limitations of Android system when the screen is switched off.

Deliver user inputs to the hardware driver.

The first limitation is that user inputs can no longer be detected by the device when the screen turns off; this is because when the screen is switched off, the OS puts hardware drivers [37] into a sleep status causing the hardware composer [12] to stops rendering images on the screen, as well as receiving user inputs. We leveraged ULPM [39] that instruments Android OS to keep receiving user inputs under this circumstance, without rendering images on the screen.

Deliver user inputs to screen readers.

Although the instrumented OS is able to detect user inputs by now, the *screen reader* still cannot receive user inputs when the screen is switched off. This is because screen readers detect user input events through accessibility services supported by the OS. These services often use a set of policies to filter user inputs, and one of the policies is to drop user inputs when the screen is switched off. We change this filter policy to allow user inputs to reach Accessibility Manager Service (as shown in Figure 3), which then delivers these inputs to a screen reader.

Retain the screen reader cursor.

The third problem is that the screen reader cursor and cache is deleted when the screen is switched off. As shown in Figure 3, Text-to-Speech (TTS) component in Android is responsible for reading out the screen content that is stored in a textual buffer. This buffer is destroyed immediately when the screen turns off causing a loss of screen reader cursor. DarkReader overcomes this problem by proactively caching the screen context as well as Text-to-Speech buffer. As a result, the screen reader cursor is always preserved, and the screen reader is able to read out screen content uninterruptedly as long as our dark mode is enabled.

Update application UI

By default, when the screen is switched off, a foreground app is moved to the background, and its UI is no longer being updated. If the UI is not getting updated in response to user input, the application will not function correctly. To resolve this problem, we leveraged an existing approach from UIWear [38], which assigns a special *non-stop* status to specified running apps. This *non-stop* status allows an application to run and update its UI regardless of the screen status, but with minimal power consumption.

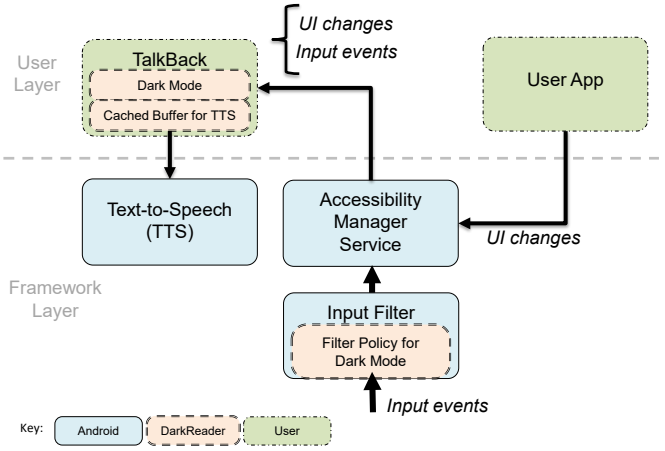


Figure 3. The overall architecture of DarkReader which is built on top of TalkBack. DarkReader introduces *dark mode* in Android system. Highlighted components including the Android OS needs to be instrumented.

The result of these four techniques is that DarkReader is able to switch the screen off, the user can continue to interact with the screen reader, *and* there is considerable power savings.

Implementation

DarkReader is built on top of TalkBack [22] running on LineageOS v14.1 [5] (Android v7.1.2). Figure 3 shows different components in both Android OS and TalkBack that is instrumented for DarkReader to work.

We deployed DarkReader on a Nexus 5 and a Nexus 6P, and used these phones for our user studies and power analysis. We believe that the design principles of DarkReader will work across different Operating Systems, including iOS and Tizen [29], as the notions of caching, filter policies, and Accessibility [14, 31] carry over to these OSes as well.

EVALUATION OF DARKREADER

We conducted another IRB-approved user study: (i) to measure the perceived difference in user experience between DarkReader and the default TalkBack screen reader, and (ii) to collect data (e.g., users' logs, activity traces) for a posterior analysis of power consumption. Since this study is distinct from the first study, there is no carryover effect. Therefore, we recruited all ten participants from our first study to also participate in the second study. We aim to validate the following hypotheses:

- **H1**: There is no difference in completion times in performing routine tasks using DarkReader with dark mode on and default screen reader.
- **H2**: DarkReader consumes significantly less power compared to the default screen reader.

