Cacheability Analysis of HTTP traffic in an Operational LTE Network

Buvaneswari A.Ramanan, Lawrence M.Drabeck, Mark Haner, Nachi Nithi, Thierry E.Klein, Chitra Sawkar
Bell Labs Research, Alcatel-Lucent
600 Mountain Ave
Murray Hill, NJ 07974, USA
(buvana.buvaneswari, l.drabeck, mark.haner, karun.nithi, thierry.klein, chitra.sawkar)@alcatel-lucent.com

Abstract— **With the rapid increase of traffic on the web, content caching reduces user-perceived latency as well as the transmission of redundant traffic on the network. In this study, we analyze the gains of HTTP content caching at the location of SGW in an LTE Wireless network. High cache hit ratio can be achieved if the proxy server caches only those contents that are guaranteed of significant revisits. In this paper, we identify such contents for optimum proxy server performance. We compare the cacheability gains for different content types such as image, video, text etc, and also for popular websites. Our analysis shows that amongst all the contents, 'image' type have the highest revisit rate, which means caching them is beneficial. Amongst the popular websites compared, cacheable contents from 'Facebook' have the highest probability of revisits. We extend the analysis by varying the interval of caching and studying its effect on the cacheability. Based on these results, we provide guidelines for configuring the proxy server for high cacheability benefits.**

Keywords-component; cacheability, Revisit, Cache Hit, LTE Network, Content caching, HTTP, Content type, Hosts, eNodeB, SGW, Data Analysis, User Plane, Traffic Statistics

I. INTRODUCTION

Web caching is a well-known strategy for improving the user experience by keeping Web objects that are likely to be used in the near future at a location closer to the user. Proxy servers play the key roles between users and web sites in minimizing the response time of user requests and saving of network bandwidth [1], [2]. With the advent of 4G LTE (Long Term Evolution) technology and its nationwide popularity and usage, data traffic in mobile networks is growing rapidly. Additionally, 4G networks have enhanced capability to support more features for social networking than previous generations. VoIP, peer-to-peer, and flexible file sharing for multimedia communications will be used extensively in 4G networks.

Extending web caching to mobile networks, it is natural to target the SGW/PGW (Serving Gateway/Packet Data Network Gateway) location for proxy caching. Taking this one step further, some popular content (that is expected to be downloaded by a large number of users and is fairly predictable) can be pre-loaded and cached at the eNB (evolved Node B, i.e. Base Station) to reduce the congestion on the backhaul link and reduce the overall download time and hence the user-perceived quality of experience.

In this paper, we aim to quantify the caching gain in terms of the ratio of requests for which the user perceived latency can be improved and that the bandwidth required is reduced. Most importantly, we provide guidelines in terms of configuration parameters for the proxy server on a per-content type / per-host basis for maximum returns. Our analysis and results are based on LTE network data gathered in one of the busiest markets in North America.

This is the first study to the best of our knowledge on the LTE/3GPP networks aimed towards providing guidelines based on content attributes for proxy caching. Studies exist in the literature that are based on users of residential Internet, 802.11 and 3G Cellular networks. In [3], a match-making content discovery solution for wireless network is proposed, also implementing configurable caching schemes. In [4], a location aware caching scheme is proposed for dynamic environments to increase cache hits and reduce cache query response times. Both [3] and [4] base their results on simulation model and not on real network data. The impacts of caching-related HTTP (Hypertext Transfer Protocol) headers on cacheability decision were investigated in several trace-base studies. C.H.Chi, et al. in [5] provide references on these studies. B.Ager, et al. [6] investigated the potential of caching for a set of application protocols, peer-to-peer and client-server including HTTP, using data from 20,000 DSL subscribers in 2007. A.Arvidsson, et al. [7] propose a distributed caching scheme and claim that their scheme is particularly suitable for cellular mobile networks in minimizing backhaul transmission. Though their study is targeted for Cellular networks, it is based on two popular content models rather than on real Cellular Network traffic. In [8], redundant HTTP transfer analysis was done based on smartphone user data in a commercial 3G network; the focus was towards handset caching and not proxy caching. J.Erman, et al [9] have explored forward HTTP caching in 3G (UMTS - Universal Mobile Telecommunications System) cellular networks by using traffic traces generated by millions of users over a period of 36 hours from one of the world's largest 3G networks. Their study focuses on a cost model for building hierarchical caching nodes at national and regional levels based on the observation that the cache hit ratio can reach as high 33% at the national level and about 27% at regional level. Our study is distinct from the 3G network studies of [7-9] in

978-1-4673-5013-6/13/$31.00

that we are the first ones to propose selective caching based on content type and host. Though this study is based on LTE data, the results on cacheability gains per category are applicable to UMTS and EVDO (Evolution Data Only) networks as well since it depends more on the usage nature of the end user devices than on the air interface technology.

The rest of this document is organized as follows. Section II gives an overview of data collection and the fields used for analysis. Section III defines two cacheability benefit metrics that are used throughout our study. Section IV lists the constraints that were used in our analysis for classifying a HTTP response as cacheable or non-cacheable. Section V shows the results of our analysis and we conclude in Section VI.

