

# Energy Efficient Offloading of 3G Networks

Nikodin Ristanovic  
EPFL, Switzerland  
[nikodin.ristanovic@epfl.ch](mailto:nikodin.ristanovic@epfl.ch)

Jean-Yves Le Boudec  
EPFL, Switzerland  
[jean-yves.leboudec@epfl.ch](mailto:jean-yves.leboudec@epfl.ch)

Augustin Chaintreau  
Columbia University, USA  
[augustin@cs.columbia.edu](mailto:augustin@cs.columbia.edu)

Vijay Erramilli  
Telefonica, Spain  
[vijay@tid.es](mailto:vijay@tid.es)

**Abstract**—The increase in data consumed by smartphones is becoming a huge problem for mobile operators. In three years, mobile data traffic in AT&T’s network rose 5000%. The US operators invest \$50 billion in the data networks every year and the technology upgrades and innovation still fail to keep up with the demand.

In this paper we design two algorithms for delay-tolerant offloading of bulky, socially recommended content from 3G networks. The first one, called “MixZones”, uses opportunistic, ad hoc transfers between users, and is assisted by predictions made by the network operator. The second one, called “HotZones”, exploits delay tolerance and tries to download contents when users are close to Wi-Fi access points; it is also assisted by predictions made by the operator. We evaluate both algorithms using a large data set, obtained from a major mobile operator and a realistic application similar to Apple’s Ping music social network. The metrics address the amount of offloading, delay and mobile energy efficiency.

We find that both solutions succeed in offloading a significant amount of traffic, with a positive impact on user battery lifetime. Surprisingly, we also find that all the benefit obtained from the operator with the MixZones algorithm (i.e with ad hoc exchanges between users) can be achieved with the HotZones algorithm and a small investment in Wi-Fi access points. Note that the latter is considerably less complex to deploy than the former.

## I. INTRODUCTION

The rapid proliferation of smartphones is pushing the existing 3G networks to the limit. Although the backbone capacities are usually sufficient, it is becoming difficult and expensive for mobile operators, with a strong smartphone offer to provide sufficient access capacity to their subscribers. After a series of reported problems [1], AT&T (until recently the only iPhone vendor in the US) purchased \$2 billion mobile bandwidth from Qualcomm in December 2010 [2].

In addition to this, the increase in the amount of video clips, music files and photos available on the Internet is changing the way mobile users search and access content. In two weeks, YouTube users upload 120-years’ worth of movies in IMDb [3]. This user generated content is often served to users through social networks, social bookmarking services and websites

for organization of social news, such as del.icio.us, Citeulike, StumbleUpon, Digg or Reddit [4].

Several studies have shown the Zipf popularity distribution of contents recommended through social networks [5]. This means that popular contents are downloaded without constraints by a large number of subscribers. Such behavior leads to bottlenecks, especially in densely populated urban areas during peak usage hours. This is a strong incentive for operators to offload a part of the traffic from their 3G access networks, while preferably maintaining the ability to charge for data.

From users’ perspective, the availability of affordable data plans and growing popularity of social networks can be mapped into systematic overuse of battery intensive 3G connection and an avalanche of community recommended content. Socially recommended content may not necessarily be needed in real time, however it is always treated as such and downloaded immediately via 3G at a high energy cost. For this reason, in the case of socially recommended content, we propose to users to trade some delay for energy, and thus extend the constrained battery life of their smartphones.

We propose two Wi-Fi based solutions for energy efficient offloading of 3G networks. The focus is on socially recommended, delay-tolerant content. The first solution, which we call the MixZones algorithm, exploits opportunistic exchanges between smartphones, in the areas called MixZones. The second solution, which we refer to as the HotZones algorithm, requires covering a fraction of cells, which we dub HotZones, with Wi-Fi access points. Both solutions replace a part of the costly 3G transfers with Wi-Fi transfers. In both algorithms the problem of high Wi-Fi scanning overhead is solved by the use of prediction, provided by the operator.

We evaluate the algorithms by using a large, operator provided data set, which contains three months of activity and mobility for more than half a million users, in a European capital and its major commuting suburbs. We compare their performances with the real time offloading solution, currently deployed by some mobile operators, which allows users to seamlessly switch between Wi-Fi and 3G (we call it RT Wi-Fi Zones). For the evaluation purposes, we design a realistic application similar to Apple’s Ping music social

N. Ristanovic and A. Chaintreau conducted this work while at Technicolor, Paris, under the French National (ANR) project PROSE.”

network. It allows users to request music, by relying on social recommendation, from a catalogue characterized by Zipf popularity distribution.

Our contributions are the following:

We design two different algorithms for delay-tolerant offloading of large, socially recommended content from 3G networks (MixZones, based on opportunistic transfers and HotZones supported by Wi-Fi access points). We evaluate the algorithms using a large data set and we compare them with the real-time offloading solution currently deployed by operators.

We find that both solutions succeed in offloading a significant amount of traffic, with a positive effect on user battery lifetime. More specifically, we show that prediction and delay (in the order of a few hours) can reduce the battery consumption coming from 3G transfers of delay-tolerant content for up to 50%. We also show that the Wi-Fi coverage needed to offload a significant amount of traffic (80 – 90%) is reduced very quickly (by a factor of 3 to 4) when some delay is tolerated. Finally, we show that both algorithms deliver content with lowest delay during the peak hours, when offloading is most needed.

Surprisingly, we find that all the benefit achieved with the comprehensive, operator supported opportunistic algorithm (MixZones) can be achieved with the less complex HotZones algorithm and a small investment in Wi-Fi access points.

The rest of the paper is organized as follows:

In Section II we present the problem background. In Section III we introduce our offloading solutions. In Section IV we describe the evaluation setup. In Section V we present the performance evaluation results. In Section VI we present the related work. In Section VII we conclude the paper and in Section VIII we discuss some interesting directions for the future work.

## II. PROBLEM BACKGROUND

### A. Mobile Data Explosion

When mobile data was introduced in the early 2000s, operators unsuccessfully looked for applications that would instigate subscribers to use slow 2.5G networks on their voice-centric phones. It was the e-mail application on the first data-centric smartphones that started to reverse the situation. The appearance of iPhone in 2007 finally changed everything and exposed users to rich data services, such as mobile video.

