OPTIMIZATION OF SMARTPHONE BATTERY LIFE USING CUSTOM BROWSER

P Amudha Valli, Assistant Professor [1], S P Audline Beena [2], Assistant Professor, M Aishwarya, R Dilli Rani, P Divya Priya [3,4,5]

[1]Department of Computer Science, Sri Muthukumaran Institute of Technology. India
[2]Department of Computer Science, Sri Muthukumaran Institute of Technology. India
[3,4,5]Department of Computer Science, Sri Muthukumaran Institute of Technology. India
[1] amudhabadmanaban@gmail.com
[2] audlinebeena@gmail.com
[3] visitaishu@gmail.com
[4]dillirani94@gmail.com
[5]divyapriya2595@gmail.com

***Abstract*—Smartphone based web browsing wastes a lot of power when downloading webpages due to the special characteristics of the wireless radio interface. In this paper, we identify these special characteristics, and address power consumption issues through two novel techniques. First, we reorganize the computation sequence of the web browser when loading a webpage, so that the web browser can first run the computations that will generate new data transmissions and retrieve these data from the web server. Then, the web browser can put the wireless radio interface into low power state, release the radio resource, and then run the remaining computations. Second, we introduce a practical data mining based method to predict the user reading time of webpages, based on which the smartphone can switch to low power state when the reading time is longer than a threshold. To demonstrate the effectiveness of our energy-aware approaches, we develop a test bed with Android phones on T-Mobile UMTS network. Experimental results show that our approach can reduce the power consumption of smartphone by more than 30 precent's during web browsing. Moreover, our solution can further reduce the webpage loading time and increase the network capacity.**

Keyword- Mobile Computing, Web Browsing, Wireless Communication, Portable Devices

I. INTRODUCTION

Web browsing is one of the most important and commonly used services provided by smartphones. How- ever, the current smartphone web browser wastes a lot of power when downloading webpages. There have been lots of research on optimizing the power usage of smartphones but they only focus on reducing the power consumption of one component such as display. Although some of them also consider the power consumption of the wireless interface most of them focus on the Wi-Fi interface which has different characteristics from the cellular interfaces, such as 3G and 4G LTE, which consume much more power. In UMTS 3G networks and 4G LTE network, multiple timers are used to control the radio resource, and the timeout value for releasing the resource can be more than 15 seconds. Thus, it is possible that the wireless radio interface continues to consume a large amount of power before the timer times out, even when there is no network traffic. One advantage of this approach is that it can reduce the latency of next possible data transmission that arrives before the timer expires, because the connection between the smartphone and the backbone network is still available. Otherwise, the backbone network has to allocate the radio resource again, which will consume more time and power. As a result, simply adjusting the timer may not be a good solution for saving power.

Due to the limited computation capability, when opening a webpage, the current smartphone web browser takes a long time for downloading and processing all objects (e.g., JavaScript) of a webpage. As a result, the data transmissions are distributed along the whole webpage downloading duration, and then the data rate at any instant time is quite low. Although there are many idle times between these data transmissions, each idle time is still smaller than the time out value, and these data transmissions reset the timers again and again before they expire. Consequently, the radio interface is always on and the radio resource cannot be released, which consumes lots of power and decreases the network capacity.

We address the power consumption issue in smartphone based web browsing through two novel techniques. First, we reorganize the computation sequence of the web browser when loading a webpage. There are various computations when loading a webpage such as HTML parsing, JavaScript code execution, image decoding, style formatting, page layout, etc. These computations generally belong to two categories based on whether they will generate new data transmissions from the web server.

2. RELATED WORK

Current smartphone web browser such as Google chrome takes a long time for downloading and processing all objects (e.g., JavaScript) of a webpage. It displays the downloaded content simultaneously it downloading the remaining content. This web browser loads the URL resources and renders the content into the screen. Lot of power gets wasted when downloading webpages. Since the mobile consumes more energy when the internet is enable. An attempt is made in this project to save the battery life by creating the custom browser that will turn off the wireless connection while rendering the contents in the screen.

3. PROPOSED SYSTEM

We will solve the power consumption issue in smart phones by creating our own custom browser. To achieve this, two techniques are used in this project. First, we reorganize the computation sequence of the web browser when loading a webpage. These computations generally belong to two categories based on whether they will generate new data transmissions from the web server.

We want to separate these two types of computations so that the web browser can first run the computations that will generate new data transmissions and retrieve these data. Then, the web browser can put the wireless radio interface into low power state, release the radio resource, and then run the remaining computations. For webpages that have short processing time, we need to predict the user reading time on the webpage after it is downloaded. If this time is longer than a threshold, the radio interface can be put into low power state.

