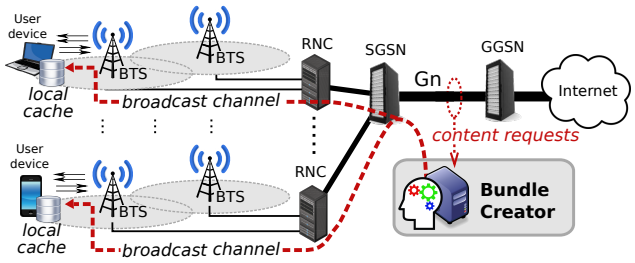

Figure 1: System architecture.

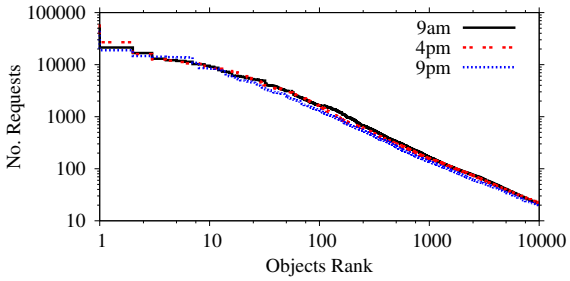

Figure 2: Number of requests for the most downloaded objects.

broadcasting to mobile devices a "bundle" with the most expensive (popular and voluminous) content observed in past periods. Using our traces, we show that such a system, if present, would i) reduce the wireless link load by up to 40% while ii) improving the end-users' performance by avoiding the download of up to 20% of objects in optimistic scenarios with a bundle of 100 MB.

However, performance of such an optimal system is dependent on the bundle (and local cache) size. Indeed, bundling the most expensive content would entail broadcasting popular elephants, creating bundles of several GBs. We further discover that some popular objects are forced to be "not-cacheable" by content providers, despite the objects being static and very popular. Considering conservative scenarios where the cache size is limited to 100 MB and only cacheable objects can be bundled, overall savings reduce to about 10% for both wireless link load and end-users' downloaded volume. Given our results, there could be network deployments that could benefit from content pre-staging on phones, but such a decision would have to be based on the perceived value gain vs. associated overheads, and such a decision is bound to vary across networks.

2. ARCHITECTURE

The main contribution of this paper is to quantify any potential benefits of pre-staging content to mobile terminals using large data sets. In order to assist in this quantification process, we describe an architecture that can enable content pre-staging at a high level.

Fig. 1 presents the architecture of such a system and how it would be incorporated in a cellular network today. Geographical areas are divided into cells. Each cell is served by a Base Transceiver Station (BTS) that provides network access to mobile devices by means of shared radio channels. A group of BTSs is managed by a Radio Network Controller (RNC) that acts as a bridge between the radio access network and the packet switched network represented by a Serving GPRS Support Node (SGSN). Finally, a Gateway GPRS Support Node (GGSN) is responsible for the internetworking between the traffic of a specific geographical area and the rest of the Internet. Today, a typical spot to monitor traffic, as well as to deploy caching solutions, is on the Gn interface between SGSN and GGSN. We envision a solution that performs pre-staging via a new component that we call *bundle creator*, that is capable of selecting objects to push/pre-stage to mobile terminals' local cache to optimize some objective function (e.g., by minimizing the number of popular object re-downloads). For efficiency reasons, objects are grouped in *bundles* that are periodically (e.g., every hour) pre-staged on devices by means of a broadcast channel. For instance, the system could rely on emerging technologies such as eMBMS [6], that offer broadcast channels in 3G/4G networks. Multicast can be also exploited on the wired back-haul up to the bundle creator. Mobile terminals receive the bundle and store content in a local cache. The location of the bundle creator, the size of the bundle, and the frequency of the transmission of the bundle are all design choices that have to be carefully evaluated.

We reiterate that our objective here is not to design and compare different systems that enable such pre-staging, but rather to quantify the benefits that the system could produce, if any. We assume the existence of a bundle creator that observes the popularity of objects during a given time period, then forms and broadcasts the bundle to mobile devices which in turn use pre-staged content in the next time period.