This event transformed users' perception of mobile Internet, but it also transformed the problem of unused capacity in cellular data networks into a problem of enormous growth of mobile data traffic. According to AT&T, from March 2006 to May 2009, mobile data traffic in AT&T's network rose 5000% on the national level. Figure 1 compares the growth in voice and data

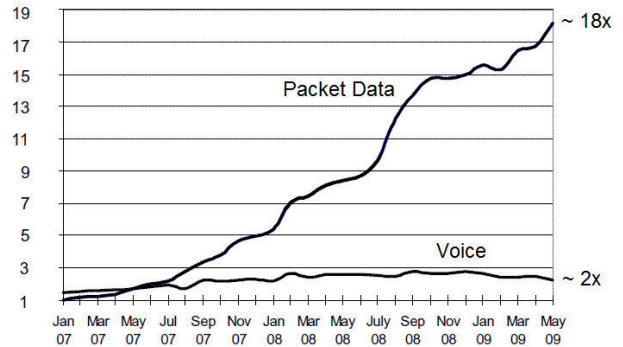


Figure 1. The growth of data relative to voice traffic in North America. The two inflection points for data correspond to releases of iPhone (July 07) and iPhone 3G (July 08). (Source: Rysavy Research)

traffic in the North America, from January 2007 to May 2009. The impact iPhone releases is clearly visible.

### B. Offloading vs. Capacity Increase

The problem the growth of mobile data traffic creates is particularly difficult to solve in the radio access part of the network. The part of the spectrum that an operator has at its disposal is limited and the efficiency of its exploitation depends on the deployed technology.

Building new cell sites or upgrading to new technologies are expensive fixes that have been applied for the past few years. It is estimated that cellular operators in the US invest \$50 billion in their data networks every year [6]. Innovation on the other hand evolves the efficiency of transmission and reception, but it can not eliminate the fact that the number of bits that can be sent in a radio stream is limited. In spite of the continuous effort to deliver higher bandwidth over more spectral efficiency, even the new generation 4G/LTE networks are not capable of serving growing demand in the densely populated urban areas.

An alternative to capacity build-up is traffic offloading through orthogonal solutions. Licensed spectrum femtocells or unlicensed Wi-Fi can allow to increase capacity in an area, while avoiding network bypassing (connections via alternative Wi-Fi networks). Operators want to avoid bypassing and maintain control over data exchanged through the unlicensed spectrum, in order to monetize it. Since May 2010, AT&T has been deploying Wi-Fi access points in areas with consistently high 3G traffic and mobile data use [7]. We compare this solution with our delay-tolerant, prediction based algorithms.

### C. The Challenges of Wi-Fi Offloading

Given the classification in II-B, our approach can be classified as orthogonal offloading through the unlicensed spectrum, namely Wi-Fi. As previously explained, our principal goals are (i) to offload data from the 3G network and (ii) to offer users a possibility to

trade delay for extended battery lifetime. We want to achieve these goals by replacing the energy costly 3G transfers with more efficient Wi-Fi exchanges. The challenges are however the Wi-Fi's limited range, inefficient idle state and high scanning overhead.

The energy consumption of different networking interfaces present on today's smartphones depends on multiple factors, such as distance, interference, signal strength or device model. Thus, the dependency between the size of the transferred data and the energy consumed by the used networking interface is commonly obtained by averaging series of measurements at different locations, at different times of the day and by different devices [8], [9], [10], [11]. In Table I we summarize the results presented in [8] and [12].

|           | Transfer (J/MB) | Idle (W) | Scan (W) |
|-----------|-----------------|----------|----------|
| 3G        | 100             | 0        | 0        |
| Wi-Fi     | 5               | 0.77     | 1.29     |
| Bluetooth | 0.1             | 0.01     | 0.12     |

Table I  
CONSUMPTION OF SMARTPHONE NETWORKING INTERFACES.

We see that, observed purely from the aspect of energy required for data transfers (and ignoring the range), Wi-Fi is much more efficient than 3G. However, any solution that requires smartphones to keep their Wi-Fi interfaces switched on, constantly scanning for transfer opportunities, would actually consume more, and not less energy than 3G transfers. Let's see it on the example of an iPhone 4 and its  $5.25Wh$  battery. When switched on, an iPhone's Wi-Fi interface interchangeably scans for  $1s$  and then spends  $8s$  in idle state. Given the values in Table I, simple calculus gives us that in this regime the daily consumption of iPhone's Wi-Fi interface would be  $19.87Wh$ . This means that the battery of an iPhone that performs continuous Wi-Fi scanning empties in less than  $6.5h$ .

So, in the ideal case, 3G transfers would be replaced with energy more efficient Wi-Fi transfers whenever possible, but Wi-Fi interfaces would be switched off whenever transfer opportunities are not present. In other words, Wi-Fi craves for alternative solutions for the detection of transfer opportunities. In order to solve this problem, we use prediction provided by the operator.

### III. OUR OFFLOADING SOLUTIONS

#### A. HotZones Algorithm

A HotZone is a cell, partly covered by the operator owned Wi-Fi access points. We do not expect this coverage to be perfect. So, when in a HotZone, a user can expect to receive a requested content through one of these access points with probability  $p$ . We assume that an operator deploys the Wi-Fi access points in

---

#### Procedure 1 Serving user's requests in a network with HotZones.

---

```

if ( $S_r^u(t) \neq \emptyset$ ) then
  if ( $c \in H$ ) then
    Turn on WiFi interface;
    Try to serve all  $r \in S_r^u(t)$  via WiFi;
    //a success with probability  $p$ 
  else
    Get  $\tau_H^u =$  time before  $u$  enters a cell  $\in H$ ;
    for all  $r \in S_r^u(t)$  do
      if ( $\tau_r$  expires in  $\leq \tau_H^u$ ) then
        Serve  $r$  via 3G;
      else
        Do nothing;
    end if
  end for
end if
end if

```

---

addition to the existing 3G infrastructure, with the goal of offloading a part of the traffic from the 3G network.

