Overnight Pre-caching on Mobile Devices

Troy Johnson
Department of Computer Science
Central Michigan University
Mount Pleasant, MI 48859
johns4ta@cmich.edu

ABSTRACT

In this paper, we evaluate the potential savings achievable on mobile devices by pre-caching websites over night that the user visited the previous day. We hope to help alleviate the problem of short battery lifetimes on mobile devices by reducing the number of requests that have to be made by a mobile device when requesting a common webpage that a user often visits.

General Terms

Measurement, Experimentation

Keywords

Mobile, Caching

1. INTRODUCTION

Mobile device and data growth is at an all-time high and is expected to keep growing. According to figures released by Cisco [1], the number of mobile-connected devices is expected to exceed the world's population by 2014 and global mobile data traffic is predicted to increase nearly 11-fold between 2013 and 2018. A figure of the projected growth of mobile data can be seen in Figure 1.

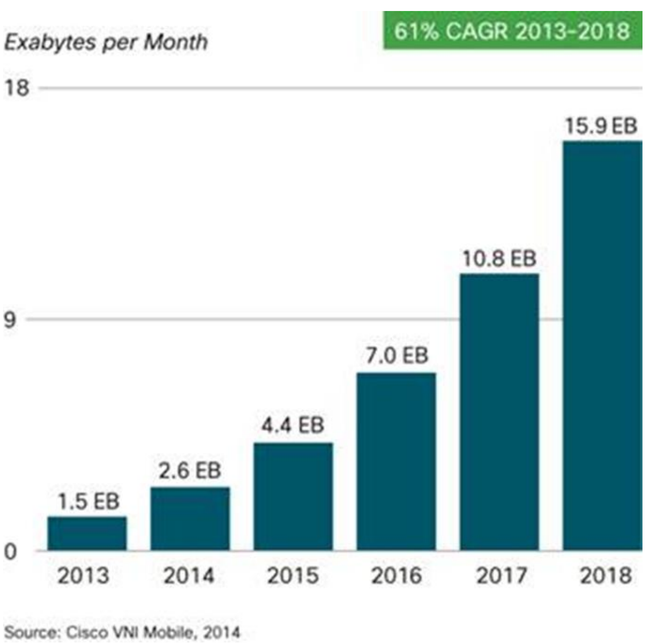

Figure 1. The expected growth of mobile data as predicted by Cisco.

This forthcoming exponential increase in mobile data usage signals the likely need for better and more efficient ways of handling data and reducing the data load that is going to be put on mobile networks. One way of reducing the amount of data that needs to be downloaded over the mobile network is through caching. Caching on mobile devices is a mechanism that causes mobile devices to hold on to recently accessed data from webpages for a desired time period, which is specified by each cache objects expiration age. In the next sections we'll evaluate current works that are centered on developing more efficient and smarter caching systems and then we'll talk about our system and our set up for determining the possible savings that are achievable by pre-caching web pages overnight and then finally we will discuss our results from the simulation.

Related Works

Caching techniques and strategies are currently a hot topic in the mobile computing research domain. In [2], the authors develop schemes to predict user movement patterns based off the previous day's movement patterns. These movement patterns can then be combined with data about the network connection information at each location (type, speed, etc...) to determine when a good time to pre-fetch email or pre-fetch webpage data may be. Caching and pre-fetching can then be performed for the user when there is a strong network connection if the movement patterns predict that the user will soon be without a strong network connection. In [3] the authors develop algorithms for cache replacement and passive pre-fetching of data objects. Their algorithms consider such things as user request patterns, mobility hand offs, and other characteristics that may affect mobile signal strength and power consumption on the mobile device. Their simulations show the possibility of more than 10% savings with the implementation of their cache replacement and passive pre-fetching algorithms.

2. Overnight Caching

Typically, most people tend to plug their mobile devices into a power source over night while they are sleeping. Also, many people tend to have Wi-Fi access in their homes, which can be accessible by the phone while the user is sleeping. Due to this, during the night or early morning, when a user is usually still sleeping, would be a great time for a mobile phone to go out and pre-cache the webpages a user visited the previous day since users are more likely to visit these same sites again the next day. By performing the pre-caching at night, the user would have the advantage of a power source and a Wi-Fi connection so they don't drain their battery or data plan. As a result, they may also have faster load times throughout the day when they try to access the webpages since the webpages will already be cached locally on their mobile devices. This can also result in power savings since many of the items on the webpages may already be cached so the mobile devices won't have to make as many requests to go out and fetch as many of the required objects for a web page.