II. Overview of Data Collection

Our data was collected in a live LTE network in a major North East dense urban metropolitan market. We use a Bell Labs developed network monitoring tool called LTE Xplorer (see Figure 1), which listens to the control (S11) and data links (S1U) and computes statistics such as per-eNB and per-bearer data volume and number of packets per minute, aggregate data volume and packet counts. It also decodes HTTP packet, retains the correlation of the HTTP Request and corresponding response with the identity of the UE (User Equipment) and the eNB(s) serving the UE.

For the market under study there are ~1300 eNB.cells (i.e. sectors) and the eNBs are assigned to one of two SGW by Tracking Area Code. This means a specific SGW will be handling most of the connections of eNBs belonging to a certain geographical area. The remaining connections (whose proportion is small) from these eNBs will be handled by the other SGW.

The records from real-time data capture are uploaded to a MySQL database hosted in a Red Hat Enterprise Linux server. The fields are: UE Id, HTTP get time & response times, eNB Id, UE & Server IP & Port number, Response Code, Content type & length, URL, Host, Referrer Cookie, Cache-control, max-age, Range – start, end and length, Set-Cookie, Vary, Last Modified Time and Expires. The analysis was done on 24 Hour worth of data from 03/01/2012 00:00 thru 03/02/2012 00:00 and consists of > 42 Million HTTP requests.

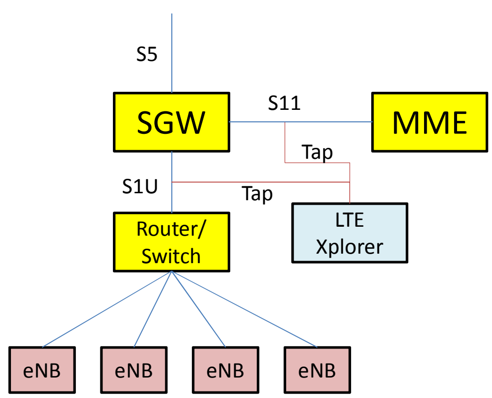

Figure 1. LTE Xplorer configuration

III. Quantifying Cacheability

Caching refers to saving a copy of a reply to a request on a server— the *cache*—with the intention to satisfy subsequent requests for the same content from the cache instead of the origin server. All subsequent requests for this cached data are provided by the proxy server which is strategically located near the users and results in quicker data retrieval and lower network bandwidth usage.

To quantify the cacheability benefits we adopt a metric similar to B. Ager, et al [6] who investigated the potential of caching for a set of application protocols, peer-to-peer and client-servers. They introduced the metric, *cacheability*, to quantify the gains of caching. This metric is based on the number of cacheable requests and the cacheable data volume.

The metric on cacheability benefits based on data volume is essentially the revisited data volume % and is given by:

$$\text{Revisited DV\%} = \frac{100 * \sum_{i=1}^{n} (k_i - 1) s_i}{\sum_{i=1}^{n} k_i s_i} \quad (1)$$

Here k_i denotes the total number of downloads for item i and s_i, is the size of item i. The metric on cacheability benefits based on the number of revisits is essentially the % of revisited contents and is given by

$$\text{Revisit \%} = \frac{100 * \sum_{i=1}^{n} (k_i - 1)}{\sum_{i=1}^{n} k_i} \quad (2)$$

Here k_i denotes the total number of downloads for item i.

IV. Criteria for 'Cacheable' Contents

There exists several studies regarding what type of data is deemed as cacheable at a proxy server [5], [10-12]. The impacts of caching-related HTTP headers on the availability decision were investigated in several trace-base studies. C.H.Chi, et al. in [5] provide a survey of these studies. We adopt their recommendations on the availability, freshness and revalidation decisions with a few minor modifications.

The following cases are deemed not-cacheable:

- Set-cookie non-NULL
- Vary non-NULL (For simplicity reasons, we do not consider them for caching. For details, refer to Section 7.3.3 in [2])
- Content-Length = 0
- Last-Modified = 0
- Cache control = "private" / "no-store" / "no-cache"
- All HTTP Response codes NOT in the set {200, 203, 206, 300, 301, 410}

Responses were grouped with respect to URL, Range_start and Range_end. We assume the entire content volume as denoted by Content Length is downloaded. It is possible that while the contents of an original visit to a given URL were being fetched, the user navigated away resulting in only the partial content available at the proxy server. This tends to lead to an overestimation of the data volume for large files (like video).

V. RESULTS

A. *Cacheable Content Volume*

For data collected over 24 hours, Table 1 shows the requests, data volume, proportion of data that is cacheable and the revisit rate (hit rate) for data stored in the cache. Over the entire day of 3/1/12, 42.6 Million content requests were recorded and this corresponded to a data volume of 12.2TB. There were 13 million cacheable requests which correspond to 9 TB of data. The response for 54.3% of the cacheable requests was revisits but this corresponds to only 810 GB of data volume.