In the process of HotZones selection, an operator first extracts typical mobility profiles of its subscribers. We refer to these profiles as User Mobility Profiles (UMPs). The process of their extraction is described in Section III-D. With the UMPs created, an operator ranks cells based on the average number of daily visits. Then, a set of HotZones  $H$  is chosen in a greedy way, so that a cell with the highest number of daily visits is added first to the set, the second most visited cell is added second, etc. The cardinality of the set  $H$  is a tradeoff between the cost of the Wi-Fi deployment and targeted benefits. As we show in Section V, this number strongly affects the observed performance measures.

The rationale behind the greedy selection of HotZones is that a user's request does not have to be served in a cell where it is created. As we target delay-tolerant offloading (keeping in mind that Wi-Fi access points are affordable, but not free) it makes sense to concentrate on cells with a high number of daily visits.

Once the set of HotZones  $H$  is created, an operator sends it to each user, along with her UMP. The operator can also send occasional updates if needed (for example if a new cell is covered by Wi-Fi). As any mobile application can obtain the real-time information about the current cell, it can use the set of HotZones and the UMP for the prediction of Wi-Fi availability. A whole class of mobile applications, where delay-tolerant content is requested can benefit from such prediction.

One such application, which we use in our evaluation is described in Section IV-B. Let us denote by  $S_r^u(t)$  the collection of pending requests of a user  $u$  (i.e. the

user  $u$ 's requests that are still not served at time  $t$ ). Let us denote by  $c$  the current cell of the user  $u$ . Finally, let us denote by  $D$  the maximum delay users permit. Each time a request  $r$  is created, a timer  $\tau_r$  with timeout equal to  $D$  is set by the application. If the request is not served before the expiry of the timeout, it is served via 3G. Using these parameters, the application on user  $u$ 's smartphone performs *Procedure 1* every  $T_P$  minutes.

We see that the application relies on the user's UMP for the prediction of possible Wi-Fi transfer opportunities within the allowed delay  $D$  (enforced with timers  $\tau_r$ ). If such an opportunity is not likely to emerge, the pending requests in the set  $S_r^u(t)$  are served immediately through 3G in order to minimize delivery delays.

### B. MixZones Algorithm

A MixZone is a  $(c, t)$  pair (where  $c$  denotes a cell and  $t$  denotes an hour of the day). The set of MixZones  $M$  is selected by an operator using the following probabilistic geometric model. Let us denote by  $A_c^e$  the effective area of a cell  $c$ . Let us denote by  $R$  the Wi-Fi radio range (90m) and let us denote by  $N_c(t)$  the number of users in cell  $c$  during hour  $t$ . A pair  $(c, t)$  is added to the set  $M$  if, on average, the following condition is satisfied for hour  $t$ :

$$p_c(t) = 1 - \left(1 - \frac{R^2\pi}{A_c^e}\right)^{N_c(t)} \geq P_{thresh}$$

Probability  $p_c(t)$  is an estimate of the probability that a user in a cell  $c$  enters the range of another user during hour  $t$ . We assume that the spatial distribution of users in cell  $c$  is uniform.  $P_{thresh}$  is the value of the probability  $p_c(t)$ , which needs to be exceeded at hour  $t$  in order for the pair  $(c, t)$  to be added to the set  $M$ . The effective area of the cell  $A_c^e$  is introduced to compensate for the assumption of uniformity, as there are regions in each cell that are less likely to be visited by users. Thus,  $A_c^e$  represents 90% of the cell area in the case of small cells ( $A < 4km^2$ ), 75% in the case of medium cells and 60% in the case of large cells ( $A > 25km^2$ ).

The HotZones algorithm has only the spatial dimension. With the MixZones algorithm, we also have the temporal dimension. A cell that is a MixZone at time  $t_1$  is not necessarily a MixZone at time  $t_2, t_2 \neq t_1$ . This is because the MixZones algorithm is based on opportunistic transfers, which means that users that want to exchange content have to be in radio range, with their Wi-Fi interfaces switched on *during the same period of time*. As we want to avoid the Wi-Fi scanning, it is the operator who decides when and where the Wi-Fi interfaces on a group of users' devices are switched on.

In the case of the MixZones algorithm the quasi-static user mobility profiles (UMPs) are not sent to users. Instead, an operator uses UMPs, along with the set  $M$ ,

---

### Procedure 2 Serving user's requests in a network with MixZones.

---

```

if ( $c \in M$ ) then
  Turn on WiFi interfaces in set  $U_c(t)$ ;
  Opportunistic WiFi transfers among users;
else
  for all users  $u$  in cell  $c$  do
    Get  $\tau_M^u =$  time before  $u$  enters a cell  $\in M$ ;
    for all  $r \in S_r^u(t)$  do
      if ( $\tau_r$  expires in  $\leq \tau_M^u$ ) then
        Serve  $r$  via 3G;
      else
        Do nothing;
      end if
    end for
  end for
end if

```

---

to concurrently signal to a group of users' smartphones if their Wi-Fi interfaces need to be switched on. As UMPs and set  $M$  are stored only on the operator side, they can be refreshed more often, using the information coming from calls and data sessions.

As the MixZones algorithm is based on opportunistic exchanges, it is assumed that every user has a cache, where she stores content that can be sent to other users. Additionally, it is assumed that an operator has the real-time insight in the content requested by users and content available in users' caches. Whenever a user creates an item request or receives an item, she notifies the operator's cloud, by sending the ID of the item.

Similarly as in the HotZones algorithm, let us denote by  $S_r^u(t)$  user  $u$ 's collection of pending requests. Let us denote by  $D$  the maximum permitted delay. Each time a request  $r$  is created, a timer  $\tau_r$  with timeout  $D$  is set by the application. If the request is not served before the expiry of the timeout, it is served via 3G. Finally, given the knowledge of items requested by users and items available in their caches, at any time  $t$  and in any cell  $c$ , an operator can select a set of users  $U_c(t)$ , such that each selected user: (i) either has items requested by some other users in  $c$  or (ii) requests items available in caches of some other users in  $c$ . Using these parameters, a server in the operator's cloud performs *Procedure 2* every  $T_P$  minutes, for every cell  $c$  in the network.

