On the Feasibility of a User-Operated Mobile Content Distribution Network

Ioannis Psaras[*] [§], Vasilis Sourlas[*], Denis Shtefan[*], Sergi Rene[*], Mayutan Arumaithurai[†],
Dirk Kutscher[‡] and George Pavlou[*]
[*]Dept. of Electronic and Electrical Engineering, University College London, UK.
[†] Institute of Computer Science, University of Goettingen, Germany.
[‡] Huawei German Research Center, Munich, Germany.
[§] Fluentic Networks, UK.
Email: {i.psaras, v.sourlas, denis.shtefan.13, s.rene, g.pavlou}@ucl.ac.uk,
mayutan.arumaithurai@cs.uni-goettingen.de, Dirk.Kutscher@huawei.com

***Abstract*—The vast majority of mobile data transfers today follow the traditional client-server model. Although in the fixed network P2P approaches have been exploited and shown to be very efficient, in the mobile domain there has been limited attempt to leverage on P2P (D2D) for large-scale content distribution (*i.e.,* not DTN-like, point-to-point message transfers). In this paper, we explore the potential of a *user-operated, smartphone-centric content distribution model for smartphone applications*. In particular, we assume source nodes that are updated directly from the content provider (*e.g.,* BBC, CNN), whenever updates are available; destination nodes are then directly updated by source nodes in a D2D manner. We leverage on sophisticated *information-aware and application-centric connectivity* techniques to distribute content between mobile devices in densely-populated urban environments. Our target is to investigate the feasibility of an opportunistic content distribution network in an attempt to achieve widespread distribution of heavy content (*e.g.,* video files) to the majority of the destination nodes. We propose *ubiCDN* as a ubiquitous, user-operated and distributed CDN for mobile applications.**

***Keywords*—mobile video distribution, D2D, Wi-Fi Direct.**

I. Introduction

Modern smartphone technology is ubiquitous and supports a multitude of D2D connectivity (*e.g.,* Wi-Fi Direct, Bluetooth) and Internet accessibility options (*e.g.,* LTE, Wi-Fi), which can be exploited simultaneously [1][2]. These connectivity opportunities together with the fact that modern smartphone devices have large amounts of storage available enables *client-server-like* communication between mobile devices themselves. The concept of opportunistic networking exploits natural human mobility as an opportunity to facilitate content dissemination [3][4]. Mobility, which is usually perceived as a challenging and degrading factor in terms of the network's performance, can now be considered as an opportunity to disseminate content in mobile devices [5].

At the same time, there is a growing gap between the amount of storage that a smartphone possesses and the amount of network data that a user can consume per month (*a.k.a.* mobile data cap). The average smartphone on the low-end of the price spectrum holds roughly 16 GBs of storage capacity, while trends want this number to roughly double every year[1]. The price of a GB in a *microSD* card is between \$1-\$3 (*i.e.,* between \$35-\$60 for 128GBs of storage). On the other hand, in terms of network data allowance, the average monthly data-cap is at about 2GBs/month (ranging from less than 1GB in EU up to 30GBs in the US for contracts costing more than \$40/month).

According to the above statistics, the ratio of *"storage capacity to network data allowance"* for cell-connected smartphones is roughly equal to 16 (32GBs of storage capacity and 2GBs of data per month), *i.e.,* users can have at least 16 times more data stored in their devices than can send/receive over the mobile/cell network within a month[2]. Adding a 64GB *microSD* card (presently costed around \$20), the ratio of *"storage capacity to data cap"* skyrockets to 48.

In terms of mobile data consumption, a Standard Definition YouTube video consumes approximately 2MB/min of network data, translating to 120MB for one hour of streaming. This figure varies widely across platforms and can reach up to 20MB/min for High Definition (HD) video - see Fig. 1 (left)[3]. This means that one hour of YouTube streaming per day (*e.g.,* two 30min one-way commutes) consumes a 2GB monthly plan within 16 days, iPlayer consumes a 4GB plan in 17 days and Netflix consumes a 12GB data plan in just 20 days - see Fig. 1 (middle).

There is a clear bottleneck between the modern smartphones' storage capacity, the requirements of mobile multimedia applications and the users' monthly data allowance. Given the increasing trends of mobile video and the fact that teenagers are considered *"video natives, as they have no experience of a world without online video streaming"* [6], there is a pressing need for an infrastructure that overcomes this bottleneck and natively supports mobile video delivery at scale.

Putting it all together, a 32GB smartphone can hold more than 10,000 minutes of YouTube or iPlayer video (*i.e.,* 166 hours of streaming or 1,000 ten-minute videos) - see Fig. 1 (right). That said, *the average smartphone today is a small-scale, always-on, always-connected and mobile data-centre* for

[1] https://thomas.vanhoutte.be/blog/2015/12/29/mobile-phone-timeline/

[2] Note that the same figure for broadband/Wi-Fi-connected devices tends to zero, as most broadband deals come with unlimited downloads.

[3] Note that these figures are for 720p screens. As technology progresses and for smartphones/tablets towards the higher-end of the price spectrum higher resolution will result in considerably higher demands for bandwidth.

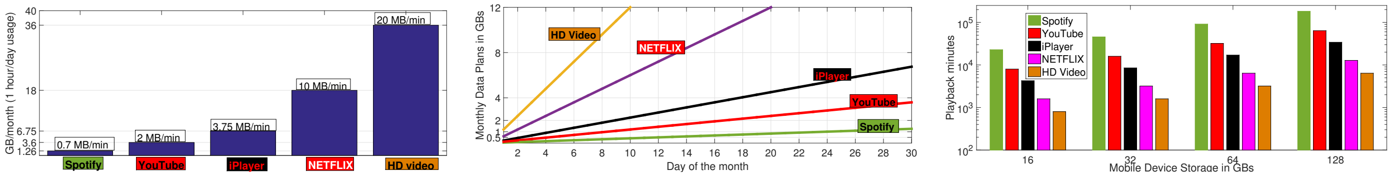

Fig. 1. Application data consumption and smartphone storage capabilities.

short video clips, songs or multimedia headline feeds.

In this paper we investigate the feasibility of a *user-operated content distribution network for mobile smartphone applications*. In terms of applications, we focus on news-related apps (*e.g.,* BBC, CNN) that update their text and video content frequently throughout the day. *Source and destination nodes* are the users' devices that have the application installed. In particular, we assume a fixed number of *source nodes* who are directly updated by the content/application provider through cell or Wi-Fi connectivity. Exploiting mobility patterns in urban environments, the source nodes disseminate content updates to *destination nodes* through smart *information-aware and application-centric connectivity* [7].

