GreenTouch: Transparent energy management for cellular data radios

Shravan Aras
Computer Science
University of Arizona
Email: shravanaras@cs.arizona.edu

Chris Gniady
Computer Science
University of Arizona
Email: gniady@cs.arizona.edu

Abstract

Smartphones come equipped with multiple radios for cellular data communication such as 4G LTE, 3G, and 2G, that offer different bandwidths and power profiles. 4G LTE offers the highest bandwidth and is desired by users as it offers quick response while browsing the Internet, streaming media, or utilizing numerous network aware applications available to users. However, majority of the time this high bandwidth level is unnecessary, and the bandwidth demand can be easily met by 3G radios at a reduced power level. While 2G radios demand even lower power, they do not offer adequate bandwidth to meet the demand of interactive applications; however, the 2G radio may be utilized to provide connectivity when the phone is in the standby mode. To address different demands for bandwidth, we propose GreenTouch, a system that dynamically adapts to the bandwidth demand and system state by switching between 4G LTE, 3G, and 2G with the goal of minimizing delays and maximizing energy efficiency. GreenTouch associates users' behavior to network activity through capturing and correlating user interactions with the touch display. We have used top applications on the Google play store to show the potential of GreenTouch to reduce energy consumption of the radios by 10%, on average, compared to running the applications in the standard Android. This translates to an overall energy savings of 7.5% for the entire smartphone.

ACM Classification Keywords

C.2.1 Network Architecture and Design: Wireless communication; D.4.4 Communications Management: Network communication

General Terms

Design, Experimentation, Measurement, Performance

Author Keywords

LTE; 4G; 3G; Energy saving, Dynamic interface switching; Greentouch

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

UbiComp '16, September 12-16, 2016, Heidelberg, Germany

© 2016 ACM. ISBN 978-1-4503-4461-6/16/09...\$15.00

DOI: <http://dx.doi.org/10.1145/2971648.2971660>

INTRODUCTION

Smartphones are becoming more advanced, more reliable, and offer a greater degree of connectivity. We have come to depend on them for everything from communication, navigation, entertainment to even monitoring our health and keeping us fit. All of this is made possible due to the exponential growth of smartphone performance and features in the past few years. Not only do these phones come equipped with larger and sharper displays but also a wider array of sensors and radios. Naturally all these additional features and components place an ever increasing demand on the battery and drive the need for more energy efficient system and component design.

While screens in modern phones can be very energy hungry as manufacturers are making them bigger and brighter to improve information presentation to users, the communication devices are close behind in energy consumption as we expect higher bandwidths and better connectivity. Primary data communication is accomplished through either WiFi or cellular interfaces, with the current shift towards cellular interfaces as they provide data connectivity on the go, they can offer higher bandwidth than a home WiFi connection, and unlimited data plans further reduce the incentive to rely on a WiFi interface for data communication. Therefore, cellular data interfaces are starting to dominate energy consumption during communication [14], and we need to consider more aggressive optimization techniques to improve their energy efficiency while not compromising performance.

The most popular family of cellular data communication interfaces (radios) today are 3G and 4G. 4G LTE (Long Term Evolution) offers much higher bandwidth capability and is gaining more popularity as providers are expanding their 4G LTE coverage. However, higher bandwidth of 4G LTE radios results in higher power demand as compared to 3G radios. Most network aware applications on mobile phones seldom require bandwidth levels offered by 4G LTE, however providing high bandwidth when an application demands it is critical to maintain a good user experience. In addition, quickly delivering large amounts of data at high bandwidth can improve energy efficiency as users are able to accomplish the task faster and the phone can transition to a lower power state. Subsequently, we are faced with optimization opportunities by trading bandwidth for energy based on

the traffic demand from the application. The challenge lies in accurately matching bandwidth demand from the application to the bandwidth offered by the given radio in a timely manner.

Standard protocol for smartphones is to connect to a 4G LTE network as long as it is available irrespective of the bandwidth demand. This approach maximizes performance and user satisfaction while sacrificing some energy efficiency when serving a low bandwidth traffic using the 4G LTE radio. The bandwidth demand depends on application features and user interaction with specific features. For instance, if a user is running a movie streaming application, one can expect a large surge of network traffic when the play button is clicked. Similarly, if the user is using a non-network intensive application to browse text pages or read messages, the user activity will produce small quantities of network traffic, which can be well accommodated by the 3G radio. Therefore, such user interactions can be successfully exploited to predict bandwidth demand, as have been successfully exploited to manage WiFi interfaces in mobile computers [6].

Different interfaces such as touch screens and cellular radios in smartphones present a diverse set of challenges in detecting user activity as well as applying prediction to radio interfaces that have much longer switching delays due to actually switching between different hardware radios. To address those challenges, we propose GreenTouch to optimize cellular data interfaces. Subsequently, this paper presents the following contributions: (1) analyze and characterize top communication intensive applications in the Android market, (2) design and implement a low overhead monitoring of user touch activity and network bandwidth behavior in smartphones, and (3) design a complete framework to manage energy efficiency of radio interfaces while adapting to dynamically changing network bandwidth demand.

MOTIVATION

The challenge in designing an energy efficient system is to minimize energy consumption without sacrificing performance. In our case, an ideal solution would provide a performance level that closely matches the bandwidth demanded by the applications. This requires timely radio switching to match the bandwidth demand needs without introduction of delays that can degrade the performance and impact users' experience with the phone and the application. To complicate the matter further, the available bandwidth and the switching delays are affected by many factors such as signal strength, network load, and geographical location.