The idea behind the creation of the set  $U_c$  is to switch on only the users that can contribute to data transfers. The problem is similar to the NP hard *set cover problem*, where a set of items is to be covered with a number of subsets. It differs in that in our case each requested item should be covered by preferably more than one copy, in order to increase the delivery probability.

### C. Implementation Aspects

From the implementation aspect, HotZones algorithm is less complex to deploy. It requires an operator to create UMPs and set  $H$  and to deliver them to users. Apart from this initial support from the operator (and possible occasional updates), the HotZones algorithm is completely distributed. All decisions with regard to the use of networking interfaces are made locally by the smartphone application. The APIs of today's smartphone operating systems (such as iOS) enable applications to switch between 3G and Wi-Fi. An interworking WLAN client application on the handset offers the ability for two functions. The switchover is seamless and presents a transparent view to the user.

In the case of MixZones algorithm, support for ad hoc exchanges between users' smartphones is needed. Such support exists in the case of iPhone and it is additionally improved with the release of the iOS 4.3.

Regarding the operator's assistance, MixZones algorithm is more demanding. First, an operator is required to maintain a fine-grained knowledge of users' requests and caches, in order to avoid switching on Wi-Fi interfaces on devices that can not contribute to data exchanges. This task can be performed by a server in the operator's cloud. The server can receive small incremental updates, sent by users, following the changes in their caches or requests. The updates, containing only item IDs, can be uploaded via 3G. Due to their small size, they would consume few resources.

Second, MixZones algorithm requires an operator to switch on Wi-Fi interfaces on users' smartphones remotely, so that a group of users in a cell have their Wi-Fi interfaces turned on during a same time period. There are multiple possible solutions to this problem. One of them is the use of control channels. In order to quickly locate called users, base stations maintain communication with subscribers, even when they are inactive. Cell phones send location updates to base stations through the access channel and base stations occasionally page users using the paging channel. Control channels are also used for sending text messages and similarly, an operator can use them to signal to a smartphone if a networking interface needs to be switched on.

### D. Inferring Users' Mobility

The most commonly stored users' activity (and mobility) records are Call Detail Records (CDRs). A CDR contains calling and called users' numbers (blank in case of a data session), date and time, session duration, caller's cell ID, cell coordinates, etc. As explained in Section IV-A, these are precisely the records we have at our disposal.

As both proposed algorithms rely on users' mobility, we use CDRs to obtain it. The approaches to describe

users' mobility can be classified as: (i) quasi-static, where a rather permanent list of pre-computed locations describes the mobility of a user and (ii) dynamic, where a list of cells is dynamically adjusted (with expires of cells).

Using only one month of the data set, we extract what we refer to as quasi-static, user mobility profiles (UMPs). A UMP is an array of 24 elements, which contains the most visited cell by a user for each hour of the day. For each of the half of a million users that we observe, we extract two such profiles, one for the weekdays and one for the days of weekend. We use the remaining two months of data to test how accurately UMPs predict users' mobility. With only one month of data used for the creation of profiles, we obtain a 69% match with the remaining two months. This relatively high prediction accuracy, based on a few weeks of data, is the result of a high correlation between daily mobility patterns of individual users, especially for the weekdays. Users tend to visit the same cells at the same hours.

Once an operator has the UMPs extracted, these can be sent to users (i.e. every user is provided with her own mobility profile). Although we find that UMPs show little change over time, it is possible for an operator to occasionally recalculate UMPs. This way, the quasi-static mobility profiles can be made more dynamic and adjustable to possible changes in mobility, which can occur over time (the change of workplace, address, etc.).

Our algorithms use UMPs for predictions of upcoming areas suitable for Wi-Fi transfers to switch between networking interfaces. More generally, the mobility profiles can be used by a wider range of smartphone applications (for example, any application that sends push notifications to users based on expected mobility).

Finally, from the perspective of HotZones algorithm, it is interesting to check if most users generate their requests from a small subset of frequently visited cells. Unfortunately, our data shows that this is not the case. By observing only data session CDRs, over the period of three months, we can see that users tend to download content from a wide range of locations. Similarly, by focusing on MMS records we can see that uploads<sup>1</sup> are made from a variety of locations. Although users request content from a variety of locations, we notice that a relatively small subset of cells reoccurs in the majority of UMPs. These are precisely the highly frequented cells that are top candidates for HotZones.

CDRs are not the most detailed location logs an operator can store. Every cellular operator has access to more detailed location records. They contain information exchanged via the paging and access channels. Log files containing this additional information would allow

<sup>1</sup>Uploads are less important for the application studied in this paper.

us to recreate UMPs with more accuracy. However, from the aspect of our goal, CDRs seem to naturally fit the purpose. They permit us to observe mobility through activity and as the goal is an activity driven offload, what is needed are the areas with high user activity.

#### IV. EVALUATION SETUP

##### A. About the Data Set Used in the Study

The data set we use is obtained from a major mobile operator and it consists of CDRs from 1 million users for a period of three months (October-December 2009). The data covers an area of a Western European capital and its major commuting suburbs. We focus our analysis on 533,189 users, which had more than 50 records (calls/data sessions) per month.

##### B. Social Music Sharing Application

In order to estimate the proposed algorithms' potential for offloading of socially recommended contents, such as music or video, we consider an application that allows users to request media items based on social recommendation. All items are part of a catalogue of size  $I$ , characterized by Zipf(1) distribution. It was shown that Zipf distribution describes content popularity in many social and content sharing networks; a recent study of del.icio.us [15] found Zipf distribution in tags associated with the URLs flickr.com (photos), del.icio.us (social bookmarking), pandora.com (music) and youtube.com (video). The same distribution was found to describe the popularity of YouTube videos in [3].

We assume that each user has a cache (a library) with  $b$  items. The caches are refreshed following one of the three popular caching strategies: FIFO, LRU (Least Recently Used) and LFU (Least Frequently Used). The LRU and LFU algorithms are completely distributed; they are based only on user's local observations of the requests for each of the items in her cache.