3. DATASET AND CONTENT PROPERTIES

We rely on a data set of anonymized HTTP traffic. The data sets consists of HTTP requests generated by mobile terminals located in an European metropolis during a whole day (November 26th, 2012). Data has been collected by vantage points monitoring the Gn interfaces between SGSNs and GGSNs (see Fig. 1). At each interface, a proxy accelerator handles HTTP traffic to speed up data delivery. In addition, it logs information about the requested objects on a text log file for each HTTP transaction. Among all exposed information, we focus on the (anonymized) terminal ID, requested URL, number of downloaded bytes and a flag stating if the object has been locally cached by the proxy handling the request. Overall, the data set contains 48M HTTP transactions corresponding to 721 GB of volume downloaded by more than 200K mobile subscribers.

To validate the idea of content pre-staging, we start our analysis by investigating three fundamental content properties: *popularity*, *cacheability*, and *lifetime*.

Popularity: The data set contains more than 7 million distinct URLs over the day. For the top 10,000 objects, Fig. 2 reports their popularity measured as the number of requests observed considering 1-hour long time interval. Measurements at 9am, 4pm, and 9pm are reported using a log-log scale (similar results hold for other intervals). As expected, distributions follow a classic heavy tailed shape. The top 100 objects present more than 1,000 requests, and more than 90% of objects generate less than 100 requests.

Considering different measurement periods, we see that popularity does not change significantly. Interestingly, we also found a strong correlation between popular objects re-

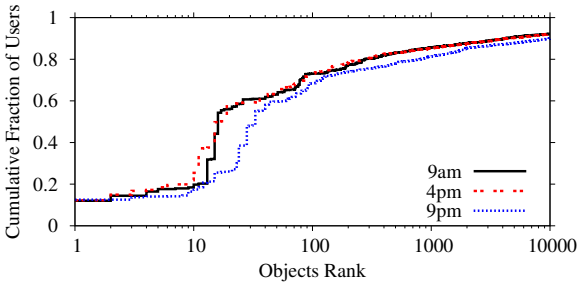

Figure 3: Cumulative fraction of users requesting the most downloaded objects. Knee-points between the top 10-30 objects are related to Apple iTunes/Store content, navigation apps/services.

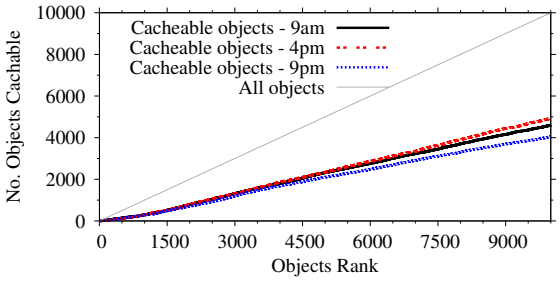

Figure 4: Cacheability of popular objects.

quested at different hours. More specifically, about 50% of the top 100 objects are still among the top 100 in the following hour. These popular objects account for a large fraction of the traffic. For example, the top 1,000 requested objects account for 15% of volume and 48% of requests during the peak hour.

To better understand the popularity of a given content among different users, Fig. 3 shows the cumulative fraction of users requesting the top downloaded objects. We can see that the top 10 objects are downloaded by about 20% of terminals. Investigating further we found these objects to be application control messages of mobile apps, and advertising content.

Overall, the top objects accounts for a large amount of requests and they are very popular among the majority of users. This indicates that, despite the increasing growth of dynamic and personalized applications, object popularity in cellular networks exhibits the desirable properties that would make caching effective. Our finding is in line with previous work [2].

Cacheability: It is well known that not all content is cacheable. By checking HTTP responses, a recent work found 66% of objects to be cacheable considering a U.S. mobile network [7]. However, we are interested in observing if popular objects are allowed to be cached. To investigate this, we rely on cache hit/miss indications reported in the logs. Intuitively, an object that a proxy was able to cache is cacheable. Notice that this does not imply that an object that the proxy was not able to cache is not cacheable.

