Taming the Mobile Data Deluge with Drop Zones

Ionut Trestian, *Student Member, IEEE,* Supranamaya Ranjan, Aleksandar Kuzmanovic and Antonio Nucci

***Abstract*—Human communication has changed by the advent of smartphones. Using commonplace mobile device features they started uploading large amounts of content that increases. This increase in demand will overwhelm capacity and limits the providers' ability to provide the quality of service demanded by their users. In the absence of technical solutions, cellular network providers are considering changing billing plans to address this.**

Our contributions are twofold. First, by analyzing user content upload behavior, we find that the user-generated content problem is a *user-behavioral* problem. Particularly, by analyzing user mobility and data logs of 2 million users of one of the largest US cellular providers we find that (i) users upload content from a small number of locations; (ii) because such locations are different for users, we find that the problem appears ubiquitous. However, we find that (iii) there exists a significant lag between content generation and uploading times, and (iv) with respect to users, it is always the same users to delay.

Second, we propose a cellular network architecture. Our approach proposes capacity upgrades at a select number of locations called Drop Zones. Although not particularly popular for uploads originally, Drop Zones seamlessly fall within the natural movement patterns of a large number of users. They are therefore suited for uploading larger quantities of content in a postponed manner. We design infrastructure placement algorithms and demonstrate that by upgrading infrastructure in only 963 base-stations across the entire United States, it is possible to deliver 50% of content via Drop Zones.

I. Introduction

Cellular network providers are faced with an increasing challenge when offering data services over their networks. In the last several years, the production and consumption of digital media over cellular networks has evolved dramatically, and it is continuing to grow at an exponential pace [11]. As an example, it is expected that more than 140 million mobile subscribers worldwide will use social networking applications that enable them to share photos and videos with their friend circle on their phones by 2013 [13].

The problem incurred by the booming activity on mobile devices is that users are no longer only consuming data but have also started *producing* content that grows at an exponential pace. This happened due to high processing power and high capability mobile devices (*e.g.*, enabled with high-resolution cameras) that became available for mass-market prices around the world.

The load induced by the user-generated content creates problems to mobile network providers on a daily basis [5, 9]. AT&T officials warned that the Internet will not be able to cope with the increasing amounts of video and user-generated content being uploaded [2]. For example, users are likely to upload 'heavy' content, *e.g.*, photos and videos, that range from several tens of KBytes up to several MBytes, to popular sites such as Flickr, Facebook, or Youtube, or send directly to their friends. Contrary to 'traditional' content (*e.g.*, the one shared by peer-to-peer applications), user-generated content is unique and often meaningful only to a user and his social circle. Hence, traditional content delivery methods, including caching that would at least reduce the long-haul burden on the provider are incapable of addressing this issue.

In light of the above changes, cellular network providers are rushed to address the problem and keep up with the explosion of content production and consumer interest that drives the traffic increase. In the absence of viable solutions some providers are considering charging special usage fees to heavy data users [3]. AT&T, concerned by the data usage habits of iPhone customers took further steps and changed billing plans [1]. Moreover, the current efforts conducted by the providers are focused on "educating customers about what represents a megabyte of data and improving systems to give them real-time information about their data usage" [3].

Our key contribution lies in demonstrating that a feasible *win-win* solution to this emerging problem *does* exist. In particular, our approach enables users to freely upload their content. At the same time, it helps providers to cope with growing uploading trends. We demonstrate that providers can reach this goal by strategically upgrading small parts of their networks, (that we call Drop Zones) in which users can upload heavy content as they pass by in their daily commute. We base our approach on the following observations.

First, by analyzing mobility and upload properties of nearly 2 million users of a mobile 2.5G, and 3G network, we confirm that users are likely to upload 'heavy' content from most locations, implying that the problem is wide-spread. However, a structural analysis of joint user mobility and uploading properties shows that the user-generated content problem is vastly a *user behavioral problem*. Indeed, we find that an individual user is likely to upload 'heavy' content only from a small subset of locations, typically corresponding to his home, or work or school locations. Still, given that such locations are different for different users, the problem appears ubiquitous since the user-generated content uploads grow exponentially at *most* locations.

Second, we analyze properties of user-generated content: (i) uploaded via mobile devices to popular sites such as Flickr, or (ii) directly sent to friends. We find that large amounts of such content is uploaded in a *postponed* manner, *i.e.*, there exists a time lag ranging from several hours to weeks, from when the content is generated to when it is uploaded. For example, from our trace, we find that 40% of images are sent via mobile devices at intervals longer than 10 hours since such content was generated; likewise, in more than 55% of scenarios, the

I. Trestian and A. Kuzmanovic are with the EECS Department, Northwestern University, Evanston, IL 60208 USA (e-mail: ionut@northwestern.edu; akuzma@northwestern.edu).

S. Ranjan and A. Nucci are with Narus Inc., Mountain View, CA, 94043 USA (e-mail: soups@narus.com; anucci@narus.com).

difference between content generation and uploading events is longer than a day in the Flickr case.

Our Drop Zone approach is based on (i) changing a user's upload patterns (not the user behavior just where the uploads physically take place), and (ii) disproportionally upgrading bandwidth in a small subset of existing networks. In particular, users can tag content for postponed delivery immediately after generating it, and remove the burden of worrying about uploading such content from home, or work, or school locations. At the same time, providers can take advantage of users' daily commute properties to increase bandwidth at a smaller number of locations. We call these locations Drop Zones, and let users opportunistically upload their content while in such zones. The underlying intuition, that we confirm in our analysis, is that most users visit a much smaller number of *common locations* during daily commutes. Thus, by strategically upgrading small portions of their networks, providers can effectively serve growing user-generated content with minimal resources.

The key research questions we explore in this paper is where to place Drop Zones such that they absorb the most content possible? How to design effective algorithms to approximate this placement problem? What is the relationship between postponed content delivery intervals users can tolerate and needed infrastructure? Can we perturb user movement to achieve better performance? What are the advantages of wireless technologies with a higher coverage?

Our analysis shows that by upgrading only 963 base-stations of the current United States nationwide infrastructure and assuming users would postpone content delivery by 3 days, the analyzed provider can become capable of absorbing 50% of user-generated content delivered in a postponed manner as part of the user daily movement routine. Furthermore we show that when considering spatial proximity of users to our Drop Zone infrastructure, 65% of content could be delivered if users would travel 2 kilometers for the same Drop Zone placement or if better radio technology would extend the radius of the cell by the same amount.

The rest of this paper is structured as follows. In Section II we introduce our Drop Zone content upload approach and we give insights into how users currently upload content. In Section III we present our Greedy Drop Zone placement algorithm. In Section IV we thoroughly evaluate the performance of our Drop Zone placement and investigate how Drop Zones will be used. In Section V we demonstrate the feasibility of our approach by describing a possible implementation. We discuss related issues in Section VI. We present related work in Section VII and conclude the paper in Section VIII.

II. The Case for Drop Zones

Here, we briefly introduce the Drop Zone architecture. Then, we show empirical results that motivate our approach.

A. A Drop Zone Architecture

Figure 1 shows our proposed Drop Zone architecture. The network is fragmented into normal connectivity zones. These correspond to base-stations using the technology that is common place in the provider network, *e.g.*, 3G, or 2.5G. On the other hand, there exist better connectivity zones, that we call

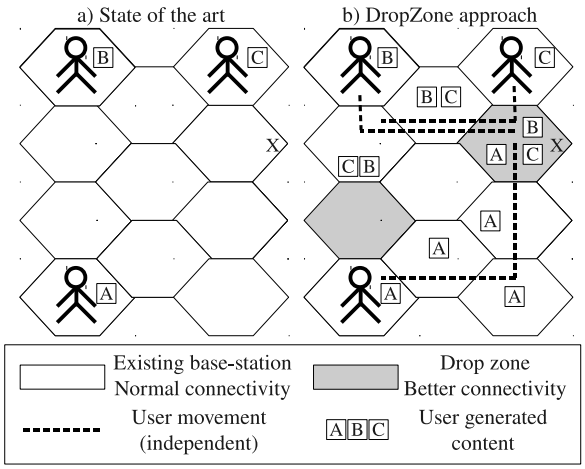


Fig. 1. Postponed delivery example

Drop Zones, shown with a darker color in Figure 1(b). We intentionally do not tie our approach to a particular technology that can be used in Drop Zones for two reasons. First, because it can come in different forms. For example, this could be WiMAX [14] or LTE [10], for which base-station ranges can be roughly matched among 3G, 2.5G and WiMAX and LTE. WiFi also can be one such technology (*e.g.*, [16, 32, 35]). Second, our goal is to understand system performance in limiting scenarios. In particular, if the user-generated content will keep growing at an exponential pace, we want to explore where should the Drop Zones be placed and how should their capacity scale.

Figure 1 illustrates the difference between the state-of-the-art user uploading and our proposed Drop Zone approach. Consider three users, who generate three independent pieces of content, marked by A, B, and C. Figure 1(a) shows how the content is currently uploaded. *Independently* from where a user may generate the content, we find that with a high probability, the user uploads the content from a certain set of locations. We call such user- centric locations (we call them user- centric since users are seen spending a majority of their time here) as the user's 'comfort zones', that most of the time correspond to the user's home, or work, or school locations. We validate this phenomenon in Section II-B1 below. Because such locations are different for different users, the user-generated load grows nearly uniformly at *most* locations.

Note that businesses and sometimes even users decide to take matters into their own hands and employ femtocells (small cellular base stations) in order to get bigger capacity and increased reliability for themselves. They are however complementary to our Drop Zone upgrades as they are targeted at covering a small number of users; unlike the Drop Zones that we envision that will cover significantly larger areas and numbers of users. We also do not have femtocell statistics for the provider that we have analyzed and the results we provide should be considered with this in mind.

Figure 1(b) shows the Drop Zone uploading scenario. Users do not upload content from comfort zones, but rather upload it in a postponed manner from Drop Zones. In particular, all pieces of content, A, B, and C, are uploaded from the same Drop Zone marked by X in the figure. As shown in Section II-B3, users upload content in a postponed manner; we show

TABLE I
SENDING STATISTICS

	Total [MB]	Nr. messages	Avg. size [bytes]	Max. size [MB]
Text	73	1,231,411	58	0.42
Appl.	826	2,193,443	376	3.5
Image	77,495	2,022,361	38,318	3.1
Audio	34,831	531,133	65,577	3.2
Video	5,998	31,345	191,339	3.5

it by considering the difference in content creation and upload times. In this paper we aim to quantify benefits and trade offs involved in using the architecture shown in Figure 1.