The total of  $N$  users request items following two request dynamics: (i) every time a user  $A$  calls a user  $B$ , she requests an item from  $B$ 's cache, with the constraint that the item is not already requested by user  $A$  or that it is not in her cache; (ii) every time a user initiates a data session she requests an item from the catalogue, following the Zipf distribution of items' popularities.

Given these request dynamics, at each moment in time  $t$ , the state of a user  $u$  is described by: (i) the current cell  $c$ , (ii) the collection of pending requests  $S_r^u(t)$  and (iii) the collection of available items  $S_a^u(t)$  (i.e. the  $b$  items in the user's cache).

The described application has certain similarities with two recent Apple projects. In May 2010 Apple filed a patent application that describes a system for targeted ads based on the contents of friends' media libraries. In September 2010, Apple added a music social network

to iTunes, called Ping, that enables users to share music preferences with friends [16].

##### C. Trace Driven Simulation

We design a Java simulation framework that enables us to perform discrete event simulations, exploiting the real user mobility and activity, extracted from the data set described in Section IV-A. The framework permits, at any moment in time, to keep track of users present in different cells, the contents of their caches ( $S_a^u(t)$ ) and the lists of their requests ( $S_r^u(t)$ ); they come as a result of real calls and data sessions initiated by users. It also allows us to simulate different caching strategies and different cache sizes.

We simulate the proposed algorithms with  $N = 533189$  users, who move between  $C = 1141$  cells, following their real mobility recorded in the CDRs. The simulation lasts for 30 days, which are different from the month used to extract UMPs.

With both algorithms, the turning on of a Wi-Fi interface incurs the energy cost of two  $1s$ -scanning intervals and  $8s$  of idle state, even if no data transfers occur. The energy consumption is calculated using the values in Table I. The parameter  $T_p$  is set to  $10min$ .

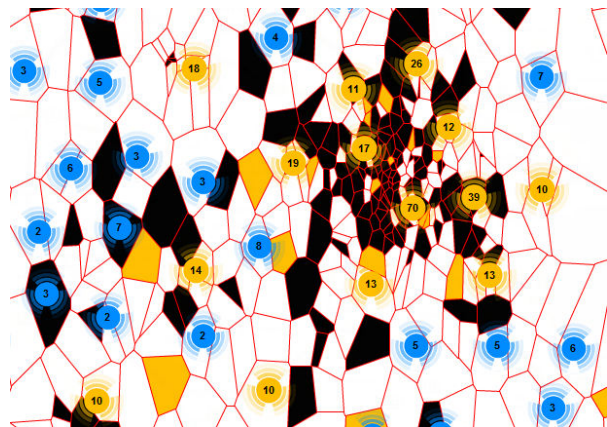


Figure 2. HotZones (light cells) and the cells that form the MixZone pairs (dark cells) in the city center. The circle-shaped markers contain numbers of cell towers in different regions.

There are  $I = 100K$  items in the considered media catalogue. The popularity of items follows Zipf(1) distribution. Users' caches are initially filled with items following the same distribution (i.e. the probability that an item is found in a user's cache depends on its popularity obtained from Zipf(1) distribution). The caches remain full throughout the simulation. The items in them are replaced following one of the caching strategies. The item size is uniformly distributed between 5 and 10MB, which is comparable to the size of a large music file or an average YouTube video. We run simulations with cache sizes of  $b = 100$  and  $b = 1000$  items, which

corresponds to  $0.75 - 7.5GB$  of storage space. The media catalogue size can be compared with sizes of large music catalogues (such as iTunes), whereas the simulated cache sizes are a reasonable estimate of the sizes of personal smartphone media libraries.

We first infer the set of MixZones by setting the parameter  $P_{thresh}$ . The choice of this parameter is conditioned by the energy efficiency requirement. As shown in Figure 9, the value  $P_{thresh} = 0.8$  saves most energy and allows to 225 cells to form 2612 MixZone  $(c, t)$  pairs. Next, we look for the set of HotZones that provides comparable performance to the HotZones algorithm. We find that the top 30 cells, selected following the procedure described in Section III-B meet this goal. The HotZones specific parameter  $p$  (which denotes the probability that a request is served via a Wi-Fi access point in a HotZone) is set to 0.9.

The HotZones in the city center, and the cells that participate in the MixZone pairs, are shown in Figure 2.

We simulate both algorithms with the value of parameter  $D$  (maximum permitted delay) equal to  $1h$ ,  $6h$  and  $24h$ . In order to evaluate the impact of prediction and delay tolerance, we also simulate the special case of the HotZones algorithm, with  $D = 0$ , which we dub the Real-Time Wi-Fi Zones. This solution is currently considered (or deployed) by a number of operators. Smartphones (such as iPhone) support it with seamless switchover between 3G and Wi-Fi.

## V. PERFORMANCE EVALUATION RESULTS

**Both algorithms achieve significant energy saving. Up to 75% of traffic offloaded by only 30 HotZones.** For the selected sets of HotZones and MixZone pairs, we plot the traffic offloaded from the 3G network and the amount of energy saved, as a function of the maximum permitted delay  $D$  (Figure 3). We see that for  $D = 1h$ , roughly 20% to 40% gets offloaded to Wi-Fi and 20% to 35% less energy is consumed by the application. For  $D = 6h$ , this fraction goes up to 50%. In the case of  $D = 24h$ , the impressive 60-75% are offloaded with as few as 30 HotZones.

We can also see that the HotZones algorithm is less efficient than the MixZones algorithm in case of lower permitted delays ( $D = 1h$ ). This is because one can not expect users to enter one of the very few HotZones every 60 minutes. However, as the permitted delay increases, users become more likely to enter the HotZones and the performance of the algorithm improves.

