Proactive Caching for Hybrid Urban Mobile Networks

Afra J. Mashhadi, Pan Hui
Deutsche Telekom Laboratories
Ernst-Reuter-Platz 7
10587 Berlin, Germany
(Paper ID: 1569334407 , 12 Pages)
{A.JahanbakhshMashhadi@cs.ucl.ac.uk | pan.hui@telekom.de }

ABSTRACT

Consuming digital multimedia (such as videos) on move has become evermore popular, all thanks to the widespread success of powerful, networked handheld devices as well as availability of 3G services in urban areas. The storage sizes and Wi-Fi networking capabilities of such devices have made them a good platform for opportunistic content sharing; however given the bulky nature of multimedia files, the question arises as to how we can increase the number of successfully serviced requests by caching content locally.

In this work we study the state of classical caching and show that while those strategies are considered to be *good enough* in their intended domain (such as web proxies); they fail to perform effectively when applied to mobile networks due to lower usage. Therefore, we propose an opportunistic proactive caching strategy which exploits available access points to *proactively* push contents to nodes through Wi-Fi. We demonstrate the effectiveness of this approach, in terms of successful cache hit rate by means of simulation using large-scale real world traces. In addition, we show that up to 70% of content requests can be successfully satisfied by the proposed opportunistic proactive caching, cutting the delivery up to half its original perceived delay.

1. INTRODUCTION

Mobile devices have undergone a major evolution. The new generation of mobile phones (e.g., iPhone, Android- powered devices, etc.) has seen their computing and memory capabilities grow significantly and in line with Moores Law. A variety of functionalities have become a commodity, including multiple network interfaces of increasing bandwidth (e.g., 3G, Wi-Fi), thus enabling users to consume content on the go. Indeed it is forecasted that by 2014, 500 million Wi-Fi Certified handsets will be available [1]. The area of Delay Tolerant Networks [1] has been introduced to exploit availability of such networked devices in proximity (i.e., nodes), facilitating opportunistic downloading from collocated participants. With the recent launch of Wi-Fi Direct[2] such opportunistic downloading has taken yet another major step forward. In this work, we consider an opportunistic content sharing system built over a network of participatory mobile nodes, where users download content either opportunistically from each other (be it a mobile node or an available access points), or directly through 3G after a given patience time.

To illustrate the need for opportunistic downloading, we can describe a scenario where many users with similar interest (e.g. sports) are geographically located in an area for a relatively considerable duration of time, such as London 2012 Olympics. We assume all users to have handheld devices with limited yet decent storage sizes, as well as having space and time for sharing content on move. The target content type is delay tolerant in nature and considered relatively sizeable, such as multimedia content.

In such a scenario, opportunistic downloading can be useful where some users do not have access to 3G, either due to cost of such services (e.g., roaming charges for tourists) or due to service limitation (e.g., overcommitted networks or download limitations due to the fair usage policy placed by operators such as Vodafone[3]). However, in order for opportunistic downloading to take place, content must be locally available at the encountered nodes. Given the limitation on storages of mobile devices as well as bulky nature of multimedia contents, a challenging question arises regarding the content that should be cached in order to successfully respond to futures' opportunistic content requests.

During the past decade due to the development of web-based services, caching replacement strategies for web proxies has been an active area of research. As a result, proxy caching is used to reduce network bandwidth usage, user perceived delays and load on the origin servers [2].

However due to various factors influencing the Web, many researches have since concluded that the proposed caching replacement strategies are already *good enough*

[1]http://www.wi-fi.org/news-articles.php?f=media-news&news-id=969
[2]http://www.wi-fi.org/news-articles.php?f=media-news&news-id=909
[3]http://www.vodafone.com.au/personal/aboutvodafone/legal/fairusepolicy/

[3]. The two main factors are: firstly, the ever increasing caching capacity of web servers, and secondly the dynamic nature of Web 2.0 [3] . The former puts forward the argument that capacity growth ensures that replacement strategies are not a limiting factor for caching. However, while this argument maybe true in the domain of Web services, it does not fully apply to mobile networks where storage limitations still persist on handheld devices.

The second factor contributing to the abandonment of more research on caching replacement strategy is the nature of the traffic which is to be cached. The dynamic and fast nature of user generated content in Web 2.0, certainly makes caching replacement strategies less useful as the content will frequently be modified. Even though this factor also impacts caching in mobile networks, it is a lesser problem where content is delay tolerant in nature, as established in our scenario (e.g., videos from YouTube). This especially holds where users do not have a direct connection to Internet through Wi-Fi or 3G but are satisfied with a cached version of potentially out-of-date content.

Therefore in this work, we investigate the impact of applying classical caching strategies to mobile networks. We show that while those strategies are considered to be *good enough* in the web domain, they fail to have the same impact for mobile networks where caching is highly influenced by network topological properties. The main contribution of this work is addressing classical caching limitations and proposing a fully distributed opportunisticlly proactive caching which well adapts to the needs of mobile environment. We evaluate our proposed approach by means of simulation by using large scale real mobility traces of an urban city, as well as realistically modeling users request behaviour and content distribution according to prior research on Web 2.0. Catering for user experiences, we show that the opportunistic proactive caching proves to be successful in reducing user dependency to 3G network (potentially cost beneficial both for users and network operators) as well as reducing the encountered delay. To the best of our knowledge, our work is the first to thoroughly analyse the state of caching for mobile networks in a large scale realistic environment.

The remainder of the paper is structured as follows: first, we briefly describe the state-of-art of caching in web proxy domain in Section 2. We then describe how such caching strategies can be applied to mobile networks in Section 3, before moving to Section 4 where we present the results of a comparative evaluation of some of those popular caching replacement strategies (Section 4.2); demonstrating where they all suffer from shortages impacting their effectivity of caching. We then propose and evaluate our proposed opportunistic proactive caching approach in Section 4.3; and further evaluate it under the described London Olympic scenario in Section 4.4. We then position the proposed approach with respect to related works in domain of mobile cooperative caching in Section 5, before presenting our concluding remarks in Section 6.