Apparatus and Materials

We designed 3 real-world tasks the participants routinely do on their smartphones. For instance, making a phone call, browsing the web, consuming multimedia content. More specifically, we asked participants to perform the following tasks:

- **T1**: (Essential phone operation) make a phone call to a given number.
- **T2**: (Internet Browsing) Read the first two paragraphs of a given article from Wikipedia.
- **T3**: (Multimedia consumption) Listen to a given video for 30 seconds from YouTube.

The experiment was conducted on a Nexus 6P smartphone running our custom Android ROM, Nougat v.7.1.2. All user actions (e.g., taps, swipes, timestamps) along with application states were logged in a file on the device.

Procedure

The study was a within-subject design. Each participant performed the above 3 tasks under the following 2 study conditions:

- **TalkBack**: Participants could use either VoiceOver or native TalkBack gestures. They were not allowed to press the power button. This was our baseline.
- **DarkReader**: Participants could use either VoiceOver or native TalkBack gestures. The screen was in *dark mode* (i.e., turned off).

We disabled the timer-based auto-lock while using TalkBack. In our custom ROM, we also simulated the gestures of VoiceOver. Links to Wikipedia articles and YouTube videos used in the study were stored in a file on the device. For tasks T2 and T3, the experimenter clicked on a link first and handed the phone to participants when that article or video was loaded in the browser or on YouTube.

To minimize learning effect, we counterbalanced the ordering of tasks and study conditions. Prior to the study, participants were given sufficient instructions and time ($\sim$10 min) to familiarize themselves with the gesture (e.g., native TalkBack, or simulated VoiceOver) of their choice. However, all participants chose to use VoiceOver gestures. Once a participant was fully comfortable with the device and gestures, we asked them to perform the three aforementioned tasks.

To observe participants' reactions on battery status notification, we played a message *"Your phone will last for X hour(s)"* at 3 different timestamps during the course of each study – one at the beginning (X = 2), one at the middle (X= 1), and towards the end (X= 0.5). These messages were artificial.

Each participant was allotted 5 minutes to complete a task. All conversations during the study were in English. The experimenter took notes during a session. Each session lasted for an hour. All sessions were video recorded and coded post-transcription. In total, we collected data and logs for 60 tasks (= 10 participants x 2 conditions x 3 tasks/condition/participant).

Study Result

Completion time

We analyzed the experimenters' notes, the recorded videos, and the traces to measure completion time. Since our sample size was small, we used Wilcoxon Signed-Rank tests to determine statistical significance. The mean completion times

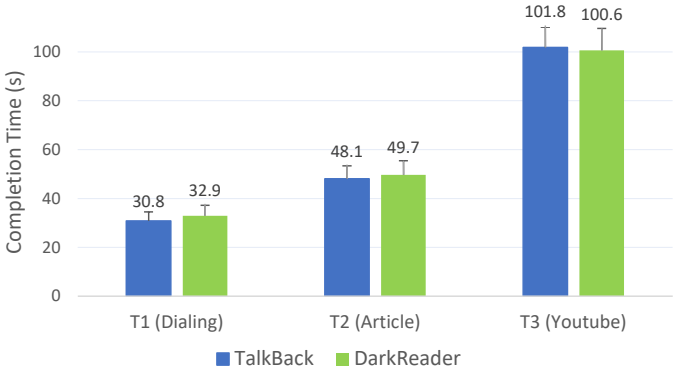


Figure 4. Mean completion times (sec.) for 3 tasks T1, T2 and T3 using TalkBack and DarkReader. Error bars show +1 SD.

for tasks T_1, T_2 and T_3 under each condition are shown in Figure 4. The Wilcoxon Signed-Rank test showed that study conditions had no significant effect on the completion time for tasks T1 ($Z = -.870$, $p = .384$), T2 ($Z = -.90$, $p = .366$), and T3 ($Z = -.307$, $p = .759$).

