Impact of Emerging Social Media Applications on Mobile Networks

Tarik Taleb*, Adlen Ksentini‡
*NEC Europe, Heidelberg, Germany
Email: talebtarik@ieee.org
‡IRISA/University of Rennes 1
Email: adlen.ksentini@irisa.fr

***Abstract*—Emerging social media applications are expected to cause severe congestion to mobile networks, both mobile core network and mobile radio access network. These social media applications are characterized by the fact that they involve sessions with frequently and dynamically updated content, shared with a potential number of mobile users sharing the same location, or being dispersed over a wide area. A method to dynamically identify such applications/sessions and initiate multicast based delivery of the relevant content is proposed. The performance of the proposed method is evaluated through computer simulations, taking Twitter as an example. Encouraging results are obtained.**

I. Introduction

Many web services and applications, and in particular social network platforms (e.g., Twitter and Facebook) or news tickers (e.g. CNN and sport events) are based on a one-to-many communication paradigm, i.e., one entity posts a message of the same content which is then received by many users that have "subscribed" to this "news feed". Other mobile web applications that involve the delivery of the same content to multiple users being in the same location are location-based "check in" services such as Foursquare (1 million users), Facebook places, Gowalla, Brightkite, Yelp, and Google's Latitude. These applications allow users, particularly mobile users, to check in at locations they visit as a way to find other friends, coordinate gatherings and exchange content of common interest among a "social network" of users. There are also many location based instant message services, such as Meebo, that adopt the same concept. There are further many emerging mobile games, allowing users to play a game relevant to their current location and with other users in the same location, e.g., SCVNGR and Zynga, resulting therefore in a frequent and dynamic exchange of content among a group of mobile users in the same neighborhood.

The problem today is that every user establishes a point-to-point communication to the Web server to request the HTML/XML data. While this solution works fine for low-interest information (i.e. where only few users are interested), for high-interest feeds (i.e. information that are followed" by many users in real-time) this solution introduces a significantly high, and above all unnecessarily duplicate load on the mobile network and the Web servers.

The overall objective of this paper is twofold: *i*) to define a mechanism for identifying mobile web applications and services that involve a dynamic and frequent transmission/reception of the same content by a group of users in the same neighborhood and *ii*) based on this data identification to apply policies to the relevant flow. The application and service identification can be done at the end terminals (e.g., User Equipment - UE, server, proxy server, and cache) or at an intermediate node in the network (e.g., gateway and router). The applied policy can, for example, enforce a dynamic offload of the relevant data to a well-designated access network [6], or reroute the relevant data traffic through a proxy server that may dynamically establish a multicast group (based on any available multicasting technology such as Multimedia Broadcast Multicast Service – MBMS [3]) for content that is pushed to many UEs and may trigger the concerned UEs to join the relevant multicast group using one or more suitable multicast technology. Regarding the latter, it should be noted that although the concept of multicasting has been defined already many years ago, there are hardly any applications that leverage this feature.

The remainder of this paper is organized as follows. Section II describes some related work. Section III introduces our proposed solution, which is evaluated in Section VI. Section V concludes the paper.

II. State of The Art

In order to optimize usage of network resources, operators have been looking into solutions that enable them to distribute, to a certain extent, the user plane of their network architecture [7] and to selectively offload IP trafc as near to the edge of operator's network as possible [6]. Other solutions, proposed in the recent literature, that intend coping with the increase in the mobile data traffic and the limited resources at the physical layer, tend towards the direction of using multiple wireless interfaces (e.g., Long Term Evolution and WiFi) and steering traffic among them based on service and application type (e.g., Voice over LTE and YouTube traffic) to ensure acceptable Quality of Service/Quality of Experience (QoS/QoE) to a large number of users sharing the wireless medium [8]. With this regard, in [14], an Android phone-based prototype for making use of all network accesses is described, along with discussion on issues, yet to be solved.

To cope with mobile user-generated traffic, some researches have considered the concept of Delay Tolerant Networks

978-1-4673-3122-7/13/$31.00 ©2013 IEEE

for uploading user's content to mobile networks, designing different delay tolerant forwarding and data transferring algorithms. For example, in [15], it was observed that mobile user generated content delivery is a user-behavioral problem, as most content uploads occur at small number of locations (e.g., users' home or work locations) with significant lag between the content generation time and the content upload time. Based on these observations, it was proposed that mobile user generated content uploads shall happen at selective locations, called drop zones, that are intelligently placed across the cellular network taking into account deployment cost and daily movement patterns of a large number of mobile users.