Surprisingly, we observe that only 7-10% of objects are cached by the proxies. This low fraction can be due to limitation of the cache size which maybe too small to effectively cache moderately popular objects. Yet, for the most popular

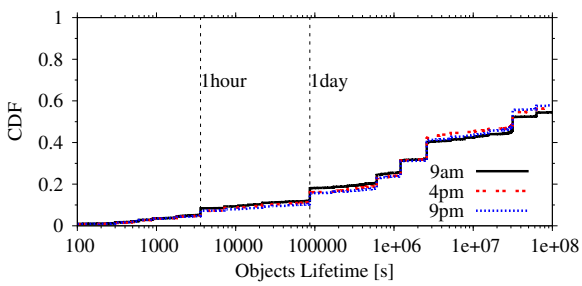

Figure 5: Lifetime CDF of cacheable objects.

objects the proxies are expected to be efficient (as testified by the 25% average hit ratio).

Fig. 4 reports the total number of cacheable objects among the top x most popular objects. If all objects were cacheable, we would observe the curve $y = x$. Unfortunately, we observe that the probability that objects are cacheable is between 0.4 and 0.5 (about 4,000/5,000 objects results cacheable within the top 10,000 objects).

Surprisingly, the most popular objects result not to be cacheable. For instance, only 23 objects are found cacheable in the top 100. Investigating further, we observe that (i) some popular objects correspond to dynamic content (e.g., pages with the same URL, but generated on a per-user basis using a cookie), and (ii) some objects which are actually forced to be not cacheable by the content owner (e.g., impression pixels[1]). For instance, a 500 Bytes long GIF image is constantly among the top 10 most popular objects (more than 15,000 requests), but it is declared as "non-cacheable". This holds true for any time period. Another interesting example are YouTube videos. Despite their static nature, the service allows caching content only on the end-users' terminal but not on any "intermediate cache".

Lifetime: The last requirement for pre-staging is that content should have a long lifetime. For this analysis we relied on active measurements. We downloaded each cacheable objects and inspected the caching directives and the declared object lifetime in the HTTP responses [8, 7].

Fig. 5 reports the lifetime Cumulative Distribution Function (CDF) of content requested at different hours. Peaks are visible at common lifetimes, e.g., at multiple of hours or days. We observe that only less than 8.4% of objects declare a lifetime smaller than 1 hour and 82% of the objects indicates a lifetime larger than 1 day. Again, no differences are observed among different hours.

4. SYSTEM BENEFITS

We now assume that the system is in place and we assess the benefit of content pre-staging by means of trace driven simulations using two different scenarios namely *conservative* and *optimistic*. In the first scenario only cacheable objects can be bundled by the bundle creator. As previously reported, we learn that the cacheability properties from the traces is conservative. Thus the savings achieved by pre-staging can represent a lower bound. In the second scenario we instead optimistically assume that all objects are

[1]Ad impressions - an Ebay affiliate quality factor, `http://goo.gl/FrJe7`

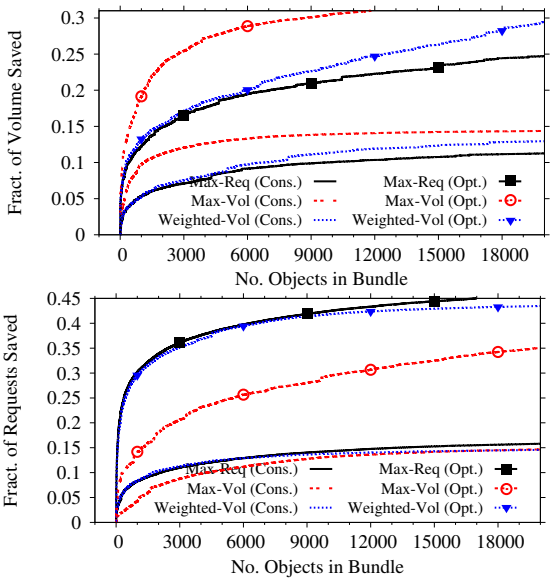

Figure 6: Fraction of volume (top) and requests (bottom) saved for different number of objects in the bundle.