B. Analyzing User Behavior

Here, we provide details about the dataset we use for this study. We use an anonymized trace collected from the content billing system for the data network of a large 3G, and 2.5G mobile service provider. The trace contains information about 1,959,037 clients during a seven day period. It preserves user privacy as all identifiers such as users' phone numbers, email addresses and ip-addresses were anonymized.

The trace provides details of user sessions defined as beginning from the time the user is authenticated by the Remote Authentication Dial in User Service (RADIUS) server to the time the user logs off. When logged in and out, the event is stored in our trace. Among the fields we store, we count the anonymized user identifier, the local timestamp and the base-station that serves the user. Further changes in location are reported to the server.

With regards to base-station location, we have the latitude and longitude of the base-stations and since the cell phone only reports the current base-station that it uses, we make the assumption that the current position of the user is given by the position of the base-station. More details about the extraction of the dataset can be found in [36, 37].

The trace contains MMS messages exchanged among users, as well as uploaded to social networking websites such as Facebook, Myspace, Flickr. For messages we have logged the content filename, the size, if it was uploaded or downloaded, the base-station that was used, and the anonymized identifiers for the sender and receiver. In order to upload pictures, Facebook Mobile users for example, receive from Facebook a unique email address that they can use to send emails or MMS with attached images from their mobile phones. The pictures they upload in such manner are shown on their Facebook profile. Our trace contains such information, yet we cannot identify individual uploads since the corresponding identifiers are anonymized. *etc*. Table I summarizes the uploading statistics. We use various attachment types to categorize given content in one of the five categories: text (plain, xml, bookmark, calendar, *etc*), application (word, excel, powerpoint, pdf, rtf, zip, *etc*), image (gif, bmp, jpg, jpeg, *etc*), audio (mp3, acc, midi, wma, wav, amr, *etc*), and video (3gpp, h264, mp4, *etc*).

Note that in our trace there exist smartphones that also use WiFi and prefer it when such a signal is present. Since our trace was collected from the content billing part of the cellular network, WiFi transfers are not recorded. Even though it would be desirable to analyze WiFi effects (transfers meant for Drop Zones could be scheduled when WiFi is available), WiFi usage does not directly impact our study.

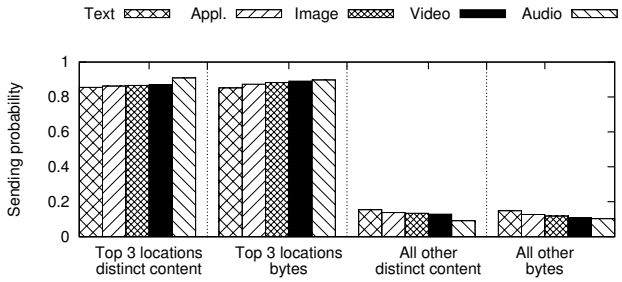


Fig. 2. Sending probability depending on location rank

It is rather obvious that our trace is only a sample of user activity across applications yet the best we could obtain at such a large scale. Indeed, user activity is diverse, users use applications such as Skype, and Facetime to communicate in real time. User-generated-content should not be neglected however. One can note how important such content is by looking at the Facebook usage statistics [6], mobile users being twice as active as desktop users.

1) Users upload content from their top locations: Here, we explore from what locations do users upload their content to the network. To answer this question, we proceed as follows. First, for each *individual* user, we rank the locations he encounters based on the amount of time the user spends in that location. We find that there exists a significant bias in user behavior. In particular, independently from the number of locations that users visit in their daily commute, they tend to upload their content from the top three locations.

Figure 2 shows this effect. In particular, more than 85% of content of all types is uploaded from a user's top three locations. This holds true both for the number of different content pieces uploaded (marked by 'distinct content' in the figure) and the content size (marked by 'bytes' in the figure). Analyzing these results more closely, using straightforward time and space analysis (mainly identifying the locations where a user spends the most time during day hours and night hours as explained in our previous work [37]), we find that in the vast majority of scenarios, two of the three locations can be confidently associated with a user's home and work or school locations. Thus, users prefer to send their content, including the 'heavy' ones that we focus on in this paper, from their top ranked locations.

2) The user-generated content problem is wide-spread: Here, we explore the user uploading behavior from the *network-wide* perspective. Above, we demonstrated that individual users tend to upload content from top locations. However, we show that the problem is the fact that different users have different top locations. Hence, the problem is wide-spread, as we demonstrate below.

Figure 3 shows the amount of uploaded content for each application type as a function of top base-stations in terms of messages sent from that location. We make the following insights. First, in terms of content size, images are dominant, then audio, then video, then applications, then text. Second, the figure shows that while some base-stations are necessarily

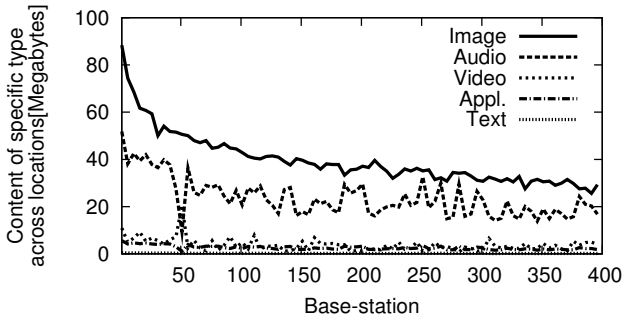


Fig. 3. Sent content type breakdown across base-stations

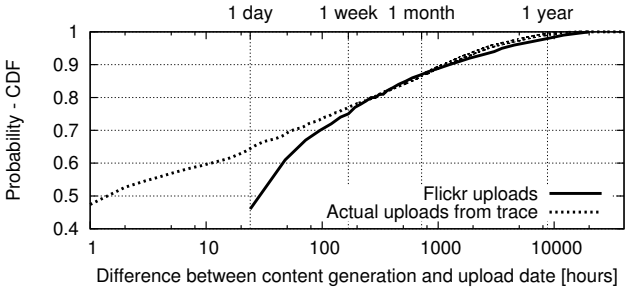


Fig. 4. Delay between shooting and uploading pictures using mobile phones

more popular than others, the popularity difference among base-stations is not dramatic, implying that user-generated content uploads grow nearly uniformly at these locations. Indeed, the peak to mean ratio across base-stations is approximately 2:1 for images and audio, that dominate the trace. Third, the relative ratio among content types stays nearly constant for most base-stations, that implies similar upload trends at most locations.

Summarizing the results from the entire trace, we find that out of all locations that users upload their content from, 80.57% of such locations are top locations for some users. We conclude that the user-generated problem is wide-spread and induced by users' habit to upload such content to the network from top locations.

3) Lag between producing and uploading content: Here, we present evidence suggesting that not all user-generated content is posted or sent immediately after it has been produced. In particular, we have crawled Flickr mobile photography groups where users upload pictures taken via their camera phones [7, 12]. The pictures are also uploaded via the phone. It contains 49,054 pictures uploaded over a period of 3 years. For this part, we were able to extract the time information at the granularity of days. In addition, we have also explored the same issue using our trace. Our mobile trace, different than the above Flickr trace, contains diverse user behavior such as sending photos to friends as well as uploading them to sites such as Facebook, Myspace and Flickr. We obtain the date and time when the content was created by observing that a subset of the image filenames in our trace contain such information (the default setting of the camera is to insert in the filename the date and time of creation). In order to determine if this sample is representative for all the pictures, we compared the distribution of picture sizes for this sample with the overall distribution of picture sizes. The two are indeed similar.

Figure 4 shows the results, implying that users do not necessarily upload their pictures as soon as they shoot them. For example, the Flickr data shows that as much as 55% (100 - 45%) of content is uploaded at a lag longer than one day, while 25% at a lag longer than a week. At the same time, the results extracted from our trace show good match for lags above one week, yet imply shorter lags between picture generation and upload times for less than a week time scales. Still, the results show that 40% of content is uploaded after 10 hours or longer since it has been generated.

The statistics about the lag between content generation and uploading show that users are already willing to tolerate delays. Moreover, while we cannot make strong statements for content that is uploaded soon after generated, we argue that a portion of this content might be possible to deliver in a postponed manner. This is because a subset of users might have a tendency to 'hand over the content immediately', while they might not require it to be uploaded so fast[2]. Nonetheless, the observed postponed content delivery behavior already validates our assumption that the bulk of user-generated content can be uploaded in such a manner. We do not expect all users to postpone content uploads. Indeed, some users have the expectation of having content available immediately after posting it. It is however hard to predict which way user behavior will change as there are factors working both ways (battery life, capped data plans, smaller upload delays, incentives that providers might offer).

Other incentives for users to upload or download content in a postponed manner include: (i) *longer battery life* - it has been shown in [19] that batching transmissions improves battery life by reducing the tail energy incurred in wireless data transmission, (ii) *pricing*, clients can be given discounts for uploading or downloading some content through Drop Zones.

4) User profiling: User profiling can prove useful for a service provider for example for better service targeting. It is not our goal in this paper to profile users of a mobile network. However, one can distinguish from the above results several characteristics on which users could be profiled. Among these characteristics, we can include: (i) how many locations they visit, (ii) how much content they upload, (iii) how long they delay uploading content to the network.

Figure 5 shows the uploading and movement profiles of the users in our trace (all users are captured on the x axis, one point on the x axis being one user). The relative amount of content in bytes uploaded per user (with respect to the y1 axis) shows that a small amount of users out of all the users upload a few orders of magnitude more content than others. In fact 10% of the users upload 54% of the content (not shown). We call these users *heavy uploaders*.

Also shown in the figure is the number of locations a user visits during the trace interval. The same behavior can be observed in this figure. A small amount of users is seen in a large number of locations. In particular 10% of the users are seen in more than 8 locations with some users being seen in more than 100 locations. We call these users *heavy travelers*. Note that in Figure 5, the users on the x axis are not the same for the two curves but are ranked in decreasing order with

[2]Certain phones offer users the option to directly upload a picture after taking it, to sites such as Facebook or to send it to a friend via MMS.