In [9], usage patterns of mobile data users in large 3G cellular networks were characterized. It was found that most of the mobile users access mobile data services occasionally, whereas only a few of heavy users contribute to a majority of data usage in cellular networks, that is due to usage of a small number of data-intensive mobile applications, video browsing and streaming, and popular social media sites. In [10], flow-level dynamics of cellular traffic are studied, proposing a ZIPF-like model and a Markov model to capture the volume distributions of application traffic and the volume dynamics of aggregate Internet traffic, respectively. The radio resource usage of mobile applications is analyzed in [11] Focus on the impact of the over-the top video on cellular networks is provided in [12]. In [13], the characteristics of cellular http-based traffic are analyzed with respect to a group of applications, namely those related to social, news, and video (e.g., Flickr, Google Videos, WorldPress, YouTube, Blogsport, etc). Many observations were made about the size of wireless sessions, the number of flows per wireless sessions, the packet size used in wireless sessions, and the temporal distribution of demands for mobile services, in comparison to wireline networks. An important observation pertains to the fact that inter-packet gaps differ significantly among different service types, suggesting advanced optimizations such as application-oriented handling of bearer and terminal states, which is in line with the objective of this paper.

III. Envisioned Solution

The main contributions of this paper are twofold, namely identifying applications with frequently updated content sent to many users and enforcing an adequate policy to cope with the congestion that may be caused by the identified applications.

Data identification can take place at a traffic data function (TDF) along the communication path from a UE to the server. Indeed, upon a trigger, UEs issue a HTTP GET request to get a content common to all of them. These HTTP GET requests are intercepted by TDF that can be operating at a dedicated node or collocated with a data anchor gateway (e.g., PDN-GW in case of Evolved Packet System – EPS). If these HTTP GET requests are issued at a frequency higher than a predetermined threshold (i.e., inter-arrival time between two consecutive HTTP GET requests from the same UE is shorter than a certain threshold, and/or the number of HTTP GET requests from the same UE issued during a time interval exceeds a certain threshold), TDF qualifies/identifies the application relevant to the HTTP GET requests as "an application with frequently and dynamically requested content". Alternatively, TDF can also monitor the traffic sent to UEs and identify sessions that are delivered to many UEs and send frequent content updates, based on configurable thresholds.

If the network identifies requests/sessions that lead to the delivery of frequent content updates to many UEs (i.e., to more than a pre-determined number of UEs), the network enforces a suitable policy with regard to the relevant data traffic, such as offloading the relevant data traffic, rerouting the relevant data traffic through a proxy server, which dynamically establishes a multicast group if that content is sent to many UEs, or requesting the concerned UEs to join a relevant multicast group.

Data identification can be also performed at the UE. For this purpose, the UE analyzes *i*) the UE's requests and identifies applications/content that are frequently requested (i.e., at a frequency higher than a threshold – inter-arrival time between two consecutive requests from/for the same applications is shorter on average than a certain threshold and/or the total number of requests from an application issued during a time interval exceeds a certain threshold), and *ii*) the received data traffic. For applications/sessions with frequently updated content, the UE may enforce a suitable policy, provided by the operator, such as offloading the relevant data traffic to a particular access network, using a particular Access Point Name (APN) for receiving/transmitting the relevant data traffic, rerouting the relevant data traffic to/from a different server, and searching for and joining a relevant multicast group.

It shall be noted that the above mentioned thresholds can be dynamically updated, depending on the time of the day and the location. The applied policy could also depend on these thresholds. For instance, if the frequency of the HTTP GET requests exceeds a certain value (Val_1), offload the relevant data traffic to WIFI; if the frequency of the HTTP GET requests is within the range of $[Val_2; Val_1]$, join an adequate multicast group; if the frequency of the HTTP GET request is lower than (Val_2), do nothing.

