DataSpotting: Exploiting Naturally Clustered Mobile Devices to Offload Cellular Traffic

Xuan Bao
Duke University
xuan.bao@duke.edu

Yin Lin
Duke University
yin.lin@duke.edu

Uichin Lee
KAIST
uclee@kaist.ac.kr

Ivica Rimac
Alcatel-Lucent
ivica.rimac@alcatel-lucent.com

Romit Roy Choudhury
Duke University
romit.rc@duke.edu

Abstract—**The proliferation of pictures and videos in the Internet is imposing heavy demands on mobile data networks. Though emerging wireless technologies will provide more bandwidth, the increase in demand will easily consume the additional capacity. To alleviate this problem, we explore the possibility of serving user requests from other mobile devices located geographically close to the user. For instance, when Alice reaches areas with high device density – *Data Spots* – the cellular operator learns Alice's content request, and guides her device to nearby devices that have the requested content. Importantly, communication between the nearby devices can be mediated by servers, avoiding many of the known problems of pure ad hoc communication. This paper argues this viability through systematic prototyping, measurements, and measurement-driven analysis.** [1]

I. Introduction

Mobile broadband usage has been increasing at an alarming rate due to an ever increasing popularity of mobile devices (e.g., smartphones, netbooks, iPad, Kindle) and Web 2.0 services (e.g., Pandora, YouTube). While on the move, mobile users listen to music, browse pictures, and watch videos. Such continued rise in demand is predicted to overload cellular networks.

Cellular operators are currently relying on expensive solutions to mitigate the problem. Examples include additional hardware/spectrum deployment, dispatching "Cell Towers on Wheels" to overloaded places, or upgrading access networks to 4G/LTE. Also, some researchers have been viewing WiFi hotspots [1] as a means to offload data from their 2G/3G infrastructure. However, under the flat rate billing policy, there is no clear justification on revenue growth for such huge capital expenditure. Given this, major operators in the US introduced new pricing mechanisms that bind price with data usage, hoping to ease the congestion in their wireless access network.

So far P2P solutions failed to receive hospitable treatments for several reasons. (1) Mobile devices need their WiFi/Bluetooth interfaces to continuously scan for peers and exchange content availability – this prohibitively reduces the phone's battery life. (2) Due to limited storage and lack of YouTube-like popular services in the past, the opportunities for serving requests from peer devices were limited. (3) Finally, concurrent ad-hoc connections may interfere with each other and reduce the overall throughput.

Today, trends are probably changing in favor of P2P communication. A small number of web sites and services are becoming extremely popular, drawing many users to access content from them (e.g., CNN, YouTube, ESPN, Hulu). Alongside this popularity trend, limited screen size on mobile devices permits users to access only a limited number of pages per site – this further skews the content popularity distribution by only permitting some pages to become very popular while other pages remain much less noticed [2]. Moreover, IEEE 802.15 TG8 (Peer Aware Communications) was formed this year to provide a global standard for scalable, low power, and highly reliable wireless P2P communications for emerging mobile services [3]. Further, state-of-art localization techniques for smartphones makes location-tracking at finer grade feasible and power efficient. These factors together present a greater potential for exploiting P2P data exchanges to offload cellular traffic, particularly during peak hours.

The goal of this paper is to explore a practical way of offloading cellular traffic via device-to-device (D2D) content transfer. We exploit the observation that cellular networks are most strained during high-density events (sports stadiums, concerts, train stations, rush hours, etc.), when many people located in a small area request for content. Importantly, these are also the scenarios where the opportunity of P2P transfers is maximized. Thus, the cellular operators can track the location of different phones and builds a map that indicates where the phone clusters are reasonably dense – we call these places *data spots*. These data spot maps are periodically pushed to the phones.