2. BACKGROUND TO CACHING REPLACEMENT TECHNIQUES

The survey of web cache replacements [3] groups the caching replacement strategies into three main categories: *Recency* based, *frequency* based and finally *recency and frequency* based strategies. Common to all categories is that when the cache is full, an already stored object (data) is deleted from the cache space in order to make free space for future objects. In here we briefly describe each category, selecting a strategy from each category for our comparative analysis:

- Recency Based: these strategies use time as the main factor to decide which object in cache should be discarded. The most known strategy of this type is LRU or Least Recently Used, which removes the least recently referenced object. LRU is used in many different areas such as thread scheduling , databases and so on. The idea behind recency based approaches such as LRU is that the objects that have been recently requested are more likely to be requested again within short time.

  We select LRU for its simplicity and adapt it to the described mobile network settings. In such cooperative mobile setting, the references to a cached object comes from both the device itself and the requests of users in the devices' vicinity. Hence we expect LRU to fit well in such environment as it can integrate and react to mobility of users: if a user is currently in a public place encountering many other devices, he is more likely to be asked for the cached popular contents multiple times in short duration of time while still in the vicinity of devices.

- Frequency Based: these caching replacement strategies use the frequency of an object request as main factor. The most known of these strategies is LFU or Least Frequently Used, which removes the least frequently requested object. We select LFU algorithm for our comparative analysis and adapt it to our settings as before by considering the references (i.e., requests) to an object to come from both the device itself and the neighbouring nodes.

- Recency/Frequency Based: strategies in this category take both time and popularity in to account. LRU* [4] is one of these cache replacement strategies which we will describe in Section 3.3.

We may also consider *Function-based* strategies. These strategies take into account properties such as size and cost to calculate the value of cached objects as a basis for replacing the least valued object. Defining the cost of each object is non-trivial and highly tied to the application domain (e.g., cost of a request remaining unsatisfied). For this reason we will focus in the context of this paper on time and frequency based approaches, though propose potential ways of defining such function-based strategies using social network reasoning as a basis for future work, as discussed in Section 6.

3. SYSTEM DESCRIPTION

In this section, we detail the process of requesting and obtaining content in the network in order to provide a background to the concept of opportunistic downloading. We then model the described network in terms of content population and user behaviour and further describe how we apply the selected classical caching strategies to the mobile network as a basis for the comparative evaluation in Section 4.

3.1 Opportunistic Download

When a user requests content (e.g., a video), the local cache of the user is first searched to determine whether or not the content may be already available. If not a *patience time* is assigned to the request: this time indicates an initial limit that the user is willing to tolerate before obtaining the content via the standard network (e.g., 3G). Should the user encounter other neighbouring nodes (i.e., other user devices) during this time, the request will be passed on in the hope that they may have a cached copy of the content available. This would constitute an opportunity for an opportunistic download to take place, via Wi-Fi.

We refer to measurements in [5] and assume that the download rate using Wi-Fi between any pair device in range is 6 Mbps. However should the opportunistic download not take place during the patience time (i.e., due to the lack of encounters or cache miss on other devices), then the content is eventually downloaded through 3G with downloading rate of 600Kbps as measured in [6].

3.2 Content Distribution

In order to realistically model users' downloading behaviour and desire for content, we refer to the analysis of content popularity and properties conducted for the online video provider YouTube. In [7], authors studied the size distribution of the YouTube videos, demonstrating that this follows a long tail distribution with many smaller files and fewer larger ones; corresponding to previously observed file size distribution of Internet traffic. Their measurements present that more than 40% of videos were within four minutes in duration and within 5 MB, reflecting that YouTube is primarily a site for very short videos (the study was done when there was a maximum 10 mins video length enforcement by Google).

Given that in our scenario we also consider users requesting content in the form of videos, we will hence follow the same size distribution as YouTube for modeling content population. That is, 40% of content will be of size 5 MB, 30% will be 10 MB, 20% 15MB, and finally the bulky 25 MB content will correspond only to 10% of the population.

Alongside content size distribution, we are also required to model users request behaviour (i.e., preferred user content). We do so by modeling the correlation between user requests and content popularity, following prior research on YouTube. In [8], the authors investigated the popularity of YouTube content on a large scale and have shown that the video requests are highly skewed towards popular files, following the Pareto Distribution [9], with 20% of the files satisfying 80% of the requests. We hence adopt the same mapping between requests and popularity of files (formal model is provided in Section 4.3.3). In doing so we assume no correlation between content size and popularity. Furthermore, in this work we assume the existence of a static non-aging content distribution model in terms of the popularity measure, and the effect of content evolution remains as part of our future work.

3.3 Adapting Caching Strategies to Mobile Networks

Common to all the chosen cache replacement strategies is that they are triggered locally when the total size of cached objects exceeds a predefined upper threshold, H. Once triggered, the caching replacement strategy discards as many objects as indicated, in order for the total size to fall below a predefined low threshold, L (with $L < H$). For simplicity and efficiency reasons, we assume H to be equal to the storage capacity and L to be a content element (i.e., single video) less than storage capacity. Indeed, this design decision addresses the rarer cache insertion events in mobile networks in comparison to web proxies and operating systems, hence allowing us to maximise the number of cached elements at each node.

As previously described, in order to adapt the caching strategies to mobile networks, we assume the references (i.e., requests) to cached content are made both locally (user's own requests) and globally (from nodes in vicinity). The actions of any caching strategies can be broken into refreshing and replacement procedures. Refreshment procedure is triggered when a request is issued. If the request is a hit (the requested content exists in the cache), the following actions happens for each different caching replacement strategies:

- LRU: the content is moved to the head of the list, reflecting that it has been recently accessed.
- LFU: the content's counter is incremented.
- LRU*: the content's counter is incremented and it is moved to the head of the LRU list.

However if the request is a miss (i.e., the requested content did not exist in the cache), caching strategies would traditionally indicate that content should be fetched. However in adapting those strategies to mobile networks, we assume that no action is taken upon a miss: the content will eventually get downloaded either opportunistically or through 3G after the established patience time.

If the miss request was from a node in vicinity, then there is no incentive for the receiver of the request to allocate its power and other possible costs downloading the content through 3G for someone else. Hence we assume the requester will eventually download and place the content in its cache even though this event may not happen immediately.