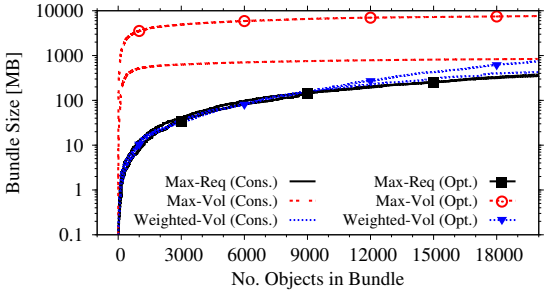

Figure 7: Bundle size with respect to number of objects in the bundle.

cacheable, thus bundled. This defines an upper bound for the performance.

Once the set of candidate objects to be pre-staged is defined, different strategies can be used to compose the bundle. For instance, selecting content i by its popularity $N_{req}(i)$ optimizes the hit probability. Conversely, assuming there is no constraint by the device cache size, bundling content according to the total generated volume $Size(i) \times N_{req}(i)$ guarantees the highest volume savings theoretically. In the rest of the paper we refer to these two strategies as *Max-Req* and *Max-Vol* respectively. We further consider a third bundling strategy, named *Weighted-Vol*. For this strategy, an object i is ranked based on the function $R(i) = \log(Size(i)) \times N_{req}(i) \times Users(i)$, where $User(i)$ is the number of users requesting content i. This represents a hybrid strategy where large objects are more likely to be bundled if they are popular too.

In the remainder of this section we evaluate the system with respect to different scenarios and bundling strategies. We start by considering savings from the 3G network point of view, and then we further dig into the users' perspective. Based on the results reported in Fig. 5, for all the analysis we consider a single time window of 1 hour. Most of our results refer to a bundle created observing traffic from 5pm to 6pm, then broadcasted at 6pm, and used between 6pm and 7pm. This corresponds to peak hours in the data set.

4.1 Savings for the Network

Peak hour: Fig. 6 reports the fraction of total volume (top) and requests (bottom) saved during the peak hour with respect to the number of objects in bundle. Let us consider the top figure first. As expected, the conservative scenario leads to lower savings than the optimistic one. The *Max-Vol* strategy guarantees the best performance, with 1,000 objects already offering 9.7% (19.1%) of savings in the conservative (optimistic) scenario. The *Max-Req* and *Weighted-Vol* present lower savings with only 5.3% and 5.2% of volume reduction for the first 1,000 objects in the conservative scenario. Benefits obtained by increasing the number of objects in the bundle diminish, and savings stabilize after about 5,000 (10,000) objects in the conservative (optimistic) scenario.

From Fig. 6 (bottom), the fraction of total saved requests present similar trends as for volume savings. However, in this case *Max-Req* is the optimal strategy that already allows to save 8.2% (30%) of requests in the conservative (optimistic) scenario with a bundle of only 1,000 objects.

Overall, results show that a few thousands of objects are sufficient to achieve some savings. But how large should the bundle be? Intuitively, the smaller the bundle, the lower the requirements of the broadcast capacity, energy and local storage on users' devices. Fig. 7 shows the relation between the bundle size and the number of objects in the bundle. Notice the log scale on the y-axis. We see that the *Max-Vol* strategy is not practical. In fact, to achieve 5% of savings in the conservative scenario about 200 objects are required, i.e., a bundle size of 500 MB. This becomes more than 3 GB in the optimistic scenario! We stress that the bundle size, i.e., the local cache size on mobile terminal, is a severe technological constraint. Conversely, with only 10 MB (corresponding to 1,000 objects) the *Max-Req* strategy achieves 5% and 7% of savings for volume and requests respectively in the conservative scenario. Since popular objects are typically small (22 kB on average), the *Max-Req* strategy allows to bundle more objects than the *Max-Vol* strategy. This allows to increase the hit probability and the achievable savings for a given bundle size constraint. The *Weighted-Vol* strategy presents almost identical performance to the *Max-Req* up to bundle size smaller than 100 MB. Overall, the *Max-Req* strategy represents a simple and effective solution to achieve some savings for both volume and number of requests. We now focus only on the *Max-Req* bundling strategy which proves to be practical and effective.