When a mobile phone enters a data spot, it alerts the cellular operator of its content request. Given the operator can maintain a digest of available content in different devices in that area (including WiFi access points), it now performs the match-making service. The cellular operator notifies the requesting phone to directly connect to the appropriate phone through WiFi adhoc mode and retrieve the desired content. In this process, 3G connection is used as a control channel to wake up WiFi interface only when the transfer is taking place, thus the energy cost is minimized. This form of centralized mediation also obviates the need for a phone to constantly scan for other devices in the vicinity, making it an energy-effective solution. Moreover, the knowledge of cached content in each phone does not spread among peer devices, avoiding privacy concerns. Finally, cell-tower mediated transfers are also amenable to accounting – a mobile device that has served content may be appropriately rewarded at the end of the month.

Example Applications

Dataspotting can be an enabler for a variety of applications, especially those heavy on bandwidth consumption. We present two examples here and discuss more later.

Subscription Based Services: Many popular applications today provide downloading/streaming services to subscribed users. For example, Pandora radio provides randomized music playing based on clients' taste. These kinds of services are certainly amenable to Dataspotting.

[1]The research leading to these results has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement n318398.

978-1-4673-5946-7/13/$31.00 ©2013 IEEE

The cellular provider can monitor music files in nearby mobile caches to deliver a music service directly from neighboring phones. Similar examples are TV catch up services, news feeds, etc.

Special Events Broadcast (e.g., World Cup, NCAA): Special events, such as sports games, usually cause a peak in demand for related content. For example, during the soccer world cup, soccer related videos become highly popular. Moreover, the popularity for some content may also be region specific, e.g., German games may be more popular among German communities. Dataspotting naturally leverages these opportunities due to its inherent spatio-temporal content sharing model.

Our overall contributions in this paper may be summarized as follows: (1)Combining advanced localization techniques, we identify the opportunity to exploit spatially clustered devices for offloading cellular traffic. (2)We propose using 3G connection as a control channel for operator mediated device-to-device connection. (3)We perform real life measurements, along with basic theoretical analysis and simulations, to understand the viability of the system in large-scale deployment. (4)We prototype the Dataspotting system on the Android Nexus One platform to understand system-level/practical challenges.

The paper is organized as follows: Section II introduces the overview of the system; Section III provides measurement, theoretical analysis and simulation results regarding the potential benefit of a large-scale deployment; Section IV explains the implementation and provides a system level evaluation. Section V discusses related and ongoing work.

II. System Overview

During peak hours, operators activate the *DataSpotting* service as in Figure 1. Mobile clients are instructed to report their locations to the operator periodically (e.g., every ten minutes or when the user is estimated to have moved a threshold distance). The employed localization method is similar to CompAcc [4], a sensor assisted localization method with low energy budget.

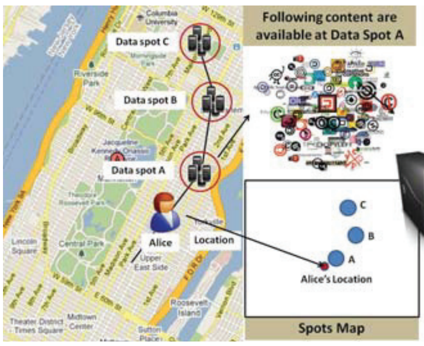

Fig. 1. DataSpotting overview: Black lines represent Alice walking through data spots. The DataSpotting server maintains a data spot map and creates content profiles for each data spot.

The server uses the location information from a large number of clients to estimate locations/radii of the data spots (where the clients are clustered). At the same time, the server also tracks what content are available at each data spot and establishes content profiles. In Figure 1, three data spots are detected and recorded in a data spot map. The server can also use historical data to facilitate the mapping process. An incoming user $Alice$ reports her location and receives the related part of this spot map. Therefore, she even has the choice to intentionally approach a data spot [5]. When $Alice$ enters a nearby data spot A, the server examines the content profile of data spot A and determines whether any nearby mobile client Bob in data spot A has the content of interest. If so, the server instructs $Alice$ and Bob to wake up their WiFi interfaces to ad-hoc mode and assign temporary IP addresses. Then, $Alice$ can retrieve the content from that client via WiFi.