The second procedure which distinguishes different caching strategies is replacement. Replacement is triggered when new content is downloaded and the cache size is bigger than H watermark (i.e., in our case when the cache is full). The following strategies define which piece of content would be nominated for replacement:

- LRU: delete from the end of the list and place the newly downloaded element in the head.
- LFU: delete the content with the least value for the counter, set the frequency of the newly downloaded content to 1 and insert to the storage.
- LRU*: iterate from end of the LRU list:

  If the contents frequency counter is zero then delete;

  otherwise decrement its frequency counter and place at the head of the list.

  Finally insert newly downloaded content in head of the list, setting its frequency counter to 1.

4. EVALUATION

In this section we present our insights to caching in cooperative mobile network. We first describe our simulation settings in Section 4.1. We provide a comparative evaluation of the selected classical caching strategies and investigate the issue of *is classical caching good enough for mobile networks* in Section 4.2. From the obtained results, we claim that in such networks caching is highly influenced by users's requesting behaviour and encounter rate; hence we propose a more suited caching strategy for mobile networks, by modeling caching proactively (Section 4.3).

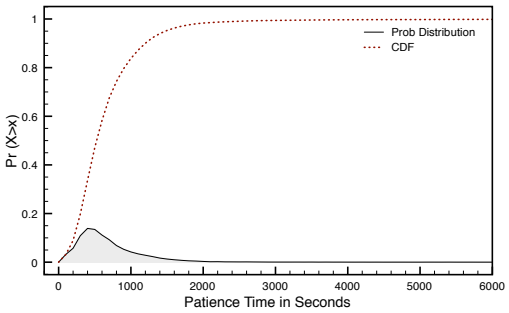


Figure 1: Patience time (in seconds) for requests issued by 100 nodes in the network

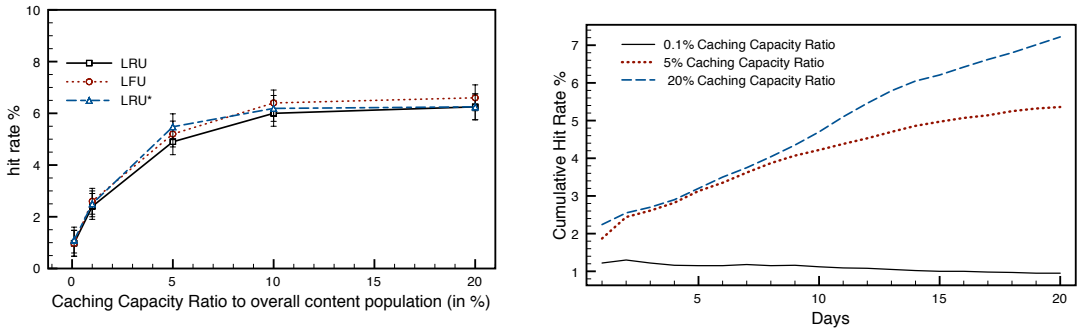
4.1 Simulation Settings

All the evaluations have been conducted by means of simulation, and averaged across multiple runs. We used the previously described content model (Section 3.2), and assumed a fix population size of 10000 pieces of content available to the users. In terms of mobility, we required traces to be reflective of a real urban scenario, therefore we have chosen the only available large scale mobility traces, the San Francisco Cab traces [10].

The traces recorded the GPS coordinates of 500 cabs, logged every 10 seconds, over a period of 20 days, in the San Francisco Bay Area. We model each cab as a mobile node in the network. In order to infer colocation information from GPS coordinates, we have assumed that two cabs are colocated if their physical distance is less than 100 meters (i.e., within Wi-Fi range). Furthermore, the dataset contains information about cabs occupancy (i.e., the time a cab becomes occupied and the time it becomes vacant). Even-though in our simulations we model the cabs as nodes (users carrying one unique device), we use the occupancy information for modeling content requests. We simulate the content requests for every user by assuming that a request is issued every-time a passenger enters a cab, and the patience time is set to the duration of the cab being occupied, as previously modeled in [11]. Figure 1 presents the patience time distribution for requests issued by subset of 100 nodes during the course of 20 days. As it can be seen 80% of the requests have patience time of less than 15 mins (i.e., 1000 seconds in the graph), with the average patience time being around 650 seconds (i.e., 10 min).

4.2 Is Classical Caching good enough for Mobile Networks?

In order to answer this question, we break our evaluations into two parts: we first analyse the classic approaches under various caching capacities, we show that only for a range of scenarios caching has an impact. We then compare those strategies under different net-



(a) (b)

Figure 2: Caching performance in mobile network: (a) The hit rate of classical caching strategies (b) The cumulative hit rate over time for LFU

work sizes and show that a frequency-based strategy can have potentially a higher impact when more users participate.

4.2.1 *Effect of caching on a mobile network*

We evaluate the adaptation of classic caching approaches described in Section 3.3 for a network of 100 randomly selected users from the Cabs traces.

Figures 2a and 2b present hit rate and time-based cumulative hit rate for various strategies, respectively. In Figure 2a, the x-axis presents the ratio of nodes caching capacity to the total content population size, whilst Y-values present efficiency of caching (hit rate) in percentage. As it can be observed, for smaller caching capacity ratio (i.e., where node can only store small percentage of overall available content), the hit rate of all the classical approaches is negligible (less than 1%). However as the caching capacity ratio increases and nodes accumulate more contents locally, the hit rate increases getting up to 6%. However this incline in hit rate only applies to the caching capacity ratio of less than 10%, and all the protocols start to flatten out afterwards. In order to understand this insight better we have plotted the hit rate percentage for the LFU caching strategy on timely basis throughout the course of 20 days. Figure 2b presents this result for capacity ratio of 0.1%, 5% and 20%. As expected, the hit rate is low at the beginning of the simulation (i.e., day 1), reflecting users empty caches which in turn cause missed requests. As time increases and nodes fill up their cache by requesting content, we observe that the cache hit rate stabilizes for case of smaller cache capacity (0.1% and 5%). The opposite is observed for bigger caching capacity in which the hit rate grows according to time, reflecting potential space in the nodes' cache. Indeed our analysis of Cab traces shows on average 15 requests per day for the same subset of nodes, requiring on average 7 GB storage each for period of 20 days.

4.2.2 *Recency Vs Frequency Based strategies*

In order to highlight the differences between the presented classic caching strategies, we compared them under different network topologies. For this evaluation, we stayed with the caching capacity ratio of 5% as it was proven to be big enough to have an impact on hit rate while being small enough to get full during a period of 20 days, under our system model.

