Experimenting with Floating Content in an Office Setting

Shahzad Ali*†, Gianluca Rizzo§, Vincenzo Mancuso*†, Vittorio Cozzolino*, and Marco Ajmone Marsan*‡
{name.surname}@imdea.org, gianluca.rizzo@hevs.ch, ajmone@polito.it
*IMDEA Networks Institute, Spain †University Carlos III of Madrid, Spain
‡Politecnico di Torino, Italy §HES SO Valais, Switzerland

***Abstract*—The growing ubiquity and pervasiveness of sensors and smart devices, with the consequent availability of a large amount of "local" and contextualized information, are giving rise to a wealth of "context-aware" applications and services. In this work we consider an opportunistic communication scheme called "Floating Content" (FC), which was specifically designed for server-less distributed context-aware applications. So far the performance analysis of FC, and in particular of content lifetime and availability, was based on simplified system models and fluid system analysis. The resulting performance estimates did not account for the effects of propagation characteristics, of mobility patterns, and of communication protocols, all factors which are known to affect significantly the performance of opportunistic communication schemes. This paper studies the main issues related to the performance of the FC service in a realistic office setting, through a first experimental evaluation of a *mobile app* for Android devices. Our experimental results confirm the feasibility of the FC service for supporting practical context-aware mobile applications in office settings.**

I. Introduction

The growth of mobile computing, and the pervasiveness of smart user devices is progressively driving applications towards context-awareness, i.e., towards the exploitation of "any information that can be used to characterize the situation of an entity" [1], in order to implement new services that better suit the needs of users. For most of those context-aware applications, the scope of the generated content is limited in time and space. The local relevance of the content generated and exchanged by context-aware applications makes traditional infrastructure-based content distribution services unfit for supporting those applications. In fact, traditional approaches require dedicated servers, which implies a heavily suboptimal use of storage and connectivity resources.

In contrast, opportunistic communications may play a special role when coupled with context-awareness. Indeed, not only they offer a solution to deploy server-less and infrastructure-less content distribution systems, but they also naturally incorporate context and locality, as spatial proximity is closely associated with connectivity. In the present paper we consider a specific opportunistic communication paradigm, known as *Floating Content* (FC) [2], conceived to support server-less and infrastructure-less distributed content storage and sharing. By letting users within a geographical region (hereby called Anchor Zone or AZ) and within a given time interval, replicate the content opportunistically, the FC service aims at ensuring the availability of that content to new users entering the area, by letting it "float" within the AZ.

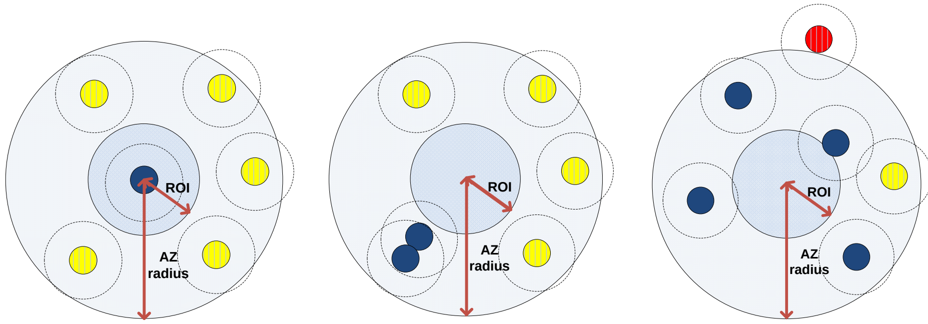
Previous studies of the performance of the FC service focused on determining the conditions under which the content floats with very high probability [2], [3] and on application-level performance modeling [4], [5]. All these works are based on simplifying assumptions for user mobility, and on a simplified model for data exchange. The resulting models give indications about the potential and limitations of FC, but they are only first-order predictions of performance in realistic settings. Indeed, important aspects, such as the combined impact of the actual mobility patterns of users, of the communication protocols, of the specific propagation characteristics in the chosen area, of localization accuracy, are difficult to investigate analytically and to evaluate by simulation. Moreover, it is important to understand the impact of the limitations and of the specific features of available protocols for opportunistic communications (such as Bluetooth and WiFi Direct) on the performance of the FC service. This paper presents a first evaluation of the impact of all those factors on FC performance, through experiments of the FC service in an office setting.

In the rest of this paper we first present the FC service, and discuss the main existing results related to its performance. Then we describe our experimental setting, and our implementation of the FC service over an Android *mobile app*. We present the results of our experiments in an office setting, and discuss the practical feasibility of the FC service for context-aware mobile applications.