III. Measurement Driven Simulation

This section is focused on understanding the viability and rough performance gains expected with data spots. The main question we ask is: *how likely is it to find a requested content in one of the nearby devices, i.e., the cache hit rate*. We break down the question into three parts: **(1) Measurement:** Measuring distribution of data spots in real life. **(2) Analysis:** Analyzing theoretical hit rate inside one data spot. **(3) Simulation:** Simulating system performance combining the measurement data with human mobility model and system parameters.

The measurement underpins the feasibility of the system by understanding user densities in urban areas (i.e., occurrence of data spots), as well as contact durations between devices. The analysis uses content popularity distributions, varying node densities, and different cache sizes, to characterize hit rate inside one data spot. The simulation synthesizes measured parameters with a human mobility models (for Manhattan) to compute realistic hit rates and benefit from network offloads. We begin discussion with the real-life measurement.

A. Measurement - Data Spot in Manhattan

The first question to ask for DataSpotting is *"Are there many data spots in the real life?"*. Therefore, we conducted a real-life measurement in Manhattan area in order to understand users' location distribution and the typical contact duration – the time that two devices stay within communication range – for pedestrians. The results are used for realistic simulations as input parameters.

Measurement Setting: The measurement is conducted by wardriving (through biking) in Manhattan area with Bluetooth scanning and GPS logging. Since it is not possible to detect WiFi availability in mobile devices, our detection was limited to Bluetooth devices in the discoverable mode. Considering these factors and the fact that Bluetooth range is significantly shorter than WiFi, the user density statistic here is very conservative. Yet the measured result still shows enough density for our purpose. In our calculation, dumb devices such as headphones, printers, etc. are filtered out from the final list thus the statistics only include phones and laptops which can be used in the DataSpotting system. Moreover, after identifying data spots, we select a few representative data spots and analyze the characteristics of data spots (e.g., contact duration among mobile users). Figure 2 shows the area covered in the wardriving process. All north-south oriented streets were covered by biking. A static analysis for different locations at NY Penn-Station was also conducted.

Biking Scenario: The biking scenario captures the typical situations that a pedestrian may encounter. The metrics in interest are: (1) Devices around the user at any time. This determines how often the user may pass data spot

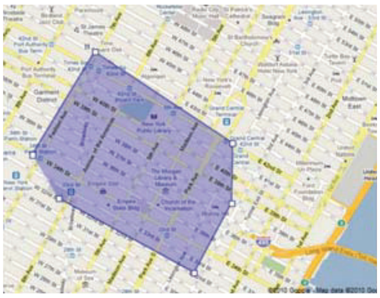

Fig. 2. Area Covered In Manhattan Wardriving

if we define data spot as crowded areas with certain user density threshold. (2) Contact duration. This determines how much time a pair of devices usually stay within Bluetooth communication range.

Figure 3(a) shows devices around the user. The shown time steps every 10 seconds, the number of devices shows the laptops and mobile phones detected during the 10 seconds. Though the number is changing all the time, most of the time there are more than 30 devices around the user. The variation in the number of nearby devices also shows the need for localization. Derived from the data, the average user density in this region of Manhattan area to be $2.9124 * 10^3/km^2$. This parameter is used in our simulation later.

Though there are many devices around the user at all time, these devices also need to stay within range to facilitate communication. Figure 3(b) shows the histogram of contact duration. Many devices actually stay for 30 seconds. Based on our implementation, the measured data rate is around 1 MB/s, so 30 seconds is more than enough to transfer a medium sized file.

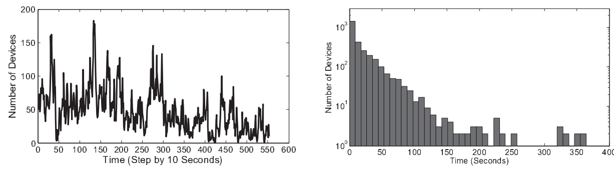

Fig. 3. (a)Devices Around the User/(b)Histogram of Contact Duration

NY Penn Station: To understand how user density changes inside a data spot, we choose NY Penn Station as an example. The measurement was conducted at two different waiting areas at 4:30PM on a Thursday. Figure 4(a) shows there are always many devices around the user and the number of devices is quite stable for different locations. Measurement at the other waiting area also shows similar behaviors. Figure 4(b) shows the histogram of contact duration. The average contact duration is 80 seconds, more than enough to transfer multiple files.