While theoretical bandwidths of different interfaces can be very high, such as 300Mbps for the first revision of LTE [4], the practical bandwidth observed by users can vary significantly. To obtain a more representative bandwidth levels that users may encounter, we have conducted numerous bandwidth measurements for 4G LTE, 3G, and 2G on an AT&T network using Samsung

Figure 1. Radio transition diagram.

Galaxy S2 i727 radios. The measured bandwidth and resulting power demand as well as switching delays and corresponding switching energy are shown in Figure 1. The empirical bandwidth levels are the average of several measurements throughout the day at our lab. As we observe from Figure 1, the available bandwidth levels are much lower than the theoretical levels.

The power levels for the radio interfaces in Figure 1 correspond only to the power demand of the given interface. Power measurements were obtained using National Instruments digital acquisition card (NI DAQ 6230) with the phone connected to a power supply. To measure energy consumption associated with data transmission, we first measured energy consumption of the device with the radios disabled and screen brightness set to minimum to establish a base power level which we found to be 516.92mW. Subsequently, we enabled the individual radios, measured the power, and subtracted the base power to obtain the radio only power levels. Finally, we measured the switching time and energy between various radios by using Android intents to monitor when the network went down and came back up again during transitioning. It must however be noted that, the power consumption of a cellular radio is closely tied with its signal strength, indoors or outdoors usage as well as network congestion, which comes into play during cellular hand-offs between different interfaces as shown in Bartendr [18] and Stratus [2]. For the scope of this paper however we have used average static values, which were taken when the signal strength was full for all 3 interfaces in our lab (indoors) to demonstrate our proposed approach. However, dynamic measurement of bandwidth and signal strength would be critical to further improve the robustness of the proposed solution.

We can draw several conclusions from our measurements. First, there are significant differences in power level between the interfaces in the active state. This provides the opportunity to trade bandwidth level for power demand. Second, 2G offers such low bandwidth that it is not suitable for meeting data transfer demand generated during the interactive episodes in the studied applications. Subsequently, we are left with 4G LTE and 3G radios for any demanding network application that a user is interact-

Figure 2. Percentage of data transmitted in 4G LTE and 3G band.

ing with. The remaining 2G radio may be suitable for maintaining network connectivity when the user is not executing any network aware applications or the phone is in standby mode. Third, the standby power of the interfaces is similar; therefore, we need to properly address the situation of transitioning from an interface already in idle state as it may not offer any significant energy savings while introducing transition delays. Finally, the long transition delays need to be addressed properly to provide efficient transitions and prevent breaks in network connectivity and subsequently impact user experiences with the phone and the application. While it would be desirable to enable one radio while still serving traffic with the other one to hide the transition delay, it is not possible due to the limitations imposed by cellular networks and current technology.

Application Characteristics

While the above observations show that there are both potential opportunities and challenges in improving energy efficiency of cellular data communication, it is critical to understand the application behavior with respect to the bandwidth and traffic patterns resulting from the interactions. We selected representative applications from the top ten applications on the Google Play store in each of the following categories: Video streaming (YouTube, Netflix, Hulu), Music streaming (Spotify, Tunein, Pandora), News (Fox News, USA Today), and Social Media (Facebook, Twitter). These categories were chosen to broadly classify popular user activity on smartphones and understand the behavior of data communication in such application types.

Figure 2 shows the data transfer breakdown according to the bandwidth demand in the studied applications. To provide unrestricted access to bandwidth these experiments were conducted using a dedicated WiFi connection. This guaranteed that the application throughput was not throttled due to congestion in cellular networks. Multiple runs from each application were then aggregated together and data transmitted in each bandwidth

Figure 3. Fraction of time attributed to 4G LTE, 3G, and Idle radio states in the studied applications.

level was normalized to the total data transmissions in each application. Any data that was transmitted at or below 3G bandwidth was classified as 3G and data transmitted above 3G bandwidth was classified as 4G LTE. We observe that applications can take advantage of the higher bandwidth and on average 51% of data is transmitted in 4G LTE bandwidth range. Transmitting the data at high bandwidth allows faster transfer completion and a potential for the interface to transition to idle and conserve energy. Furthermore, faster data transmissions can improve user experience as users may encounter shorter delays. Therefore, it is critical to offer the 4G LTE bandwidth when an application demands it, but we also observe that there is significant amount of data transferred in 3G. We see this as a potential opportunity to save energy.

To contrast Figure 2, Figure 3 shows the amount of time each radio interface would be on to account for data transmissions in the given state, either 3G or LTE, given oracle based switching based on the future traffic arrival. It is important to note that the active times for 4G LTE and 3G shown in figure 3 also include periods when the interface is active but temporarily not transmitting any data. Finally, when no data transfers occur during the timeout period, the radio transitions to the idle state and this time is shown for both radios as Idle time in Figure 3. We combine both 4G LTE and 3G idle times since there is little difference in idle power and any optimization during the idle time would not yield any benefits.

It is evident from Figure 3 that applications need to spend very small amount of their total execution time requiring 4G LTE radio, about 1% on average. Large data transfers can be quickly and efficiently transmitted at LTE bandwidth levels and therefore, it is critical to provide 4G LTE bandwidth when needed. However, 99% of the time the applications require low bandwidth to either maintain buffer after initial fill in video or music streaming applications or load low bandwidth context such as text in news or social media applications.

