Taming User-Generated Content in Mobile Networks via Drop Zones

Ionut Trestian
Northwestern University
Evanston, IL, USA
ionut@northwestern.edu

Supranamaya Ranjan
Narus Inc.
Sunnyvale, CA, USA
soups@narus.com

Aleksandar Kuzmanovic
Northwestern University
Evanston, IL, USA
akuzma@northwestern.edu

Antonio Nucci
Narus Inc.
Sunnyvale, CA, USA
anucci@narus.com

***Abstract*—Smartphones have changed the way people communicate. Most prominently, using commonplace mobile device features (*e.g.,* high resolution cameras), they started producing and uploading large amounts of content that increases at an exponential pace. In the absence of viable technical solutions, some cellular network providers are considering to start charging special usage fees to address the problem.**

Our contributions are twofold. First, we find that the user-generated content problem is a *user-behavioral* problem. By analyzing user mobility and data logs of close to 2 million users of a cellular network, we find that (i) users upload content from a small number of locations, typically corresponding to their home or work locations; (ii) because such locations are different for different users, we find that the problem appears ubiquitous, since user-generated content uploads grow exponentially at most locations. However, we also find that (iii) there exists a significant lag between content generation and uploading times. For example, we find that 55% of content that is uploaded via mobile phones is at least 1 day old.

Second, based on the above insights, we propose a new 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 better 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 total content via the 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 [7]. 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, videos with their friend circle on their phones by 2013 [9].

The problem incurred by the booming activity on mobile devices is that users are no longer only consuming data but have started *producing* content 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 [3], [5]. 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 [1]. 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 at popular 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 the 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 [2]. 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" [2].

Our key contribution is 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. Also, it helps providers effectively 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) where users can upload their 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 a *user behavioral problem*. We find that an individual user is likely to upload 'heavy' content from a small subset of locations, typically corresponding to his home or work locations. 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, for 55% of the content the difference is longer than a day in the Flickr case.

Our Drop Zone approach is based on (i) changing a user's upload patterns 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 locations. At the same time, providers can take advantage of users' daily commute properties to increase bandwidth at a small number of locations. We call these locations Drop Zones, and let users opportunistically upload content while in such zones. The underlying intuition, that we confirm in our analysis, is that most users visit a smaller number of *common locations* during daily commutes. Thus, by strategically upgrading small parts of their networks, providers can serve growing user-generated content with minimal resources. In our approach, a user would not necessarily have to plan moving to a Drop Zone in order to upload content. He would hand-over content to a background application that will transparently upload content at the first Drop Zone that he encounters during his regular movement.

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 infrastructure placement problem? What is the relationship between postponed content delivery intervals users can tolerate and needed infrastructure? What is the limiting behavior of this approach as content keeps increasing?

Our analysis shows that by upgrading only 1,303 base-stations of the current nationwide infrastructure and assuming users would postpone content delivery by 1 day, 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.

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. We present related work in Section VI and conclude in Section VII.

II. The Case for Drop Zones

Here, we briefly introduce the Drop Zone architecture. Then, we show empirical results that motivate our approach. In particular, we demonstrate that (i) users tend to upload content from 'comfort zone' locations, (ii) this makes the user-generated problem widespread, and (iii) the time lag between content generation and upload can be significant for large content fractions.

A. A Drop Zone Architecture

Figure 1 shows our 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 Drop Zones, shown with a darker color in Figure 1(b). We 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 [10] or LTE [6], for which base-station ranges can be roughly matched among 3G, 2.5G and WiMAX and LTE. 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.

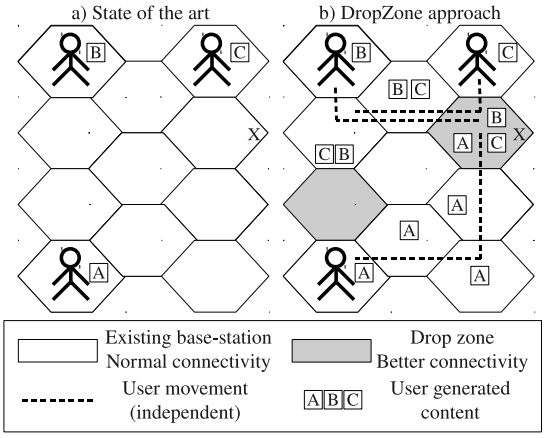


Fig. 1. Postponed delivery example

Figure 1 illustrates the difference between state-of-the-art user uploading and our Drop Zone approach. Take 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 certain locations. We call such user- affine locations as the user's 'comfort zones', that most of the time correspond to the user's home or work 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. We demonstrate that this is indeed the case in Section II-B2 below.