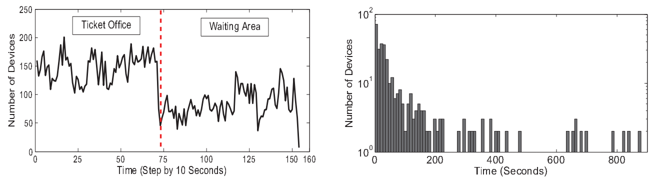

Fig. 4. (a)Devices Around the User at NY Penn Station/(b)Histogram of Contact Duration at Penn Station

This measurement shows that, even the active users constitute only a small portion of the whole population [6], the user density in populous cities like New York is more than enough to support massive D2D data transfer at many different Data Spots.

B. Theoretical Analysis – Inside One Data Spot

Now we have an estimate of how data spots are distributed. The next question is *"What hit rate can we expect inside one data spot?"*. To obtain a sense of the hit rate, we assume that all devices in a data spot are in communication range of each other. We populate the cache of each device by drawing from a universal content set – content is drawn as per the standard content popularity distribution (*Zipf*) [7], [8]. Given the contact time presented in the previous section, a medium-sized file of 10MB can easily be successfully transferred during one contact. Therefore, we consider files as the smallest units and consider file blocks as an optimization in the future. We consider two different types of services here.

(1) *Video on Demand (e.g., YouTube)*: The first analysis is for popular video websites like YouTube, where the interests towards different content items are diverse. According to previous research [7], [8], the content popularity distribution generally follows Zipf distribution with parameter ranging from 0.5 to 0.7[2] In our simulation, we assume a Zipf distribution with parameter $\alpha = 0.66$, a library set $U = 10,000$, and $N = 200$ content requests generated by the user. Figure 5(a) shows the expected hit rate for varying number of files per device cache, and increasing users in the data spot. Evidently, the hit rate is above 22% with less than 20 devices within the data spot each holding 50 files. More importantly, the hit rate increases sharply with more number of devices, validating that the core notion that nearby devices may together be a reasonably good cache for serving content requests.

(2) *Subscription Based Services (e.g. personalized music radio, TV catch up service):* Many popular applications today provide downloading/streaming services to subscribed users. Previous research shows these content exhibit more skewed popularity distribution and limited number of active items. We use a content popularity distribution extracted from real life services – this suggests a Zipf distribution with parameter $\alpha = 1.66$ Figure 5(b) shows the expected hit rate with different cache sizes and node density. Naturally, the cache hit rates are greater in these cases, since the popularity distributions are more skewed towards popular items. A user in such scenarios may expect a hit rate of around 71% with less than 50 users within her WiFi vicinity each holding 50 files.

Insights: We draw two insights from the simple calculations above. (1) Increasing the number of cached files per device is more effective (for augmenting hit rate) than increasing device density, especially when the content popularity distribution is significantly skewed. (2) Subscription based services offer a promising application space for DataSpotting, given their overly skewed content popularity distributions.

C. Simulation - Effect of Mobility and System Parameters

The measurement confirmed there are many data spot in metropolitan areas like Manhattan, indicating promise for the system. The theoretical study in the beginning of this section provided initial insights into retrieval opportunities when a user enters a data spot. However,

[2]The tail of the actual distribution is actually more popular than Zipf. We still use the Zipf distribution to show an approximation of reality.