Figure 4. Flowchart of the GreenTouch system.

This low bandwidth demand gives us a great opportunity for improving energy efficiency by utilizing the 3G radios for a majority of the time. This contrast can be best observed in Tunein in which the 0.7% of time spent in 4G LTE bandwidth level accounts for 94% of data transferred. Subsequently, we argue that providing applications with the 3G radio only would lead to noticeable transmission delays as compared to 4G LTE, while using the 4G LTE radio for all transfer is inefficient. Hence, the challenge is to conserve energy while providing the high bandwidth when needed.

DESIGN

To address the issue of energy efficiency in applications we approach the optimizations in three mostly independent steps that decide: (1) What radio would best accommodate the application startup with respect to delays and available bandwidth; (2) When to switch to high bandwidth 4G LTE radio so the arriving traffic is accommodated without excessive delays; (3) Finally, when to switch from high bandwidth 4G LTE radio to 3G radio when demand for bandwidth drops. The radio adaptation decisions of the three steps above, during the application execution, are summarized in the decision flow diagram in Figure 4, which will guide the design and execution of the proposed GreenTouch framework.

Application Initialization

In this paper, we rely on 2G radio to provide data connectivity or serve low bandwidth traffic such as push notifications. While the idle power for the interfaces is similar, long periods that the phones spend in sleep state can significantly impact battery life even if we are considering a difference of 20mW & 40mW with 3G and LTE idle power respectively. Since network providers are looking to eliminate 2G bands to improve their spectrum utilization, future phones may not have the 2G low power

radio. In that case, we believe that there will always be multiple radios to maintain backward compatibility and in future implementation of the proposed mechanisms, the 2G radio can be replaced by the lowest power alternative such as 3G or any other future technology for maintaining connectivity.

To properly select the low power radio, GreenTouch monitors system activities, and every time the application is started in the system, GreenTouch mechanisms classify the application as network aware or not. Non network aware applications can be quickly identified by the permissions requested from the Android system (by checking `/data/system/packages.xml`), as it indicates whenever the application is using network or not. Any application that is network aware is monitored by GreenTouch and if bandwidth is very low, such as with background weather updates, the application is classified as 2G bandwidth level. This is a very straightforward task, and we leave low bandwidth applications out of the focus of this paper. On the other hand, for applications whose bandwidth requirements exceed that of the 2G band GreenTouch needs to decide if initial application demand can be satisfied by 3G or LTE bandwidth as shown in Figure 4.

Subsequently, we propose a simple predictor that monitors application bandwidth demand to select between 2G, 3G, and LTE radios upon application start. The predictor is a simple three state mechanism that dictates what radio to switch to when the application starts. If an unknown application is started and the network permissions for the applications are set, GreenTouch initializes the radio to 3G to balance the delay and energy trade-offs. If application does not show any network activity above 2G, the application is marked as not network intensive and future execution of the application will not trigger a radio switch from 2G radio, unless a different network activity is detected at a later time.

Once LTE level traffic in an application is detected, we need to decide if we want to start the application in future execution with 3G or LTE radio on. To classify applications that have 3G and LTE level bandwidth, GreenTouch uses an energy-delay² metric to compare starting the application in 3G and LTE. As this paper focuses on preserving user experience we use delay² as it puts more weight on performance. The calculation and comparison of energy delay is made between application start and the first LTE level traffic. Once the benefit comparison of starting the application with the given radio is performed, the application is labeled with the radio that results in the lowest energy-delay² product. The monitoring continues during every application start and if the energy-delay² product changes the application is relabeled with the corresponding radio.

Anticipating LTE Demand

Once we have determined the desired radio to start the application in, we need to continue matching the application bandwidth demand throughout application execu-

tion. A reactive approach that switches to LTE once the 3G bandwidth is saturated is simple, but it would expose users to switching delays (1.6 seconds), thus degrading the user experience. Therefore, we need to design mechanisms that match the bandwidth demand from the application while minimizing delays exposed to the users. Fortunately, previous research has shown that keyboard and mouse interaction can be exploited to predict application behavior with respect to network activity [6]. While smartphone interactions use touch to control the device, the general idea of user interaction invoking network activity still applies. However, there are some significant challenges that need to be addressed. First, touch based interaction may be significantly more complex as users drag, rotate, use multi-touch, etc., to invoke activities in smartphones. Second, significantly larger switching overheads (of the order of seconds when compared to milliseconds in the case of WiFi) and multiple radios with corresponding idle and active states make it more challenging to predict the best state under dynamic conditions.

Characterizing Touch Based Interactions

In order to accurately predict bandwidth demand, GreenTouch must uniquely correlate various user interactions including complicated gestures with the network activity they might generate. In order to achieve this, we exploit a fundamental property of the Android framework; all components capable of processing user input, like buttons, progress bars, text boxes, etc., are children of the *View* class. We refer to these components simply as UI-components for the remainder of this paper. Each of these UI-components has an ID associated with them which are declared in their application's layout files. We use these IDs as our starting point for the construction of a unique event ID value.