We compare the frequency based strategy (LFU) versus recency based strategy (LRU), omitting LRU* as it has a high processing cost associated to it, due to continuously rearranging the queue on each miss.

Figure 3 presents the hit rate for various network sizes. As it can be observed LFU's performance gap to LRU widens as population increases with the hit rate performance, converging and stabilizing for both protocols.

For the network of 300 nodes, LFU reaches peak performance having obtained enough popularity knowledge from neighbouring nodes, and effectively caches only the most popular contents. However after that point as the population increases to include more nodes, not only the caching quality does not improve further, the hit rate also slightly declines. Indeed as network population increases more nodes become collocated (on average) causing the number of requests received per node to increase. This in turn increases the number of missed requests when the caching capacity is not big enough to include all available content.

4.2.3 *Conclusion*

To sum up our observations, we claim that classical caching has negligible impact when the caching capacity is small in ratio with the overall available content population. This means either small storage sizes or enormous available content population (or both). In terms of the former one, research has shown that storage size for mobile devices has grown according to Moors

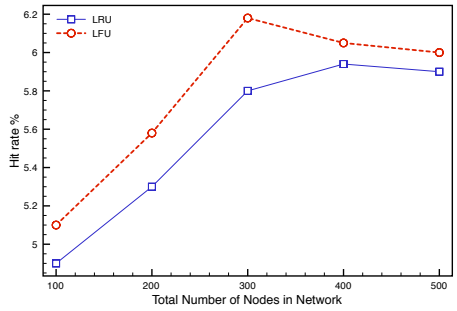


Figure 3: Comparison between frequency and recency caching strategy for increasing network size

Law, allowing many of mobile phones in todays' market to have more than couple of Giga Bytes storage (with iPhone and Android phones having up to 30 GB storage). While it is perhaps argumentative to blame the low capacity ratio on the storage size of todays devices, it is very likely that the latter reason (the enormous content population) plays a major role. Let us assume devices each with 10 GB storage size, in order for the caching capacity ratio to be less than 1%, users must request from 1 TB content pool at a relatively short period of time (couple of weeks). We claim that for the described scenario and by considering humans locality of movement, it is unlikely that users belonging to a local community request 1 TB different content in a short period of time (weeks). While in this work we do not model dynamic properties of contents (such as aging and popularity evolution), it can nevertheless be assumed that at each point of time only a subset of a bigger universal pool of content is of interest amongst users [12, 8].

We further showed that LFU can be a better caching strategy due to catering for the popularity of contents. However, for largely populated networks, there is perhaps little difference between the performance offered by LRU and LFU as they both saturate by number of received requests (references to cached contents). However both protocols under ideal network size and caching capacity ratio still suffer from low hit rate. We showed that this is due to the low usage pattern (i.e., requests issued from users on the go) together with no miss replacement action. Therefore, for caching to work effectively, a mechanism to proactively cache contents at nodes regardless of their usage pattern is needed.

4.3 Opportunistic Proactive Caching Strategy

In this section we discuss how the classic frequency based strategy can be adopted to mobile networks where the performance is extremely dependent on the network topology (in terms of number of nodes, user request patterns and mobility). We built our approach upon the LFU protocol as it brings a popularity dimension which can be used by available access points in the network.

We previously showed that the main cause of a poor cache hit rate was the low number of requests issued by users together with the absence of any immediate action to replace the missing content. As described earlier, a node has two options upon a cache miss event from a non-local request (i.e. from another neighbouring node), both of which we feel are not appropriate courses of action. The node can either download the missing content via 3G, which would be unreasonable due to potential cost and absence of direct incentive, as the requesting node would download it via 3G eventually. Alternatively the recipient of the request can try and download it opportunistically, but in this case little gain is expected as both nodes are likely to be in the same range as any neighbours holding the cache content.

In order to facilitate an effective caching strategy that is independent from the requesting patterns of users, and to compensate for the lack of cached content, we introduce a proactive caching approach which aims to exploit storage availability on devices: this strategy will *proactively* (i.e., without nodes requesting for the content) push content to nodes for caching purposes. We also rely for this purpose on the availability of access points (i.e., APs) common in today's urban cities, possible by Broadband Sharing schemes [13, 14] and assume that they are available to all users in the network.

We model here access points as fixed nodes, monitoring network requests and downloading missing content directly from their gateways to Internet (assuming downloading is immediate with no communication delay). In our evaluation, we assume that access points have enough storage to hold a history of requests and apply the same frequency based strategy to identify the popularity of different pieces of content from their local community. The access points then *proactively* push popular content to users in their Wi-Fi range, assuming that such transmission can be done transparently and without the user's need to accept the content. Relying on the described architecture, we detail our proposed proactive caching approach next.

4.3.1 *Proactive Caching Strategy*

The access points' behaviours can be distinguished as follows: *Reactive* and *Proactive*. An access point behaves *reactively* when it receives a content request by a node in its range. The requested content is opportunistically transfered to the node for the duration of the colocation. Once there is no more requests from the user, the access point *proactively* pushes a selected set of contents into the node's cache. This selection is primarily based on content popularity. The static nature

of access points allows them to encounter more nodes in comparison with mobile users on average [15, 16], which in turn allows them to obtain a better estimate of a communities' interest.

This means that for all the content that is proactively pushed into a node's cache by access points, it is likely that they have a relatively higher frequency counter in comparison with reactively downloaded ones; causing them to be favored by frequency-based caching strategy and be replaced later in time.

The impact of the proposed proactive caching strategy depends mainly on two factors: the availability of the access points, and the selection process of what content is to be proactively cached by nodes. While the availability of access points can be a variable in our evaluation, the selection process needs to be predefined by the protocol. We define content selection to be primarily based on popularity and secondarily based on size. In this work, we concentrate on two basic selection approaches, defined as follows:

- Bulky selection: access points proactively push the bigger popular files to adjacent nodes. This is to maximise the amount of data received opportunistically through Wi-Fi.

- Random selection: in which *any* popular content can be pushed by access points and proactively cached by adjacent nodes.

