Energy Consumption on Android Phone Web Browsing in 3G network

Prajakta Kulkarni
Computer Science and Engineering
G. H. Raisoni College of Engineering
Nagpur, India
prajaktakul@gmail.com

Prof. Prachi Jaini
Computer Science and Engineering
G. H. Raisoni College of Engineering
Nagpur, India
Prachi.jaini@raisoni.net

Abstract— In today's era mobile applications have reached great heights. Processing power, available user memory and storage have seemingly increased. Considering all this, a concern about energy consumption is raised as all these require significant amount of power consumption. Also keeping the battery backup of a smartphone in mind, the need to conserve and minimize Energy consumption comes to play. This power consumption is due to the characteristics of wireless radio interface, power consuming display. Hence, in this paper the power consumption reduction will be achieved by addressing those various issues of the wireless radio interface. For this, survey of various techniques is done. This paper helps us to focus and resolve the issues because of which the energy consumption in the smartphone is increasing. Again two methods i.e. Prefetching and caching will also be added for the performance enhancement and reduce the latency of webpage loading.

Keywords- *Web Browser, Mobile Computing, Wireless Communication, Portable devices, prefetching, Reading time Threshold, Wireless radio interface.*

*****

I. INTRODUCTION

Smartphones are becoming a key element of our day-to-day life. We all are connected by the internet throughout the world. The survey says that there are 3% people worldwide who use internet on mobile phones. And some of them use more than one smartphones for their own. Samrtphones have tremendous demand in the market because of its amazing applications. Though smartphones are doing well, there is one issue which must be focused and resolved. And that is, the energy consumption while browsing the web.

For resolving this issue, lot of research has been done on various interfaces like, Wi-Fi, Bluetooth. But these have different characteristics than the wireless radio interface or the cellular interface like 3G, 4G LTE. The cellular interface characteristics if studied, we come to know that these things consume much more power. So, here those special characteristics will be addressed and the issues on them will be identified.

To utilize the limited radio resource of the backbone network efficiently, the 3G Radio Resource Control (RRC) protocol defines the following three states for smartphones to control their wireless radio interface.

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

DCH state: This is the dedicated channel state in which the backbone network allocates dedicated transmission channels (uplink and downlink) to the smartphone, so that the smartphone can transmit user data and signaling information at high speed. This consumes highest level of power.

FACH state: This is forward access channel state in which the smartphone no dedicated transmission channel. Hence, it can only transmit user data and signaling information through common shared transmission channels at low speed (up to a few hundred bytes/second). Data transmission in the FACH state requires about half of the power in the DCH state.

When the smartphone wants to transmit the data, it has to be switched from IDLE to the DCH state. It first establishes the signaling channel and then obtains the dedicated channel for the transmission. This process requires lot of message exchanges. To determine when to release the dedicated transmission channels allocated to the smartphone, the backbone network uses timers.

Smartphones can't do many computations. While loading a webpage there are various computations such as HTML parsing, JavaScript code execution, image decoding, style formatting, page layout, etc. Based on whether they will generate new data transmissions from the web server, these computations generally belong to two categories i.e. Data transmission computation and layout computation. These two computations are mixed in current web browsing.

Again there are two types of web pages based on the processing time i.e. long processing time and short processing time. The web pages which require the long processing time, should separate the computations to reduce the power consumption. And those webpages which require the short processing time, the reading time will be predicted so that the next data object should be prefetched.

Caching and prefetching techniques can also be used for reduction of traffic overhead. These techniques will help the browser to prefetch the data objects distributed on the webpages in the local cache. And when user is accessing the webpage he will not have to wait for loading the data objects as it will be already prefetched in the cache. This will improve the user experience and reduce latency.