We note that reported results do not consider the effect of aborted downloads. Considering the top 10,000 cacheable objects, for which we can assess the actual size using active experiments, we indeed verified that such an effect is marginal (99.6% of such objects are smaller than 1 MB and do not present aborted downloads). Unfortunately, we cannot generalize this result since we do not have the actual size of all requested objects, and we leave a more precise study of this effect as future work.

Full day: To complete the analysis we evaluate the hourly savings for the whole day in Fig. 8. The top plot refer to volume savings while the bottom plot shows the fraction

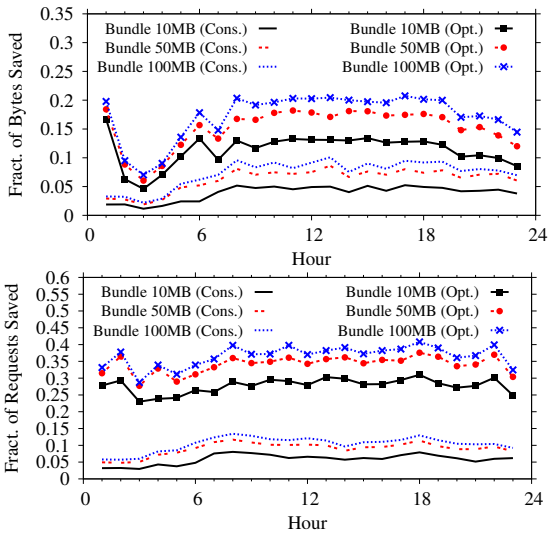

Figure 8: Fraction of saved volume (top) and requests (bottom) over the day.

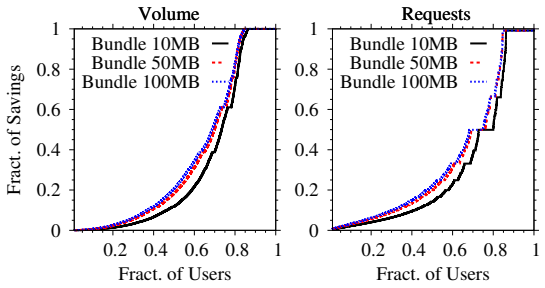

Figure 9: Fraction of volume (left) and requests (right) saved per user. Conservative scenario.

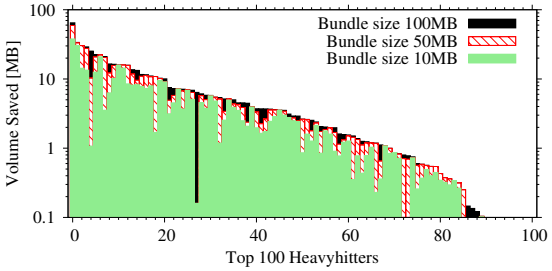

Figure 10: Saved volume by the top 100 heavy hitter at peak time (6pm to 7pm).

of saved requests. We can see that up to 10% (20%) of volume can be saved in the conservative (optimistic) scenario using a bundle size of 100 MB. For the same bundle size, the fraction of requests saved is higher and tops 13.2% (40%) in the conservative (optimistic) scenario. Moreover, with the exception of day/night shifts, savings are surprisingly stable across the day. This consistency is related to specific characteristics of the content accessed from 3G networks as discussed in Sec. 3.

4.2 Savings for the User

We conclude the analysis investigating the benefits of content pre-staging for end-users. We first investigate what is the fraction of users that can benefit from pre-staging. Results, not reported here for sake of brevity, show that even for a bundle of 1 MB, 49% (79%) of customers would have used at least one bundled object in the conservative (optimistic) scenario.