Indeed, news applications for smartphone devices today *i)* mainly come in the form of text content, *ii)* they get regular updates during the day, and, *iii)* offer limited multimedia/video content which is mainly consumed when connected to broadband networks. The reason behind consuming video content under broadband connections is that users cannot afford (in data terms) to watch video content when connected to the cell network.

In contrast to the traditional CDN *modus operandi*, where users receive content reactively (*i.e.,* after they request for the content), here, we argue that in the mobile space users' applications need to be updated *proactively*. Therefore, the system's target is to disseminate the most up-to-date content to *destination nodes* before users actually attempt to read the latest news.

Through extensive simulations we find that in densely populated urban environments a small percentage of source nodes (*e.g.,* around 5%) is enough to disseminate content to almost 70% of destination nodes (*i.e.,* the population that has the specific application installed). Application updates are based on information-aware D2D connectivity [7], while transfers are realised through WiFi Direct. We test the concept of a ubiquitous, user-operated CDN (*ubiCDN*) under a variety of network conditions and compare its performance with an information-agnostic D2D, floating content scheme. We measure the energy-consumption of the system in a small proof-of-concept prototype and find that if content distribution targets the right nodes and is kept to reasonable levels (in terms of exchanged messages), then energy consumption is not an issue (*i.e.,* consumes roughly 1% of battery per hour).

II. System Architecture

We argue that the current *client-server centric* data transfer model leaves enormous amounts of resources unexploited. Powerful end-user devices can act as data sources and take advantage of local connectivity (through Wi-Fi Direct, Bluetooth, Google Nearby[4]). Although these technologies have been around for quite a while (the first Bluetooth distribution came out more than a decade ago[5]), there have been surprisingly few applications (mainly gaming and chat applications, *e.g.,* FireChat) that exploit such connectivity in a user-transparent way. That said, links between devices (with speeds that can reach up to 250Mbps for Wi-Fi Direct and 25Mbps for Bluetooth 4.0) remain largely unused.

In this paper we set off to build an *information-aware and application-centric* D2D connectivity framework and realise a distributed and ubiquitous content distribution platform, acronymed *ubiCDN*. *ubiCDN* focuses on smartphone applications that receive *frequent* (in the order of 1-3 hours) and *heavy* (in the order of 5MBs or more) updates. Such applications are normally news applications, *e.g.,* newspapers, national broadcasters, or topic-specific online TV-like channels, *e.g.,* YouTube, Periscope. Such applications receive updates regularly throughout the day, which, however, are not directly pushed to the user's device to avoid consuming their cellular data[6].

In the current/traditional *client-server* model, content is pulled from the (CDN) server upon the user's request. Instead, *ubiCDN* attempts to prefetch, distribute and make content available to users by default (*i.e.,* before the user attempts to check for updates). Although someone might argue that prefetching vast amounts of data to mobile devices will result in waste of bandwidth resources (given that the user will only read/watch/listen to a small subset of updates), *ubiCDN* utilises local D2D connectivity to complete content transfers. That said, it does not consume network bandwidth resources, while the impact on the device's energy consumption is negligible, as we show through our proof-of-concept testbed measurements in Section IV. *ubiCDN* consists of the following node groups:

- **Source nodes** receive updates directly from the content/application providers in a push mode. Their job is to disseminate the content further to users that have the same application installed on their device, or are subscribed to the same YouTube-, or Spotify-like channel.
- **Destination nodes** are passive nodes that have a number of applications installed on their device (or are subscribed to a number of YouTube-, or Spotify-like channels) and expect to receive updates on those while roaming. Local D2D data transfers take place when a source node meets a destination node whose application or channel has outdated content.
- **Relay nodes** are destination nodes that temporarily become (act like) source nodes once they get updated. This mode of operation reduces the number of source nodes needed and accelerates significantly the distribution of content in the mobile, D2D domain.

[4] https://developers.google.com/nearby/

[5] https://www.bluetooth.com/about-us/our-history

[6] In some cases, heavy content can be pushed when the user is connected to a Wi-Fi AP, although we have not found any application that pushes heavy updates transparently, *i.e.,* without the user's intervention.

A. D2D and Information-Centric Connectivity

We assume that nodes use the Wi-Fi Direct specification to exchange application content updates. Wi-Fi Direct is a Wi-Fi Alliance technical specification, that allows devices to connect to each other, form groups and transfer content in a D2D manner without requiring an access point (AP). Smartphone devices with the appropriate hardware (*e.g.,* any Wi-Fi device using nl802.11 driver[7]) can connect directly to each other. For example, Android phones use the Wi-Fi P2P framework that complies with the Wi-Fi Direct certification program. Wi-Fi Direct devices need to negotiate their role in the communication: one of them plays the role of the AP, called *Group Owner* (GO) and the other device(s), that may include legacy non Wi-Fi Direct devices, associates with the GO as clients. In the following we describe some of the details of the protocol.

Device Discovery: Wi-Fi Direct devices listen and send probe requests on the so-called social channels (channels 1, 6 and 11 in the 2.4GHz bands), and respond to the requests with probe responses, prior to the group formation.

Service/Application discovery: Before the establishment of groups, devices need to exchange information about their available applications. Wi-Fi Direct devices can advertise services/applications by attaching information at the management frames (*i.e.,* beacons, probe requests and responses) through the usage of the Generic Advertisement Protocol (GAS) specified in 802.11u [8]. *ubiCDN* exploits the management frames of GAS in order to exchange information related to the device's applications. Through management frames, *ubiCDN* source nodes share the applications they distribute content for, as well as the latest update they have, *e.g.,* `BBC-Sports-1100am`. GAS management frames can be used to share information regarding *application name*, *transport protocol*, *port number*. This way, users can share necessary application information before forming groups.

Group formation: Once two devices have found each other and are willing to share information, they start the group formation following one of the three different ways:

- **Standard mode:** The basic GO Negotiation phase is implemented using a three-way handshake, sending the GO negotiation Request, Response and Confirmation messages. The two devices agree on which device will act as GO and on the channel where the group will operate. For the prototype results we present later in Section IV, we use this Standard mode, since it is the default mode for the Android implementation.
- **Autonomous mode:** A device may autonomously create a group, where it immediately becomes the GO. It starts sending beacons at a chosen channel, without initiating any negotiation with any other device. Other devices can discover and participate in the established group using traditional scanning mechanisms. In this mode no GO Negotiation phase is required.
- **Persistent mode:** Devices can declare a group as persistent, by using a flag in the capabilities attribute present in beacon frames, probe responses and GO negotiation frames. In that way, the devices forming the group store network credentials and the assigned GO and client roles for subsequent re-instantiations of the group.