3. Simulation

3.1 HTTP Archive

To simulate a user's browsing session over a 24 hour period we needed website data that contains information about the cache lifetime of objects for the most popular webpages along with the sizes of these objects in order to determine potential savings by caching these webpages overnight. HTTPArchive.org is an online repository of web page performance information. HTTP Archive utilizes WebPageTest.org to collect web page data on a bi-monthly basis for both the mobile and the desktop versions of the top 1,000,000 websites based on the Alexa index (information from only 4776 mobile web pages is collected). For our purposes, we will only be examining the web page data for mobile versions of webpages. Also, it is important to note that only information regarding the landing page of a website by is collected by HTTP Archive. HTTP Archive sets the settings for WebPageTest.org to that of an iPhone 4 with iOS 5. HTTP Archive records page level information such as total requests, total bytes, total number of bytes from jpeg images, etc... as well as information about all the responses a webpage receives. The response information includes information such as the mime type, response length, url, expiration age, etc... The October 15, 2013 data set was used for my set up. Once downloaded, the data had to be cleaned, sorted into ascending order based on page rank, and then put into a spreadsheet where each row contains all the relevant information (expiration ages and respSizes) that pertains to each landing page.

3.2 Simulation Set up

To determine the overall potential savings we will simulate a 24 hour browsing session for a user. The two behavioral attributes of the user that we incorporate into the simulation model are the user think time and the webpage visited. User think time is the time in between page requests that a user spends viewing a web page before navigating to another webpage. The user think time is determined similarly to the process outlined in [4]. The user think time follows a Pareto distribution with $\beta = 1.5$ and $k = 30$. Website popularity was shown in [5] to follow a zipf distribution with $\alpha = 0.85$. So, to determine what webpage the user visits, an integer will be drawn from a zipf distribution with $\alpha = 0.85$ and the webpage with the corresponding page rank will be the webpage that we simulate the user navigating to. The overall algorithm for simulating a 24 hour user browsing session can be seen below.

```
While time <  86400: //86,400 seconds = 1 day
    Select webpage based on zipf distribution
    For each cacheItem in webpage:
        If expAge > time //Item hasn't expired
            AccumulateSavings()
    time += Pareto(β = 1.5, k = 30)
```

3.3 Overall Cache Lifetime Behavior

The overall lifetime of cache objects from mobile sites is shown in Figure 2. Figure 2 depicts the total bytes remaining in cache over a 24 hour period. Roughly 35% of bytes expire in cache almost immediately. Subsequently, there are also noticeable drops at periods of one hour, four hours, and twelve hours before slowly starting to level off to about 52% of bytes remaining in cache after 24 hours.

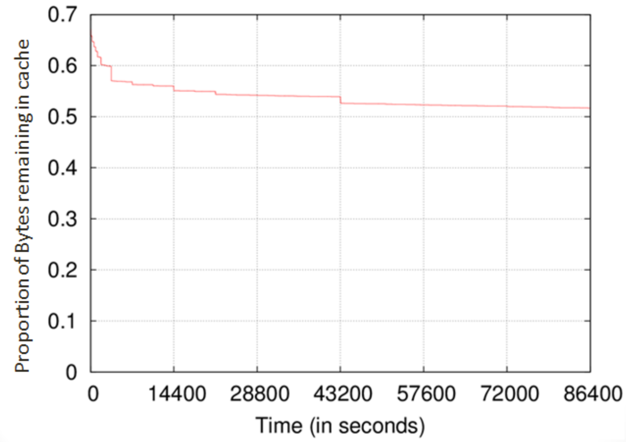

Figure 2. The proportion of bytes remaining in cache over a 24 hour period.

Figure 3 depicts the total responses remaining in the mobile device's cache over a 24 hour period. There is also a steep initial drop to about 55% after the responses with an expiration age of zero expire. There are subsequent noticeable drops at periods of one hour, four hours, six hours, and twelve hours, before starting a steady decline to about 45% of responses remaining in cache at the end of the 24 hour period.

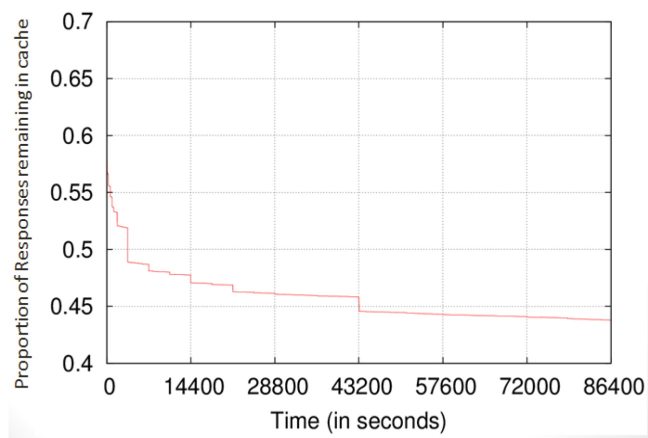

Figure 3. The proportion of HTTP responses remaining in cache over a 24 hor period.