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 three pieces of content, A, B, and C, are uploaded from the same Drop Zone marked by X in the figure. As we show in Section II-B3, users even now upload content in a postponed manner. 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 across 64,670 base stations during a seven day period. It preserves user privacy as all identifiers such as users' phone numbers, email addresses and ip-addresses were anonymized. More details about the dataset can be found in Appendix A.

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

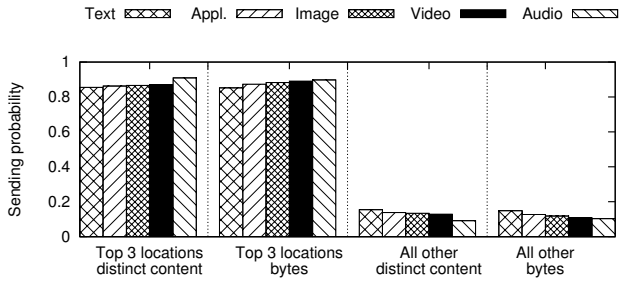


Fig. 2. Sending probability depending on location rank

The trace contains MMS messages exchanged among users, as well as uploaded to social networking websites such as Facebook, Myspace, Flickr, [1] *etc*. Table I summarizes uploading statistics. We use various attachment types to categorize content in one of the five categories: text (plain, xml, *etc*), application (word, excel, pdf, *etc*), image (gif, jpg, jpeg, *etc*), audio (mp3, acc, midi, *etc*), and video (3gpp, h264, mp4, *etc*).

The trace provides the location of a user in terms of the base-station. The area serviced by a base-station in this network varies from hundreds of square meters (in densely populated areas) to several square miles (in sparsely populated areas). In the remainder of the paper, we use the term location to refer to the area serviced by a specific base-station. Thus, while our trace does not provide finer-grained location information, it serves our capacity provisioning and infrastructure placement needs perfectly.

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. Analyzing deeper these results, using straightforward time and space analysis (details omitted for space constraints - identifying locations where a user spends most time during day hours and night hours), 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 locations. Thus, users prefer to send their content, including the 'heavy' ones that we focus on in this paper, from their top ranked locations.

[1]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.

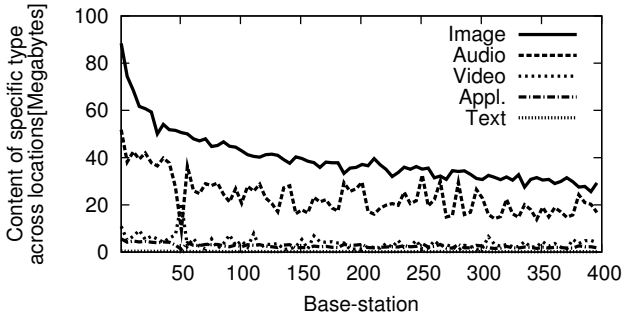


Fig. 3. Sent content type breakdown across base-stations

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 widespread, 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 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 content problem is widespread 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 showing that not all user-generated content is posted or sent after it has been produced. In particular, we have crawled Flickr mobile photography groups where users upload pictures taken via their camera phones [8]. 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. We obtain the 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). For this part we have the results at the granularity of hours.

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

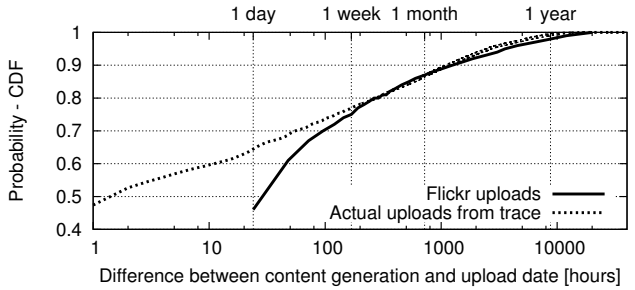


Fig. 4. Delay between shooting and uploading pictures using mobile phones

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. While we cannot make strong statements for content that is uploaded after generated, we argue that a portion of this content might be possible to deliver in a postponed manner. This is because some 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 validates our assumption that some user-generated content can be uploaded in such way.