In the remainder of this paper, we show, once a mobile application is identified as an application with frequently and dynamically updated content (either at the UE or a TDF), how its subsequent relevant data traffic is sent via multicast. For this purpose, the core idea is to extend Web servers/proxies and clients (e.g., browsers) with a functionality (e.g., through a plug-in) that enables efficient delivery of the same Web content requested by many users by Web servers/proxies using multicasting technologies (i.e., IP multicast or 3GPP MBMS [3]) and seamless integration/embedding of the multicast content into normal Web pages/services by the clients (i.e., Web browsers). A Web server/proxy server enhanced with this functionality would be either statically configured or would dynamically decide (e.g., upon receiving many requests for the same content feed) to allocate a multicast address to this content feed and then start multicasting the content (i.e., text, images, etc.) using HTML/XML encoded via UDP (or an alternative multicast transport protocol). It shall be noted that even if a Web server does not directly support this functionality, a Web proxy server (e.g., a cache) in the operator network could provide this functionality. From that point onwards, the Web server/proxy server would respond to any HTTP request for that content with a well-defined Content Type and the Multicast Address, which would, upon arrival at the Web browser, activate the Multicast Browser functionality (or launch the respective browser plug-in). The multicast plug-in would then join the respective Multicast Address/Group and start listening for content message. The content messages received via the multicast channel would then be rendered according to the XML/HTML format in the browser window.

For the Web client/browser to support multicasting, it is enhanced by an internal cache where it stores any content that is received via the multicasting channel. Prior to requesting any missing content objects for the server – as it would do normally – the client/browser checks if the content object has already been received via the multicasting channel. If so, and the cached content is still topical, it will omit the HTTP request to the server and use the cached content. This allows the Web server/proxy server to push embedded content objects (e.g., images or other content types) in addition to normally configured XML/HTML content. If the content is not in the local cache, the Web client/browser could request the content as usual. It shall be noted that to avoid that the client/browser requests any missing content based on a regular HTTP request, the server/proxy server needs to make sure that all embedded contents are delivered prior to the XML/HTML of the page.

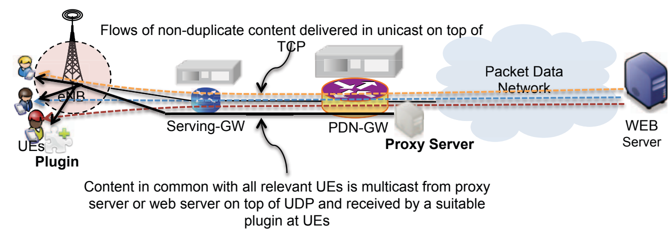

Fig. 1. Envisioned solution, illustrated for 3GPP's Evolved Packet System [4]; content in common with all relevant UEs is multicast from a Web server or proxy server on top of UDP and received by a suitable plugin at UEs.

Fig. 2 shows an example implementation scenario of the proposed solution whereby the server does not belong to the mobile operator or does not run the mobile network's multicast plugin presented here. In the envisioned implementation scenario, a number of UEs send HTTP request messages asking for a particular content from the same server. TDF initially analyses the frequency of these HTTP requests and the number of requesting UEs. If they satisfy particular conditions as explained above, TDF intercepts new HTTP requests to the same server and forwards them to a proxy server owned by the mobile network (e.g., in case the Web server is not owned by the mobile network or the application server does not run the mobile network's plugin for multicasting common content), the proxy server then allocates a multicast address to

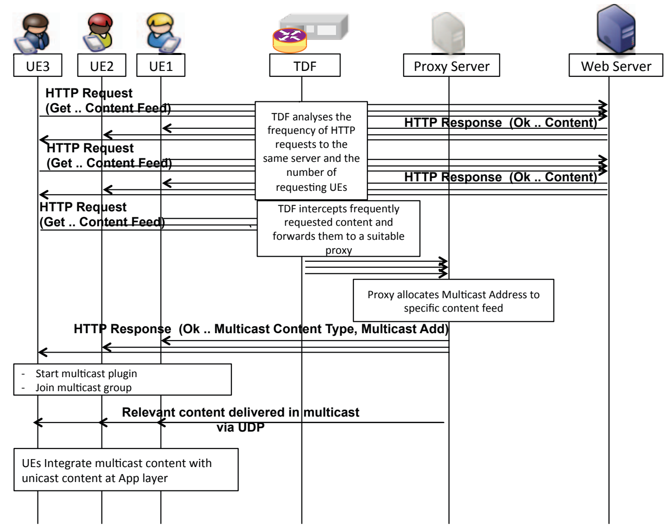

Fig. 2. Detailed message flow of Web-based multicast content delivery in the proposed solution.

the specific content feed, and sends back a HTTP response to the relevant UEs indicating the Multicast Content Type and the Multicast Group Address. To further optimize the transmission over the 3G/4G, the mobile operator GW (e.g. P-GW/GGSN) and/or proxy server could also trigger the establishment of a MBMS session over which the common content can be transmitted efficiently. In this case, the necessary information for the UE to join the MBMS session could also be delivered via the initial HTTP response.