We focus here on a purely participatory network (i.e., where nodes all proactively cache for each other) , but the potential for refinement is evidently great: we discuss in future work potential selection approaches based on social networking analysis (i.e., who should proactively cache and for whom). We aim however here to first demonstrate the potential performance boosts that opportunistic proactive caching can bring.

4.3.2 *Deployment of Access Points*

We deployed access points for our simulation settings using randomly selected points in the San Francisco Bay area. We have done so by randomly selecting 500 points each corresponding to the GPS coordinates of at least one node at some point during the Cab traces. We then repeated the process 20 times and measured the number of encounters between nodes and access points for each set (presentation of this result is omitted due to space constraints). We finally selected a set closest to the average encounters to present 500 access point locations in San Francisco Bay. Therefore all our evaluations are based on the average case in terms of impact of access points on caching. Nonetheless the observations hold for various other ways of deploying access points, as all our evaluations are based on comparisons within the same topological configuration,

4.3.3 *Results*

As stated earlier, the proactive caching strategy heavily relies on two factors: the availability of access points, and proactive content selection by access points. Accordingly, we arrange our evaluations in two parts; we first measure the impact of access point availability in the network and thoroughly analyse the performance of proactive caching against benchmarks presenting upper bound limits (which we will define later); secondly we do a cross comparison between the two selection strategies and show their impact on the user experience and their offered incentives.

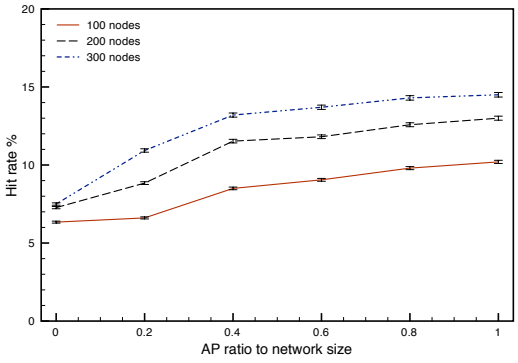


Figure 4: Effect of proactive caching for increasing number of deployed access points

Impact of APs

We first experimented by varying the number of deployed APs in ratio with the network population. Figure 4 presents these results for caching capacity of 10 GB per node.

While it can be observed for all network sizes that the more deployed access points the better the cache hit rate, the incline is reduced when passed the 40% mark. This means, for instance for a network of 200 users, adding 80 access points is adequate and adding more would not dramatically change the caching performance. Such insight can help network operators quantify the required number of access points based on population and hence estimate a minimum cost associated for the proposed scheme.

Benchmark comparison:

In order to get an insight of how good the presented hit rates are in reality, we have conducted experiments with 200 nodes and 80 access points, and measured the performance of proactive caching against two upper bound benchmarks defined as follows:

The *best case theoretical* caching: we define this theoretical approach as the maximum hit rate that can be achieved if the hit rate was independent of the physical network and the environment. This means no factor other than limit on capacity affects the caching. We

mathematically defining this best case as follows:

Let us assume a set P containing contents c_1,c_2,. . . ,c_n and presenting all the universal available contents; denoted by $|P|$ is the size of the population. Similarly a set of popular contents D (the dominate contents), where D $\subset$ P. Let the set C present the cache on each node (with $|C|$ being the caching capacity), and X be a random variable for requesting popular content, we can define the probability of a node asking for a popular content (i.e., Rate of dominative or $Rate_d$) following the Paerto distribution as:

$$Rate_d = Pr(X > x) = \begin{cases} \frac{x_m}{x}^{\alpha} & for \quad x \geq x_m \end{cases}$$

Where x_m is minimum possible value for x and can be defined as probability of choosing a popular content out of all the available contents and formulated as $\frac{|D|}{|P|}$; and α is the Paerto index. It follows from the above formula that the cumulative distribution function is:

$$F_x(x) = \begin{cases} 1 - (\frac{x_m}{x})^{\alpha} & for \quad x \geq x_m \end{cases}$$

As previously described, prior research [8] has shown that content request on domain of Web correlates to content popularity following the 80-20 rule. Hence, choosing α value suitably, the above formula results in request rate for popular content to be:

$$Rate_d = Pr(X > 0.2) = 0.8$$

Based on this model, the cache *theoretical* hit rate is then calculated as follows:

$$\begin{cases} If \ |C| > |D| \quad then \\ Rate_{hit} = Rate_d + (\frac{|C|-|D|}{|P-D|} * (1 - Rate_d)) \\ Otherwise \qquad Rate_{hit} = (\frac{|D|-|C|}{|D|} * Rate_d). \end{cases}$$

Following the above formula, for a cache capacity ratio of 50% (in our data model $|C|$=50), the hit rate would be calculated as 80% + ($\frac{50-20}{100-20} * 20\%$), which is 87.5% hit rate.

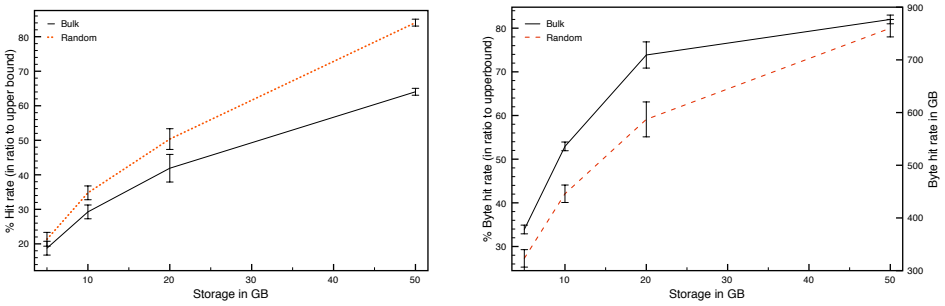
Upper Bound limit: While the theoretical caching puts a far from reachable upper bound on our expectations, it fails to take network properties such as node mobility, request rate, bandwidth and etc. into account. Hence, in order to get a fair expectation for performance of any applied caching strategy, we require a benchmark which takes all the factors concerning mobile network into account while still presenting the best case caching. Intuitively this corresponds to nodes having *unlimited* caching capacity. This means that once they receive contents, they store them indefinitely without any need for replacement (i.e., hence independently from any caching strategy). In such a case, the only factors affecting the hit rate performance are bandwidth, user mobility and distribution of requests. Bearing in mind that such an upper bound would be also unachievable for any caching strategy applied to devices with *limited* storage, we present our results.