Each UI-component is assigned an ID, which is declared by the application developer in layout files during development as part of the standard Android application development process. Along with UI-component IDs we also make use of motion IDs, which are Android's way of associating numerical values to various touch events like drag, up, down, move, scroll, etc. While UI-component IDs help us identify the container on which the user first touched, motion IDs enable us to track the users finger on screen as he performs a touch gesture.

Ui-components are often stacked in a hierarchical manner, and when a user interacts with any node in this hierarchy the events are propagated up the tree all the way to the root node. We made use of this property and instrumented the Android framework code to record both the UI-component ID and the motion ID. The instrumentation used to record the IDs as the control flow travels up to the root is negligible as it needs only to record the motion IDs which are simple integers. In order to generate the event ID, we simply aggregate all the motion ID's and append them to the child UI-component ID which initiated the user interaction.

Figure 5. Event anatomy for Energy-Delay² calculation.

Calculating Energy-Delay² Metric

As shown in Figure 5, user interaction at time T_1 triggers the sequence of events resulting in two potential scenarios: (1) serving traffic in 3G; or (2) switching the radio to LTE and serving the traffic in LTE. The application initiates the network activity and the traffic arrives at T_2 . T_{idle} gives us the opportunity to potentially switch to LTE radio, hide the delays, or wait for the traffic to arrive in 3G mode. Switching to LTE will result in switching energy being consumed (E_{switch}) and before any traffic can be serviced in the LTE mode. In addition, if we switch too early as shown in Figure 5 there is overhead of consuming the energy in LTE active mode waiting for the network traffic to arrive (T'). GreenTouch also needs to consider the exposed delays when the demand for network traffic arrives before the radio switch completed. This scenario is not illustrated due to space constraints; however, the calculations are similar. Finally, GreenTouch needs to calculate the amount of time to serve the traffic in each mode. T_{LTE_ACTIVE} and T_{3G_ACTIVE} are calculated by dividing the amount of data transmitted by the average observed bandwidth in each mode.

The resulting energy to complete the activity due to the interaction for 3G mode is: $E_{3G} = E_{3G_idle} + \Gamma_{3G} + E_{3G_active}$. Γ_{3G} is energy required to transition the interface from idle to active, this energy is negligibly small compared to other energies involved in servicing the event (and not shown in Figure 5 for clarity), but GreenTouch includes it in calculations for completeness. The resulting execution time of the given user interaction: $D_{3G} = T_{idle} + \tau_{3G} + T_{3G_ACTIVE}$. τ_{3G} is the amount of time it takes to transition the interface from idle to active, which is also negligible, but included in the calculation by GreenTouch mechanisms.

Similarly, GreenTouch calculates the energy E_{LTE} and delay D_{LTE} for LTE. The resulting energy is: $E_{LTE} = E_{switch} + E' + E_{LTE_active}$. E' is energy consumed during the time T' when the interface is waiting in LTE active mode or the network activity to arrive. The resulting

execution time is: $D_{LTE} = T_{switch} + T' + T_{LTE_active}$. Finally, the corresponding ED^2 are calculated and compared to evaluate which radio is most beneficial for the given set of events. While we only presented and discussed one scenario in Figure 5, GreenTouch performs calculations of ED^2 based on the current radio state, event timing, and resulting network traffic.

Touch Based Prediction of LTE Bandwidth

In order to predict the occurrence of LTE bandwidth, we took inspiration from classic branch prediction algorithms used in modern day processors. Our approach is divided into two parts: the training stage - which associates user interactions with bandwidth demand, and prediction stage - which uses historic information to predict when to switch to LTE. In order to carry out these steps we maintain a system wide hash table with entries indexed by the key that is a combination of the application name and event ID (com.example.myapp:video01). Each entry in the table contains a simple 4-bit predictor that is updated continuously based on a past behavior. We have considered several predictors and the 4-bit predictor offered the best stability. In order to reduce the memory footprint of our hash table we make use of the observation that there are far less interactions which lead to LTE activity when compared to 3G activity as seen in Figure 3 and thus only record those interactions.

The table lookup is performed when GreenTouch encounters a user-interaction, as illustrated in Figure 4. The interaction description is first converted into a key and the hash table lookup is performed. Lookup miss can indicate two possibilities: either the interaction has never been seen before or the interaction does not result in LTE level bandwidth demand. The default action when the lookup in the table fails that addresses both scenarios is to serve the forthcoming traffic in 3G and switching to LTE on demand once a high bandwidth demand is detected. Subsequently, all training occurs in 3G mode with on demand switch once 3G bandwidth is saturated. Some delays during training may be exposed to users; however, this only occurs for the first time an interaction is encountered in an application, and with a small number of LTE bandwidth demanding interactions the impact on user perceived delays is limited, and will only occur during the first encounter of a given interaction for that application execution. Due to its small size, the table is maintained across application executions which will limit or eliminate impact of training in future executions. Further these keys model how an application reacts to network usage, and hence can be shared between different users using the same application to enable a cold start. Moreover as GreenTouch relies on component ID's declared by application developers in layout files, the contents of the hash table are not invalidated at every application update but only when these change, a task that is undertaken rarely.

On the other hand, if GreenTouch encounters a match in the hash table, it uses the 4-bit counter to predict

and select between LTE and 3G radios. Once the radio is selected GreenTouch continues to monitor bandwidth levels to train and update predictions. The training is continuous for all interactions and the table updates only occur if the interaction has a potential for generating LTE level traffic demand. If the arriving traffic is better served in LTE according to the energy-delay², the 4-bit counter is incremented, and decremented otherwise. The monitoring continues as some applications may show varying behavior based on the data the user is accessing, such as low or high resolution video, or a server and network may encounter congestion and slow down the traffic.