Upon receiving these HTTP responses from the proxy server, UEs launch their multicast plugin and join the multicast group. In the envisioned scenario, the service that the UEs are receiving may have content that is common to the UEs and other part of the content that is specific to each UE. The uncommon content is delivered in unicast using TCP from the server or the proxy server. The content in common is then delivered in multicast using UDP to the UEs. The content could be either received by a proxy server caching the information locally, or a proxy server could request it from the web server and immediately relay it to the UEs in multicast. The UEs integrate the content delivered in multicast and the content delivered in unicast at the application layer.

IV. Performance Evaluation

A. Simulation model and scenarios

In order to evaluate the proposed solution, we used the NS3 simulator. We assume that multicast communications are enabled at the client navigator as well as by the LTE network using the enhanced MBMS (eMBMS) system. Further, we envision a scenario whereby a group of mobile users are connected to the same eNodeB, since they are assumed to be present in the same area. To simulate the social network traffic, we used the model of twitter traffic presented in [5] with slight modifications, to tailor it to our studied case. As in [5], our aim is to derive the number of hits rising after a person twits a tweet message. We consider the same approach:

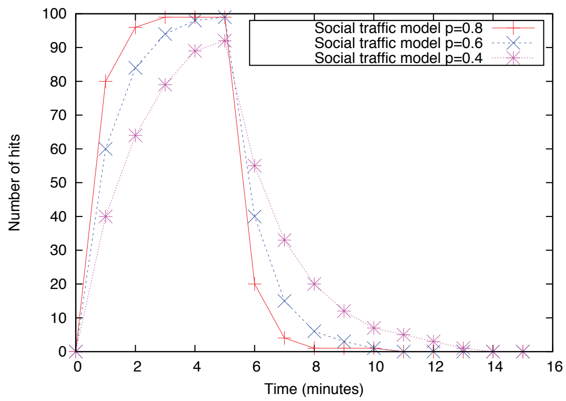

Fig. 3. Traffic model assuming 5% of users re-twitting the tweet message during the first 5 minutes.

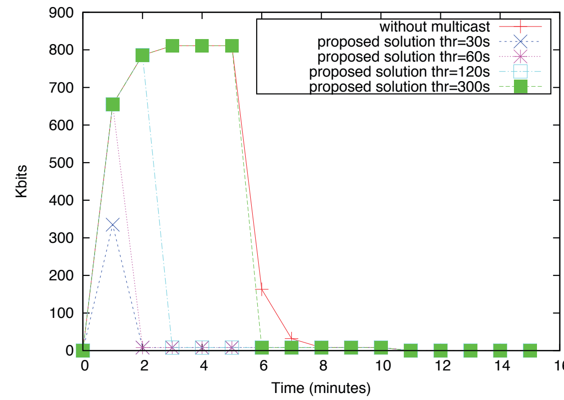

Fig. 4. The amount of data exchanged between the remote server and the group of mobile users when $p = 0.8$

followers are persons that follow the tweets of one user; watchers are some of these followers that are actually watching their twitter stream. During the first minute, we assume that the number of hits is following Equation (1), which depends on the probability that watchers hit the link in the first minute after the tweet.

$$N_i = \frac{w}{p} \quad (1)$$

where N_1 and w denote the number of hits and the number of watchers of the tweet, respectively. p denotes the probability of clicking on the tweet in the first minute. The main difference with the model used in [6] consists in the fact that we consider only watchers, since we want to simulate the case of mobile users being in the same location and sharing content. These watchers will surely hit the link of the tweeted message. This, for instance, can represent the case that a group of persons exchange messages (tweets) in a close area during a specific event. Equation (2) gives the number of hits for the following minutes:

$$N_i = \frac{(w - \sum_{j=1}^{i-1} N_j) * p}{c} \quad (2)$$

where N_i and c denote the number of hits at the ith minute and a factor representing the fact that after some time the probability of clicks decreases, respectively. Since we have modeled the traffic of one tweet, we assume that w' is the number of persons that responded to this tweet within one minute interval, which in fact represents the case that this group of users actively communicates and shares the same content among them. All tweet messages and responses were simulated as echo request/reply messages of a size of 1024 bytes. In the simulations, a static method based on varying thresholds is used for detecting social media applications. Once an application is detected as a frequently requested social media application involving the delivery of the same content among many users, its traffic data start being multicast rather than being delivered in unicast.