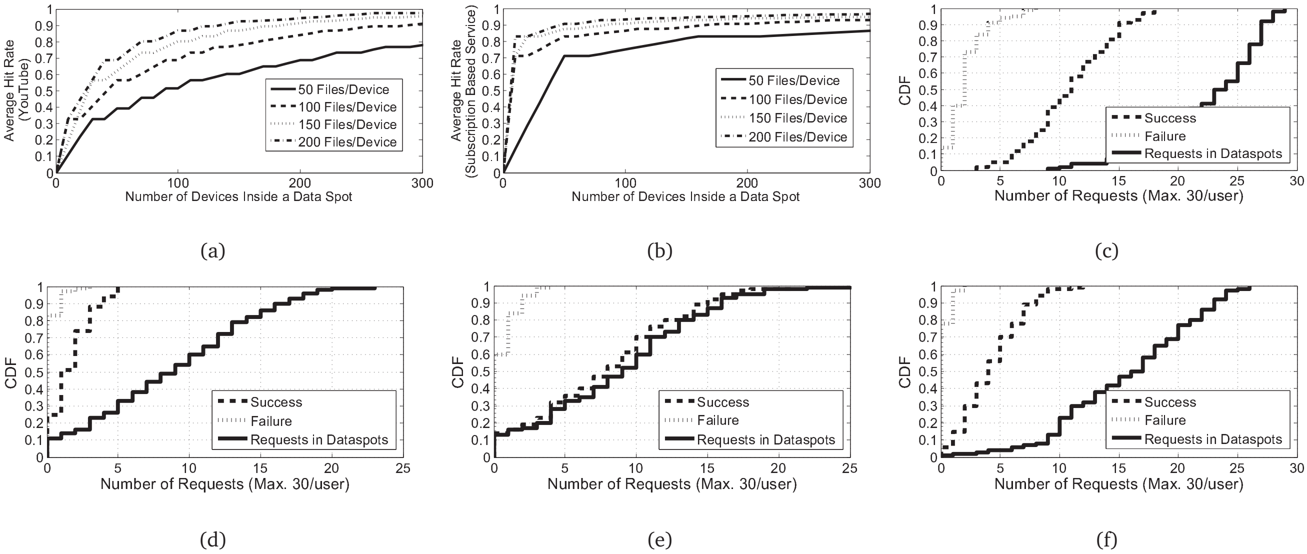

Fig. 5. Expected hit tate vs. cache space and user density inside a data spot: (a)Video on demand (b)Subscription based service; Overall performance: (c)Crowded areas (Manhattan) (d)Less crowded areas (1/10 of Manhattan) (e)Subscription based services (f)Increased cache storage

the simulation of overall system performance depends on other factors such as users' mobility patterns, localization accuracy and cache sizes. The simulation structure is shown in Figure 6.

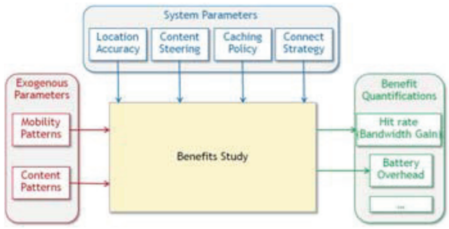

Fig. 6. Simulation overview

Simulated users walk on a generated map using a realistic human mobility model [9]. Now, as the mobile users request different contents, the server tracks the users' location and checks if requested content is cached in any of the devices in her vicinity. Of course, localization error may prevent a content exchange, even though the server prescribes one. This results in a failure. At the end of the simulation, hit rate and failures (representing energy overhead) are computed based on the generated traces. Different user densities and preloading schemes (for subscription based service) are simulated. Table III-C summarizes the parameters of the simulation. We discuss the performance next, using measured parameters from crowded and sparse areas.

Table 1: Parameter	Value	Explanation
Content Library Size	10000 files	No. of Active Items
Cache Per Device	50 files	
Popularity Ranking	Zipf(0.66, 1.66)	
Request Rate	6 per hour	5 Hours in Total
Localization Error	10 m to 150 m	Reset by GPS
Node Density	$3000(300)/km^2$	
No. of Mobility Pattern	100 per setting	

1) Crowded Area (Manhattan): Figure 5(c) shows the performance of DataSpotting in densely populated areas for YouTube-like video services. The X axis shows the number of content requests made by a new mobile user. To make the simulation realistic, observe that not all requests are made inside data spots – the solid black curve shows the number of requests that were actually from within some data spot. Of these some of the requests are successfully served by peer caches, while other requests fail due to localization errors. The dotted curves (black and gray) show these results. Of course, the sum of succeeded and failed requests do not add up to the total number of requests within a data spot; some requested files are not in any of the peer caches, and the server naturally does not prescribe a P2P exchange. Under these scenarios, Figure 5(c) shows the average cache hit rate is 36.27%, implying that more than a third of the content requests can be served by the DataSpotting.