Predicting 3G Bandwidth

To complete the design of GreenTouch we also need to consider switching from LTE to 3G, once the demand for high bandwidth subsides, to improve energy efficiency. A simple approach is to rely on a timeout based mechanism to switch from LTE to 3G. Using the switching overheads shown in Figure 1, we calculated the break-even time (serving in LTE vs switching from LTE to 3G and back again to LTE) to be 121 seconds. The break-even time is then used as a timeout value in the transition mechanisms. While timeout based mechanisms are simple and perform adequately well, they waste a considerable amount of energy waiting for the timer to expire. Furthermore, the large timeout interval and the fact that applications spend majority of their time well within the 3G bandwidth range (as shown in figure 3), motivated us to use a dynamic approach to switch down from LTE to 3G.

We propose a simple modification that augments the prediction table discussed in section 3.2.3 with an additional 1-bit predictor we call H.down to indicate if GreenTouch should transition back to 3G immediately after the LTE bandwidth subsides. In order to determine the value of H.down during our training stage, GreenTouch performs an energy-delay² comparison of keeping the device in LTE or 3G state, following the LTE activity. If the analysis indicates that it is beneficial to leave it in LTE state then we set the field to 1, else we set it to 0. The selection of the simple 1-bit predictor was motivated by the observation that the majority of time the LTE bandwidth subsides, it is followed by long periods of low bandwidth traffic. This behavior follows from recommendation that the traffic in mobile devices should be served in bursts [1].

The prediction remains simple and efficient since upon a touch interaction that predicts the arrival of LTE bandwidth level traffic, GreenTouch only needs to read one more prediction counter to decide how to react to bandwidth demand subsiding to 3G levels. If GreenTouch does not find an entry in the hash table and the LTE switch was required (eg. the event has not been observed before), GreenTouch simply falls back to the timeout based mechanism, switching back down to 3G after 121 seconds. We also use a timeout based mechanism as a

backup in case GreenTouch wrongly predicts to stay in LTE and the LTE bandwidth traffic level does not arrive. The collection of these basic mechanism completes the design of GreenTouch mechanisms.

RESULTS

In order to collect application traces, we ran each media application for an average of 10 hours and 3 hours for each news and social application. These applications were run in isolation, generating event IDs as described in Section 3.2.1 while data from */proc* file system was used to calculate bandwidth consumption of the application.

Prediction of LTE Bandwidth

Figure 6 shows energy distribution between radios and their states for the applications, normalized to LTE radio for all traffic. Energy consumption for each application and mechanism is divided into (1) active energy of the LTE radio (LTE), (2) active energy of the 3G radio (3G), (3) combined idle energy of 3G and LTE radios (Idle), and (4) switching energy between 3G and LTE radios (Switching). Due to the small contribution of individual 3G and LTE idle and switching energies to the overall energy consumption of the application, they were lumped together for clarity in Figure 6. Standard Android systems will use only the LTE radio, if available, to serve all traffic during application execution; therefore, we use it as a reference for studied mechanisms (shown as LTE). Alternatively, one may disable the LTE radio and use only 3G radio to serve all of the traffic (shown as 3G). While such an approach reduces the overall energy by 11.5%, it can potentially introduce delays during high bandwidth traffic transfers, delaying users, thus impacting the user experience. Keeping the smartphone only in 3G band also sets a higher bound on potential energy savings we can achieve, if we overlook resulting delays.

Oracle mechanisms, on the other hand, are able to see the future and switch from 3G to LTE radio just before the traffic arrives and back to 3G once the traffic level drops to the 3G level. Oracle mechanisms are able to reduce energy consumption by 10.4%, while preventing any delays from being exposed to the users, and show the optimal behavior that should be targeted by a predictor. The next mechanism we evaluate is purely reactive and switches to the LTE radio when the 3G radio bandwidth saturates. To focus our comparison on minimizing the transition delays from 3G to LTE, all switching mechanisms use oracle to switch from LTE to 3G once traffic falls below LTE level. While the energy savings of reactive mechanisms is comparable to Oracle, the on demand switching exposes delays when users are waiting for high bandwidth data transfers. Finally, we consider our proposed GreenTouch (abbreviated as GT in the graphs) predictor which also matches the oracle behavior very closely, and is able to reduce energy consumption of the cellular data radios during the application execution by 10%.

It is interesting to note that both Netflix and Fox News spent 39% and 24% of their energy in the LTE band. This was primarily due to the way in which these applications served video. Contrary to applications like YouTube and Hulu which have LTE burst at the video start to buffer video while maintaining buffer at the 3G bandwidth level, Netflix and FOX News initially buffer and maintain the video buffer in short LTE bandwidth bursts. Subsequently, the majority of the traffic is served in LTE, and the LTE radio is a significant energy consumer. In the case of Netflix, our observation confirms previously observed behavior [11]. While Netflix can adopt to lower bandwidth by degrading the video stream, streaming FOX News in 3G takes more energy due to transfer delays and we can see a small increase in energy consumption.