Subjective Feedback

With the exception of participants P1, P7, and P8, who had some light perception, no other participants were aware of which screen reader they were using to complete a task at any instant. P1, P7 and P8 noticed that the screen went dark occasionally, but they thought this was because of using TalkBack with curtain mode.

Battery Indicator

Participants commented that the knowledge of remaining battery time would help them better decide whether to recharge their phones. For example, after hearing the last announcement, 6 participants asked the experimenter to put the study phone on charge, whereas others deemed it unnecessary, as the study was almost over.

The above objective and subjective results thus validated our hypothesis H1.

Posterior Analysis of Power Consumption

Next, we report the amount of power DarkReader and TalkBack consumed for each task based on the user traces collected from the study. Note that this posterior analysis allowed us to report power statistics of TalkBack with multiple screen conditions (e.g., different brightness), without repeating the study multiple times in each of those screen conditions. Using Android's ADB tool, we emulated the aforementioned tasks by replaying user traces and activities.

In order to measure power consumption with high precision, we utilized BattOr [1], a portable, hardware power monitoring tool that reports power consumption for every 100 microseconds. Energy consumption was then estimated as a function of power and time.

Each user trace was replayed under 3 screen conditions (with screen brightness varied from 100% to 50% to 0%) when using TalkBack, since these variances can affect the power

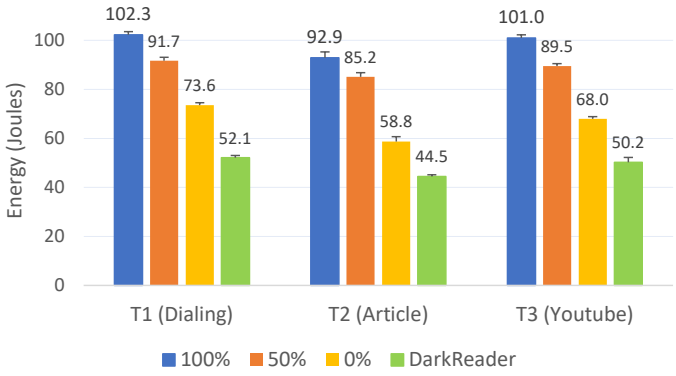


Figure 5. Power consumption in tasks T1, T2, T3 when using TalkBack with 3 different screen conditions (e.g., brightness 100%, 50%, 0%), and DarkReader with *dark mode* on.

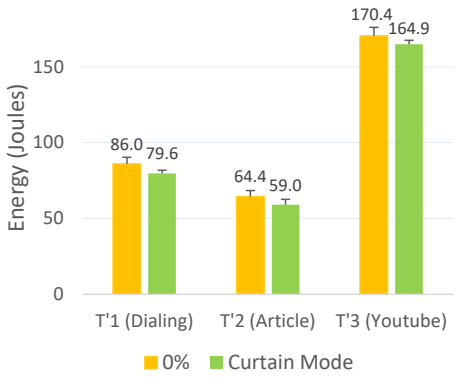
consumption. Figure 5 shows energy consumption in each task under different conditions. From this figure, it is evident that as the brightness increased from 0% to 50% to 100%, DarkReader saved 29% to 43% to 49% energy for T1, 24% to 47% to 52% energy for T2, and 26% to 43% 50% energy for T3—all of these measures were found to be statistically significant, as reported by Wilcoxon Signed-Rank tests ($Z = -2.803$, $p < .005$). So, we conclude that even if the screen is completely dimmed (i.e., 0% brightness), DarkReader could still save at least 24% of battery. These measures validated our hypothesis H2.

Power Consumption in Curtain Mode

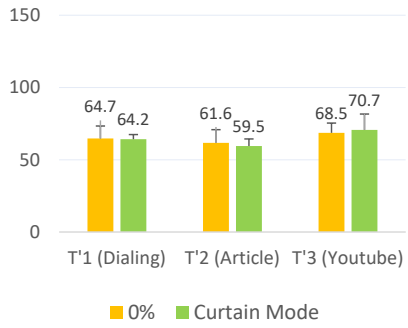
We measured the effect of using curtain mode in iOS and Android devices in terms of power consumption. Since curtain mode is only available in iPhones and in selected Android phones, we conducted our experiments in an iPhone 5s (iOS v.11.4.1) and a Samsung Galaxy S9 (Android 9 Pie). We used GameBench [7] software tool to measure power consumption in iPhone 5s. Since BattOr only measures power in Nexus 5, we used *dumpsys* [21] software tool to measure power on Galaxy S9.