**Caching strategy has little effect on performance. Cache size is crucial for the MixZones algorithm.** One of the first things we observe is that for the mobility and request dynamics obtained from our data set, caching strategies have a very limited effect. With an average of 80 requests per month, and the user cache

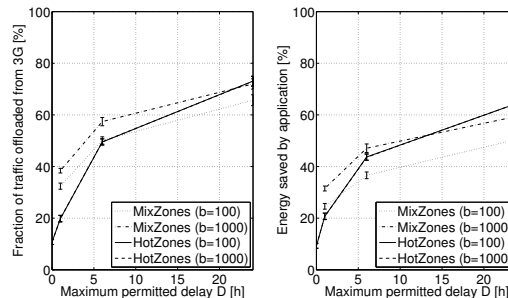


Figure 3. Offloaded traffic and saved energy as a function of permitted delay  $D$ . The curves are obtained for the MixZones (225 cells) and HotZones (30 cells) algorithms, with LRU caching strategy. The curves for the HotZones algorithm almost overlap.

sizes of  $b = 100$  and  $b = 1000$ , the initial Zipf(1) distribution of items in users' caches is well maintained after 30 days, for all three simulated caching strategies. The Complementary Cumulative Distribution Functions (CCDFs) in Figure 4 show the initial distribution of items in users' caches and the distributions after 30-day simulations, with LRU, LFU and FIFO caching strategies. We see that even with the caches of  $b = 100$  items, the system stays stable. Consequently, the values of the performance metrics that we obtain for these caching strategies are very similar. In order to avoid the unnecessary repetition, in the rest of this section we show the results for the LRU caching strategy only. The other figures are provided in the technical report.

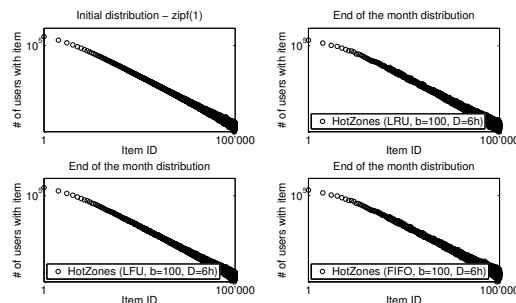


Figure 4. Evolution of item popularity distribution: The four curves are plotted on the log-log scale and they show the initial distribution of items in users' caches and the distributions after 30-day simulations, with LRU, LFU and FIFO caching strategies.

Unlike caching strategy, the cache size plays a major role in the case of the MixZones (Figure 3). Larger cache sizes increase the probability that an encountered user can serve a request. Hence, the improvement brought by the cache size, is significant. On the contrary, as expected, the cache size does not affect the HotZones algorithm, where serving requests depends only on users' mobility and the selected set of HotZones. Thus, the curves for  $b = 100$  and  $b = 1000$  almost overlap.

**Most energy saving comes from prediction and delay tolerance.** The special case of the HotZones algorithm with  $D = 0$  (which we refer to as the RT Wi-Fi Zones), allows us to estimate the offloading and energy saving that do not come from prediction and delay tolerance, but purely from the placement of Wi-Fi access points. As we see in Figure 3, with 30 HotZones and  $D = 0$ , only about 10% of traffic is offloaded and about the same amount of energy is saved. This means that the rest of the improvement observed for higher values of  $D$  comes from prediction and delay tolerance.

The energy improvement brought by prediction can be better observed in Figure 5. The figure contains the Cumulative Distribution Functions (CDFs) for the daily energy consumption of the application, for both evaluated algorithms and cache sizes. The dotted curve in two bottom figures is the CDF for the case of RT Wi-Fi Zones. We see that it almost overlaps with the CDF coming from pure 3G transfers, yielding less than 10% improvement (as shown in Figure 3). Again, the increase in cache size affects only the MixZones algorithm.

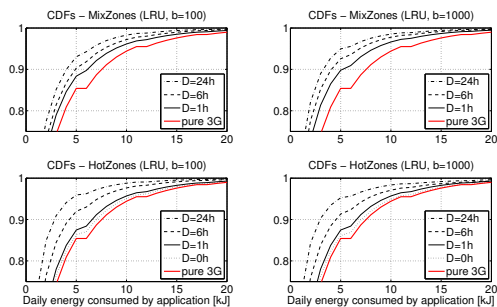


Figure 5. Cumulative Distribution Functions (CDFs) for the daily energy consumption of the application. The subfigures correspond to different combinations of the offloading algorithm and cache size  $b$ .

In order to better understand the origin of the energy savings with MixZones and HotZones, we plot two histograms that show the energy consumed to serve users' requests (Figure 6). We see that in the case of both algorithms a portion of requests is served via 3G. The energy required to serve such a request ranges from 500 to 1000J, depending on the item size (as explained in Section IV-C, item sizes are uniformly distributed between 5 and 10MB). In the case of a pure 3G delivery (without either of the proposed algorithms) only this part of the distribution would exist.

However, with MixZones and HotZones algorithms, we observe a mode on the left, which comes from the requests served via Wi-Fi. In the case of HotZones, the mode is formed around the value that corresponds to the energy needed for a single item download via a Wi-Fi access point, plus the energy needed for switching on a Wi-Fi interface. In the case of the MixZones algorithm,

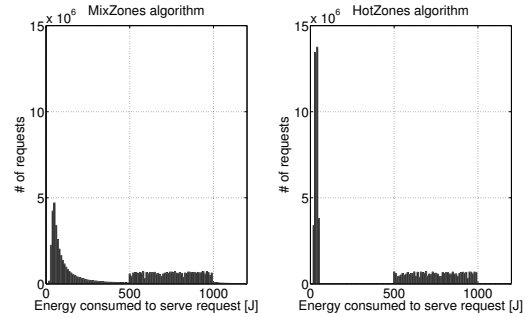


Figure 6. Request energy histograms show energy consumed to serve users' requests. The uniform portion on the right comes mostly from the requests served via 3G. The modes on the left come from the requests served via Wi-Fi.

the mode is moved towards the value corresponding to two Wi-Fi transfers (sending and receiving users), plus the energy cost of turning on of two Wi-Fi interfaces. Additionally, in the case of the MixZones algorithm, this part of the distribution is more skewed, as it is more likely that a user, with her Wi-Fi interface turned on, would miss a transfer in a MixZone, than in a HotZone. This comes from the fact that a user in a HotZone finds an access point (with access to all items) with probability  $p = 0.9$  and a user in a MixZone meets another user (with only  $b$  items) with probability  $P_{thresh} = 0.8$ . Thus, the MixZones algorithm sometimes requires users to have their Wi-Fi interfaces switched on several times before a request is served.