Table 1. Cacheable Data Statistics for a Given Measurement Period

Category	Value
Data Collection Time	3/1/12 (24hrs)
Number of Request	42.6 Million
Data Volume (all Requests)	12.2 TB
Cacheable % (Requests)	30.6%
Cacheable % (Data Volume)	73.9%
% Revisits (Requests)	54.3%
% Revisits (Data Volume)	9.0%
% Revalidation of *Revisits* (Requests)	26.9%

A network proxy installed at the SGW that adheres to all the constraints of Section IV would improve the user perceived latency of 7 million requests (16%) and reduce the backhaul by 810 GB (6.6%) The proxy would have to save 13 Million records that occupy about 9 TB in order to achieve this efficiency.

A high revisit rate but low revisit data volume makes it clear that there is a set of small objects that are frequently revisited and there are some bulky contents that are not revisited. Is it possible to identify them based on attributes such as content-type and host and exclude them from caching? This will result in smaller cache that can be maintained for longer.

State of the art proxy servers continuously perform the updates and eviction; the revisit rate will certainly increase as the interval of caching increases. A limiting factor for a proxy server is the space available for storage. It would be typical for a network operator to store contents with high predicted revisit rate and exclude those bulky contents with low predicted revisit rate. By performing a per-content type / per-host analysis, we can predict the revisit rate of contents and configure the proxy server with a decision policy for storing contents based on the content type, length and host fields.

Please note that although the results presented here are for a 24 hour period on 3/1/2013, we found that there was no considerable variation for other weekdays between 2/13/2012 and 3/20/2012.

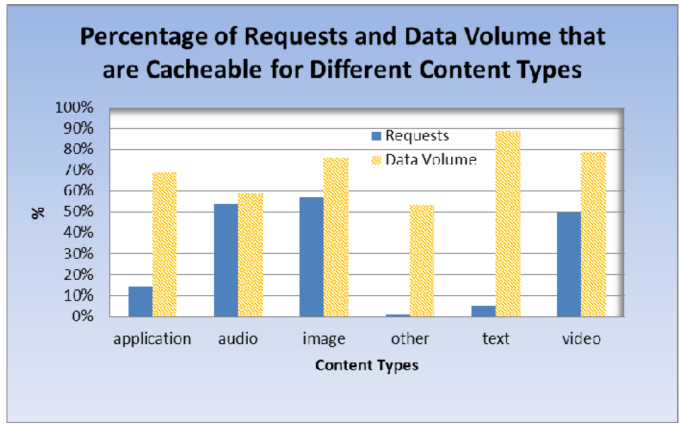

Figure 2. Percentage of Requests and Data Volume that are cacheable for Different Content Types

B. *Cacheability Benefits of different content types*

The contents are grouped into 6 major categories and we study the metrics on cacheability benefits, the *Revisited_DV_%* and *Revisit_%* for each content type.

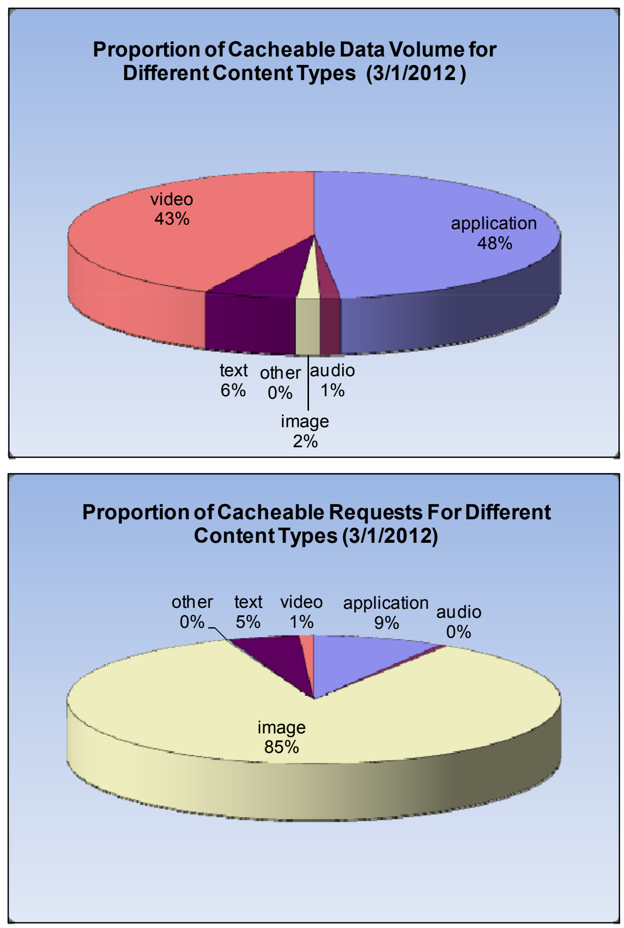

Figure 3. Distribution of data volume (top) and number of cacheable requests (bottom) among the different content types