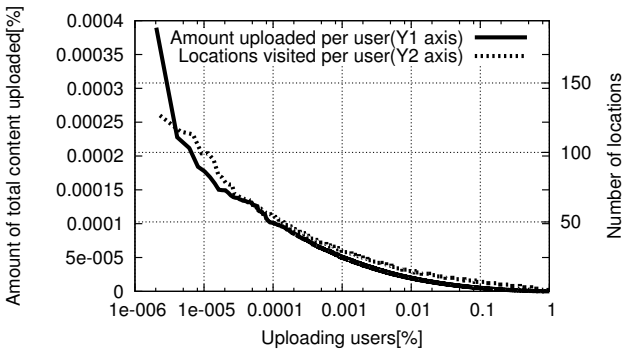


Fig. 5. User movement and uploading profiles. Certain users visit more locations and other users upload more content. Each of the curves corresponds to a different y axis therefore the figure does not show statistical correlation between movement and uploading.

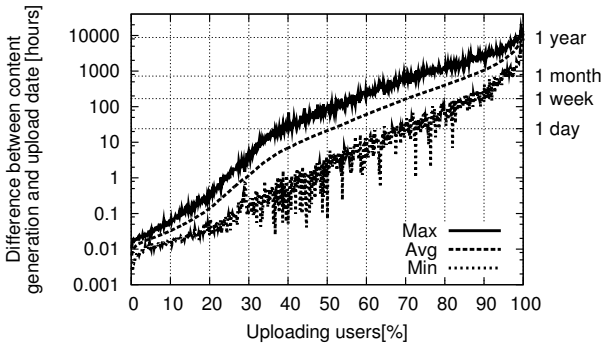


Fig. 6. User delay profiles. Certain users consistently delay most uploads.

respect to their corresponding y axis.

With regards to users delaying content we use the same data as described in Figure 4. Figure 6 shows the content delaying profiles of the users in our trace (as in the previous figure all users are captured on the x axis, one point on the x axis being one user). Three curves are shown in the figure. For each user on the x axis we have a minimum value, a maximum value, and an average value for all the delays observed for that user. We can note that the figure captures the fact that distinct users prefer to upload content right away and other distinct users prefer to almost always delay uploads. In fact 48% of the users delay uploads by more than one day. We call these users *heavy delayers*.

III. Methodology

In this section, we will introduce and analyze the mechanics of our approach for providing better infrastructure for content delivery at certain special locations. Some content can be marked as postponed for delivery by the user and will be delivered only at these locations that have better connectivity. Below we introduce the specific methodology we use for identifying candidate locations for better connectivity.

A. Problem Statement

Our Drop Zone placement problem formulation is based on the following observations. First, that users already inherently postpone delivery of content after generating the same as shown via Figure 4. Further, we argue that once an architecture such as that proposed here is in place, users can be given the option to either deliver content immediately (using whatever type of infrastructure is available at the current location) or asked about how much delivery delay are they willing to tolerate. Hence, in the Drop Zone placement problem, we assume a tolerable delivery delay for all users to come up with a placement. Second, due to users' mobility patterns, there exist a set of *common locations*, through which many users pass by at some point in time. Hence, in our problem formulation, we combine the two observations and determine the common locations through which users will pass by after generating content within the tolerable delay assumed. The Drop Zone placement problem can be stated:

Problem Statement 1: Given:

- B base-stations and U nomadic users with the associated tempo-spatial mobility patterns, *i.e.*, which base-station is serving each user at any time;
- a description of the temporal content generation process for each user, *i.e.*, number of content units being generated by any user at any point in time;
- for all content, a description of the delay that would be encountered by content generated by a user at time t_i, if it is delivered at time t_j, that is quite simply: $t_j - t_i$;

Find the minimum number of Drop Zones to be co-located at the base-stations, such as to satisfy the below constraints:

- the amount of content that a Drop Zone can deliver at a point in time is less than a maximum capacity, (in terms of aggregated rate across users);
- the delay between original and postponed delivery for any content in the system is less than a maximum tolerable delay.

1) Inputs: For the Drop Zone placement problem, we use a one week long trace from one of the largest cellular providers in North America. The trace provides information about users' trajectories in terms of locations (base-stations) they were present at, and at what time. The trace provides details about user's activity at times when (s)he was active.

While the trace provides the tuple, "time, location, content" per user, we extract separately the content uploaded by a user and the user's trajectory. First, we define a single indivisible unit of content as *content chunk* of maximum λ bits. We assume that any solution to the Drop Zone placement problem must ensure that a content chunk is delivered from within one location only. We divide time into discrete units of length τ seconds each, such that the entire trace spans over the set of bins: $T = \{t_1, t_2, ..., t_T\}$. So a user's trajectory that straddles across two time bins is modified such that it begins at the beginning of the first bin and ends at end of the second bin.

What we need as variables in our algorithm is a mapping of content to the base stations where it might potentially be delivered by means of delay and user travel. We formalize as follows. Let C be the set of content chunks, $c \in C$ denote a chunk and $|c|$ represent the size of a chunk in bits. let the function $u = ctou(c)$ map the user $u \in U$ who produced the content chunk c. Let $T = \{t_1, t_2, ..., t_T\}$ be the sequence of temporal snapshots at which the system is observed. Let Δ_c^i represent the number of new content bits generated by the user for content chunk $c \in C$ at time $t_i \in T$. Let R_c^{ij} be the delay for content chunk $c \in C$ generated at time $t_i \in T$ and

delivered at time $t_j \in T$ with $t_i \leq t_j$. Since the base stations necessarily have finite capacity let $\zeta_b^{\max}$ be the maximum number of content bits that can be uploaded at the Drop Zone placed at base-station $b \in B$ within any time bin, and since we do not want content delivery to be postponed indefinitely let $D^{\max}$ be the maximum delay allowed for any content chunk to be uploaded since its generation. Furthermore, let $n_c^i \in \{0, 1\}$ indicate whether content chunk $c \in C$ was generated at time t_i (i.e., $n_c^i = 1$) or not (i.e., $n_c^i = 0$). Similarly, let $m_c^{jb} \in 0, 1$ indicate whether user u corresponding to content chunk $c \in C$ is covered by base-station $b \in B$ at time t_j (i.e., $m_c^{jb} = 1$) or not (i.e., $m_c^{jb} = 0$).

We explain the variables via an example. Suppose from the trace, we obtain information about a user who uploads a content of 10 Kb in a session over which the user was present at base-station B_1. Let the maximum chunk size $\lambda = 10$ Kb, time bins of $\tau = 1$ minute, maximum capacity at any Drop Zone, $\zeta_b^{\max} = 100$ Kbps, $\forall b \in B$, and maximum tolerable delay $D^{\max} = 5$ minutes. Let the binned trajectory for the user be $\{B_1, 0, 0, B_2, 0, B_3, B_3, 0\}$ over $T = \{t_1, ..., t_8\}$. We assume that the content generation time for the chunk is the beginning of time bin t_1. Hence, $\Delta_{c_1}^i = \{10, 0, 0, 0, 0, 0, 0, 0\}$ over $t_i \in T$. The variable $R_{c_1}^{1j}$ provides an idea of delay that is suffered by content c_1 generated at t_1, if it is delivered at time $t_j \in T$ and is given as: $R_{c_1}^{1j} = \{0, 1, 2, 3, 4, 5, \infty, \infty\}$. This variable computes the time delay regardless if the user's presence is known or not at the time, *e.g.,* at time t_5, the value is 4. User u is known to have been in location B_3 at time bins t_6, t_7 as well. However, the content can be delivered in B_3 at time t_6 only and not at t_7 since that would violate the maximum tolerable delay of 5 minutes. Hence, the value ∞ we have at time t_7 as well as t_8. Finally, $n_{c_1}^i = \{1, 0, 0, 0, 0, 0, 0, 0\}$, $m_{c_1}^{jB_1} = \{1, 0, 0, 0, 0, 0, 0, 0\}$, $m_{c_1}^{jB_2} = \{0, 0, 0, 1, 0, 0, 0, 0\}$, $m_{c_1}^{jB_3} = \{0, 0, 0, 0, 0, 1, 1, 0\}$, over T. This notation is further used in Appendix A where we introduce an Integer Linear Program.

B. Greedy Algorithm

As described in the problem formulation above, we wish to place the minimum number of Drop Zones that would cover all the content that was uploaded originally (under no delivery postponement) under a maximum tolerable delay. This can be mapped to a set covering problem, where given a universe set of content, and given a set of base-stations, where each base-station covers a subset of the content universe, we are interested in choosing the minimum number of base-stations that cover the entire content universe set. Determining the minimum cover in the set covering problem is a well known NP-Hard problem [23]. Given the large size of the data we are dealing with (a cover over a set of several millions of elements), in this paper we take a Greedy approach as shown in Algorithm 1. It has been shown [27], that the worst case approximation ratio achieved by our Greedy algorithm when base station capacity is ignored is $H(s)$, *i.e.,* the solution achieved by Greedy can not be more than $H(s)$ times worse than optimal. In our case, s is the number of distinct content chunks covered by the base-station that covers the maximum number of distinct content chunks and $H(s)$ is the corresponding Harmonic number given as: $H(s) = \sum_{k=1}^{s} 1/k \leq ln(s) + 1$.

The greedy algorithm is iterative and determines which base-stations should be considered for placing Drop Zones until all content is covered by at least one Drop Zone. At each step, the greedy algorithm selects the base-station that has the maximum number of distinct content chunks that have not been covered yet. While the algorithm is intrinsically similar to the greedy set cover algorithm, in addition, it incorporates the capacity constraint, that the aggregate content uploaded from candidate Drop Zones should not exceed a maximum capacity (in terms of content bits per time unit). In this regards, the algorithm assigns priority to each uncovered content in terms of how many chances a content has to be covered. More precisely, the priority of each uncovered content at a base-station is computed as an aggregate of unused capacity across time bins where the user corresponding to this content was present, since the content was originally uploaded and within the maximum tolerable delay bound (see Function 2).