While energy savings are comparable between reactive and GreenTouch mechanisms, the goal of this research is to minimize performance degradation caused by energy optimization. Figure 7 presents delays exposed to the user when serving traffic that demands LTE bandwidth. The delays are normalized to the number of LTE bandwidth transfers in the application. The delays were further separated into delays due to transmissions, when the smartphone is transmitting at 3G instead of LTE, and switching delays that are exposed to the user during an interface switch from 3G to LTE. In addition, GreenTouch switching delays were separated into regular switching delays, and switching delays during initial training that are equivalent to reactive approach as GreenTouch has never seen the given interaction and is switching reactively.

If we consider execution of the entire application using 3G radio only, the transfer delays exposed to the user are prohibitive and average 12 seconds for LTE level data transfer. The on demand approach waits for the bandwidth to saturate the 3G radio after which the radio is switched, exposing the entire switching delay of 1.59 seconds (Figure 1) to the user. There are no transmission delays for the reactive approach, since as soon as the bandwidth is saturated the switch is initiated and the rest of the traffic is served through LTE radio. The entire switching delay is exposed, resulting in the possibility that it would be quicker to finish transferring the data through the 3G interface when the amount of data is relatively small.

GreenTouch addresses both of those issues. First, it triggers the transition of the radio upon an interaction that results in the prediction of LTE level bandwidth. The earlier the prediction is made the more time GreenTouch has to overlap the switching time before the traffic arrives. On average, the delay exposed to the user by GreenTouch is 0.5 seconds which is 69% shorter than delays of the reactive approach, if we ignore training. Including training, increases the average delay to 0.8 seconds. The training delays are a full 1.59 seconds since GreenTouch has never seen the interaction and is un-

Figure 6. Normalized energy distribution.

Figure 7. Average amount of delays exposed to users.

able to initiate the switch earlier. Since popular applications are executed frequently on smartphones, we expect the initial training to be insignificant in real life scenarios, however we show the impact of initial training for completeness. Second, GreenTouch predicts that in some scenarios paying switching overhead may not benefit the energy and the resulting delays and serves traffic in 3G. This is the case with Spotify which bursts into LTE bandwidth but the data transferred is small and energy-delay² comparison suggests using 3G radio optimizes energy and minimizes user encounter delays.

To minimize delays due to radio switching during data transfers, GreenTouch is very aggressive in predicting LTE behavior. We have investigated several predictor configurations and settled on a 4-bit predictor that is fully preloaded towards LTE prediction to aggressively predict potential LTE traffic generating interactions. Since, GreenTouch is very aggressive in predicting LTE events, there is the possibility that there will be incorrect switches. Figure 8 presents switching activity for GreenTouch and the given application. As we observe, GreenTouch is very aggressive and switches ahead of time for 83% of LTE events on average. During training GreenTouch uses a reactive approach and all misses from the Figure 8 are due to initial training. As we mentioned before, initial training will be insignificant

Figure 8. Switching prediction accuracy for GreenTouch.

in common applications as they are executed frequently and initial training occurs during the first execution or interaction type only.

The unnecessary switches shown in Figure 8 are due to GreenTouch predicting the arrival of LTE bandwidth traffic and switching the radio to LTE, when the actual traffic did not require LTE bandwidth. Upon careful examination, we discovered that unnecessary switches

Figure 9. Extension in the application execution time.

were still followed by the same traffic pattern as the LTE events, however the bandwidth failed to burst into LTE levels. This lower than expected traffic can be attributed to network or server side throttling, either due to higher server load, network load, or decrease in signal quality of the cellular data connection. Hence the same event can at times produce both traffic in LTE and 3G band making it challenging to predict the upcoming bandwidth perfectly.

The unnecessary switches can introduce additional delays as the traffic could have been served by 3G radio without encountering switching delay. However, we argue that their impact on delays and thus user satisfaction is far less as compared to missed switches. The reason being, a missed switch would result in a delay penalty of 1.59 seconds as GreenTouch would reactively switch upon 3G bandwidth saturation. On the other hand, an unnecessary switch would take place before the actual arrival of the traffic, partially overlapping switching overhead. Figure 9 shows total delays exposed to the users normalized to the total application execution time. We observed that on average GreenTouch hides 44% of the total delays when compared to the reactive mechanism even after taking into account the delays caused by unnecessary switches. Furthermore, on an average, GreenTouch increased the application running time only by 0.05% when compared to 0.10% in case of running the applications under reactive mechanisms. Subsequently, the GreenTouch's switching aggressiveness results in energy savings comparable to oracle mechanisms while significantly reducing delays exposed by the reactive mechanisms.

Prediction of 3G Bandwidth

In the previous section we assumed that GreenTouch uses oracle to switch from LTE to 3G radio when the LTE bandwidth demand ends. Figure 10 presents the impact of replacing the oracle LTE to 3G switching by static timeout (GTS) and finally the full GreenTouch mechanisms that use interactions to predict when to

Figure 10. Impact of predictive LTE to 3G switching on energy.

switch from LTE to 3G radios. The figure presents additional energy savings as compared to the GreenTouch mechanisms with oracle LTE to 3G switching. In addition, we present LTE as a reference for comparison to Figure 6. For the static timeout based transition we set the timeout to breakeven time, which we calculated to be 121 seconds.