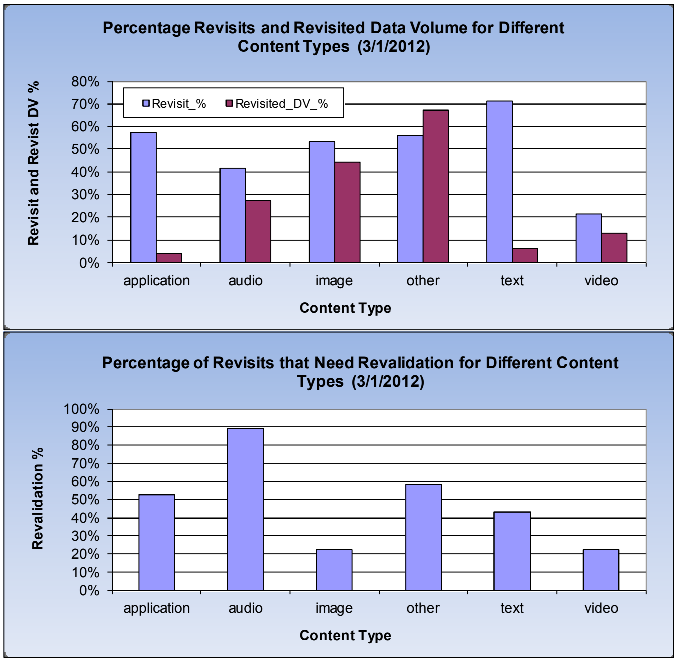

Figure 4. ***Revisited_DV_%*** **and** ***Revisit_%*** **(top) and Revalidation% (bottom) for different content types**

Figure 2 shows the % of cacheable requests and data volume for the different content types. Audio, images and video have >50% of the request being cacheable but application, text and others have < 10% of the requests being cacheable. The data volume shows a different story with > 50% of the data volume for all categories being cacheable.

Now looking at the total cacheable requests and data volume amongst all categories, Figure 3 top shows that application and video have the largest data volume for cacheable content. Their size exceeds the other content types by almost an order of magnitude. If their *Revisited_DV_%* value is also good, then caching them will result in considerable bandwidth (S5) saving.

Figure 3 bottom shows, by content type, the number of requests for cacheable contents. Image content type has the most cacheable requests with their cacheable requests almost an order of magnitude large than the other content type. If their *Revisit_%* value is also relatively high, then caching them will result in significantly improving the user perceived latency. Audio and video content types have the smallest number of cacheable requests.

For the proxy server to be effective, we not only require the requests to be cacheable (Figure 3), but also the number of times the contents are revisited to be ≥1 (cache hit). Figure 4 shows the *Revisited_DV_%, Revisit_%* and Revalidation% of different content types. Remember, the *Revisit %* and *Revisted_DV_%* are the percentage of requests and data volume that are replicates and could be served by the cache. Here are some results and observations:

- Image contents show both a large *Revisited_DV_%* and *Revisit_%*. Since the image requests are the highest among all the cacheable requests, excellent user perceived latency can be achieved by caching them. Considerable bandwidth reduction is also possible. The cost incurred for these gains is small (only ~20% revalidations).
- Text contents have high *Revisit_%* but very low *Revisited_DV_%* showing that small text objects are revisited more often than large text objects. Caching such text objects will result in improved user perceived latency but will not really reduce backhaul data volume and will have fairly high revalidations (~40%).
- Application has the largest cacheable data volume and second largest cacheable requests but, like text, shows large *Revisit_%* and fairly small *Revisited_DV_%* (many small objects revisited but no large ones). This again will improve latency but not help with backhaul reduction with caching .Also, the revalidations are high at ~50%.
- Audio shows fair *Revisited_DV_%* and *Revisit_%,* but it's revalidation is extremely high and it's data volume and number of requests are low making this not a good candidate for caching
- Video content has low revisits and revisited data volume and a very small number of requests (1%) but it has a very large data volume. Caching video could result in a reduction in backhaul traffic but a large cache space would be needed. There would be very little if any impact on latency with video caching.

Thus, if the aim of proxy server is to reduce user perceived latency, then configure it to cache image, text and application. If the aim is to save on bandwidth, then configure it to cache just video.

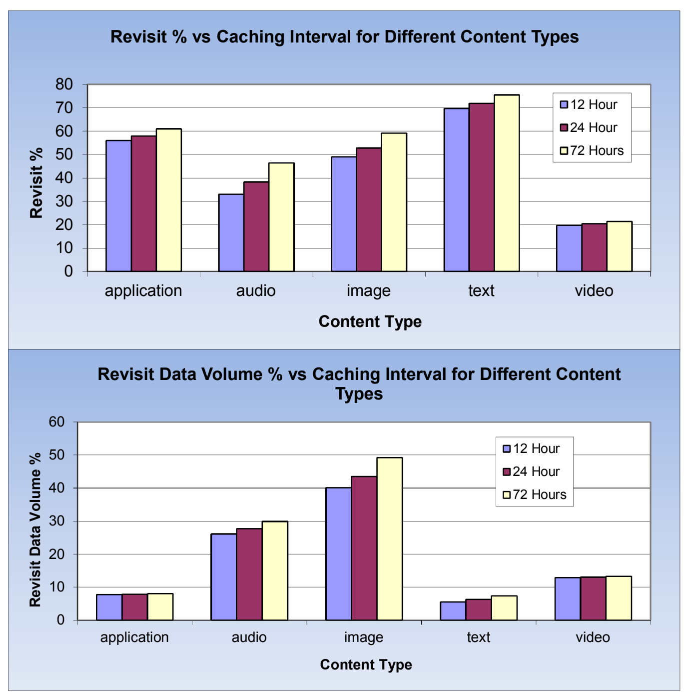

Figure 5. Revisit_% (top) Revisited_DV_% as a function of caching interval for different content types

Next, we study the behavior of *Revisit_%* and *Revisited_DV_%* of different content types as a function of caching interval. The aim is to see if increasing the caching interval will result in higher gains for one content type over the others. This study has direct implication on the eviction policy of the cache.