Algorithm 1 Greedy algorithm to determine which base-stations serve as candidate Drop Zones

```
Initialize X = ∅, where X is set of base-stations selected as Drop Zones.
Create C = Set of content chunks in the system over all t_i ∈ T.
Create B = Set of base-stations at which we have at least one chunk not yet covered, c ∈ C at any time.
Create ζ(b, t_i) = Unused capacity at base-station b at time bin t_i.
At any time, ζ(b, t_i) ≤ ζ_b^max.
while |C| > 0 do
    b=RankBaseStations(B);
    X = X ∪ b;
    RC=RankContent-AT-BaseStation(C, b);
    for (c,b) in RC do
        t_h=DeliverContent (c,b);
        if t_h ≠ −1 then
            ζ(b, t_h) = ζ(b, t_h) − |c|;
            C = C − c;
            RC=RankContent-AT-BaseStation(C, b);
        end if
    end for
    Create B;
end while
```

Function 1: **RankBaseStations(B)** assigns priority to base-stations $b \in B$ by counting the maximum number of distinct content chunks not yet covered, that can be served by each base-station over all time $t_i \in T$. It then sorts these base-stations in ascending order and returns the base-station with largest number of distinct content chunks.

Function 2: **RankContent-AT-BaseStation(C,b)** assigns priority to each content chunk $c \in C$ served by input base-station b by counting the number of capacity units that content chunk c will have at base-station b within the time the content was originally uploaded (t_i) and the maximum tolerable delay, *i.e.* $t_j \in [t_i, t_i + D^{\max}]$. Then it sorts these pairs in ascending order, with the most critical pair as the first one to be served, *i.e.* with the fewest number of capacity units available to it for being served. It returns this list in RC=(c,b).

Function 3: **DeliverContent(c,b)** delivers the content c at base-station b by selecting the earliest time bin $t_h \in [t_i, t_i +$

$D^{\max}$] at which $\zeta(b, t_h) > 0$. Then it returns the time bin t_h. It returns $t_h = -1$, in case no time bin is available.

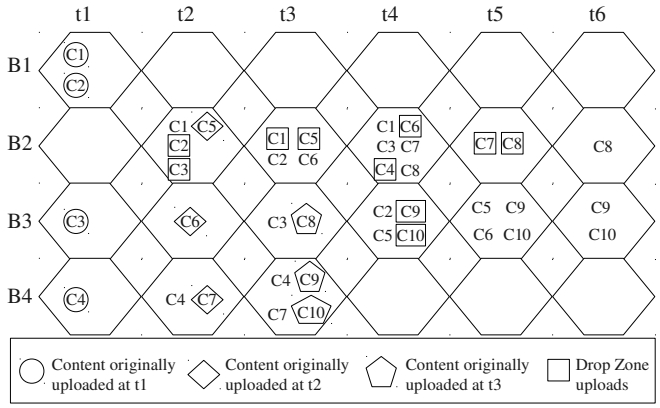


Fig. 7. Example for Greedy Drop Zone Placement.

We further explain the Greedy Drop Zone placement algorithm (see Algorithm 1 via an example as shown in Figure 7. Assume time bins of length $\tau = 1$ bin and content is chunked in to sizes of 1 unit. Further, assume maximum capacity at any Drop Zone is 2 units per time bin and maximum tolerable delay to deliver any content chunk since original upload is $D^{\max} = 3$ time bins. Initialize the unused capacity at each base-station b for each time bin t_i as $\zeta(b, t_i) = 2$. Assume the following original content upload process, where at time t_1, the following chunks were first uploaded: $\{c_1, c_2, c_3, c_4\}$, at time t_2: $\{c_5, c_6, c_7\}$ and at t_3: $\{c_8, c_9, c_{10}\}$. For simplicity, we assume that all content chunks are of the same size: 1 unit. Moreover, the user corresponding to content chunk c_1 had the following trajectory: (B_1, t_1), (B_2, t_2), (B_2, t_3) and (B_2, t_4). Note from Figure 7, that the content that was originally uploaded at time t_1 is shown with circles, that in t_2 with diamonds and in t_3 with pentagons. Content that was not uploaded at a location, but it could be potentially delivered there by virtue of user's movement there, is shown in plain black. Hence, note that while the user corresponding to content c_1 did not upload any new content at time bins t_2, t_3, t_4, these times are candidates for him to still upload content in a postponed manner. The final allocation of which content is uploaded from which Drop Zone is as shown by content enclosed in squares.

The algorithm initializes: $X = \{\}$, $C = \{c_1, ..., c_{10}\}$ and $B = \{B_1, B_2, B_3, B_4\}$. At the first Iteration ($|C| = 10$ that is > 0) and hence, we evaluate RankBaseStations(B) to obtain $\{(B_1, 2), (B_2, 8), (B_3, 7), (B_4, 4)\}$ and output $b = B_2$ and $X = \{B2\}$. RankContent-AT-BaseStation(C, B_2) produces the following initial set of priorities for users at B2, $\{(c_1, 6)$, $(c_2, 4)$, $(c_3, 4)$, $(c_4, 2)$, $(c_5, 4)$, $(c_6, 4)$, $(c_7, 4)$, $(c_8, 6)\}$. Hence, first content that is selected is $(c_4, 2)$ that is then delivered at time t_4 as determined by DeliverContent(c_4, B_2). Next, we reduce unused capacity by one to obtain $\zeta(B_2, t_4) = 1$, and modify the set of content chunks not yet placed as $C = C - c_4$. Next, RankContent-AT-BaseStation(C, B_2) produces the following set of priorities amongst users at B2, $\{(c_1, 5), (c_2, 4), (c_3, 3), (c_5, 4), (c_6, 3), (c_7, 3), (c_8, 5)\}$. Next content to be selected is $(c_3, 3)$, that is then delivered at B_2 at time t_2. Thus, we reduce unused capacity at B_2 at time t_2 by one unit and obtain $\zeta(B_2, t_2)$ as 1, and set $C = C - c_3$. We continue this way and produce the following final allocation of content at B_2: (c_4, B_2, t_4), (c_3, B_2, t_2), (c_2, B_2, t_2), (c_5, B_2, t_3), (c_1, B_2, t_3), (c_6, B_2, t_4), (c_7, B_2, t_5), (c_8, B_2, t_5). We exit the for loop with $C = \{c_9, c_{10}\}$, and $B = \{B_3, B_4\}$. In the second iteration, $|C| = 2 > 0$, and hence we enter the while loop. RankBaseStations(B) produces the following $\{(B_3, 2), (B_4, 2)\}$, and since both base-stations are of equal priority, it randomly chooses $b = B_3$. Hence, the final Drop Zone placement as output by Greedy algorithm is $X = \{B_2, B_3\}$. Finally, RankContent-AT-BaseStation(C, B_3) produces the following set of priorities for users at B_3: $\{(c_9, 6), (c_{10}, 6)\}$ and the consequent allocation $(c_9, B_3, t_4), (c_{10}, B_3, t_4)$.

C. Parameters

In the next section, we evaluate the performance of Greedy and Optimal algorithms (described in Appendix A). Where not specified, we use the following values for parameters. We assume $\tau = 1$ minute, *i.e.*, time is divided in to bins of length 1 minute. We evaluate the performance of the algorithms assuming that Drop Zones are to be serviced by LTE, and hence we use the maximum capacity at any Drop Zone, ζ_b^{max} to be 75 Mbps, $\forall b \in B$. Many factors such as errors due to signal propagation obviously decrease this aggregate capacity yet we ignore them for the purpose of this study as we do not have access to them. We choose maximum chunk size $\lambda = 3.5$ MB as this is the biggest content piece in our dataset and can safely fit in one minute considering the LTE technology. We vary the maximum tolerable delay over the duration of the trace, as [1-168] hours.

IV. Evaluation

In this section we evaluate the Drop Zone architecture and the effectiveness of various infrastructure placement algorithms. We then explore multiple system parameters and their impact on performance.

A. Greedy vs. Optimal

Here, we present results to compare the placement obtained by the Greedy algorithm with respect to the Optimal shown in Appendix A. In the following experiments, we assume the maximum capacity of each Drop Zone to be the same as the maximum aggregate upload rate possible under LTE, 75 Mbps *i.e.* $\forall b \in B, \zeta_b^{\max} = \zeta^{\max} = 75$ Mbps. We solve the Integer Linear Program by using the ILOG CPLEX software [8]. Because of the large scale of the data involved, we compare the optimal placement given by the Integer Linear Program with our Greedy algorithm on a limited dataset extracted from the original dataset. We extract uploads across 98 base-stations that cover a medium size United States town. We only extract uploads originally carried out across the first day of our dataset. Figure 8 shows the results. We vary the maximum postponed delivery interval among the values of 6, 12, 18, 24, 48, 72, and 96 hours. The reduced dataset contains uploads originally carried out over a single day, yet we also extract

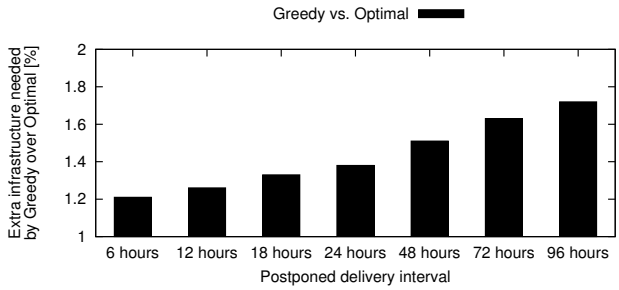


Fig. 8. Greedy placement compared to the Optimal placement obtained from the Integer Linear Program.

the upload opportunities that such content has across the above longer time intervals.

The insights from Figure 8 are as follows. First and foremost, Greedy stays very close to optimal. Indeed, for all maximum postponed delivery intervals we considered, Greedy selects only 1%-2% more Drop Zones than the optimal placement does. Second, we can see a tendency for Greedy to select a relatively larger number of Drop Zones as compared to optimal, when the maximum postponed delivery interval increases. We make two points here: (i) despite the increased difference, the absolute difference is still very small, *i.e.*, less than 2% in all cases. (ii) We will demonstrate below that in any case we cannot obtain significant gains for maximum postponed delivery intervals greater than a few days.

B. Greedy vs. Greedy Zero

Here, we evaluate the impact of postponed content delivery intervals on the infrastructural requirements needed by the Drop Zone approach. For comparison, we use Greedy Zero, an instance of our Greedy algorithm that greedily selects as Drop Zones the locations from where users originally uploaded the largest quantities of content and evaluates them under the considered postponed delivery assumption.