Furthermore, we were unable to replay user traces in these phones because of the differences in platforms and OSes. Therefore, we manually emulated the study tasks as follows: (i) T′1: we emulated the task T1 by entering a single digit in each second from Dialer for 60 seconds; (ii) T′2: we emulated the task T2 by performing a swipe right gesture in Wikipedia articles every 5 second for 60 seconds; and (iii) T′3: we played a YouTube video for 60 seconds to emulate task T3. Each of these three tasks was repeated 5 times. We conducted these experiments using VoiceOver (iPhone) and TalkBack (Android) under two screen conditions: keeping the screen brightness to 0% while enabling and disabling curtain mode.

As shown in Figure 6.a, curtain mode saved 7% for T′1, 8% for T′2, and 3% for T′3 power compared to no curtain model under zero brightness conditions on iPhone. On Android phones (shown in Figure 6.b), the curtain model saved only 1% power compared with no curtain mode, under zero-brightness conditions.



(a) Power draw in iOS



(b) Power draw in Android

Figure 6. Power consumption in simulated tasks T′1, T′2, and T′3. We turned on (a) curtain mode with VoiceOver on iPhone 5s and (b) dark-screen mode (which is the curtain-mode equivalent in Android) with TalkBack on Samsung Galaxy S9 to compare with 0% brightness on each device. Note that iPhone 5s has a LCD screen, whereas Galaxy S9 has an AMOLED screen.

We theorize that these modes occlude the screen and turn off the back-light, but do not switch the screen off, because (a) the power consumption is commiserate with this observation, and (b) current applications stop working when the screen is switched off. However, we do not have the access to the internals of iOS and Samsung Galaxy OS, so we are unable to verify this theory.

DISCUSSION AND FUTURE WORK

DarkReader offers people with vision impairments a "disability gain", as they are capable of interacting with phones without keeping the screen on, thereby conserving power that would be otherwise consumed by the screen. Our results have shown that depending on the task and screen condition, DarkReader can save more than 50% battery. For example, if a blind user is browsing a webpage in broad daylight, most likely the brightness of her phone screen will be set to 100% by the automatic-brightness feature built-in to her iOS or Android device. In this circumstance, using DarkReader yields her a net saving of 52% power, without interrupting her interaction experience.

One of our limitations while experimenting with curtain mode in iPhone and Android was we emulated the tasks manually.

In future, we will automate these experiments by leveraging accessibility or automated GUI testing tools in iOS. The GameBench software tool we used to measure power consumption in iPhone is not precise. We plan to use a more precise, hardware power monitoring tool, such as Monsoon Monitor [34] in the future, but this requires tearing down the iPhone.

The reported power savings in this paper are applicable for traditional smartphones equipped with LCD (Liquid-Crystal Display) screens. But AMOLED (Active-Matrix Organic Light-Emitting Diode) screen is becoming popular. In fact, the Samsung Galaxy S9 phone we used in our experiments shown in Figure 6.b uses an AMOLED display. As shown in that figure, the display technology of AMOLED is more power efficient than LCD, particularly for the simulated task T′3 in comparison with that in Figure 6.a. We want to further investigate the effect and benefit of DarkReader on smartphones equipped with AMOLED screens.

From our first study, we learned that blind users interact with many apps that use power-intensive modules, such as camera, GPS, cellular data. We will conduct more studies to understand the effect of DarkReader while running these power-intensive apps in real-world scenarios. Finally, we aim to incorporate a predictive model into DarkReader to report estimated remaining battery time in realtime.