From Figure 5, we find that except for video, all other content types show increasing *Revisit_%* for longer caching intervals. Caching of Audio contents for longer duration will have the largest benefit towards *Revisit_%*. Except for video & application, the *Revisited_DV_%* of other content types increases for longer caching intervals. The largest benefit comes for image content types.

Thus, we recommend to configure the proxy server with longer eviction time (days) for image and shorter time for everything else for great cacheability benefits.

These increased *Revisit_%* and *Revisit_DV_%* with longer caching intervals normally come at a cost of (a) larger storage space and (b) additional revalidations. However, we found that the *Revalidation%* remains almost the same for the different caching intervals. The reason for this behavior can be explained from the max-age cumulative distribution function (cdf) (see Figure 6). There is very little change in the max-age between 12 and 72 hours (except for video) and this leads to little change in the *Revalidation%* over the different caching intervals that we considered. It is interesting to note that for the different content type, most max-age values are <3 Hrs or >24 hrs with almost none between these two values. The cost in terms of revalidation for longer duration of caching is insignificant when compared to the benefits. However, there is still the cost associated with large storage and retrieval.

Since video and application contents tend to be bulky, storing them for longer duration may not provide good returns. Their eviction interval should be kept shorter. Even though audio contents show increasing trend in *Revisit_%*, owing to their high *Revalidation%* (~90% for all the intervals considered), their eviction interval should be kept shorter. Image contents will benefit significantly from longer duration caching at a very minimal cost. Since their size is also small, caching them for longer duration would be very beneficial.

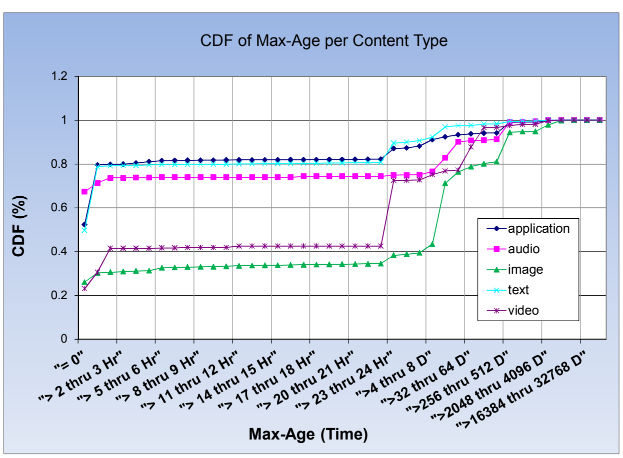

Figure 6. Cumulative distribution function (cdf) of the max-age of different content types

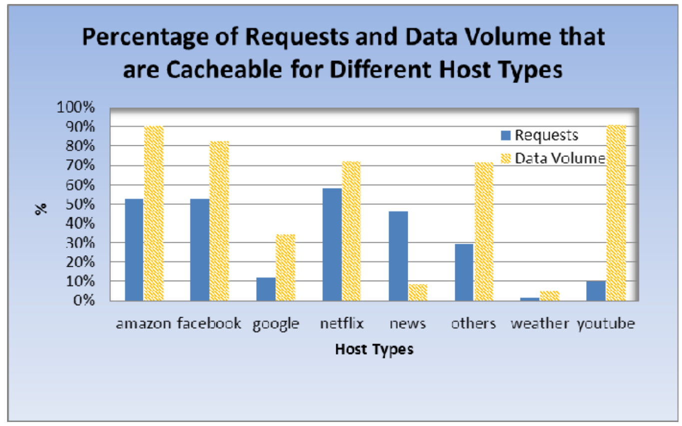

Figure 7 Percentage of Requests and Data Volume that are cacheable for Different Hosts

C. *Cacheability of different hosts*

Here, we identified a few popular hosts and repeated the same type of analysis as done for the content types. The hosts considered are – Amazon, Facebook, Google, Netflix, news, weather and Youtube. 'news' is a general term that we use for a group of news channels - ABC, FOX, CNN, MSN and ESPN. 'weather' is a general term for any host with 'weather' in its name (e.g. weather, weatherbug, accuweather).

Figure 7 shows the availability of the different hosts for caching. Amazon, Facebook, Netflix and News shows ~ 50% of the requests to be cacheable while Google and Youtube show very low ~10% of the requests to be so. The data volume cacheability is >70% for Amazon, Facebook, Netflix, Youtube and "others" but quite low for news and weather.