Figure 5 presents the theoretical boundary for variable caching capacities. The hit rate for this theoretical boundary is as expected very high. The figure also presents our proactive caching approach under two different states, stabilized and dynamic. The stabilized approach corresponds to a situation where the cold start period (empty cache) was omitted by initializing all caches with *randomly* chosen content at time zero. The dynamic proactive approach on the other hand presents a system starting with empty caches, therefore the hit rate suffers from a cold start period. It is interesting to observe that for the smaller cache sizes, the dynamic case performs similarly to the stabilized case, as caches are small and hence they fill up faster, this also validates our previous observation from the time analysis of Figure 2b. However as the storage capacity increases the stabilized proactive curve converges towards the theoretical case: this is due to the fact that most of the available content is initialized in the beginning in the caches of all nodes, hence converging to a 100% hit rate.

We next evaluate our approach against the second benchmark; the upper bound in which nodes have unlimited caching capacity but are still affected by network properties in Figure 6. The figure also presents the gap performance for both proactive and reactive only (where access points only respond to reactive requests and do not push contents proactively to node caching strategies). As expected the reactive only strategy performs poorly in comparison to a proactive strategy; this is because contents are only inserted to the nodes' caches upon a download request, limiting the performance to users' usage patterns. Hence the hit rate stays flat even for bigger caching capacities. Whereas for the proactive caching strategy, the bigger the capacity, the more proactively content is pushed from access points to nodes, therefore increasing the hit rate. It is interesting to observe that even for a caching capacity of 20 GB, a proactive caching strategy manages to effectively satisfy requests with up to 60% hit rate in ratio to the upper bound.

Pushing Strategies

We next evaluate the effect of different pushing strategies for the same network (i.e., 200 nodes and 80 deployed access points), while varying the caching capacity ratio. Figure 7a and 7b present the hit rate and byte hit rate respectively, for the described pushing strategies. As expected the *Bulky* proactive pushing performs worse in terms of hit rate as it allocates the caching capacity to bigger chunks of data and therefore smaller number of hits per requests. Adversely we expect the Bulky strategy to have an advantage in terms of byte



(a)*Hit rate* (b)*Byte hit rate*

Figure 7: Effect of different pushing strategies on: (a) Hit rate (b) Byte hit rate, both in ratio to the upper bound

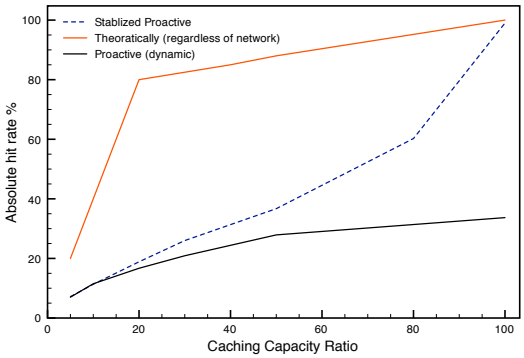


Figure 5: Absolute hit rate comparison with the best case theoretical hit rate in the absence of networking conditions (bandwidth, mobility and request rate)

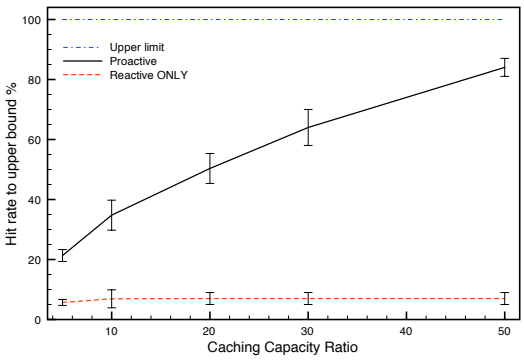


Figure 6: Hit rate performance compared to the upper limit benchmark

hit rate. Figure 7b proves this expectation by presenting byte hit rate percentage for the same value of x-axis. We observe that for the larger caching capacity ratio, while the hit rate gap between Bulky and *Random* approach remains wide, the *Random* approach manages to deliver the same amount of bytes hit as the Bulky approach. This observation is well reflective of the size distributions of content in the network, with only 30% of available content being greater than 10 MB.

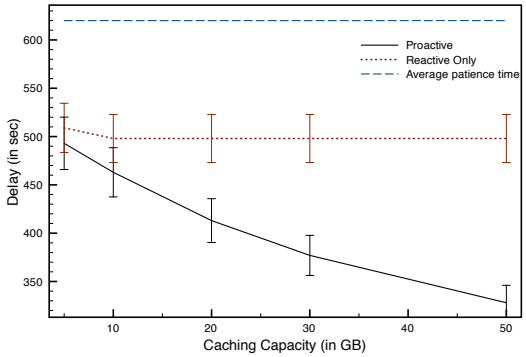
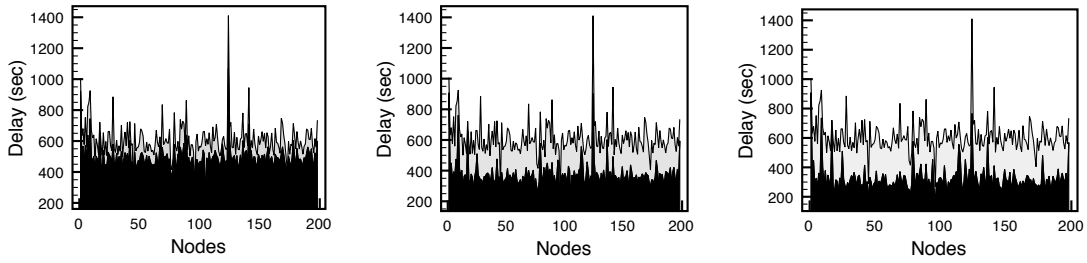


Figure 8: Average delay for caching strategies Vs. the original average patience time

To conclude, we would like to point out the incentives that such caching strategies can offer users as well as network operators. We have thus evaluated the impact of the caching on the delay experienced by users. In doing so we have measured the delivery time from the time that first bytes of the requested contents were received by users. In absence of caching or opportunistic downloading, the delay corresponds to the patience time a node has to wait before downloading the content from 3G network. Figure 8 presents the average delay in seconds for the original network's patience time in comparison with the measured delay for both proactive and reactive only caching. From this figure we can see that the bigger the caching capacity the shorter delay using proactive caching strategy. The delay is cut up to half original patience time for bigger capacities, reducing the users waiting time to on average 5 mins. It is interesting to note that as before where the content is not proactively pushed by to nodes (the reactive only case) the delay remains constant for an increasing ca-



$(a)5GB$ $(b)30GB$ $(c)50GB$

Figure 9: Average delay per node for *proactive* caching (black) in comparison with original patience time (white) for various caching capacities

pacity ratio (due to its dependency to request rate by users). Finally, Figure 9 presents each individual user experience (i.e., delay analysis per node), where the bigger contrast reflects the faster delivery of the content to the users (in comparison to the original patience time).