Figure 9 shows the results. The x-axis shows the number of Drop Zones, while the y-axis shows the ratio of the content delivered by our Greedy algorithm vs. Greedy Zero. For example, point (x,y) = (200,1.24) shows that the Greedy algorithm manages to deliver 24% more content than the Greedy Zero algorithm when 200 Drop Zones are used in both cases and a maximum postponed delivery interval of 96 hours is considered. This is not a surprise: when the postponed delivery is considered during the selection process, locations that can deliver more content in a postponed manner are selected. Thus, a better infrastructural placement is possible to achieve, and hence more content is delivered.

Figure 9 shows that the Greedy approach manages to deliver approximately 5%-25% more content than Greedy Zero. Necessarily, the gap between the two steadily increases as the maximum postponed delivery interval increases. Also, the gap between the algorithms is particularly high within the first 200 Drop Zones. This happens because the Greedy algorithm manages to quickly select locations that were not so popular originally, yet they are excellent Drop Zone locations when postponed delivery is considered during the selection process. Because Greedy Zero has no advanced knowledge about user

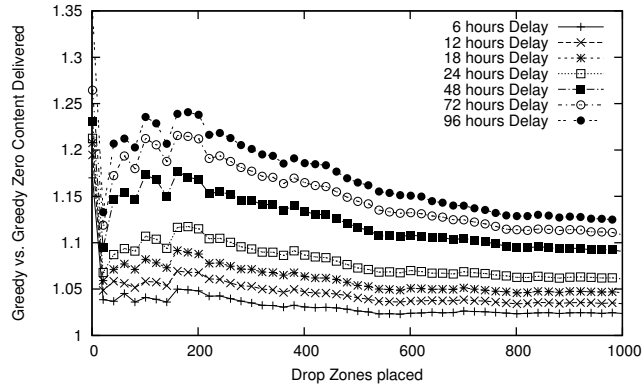


Fig. 9. Greedy Drop Zone placement compared to Drop Zone placement based on popular locations (ranked by content delivered).

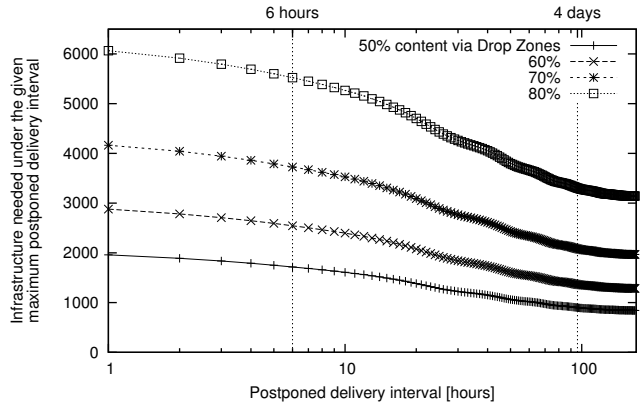


Fig. 10. Number of locations where Drop Zones are installed when postponing delivery.

mobility, it either neglects such locations or selects them much later in the process.

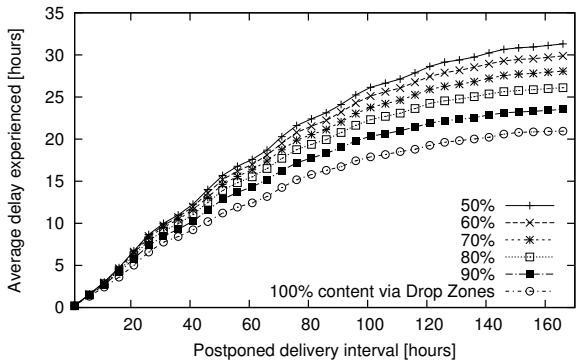
Furthermore, the gaps shown in the figure translate to additional infrastructure the order of a few hundred additional Drop Zones needed to deliver the particular amount of content. In particular, for 1,000 Drop Zones placed by Greedy with 96 hours postponed delivery, the Greedy Zero needs 1,201 Drop Zones (the result is not shown in the figure). Thus, an approach that does not consider user mobility and postponed content delivery during the selection process requires 20% larger infrastructural deployment to achieve the same performance.

C. Infrastructural Needs

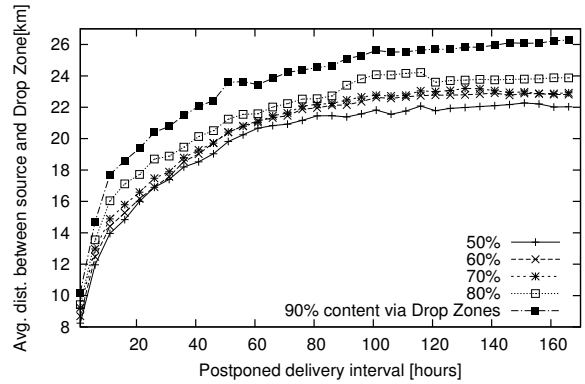
Here, we explore the infrastructural needs as a function of postponed delivery intervals. In this scenario, we take the percent of delivered content as a parameter.

Figure 10 shows the results. It depicts the number of Drop Zones (y-axis) needed to serve the given percent of content by assuming the maximum postponed delivery interval (x-axis) varied in the range from 1 to 168 hours. Necessarily, Drop Zone architectures that target to absorb larger amounts of traffic need more Drop Zone locations. Indeed, to deliver 80% of traffic via Drop Zones for 1 hour postponed delivery interval, one needs to deploy three times more Drop Zones (6,066 vs. 1,960) relative to the 50% content case.

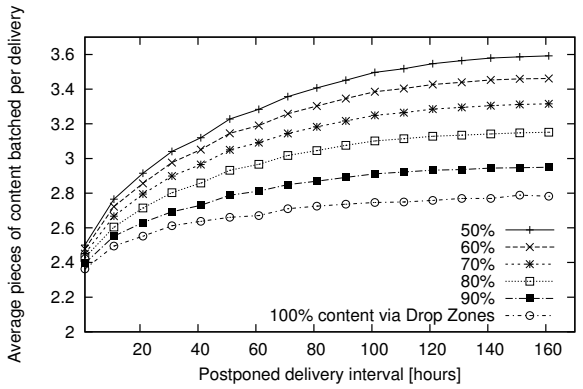
Another insight is that the Drop Zone deployment rate reduces as the postponed delivery increases. Note that the



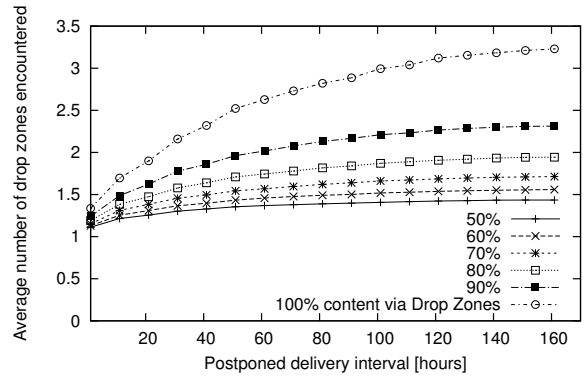
(a) Average delay experienced by content.



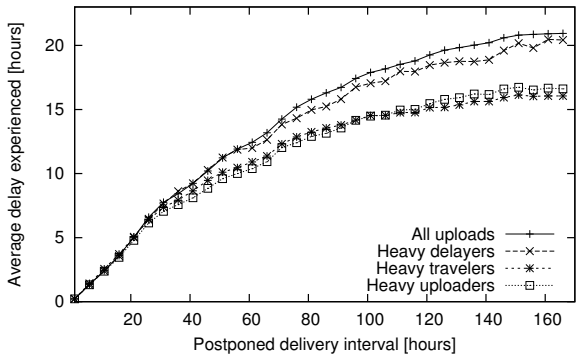
(b) Average distance between source and Drop Zone.



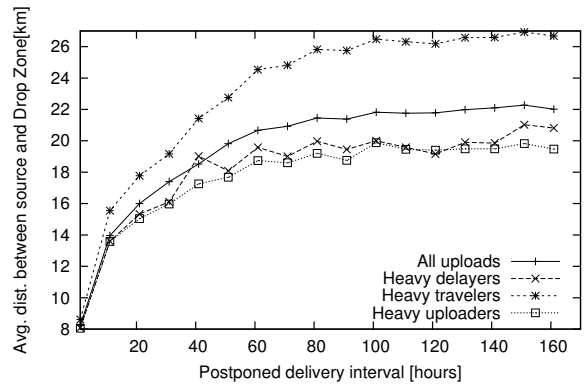
(c) Average number of content pieces batched per delivery.



(d) Average number of Drop Zones a user interacts with during the trace interval.



(e) Average delay experienced by content for different user types - 100% of content delivered via Drop Zones



(f) Average distance between source and Drop Zone for different user types. - 50% of content delivered via Drop Zones

Fig. 11. Infrastructural usage

largest benefits come early. Focusing on the 50% content delivery case, one needs Drop Zones in 12% less places when comparing 1 hour (1,960 Drop Zones needed) to 6 hours for maximum postponed delivery (1,716 Drop Zones needed). This is because the probability that users change their location within 6 hours intervals is high. Thus, it becomes possible to offload the same content at a smaller number of Drop Zones.

Figure 10 shows that all curves 'flatten' as the postponed delivery interval increases over 4 days. One would expect that as the postponed delivery interval increases, users see more locations, and hence, infinite gains can be obtained from mobility. This is not the case as shown in Figure 10.

Previous studies on human mobility, reported on the high predictability of movement and observed that users spend significant time in just a few locations. For example in [30], the authors note that even for users seen at as much as 30 or 50 different locations they spend more than 40% of their time in just two locations. This effect can be observed in Figure 10. After a time interval of approximately 4 days, the curves level off and marginal gains can be obtained on increasing the delivery interval. Our explanation is that since users spend time in a few locations (as shown in previous work), benefits in Drop Zone placement come from considering these locations. However, as the time interval increases, the probability to visit other locations increases. However, after 4 days, it is unlikely for users to visit locations not seen before.

D. Infrastructural Usage

Here, we analyze how users will interact with the Drop Zone architecture. We explore the following aspects: (*i*) Average

content delay: even though we specify a maximum postponed delivery interval, content could be delivered much earlier; hence, we take a look at the actual delay experienced by users, (ii) average distance between source and Drop Zone, (iii) average pieces of content batched: since users postpone content delivery, they might carry a larger number of pieces of content when encountering a Drop Zone, and (iv) average number of drop zones encountered during the seven day interval. In all scenarios, we take the percent of delivered content as a parameter. For the delivery, we have users opportunistically deliver their postponed content upon encountering the first Drop Zone with available capacity to deliver the content.