Figure 8 shows the proportion of cacheable requests and data volume for these different hosts. Netflix and Youtube have large data volume (predominantly videos) but very low requests (a few requests for big files). If they exhibit good *Revisited_DV_%*, then caching them will certainly result in great bandwidth savings (assuming that the proxy server can deal with the excessive storage required). Facebook, on the other hand has many requests but small data volume. Still, excellent user perceived latency can be achieved by caching these contents provided their *Revisit_%* is high. Google and News requests are the next most populace requests but also show low data volume (when compared to Netflix and Youtube). Weather data is very small in DV and requests, meaning the likely gains from caching will not be high in the absolute sense.

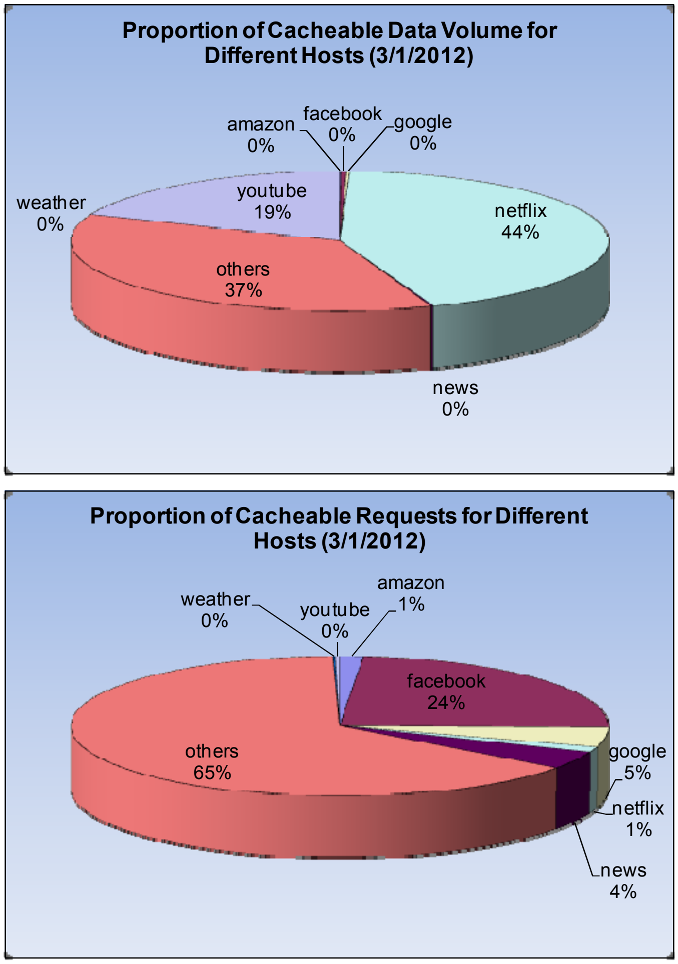

Figure 8. Distribution of cacheable data volume (top) and cacheable requests (bottom) amongst popular hosts

Figure 9 shows the *Revisited_DV_%* and *Revisit_%* of the hosts considered.

- Netflix has the smallest *Revisited_DV_%* and moderately high *Revisit_%*. Small objects from Netflix are likely to be visited often (such as icons and button figures). There may be a little benefit caching only the small objects of Netflix.
- The contents of Youtube are not revisited often as compared with other hosts and the *Revisited_DV_%* for Youtube is also very small. There will be bandwidth saving in the absolute sense owing to their bulky nature.
- Weather's contents have excellent *Revisited_DV_%* and *Revisit_%* values; but their small size, low number of requests and high revalidation % makes them unattractive for caching. The icons on the weather pages are mostly what are cacheable.
- News contents have excellent *Revisited_DV_%* and the highest *Revisit_%* values. Earlier, we saw that their size and number of requests are considerably good. They seem to be attractive candidates for caching, with the only caveat that 36% of revisits need revalidation.

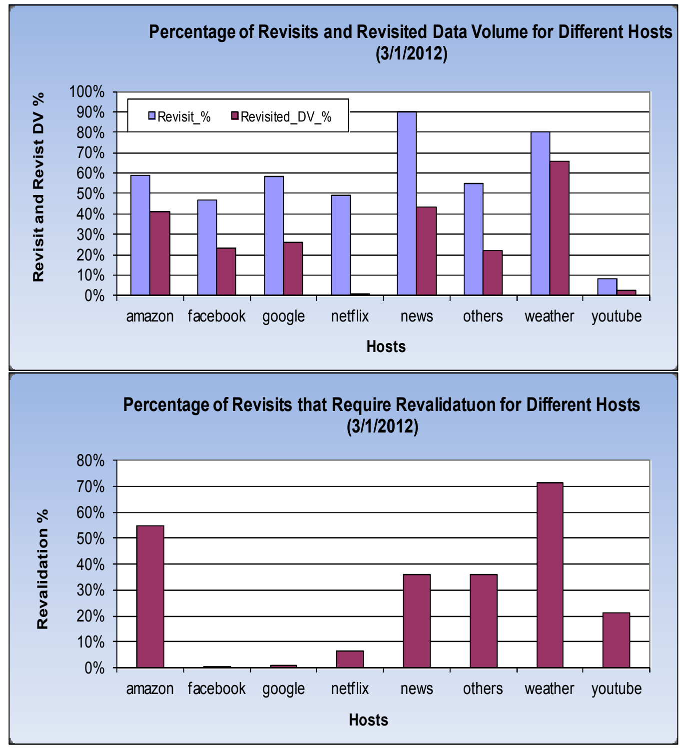

Figure 9. *Revisited_DV_%* and *Revisit_%* (top) and Revalidation% (bottom) for different hosts.