So far we have presented the impact of proactive caching strategy on users in terms of their perceived delay. We now discuss another potential benefit brought to users as well as network operators. As we observed earlier from Figure 7b, for a modern handset like iPhone or Android phones with an average storage capacity of 20 GB, the opportunistic proactive caching strategy can offer up to 700 GBs of data downloaded through Wi-Fi. This value corresponds to 35 GB of data on average per day for a network with only 80 deployed access points.This amount of opportunistically downloaded data can benefit users who may not have access to 3G network due to either cost or availability of the services, as well as benefiting network operators by offloading traffic from 3G. We next describe and evaluate suitability of our work for such scenarios.

4.4 Reality Check

In this section we evaluate our work for scenarios where dense urban community of users exist, but 3G services are not always available to all the users in the network. For instance events such as the London Olympics where many fans attending the event are geographically collocated, with fans belonging to the local community as well as a percentage of tourists who may prefer to avoid 3G services due to the roaming costs.

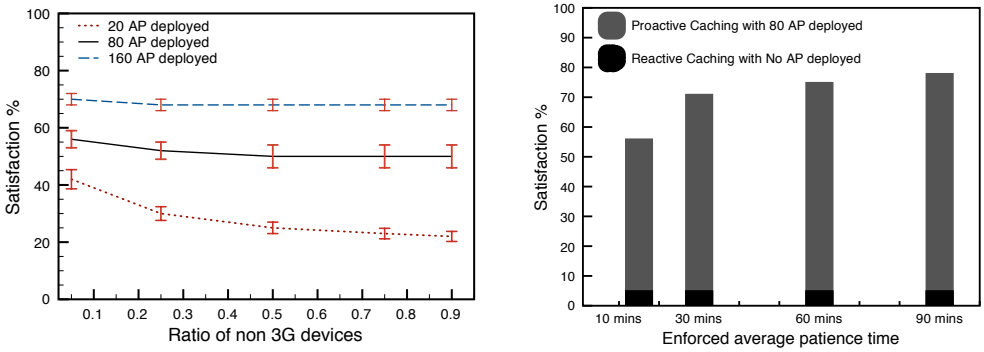
To cater for the described scenario, we have experimented on a real large scale network population of 500 nodes based on complete Cab traces; and introduced heterogeneity to this population by designating 3G services to only a percentage of the nodes. While the heterogeneity applies to networking capabilities, we still assume the storage sizes for all devices are uniform and 10 GB caching capacity per device. We further assume that users without 3G services, or as we refer to them *tourists*, are willing to tolerate more delays and we model their behaviour by extending their requests' patience times (we refer to this extended patience time as *enforced patience time*). While in our evaluation we experiment by varying the enforced patience time, we assume that it is nonetheless constant throughout simulation time and across all requests issued by tourists.

In order to quantify the impact of caching on non 3G portion of the network, we introduce a *satisfaction* metric, which presents the number of successfully received contents in ratio to all the issued requests by tourists. It is worth noting that this metric is only applied to tourists as 3G capable nodes always have a complete satisfaction, due to the capability of downloading their requested content from 3G by the end of their patience time (if not opportunistically earlier).

Figure 10a presents the satisfaction results for the network of 500 nodes with an enforced patience time of 10 minutes (i.e., average user patience time as presented in Figure 1), for various numbers of deployed access points. In this figure the x-axis presents ratio of tourists to the population and y-values present average satisfaction for tourists.

From 10a we observe that for the smaller number of deployed access points, the satisfaction drops sharper as ratio of tourists increases. This is because in the absence of access points in range, the satisfaction would mainly depend on the caches of the local users who download contents from 3G. The adverse is observed for cases with more deployed access points, the satisfaction stays the same no matter the ratio of tourists in the network, making the satisfaction of tourists independent from number of 3G users.

The second observation is the high satisfaction obtained even when the enforced patience time is not highly extended and is same as the average original patience time. To address this result, we have experimented with a network where half population are tourist (0.5 ratio), and have fixed the number of deployed access points in the network to 100 (assuming that number of de-



(a) (b)

Figure 10: Achieved satisfaction: (a) for increasing ratio of tourists to network population, (b) for increasing tourists' delay tolerance.

ployed access points are in 0.4 ratio with the local users only). Figure 10b, presents the satisfaction for various enforced patience time. The result shows that when tourists are willing to tolerate more delay, the opportunistic proactive caching can offer up to 70% perceived satisfaction.

5. RELATED WORK

Cache management techniques for mobile devices and mobile networks have recently become the focus of the research community. In [17], authors propose a greedy caching strategy for scenarios where devices can potentially become detached from the network due to factors such as over committed network bandwidth or geographical areas with no base station coverage. While [17] provides a valuable insight for greedy cache management for mobile devices, it fails to address cooperative caching where caching amongst devices is used in order to maximise a global metric such as number of references served without contacting the base station.

Cooperative caching for mobile networks has been studied in [18, 19, 20]. [20] follows the quest of optimising caching policies to minimise delay by proposing a distributed replication mechanism that yields to an optimal replication ratio for a homogenous network where mobile users meet each other with the same rate.

[19] builds upon [20] by proposing a cooperative caching strategy for heterogeneous networks. It applies a distributed caching replacement approach based on users computed policy in the absence of a central authority and uses a voting mechanism for nodes to decide which content should be replaced. Unlike [20], authors model a heterogeneous network where users have different storage capacity and access the infrastructure at a different rate. Furthermore authors theoretically show that the proposed voting based caching policy is optimal. While [19] formally demonstrates maximisation of social welfare, it is solely based on theoretical proofs and no evaluation of their proposed approach is presented. It would be of interest to see the quantified result when the protocol is used in large scale network as well as the overhead associated to the voting scheme. Our work differs from [20, 19] in modeling users behaviour and network topology corresponding to real scenarios. We introduce an additional step by providing extensive evaluation of the proposed proactive caching as well as various classical caching approaches, under different network topologies, and quantify the impact that caching can have on users experience (both in terms of cost and delay).