Other incentives for users to upload or download content in a postponed manner include: (i) *longer battery life* - it has been shown in [13] 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) Summary: We have demonstrated that users tend to upload content from a subset of top locations, unique to each user. Because such locations are widely dispersed for different users, the observed load increase incurred by such uploads is widespread. At the same time, we have demonstrated that large portions of user-generated content is uploaded in a postponed manner, but still from top locations. Putting all the pieces together, we argue that a Drop Zone approach can help both users and providers. Users can mark content for postponed upload and do not worry about it. Providers can strategically place upgraded technology at a small number of locations that can absorb large portions of heavy content. Below, we develop infrastructure placement algorithms, to determine where to place better connectivity infrastructure.

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.

[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.

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 delay.

1) Inputs: For the Drop Zone placement problem, we use a one week long trace from one of the largest cellular providers in the US. The trace provides information about users' trajectories in terms of what locations (base-stations) they were present at, and at what time. It also provides information about uploaded content. First, we define a single indivisible unit of content as *content chunk* of maximum λ bits. Hence, each content could consist of several chunks. 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 in to discrete units of length τ seconds each, such that the entire trace spans over the set of bins: $T = \{t_1, t_2, ..., t_T\}$. 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 $T = \{t_1, t_2, ..., t_T\}$ 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 $t_i \leq t_j$. 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 $D^{\max}$ 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., $m_c^{jb} = 0$). This notation is also used in Appendix B.

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 [15]. 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 18. It has been shown [18], 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.

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.

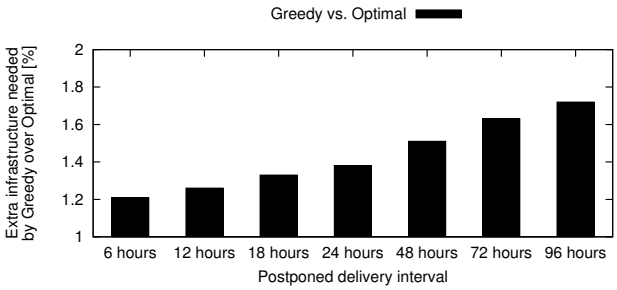


Fig. 5. Greedy placement versus Optimal placement obtained from the ILP.

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.* $t_j \in [t_i, t_i + D^{\max}]$. It sorts these pairs in ascending order, with the most critical as the first 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.

C. Parameters

In the next section, we evaluate the performance of Greedy and Optimal algorithms (described in Appendix B). 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, $\forall b \in B$ [6]. 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 it is the biggest content piece in our dataset and can fit in one minute considering the LTE technology.

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 B. We solve the ILP by using the ILOG CPLEX software [4]. Because of the large scale of the data involved, we compare the optimal placement given by the ILP 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 5 shows the results. We vary the maximum

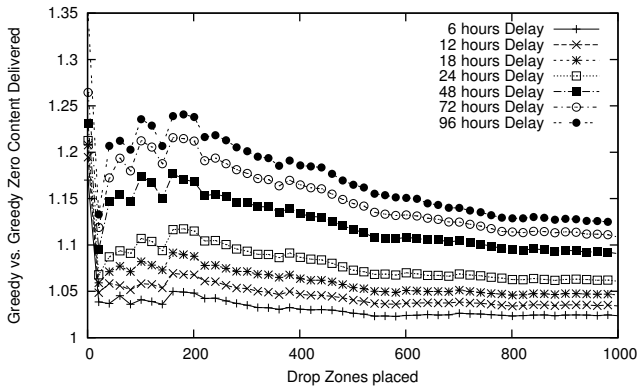


Fig. 6. Greedy Drop Zone placement compared to Drop Zone placement based on popular locations (ranked by content delivered).