Focusing on these users, Fig. 9 details the CDF of the fraction of volume (left) and requests (right) they can save. Results refer to the conservative scenario during peak hour (similar results are true for other hours). Consider saved volume first. We can see that 20% of users have a negligible benefit (<<1%) for almost any bundle size. These users are typically heavy hitters, i.e., users downloading a small number of very large (and unpopular) objects. On the opposite side, 15% of users presents a 100% of savings, i.e., they would not need to download any content. We found the majority of these customers using a GPS/navigation service that was downloading/accessing map related objects. As these are popular objects, they fit in the bundle offering "offline" navigation for free. Overall, with only 10 MB of bundle, 50% of customers can save more than 10% of volume and larger bundle size permits further savings.

Similar considerations hold for the number of saved requests (right plot in Fig. 9). Notice the horizontal steps in the figures corresponding to popular objects accessed by a large fraction of users, e.g., the map objects previously mentioned.

To conclude the analysis, Fig. 10 reports the absolute amount of volume saved by the top 100 heavy hitters at 6pm. We can notice that a bundle of 10 MB is sufficient to guarantee most of the savings. Observe that not all heavy hitters can take advantage of the pre-staging. In particular, 20 of these top 100 users present negligible savings. As previously said, they access large objects that are not popular. For such cases, pre-staging does not represent an optimal strategy for saving network resources.

5. DISCUSSION

In this paper we considered the opportunity to exploit the peculiarity of the wireless broadcast channel in 3G/4G networks to pre-stage popular content to mobile terminals. Using actual traces collected from an operational network, we analyzed the savings such a system would offer if deployed. In an optimistic scenario where all popular content can be bundled, benefits for both the wireless network and the end-users are significant with a reduction up to 20% of downloaded volume for a bundle size of 100 MB. However, considering only objects that are nowadays cacheable, savings reduce to about 10% for the same bundle size. One can argue whether this is enough to justify the engineering of the system.

First of all, according to recent estimates, mobile carriers are expected to invest 10.5 billions of dollars in network infrastructures in 2013 to keep pace with the ongoing increasing requirement of resources[2]. In this context, even a 0.5% of traffic saved can correspond to millions of dollars. Fig. 11 shows a rough measure of Return Of Investment (ROI) defined as the ratio between the volume saved (i.e., the benefit) and the bundle size (i.e., the cost). We can see

[2]US Carriers Will Outlay $10.5 Billion for Network Investments in 2013 as they Chase the 4G Dream. `http://goo.gl/9FZau`

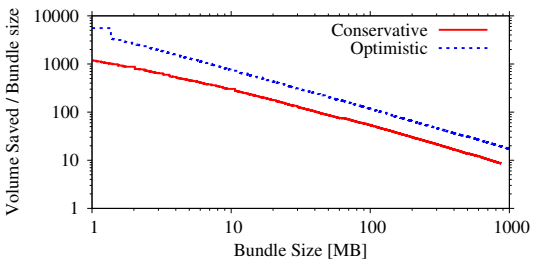

Figure 11: Return of Investment at peak time.

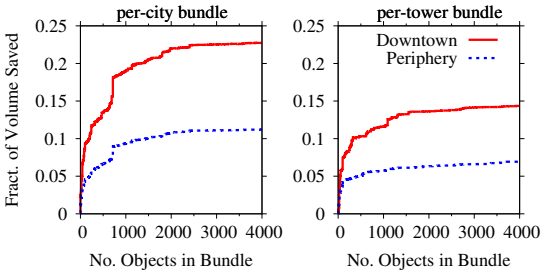

Figure 12: Comparison of per-tower and per-city bundle, optimistic scenario.

that by devoting a little fraction of capacity it is possible to obtain high returns.

Savings are also strictly related to content characteristics. On the one hand, if content becomes more and more personalized, then the return would diminish. On the other hand, content owners can cooperate with the system, increasing the content cacheability and system efficiency. It is even possible to envision a scenario in which content owners suggest to the system which content to bundle (e.g., pushing application/system updates during the night). This calls also for standardization efforts. In such a case, savings would be enormous.