II. Literature Survey

Much research have been done to resolve the issue of power consumption. But many of the researchers have focused on the issue of power consumption on display. Some researchers have focused the interfaces like wifi or Bluetooth. The wireless radio interface such as 3G, 4G LTE have different characteristics from other interfaces and they consume much more power. To resolve this issue, the two techniques are used [1] i.e. Reorganization of computing sequence for those web pages which require long processing time and prediction of user reading time for the web pages which require short processing time. These techniques help to reduce the power consumption, delay and to increase the network capacity. If the threshold value is not exceeded the energy consumption will be more due to state switching. In the reorganization of computing sequence, the two computations are separated i.e. data transmission computation and the layout computation. The main focus is on the data transmission computation. So, those data is parsed which generate the new data transmissions then the radio resources are kept into the low power state. And then the remaining layout computations are done. In this way, it enhances the performance of the smartphone in terms of power consumption, delay and network capacity.

There are various prefetching and caching techniques to reduce the latency of webpage loading. But most of them rely on the history for prefetching the objects of webpages and it can increase the traffic. So, congestion problem occurs. An ideal pre-fetching caching scheme is a system that is able to predict the next request or number of next requests and pre-fetch those into the cache .The pre-fetched objects are stored in the local cache which will reduce the latency**.** So, the study of various algorithms is done for the best scheme of prefetching and caching .

The prefetching technique has two main components: The prediction engine and the prefetching engine. A prediction algorithm will be run by prediction engine to predict the next user’s request. The prefetching engine decide to prefetch them or not depending on some conditions like available bandwidth .Each engine can work at any element of the web architecture. The predictions (PD) i.e. the number of objects which are predicted by the prediction engine and the prefetch request (PR) represents the number of objects prefetched. Considering these things, various prediction based techniques[5] The Prediction Algorithm based on Maximum-Weight-Matrix which is used to train the machine (caching system) to learn the request pattern from the client. The learning process is by prediction on next request following the current one. Then, in the Dynamic web prefetching subsequent links are pre-fetched only if bandwidth usage of existing network is less than a predefined threshold. Likewise other algorithms are proposed for efficient prefetching and caching.

There is one of the best techniques for optimal prefetching [2] which can be done. The algorithm based on the current content of the web documents is used so that there is no requirement of maintaining past history of the users and is also beneficial for first retrieval of access of web resources. The algorithm very intelligently fetches only those hyperlinks which are required by the user. For this, the two modules are used i.e. prefetch module and prediction engine. The prefetch module consists of two parts, one is the extraction module and the another one is prefetch module. The prefetch module receives the request from client and serves the request to the client. As per the request of the client it prefetches the hyperlinks according to the list of hyperlinks supplied by the prediction module and puts them into the prefetch cache. The extraction module collects the keywords from the user specified list of keywords. Then the list of all hyperlinks which contain the user specified keywords is made by this module. Prediction module compares the keywords in the user specified keyword-list with the list of keywords describing the hyperlinks on the actual HTML page using the optimal perfecting algorithm and sends the list of the hyperlinks to the prefetch module.

OLED (Organic Light emitting diode) technology is used on many mobile phone displays. But the OLED display are not so efficient to display bright colors and it consumes different power for showing different colors. For reduction of power consumption on displaying webpages, Chamelon [3] (A Color-Adaptive Web Browser for Mobile OLED Displays) technique is used. This web browser renders webpages with power-optimized color schemes under the constraints which are user-supplied. Chameleon gives user to choose the preference for color transformation using 4 different algorithms which gives 4 formats of webpages as dark, green, arbitrary and inversion. This technique is able to reduce average system power consumption for web browsing by 41% and is able to reduce display power consumption by 64% without introducing any noticeable delay. But, it only works on energy consumption of display only.

As we know, mobile phones have low computation power, to reduce the computational burden, some techniques should be used to reduce the computational burden. Hence, Bo Zhao has proposed the technique which shifts the burden of computation from smartphones to VMP (Virtual machine based proxy) [4]. A proxy is added between the web server and the smartphones, and a new client is added on the side of the smartphone which interacts with the proxy. Hence the request is sent to the proxy instead of sending it to the web server, and proxy sends that request to the web server. The proxy can reduce the resource consumption by running different optimization techniques at the smartphone. It reduces the delay by more than 80% and reduces the power consumption during web browsing by more than 45%. Although this approach has low cost, public cloud may not meet the security level.