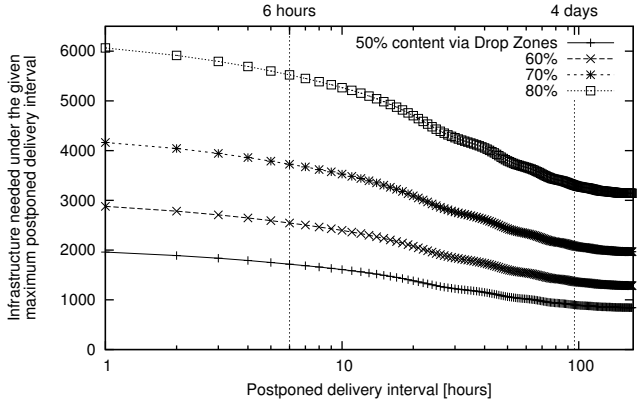


Fig. 7. Number of locations where Drop Zones are installed.

postponed delivery interval among the values of 6, 12, 18, 24, 48, 72, and 96 hours.

The insights from Figure 5 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 6 shows the results. The x-axis shows the number of Drop Zones, while the y-axis shows the ratio of 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 Greedy Zero when 200 Drop Zones are used in both cases and with a maximum postponed delivery interval of 96 hours. 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.

Figure 6 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 select locations that were not so popular originally, yet they are good Drop Zone locations when postponed delivery is considered during the selection process. Because Greedy Zero has no knowledge about user mobility, it either neglects such locations or selects them much later.

The gaps shown in the figure translate to additional infrastructure the order of a few hundred additional Drop Zones needed to deliver the amount of content. In particular, for 1,000 Drop Zones placed by Greedy with 96 hours postponed delivery,Greedy Zero needs 1,201 Drop Zones (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.

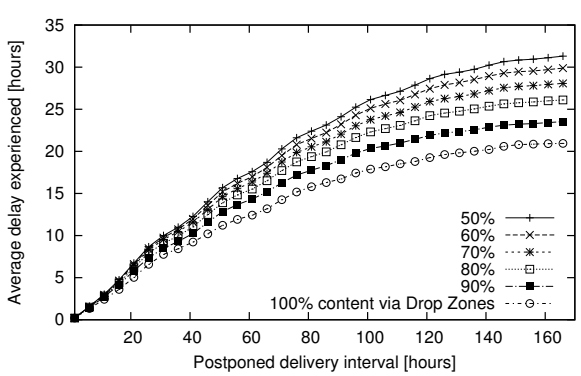
C. Infrastructural Needs

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.

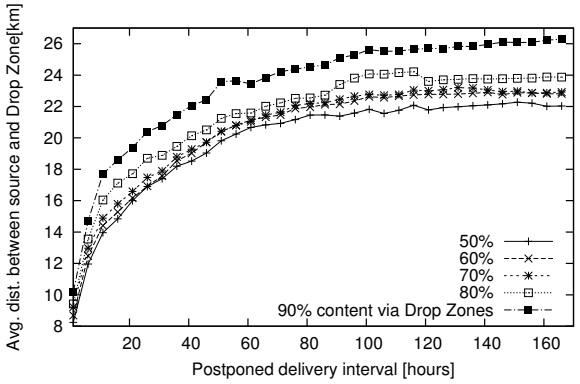
Figure 7 shows the results. It depicts the number of Drop Zones (y-axis) needed to serve the given percent of content by assuming the maximum postponed delivery interval (x-axis) varied in the range from 1 to 168 hours. Necessarily, Drop Zone architectures that target to absorb larger amounts of traffic 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 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.

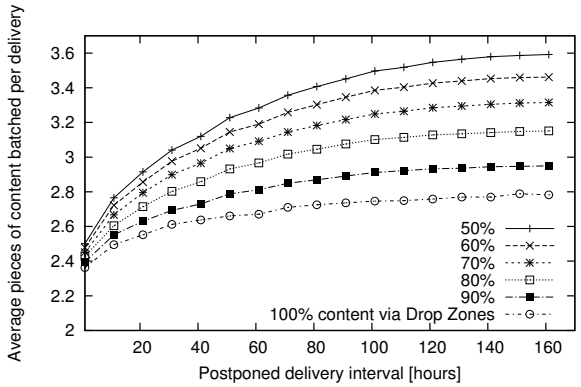
Figure 7 shows that all curves 'flatten' as the postponed delivery interval increases over 4 days. One would expect that as the postponed delivery interval increases, users see more locations, and hence, almost infinite gains can be obtained from user mobility. However, previous studies on human mobility, reported on the high predictability of human movement and observed that users spend significant time in just a few locations [19]. This effect can be observed in Figure 7. After a time interval of approximately 4 days, the curves level off and only marginal gains can be obtained. Our explanation is that since users spend time in just a few locations, benefits in