Another direction worth investigating are temporal and spatial correlations. We considered two towers (downtown vs periphery) appearing in the top 10 cells ranked with respect to both number of users and volume served. Fig. 12 (left) reports the savings using a per-city bundle, i.e., bundling objects found to be popular considering the city-wide population. Instead, Fig. 12 (right) reports the savings using a per-tower bundle containing popular objects among users in that cell. One would expect the latter strategy to provide more customized content and thus better savings. Surprisingly, this turns out not to be the case, the reason being that the per-tower popularity estimation results are less accurate and less stationary than the per-city one.

In this work we considered a bundling time window of 1 hour, a reasonable value based on the observed characteristics of the content in the data set (see Fig. 5). However, a more fine-grained sensitivity analysis could better pinpoint the implication of this parameter on the achievable savings. Moreover, when generating the bundle, we have not considered the effect of previous broadcasts. In fact, in a real implementation the local cache suppresses some of the users' requests, which raises the question of how to properly estimate popularity. We can consider different strategies to address this issue. For instance users' terminals can help the system to compute popularity by sending reports about local cache hits, or content providers can specify which content has to be bundled. Moreover, by comparing the popularity of objects included and not included in the bundle, the system can automatically identify new hot content. At last, we can envision a system exploiting some statistical models to predict content popularity. We leave the evaluation of all these aspects as future work.

Finally, other optimizations can be done as well. For example, content with very long lifetime can be broadcast less frequently, reducing broadcast bandwidth usage; similarly, compression and delta encoding among consecutive bundles can be easily built-in. Considering mobile terminal requirements, local cache size can be optimized as well (e.g., by discarding the content that with high probability will not be accessed due to a different operating system). Energy impact has to be carefully evaluated too. In particular, a recent work shows that apps using push-notification mechanisms and running in background can significantly contribute to battery draining [9]. Unfortunately, the data set we used does not include any information about the users' devices radio state and cannot be used to investigate energy consumption. In conclusion, pre-staged content would cause a reception cost and further analysis is needed to quantify such effects.

6. ACKNOWLEDGEMENT

We thank the anonymous reviewers, the shepherd for their constructive comments, Prashant Kumar and Guy Loureiro. This work is partially funded by the mPlane EU-IP project supported by the European Commission under the grant n-318627.

7. REFERENCES

[1] E. Nygren, R. K. Sitaraman, and J. Sun. The akamai network: a platform for high-performance internet applications. *SIGOPS Oper. Syst. Rev.*, 44(3):2–19, August 2010.

[2] J. Erman, A. Gerber, M. Hajiaghayi, Dan Pei, S. Sen, and O. Spatscheck. To cache or not to cache: The 3g case. *Internet Computing, IEEE*, 15(2):27–34, 2011.

[3] B. D. Higgins, J. Flinn, T. J. Giuli, B. Noble, C. Peplin, and D. Watson. Informed mobile prefetching. ACM MobiSys '12.

[4] S. Traverso, K. Huguenin, I. Trestian, V. Erramilli, N. Laoutaris, and K. Papagiannaki. Tailgate: handling long-tail content with a little help from friends. ACM WWW '12.

[5] X. Bao, M. Gowda, R. Mahajan, and R. R. Choudhury. The case for psychological computing. ACM HotMobile '13.

[6] CES 2013: Verizon Eyes Broadcast Over LTE for Super Bowl 2014. `http://goo.gl/vPfVxX`.

[7] F. Qian, K. S. Quah, J. Huang, J. Erman, A. Gerber, Z. Mao, S. Sen, and O. Spatscheck. Web caching on smartphones: ideal vs. reality. ACM MobiSys '12.

[8] RFC 2616 - Hypertext Transfer Protocol - HTTP/1.1. `http://www.w3.org/Protocols/rfc2616/rfc2616-sec13.html`.

[9] A. Aucinas, N. Vallina-Rodriguez, Y. Grunenberger, V. Erramilli, K. Papagiannaki, J. Crowcroft, and J. Wetherall. Staying online while mobile: The hidden costs. ACM CoNEXT '13.