Note that devices implementing Wi-Fi Direct may support concurrent operation through multiple groups simultaneously.

B. Incentives and data integrity

As mentioned in [9] the integration of an effective incentive mechanism in a user-operated offloading mechanism is a challenging problem. The application content provider reduces its expenses to the CDN provider that manages its content and the corresponding cellular providers significantly decrease the stress on their links. Incentives should be provided to both *Source* and *Relay* nodes in order for a user-operated CDN to succeed. In an attempt to encourage source nodes to participate and destination nodes to extent the corresponding relay period, a *"couponing"* scheme similar to the one presented in [10] can be exploited. *Coupons*/Compensation comes from the content/application provider and in terms of monetary amount should be smaller than the CDN savings discussed above. According to [10], compensation is based on the actual volume of the delivered/offloaded content. This approach is highly suitable for the case of *ubiCDN* presented here as different source (and relay) nodes have different mobility patterns (*e.g.,* office worker vs. bus driver) and will therefore, deliver different amounts of content. In case of scenarios of mutual interest, such as a football match or a music concert, incentives are built among fans of the same team/artist [2].

In order to ensure data integrity (*i.e.,* content is what it claims it is and has not been modified by intermediate users), *ubiCDN* integrates digital signatures (*e.g.,* HMAC) based on Public Key Infrastructure (PKI). This setup prerequisites that the digital certificates used by the application provider do not expire while the users are disconnected. This way, users can easily authenticate the content they are receiving. Other security vulnerabilities, such as, eavesdropping, privacy violation, or denial-of-service (DoS) attacks, are out of the scope of this paper. However, related literature provides ways to deal with such issues in D2D communications (*e.g.,* [11]).

III. Performance Evaluation Setup

A. Evaluation Setup and Assumptions

For the evaluation of the proposed user-operated CDN we extended the *ONE* simulator [12]. ONE is a discrete event simulator for opportunistic network environments, and is capable of generating different node map-based movements using various models, routing messages using different DTN routing schemes and provides interfaces for application level extension.

The scenario chosen for assessment was that of a busy city environment, namely Helsinki city center, with a fixed population of users carrying a mobile device capable of D2D connectivity and support for multiple smartphone applications, and simulating different mobility models for multiple set of users defined in Section III-B.

Applications: We assume a given set of ten different smartphone applications, where each application generates content updates periodically (*e.g.,* a news application like BBC news could update its content every hour). We call the update interval *"application update period/interval"*. For the purposes of this study and unless otherwise stated, we assume that the *application update period/interval* is $1 hour$ and that the default

[7] https://wireless.wiki.kernel.org/en/developers/documentation/nl80211

size of each update message is equal to 5 MB (although we experiment with updates up to 200MB - Section IV-C).

We approximate application popularity by a Pareto distribution, given ample evidence from related literature that smartphone application popularity follows such distribution, *e.g.,* [13], [14]. According to the used Pareto distribution, 28% of the total number of installed applications in the mobile devices of the destination nodes are instances of the most popular application, whereas the least popular application counts only 4.5% of that number (*i.e.,* skewed, heavy-tailed distribution, where some applications are installed in almost all devices, whereas the remaining are present in a smaller number of devices). Moreover, applications are randomly assigned to users, but each user has at least one application installed.

For the purpose of the present study, we assume that the D2D transmission capacity of Wi-Fi Direct is equal to 31.25 Mbps and the range of transmission of each device is equal to 60 meters, with this capacity being different based on the distance of two nodes (*i.e.,* inversely proportional to the square of their distance). Alternative direct communication, *e.g.,* through Bluetooth, is also an option, although the details are left for future investigation. The main difference between the two technologies lies in their transmission range capabilities and energy consumption.

According to *ubiCDN*, whenever a source node is in range with a destination node it will first check for matching applications, and if found, the corresponding content will be compared. If the discovered node has out of date content the source node will push the update via a direct message to the destination node. These messages are also *prioritised by the application popularity*, meaning that more popular applications will be updated first and less popular afterwards. If relaying is also enabled, the updated destination node will become a relay node either for a fixed amount of time or until it transmits one successful update (*i.e.,* single contact relay), depending on the node's settings.

What we compare with: For comparison purposes we have extended the ONE simulator with an application- and information-agnostic content delivery scheme. According to this last scheme, source nodes blindly send blanket update messages to nodes they encounter without comparing content of the same application or checking whether the encountered node has the latest update or not.

Effectively, this scheme imitates the behaviour of the *"Floating Content"* concept [15], where distance vector is set to infinity (*i.e.,* the borders of the city), while the time anchor is set to the *application update period/interval*. In the following, we denote this distribution method as *fltCDN*. Intuitively, and as we show later in our evaluation results, it is clear that *fltCDN* introduces huge amount of duplicate messages exchanged between nodes. To make the comparison more pragmatic, we have built a combination of the Floating Content concept with the *"First Contact"* dissemination strategy [16]. Single contact relay or First Contact [16] is a well-known traditional DTN routing and replication strategy, according to which nodes forward the messages they have to the first node they encounter only. The combination of First Contact with Floating Content effectively reduces the number of replicas in the system in an attempt to reduce duplicate messages and overhead. For our proposed *ubiCDN* approach, the "First Contact" is an application-based decision. Table I contains the basic schemes (*ubiCDN* and Floating Content (*fltCDN*)) that we will compare in the following section, whereas Table II gives the default values for the various system parameters.

TABLE I. EXAMINED CONTENT DISSEMINATION MECHANISMS.

Strategy	Dissemination by
ubiCDN - w/o Relay	Sources only
ubiCDN - w/ Relay	Sources & Destinations for t time units (*i.e.,* Relaying)
ubiCDN - Single Contact	Sources & Destinations for one contact only
fltCDN - w/ Relay	Sources & Destinations for t time units (*i.e.,* Relaying)
fltCDN - Single Contact	Sources & Destinations for one contact only

TABLE II. SUMMARY OF THE EVALUATION PARAMETER SETTINGS.

Parameter	Value
Number of Applications	10
Number of Source Nodes	50
Number of Destination Nodes	1000
Size of each message	5 MB
App. update period	1 hour

B. Urban Movement

We have used a variety of movement models/patterns (included in the ONE simulator) in order to assess the efficiency of the proposed user-operated, *ubiCDN* approach to mobile content dissemination. We have used the Helsinki city population and city centre as the default urban environment (area of size equal to 8.3km × 7.3km). By default, we assume that the population of the destination nodes is equal to $D = 1000$, whereas the default population of the source nodes is equal to $S = 50$ (although we also experiment with different ratios of Source-Destination nodes in Section IV-B).