CONCLUSION

In this paper, we conducted two studies with 10 blind participants. Our first study revealed blind users' perceptions of power consumption in smartphones. We found that a widely used power saving mechanism for smartphones–pressing the power button to put smartphone to sleep–has a serious usability issue for screen reader users. Blind users cannot resume screen reading from where they left off if their phones go to sleep mode in between. For this reason, many blind users do not let their phone to sleep. These users resort to a privacy preserving mode, known as curtain mode, that darkens the screen. A majority of our participants thought that this curtain mode also saves power, but that was not the case. Informed by our first user study, we designed a screen reader, DarkReader, by modifying different components of Android OS and TalkBack screen reader. DarkReader bridges blind users perception to reality since it works as if the phone were in curtain mode, but saves power as if it were in sleep mode. A second user study with 10 blind participants shows that participants perceived no difference in interaction experience between DarkReader, and TalkBack with screen on or off. Yet, DarkReader can save 24% to 53% power depending on the task and the screen condition. We encourage device manufacturers to consider adding DarkReader mode as a core feature for accessibility and privacy.

ACKNOWLEDGMENTS

We thank IV Ramakrishnan, Vikas Ashok, our study participants, and the anonymous reviewers for their insightful feedbacks. We also thank Lighthouse Guild at New York City to facilitate our studies. This research was supported by NSF CSR-1717973.

REFERENCES

[1] 2011. BattOr: Portable Power Monitor for Mobile Phones. (2011). https://cseweb.ucsd.edu/~schulman/battor.html

[2] 2019. Access Service. (2019). https://accessla.org/home/

[3] 2019. Be My Eyes: Bringing sight to blind and low vision people. (2019). https://www.bemyeyes.com/

[4] 2019. BlindSquare: Pioneering accessible navigation - indoors and outdoors. (2019). http://www.blindsquare.com/

[5] 2019. LineageOS Android Distribution. (2019). https://lineageos.org/

[6] 2019. Seeing AI. (2019). https://www.microsoft.com/en-us/seeing-ai/

[7] 2019. Who uses GameBench? (2019). https://www.gamebench.net/

[8] Tousif Ahmed, Roberto Hoyle, Kay Connelly, David Crandall, and Apu Kapadia. 2015. Privacy Concerns and Behaviors of People with Visual Impairments. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 2702334, 3523–3532. DOI: http://dx.doi.org/10.1145/2702123.2702334

[9] Android. 2019a. Accessibility. (2019). https://developer.android.com/guide/topics/ui/accessibility/index.html

[10] Android. 2019b. Accessibility Manager in Android. (2019). https://developer.android.com/reference/android/view/accessibility/AccessibilityManager

[11] Android. 2019c. Activity Manager in Android. (2019). https://developer.android.com/reference/android/app/ActivityManager

[12] Android. 2019d. Android Hardware Composer. (2019). https://source.android.com/devices/graphics/arch-sf-hwc

[13] Android. 2019e. Use battery saver mode. (2019). https://support.google.com/nexus/answer/6187458?hl=en

[14] Apple. 2019. UIAccessibility. (2019). https://developer.apple.com/documentation/uikit/accessibility/uiaccessibility

[15] AppleVis. 2014. Screen curtain on. (2014). https://www.applevis.com/forum/ios-ios-app-discussion/screen-curtain

[16] Android Authority. 2017. Google explains how it makes personalized battery life predictions. (2017). https://www.androidauthority.com/google-pixel-battery-life-estimation-817194/

[17] Shiri Azenkot, Kyle Rector, Richard Ladner, and Jacob Wobbrock. 2012. PassChords: Secure Multi-touch Authentication for Blind People. In *Proceedings of the 14th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '12)*. ACM, New York, NY, USA, 159–166. DOI: http://dx.doi.org/10.1145/2384916.2384945

[18] A. Bryman and R.G. Burgess. 1994. *Analyzing Qualitative Data*. Routledge. https://books.google.com/books?id=KQkotSd9YWkC

[19] Xiaomeng Chen, Abhilash Jindal, Ning Ding, Yu Charlie Hu, Maruti Gupta, and Rath Vannithamby. 2015. Smartphone Background Activities in the Wild: Origin, Energy Drain, and Optimization. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*. ACM, 2790107, 40–52. DOI:http://dx.doi.org/10.1145/2789168.2790107

[20] Angela Dalton and Carla Schlatter Ellis. Year. Sensing User Intention and Context for Energy Management. In *HotOS*.