Building a Custom Browser App

In this module, we need to build an App which is having our custom browser. Now a day, web browser is more complex since it has to process various script codes such as JavaScript embedded in HTML documents. Further, it has to process Cascading Style Sheets (CSS), which is used to describe the presentation semantics and the style rules of a webpage such as layout, color and fonts. Document Object Model (DOM) is an interface that allows programs and scripts to update the content, structure and style of HTML documents. After the HTML code has been parsed, the nodes in the DOM tree store the HTML data. After the CSS code has been parsed, the style and layout properties are assigned to these nodes in the DOM tree. Then the web browser can display them on the screen. After the web browser gets the

main HTML page, the data transmission mainly comes from three kinds of sources: HTML, JavaScript, and CSS. In HTML and CSS, content objects such as HTML files, JavaScript files, images, and flashes, are referenced by URLs. Hence, the web browser needs to fetch and add them to the DOM tree. For JavaScript code, it is either transformed to HTML code and then fetches content objects, or it can directly get content objects from the web server.

Cut off the internet connection

We will cut off the 2G/3G connection after downloading the content from the website. We evaluate the power consumption in this section. The power consumption of Smartphone includes the power consumed by the display and system maintenance. Lists the power of the Smartphone in different states, where the DCH state has the highest power. When the Smartphone stays in FACH, it consumes almost the
same amount of power as CPU fully running at IDLE state. When the Smartphone is in IDLE, the power is mainly consumed by screen display and system maintenance. Since our energy-aware approach reduces the time at FACH and DCH states, it can save energy for web browsing. Suppose the reading time is larger than 20 seconds, in the original approach, the Smartphone will stay in FACH for Seconds, but our approach can put the Smartphone into IDLE. Our approach can reduce the power consumption by 35.7 precents. Most of this power saving comes from putting the Smartphone into IDLE during the reading time. With full version benchmark, our approach can reduce the power consumption by 30.9 precents. Different from the mobile version, most of this power saving comes from reducing the data transmission, which is achieved by separating the data transmission computation and layout computation.

Automatic Switch on 2G & 3G

Case 1: After reaching threshold value the internet connection will be switched on automatically.

Based on our collected traces the reading time of a webpage is related to both webpage features and user interests in the webpage. For example, suppose that two news webpage have similar features listed in one is about games and the other is about finance. A user may spend more time on the webpage about games if he has more interests in game than finance. To match the user interest, we need to parse the content of a webpage when opening it to collect the word frequency and classify the content based on keywords like “game”, “finance”, and “weather”. Parsing the content and classifying webpage will consume too much power on computation limited Smartphone. Instead of parsing the webpage content, we use a predefined interest threshold for prediction. If the reading time of a webpage is larger than the interest threshold, it means that the user is interested in this webpage. To improve the prediction accuracy, we only predict the reading time of the webpage that the user is interested in. After a webpage is opened, we will wait until the reading time is larger than the interest threshold. Then we can predict and decide if the Smartphone should switch to IDLE.

Case 2: If you click any hyperlink within the page it will also turns on the internet connection.

IDLE state: The Smartphone does not have any signalling connection with the backbone network, and hence it cannot transmit user data. The radio interface of the Smartphone in IDLE state consumes very little power.

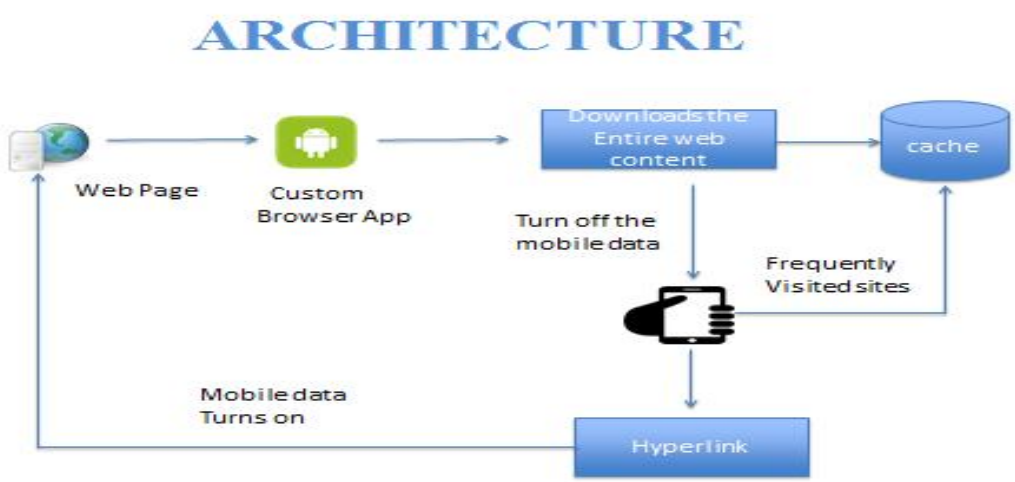
DCH state: The backbone network allocates dedicated transmission channels (uplink and downlink) to the Smartphone, so that the Smartphone can transmit user data and signalling information at high speed.