Figure 11(a) shows the actual delay experienced by users (y-axis) considering the given postponed delivery interval (x-axis). Necessarily, the experienced delay is shorter than the maximum postponed interval shown on the x axis. Indeed, the scale on y-axis is approximately 5 times shorter than on the x-axis. Another insight is that delay grows sub-linearly with the postponed delivery interval in the range (x,y) = (1hour, 15 minutes) to (100 hours, 25 hours). In all cases, users experience on average four times less delay than given by the maximum postponed delivery interval.

Figure 11(b) shows the actual average distance between the source and the Drop Zone. The figure shows that the average distance increases with the increase in content delivered by Drop Zones. A larger amount of content delivered implies a larger number of Drop Zones. This further means that the average distance between the source and the Drop Zone increases with the number of Drop Zones. This result may seem counter intuitive at first. Indeed, if there are more Drop Zones, they should be on average closer to users, not further away. By examining the data we realize that the reason is as follows: when there are a smaller number of Drop Zones, there is still a large number of users close to those locations. Hence, the smaller distance. As the number of Drop Zones increases, users who are already close to existing Drop Zones are further covered, while the larger number of Drop Zones singles out the users who are further away. Hence, the larger distance.

Figure 11(c) shows the average number of pieces batched. As mentioned above, batching content delivery is beneficial for a mobile device as it improves battery life [19]. The figure shows that in all Drop Zone placements, users deliver on average 2.4 more content per delivery for 1 hour postponed delivery interval. As the delivery interval increases, so does the batching effect. As expected, more content pieces are batched with less Drop Zones, corresponding to smaller percent of content (*e.g.*, 50%) uploaded via Drop Zones.

Figure 11(d) shows the average number of Drop Zones that users interact with during the seven day trace interval. As users 'see' only a few base-stations that are part of their predictable daily routine, the Drop Zone usage necessarily captures this effect. Hence, users interact with a small number of Drop Zones on average, *i.e.*, 1-3.5, depending on the amount of Drop Zones placed. The more content the infrastructure needs to absorb, the larger the number of Drop Zones is, and hence the larger the number of Drop Zones that users encounter.

E. Heavy Uploaders, Travellers, and Delayers

Finally, we evaluate how the users we identified in Section II-B4 perform with respect to the Drop Zone infrastructure. Figures 11(e), and 11(f) show the results. In particular, we have examined the delays, and distances experienced by heavy travelers, heavy uploaders, and heavy delayers. Figure 11(e) shows the average delay. One can see that the heavy travelers and heavy uploaders fare better in terms of average delay than the rest of the users. In fact this shows that the *heavy tail* of uploads is made up of users who do not upload, or move as much. Note that Figure 11(e) shows only the case when 100% of content is delivered via Drop Zones but all other curves behave similarly (not shown).

The average distance is shown in Figure 11(f). Here, one can see that the heavy uploaders have on average a shorter distance between them and a Drop Zone. This is natural as our algorithm tries to capture as many upload opportunities as possible, and as soon as possible. Heavy travelers are on average further away than the rest since heavy movement is not particularly related to heavy uploads (what our algorithm tries to optimize for).

F. What-If Scenarios

One needs to understand that there are multiple (often unpredictable) ways in which the mobile world can evolve. In particular, some trends might involve large increases in the number and percentages of users with smart devices, an increase in the number of mobile smart devices per-user, increases in the desire for real-time high-fidelity audio/video communications, etc. Because of this, we aim to address the following 3 problems: (i) how would our architecture deal with an exponential increase in content size in the future, (ii) what benefits might arise from users deliberately changing movement patterns to drop off content or from bigger radius wireless technologies, and (iii) what are the number of creation date? In all the cases below, we analyze the impact of a Drop Zone architecture cover 50% of the content for the maximum postponed delivery intervals of: 6 hours (1,717 Drop Zones), 24 hours (1,303 Drop Zones), and 72 hours (963 Drop Zones). In all cases, we assume the maximum upload capacity of 75 Mbps, corresponding to the LTE technology. We agree that studied in isolation some of the above scenarios can be seen as limited, however we consider them a useful exercise for predicting future growth to a limited extent.

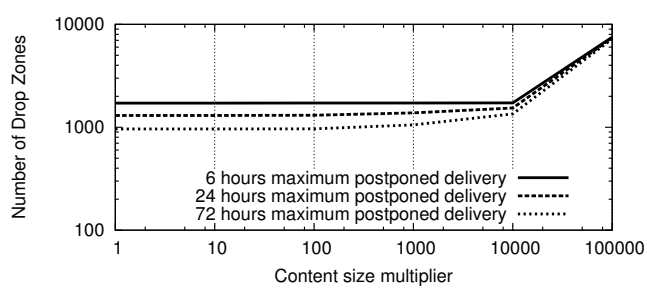


Fig. 12. Increase in infrastructure due to increasing the size of content.

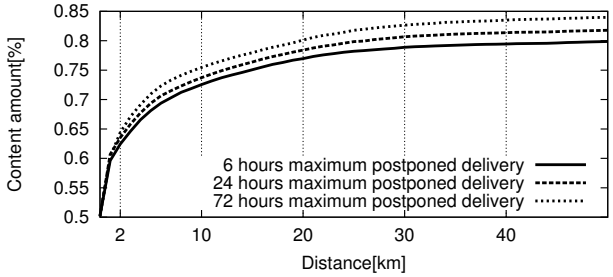


Fig. 13. Analysis of the impact of movement changes on content delivery

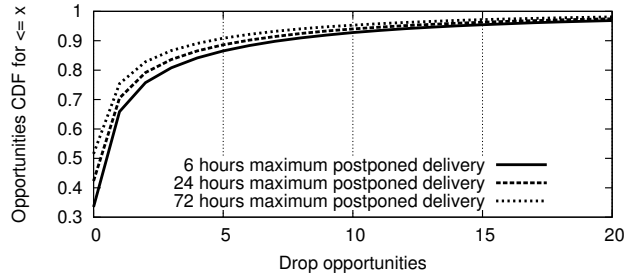


Fig. 14. Missed opportunities for pieces of content that we know the creation date.

1) Content size increase: Here, we try to understand how the proposed architecture deals with an increase in content size in the future. Figure 12 shows the results obtained. In particular, we increase the content size in our trace by the multiplier shown on the x axis in the figure, and rerun the Greedy placement. The number of Drop Zones is shown on the y axis. Note that our Drop Zone architecture can handle a five order of magnitude size increase, *i.e.*, 10,000. If we assume that the amount of content doubles every year, this gives approximately 14 years lifetime under the 75 Mbps LTE technology assumption. Further, an increase beyond a 10,000 multiplier would require a deployment of a significantly larger number of Drop Zones, as shown in Figure 12 for x=100,000, or an increase in the capacity of the existing Drop Zones.

By considering only the heavy uploaders in the above scenario, we still require close to the same amount of infrastructure (just 8% less) to serve them in the first place without considering content increase. The reason is that the heavy uploaders are not particularly clustered in just a few locations. When considering content increase we still witness the same substantial increase in infrastructure needs at 5 orders of magnitude. So the heavy uploaders are the ones driving the infrastructure changes.

2) Drop Zones influencing movement: Here, we try to quantify benefits that might arise if people would change their movement patterns to explicitly deliver content via Drop Zones or if Drop Zones would employ a higher radius wireless technology. Movement might occur on the basis of an application pointing them the areas close by with a better connectivity. While it is hard to predict whether such behavioral change is a viable option in the future, we nonetheless argue that it is worth quantifying gains of such potential scenarios. We rerun the Greedy placement algorithm with the following modification. When considering a base-station for a Drop Zone, we include the content that comes from the base-stations located at a distance given by the number of kilometers shown on the x axis in Figure 13. The y-axis shows the amount of content that would have been delivered in such a scenario. In the case of movement, we thus make the assumption that if a user is at a given distance from a certain Drop Zone, he would choose to travel that distance to deliver his content. In this way, we aim to quantify how close in space is the content to the actual Drop Zones we have placed.

Figure 13 shows that considering just content that is delivered 2 kilometers away from the given Drop Zone placements, we manage to cover more than 60% of the content for all maximum postponed delivery intervals we analyzed. This is more than the baseline when there are no changes to users' movement patterns, when 50% of content is covered by the Drop Zones in all scenarios (see $x = 0$). This further fortifies our finding stated in the context of Figure 11(b): for a smaller number of Drop Zones, there exists a large number of users close to those locations. Hence, perturbing their movement slightly brings big gains. Note also from Figure 13 that ranking of the curves suggest that even though we have a large number of Drop Zones for the 6 hours postponed delivery intervals (1,717 Drop Zones) than for the 72 maximum postponed delivery interval (963 Drop Zones), we manage to deliver more content in the later case. This is due to the increased time interval (72 hours vs. 6 hours) that brings more content closer to a smaller number of Drop Zones.

3) Missed connections: Here, we focus on a subset of users that produce and upload content for which we know the creation date, *i.e.*, photos, in a postponed manner, as explained in Section II-B3 above. In particular, we try to quantify missed upload opportunities for this content. For example, if content is created at time t1, and it is uploaded by the user at time t2, we explore how many locations our algorithm upgraded to Drop Zones did the user visit between t1 and t2.

Figure 14 shows the CDF of upload opportunities for considered content. Our figure shows that approximately 50% of users see no upload opportunity. This is consistent with insights from Figure 4, that shows that around 50% of users upload their content within the first hour (within or outside a Drop Zone). Somewhat counter intuitively, Figure 14 further shows that the Drop Zones for shorter maximum postponed intervals provide more postponed opportunities. For example, for Drop Zones for postponed intervals of 6 hours, the probability to have more than 5 opportunities is 0.15, (= 1-0.85), corresponding to (x,y) = (5, 0.85) in the figure. On the other side, the probability to have more than 5 drop opportunities is approximately 0.1, (= 1-0.9), corresponding to (x,y) = (5, 0.9) for the Drop Zones placed for the 72 hours postponed delivery interval. This happens because the shorter postponed delivery interval incurs more Drop Zones, *i.e.* 1,717 vs. 963, hence the larger number of drop opportunities exist.