- Google and Facebook qualify as the next best candidates for caching due to the fact that their size, number, *Revisit_%*, *Revisited_DV_%* and *Revalidation%* are all significantly very good.

Thus, it should be noted that if the aim of the proxy server is to reduce user perceived latency, then configure it to cache news, Facebook, and Google contents. If the aim is to save on bandwidth, then configure it to cache Youtube.

Figure 10 shows the *Revisit_%* and *Revisited_DV_%* as a function of caching interval for different hosts. There is definitely a gain in terms of additional revisits as the interval of caching increases. We also found that Revalidation% (not shown here) remains almost the same over the caching intervals considered for all hosts except Netflix.

- The increase in *Revisited_DV_%* for longer caching interval is more than the increase in *Revisit_%* for Google, news and weather, which implies that the probability of larger contents getting revisited long after they are stored for the first time in cache is higher than that of smaller contents. Caching them for longer duration benefit, more so for Google whose Revalidation% is negligible.
- Amazon and Facebook show an increase in both *Revisit_%* and *Revisited_DV_%*. Longer caching intervals for Facebook in particular should result in great benefits as its Revalidation% is almost zero for all the 3 intervals considered.

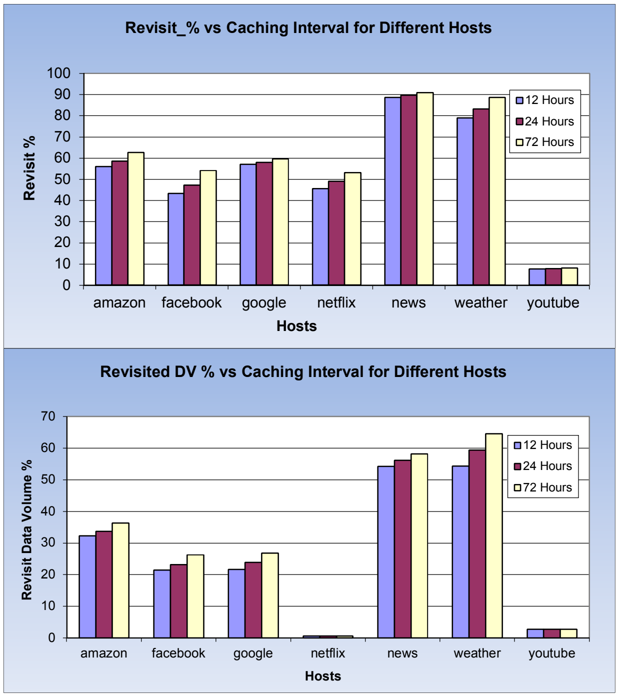

Figure 10. Revisit_% (top) and Revisited_DV_% (bottom) as a function of caching interval for different hosts

- Youtube's *Revisit_%* increases slightly for longer caching intervals and its *Revisited_DV_%* remains almost the same. The returns of caching for longer duration are very small and since they are very bulky, it's better to cache them for shorter duration.

- Netflix shows an increase in *Revisit_%* and *Revalidation%* for longer duration caching. As stated earlier, small objects from Netflix may be cached and should be configured with short eviction time.

Thus we recommend that the proxy server be configured with a longer (several days) eviction time for Facebook and larger contents of Google, news and weather and shorter eviction time for everything else.

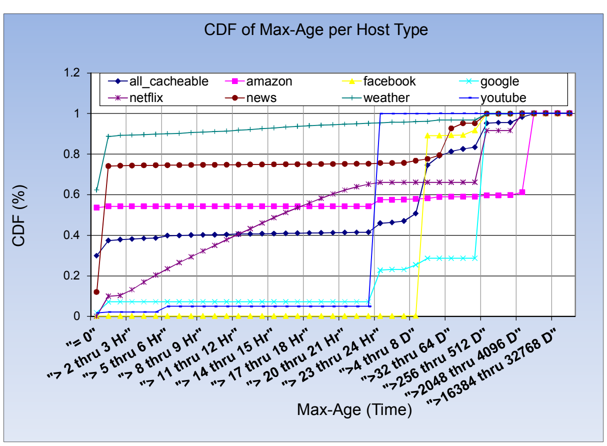

Figure 11. CDF of the max-age for cacheable content broken down content type (top) and by Hosts (bottom)

Now let us focus briefly on the max-age behavior to understand the behavior of Revalidation% for longer intervals. Figure 11 shows the cumulative distribution function (CDF) of the max-age for different host types. Facebook, Google and Youtube have a huge proportion of contents with very long max-age, meaning longer caching interval will not result in increasing Revalidation%. The behavior is almost the inverse for weather and news contents. Note that the proportion of contents with max-age between 12Hrs and 72 Hrs is small, which explains why we saw no change in Revalidation%. Netflix is interesting as it is the only one that shows a range of max-ages between 0 and 24 Hrs. This explains why the Revalidation% increases for longer caching intervals.