2) Less Crowded Area (one tenth user density): Not all places are as crowded as Manhattan. Figure 5(d) shows the performance of DataSpotting at less crowded areas where the device density is only $\frac{1}{10}$ of the Manhattan area. The average cache hit rate is still 9.57%. Moreover, a number of optimizations are possible to cope with this decreased device density. First, we can provide subscription based services. Figure 5(e) shows the performance of subscription based services (Zipf 1.66) in these less populated areas – the hit rate increases to 30.27%. Further, even for general services, increasing the caching storage per device makes DataSpotting more efficient. Figure 5(f) shows that under such change, the hit rate can be increased to 14.28%.

D. Summary of Simulation Results

Table III-D summarizes the overall performance with data spotting. The reason that such high hit rate can be achieved is that video on demand or subscription based services tend to have limited number of active items. In our simulation, we used 10000 as the size of this active item set which can be observed from real life data. The main findings suggest that in dense data spots, one third of the content requests may be served without much difficulty. However, as the user density decreases, the hit

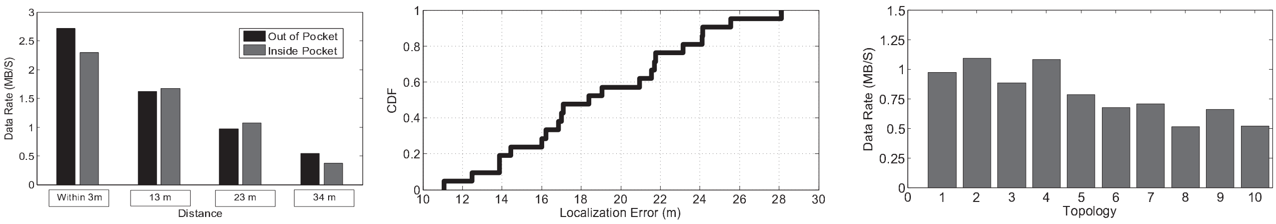

Fig. 7. (a)Data Rate Supported by DataSpotting; (b)Inertia Localization Error; (c)Throughput of User Moving Through Data Spots

rate may degrade. Nevertheless, at the expense of some increase in caching storage, this degradation can be partly compensated. Finally, current localization error does not impact the performance excessively. These assumptions are also verified through prototype implementation.

Table2: Scenario	Hit Rate	Failure Rate
Crowded Areas	36.27%	3.5%
Less Crowded Areas	9.57%	1.13%
Less Crowded Areas (Subscription)	30.27%	1.85%
Less Crowded Areas (More Storage)	14.28%	1.19%

IV. Prototype Implementation

To evaluate the performance of our system in the real world, we implemented a prototype on Google's Nexus One devices. In our experiment, we first evaluate the performance of individual building blocks and then test the overall performance of the DataSpotting system in a small scale deployment.

A. Device-to-Device Transmission Performance

To understand the capability, we first measure the device-to-device communication properties. Figure 7(a) shows the average data rates with different distances at outdoor location. Since users may put devices inside their pockets, our experiment covers this scenario. The data rates do not change significantly with different settings. In an indoor environment, the data rate depends more with conditions such as having a line of sight and interference. The observed data rate ranges between 300KB/S to 2.5MB/S, adequate for transferring files around 10MB during contact time.

B. Localization Accuracy

Figure 7(b) shows the localization error of an inertia localization scheme using step count and compass similar to CompAcc [4]. The error accumulation rate is around 10 meters per minute as in our simulation and is mostly below the communication range. This rate confirms our simulation input and validate the simulated failure rate.

C. Overall Performance

In these experiments, we install DataSpotting prototype on five devices. The experiments take place at different indoor/outdoor locations and multiple topologies are adopted to emulate possible settings of DataSpotting. Figure 8 shows the typical indoor/outdoor scenarios with different topologies and walking paths that we used and Figure 7(c) shows the throughput of DataSpotting when the user is moving around a data spot. This experiment verifies that medium sized video files, around 10MB, can be efficiently offloaded through DataSpotting under walking/running speed.