4. Results

We ran the simulation for two-thousand rounds to determine the possible savings along with their confidence intervals. The overall savings for a 24 hour period determined from our simulations seen be seen in Figure 4 and Figure 5 along with their confidence intervals.

In Figure 4, after the initial drop from cache items with expiration ages of zero, the amount of bytes remaining in cache starts out at 43% and starts to level off at about 35% by the end of the day. The most significant visible drops occur at the two hour and twelve hour periods with other smaller noticeable drops throughout the day.

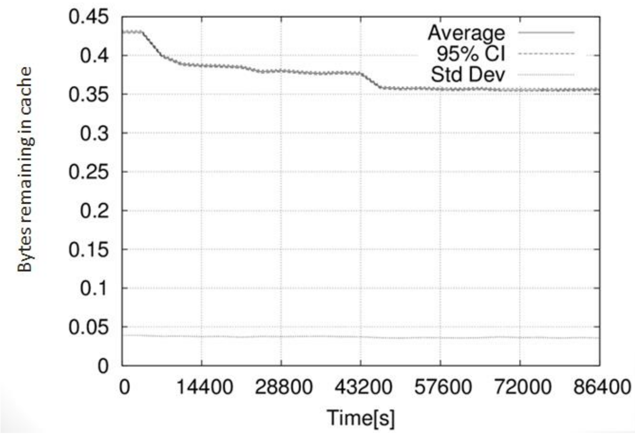

Figure 4. The proportion of bytes remaining in cache over a 24 hour period using our simulation model.

Figure 5 depicts the total proportion of responses remaining in cache over a 24 hour period along with the 95% confidence intervals. After the initial drop from cache items with expiration ages of zero, the proportion of responses remaining in cache starts out at about 37% and levels off to about 30% by the end of the day. As expected, the most visible drops are in similar locations as Figure 4, at the two hour and twelve hour periods with other smaller noticeable drops throughout the day.

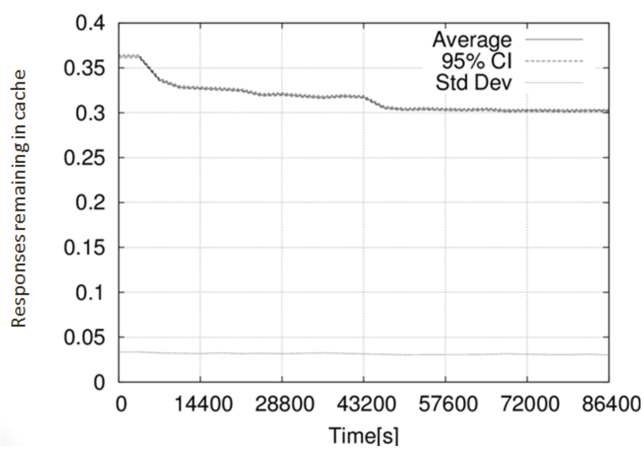

Figure 5 The proportion of HTTP responses remaining in cache over a 24 hour period using our simulation model.

5. Conclusion

Overall, we see that potential savings can be achieved by pre-caching web pages overnight. The number of bytes for web pages that have to be downloaded can be reduced by up to 43%, while the number of requests a device may have to make can be reduced by up to 37%.

In future works we can develop more complex user website visitation models by incorporating more user website visitation patterns into the model along with extending the implementation past just the landing pages of the most popular webpages. Another characteristic to account for would be how often website change their contents throughout the period of a day. If a website changes their contents after a web page has been pre-cached, then those newly added items would have to be downloaded when the webpage is requested since they weren't pre-cached earlier in the day.

6. ACKNOWLEDGMENTS

Thanks to Dr. Patrick Seeling for his guidance in the development of this project.

7. REFERENCES

[1] Cisco, Inc., "Cisco visual networking index: Global mobile data traffic forecast update, 2013–2018," Tech. Rep., Feb 2014

[2] Thyamagondlu, B.N.; Chu, V.W.; Wong, R.K., "A Bandwidth-Conscious Caching Scheme for Mobile Devices," *Big Data (BigData Congress), 2013 IEEE International Congress on* , vol., no., pp.78,85, June 27 2013-July 2 2013
doi: 10.1109/BigData.Congress.2013.20

[3] Huaping Shen, Mohan Kumar, Sajal K. Das, and Zhijun Wang. 2005. Energy-efficient data caching and prefetching for mobile devices based on utility. *Mob. Netw. Appl.* 10, 4 (August 2005), 475-486.

[4] G. Anastasi, M. Conti, E. Gregori, and A. Passarella, "Performance comparison of power-saving strategies for mobile web access," Performance Evaluation, vol. 53, no. 3, pp. 273–294, 2003.

[5] S. A. Krashakov, A. B. Teslyuk, and L. N. Shchur, "On the universality of rank distributions of website popularity," Computer Networks, vol. 50, no. 11, pp. 1769 – 1780, 2006. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128605002513