The distribution of source nodes is as follows: out of the 50 nodes, we assume that 18 are buses that follow predefined routes, whereas the rest nodes (*i.e.,* 32 in the default scenario) are assumed to be users that follow the working day movement model [17].

Destination nodes are configured as follows: out of the 1000 nodes 20% are assumed to be tourists. Tourists choose random destinations (either total random points in the map or one of the seven "points of interest" (*i.e.,* tourist attractions) in the city centre) to which they travel following the shortest path and wait randomly $2-15$ minutes. The majority of the *destination* nodes, *i.e.,* the remaining 80%, are assigned the working day movement model [17], which allows them to travel to designated office spaces on the map and travel for other evening activities later in the day. All nodes start at their base/home and travel to their office, either directly by car (50% of nodes) or by bus (remaining 50%). Once they reach the office, they spend 7 hours there and at the end of the office day there is a 50% chance the node will go for an "evening activity" and 50% chance it travels home for the rest of the day.

C. Metrics

Satisfaction Ratio. In contrast to a fixed network CDN, where the target is to reactively serve user requests within

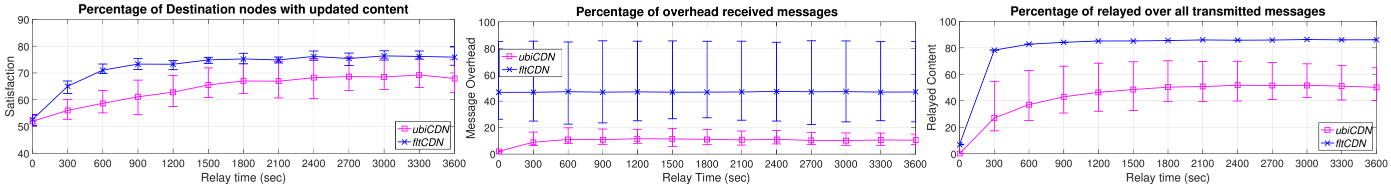

Fig. 2. The impact of the relay time in the performance of the examined content dissemination mechanisms.

deadlines (that is, after the user has requested for some content), the target of a mobile-focused distribution network should be (we argue) to proactively load content to users' devices. In the context of smartphone applications, proactively comes in terms of the *"application update period/interval"*. That said, one of the main metrics used here is the *Satisfaction ratio*, defined as:

Satisfaction ratio (in % of destination nodes): The percentage of the destination nodes with updated content over all destination nodes at the end of each application update period.

Message Overhead. Given the population (and therefore device) density in urban environments, D2D proactive dissemination of content raises questions of overhead in terms of messages transmitted to non-interested recipients, or recipients that already have received the update. Therefore, a second metric of interest is *Message Overhead*, defined as:

*Message Overhead (in % of total received messages): The percentage of messages rejected by the users as duplicates (*i.e., *nodes have already received the update) or messages of no interest (in the floating content mechanisms), or transfers that did not complete due to users moving out of range of each other, over the total received update messages at the end of each update period.*

Relayed Content. One of the main challenges in realising a user-operated CDN comes in terms of costs, that is, how many source nodes would be needed (and therefore, compensated) in order to achieve proactive distribution of content to the majority of destination nodes and at what timescales would delivery be achieved. Relaying is an important feature of message distribution and therefore, the amount/percentage of messages that is being transmitted through relaying is a central metric in our evaluation.

Relayed content (in % of total transmitted messages): The percentage of messages delivered by relay nodes over all transmitted messages at the end of each update period.

In each of the three scenarios analysed in the next section, the above metrics are evaluated with regard to the impact of: *i) relay time*, *ii) number of source nodes*, and *iii) update message size*. We also present energy consumption statistics of source and relay nodes in a user-operated CDN based on a proof-of-concept prototype of *ubiCDN*. In what follows, the above metrics are averages over 12 application update periods.

IV. Results

In this section we illustrate the performance of *ubiCDN* and we compare it against the floating content mechanism (*fltCDN*) described above.

A. Impact of the relay time

Initially, we examine the impact of the relay time "t" in the performance of the two content dissemination mechanisms in Fig. 2. The relay time "t" is the amount of time that each updated destination node relays the newly acquired application update(s). We examine relay times that vary from $t = 0$ (no relay) to $t = 3600$ (1-hour relay), which we assume as the default update period of each application. Note that the plots are depicted as error bars, where the error bars correspond to upper and lower bound results for the most and least popular application, respectively and not the typical standard deviation.

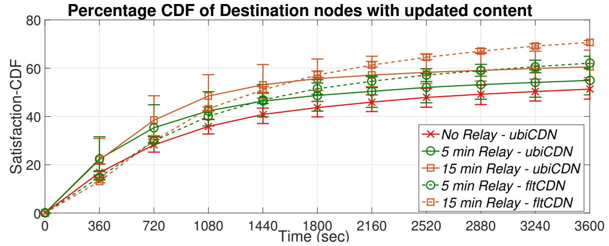

Fig. 3. The impact of the relay time in the Satisfaction CDF of the examined content dissemination mechanisms.

From the Satisfaction ratio plot in Fig. 2, we observe that when only source nodes disseminate content (*i.e.,* no Relaying) already half of the destination nodes manage to retrieve the updated content (first point in left figure). When destination nodes start to relay the received content, even for a very small amount of time (*i.e.,* 5-15 minutes), the satisfaction increases by up to an extra 40%. This is also obvious from the Relayed content plot (right plot in Fig. 2) where we see that the total number of messages distributed by the relay nodes can reach up to 80% of the total transmitted messages in the *fltCDN* case and 50% in the *ubiCDN* case. However, increasing the relay time to more than 15 minutes (900 secs onwards) brings no substantial gain in terms of satisfied users (left plot in Fig. 2). This result illustrates the fact that while some users move in the city centre and therefore can interact and receive updates, some others remain in non-reachable areas, *e.g.,* offices or outskirts of the city. This result serves as an upper bound of the performance of the examined content dissemination mechanisms, given the specific settings - Table II.