Cooperative caching techniques have also been focused for different mobile environments such as wireless home networks. In [18], Ghandeharizadeh et. al. present a novel cooperative caching approach based on asymmetric bandwidth of wireless connections between a handful devices on a home network. We believe their approach is highly efficient for small networks such as the intended home networks, but is not applicable to large scale, disconnected networks such as the urban environment that we addressed in this work.

6. CONCLUSION AND FUTURE WORK

In this work, we studied the effect of caching on mobile networks, where the requested content is delay tolerant in nature. Our contributions were as follows: first, we identified the limitation of classical caching approaches in mobile environments. In doing so, we performed a comparative evaluation of different approaches with the objective of maximising the cache hit rate. We claim frequency based caching strategies are a better fit to mobile networks, due to taking into account the popularity of content as well as for their simplicity. Our second contribution was introducing a more suitable adaptation of the frequency based caching strategy to mobile environment by addressing its limitations. We pre-

sented a quantified analysis of the proposed proactive caching, proving such approach can be highly effective under the modeled real settings. The final contribution of this work is addressing and quantifying the benefits brought to users in terms of user experience by referring to a London Olympic scenario where many tourists will be collocated with local users for a relatively long period of time. For such scenario we showed that users who are not on 3G can still benefit from the network if they tolerate longer delays. Indeed we showed that up to 70% of their requests can be serviced successfully in absence of 3G networks using the proposed proactive caching strategy.

Finally our work is the first of its kind to bring together researches from various fields (such as users' requesting behaviour and content distribution research from Web 2.0 domain, as well as infrastructure availability from research on broadband sharing), to build a realistic model for applying caching strategies.

Our future directions include, extending this work to include dynamic content evolution, where popularity of a content is a function based on its age rather than the binary model used in this work. While in this work we assumed a fully participatory network, in reality participation of the nodes cannot be taken for granted specially given the battery consumption related to proactive caching. Therefore we are interested in applying social networking reasoning to proactive caching, allowing nodes to decide for whom to allocate their storage and proactively cache for (i.e., for whom to participate).

7. REFERENCES

[1] S. Burleigh, A. Hooke, L. Torgerson, K. Fall, V. Cerf, B. Durst, K. Scott, and H. Weiss, "Delay-tolerant networking: an approach to interplanetary internet," *IEEE Communications Magazine*, vol. 41, no. 6, pp. 128–136, 2003.

[2] B. Davison, "A web caching primer," *IEEE Internet Computing*, pp. 38–45, 2001.

[3] S. Podlipnig and L. Boszormenyi, "A survey of web cache replacement strategies," *ACM Computing Surveys (CSUR)*, p. 398, 2003.

[4] C. Chang, A. McGregor, and G. Holmes, "The LRU* WWW proxy cache document replacement algorithm," 1999.

[5] M. Solarski, P. Vidales, O. Schneider, P. Zerfos, and J. Singh, "An experimental evaluation of urban networking using IEEE 802.11 technology," in *Proc. of 1st IEEE Workshop on Operator-Assisted Community Networks*, 2006.

[6] A. Balasubramanian, R. Mahajan, and A. Venkataramani, "Augmenting mobile 3g using wifi: Measurement, design, and implementation," in *ACM MobiSys, San Francisco, USA*, 2010.

[7] X. Cheng, C. Dale, and J. Liu, "Statistics and social network of youtube videos," in *Proc. of IEEE IWQoS*, 2008.

[8] M. Cha, H. Kwak, P. Rodriguez, Y. Ahn, and S. Moon, "I tube, you tube, everybody tubes: analyzing the world's largest user generated content video system," in *Proc. of the 7th ACM SIGCOMM conference*, 2007.

[9] W. Reed, "The Pareto, Zipf and other power laws," *Economics Letters*, vol. 74, no. 1, pp. 15–19, 2001.

[10] M. Piorkowski, N. Sarafijanovoc-Djukic, and M. Grossglauser, "A Parsimonious Model of Mobile Partitioned Networks with Clustering," in *COMSNETS*, 2009.

[11] S. Dimatteo and P. Hui, "MADNet: Metropolitan Advanced Delivery Network," Deutsche Telekom Laboratories, Tech. Rep., 2009.

[12] C. Wallenta, "Analysing and modelling traffic of systems with highly dynamic user generated content," University College London, Tech. Rep. RN/08/10, 2008.

[13] L. Mamatas and I. Psaras, "Incentives and algorithms for broadband access sharing," in *Proc. of 1st workshop on Home Networks (HomeNets)*, 2010.

[14] R. Bhatia, G. Narlikar, I. Rimac, and A. Beck, "UNAP: User-Centric Network-Aware Push for Mobile Content Delivery," *IEEE INFOCOM 2009*, April 2009.

[15] A. J. Mashhadi, "Source Based Content Dissemination in Participatory DTN," University College London, Tech. Rep. RN/09/06, 2009.

[16] P. Hui and A. Lindgren, "Phase transitions of opportunistic communication," in *Proc. of the 3rd ACM workshop on Challenged nets.*, 2008.

[17] S. Shayandeh and S. Ghandeharizadeh, "Greedy cache management techniques for mobile devices," in *1st IEEE Workshop on Ambient Intelligence, Media, and Sensing*, 2007.

[18] S. Ghandeharizadeh and S. Shayandeh, "Cooperative caching techniques for continuous media in wireless home networks," in *Proc. of the 1st conf. on Ambient media and systems*, 2008.

[19] S. Ioannidis, L. Massoulié, and A. Chaintreau, "Distributed Caching over Heterogeneous Mobile Networks," in *Proc. of ACM SIGMETRICS*, June, 2010.

[20] J. Reich and A. Chaintreau, "The age of impatience: optimal replication schemes for opportunistic networks," in *Proc. of the 5th international conference on Emerging networking experiments and technologies*, 2009.