[21] Android Developers. 2019. dumpsys tool. (2019). https://developer.android.com/studio/command-line/dumpsys

[22] Google. 2019. Android TalkBack Github. (2019). https://github.com/google/talkback

[23] Songtao He, Yunxin Liu, and Hucheng Zhou. 2015. Optimizing Smartphone Power Consumption through Dynamic Resolution Scaling. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*. ACM, 2790117, 27–39. DOI: http://dx.doi.org/10.1145/2789168.2790117

[24] JuiceBox. 2019. JuiceBox. (2019). https://juiceboxbattery.com/

[25] Mophie. 2019. Mophie powerstation External Battery for Universal Smartphones and Tablets. (2019). https://www.mophie.com/shop/powerstation-smart-phones-tablets

[26] Emma Murphy, Ravi Kuber, Graham McAllister, Philip Strain, and Wai Yu. 2008. An empirical investigation into the difficulties experienced by visually impaired Internet users. *Universal Access in the Information Society* 7, 1-2 (2008), 79–91. DOI: http://dx.doi.org/10.1007/s10209-007-0098-4

[27] Earl A. Oliver and Srinivasan Keshav. 2011. An Empirical Approach to Smartphone Energy Level Prediction. In *Proceedings of the 13th International Conference on Ubiquitous Computing (UbiComp '11)*. ACM, New York, NY, USA, 345–354. DOI: http://dx.doi.org/10.1145/2030112.2030159

[28] Ahmad Rahmati, Angela Qian, and Lin Zhong. 2007. Understanding human-battery interaction on mobile phones. In *Proceedings of the 9th international conference on Human computer interaction with mobile devices and services*. ACM, 1378017, 265–272. DOI: http://dx.doi.org/10.1145/1377999.1378017

[29] Samsung. 2019a. Architecture of Tizen. (2019). https://developer.tizen.org/tizen-architecture

[30] Samsung. 2019b. Samsung Ultra Power Saving Mode. (2019). https://www.samsung.com/us/support/answer/ANS00079037/

[31] Samsung. 2019c. Tizen Accessibility. (2019). https://developer.tizen.org/development/ux-guide/basic-interactions/accessibility

[32] Kristen Shinohara and Jacob O. Wobbrock. 2011. In the Shadow of Misperception: Assistive Technology Use and Social Interactions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*. ACM, New York, NY, USA, 705–714. DOI: http://dx.doi.org/10.1145/1978942.1979044

[33] Kristen Shinohara and Jacob O. Wobbrock. 2016. Self-Conscious or Self-Confident? A Diary Study Conceptualizing the Social Accessibility of Assistive Technology. *ACM Trans. Access. Comput.* 8, 2 (2016), 1–31. DOI:http://dx.doi.org/10.1145/2827857

[34] Monsoon Solutions. 2019. Monsoon Power Monitor. (2019). https://www.msoon.com/online-store

[35] TalkBack. 2019. TalkBack: An Open Source Screenreader For Android. (2019). http://google-opensource.blogspot.com/2009/10/talkback-open-source-screenreader-for.html

[36] VoiceOver. 2019. Screen reader from Apple. (2019). https://www.apple.com/accessibility/iphone/vision/

[37] Wikipedia. 2019. Wikipedia: Device driver. (2019). https://en.wikipedia.org/wiki/Device_driver

[38] Jian Xu, Qingqing Cao, Aditya Prakash, Aruna Balasubramanian, and Donald E. Porter. 2017. UIWear: Easily Adapting User Interfaces for Wearable Devices. In *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking (MobiCom '17)*. ACM, New York, NY, USA, 369–382. DOI:http://dx.doi.org/10.1145/3117811.3117819

[39] Jian Xu, Suwen Zhu, Aruna Balasubramanian, Xiaojun Bi, and Roy Shilkrot. 2018. Ultra-Low-Power Mode for Screenless Mobile Interaction. In *31st ACM User Interface Software and Technology Symposium*. ACM. DOI:http://dx.doi.org/10.1145/3242587.3242614