This result is more clear in Fig. 3 where we depict the Satisfaction Cumulative Density Function (CDF), which shows how many users acquire the updated content over the duration of an application update interval (1-hour) for different relaying durations. From Fig. 3 we observe that the vast majority of the updated destination nodes receive updates within 20-25 minutes after the release of the update, and only a small portion of them is updated towards the end of the application update interval. For example, for the "15-min Relay" case, 62% of the destination nodes are updated at the end of the 1-hour update period, 70% of which are updated within the first 15 minutes (900 secs). The same holds for the floating content schemes too, even for relay times that increase up to the application update interval (*i.e.,* 1-hour).

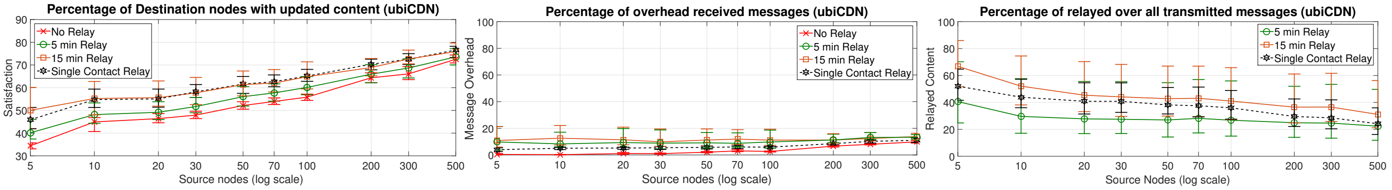

Fig. 4. The impact of the number of source nodes relay time in the performance of the *ubiCDN* mechanism.

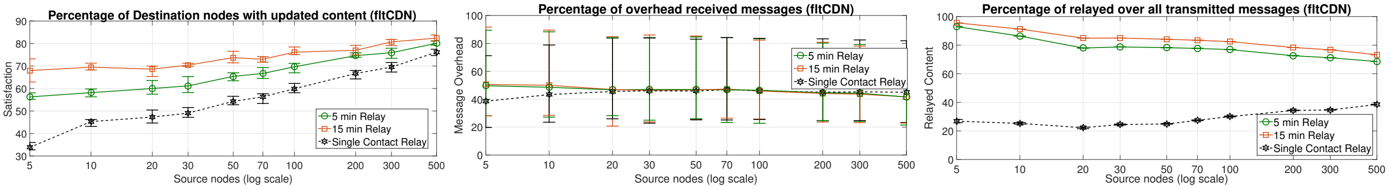

Fig. 5. The impact of the number of source nodes relay time in the performance of the *fltCDN* mechanism.

Comparing the performance of *ubiCDN's* information-aware connectivity with information-agnostic floating content in Fig. 2, we observe that the *fltCDN* scheme performs around 15% better in terms of satisfaction ratio. However, as expected with this "aggressive" approach, it creates at least four times more redundant transmissions (*i.e.,* overhead), as shown in the Message Overhead plot (middle figure in Fig. 2). In case of *ubiCDN*, message overhead is caused by unfinished transfers due to users getting out of range of each other before the transfer completes. On the other hand, as expected, the *fltCDN* approach, which effectively, floods nodes with messages has severe consequences in terms of the energy spent by user devices. We present energy consumption results later on in this section.

From the top error bars, which denote the most popular application, it is interesting to note that the most popular application is updated in almost the same amount of destination nodes using either of the two dissemination mechanisms (see 2100 secs point in Satisfaction ratio plot). However, information-aware transfers through *ubiCDN* introduce only around 20% of message overhead, whereas in the *fltCDN* case almost 90% of the transmitted messages for that particular application is redundant (Message Overhead plot in Fig. 2). On the other hand, due to the limited interest for the least popular application (bottom error bars), we see that the *fltCDN* mechanism manages to update 25% more destination nodes than *ubiCDN*. Based on that, we can deduct that a hybrid approach where popular applications are disseminated using information-aware connectivity and the least popular based on the floating content scheme would get the best of both worlds, resulting in higher satisfaction ratios and lower overheads. It is clear (and intuitively expected) that less popular applications require some kind of data mules to reach interested users, whereas popular ones can be updated with the minimum overhead, given that most users in the area are interested in this application's content. We leave the specific design details of this tradeoff for future investigation.

Based on the results of Fig. 2, in the rest of our evaluation, we only consider No Relay (*t = 0*), 5min (*t = 5 minutes*) and 15mins (*t = 15 minutes*) relay durations. From this point on, we also introduce the "single contact" relay approach presented in the previous section. Note that the No Relay *fltCDN* and No Relay *ubiCDN* mechanisms perform the same as regards Satisfaction ratio (see 0 sec relay point in Satisfaction Ratio plot, left plot in Fig. 2). This is because only source nodes disseminate updates for all applications. For that reason in the rest of the evaluation we do not depict the No Relay *fltCDN* scheme, although it performs significantly worse in terms of overhead.

B. Impact of the number of source nodes

In Fig. 4 and Fig. 5 we depict the impact that the number of source nodes S have in the performance of *ubiCDN* and *fltCDN*, respectively. Note that the number of destination nodes is always equal to $D = 1000$ nodes. The source nodes are always buses when $S \leq 18$, whereas when $S > 18$ the remaining nodes are chosen to be nodes that follow the working day movement model. This option is more rational since tourists are usually visiting a place for a limited amount of time, whereas "workers" tend to stay in one place for longer periods.

We observe exponential increase in the Satisfaction Ratio (left plots in Figs. 4 and 5) as the number of sources S increases (linear increase in the plots for logarithmic scale x-axis). This increase is more pronounced in the case of *ubiCDN* and for small relay times. With a small number of source nodes *ubiCDN* performs up to 40% worse compared to the corresponding floating content schemes, but this difference is only marginal when the number of source nodes increases to 5% of the destination nodes (*i.e.,* 50 nodes).

Message overhead (middle plots in Figs. 4 and 5) is not significantly affected by the number of source nodes, with the *fltCDN* schemes requiring on average 2.5× times more redundant transmissions than *ubiCDN*, and up to 5× times more overhead message transmissions for the most popular application (top error bar in plots). The Relayed content ratio (right plots in Figs. 4 and 5) is also significantly reduced when we increase the number of the sources (especially in case of *ubiCDN*), since users tend to get updated directly by source nodes instead of relay nodes.