D. Power Consumption

Our measurement shows that our optimized localization scheme can save energy by enabling WiFi (by sending control message through 3G connection) only at the data spot compared to a periodical WiFi scanning scheme. The average power consumption of localization is 150mW as opposed to 279mW for constantly listening on WiFi.

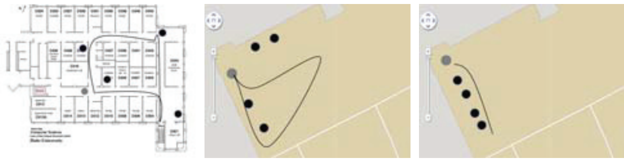

Fig. 8. Examples of Topologies and Walking Traces

V. Related Work

Previous research in related areas has greatly inspired the idea of DataSpotting. For examples, several papers have investigated the performance of offloading 3G through WiFi access point in metropolitan areas [1], [10]. Besides offloading, Combine [11] paper attempts to augment weak wireless connection through collaborative downloading with nearby peers. Several papers [12], [13] propose techniques to share video stream among users within proximity of each other.

VI. Conclusion

We propose DataSpotting, a system that explores the feasibility of offloading cellular traffic by combining localization and P2P content transfer. The key intuition is to leverage the observation that cellular networks are typically overloaded in crowded areas, and the clustered devices there can be exploited for P2P content sharing. Finally, keeping the operator in the loop facilitates the enforcement of incentive mechanisms. In light of these, we believe that DataSpotting is an early but promising research direction towards coping with the bandwidth crisis looming on 3G/4G wireless broadband operators.

References

[1] A. Balasubramanian, R. Mahajan, and A. Venkataramani, "Augmenting Mobile 3G Using WiFi," in *ACM Mobisys*, 2010.
[2] M. Zink, K. Suh, Y. Gu, and J. Kurose, "Watch global, cache local: YouTube network traffic at a campus network-measurements and implications," *ACM MMCN*, 2008.
[3] "IEEE 802.15 WPA Task Group 8 (TG8) Peer Aware Communications," *http://www.ieee802.org/15/pub/TG8.html*, 2012.
[4] I. Constandache, R. Choudhury, and I. Rhee, "Towards mobile phone localization without war-driving," *IEEE INFOCOM*, 2010.
[5] "WiFi AP Map in Seoul," *http://findzone.internet.qook.co.kr/web2/location/paranmap.asp?X=205897&Y=445055*.
[6] "New York Census," *http://quickfacts.census.gov/qfd/states/36/3651000.html*.
[7] M. Cha, H. Kwak, P. Rodriguez, Y. Ahn, and S. Moon, "I tube, you tube, everybody tubes: analyzing the world's largest user generated content video system," in *ACM SIGCOMM*, 2007.
[8] X. Cheng, C. Dale, and J. Liu, "Statistics and social network of youtube videos," in *IEEE IWQoS*, 2008.
[9] K. Lee, S. Hong, S. Kim, I. Rhee, and S. Chong, "Slaw: A new mobility model for human walks," *IEEE INFOCOM*, 2009.
[10] K. Lee, I. Rhee, J. Lee, S. Chong, and Y. Yi, "Mobile data offloading: how much can WiFi deliver?" in *ACM CoNext*, 2010.
[11] G. Ananthanarayanan, V. Padmanabhan, L. Ravindranath, and C. Thekkath, "Combine: leveraging the power of wireless peers through collaborative downloading," in *ACM MobiCom*, 2007.
[12] L. Keller, A. Le, B. Cici, H. Seferoglu, C. Fragouli, and A. Markopoulou, "Microcast: Cooperative video streaming on smartphones," *ACM MobiSys*, 2012.
[13] L. McNamara, C. Mascolo, and L. Capra, "Media sharing based on colocation prediction in urban transport," in *ACM MobiCom*, 2008.