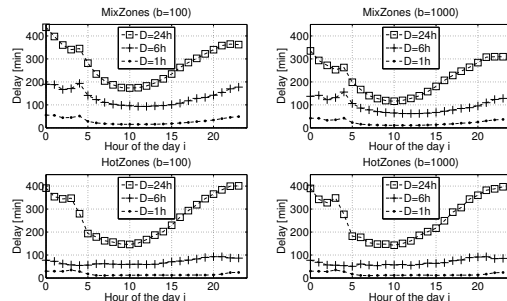


Figure 7. The average delay with which requests are served as a function of the time of the day and the maximum permitted delay  $D$ .

**Effective delay in the system is much lower than the maximum permitted delay  $D$ .** Another important performance metric is the delay with which users' requests are served. The maximum permitted delay  $D$  sets the upper limit on item delivery time. However, as we can see in Figure 7, the average time with which users' requests are served is often much lower than the value of  $D$ . For  $D = 24h$ , the requests are actually served in less than 7h, and as fast as 2h during some periods of the day (depending also on the algorithm



used). In case of  $D = 6h$ , the actual delay is between  $1.5h$  and  $3h$ , while for  $D = 1h$ , the requests are served in  $15 - 50$  minutes. In Figure 7 we also observe the time of the day dependency, with lowest delays during the peak activity hours. This means that the proposed algorithms offer best offloading performance during the hours when a 3G network is most heavily loaded.

**Real-time offloading requires 3-4 times more Wi-Fi cells than the delay-tolerant HotZones algorithm.**

It is interesting to compare the offloading potential of the RT Wi-Fi Zones with our delay-tolerant HotZones algorithm. In order to perform this comparison (in addition to the analyzed setup with 30 HotZones), we run the HotZones algorithm with 60, 120, 240, 480 and 960 Wi-Fi covered cells. We do it for the values of permitted delay  $D = 0h, 1h, 6h$  and  $24h$ .

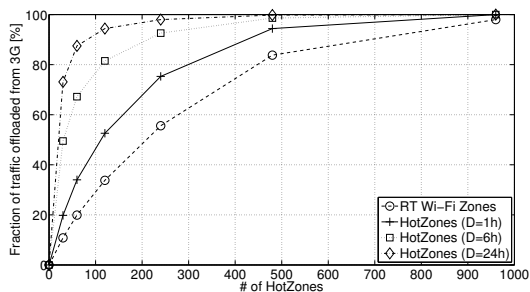


Figure 8. Offloaded traffic as a function of the number of HotZones.

On Figure 8, we can see that for  $D = 6h$ , covering only 10% of cells with Wi-Fi, results in offloading of 80% of traffic. In order to offload the same amount of traffic with  $D = 0h$ , an operator has to cover four times more cells with Wi-Fi. Similarly, the HotZones algorithm permits offloading of more than 90% of traffic with only 20% of Wi-Fi covered cells, while the RT Wi-Fi Zones require coverage of more than 70% of cells for a similar effect. This significant quantitative improvement, brought by prediction and delay tolerance in the HotZones algorithm, is even more valuable knowing that on average the delays are much lower than  $D$ .

**MixZones selection is a compromise: Impossible to maximize both offloading and energy efficiency.** When selecting the number of MixZones (i.e. the algorithm parameter  $P_{thres}$ ), we are guided by energy efficiency. The value  $P_{thres} = 0.8$  is most energy saving and it offers a solid offloading performance. Nevertheless, one can opt for another criterion when choosing the value of  $P_{thres}$ . On Figure 9, we plot the amounts of offloaded traffic and energy saved for the values  $P_{thres} = 0.2, 0.5, 0.8$  and  $0.9$ . For these values we get 732, 590, 225 and 131 MixZones respectively.

We see that although the value  $P_{thres} = 0.8$  guarantees most energy saving, more traffic gets of-

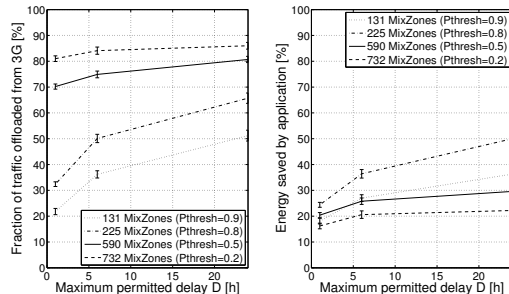


Figure 9. The offloaded traffic and saved energy as a function of the number of  $P_{thres}$  (i.e. the number of MixZones).

floaded for  $P_{thres} = 0.2$  and  $0.5$ . On the other hand for  $P_{thres} = 0.9$  both offloading and energy saving deteriorate. This can be interpreted in the following way. With the decrease of  $P_{thres}$  (increase in the number of MixZones), the number of Wi-Fi transfers increases. However, these new MixZones have lower probability of meeting between users, which results in the increased number of Wi-Fi scanning events without data transfers. This decreases the energy efficiency. On the other hand, the increase of  $P_{thres}$  beyond the value of  $0.8$ , reduces both, the amount of offloaded traffic and the energy saving, due to too few cells that satisfy this condition.

VI. RELATED WORK

A body of work proposes exploiting different smartphone interfaces and mobility for improving download and energy efficiency. In [17], the authors propose collaborative downloading as means of increasing download speeds and battery life. In [11] policies for switching between multiple interfaces are proposed, with the goal to increase battery lifetime. Modeling and comprehensive measurement studies of energy consumption by smartphone interfaces were performed in [8], [9], [10]. In [18], the authors investigate the correlation between locations and types of users' activities. Ananthanarayanan et al. [12] try to improve the energy efficiency of Wi-Fi by replacing Wi-Fi scanning with Bluetooth scanning. However, we are the first to estimate the energy saving in a cellular network coming from the use of opportunistic bandwidth, while accounting for the real costs of data transfers and Wi-Fi scanning.