.

FACH state: Smartphone in this state has no dedicated transmission channel. Hence, it can only transmit user data and signalling information through common shared transmission channels at low speed (up to a few hundred bytes/second).

The state switch component is used to switch the state of the Smartphone from FACH to IDLE. This can be easily implemented if the source code of the radio firmware interface is available to the developers. However, in Android Smartphone the source code of the radio firmware is not available, although the source codes of the applications, framework, and OS are open. Android only provides some interfaces to access the radio firmware through the 3G wireless radio interface called Radio Interface Layer (RIL).

System Architecture



Future Work- Caching

- We can also cache the images, css, js files from the frequently visited sites.
- So that we can save energy as well as mobile data.

We will apply other techniques such as caching and prefetching to save energy. By prefetching and caching future useful data at the right time, wireless data transmissions can be aggregated to save energy. We will also extend our techniques to 4G/LTE network after it is widely deployed in our area.

4. CONCLUSION

We proposed an optimization of smart phones battery life using custom browser in 3G based Smartphone. First, we reorganize the computation sequence for loading webpage so that the web browser can first run the computations that will generate new data transmissions and retrieve these data. Then, the web browser can put the 3G radio interface into IDLE,

release the radio resource, and then run the remaining layout computation. This not only saves power, but also reduces the webpage loading time by removing the computation intensive redraws and reflows. Second, we predict the user reading time on the webpage after it is downloaded. If this time is longer than a threshold, the radio interface can be put into IDLE to save power. Since Smartphone have limited computation capability, we propose a low overhead prediction algorithm based on gradient boosted regression trees. Additionally, our approach can also increase the network capacity, since the radio resource can be released earlier. Experimental results show that our approach can reduce the power consumption of Smartphone by more than 30 precents during web browsing. Moreover, our solution can reduce the webpage loading time by 17 precents, and increase the network capacity by19.6 precents.

5. REFERENCES

[1] J. Flinn and M. Satyanarayanan, "Managing battery lifetime with energy-aware adaptation," ACM Trans. Comput. Syst., pp. 137 179, May 2004.
[2] J. Sorber, N. Banerjee, M. D. Corner, and S. Rollins,"Turducken: Hierarchical power management for mobile devices," in Proc. ACM 3rd Int. Conf. Mobile Syst., Appl., Serv., 2005, pp. 261 274.
[3] M. Dong and L. Zhong, "Chameleon: A color-adaptive web browser for mobile OLED displays," in Proc. ACM 9th Int. Conf. Mobile Syst., Appl., Serv., 2011, pp. 85–98.
[4] E. Shih, P. Bahl, and M. J. Sinclair, "Wake on wireless: An event driven energy saving strategy for battery operated devices," in Proc. ACM 8th Annu. Int. Conf. Mobile Comput. Netw., 2002, pp. 160–171.
[5] F. R. Dogar, P. Steenkiste, and K. Papagiannaki, "Catnap: Exploiting high bandwidth wireless interfaces to save energy for mobile devices," in Proc. ACM 8th Int. Conf. Mobile Syst., Appl., Serv., 2010, pp. 107–122.
[6] E. Rozner, V. Navda, R. Ramjee, and S. Rayanchu, "NAPman: network-assisted power management for wifi devices," in Proc. ACM 8th Int. Conf. Mobile Syst., Appl., Serv., 2010, pp. 91–106.
[7] 3GPP Specification TS 24.008: Mobile Radio Interface Layer 3 Specification; Core Network Protocols; Stage 3, Std., Rev. Rel. 9, 2009. [Online]. Available: http://www.3gpp.org.
[8] N. Thiagarajan, G. Aggarwal, A. Nicoara, D. Boneh, and J. P. Singh, "Who killed my battery?: Analyzing mobile browser energy consumption," in Proc. Int. World-Wide Web Conf., 2012, pp. 41–50.
[9] L. A. Meyerovich and R. Bodik, "Fast and parallel webpage L ayout," in Proc. Int. World-Wide Web Conf., 2010, pp. 711–720.
[10] J. H. Friedman, "Greedy function approximation: A gradien boosting machine cite," Ann. of Statist., pp. 579–588, 1999.smartphones: Ideal vs. reality,"