The timeout based prediction, usually suffers from higher energy dissipation due to waiting in high power state during the timeout intervals. On average, the energy savings by the GTS are reduced by 5% which is half of the gap between the LTE and GreenTouch with oracle LTE to 3G switching. GTS suffers most in Pandora, since Pandora has distributed LTE burst due to pre-buffering of songs at the end of the previous song. This is a very regular behavior with an interval comparable to the timeout interval as many songs last between 3 to 5 minutes. Similar behavior is observed in YouTube. Fox News or Tunein, on the other hand, exhibit longer bursts of LTE activity in 3G bandwidth level and as a result GTS is comparable to optimal switching.

GreenTouch uses user interactions to classify how the LTE traffic will behave and switches immediately after the bursts of LTE activity to the 3G radio, once the prediction dictates to do so. Again, the interaction based classification offers us significant benefits and GreenTouch consumes only 0.8% more energy, on average, as compared to GreenTouch with oracle LTE to 3G switching. Subsequently, the complete GreenTouch mechanisms save 9.5% of energy as compared to LTE, and closely match the energy savings of the oracle predictor. The impact on delays is insignificant as the switches are occurring when the users are not interacting with the application.

We argue that switching from LTE to 3G does not incur any additional delays since the data has already been buffered and the buffers can easily absorb the switching delays. Table 1 presents the average measured buffer

Application	Size[s]
YouTube	115
Hulu	478
Netflix	81
Spotify	28
Tunein	Variable
Pandora	10

Table 1. Buffer sizes for media applications with non-interactive activity.

sizes for streaming applications that maintain and manage the buffers while users are not interacting but watching or listening to the streaming media. To measure buffer sizes, we executed streaming applications. Once the network interface became idle, indicating the initial buffer fill was completed, we severed data connection and measured the amount of time for which the video or audio continued to play, indicating the buffer size. Significant buffer sizes in the studied applications indicate that buffering is capable of fully overlapping the switching delays in our system. This is expected as streaming mobile applications need to accommodate fluctuating network capacity and cell handovers. In the case of Tunein, buffer size varies as it tries to buffer the entire data stream, e.g.; if a user plays a 60-minute-long podcast, then Tunein would buffer all 60 minutes at once.

RELATED WORK

There have been a large number of studies revolving around optimizing energy for smartphones which have ranged from optimizing the display component [17] to radios or even monitoring [14] and reporting possible software bugs [15, 13] that can cause energy drain. The idea of switching between multiple interfaces in order to conserve power for mobile systems has previously been explored in WISE [12], where energy was conserved by switching between 3G and WiFi. Similarly Trevor Pering et. al. outline a tool, Coolspots [16], wherein they conserved energy by switching between WiFi and Bluetooth. Besides switching between multiple interfaces, switching between different states for WiFi has also been explored in STPM [3] which switches between PSM and CAM. However the proactive nature of STPM revolved around user defined hints, which requires modifications to applications. As user interactions can be cleverly leveraged to predict bandwidth demand, [6] uses them to switch between CAM and PSM before bandwidth arrival. Similarly, Shuo Deng et. al. [7] used a statistical prediction model to switch between active and idle mode for the LTE radio in case of background applications.

In order to understand the impact of modern day, high throughput radios Huanh et al. created a power model for 4G LTE radios and show that they are much less energy efficient when compared to 3G and WiFi [8]. In their study they use traces from more than 3000 users to study the network throughput offered by several interfaces (LTE, WiFi, HSDPA1, etc.) and their implications on energy consumption. Furthermore, Qiang Xu

et. al. [19] conducted a comprehensive user study, where they analyzed the type of percentage of traffic volume associated with various applications. As the user does continuously interact with their phone, studies to understand screen off traffic using 3G/4G networks has been explored in the past by [9]. While there is a power difference between various radios and their states, tail time for these devices also contributes to a considerable amount of energy consumption. While most solutions rely on timeouts, RadioJockey [5] uses system calls to predict the possible start of an idle period, forcing the interface to idle if successful. As the demand for video streaming on mobile devices increases, studies to cache video content more efficiently like GreenTube [10] have been proposed. While a plethora of research in the past has been focused on switching between interfaces for background applications, tail time optimization, cache optimization and screen optimization, to the best of our knowledge user interactions to predict and then switch between 2G, 3G and LTE interfaces for *foreground interactive applications* has not been done before.

CONCLUSIONS

A wide selection of radio interfaces in smartphones allows us to be flexible in selecting the interface for the given network traffic. Switching network interfaces introduces delays into the application execution and makes it challenging to maintain performance while reducing energy consumption. The proposed GreenTouch is able to successfully navigate changing user behavior and the resulting bandwidth, preserving users' expectation of high performance while improving the battery life. GreenTouch successfully exploited the idea that user interactions can indicate a demand level for bandwidth and use the interactions as hints to manage radio selection and timely switching between radios. We have shown that while timeout or reactive mechanisms are simple, they consume energy waiting for the timeout interval to expire or introduce significant delays to users. GreenTouch is able to proactively switch ahead of the LTE bandwidth traffic's arrival, overlapping switching delays and hiding 50% of delays from the user as compared to the reactive approach. Furthermore, dynamic prediction of the traffic patterns reduces reliance on timeout based LTE to 3G switching and is able to improve the energy on average by 5% as compared to timeout mechanisms. Overall GreenTouch reduces the total energy consumption of radio interfaces by 10% which is about 7.5% when we consider the entire smartphone energy budget. While energy conservation can expose some delays, GreenTouch's proactive approach is able to hide most of the delays and the overall delays encountered by the user are less than 0.05% on average during the entire application execution.