B. Simulation results and discussion

Fig. 3 plots the obtained traffic model for different values of p. Here, we consider the following settings: $w = 100$ and $w` = 5$ That is, five persons among 100 respond to the initial tweet within one minute interval. Clearly, from this figure we see that the first tweet generates a certain amount of hits that grow and reach a maximum during the first five minutes; i.e., as five persons respond to the initial tweet by another tweet during the first five minutes. Depending on the probability of clicking on the link, the number of hits reaches 100 between $t = 3$min and $t = 5$min in the case of $p = 0.8$, between $t = 4$min and $t = 5$min in the case of $p = 0.6$, and at $t = 5$min for the case of $p = 0.4$. The number of hits begins to quickly decrease in the case of $p = 0.8$ and more faster for the case of $p = 0.4$. Furthermore, we notice that most of the traffic occur before 5 minutes in the case of $p = 0.8$, whereas the traffic is more distributed during the first 10 minutes and 12 minutes, respectively for $p = 0.6$ and $p = 0.4$. Fig. 4

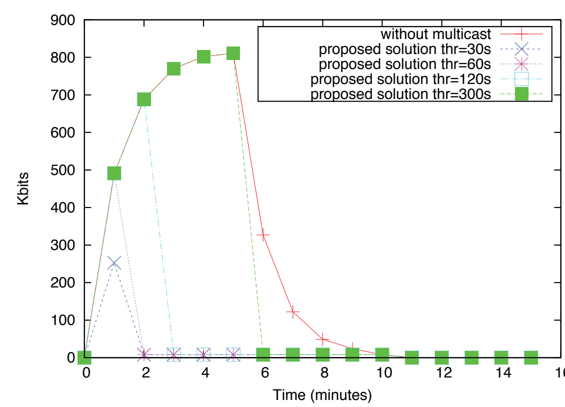

Fig. 5. The amount of data exchanged between the remote server and the group of mobile users when $p = 0.6$.

illustrates the amount of data exchanged between the group of mobile users and the remote server when $p = 0.8$. It compares the case when no multicast adaptation is used (classical solution) and the case of employing the proposed adaptive multicast solution using different adaptation thresholds. These thresholds represent the time taken by the proposed solution to recognize the social network traffic and create a multicast group to deliver the shared content among users. Clearly, we notice that the proposed solution outperforms the classical one in reducing the amount of traffic exchanged between the group of mobile users and the remote server. This good performance is intuitively attributable to the multicast adaptation where the content in common is sent in a multicast fashion, which reduce the amount of traffic exchanged between the users and the remote server, as in comparison to the classical solution where all content is delivered in unicast. Moreover, the latency in detecting the social application traffic has a significant impact on the performance of the proposed solution, particularly in case of $p = 0.8$ as most of the social traffic is exchanged during the first five minutes. So the longer the social traffic detection latency is, the lower the gain is in reducing the exchanged traffic. Thus, when the threshold of adaptation is 5 min, the gain is around 5%, whereas this gain reaches 90% when the threshold of adaptation is 30 seconds. In Fig. 5, we plot the amount of data exchanged by the mobile users and the remote server when $p = 0.6$. Similar to the case of $p = 0.8$, we notice that the best performances are achieved by the proposed solution. However, the main difference consists in the fact that even with a high threshold for detecting the social application traffic, the gain is higher in comparison to the case of p=0.6. This is mainly due to the fact that in case of $p = 0.8$, the traffic is distributed during the first ten minutes. For instance, the gain is 68% in case the detection threshold is equal to 300s, and reaches 91% in case the detection threshold is 60s.

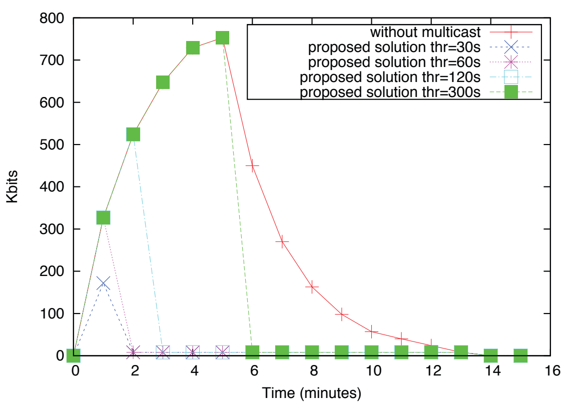

Fig. 6. The amount of data exchanged between the remote server and the group of mobile users when $p = 0.4$.