Another related body of work concerns studies of human mobility [19]. In [14] and [13], the authors use operator provided data to show that contrary to common beliefs, humans follow repetitive and reproducible patterns. We show how this predictability is a key to solve the issue of energy efficient 3G data offloading.

Finally, closely related to the application analyzed in this paper are the applications that leverage the cloud to surpass the limitations of mobile environment [20].

## VII. CONCLUSIONS

In this paper we explore the use of prediction and delay-tolerance for offloading of large, socially recommended contents from 3G networks. We show that the two algorithms we design enable users to trade delay for energy and easily reduce battery consumption coming from 3G transfers of delay-tolerant content for 50%. We show that the real-time offloading requires Wi-Fi coverage of 3 to 4 times more cells, than our delay-tolerant algorithm. We find that both algorithms have lowest delay during the peak hours, when offloading is most needed. We also demonstrate how operators can benefit the collected data to offer cloud solutions, appealing to users (extending battery lifetime) and to the operators (load balancing between orthogonal technologies).

Finally, we believe that performance evaluation of the algorithms that uses a realistic application and a large data set is a contribution on its own. It helps community get better idea of the performance of a large scale delay-tolerant application in a mobile network. It also allows us to gain insight into the possibilities of orthogonal 3G offloading, which is a topic that is likely to become increasingly important in the days to come.

## VIII. DISCUSSION AND FUTURE WORK

In this section we discuss some possible extensions of this study and a few viable directions for future work.

First, a complementary study of an application that relies on uploads could take advantage of the MMS entries available in the CDR logs. Second, adding the opportunistic component to the HotZones algorithm could additionally improve performance. The solid performance of the HotZones algorithm with very few Wi-Fi covered cells (30) shows that little infrastructure can match the performance of a comprehensive opportunistic solution. Although many factors (primarily increased storage) give greater capabilities to opportunistic nodes, the growing popularity of social networks and the increase in user generated content makes caching of relevant content difficult and complicates the design of pure opportunistic solutions.

It is also worth mentioning that although covering a cell with access points incurs certain costs, such a deployment could be facilitated by the use of home wireless routers that provide the Internet access to millions of operator's customers. The fairness scheme proposed in [21] could be extended to mobile subscribers in order to reduce cost and provide better coverage.

## REFERENCES

- [1] "New York Times. iPhones Overload AT&T Network, Angering Customers, September 2009."
- [2] "AT&T to Pay \$1.93 Billion for Qualcomm Mobile Spectrum," <http://www.bloomberg.com/news/2010-12-20/at-t-agrees-to-acquire-wireless-licenses-from-qualcomm-for-1-93-billion.html>.
- [3] M. Cha, H. Kwak, P. Rodriguez, Y.-Y. Ahn, and S. Moon, "I tube, you tube, everybody tubes: analyzing the world's largest user generated content video system," In *IMC '07*.
- [4] T. Hammond, T. Hannay, B. Lund, and J. Scott, "Social bookmarking tools (i): A general review," *D-Lib Magazine*, vol. 11, no. 4, 2005.
- [5] B. Lund, T. Hammond, M. Flack, and T. Hannay, "Social bookmarking tools (ii): A case study-connotea," *D-Lib Magazine*.
- [6] "CNNMoney.com: Billions For Wireless Networks, Dec 2010," [http://money.cnn.com/2010/12/28/technology/billions\\_for\\_wireless\\_networks/index.htm](http://money.cnn.com/2010/12/28/technology/billions_for_wireless_networks/index.htm).
- [7] "AT&T Expands Wi-Fi Pilot Project," <http://www.att.com/gen/press-room?pid=18162&cdvn=news&newsarticleid=30982&mapcode=consumer>.
- [8] A. Rahmati and L. Zhong, "Context-for-wireless: context-sensitive energy-efficient wireless data transfer," in *MobiSys '07*.
- [9] P. Mohan, V. N. Padmanabhan, and R. Ramjee, "TrafficSense: Rich monitoring of road and traffic conditions using mobile smartphones," in *In Microsoft Technical Report*, 2008.
- [10] N. Balasubramanian, A. Balasubramanian, and A. Venkataramani, "Energy consumption in mobile phones: a measurement study and implications for network applications," in *IMC '09*.
- [11] T. Pering, Y. Agarwal, R. Gupta, and R. Want, "Coolspots: reducing the power consumption of wireless mobile devices with multiple radio interfaces," in *Proceedings of MobiSys '06*.
- [12] G. Ananthanarayanan and I. Stoica, "Blue-fi: enhancing wi-fi performance using bluetooth signals," in *MobiSys '09*.
- [13] H. Zang and J. C. Bolot, "Mining call and mobility data to improve paging efficiency in cellular networks," in *MobiCom '07*.
- [14] M. Gonzalez, C. Hidalgo, and A.-L. Barabasi, "Understanding individual human mobility patterns," *Nature*, 2008.
- [15] R. Angelova, M. Lipczak, E. Milios, and P. Pralat, "Investigating the properties of a social bookmarking and tagging network," *International Journal of Data Warehousing and Mining*, 2010.
- [16] "Belfast Telegraph. Apple's Ping Rival to Facebook and Twitter Launches with 160m Users, September 2010."
- [17] G. Ananthanarayanan, V. N. Padmanabhan, L. Ravindranath, and C. A. Thekkath, "Combine: leveraging the power of wireless peers through collaborative downloading," in *MobiSys '07*.
- [18] I. Trestian, S. Ranjan, A. Kuzmanovic, and A. Nucci, "Measuring serendipity: connecting people, locations and interests in a mobile 3g network," in *IMC '09*.
- [19] E. Halepovic and C. Williamson, "Characterizing and modeling user mobility in a cellular data network," in *PE-WASUN '05*.
- [20] K. Church, J. Neumann, M. Cherubini, and N. Oliver, "Socialsearchbrowser: a novel mobile search and information discovery tool," in *Intelligent User interfaces 2010*.
- [21] D. Giustiniano, E. Goma, A. Lopez Toledo, I. Dangerfield, J. Morillo, and P. Rodriguez, "Fair wlan backhaul aggregation," in *Proceedings of MobiCom '10*.