V. Network Architecture

The Drop Zone architecture implementation that we describe in this section was motivated by two requirements that we subsequently incorporated in the design: (i) as noted above we wish to minimize the network impact that concentrating uploads in a smaller number of locations might have, (ii)

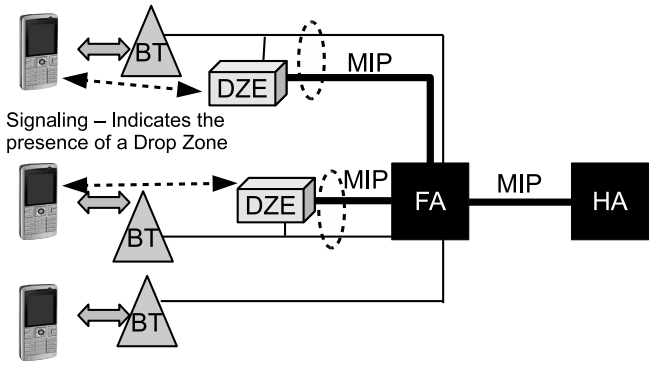


Fig. 15. Network architecture. Some base-stations (BT) are enhanced with Drop Zone Equipment.

the equipment and implementation we suggest should use the available standards as they are.

Note that this design was obtained from an existing (IPv4 based) cellular network in which we have integrated our Drop Zone architecture. The usage of a Foreign Agent in IPv4 implies MIPv4 that is currently not compatible with IPv6 and inconsistent with the portions of 3GPP2 specifications that are built on Ipv6. Switching to IPv6 would imply changing the some Drop Zone functionality also.

In current mobile networks, a base-station contains 3 radio sectors and a switch/router. The router has a T1/T3 line that goes out to the Relay Network Control (RNC) that is also known as the Base Station Controller (BSC). This T1/T3 line is referred to as the back haul link. The radio interface is the most stressed in a typical 3G network. The second most stressed portion is the back haul link. In architectures such as ours where the radio link in drop zones would have a higher capacity, the back haul link will be the main bottleneck and some expensive ways to increase the capacity of the back haul link include: (i) using optical fiber, (ii) connecting to a third party that runs a metropolitan area network (usually Ethernet based), (iii) microwave point to point connections, (iv) WiMax-D.

Regardless of the rate at which the radio interface and back haul capacity will be improved in newer networks, it is reasonable to expect that the user appetite for content will grow faster than capacity. This fact provides a solid motivation for why postponing content uploads can serve to reduce the utilization on older technology base-stations and at the already stressed back haul links.

Regarding our main addition, the Drop Zone Equipment, it consists of two interfaces: (i) the first interface called the Mobile IP interface has an IP-Address assigned by the Home Agent (the Drop Zone Equipment appears as a Mobile IP client towards the Foreign Agent/Home Agent and has Mobile IP capability as a client, (ii) the second interface is the sniffer that sniffs all traffic that goes in and out of the base-station. The reason for the sniffing interface is that this way, we do not need to change the way the base station functions. It still receives and handles traffic for the mobile phone yet the traffic is stopped from going over the backhaul link.

On the device we have a simple uploader application. The user can select a list of services that his files can be uploaded to (Facebook, Twitter, MySpace etc.). Whenever he wants to upload a set of files he goes to the application, selects the files, and sets a maximum delay he tolerates or if he wants them to be uploaded instantly. Even now the photo camera of particular phones is integrated with Facebook. Users are directly being asked upon snap if they want to share the photo via Facebook or Flickr and the photo is uploaded when network connectivity is available. In our scenario they will also be asked if they can tolerate upload delays in which case the picture upload is handled by a daemon that uploads it only when a Drop Zone is available. Newer services can also be integrated with the phone camera and given the same options for upload.

The flow for the transfer goes as this. If the client has data to send but the current tower is not Drop Zone enabled it waits for a tower which is Drop Zone enabled. When it gets handed over to a tower that is, it sends a packet on a specific IP address that is seen by the Drop Zone Equipment at the tower. The Drop Zone Equipment reads the packet (the packet has inside the client's IP address). The Drop Zone Equipment sends a message to the client on the port listed in the signaling packet telling it that it is present at the tower and to start sending the data (detecting if a Drop Zone is present can be done as simple as sending a packet and obtaining a reply from a special IP address such as 0.0.0.0 with the Drop Zone equipment providing the reply). The client starts sending the data and The Drop Zone Equipment picks it up. The stream is stopped from crossing the back haul link by an Access Control List. After receiving it, the Drop Zone Equipment eventually sends the file to the designated server.

VI. Discussion

Advanced Content Drop-off. In our evaluation of delay experienced by users who drop-off content at a Drop Zone, (see Figure 11(a)), we assumed an opportunistic drop-off policy, where a user uploads his content to the first Drop Zone that he meets. A more sophisticated drop-off policy could be deployed by the service provider as follows. The service provider could keep track of locations visited the most by each user, and the times of day when the location is visited. Next, when the user presents content to be uploaded in a postponed manner, the network determines the amount of available capacity at the Drop Zone, that is nearest to the user as well as predicts the capacity that would be available at the next Drop Zone(s) where the user is expected to move. The service provider selects the Drop Zone with the most unused capacity to upload the content. Deploying such a sophisticated delivery mechanism requires prediction of users' trajectories as well as sharing of capacity at each Drop Zones with a centralized system. Several other enhancements could be included such as a user interface that asks the user if he intends to change his normal routine. We consider this a challenging problem out of scope of our current work, but certainly the most interesting problem that we wish to study.

Generality. Although the data that we use for our study represents user generated content uploaded by users, the problem we tackle is more general and any content that potentially has a delay tolerant nature could be delayed and eventually uploaded/downloaded only when users encounter better connectivity options.

Also, it is generally accepted that growth in mobile data demand outpaces growth in capacity provided through upgrades. One only needs to look at the Cisco Global Mobile Data Traffic Forecast [4] and can see that the CAGR for data consumption is at about 92% depending on device while the increase in connection speeds has a CAGR of around 60% (from 2010 to 2015). So, on the long run, providers will most likely have to upgrade their networks again and again making our Drop Zone approach a viable first step to conduct upgrades.

VII. Related Work

The content delay-energy saving trade off has been recognized in several other recent projects [18, 29] that propose offloading 3G data to already existing WiFi networks. We too believe in the effectiveness of such offloading approaches and in this paper we take a more systematic view by considering the natural perspective of a mobile provider that selectively upgrades the cellular network.

When addressing increased load in cellular networks, one argument is on how pricing should be done, *e.g.*, [22] assumes the existence of multiple technologies that offer different performance and focuses on competitive pricing. Others, *e.g.*, [15, 16, 35] assume the existence of a diversity of networks in certain locations and introduce systems for exploiting [15, 35] or predicting locations that manifest such diversity [16]. On the contrary, our goal is to determine where placing such new technology is meaningful. While there is work on placement of relays in multi-hop wireless networks (*e.g.*, [34]) or vehicular networks (*e.g.*, [25]), our work differs as we incorporate both content postponement and mobility in to the problem.

Delay-tolerant networking has been widely studied in the recent years [24]. Most research on delay tolerance for human-carried devices considers encounters between different users (*e.g.*, [26, 28, 31, 33, 38]). Recent work in this area examines how human mobility influences the design of different forwarding algorithms [21], or how performing delay tolerant transfers helps reduce energy consumption of a mobile phone [19]. While we also take delay tolerance as a building block of our approach, our key goal is to study how human mobility influences infrastructural placement at large scale.

Another body of work deals with reducing the amount of data delivered on the wireless interface. Proxies are employed that customize content such as images to specific device characteristics (*e.g.*, device resolution) [17]. Our approach is orthogonal to such approaches. Finally, our work also relates to prior work on predicting user movement to effectively schedule network usage (e.g., [20, 32]). Indeed, by knowing where the user is currently, one could predict his next location based on his past movement history.

VIII. Conclusions

In this paper we have presented a novel cellular network architecture that attempts to deal with the emerging problem of increase in user generated content. The key idea is to selectively upgrade infrastructure in a few select locations we call Drop Zones. We developed and evaluated placement algorithms that position Drop Zones in locations that fall within the daily movement patterns of a large number of users and could manage to deliver larger quantities of content in a postponed manner. We show that users already postpone content uploads in a substantial number of cases and argue that they could be further incentivized to postpone uploads by pricing schemes. We demonstrated that our algorithm manages to place Drop Zones in a way that is very close to optimal. Thus, it can be effectively used by network operators.

Our findings are as follows: (i) A Drop Zone architecture reduces infrastructural deployment requirements by up to 24% relative to a mobility-oblivious and delay-unaware architecture. (ii) Our approach can effectively tame the exponentially increasing user-generated content surge for the next 14 years, under the LTE technology assumption; after that, a faster underlying technology or a much wider Drop Zone deployment must be applied. (iii) Slight perturbation in human movement or slightly bigger coverage wireless technologies can bring substantial gains both for users and network operators; thus, such an approach should be seriously considered. (iv) Considering user profiles in our Drop Zone architecture, heavy uploaders, and heavy travelers fare better in terms of average delay than heavy delayers. On the research side, our key contribution lies in advancing the field in delay tolerant transfers by shifting focus from random interactions between human carried devices to performing infrastructure placement and upgrades at a large network-level scale.

References

[1] AT&T Changes iPhone/iPad Data Plans,shows up Tethering Prices. http://www.tech-mania.com/2010/att-changes-iphoneipad-data-plansshows-up-tethering-prices/.

[2] AT&T: Internet to Hit Full Capacity by 2010. http://www.zdnet.com/news/at-t-internet-to-hit-full-capacity-by-2010/197822.

[3] AT&T Moves Closer to Usage-Based Fees for Data. http://www.computerworld.com/s/article/9142012/AT_T_moves_c\loser_to_usage_based_fees_for_data.

[4] Cisco visual networking index: Global mobile data traffic forecast update, 20-2015. http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-520862.html.

[5] Customers Angered as iPhones Overload AT&T. http://www.nytimes.com/2009/09/03/technology/companies/03att.html.

[6] Facebook Statistics. http://www.facebook.com/press/info.php?statistics.

[7] Flickr Mobile Snaps Group. http://www.flickr.com/groups/96383109@N00/.

[8] ILOG CPLEX: Optimization Software. http://www.ilog.com/products/cplex/.