Today, as the use of smartphone has increased a lot, the reason behind that is the excellent applications. The applications of android are also playing a big roleon energy consumption. So, developing the applications which give the excellent service in minimum energy consumption is really very necessary today. Considering this, specific coding practices are proposed [9] which will help the developer to lower down the energy consumption of smartphone and improve the usability of their applications. In this paper some tools are proposed which will help the developers to gain insight into energy usage patterns of applications. Those techniques are, Cycle- accurate simulators, power monitors, statistical based measurement techniques, Program analyzers. The investigation is done on the small scale which included the topic of optimizing the memory usage, sending of HTTP packets, field access, array length and static invocations. This investigation helped the developers and based on this insight developers could bundle the several small HTTP bundles into larger ones for energy getting more efficiency.

III. ALGORITHM

Algorithm For Optimal Prefetching

Input:
1) The list of “n‟ hyperlinks of the current web page.
2) Minimum support i.e. “minsup‟ which is used to control the number of hyperlinks to be prefetched.
3) User-specified keyword list

Output:
1) The optimal list of hyperlinks that are to be prefetched i.e. “L‟ which is large set of hyperlinks when support >= minsup

Variables:
1) “n‟: integer which contains total number of unique keywords
2) i1, i2---------- in are set of unique keywords from list of hyperlinks
3) “k‟: integer
4) “termination‟: Boolean
5) “support[]‟ which contains the support value for the keywords which is calculated with the help of formula
6) “C‟ which contains the set of keywords that are candidate to be prefetched and C1, C2------------ Ck are the candidate keyword set which contains set of keywords of length k with their support value.

Algorithm optimal_prefetching ()

```
{
1. Count unique individual keywords from list of hyperlinks
that matches with the user-specified list by scanning all the
hyperlinks once and say it as “n‟.

2. for j=1 to n do

{
Compute support[ij]=count(ij)/m by scanning all the
hyperlinks once and
counting the number of hyperlinks that keyword ij appears in (
i.e.
count(ij)).
}
3. Now, create the candidate1 keyword set i.e. C1which will
be the set of keywords i1, i2---------- in with their support
value.

4. for j=1 to n do

{
Compute the L1 which contains the subset of keywords from
C1
where support( ij)>=minsup.
}
5. Let k=1 and termination = false
6. Repeat steps (a) to (e) until termination=true

{
a. Let Lk+1=empty.

b. Create the candidate (k+1) keyword set i.e. Ck+1 by
combining members of Lk by selecting and extending k-
keyword set by one more keyword so that set of keywords will
be unique and no keyword will repeat in each set.

c. In addition, only consider as keywords of Ck+1 those k+1
keywords such that every subset of size k appears in Lk.

d. Scan the hyperlinks once and compute the support for each
member of Ck+1. If the support for a member of
Ck+1>=minsup then add that member to Lk+1.

e. If Lk+1 is empty then

termination=true
else
k=k+1
}
7. Now Lk contains the list of hyperlinks that are to be
prefetch
8. End.
}
```