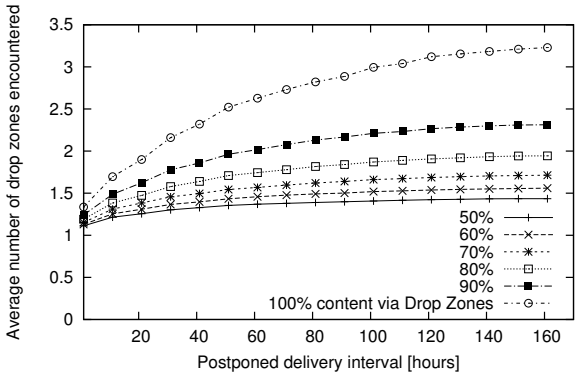
(a) Average delay experienced by content.



(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.

Fig. 8. Infrastructural usage

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 take a user view and analyze how users interact with the Drop Zone architecture. We explore the following: (i) average content delay: even though we specify a maximum postponed delivery interval, content could be delivered much earlier, (ii) average distance between source and Drop Zone, (iii) average pieces of content batched: since users postpone delivery, they have a larger amount of content when encountering a Drop Zone, and (iv) average number of drop zones encountered during the trace interval. In all scenarios, we take the percent of delivered content as a parameter. For delivery, we have users opportunistically deliver their postponed content upon encountering the first Drop Zone with available capacity to deliver the content.

Figure 8(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 all cases, users experience on average four times less delay than given by the maximum postponed delivery interval.

Figure 8(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. Given that the more delivered content necessarily correspond to 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: with 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 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 8(c) shows the average number of content pieces batched per delivery. As mentioned above, batching delivery is beneficial for a mobile device as it improves battery life [13]. 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.

Figure 8(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.

E. What-If Scenarios

Here, we study implications derived from the research presented in this paper. In particular, we address the following 3 problems: (i) how would our architecture deal with an exponential increase in content size in the future, ,and (ii) what are the number of missed upload opportunities for the content pieces for which we have determined the actual creation date? In all the cases below, we analyze the impact of a Drop Zone architecture placed to 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.

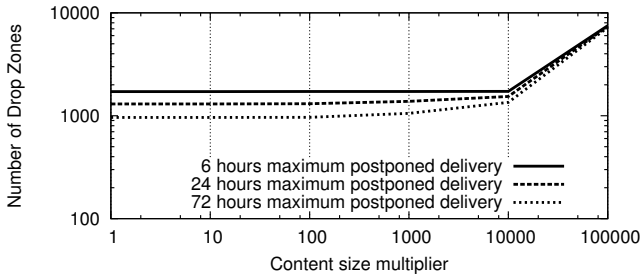


Fig. 9. Increase in infrastructure due to increasing the size of content.

1) Content size increase: Here, we try to understand how the proposed architecture would deal with an increase in the content size in the future. Figure 9 shows the results we 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. An increase beyond a 10,000 multiplier would require a deployment of a significantly larger number of Drop Zones, as shown in Figure 9 for x=100,000, or an increase in capacity for existing Drop Zones.

2) Missed connections: Here, we focus on a subset of users from our trace that produce and upload content for which we know the creation date, *i.e.*, photos, in a postponed manner, as explained in detail in Section II-B3 above. In particular, we try to quantify missed upload opportunities. 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 10 shows the CDF of upload opportunities for considered content. Our figure shows that close to 50% (more in some cases) of content can be uploaded through Drop Zones. This is consistent with the infrastructure dimensioning that we have performed.

V. Discussion

Advanced Content Drop-off. In our evaluation we used 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

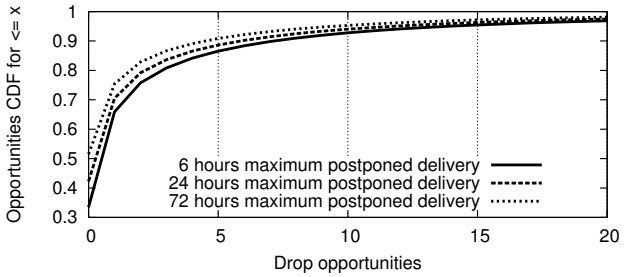


Fig. 10. Missed opportunities for content with known creation date.