An interesting observation in Fig. 4 is the performance of the Single contact relay *ubiCDN* mechanism. We observe that Single Contact relay performs as good as the 15-min relay scheme and significantly better than the other relay schemes. This shows the efficiency of the more sophisticated information-aware and application-centric dissemination scheme over the simplistic floating content one, where a relay node interacts with other nodes in its vicinity only when both share similar interest. This allows the relay nodes to maintain the single relay attribute (the single contact constraint) for

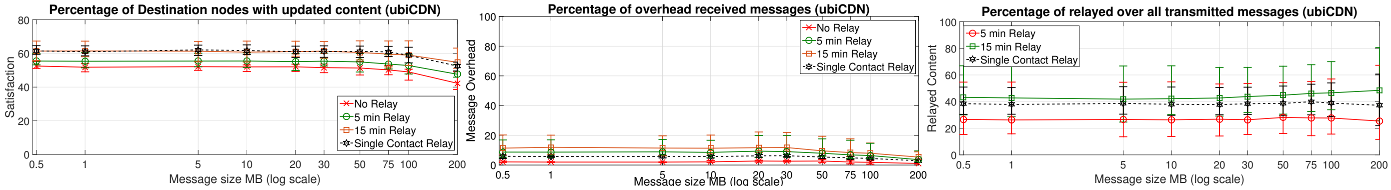

Fig. 6. The impact of the message size in the performance of the *ubiCDN* mechanism.

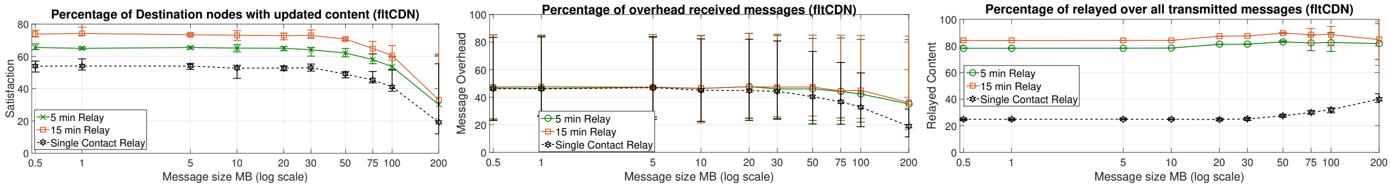

Fig. 7. The impact of the message size in the performance of the *fltCDN* mechanism.

more time, whereas in the *fltCDN* scheme a node interacts and loses the relay attribute almost instantly after the retrieval of an update message.

C. Impact of the update message size

Up until now, we assumed that each application update message is 5MB (*e.g.,* a 2-3 minutes standard YouTube video). In Fig. 6 and Fig. 7 we examine the impact of the update message size on the performance of *ubiCDN* and *fltCDN*. The overall observations are similar to the previous experiments for all metrics, *e.g.,* the satisfaction ratio is similar to the previous experiments, with *fltCDN* performing $10\% - 15\%$ better than *ubiCDN* as long as the size of each update message is smaller than 100MB (left plots in Figs. 6 and 7). From that point onwards, the *fltCDN* scheme performance declines sharply to the extent that the information-aware mechanisms perform almost 90% better than the corresponding floating content ones (for update message size of 200MB).

We remind that we assume Wi-Fi Direct for the D2D communication (*i.e.,* significantly larger transmission capacity compared to Bluetooth). This means that as long as the messages are relatively small, even the naive *fltCDN* mechanism manages to successfully transmit the updated content between two nodes that "meet" for a small period of time. On the other hand, when the messages are larger (*e.g.,* large or HD video messages) a relay or a source node following the aggressive *fltCDN* mechanism does not manage to disseminate the updated content to other destination nodes during the contact period. Note that in *fltCDN*, and assuming ten applications, a node has to deliver 2GB of data, assuming 200MB of message size. From this finding we further deduct that the proposed information-aware and application-centric D2D dissemination mechanism (*ubiCDN*) is not only efficient due to the low overhead of the communication, but is also the only viable solution when larger amount of data needs to be transmitted. Furthermore, in the Helsinki city scenario, the majority of source nodes and all the relay nodes are slow-moving "workers". In a futuristic faster-moving vehicular D2D scenario, where vehicles meet for even smaller periods of time a naive information-unaware scheme (*fltCDN*) will perform bad even for very small update message sizes. This further enhances the necessity of a more sophisticated context-aware and application-centric dissemination approach.

D. Impact on Energy Consumption and Battery Depletion

Last, but certainly not least, we look into the energy consumption of user-operated CDNs. Energy is the price paid by the system in order to disseminate content in a D2D manner and therefore, cannot afford to be overlooked in our feasibility study.

In Tables III and IV we show preliminary results using real devices (Galaxy Tab A tablet and Samsung Note 4 smartphone) and transmitting different file sizes using Wi-Fi Direct. These results are obtained with two static devices separated by around 10 meters and with a RSSI value close to -50dbm. We carried out a set of experiments using file sizes of 5 MBs, 50 MBs and 100 MBs and we extrapolated these results to the number of messages sent by source and relay nodes using the *ubiCDN* and *fltCDN* mechanisms during an update interval of 1 hour. In *fltCDN*, each source node has to transmit between 660-785 messages during one update interval (*i.e.,* 660 messages in the 1 hour relay case and 785 in the no relay scenario), whereas each relay node transmits up to 210 messages (*i.e.,* 1 hour relay case), including the overhead messages. The corresponding number of transmitted messages in the *ubiCDN* mechanism are 41-62 messages for the source nodes and up to 3 messages for the relay ones. We also extrapolated from this energy consumption the percentage of battery consumed using a tablet (*i.e.,* 7000 mAh/25.9Wh capacity) or using a smartphone (*i.e.,* 3200 mAh/12.4Wh capacity). The energy consumed by the Wi-Fi Direct application is measured using the Trepn Profiler tool [18].

From Tables III and IV, we can observe that the energy consumption for the *ubiCDN* source nodes goes from 106.6 mWh in the best case (41 messages sent) when sending 5 MB messages, to 321.16 mWh in the worst case (62 messages) when sending 100 MBs (including the energy required for the group formation mentioned in Section II-A). This means the percentage of the battery consumed is between 0.86% and 2.59% for a smartphone, and between 0.41% and 1.24% for a tablet. In case of relay nodes the energy consumption goes from 7.8 mWh to 15.54 mWh, meaning from 0.06% to 0.12% of the battery for a smartphone and 0.03% to 0.06% for a tablet, respectively. From this analysis we can consider that energy consumption is not an issue in the case of the *ubiCDN* case even if we assume large update messages. However, in the *fltCDN* case the non-application aware content delivery and the relatively increased message overhead will deplete quite fast the battery of a user's device for source

TABLE III. Source nodes energy consumption results. Summary of 1 Hour, ubiCDN: 41-62 Msgs, fltCDN: 660-785 Msgs