[9] Lots of iPhone/AT&T woes at CES. http://voices.washingtonpost.com/posttech/2010/01/at_the_worlds_largest_high-tec.html.

[10] LTE. http://www.3gpp.org/article/lte.

[11] Managing Growth and Profits in the Yottabyte Era. http://www.chetansharma.com/Managing_Growth_and_Profits_in_\the_Yottabyte_Era.pdf.

[12] Mobile Phone Photography Group. http://www.flickr.com/groups/mobilephonephotography/.

[13] Mobile Social Networking Set for Growth. http://www.emarketer.com/Article.aspx?R=1006514.

[14] WIMAX Forum. http://www.wimaxforum.org/.

[15] G. Ananthanarayanan, V. N. Padmanabhan, L. Ravindranath, and C. A. Thekkath. Combine: Leveraging the Power of Wireless Peers Through Collaborative Downloading. In *Proc. ACM MOBISYS*, pages 286–298, San Juan, Puerto Rico, 2007.

[16] G. Ananthanarayanan and I. Stoica. Blue-Fi:Enhancing Wi-Fi Performance using Bluetooth Signals. In *Proc. ACM MOBISYS*, pages 249–262, Krakow, Poland, 2009.

[17] T. Armstrong, O. Trescases, C. Amza, and E. de Lara. Efficient and Transparent Dynamic Content Updates for Mobile Clients. In *Proc. ACM MOBISYS*, pages 56–68, Uppsala, Sweden, 2006.

[18] A. Balasubramanian, R. Mahajan, and A. Venkataramani. Augmenting Mobile 3G using WiFi. In *Proc. ACM MOBISYS*, pages 209–222, San Francisco, CA, USA, 2010.

[19] N. Balasubramanian, A. Balasubramanian, and A. Venkataramani. Energy Consumption in Mobile Phones: A Measurement Study and Implications for Network Applications. In *Proc. IMC*, pages 280–293, Chicago, IL, USA, 2009.

[20] A. Bhattacharya and S. K. Das. Lezi-Update: An Information-Theoretic Approach to Track Mobile Users in PCS Networks. In *Proc. ACM MOBICOM*, pages 1–12, Seattle, WA, USA, 1999.

[21] A. Chaintreau, P. Hui, C. Diot, R. Gass, and J. Scott. Impact of Human Mobility on Opportunistic Forwarding Algorithms. *IEEE Transactions on Mobile Computing*, 6:606–620, June 2007.

[22] R. Chakravorty, S. Agarwal, S. Banerjee, and I. Pratt. MoB: a Mobile Bazaar for Wide-Area Wireless Services. In *Proc. ACM MOBICOM*, pages 228–242, Cologne, Germany, 2005.

[23] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. Introduction to Algorithms 2nd edition. McGraw-Hill Science/Engineering/Math, 2003.

[24] K. Fall. A Delay-Tolerant Network Architecture for Challenged Internets. In *Proc. ACM SIGCOMM*, pages 27–34, Karlsruhe, Germany, 2003.

[25] F. Farahmand, I. Cerutti, A. Patel, J. Jue, and J. Rodrigues. Performance of Vehicular Delay-Tolerant Networks with Relay Nodes. volume 11, pages 929–938. John Wiley and Sons, Ltd., 2011.

[26] S. Guo, M. Falaki, E. Oliver, S. U. Rahman, A. Seth, M. Zaharia, U. Ismail, and S. Keshav. Design and Implementation of the KioskNet System. In *Proc. ICTD*, pages 1–10, Bangalore, India, 2007.

[27] D. Johnson. Approximation Algorithms for Combinatorial Problems. In *Proc. ACM STOC*, pages 38–49, Austin, TX, USA, 1973.

[28] T. Karagiannis, J.-Y. L. Boudec, and M. Vojnović. Power Law and Exponential Decay of Inter Contact Times Between Mobile Devices. In *Proc. ACM MOBICOM*, pages 183–194, Montreal, QC, Canada, 2007.

[29] K. Lee, J. Lee, Y. Yi, I. Rhee, and S. Chong. Mobile data offloading: How much can wifi deliver? In *Proc. ACM CONEXT*, pages 26:1–26:12, Philadelphia, PA, 2010.

[30] M. Gonzalez and C. Hidalgo and A. Barabasi. Understanding Individual Human Mobility Patterns. volume 453, pages 779–782. Nature Publishing Group, Jun 2008.

[31] A. Natarajan, M. Motani, and V. Srinivasan. Understanding Urban Interactions from Bluetooth Phone Contact Traces. In *Proc. PAM*, pages 115–124, Louvain-la-Neuve, Belgium, 2007.

[32] A. J. Nicholson and B. D. Noble. Breadcrumbs: Forecasting Mobile Connectivity. In *Proc. ACM MOBICOM*, pages 46–57, San Francisco, CA, USA, 2008.

[33] J. Reich and A. Chaintreau. The Age of Impatience: Optimal Replication Schemes for Opportunistic Networks. In *Proc. ACM CONEXT*, pages 85–96, Rome, Italy, 2009.

[34] J. Robinson, M. Uysal, R. Swaminathan, and E. Knightly. Adding Capacity Points to a Wireless Mesh Network Using Local Search. In *Proc. IEEE INFOCOM*, pages 1247–1255, Phoenix, AZ, USA, 2008.

[35] P. Rodriguez, I. Pratt, J. Chesterfield, R. Chakravorty, and S. Banjeree. MAR: A Commuter Router Infrastructure for the Mobile Internet. In *Proc. ACM MOBISYS*, pages 217–230, Boston, MA, USA, 2004.

[36] I. Trestian, S. Ranjan, A. Kuzmanovi, and A. Nucci. Taming User-Generated Content in Mobile Networks via Drop Zones. In *Proc. IEEE INFOCOM*, pages 2840–2848, Shanghai, China, 2011.

[37] I. Trestian, S. Ranjan, A. Kuzmanovic, and A. Nucci. Measuring Serendipity: Connecting People, Locations and Interests in a Mobile 3G Network. In *Proc. IMC*, pages 267–279, Chicago, IL, USA, 2009.

[38] W. Wang, V. Srinivasan, and M. Motani. Adaptive Contact Probing Mechanisms for Delay Tolerant Applications. In *Proc. ACM MOBICOM*, pages 230–241, Montreal, QC, Canada, 2007.

Appendix A
Optimal algorithm

We present an Integer Linear Programming formulation to determine the optimal Drop Zone placement. We use the notation introduced in Section III-A1.

A. Decision Variables

Two types of binary variables are introduced into the formulation: x_b and δ_c^{ij}. The variables $x_b \in \{0,1\}$ describe whether a Drop Zone is placed at base-station b (i.e., $x_b = 1$) or not (i.e., $x_b = 0$). The variables $\delta_c^{ij} \in \{0,1\}$ describe whether the content chunk $c \in C$ that was generated at time $t_i \in T$ is delivered at time $t_j \in T$ with $t_i \leq t_j$ (i.e., $\delta_c^{ij} = 1$) or not ($\delta_c^{ij} = 0$).

B. Constraints

- Drop Zone Placement:

$$x_b \leq \sum_{c \in C} \sum_{i,j \in T: i \leq j} \delta_c^{ij} m_c^{jb} \quad \forall b \in B \tag{1}$$

$$x_b \geq \delta_c^{ij} m_c^{jb} \quad \forall b \in B, \forall c \in C, \forall i,j \in T : i \leq j \tag{2}$$

- Content Delivery (No Splitting):

$$\delta_c^{ij} \leq n_c^i \quad \forall c \in C, \forall i,j \in T : i \leq j \tag{3}$$

$$\sum_{j \in T: j \geq i} \delta_c^{ij} = n_c^i \quad \forall c \in C, \forall i \in T \tag{4}$$

- Drop Zone Capacity:

$$\sum_{c \in C} \sum_{i \in T: i \leq j} \delta_c^{ij} m_c^{jb} \Delta_c^i \leq \zeta_b^{\max} \quad \forall b \in B, \forall j \in T \tag{5}$$

- Maximum Delay Allowed:

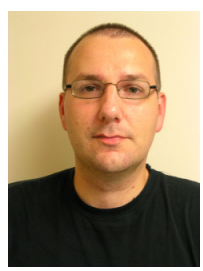
$$\delta_c^{ij} R_c^{ij} \leq D^{\max} \quad \forall c \in C, \forall i,j \in T : i \leq j \tag{6}$$

C. Objective Function

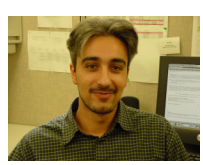
$$\min \sum_{b \in B} x_b \tag{7}$$

Ionut Trestian received his B.S. degree in Computer Science from the Technical University of Cluj-Napoca, Romania, in 2007. He is currently working toward the Ph.D degree at Northwestern University under the supervision of Aleksandar Kuzmanvoic. His research interests include network measurement, network security, overlay networks and social networks.

Supranamaya Ranjan (S'00, M'06) is a Senior Member of Technical Staff at Narus Inc. He received the M.S. and Ph.D. degrees from Rice University in 2002 and 2005 respectively. He served on the technical program committee for IEEE INFOCOM 2007 and IEEE ICDCS 2008. His research interests are in the areas of network security, anomaly detection and high-performance distributed systems.



Aleksandar Kuzmanovic is an Associate Professor in the Department of Electrical Engineering and Computer Science at Northwestern University. He received his B.S. and M.S. degrees from the University of Belgrade, Serbia, in 1996 and 1999 respectively. He received the Ph.D. degree from Rice University in 2004. His research interests are in the area of computer networking with emphasis on design, measurements, analysis, denial-of-service resiliency, and prototype implementation of protocols and algorithms for the Internet. He received the National Science Foundation CAREER Award in 2008.



Antonio Nucci is the Chief Technology Officer at Narus Inc, and received his M.S. and Ph.D. degrees in Electrical Engineering from Politecnico di Torino, Italy, in 1998 and 2003 respectively. In his career, Antonio has published more than 70 technical papers, filed 22 patent applications covering various aspects of networking and co-authored a definitive textbook on managing large IP networks, titled Design, Measurement and Management of Large-Scale IP Networks. Bridging the gap between Theory and Practice, published by Cambridge University Press. In 2007, he was awarded the prestigious InfoWorld 2007 CTO Top 25 for his vision and leadership within Narus and the IT community. His research interests include network design and measurements, traffic analysis and security.