The optimal_prefetching algorithm is taking the hyperlinks of the current web page and the keywords of the user-specified keyword list as input as well as minimum support value i.e. minsup to control the number of hyperlinks to be prefetched. At the first, it will count unique individual keywords from list of hyperlinks that matches with the user-specified list by scanning all the hyperlinks once. For each unique individual keyword it will then compute support by scanning all the hyperlinks once and counting the number of hyperlinks that keyword appears in. By considering the set of keywords of length 1 with their support value, it will create the candidate1 keyword set i.e. C1. Again for each individual keyword it will compute the L1 i.e. Large set of hyperlinks which is containing the subset of keywords from C1 where support is greater than equal to minimum support value. Now by letting k=1, Lk+1=empty and termination = false, it will create the candidate (k+1) keyword set i.e. Ck+1 by combining members of Lk by selecting and extending k-keyword set by one more keyword so that set of keywords will be unique and no keyword will repeat in each set. In addition, only consider as keywords of Ck+1 those k+1 keywords such that every subset of size k appears in Lk.. It then scans the hyperlinks once and support for each member of Ck+1 will be calculated. If the support for a member of Ck+1>=minsup then that member will be added to Lk+1. If Lk+1 is empty then termination=true else k=k+1. Same steps will be repeated if termination is false otherwise it will give Lk as output which will contain the set of keywords for the list of hyperlinks that are to be prefetched.

IV. CONCLUSION

AS THE DEMAND OF ANDROID SMARTPHONES IS INCREASING DAY BY DAY, USE OF THE SOFTWARE WHICH WILL IMPROVE THE PERFORMANCE OF THE ANDROID SMARTPHONE IN TERMS OF ENERGY CONSUMPTION, LOADING TIME AND NETWORK CAPACITY IS ALSO NEEDED.THIS PAER HAS PROPOSED THE DIFFERENT TECHNIQUES WHICH DEALS WITH THE WIRELESS RADIO INTERFACE TO REDUCE THE POWER CONSUMPTION OF THE SMARTPHONE. AGAIN, THE DIFFERENT TECHNIQUES OF PREFETCHING AND CAVHING ARE STUDIED TO REDUCE THE LATENCY IN WEBPAGE LOADING.

REFERENCES

[1] Bo Zhao, Wenjie Hu, Qiang Zheng, and Guohong Cao “Energy aware web browsing on smartphone” in proc. of IEEE Transactions on Parallel and Distributed Systems, 2014.

[2] Dinesh Kumar, Reena Patel, “An Efficient Approach For Optimal Prefetching To Reduce Web Access Latency”, International Journal Of Scientific & Technology Research Volume 3, Issue 7, July 2014.

[3] Mian Dong, Student Member, IEEE, and Lin Zhong, Member, IEEE “Chameleon: A color adaptive web browser for mobile OLED displays”,

[4] IEEE Transactions On Mobile Computing, Vol. 11, No. 5, May 2012

[5] [4] Bo Zhao, Byung Chul Tak and Guohong Cao, “Reducing the Delay and Power Consumption of Web Browsing on Smartphones in 3G networks”, 31st International Conference on Distributed Computing Systems, 2011

[6] Sandhaya Gawade 1, Hitesh Gupta2,”Review of Algorithms for Web Prefetching and Caching” in proc. Of International Journal of Advanced Research in Computer and Communication Engineering Vol. 1, Issue 2, April 2012

[7] H. Zhu and G. Cao, “On supporting power-efficient streaming
applications in wireless environments,” IEEE Transactions on Mobile Computing, 2005.

[8] W. Hu, G. Cao, S. V. K. , and P. Mohapatra, “Mobility-assisted energy- aware use contact detection in mobile social networks,” in IEEE ICDCS, 2013.

[9] J.-H. Yeh, J.-C. Chen, and C.-C. Lee, “Comparative analysis of energy saving techniques in 3GPP and 3GPP2 systems,” in IEEE transactions on vehicular technology, 2009.

[10] Ding Li, William G. J. Halfond, “An investigation into Energy- saving programming Practices for Android Smartphone App Development”, ACM, June 1, 2014.

[11] E. Rozer, V. Navda, “Napman: network-assisted power management for wifi devices MobiSys’10, June 15–18,2010 ACM 978-1-60558-985- 5/10/06

[12] Kaimin Zhang, Lu Wang “Smart Caching for Web Browsers” Proc WWW 2010, April 26–30, 2010, Raleigh, North Carolina, USA. ACM, 2010.