VI. CONCLUSION

In this paper, we aimed to identify those contents which when cached at the location of SGW in an LTE network, would result in high cacheability benefits. We presented results on availability (to cache) and revisit rate of HTTP data based on real measurements in LTE networks. We found that 73% of the data volume and around 30% of the responses are 'cacheable'. Within the cacheable category, around 9% of the data volume and > 54% of the request / responses are revisited.

We identified the contents for selective caching based on the content type and host attributes. Amongst the different content types, application and video constitute the bulk of the cacheable data volume whereas images constitute the bulk of the cacheable URL requests. Since image content have a very high *Revisited_DV_%* and *Revisit_%* values, caching them will result in extremely good returns in terms of improving user perceived latency. Their revalidation overhead is also very small owing to their long max-age duration. Because of this, their caching interval can be chosen much longer than others. Text contents exhibit excellent *Revisited_DV_%* and *Revisit_%* values, but the absolute gain will be small since their size and number is smaller compared to other types. Video contents exhibit around 10% *Revisited_DV_%* and 20% *Revisit_%* values. Caching them will result in significant bandwidth savings at a high storage overhead. In summary, configuring the proxy server to cache image contents with longer eviction time will yield excellent benefits.

Amongst the popular hosts considered, Netflix and Youtube constitute the bulk of the cacheable data volume where as Facebook and Google constitute the bulk of the cacheable URL requests. Caching Netflix will not result in significant savings in bandwidth whereas caching Youtube will result in some bandwidth savings at a high storage overhead. Caching news and weather contents will result in excellent user perceived latency with some compromise in terms of revalidation overhead. Caching Facebook and Google would improve the user perceived latency. Choosing a larger caching interval for Facebook and a smaller caching interval for Google will provide best returns. In summary, configuring the proxy server to cache Facebook and Google contents with longer eviction time will yield excellent benefits.

In our further study, a cost function involving revisit rate, revalidation rate, content length distribution (which has direct impact on the storage requirement) and number of requests that need to be cached (which has direct impact on the lookup time) on a per-attribute basis will be utilized to precisely arrive at a decision policy for caching and eviction. Additionally, benefits due to pre-fetching at the SGW location will be evaluated. Caching and pre-fetching at the eNB, which is located much closer to the user than the SGW will also be investigated.

ACKNOWLEDGMENT

The authors would like to thank their chief customer contact Kerry I. for providing all the facilities needed for the data capture and also for the valuable discussions.

REFERENCES

[1] W.Ali, S.M.Shamsuddin, and A.S.Ismail, A Survey of Web Caching and Prefetching, *Int. J. Advance. Soft Comput. Appl., Vol. 3, No. 1, March 2011*

[2] B.Krishnamurthy and J.Rexford,Web Protocols and Practice: HTTP/1.1, Networking Protocols, Caching, and Traffic Measurement, May 14, 2001 | ISBN-10: 0201710889

[3] F. Malandrino, C. Casetti, and C. Chiasserini, Content Discovery and Caching in Mobile Networks with Infrastructure, *IEEE Trans. on Computers*, 61, 2012, pp. 1507-1520.

[4] Bo Yang; Mareboyana, M.; , Caching for Location-Aware Image Queries in Mobile Networks, *IEEE International Symposium on Multimedia* , 5, 2011, pp. 410-415.

[5] Chi-Hung Chi, Lin Liu, LuWei Zhang, "Quantitative Analysis on the Cacheability Factors of Web Objects", *COMPSAC 2006*.

[6] B. Ager, F. Schneider, J. Kim, A. Feldmann, „Revisiting Cacheability in Times of User Generated Content", *IEEE Infocom 2010*.

[7] Arvidsson, A. Mihaly. L. Westberg, "Optimise Local Caching in Cellular Mobile Networks", *Computer Networks 55, 2011, pp. 4101-4111*.

[8] F.Qian, K.S.Quah, J.Huang, J.Erman, A.Gerber, Z.M.Mao, S.Sen, and O.Spatscheck, "Web Caching on Smartphones: Ideal vs. Reality", *MobiSys 2012, Low Wood Bay, Lake District, UK*

[9] J. Erman, A. Gerber, M. Hajiaghayi, D. Pei, S. Sen and O. Spatscheck, "To Cache or Not To Cache: The 3G Case", *IEEE Internet Computing, Mar./Apr. 2011, pp. 27-34.*

[10] http://wiki.squid-cache.org/SquidFaq/InnerWorkings

[11] A.Wolman, G. M. Voelker, N. Sharma, N. Cardwell, A. Karlin, and H. M. Levy., "On the Scale and Performance of Cooperative Web Proxy Caching", *Proceedings of the Seventeenth ACM Symposium on Operating Systems Principles*, Kiawah Island, SC, December 1999.

[12] A.Feldmann, R. Cáceres, F. Douglis, G. Glass, and M. Rabinovich, "Performance of Web Proxy Caching in Heterogeneous Bandwidth Environments", *Proceedings of IEEE Infocom'99*, March, 1999, pp. 107-116.