Message Size:		5 MB	50 MB	100 MB
ubiCDN	mWh	106.6-161.2	179.99-272.18	212.38-321.16
	% Battery phone	0.86-1.3	1.45-2.19	1.71-2.59
	% Battery tablet	0.41-0.62	0.69-1.05	0.82-1.24
fltCDN	mWh	1716-2041	2897.4-3446.15	3418.8-4066.3
	% Battery phone	13.83-16.45	23.36-27.79	27.57-32.79
	% Battery tablet	6.62-7.88	11.18-13.3	13.2-15.7

TABLE IV. Relay nodes energy consumption results. Summary of 1 Hour, ubiCDN: 3 Msgs, fltCDN: 210 Msgs

Message Size:		5 MB	50 MB	100 MB
ubiCDN	mWh	7.8	13.17	15.54
	% Battery phone	0.06	0.1	0.12
	% Battery tablet	0.03	0.05	0.06
fltCDN	mWh	546	921.9	1087.8
	% Battery phone	4.4	7.43	8.77
	% Battery tablet	2.1	3.55	4.2

nodes. For example, in Table III we see that smartphones can spend up to 32.79% of the battery, or tablets can spend up to a 15.7% of the battery, in a single update interval (*i.e.,* 1-hour). This means that a more sophisticated content aware dissemination mechanism is required in order not to discourage users from participating in a user-operated mobile CDN.

The results obtained in this energy consumption analysis, despite being preliminary with simple tests, are in line with the results presented in [19]. In [19] the authors report that an average smartphone can transmit up to 44GB of data before depleting the battery, with an average consumption of a 1 J/MB (*i.e.,* 1.38 mWh for a 5MB file) in a walking speed mobility scenario.

V. Related Work

Opportunistic networks can increase network capacity [20] and offload traffic from a cellular to a cellular-assisted device-to-device network [9][21]. They can also support communication and content exchange when the cellular infrastructure is under severe stress [2][22][23], whereas at the same time, are the only means of communication when the network infrastructure is down or inaccessible due to natural disasters or government censorship. In opportunistic networks, connectivity among devices is intermittent and communication can be very lossy. This might decrease the possibility for successful content forwarding. However, the fact that user movement and mobility patterns have limited degree of freedom and variation, and rather exhibit structural patterns due to geographical and social constraints [5][24], minimizes this uncertainty.

Smartphone usage is largely application-centric, meaning that users mainly rely on applications to gain access to information. For instance, according to [25] the top 100 applications (*i.e.,* Google Play Store or Apple iTunes) are responsible for almost 90% of the access time and 80% of the traffic volume. In most of the cases application usage is all about the exchange of content (*e.g.,* news, video-on-demand, emergency announcements), which in many cases can be of interest to multiple participants; *e.g.,* updates on concurrent sports matches, departure times for public transportation after a football match. Furthermore, a very big proportion of the information that users are interested in does not actually require access to the global network, but are rather targeting non-personalised services or information related to a local event (*e.g.,* a football match in [2]). Relaxing the stress to the network by looking for information locally increases the availability of Internet resources to those who do actually require access to remote services. In [26] the authors claim that enabling content-sharing between devices in sport events decreases bandwidth consumption by almost 50%.

The increased demand for mobile data and the corresponding relatively high skewness in the popularity of the requested content [13] [14] has led to an increased interest in the area of data offloading through opportunistic communication with or without the assistance of the cellular infrastructure. For example in [27] authors present a cellular assisted mechanism to serve user requests from other mobile users located geographically close, by clustering crowded places in dataspots and by tracking the location of users, as well as the content cached in their devices. An alternative approach is presented in [28], where authors propose the usage of vehicles as mobile caches, as a cheap and more versatile approach to small cells. User interests are delayed in a controlled fashion so that a passing by vehicle fulfil their interests, otherwise their interests are served by the cellular network in an attempt to minimize infrastructure load and guarantee maximum delays.

Several solutions have also been proposed to share information between mobile devices bypassing the Internet infrastructure or the coordination through the cellular network, *e.g.,* FireChat[8]. In [29], authors investigate the feasibility of a city-wide content distribution architecture composed of short-range wireless access points and examine how to improve the diffusion of information within a group of interested users. Their approach leverages various mixtures of fixed and mobile nodes that are exterior to that group and falls under the Pocket Switched Networking (PSN) paradigm [30]. Under the same principles, Haggle [31] proposed a data-centric network oriented towards sharing content in local mobile networks. Haggle enables seamless network connectivity and application functionality in dynamic mobile environments, separating application logic from transport bindings so that applications can be communication-agnostic. Finally, in [26] authors enable content sharing between users in crowded live events by realising a key-value store abstraction for applications, providing single-hop content discovery and sharing with the participation of local access points.

Other approaches have been based on Delay-Tolerant Networks (DTN), exploiting both its inherent capability to exchange data in opportunistic environments, and its in-network storage functionality. For instance the Floating Content [15] concept leverages ad-hoc communications among mobile users to share local information. According to [15], message and information replication is limited in time and space. A twisted application of the Floating Content concept is used in [32], where the proposed Locus mechanism tries to keep objects at specific physical locations in the network using whatever devices are present nearby. The goal is to keep each data object as close to its home location as possible by decoupling the data from the nodes that carry it. This is achieved by sending a request towards the home location of the data instead of searching for specific nodes.

[8] http://opengarden.com/firechat

VI. Conclusions and Future Work

We have explored the feasibility of a *user-operated, ubiquitous CDN (ubiCDN)* that targets distribution and dissemination of heavy video content to mobile devices. We have argued that in a mobile, cell-connected setting, heavy content needs to be pre-loaded to mobile devices, in contrast to broadband connected environments where content is delivered on demand. This is not least because mobile data consumption is limited by the users' monthly data caps. We build on the premise that modern mobile devices possess large amounts of storage and can act as data mules. Our results show that a small percentage of source nodes are enough to prefetch content to the majority of mobile devices in urban settings, while relaying can also be kept to a minimum (around 15 mins per relay node). Information-centric connectivity and data transfers keep energy consumption to a minimum, cancelling incentive concerns for mobile users to participate in the system. Our results prove the feasibility of a user-operated CDN for smartphone applications. Further investigation is needed in order to cover all corner cases of the system and also investigate incentives and business relationships.

Acknowledgments

This work has been supported by the EC H2020 UMOBILE project (GA no. 645124), the EC H2020 ICN2020 project (GA no. 723014) and the EPSRC, under grant no. EP/K019589/1 (the COMIT Project) and the INSP Early Career Fellowship (EP/M003787/1).

References

[1] S. Trifunovic, B. Distl, D. Schatzmann, and F. Legendre, "WiFi-Opp: Ad-hoc-less Opportunistic Networking," in *ACM CHANTS*, 2011.