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. We consider this as a challenging problem out of scope of our current work, but certainly the most interesting problem we wish to study next.

Generality. Our proposed Drop Zone architecture is generic and refers to increased bandwidth at a base-station. In this regards, our placement algorithms can be used to determine the base station locations at which to increase capacity (WiMax or LTE) first rather than everywhere at the same time. 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.

VI. Related Work

When addressing increased load in cellular networks, most research *e.g.*, [11], [12], [22] assumes the existence of a diversity of networks in some locations and introduce systems for exploiting [11], [22] or predicting locations that have such diversity [12]. On the contrary, our goal is to determine where placing such new technology is meaningful.

Delay-tolerant networking has been widely studied in the recent years [16]. Most research on delay tolerance for human-carried devices considers encounters between different users (*e.g.*, [17], [20], [21], [24]). Recent work in the area examines how human mobility influences the design of different forwarding algorithms [14], or how performing delay tolerant transfers reduces energy consumption of a mobile phone [13]. While we also take delay tolerance as a block of our approach, our key goal is to study how human mobility might influence large scale infrastructural placement. As we have network-wide views of human mobility and mobile transfers, we can design placement strategies helpful for operators, and effectively evaluate the performance of such approaches.

VII. 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 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 technology or a much wider Drop Zone deployment must be applied. 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: Internet to Hit Full Capacity by 2010. http://www.zdnet.com/news/at-t-internet-to-hit-full-capacity-by-2010/197822.

[2] 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.

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

[4] ILOG CPLEX. http://www.ilog.com/products/cplex/.

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

[6] LTE. http://www.3gpp.org/LTE.

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

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

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

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

[11] G. Ananthanarayanan, V. N. Padmanabhan, L. Ravindranath, and C. A. Thekkath. Combine: Leveraging the Power of Wireless Peers Through Collaborative Downloading. In *MobiSys '07*.

[12] G. Ananthanarayanan and I. Stoica. Blue-Fi:Enhancing Wi-Fi Performance using Bluetooth Signals. In *MobiSys'09*.

[13] N. Balasubramanian, A. Balasubramanian, and A. Venkataramani. Energy Consumption in Mobile Phones: A Measurement Study and Implications for Network Applications. In *IMC '09*.

[14] A. Chaintreau, P. Hui, C. Diot, R. Gass, and J. Scott. Impact of Human Mobility on Opportunistic Forwarding Algorithms. *IEEE Transactions on Mobile Computing '07*.

[15] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. Introduction to Algorithms.

[16] K. Fall. A Delay-Tolerant Network Architecture for Challenged Internets. In *SIGCOMM '03*.

[17] 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 *ICITD '07*.

[18] L. Lovasz. On the Ratio of Optimal Integral and Fractional Covers. In *Discrete Mathematics, 1975*.

[19] M. Gonzalez and C. Hidalgo and A. Barabasi. Understanding Individual Human Mobility Patterns. *Nature*, 453(7196):779–782, June 2008.

[20] A. Natarajan, M. Motani, and V. Srinivasan. Understanding Urban Interactions from Bluetooth Phone Contact Traces. In *PAM '07*.

[21] J. Reich and A. Chaintreau. The Age of Impatience: Optimal Replication Schemes for Opportunistic Networks. In *CoNEXT '09*.

[22] P. Rodriguez, I. Pratt, J. Chesterfield, R. Chakravorty, and S. Banjeree. MAR: A Commuter Router Infrastructure for the Mobile Internet. In *MobiSys '04*.

[23] I. Trestian, S. Ranjan, A. Kuzmanovic, and A. Nucci. Measuring serendipity: connecting people, locations and interests in a mobile 3g network. In *IMC '09*.

[24] W. Wang, V. Srinivasan, and M. Motani. Adaptive Contact Probing Mechanisms for Delay Tolerant Applications. In *MobiCom '07*.

APPENDIX A
DATASET DESCRIPTION

The dataset used in this paper was collected for one entire week from the content billing system of a nation-wide cellular provider from the United States. 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. We also store the event type (Start, Stop, Update) [23].

With regards to content, our trace consists of MMS (Multimedia Messaging Service) messages (that carry rich content such as photos, audio or video in addition to plain text) uploaded (to friends or popular websites such as Facebook), or downloaded by users using their phones. In particular, 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.

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.

APPENDIX B
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

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