Fig. 6 plots the amount of data exchanged between the group of mobile users and the remote server when $p = 0.4$. In this case, we notice that the proposed solution achieves considerable gain for practically all the simulated detection threshold values. This is mainly caused by the fact that the traffic is distributed over the first 12 minutes, and even with high detection thresholds the proposed solution can reduce the traffic from $t =5$ min and $t =12$ min. In this case, the gain is 93% and 77% when the detection threshold is 60s and 300s, respectively.

V. Concluding Remarks

This paper provides mechanisms to identify emerging mobile web applications that may waste mobile network's scarce resources. Indeed, the paper introduces an approach for dynamic Traffic Data Identification, as a functionality collocated at a data anchor gateway such as GGSN/P-GW, that identifies an application/session as "an application/session with frequently and dynamically updated content" based on the frequency at which its content or part of its content is delivered to a UE or a set of UEs. The proposed solution also introduces a mechanism whereby if the number of UEs being in the same area/location (e.g., RAN, backhaul, eNB, location area, and routing area) and identified as "with frequently and dynamically updated conten", the network enforces a suitable policy with regard to the relevant data traffic; offloading the relevant data traffic to a particular access network, rerouting the relevant data traffic to/from a different server or via a proxy server, and/or requesting the concerned UEs to join a relevant multicast group. Simulations were conducted considering the case of Twitter and significant gain in terms of core network load reduction was achieved under different simulated scenarios.

References

[1] M.K. Shin, "Extending the World Wide Web for Multicasting an HTML. Document," in Proc. INET'97, Internet Society, June 1997.

[2] R.J. Clark and M.H. Ammar, "Providing Scalable Web Service Using Multicast Delivery," in Proceedings of the IEEE Workshop on Services in Distributed and Networked Environments, 1995.

[3] 3GPP TS 23.246, "Multimedia Broadcast/Multicast Service (MBMS); Architecture and functional description".

[4] 3GPP TS 23.401, "General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access"

[5] "How can you model twitter traffic?" URL: http://www.wired.com/wiredscience/2011/05/how-can-you-model-twitter-traffic

[6] K. Samdanis, T. Taleb, and S. Schmid, "Traffic Offload Enhancements for eUTRAN", in IEEE Communications Surveys and Tutorials journal, Vol. 11, No. 3, Aug. 2012, pp. 884-896.

[7] T. Taleb, K. Samdanis, and F. Filali, "Towards Supporting Highly Mobile Nodes in Decentralized Mobile Operator Networks," in Proc. IEEE ICC 2012, Ottawa, Canada, Jun. 2012.

[8] T. Taleb and A. Ksentini, "QoS/QoE Predictions-based Admission Control for Femto Communications," in Proc. IEEE ICC 2012, Ottawa, Canada, Jun. 2012.

[9] Y. Jin, N. Duffield, A. Gerber, P. Haffner, et. al, "Characterizing Data Usage Patterns in a Large Cellular Network," in Proc. SIGCOMM Workshop, Cellular Networks: Operations, Challenges, and Future Design (CellNet), Helsinki, Finland, Aug. 2012.

[10] M. Z. Shaq, L. Ji, A. X. Liu, and J. Wang, "Characterizing and modeling internet trafc dynamics of cellular devices," in Proc. ACM SIGMETRICS, California, USA, Jun. 2011.

[11] U. K. Paul, A. P. Subramanian, M. M. Buddhikot, and S. R. Das, "Understanding trafc dynamic in cellular data networks," in Proc. IEEE INFOCOM, Shanghai, China, Apr. 2011.

[12] J. Erman, A. Gerber, K. K. Ramadrishnan, S. Sen, and O. Spatscheck, "Over the top video: the gorilla in cellular networks," in Proc. ACM SIGCOMM Internet Measurement Conf. Berlin, Germany, Nov. 2011.

[13] Y. Zhang and A. Arvidsson, "Understanding the Characteristics of Cellular Data Trafc," in Proc. SIGCOMM Workshop, Cellular Networks: Operations, Challenges, and Future Design (CellNet), Helsinki, Finland, Aug. 2012.

[14] K. Yap, T. Huang, et. al, "Making Use of All the Networks Around Us: A Case Study in Android," in Proc. SIGCOMM Workshop, Cellular Networks: Operations, Challenges, and Future Design (CellNet), Helsinki, Finland, Aug. 2012.

[15] I. Trestian, S. Ranjan, A. Kuzmanovic, and A. Nucci, "Taming User-Generated Content in Mobile Networks via Drop Zones," in Proc. IEEE INFOCOM 2011,Shanghai, China, Apr. 2011.