[2] I. Wakeman, S. Naicken, J. Rimmer, D. Chalmers, and C. Fisher, "The fans united will always be connected: building a practical DTN in a football stadium," in *ADHOCNETS*, 2013.

[3] M. Conti and M. Kumar, "Opportunities in Opportunistic Computing," *IEEE Computer*, vol. 43, pp. 42–50, 2010.

[4] L. Pelusi, A. Passarella, and M. Conti, "Opportunistic networking: data forwarding in disconnected mobile ad hoc networks," *IEEE Communications Magazine*, vol. 44, pp. 134–141, 2006.

[5] T. E. Amah, M. Kamat, W. Moreira, K. A. Bakar, S. Mandala, and M. A. Batista, "Towards next-generation routing protocols for pocket switched networks," *Journal of Network and Computer Applications*, vol. 70, pp. 51–88, 2016.

[6] "Ericsson mobility report," November 2016. [Online]. Available: https://www.ericsson.com/assets/local/mobility-report/documents/2016/ericsson-mobility-report-november-2016.pdf

[7] K. V. Katsaros, V. Sourlas, I. Psaras, S. Rene, and G. Pavlou, "Information-centric connectivity," *IEEE Communications Magazine*, vol. 54, pp. 50–57, 2016.

[8] "Ieee standard for information technology-telecommunications and information exchange between systems-local and metropolitan networks-specific requirements. amendment 9: Interworking with external networks," *Amendment to IEEE Std 802.11*, pp. 1–208, 2011.

[9] B. Han, P. Hui, V. S. A. Kumar, M. V. Marathe, J. Shao, and A. Srinivasan, "Mobile data offloading through opportunistic communications and social participation," *IEEE Transactions on Mobile Computing*, vol. 11, pp. 821–834, 2012.

[10] A. Garyfalos and K. C. Almeroth, "Coupons: A multilevel incentive scheme for information dissemination in mobile networks," *IEEE Transactions on Mobile Computing*, vol. 7, pp. 792–804, 2008.

[11] A. Zhang, J. Chen, R. Q. Hu, and Y. Qian, "Seds: Secure data sharing strategy for d2d communication in lte-advanced networks," *IEEE Transactions on Vehicular Technology*, vol. 65, pp. 2659–2672, 2016.

[12] A. Keränen, J. Ott, and T. Kärkkäinen, "The ONE Simulator for DTN Protocol Evaluation," in *SIMUTools*, 2009.

[13] T. Petsas, A. Papadogiannakis, M. Polychronakis, E. P. Markatos, and T. Karagiannis, "Rise of the planet of the apps: A systematic study of the mobile app ecosystem," in *ACM IMC*, 2013.

[14] H. Li, X. Lu, X. Liu, T. Xie, K. Bian, F. X. Lin, Q. Mei, and F. Feng, "Characterizing smartphone usage patterns from millions of android users," in *ACM IMC*, 2015.

[15] E. Hyyti, J. Virtamo, P. Lassila, J. Kangasharju, and J. Ott, "When does content float? characterizing availability of anchored information in opportunistic content sharing," in *IEEE INFOCOM*, 2011.

[16] S. Jain, K. Fall, and R. Patra, "Routing in a delay tolerant network," in *ACM SIGCOMM*, 2004.

[17] F. Ekman, A. Keränen, J. Karvo, and J. Ott, "Working day movement model," in *ACM SIGMOBILE MobilityModels*, 2008.

[18] Q. Inc, "Trepn Profiler - Qualcomm Developer Network," Feb. 2014. [Online]. Available: http://developer.qualcomm.com/mobile-development/increase-app-performance/trepn-profiler

[19] D. Camps-Mur, X. Prez-Costa, and S. Sallent-Ribes, "Designing energy efficient access points with wi-fi direct," *Computer Networks*, vol. 55, pp. 2838 – 2855, 2011.

[20] V. Vukadinović and G. Karlsson, "Spectral Efficiency of Mobility-assisted Podcasting in Cellular Networks," in *Mobile Opportunistic Networking*, 2010.

[21] H. Luo, R. Ramjee, P. Sinha, L. E. Li, and S. Lu, "UCAN: A Unified Cellular and Ad-hoc Network Architecture," in *ACM MobiCom*, 2003.

[22] M. Z. Shafiq, L. Ji, A. X. Liu, J. Pang, S. Venkataraman, and J. Wang, "A First Look at Cellular Network Performance During Crowded Events," in *ACM SIGMETRICS*, 2013.

[23] G. Parisis, V. Sourlas, K. V. Katsaros, W. K. Chai, G. Pavlou, and I. Wakeman, "Efficient content delivery through fountain coding in opportunistic information-centric networks," *Computer Communications*, 2016.

[24] E. Cho, S. A. Myers, and J. Leskovec, "Friendship and Mobility: User Movement in Location-based Social Networks," in *ACM Knowledge Discovery and Data Mining*, 2011.

[25] Q. Xu, J. Erman, A. Gerber, Z. Mao, J. Pang, and S. Venkataraman, "Identifying diverse usage behaviors of smartphone apps," in *ACM ICN*, 2011.

[26] U. Drolia, N. Mickulicz, R. Gandhi, and P. Narasimhan, "Krowd: A key-value store for crowded venues," in *ACM MobiArch*, 2015.

[27] X. Bao, Y. Lin, U. Lee, I. Rimac, and R. R. Choudhury, "Dataspotting: Exploiting naturally clustered mobile devices to offload cellular traffic," in *IEEE INFOCOM*, 2013.

[28] L. Vigneri, T. Spyropoulos, and C. Barakat, "Storage on wheels: Offloading popular contents through a vehicular cloud," in *IEEE WoWMoM*, 2016.

[29] J. Leguay, A. Lindgren, J. Scott, T. Friedman, and J. Crowcroft, "Opportunistic content distribution in an urban setting," in *ACM CHANTS*, 2006.

[30] J. Scott, J. Crowcroft, C. Diot, A. Chaintreau, R. Gass, and P. Hui, "Impact of human mobility on opportunistic forwarding algorithms," *IEEE Transactions on Mobile Computing*, vol. 6, pp. 606–620, 2007.

[31] J. Scott, P. Hui, J. Crowcroft, and C. Diot, "Haggle: A networking architecture designed around mobile users," in *WONS)*, 2006.

[32] N. Thompson, R. Crepaldi, and R. Kravets, "Locus: A location-based data overlay for disruption-tolerant networks," in *ACM CHANTS*, 2010.