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant No. 1551057

REFERENCES

1. Optimizing downloads for efficient network access <http://developer.android.com/training/efficient-downloads/efficient-network-access.html>.
2. AGGARWAL, B., CHITNIS, P., DEY, A., JAIN, K., NAVDA, V., PADMANABHAN, V. N., RAMJEE, R., SCHULMAN, A., AND SPRING, N. Stratus: energy-efficient mobile communication using cloud support. *ACM SIGCOMM Computer Communication Review* 40, 4 (2010), 477–478.
3. ANAND, M., NIGHTINGALE, E. B., AND FLINN, J. Self-tuning wireless network power management. *Wireless Networks* 11, 4 (July 2005), 451–469.
4. ASTÉLY, D. T. I. A., DAHLMAN, E., FURUSKAR, A., JADING, Y., LINDSTROM, M., AND PARKVALL, S. LTE: the evolution of mobile broadband. *Communications Magazine, IEEE* 47, 4 (2009), 44–51.
5. ATHIVARAPU, P. K., BHAGWAN, R., GUHA, S., NAVDA, V., RAMJEE, R., ARORA, D., PADMANABHAN, V. N., AND VARGHESE, G. Radiojockey: mining program execution to optimize cellular radio usage. In *Proceedings of the 18th annual international conference on Mobile computing and networking* (2012), ACM, pp. 101–112.
6. CRK, I., BI, M., AND GNIADY, C. Interaction-aware energy management for wireless network cards. *ACM SIGMETRICS Performance Evaluation Review* 36, 1 (2008), 371–382.
7. DENG, S., AND BALAKRISHNAN, H. Traffic-aware techniques to reduce 3g/lte wireless energy consumption. In *Proceedings of the 8th international conference on Emerging networking experiments and technologies* (2012), ACM, pp. 181–192.
8. HUANG, J., QIAN, F., GERBER, A., MAO, Z. M., SEN, S., AND SPATSCHECK, O. A close examination of performance and power characteristics of 4g lte networks. In *Proceedings of the 10th international conference on Mobile systems, applications, and services* (2012), ACM, pp. 225–238.
9. HUANG, J., QIAN, F., MAO, Z. M., SEN, S., AND SPATSCHECK, O. Screen-off traffic characterization and optimization in 3g/4g networks. In *Proceedings of the 2012 ACM conference on Internet measurement conference* (2012), ACM, pp. 357–364.
10. LI, X., DONG, M., MA, Z., AND FERNANDES, F. C. Greentube: power optimization for mobile videostreaming via dynamic cache management. In *Proceedings of the 20th ACM international conference on Multimedia* (2012), ACM, pp. 279–288.
11. MARTIN, J., FU, Y., WOURMS, N., AND SHAW, T. Characterizing netflix bandwidth consumption. In *Consumer Communications and Networking Conference (CCNC), 2013 IEEE* (2013), IEEE, pp. 230–235.
12. NAM, M., CHOI, N., SEOK, Y., AND CHOI, Y. Wise: energy-efficient interface selection on vertical handoff between 3g networks and w lans. In *Personal, Indoor and Mobile Radio Communications, 2004. PIMRC 2004. 15th IEEE International Symposium on* (2004), vol. 1, IEEE, pp. 692–698.
13. PATHAK, A., HU, Y. C., AND ZHANG, M. Bootstrapping energy debugging on smartphones: a first look at energy bugs in mobile devices. In *Proceedings of the 10th ACM Workshop on Hot Topics in Networks* (2011), ACM, p. 5.
14. PATHAK, A., HU, Y. C., AND ZHANG, M. Where is the energy spent inside my app?: fine grained energy accounting on smartphones with eprof. In *Proceedings of the 7th ACM european conference on Computer Systems* (2012), ACM, pp. 29–42.
15. PATHAK, A., JINDAL, A., HU, Y. C., AND MIDKIFF, S. P. What is keeping my phone awake?: characterizing and detecting no-sleep energy bugs in smartphone apps. In *Proceedings of the 10th international conference on Mobile systems, applications, and services* (2012), ACM, pp. 267–280.
16. PERING, T., AGARWAL, Y., GUPTA, R., AND WANT, R. Coolspots: reducing the power consumption of wireless mobile devices with multiple radio interfaces. In *Proceedings of the 4th international conference on Mobile systems, applications and services* (2006), ACM, pp. 220–232.
17. RANGANATHAN, P., GEELHOED, E., MANAHAN, M., AND NICHOLAS, K. Energy-aware user interfaces and energy-adaptive displays. *Computer* 39, 3 (2006), 31–38.
18. SCHULMAN, A., NAVDA, V., RAMJEE, R., SPRING, N., DESHPANDE, P., GRUNEWALD, C., JAIN, K., AND PADMANABHAN, V. N. Bartendr: a practical approach to energy-aware cellular data scheduling. In *Proceedings of the sixteenth annual international conference on Mobile computing and networking* (2010), ACM, pp. 85–96.
19. XU, Q., ERMAN, J., GERBER, A., MAO, Z., PANG, J., AND VENKATARAMAN, S. Identifying diverse usage behaviors of smartphone apps. In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference* (2011), ACM, pp. 329–344.