II. The Floating Content Service

In this section we concisely describe our system model, and the principles of operation of the FC communication service. We consider wireless nodes moving on a plane, and communicating with each other in ad-hoc mode. We assume that each node periodically issues messages that are of interest only for nodes within the aforementioned AZ, and for a limited time. Moreover, the messages contain relevant information on a restrict area within the AZ, namely the Range of Interest (ROI), as depicted in Fig. 1(a).

Contents are spread opportunistically: whenever a node possessing a content comes within the transmission range of some other node not having such content, the content is replicated, as shown in Fig. 1(b). Eventually, as illustrated



(a) A node generates a content.

(b) The content is replicated opportunistically within the AZ.

(c) Through replication, content is made available in the AZ. Nodes exiting the AZ discard the content.

Fig. 1. Operation of Floating Content service.

in Fig. 1(c), when a node possessing a content moves out of the spatio-temporal limits for that message, it deletes the content, as it is no more of interest for the node [2]. In this way contents "float" within the AZ, i.e., they are available on a set of nodes which move within the AZ. The set of nodes carrying each content varies over time, even after the node which generated the content has left the AZ. We assume nodes have sufficiently large storage resources to host all messages which are of interest for the user. Nodes know their position in space, for instance by GPS or triangulation with base stations and Wi-Fi access points.

Through this geographically constrained opportunistic replication mechanism, a given content is stored probabilistically in a spatial region without any fixed infrastructure, and it is made available to users traversing the AZ through opportunistic exchanges with nodes in the AZ.

Early works on FC focused on the analysis the floating lifetime of the content. Through a fluid model, [2] derived a sufficient condition for the content to "float" with very high probability, based on the average number of nodes in the AZ, on the average node contact rate, and on the average node sojourn time in the AZ. If n is the total number of nodes in the AZ, and μ and ν are respectively the inverse of the mean node sojourn time in the AZ, and the average rate with which any two nodes in the system come in contact with each other (i.e., within transmission range), then the condition for the content to float with very high probability is given by $\frac{n\mu}{\nu} > 1$. This condition has been validated through simulations on a variety of mobility patterns, proving to be actually a *necessary* condition for content to float with high probability. Other works have refined these results, defining sufficient conditions for content to *start* floating with a given probability [6]. All these works show that floating content is feasible even at very low node densities. Nonetheless, for the content availability (defined as the average fraction of users with content inside the AZ) to reach sufficiently high values in the AZ, much tighter constraints hold on node density, AZ size, and transmission radius, than those which hold for content to float.

From the point of view of applications, managing to have the content float is only a necessary condition to achieve satisfactory levels of performance. For applications, an important performance metric is the *success probability* of the FC service, i.e., the probability for a user entering the AZ to receive the content by a given application specific deadline [4]. In order to decouple FC parameters from those strictly related to application level performance, [4] introduces the aforementioned ROI, as a target region used for the computation of success probability. For instance, for some applications the goal is to have all users get the content by the time they leave the ROI. One example can be advertising, when the fact of traversing a given area makes a user likely to be interested in a specific offer/discount. For other applications, users should get the content before they enter the ROI. Examples of such applications can be accident or traffic jam warnings, when a user should be notified in time to take informed decisions about alternative paths. In this last example, the ROI is the area which the user should not enter, in order to avoid getting stuck. For this last class of applications, such a performance target implies an AZ substantially larger than the ROI.

So far, the study of the performance of FC and of the effect that system parameters, such as mobility and node density, have on it, have been based on analytical and/or simulation models. Those works still leave open the issue of determining in realistic settings what are the range of operating conditions (which parts of a city or of a building, what kind of mobility pattern, etc.) in which FC is able to support context-aware applications with a given performance target. In what follows we show the results of a first attempt at addressing these issues, consisting in an experimental evaluation of a FC application in an office setting.

III. Experimental Evaluation of FC

In this section we describe our implementation of the FC service on a smartphone application, and we present our test scenarios for the study of the performance of our *FC mobile app*. The experimental setting we chose is an office (the IMDEA Networks Institute main site, namely INI), composed by a mix of open spaces, meeting rooms and classic, one-room offices (see Fig. 2). For all tests, we chose a ROI coinciding with the AZ in order to simplify the implementation and the assessment of the FC service.

A. *Floaty: a Floating Content Mobile App for Android*

In order to perform our tests, we implemented the floating content service on an Android smartphone application, namely the *Floaty app* [7]. We chose to use Android because of its wide diffusion among existing mobile users. For the exchange of messages among smartphones, we chose to employ Bluetooth, as it is available in almost all the smartphones and it typically consumes much less energy than WiFi [8]. We also verified through simulations that, for a range of node densities which is typical of office settings, and under a variety of mobility models, the Bluetooth transmission range is sufficient to achieve more than 90% success probability. Indeed, the only other technology currently available on smartphones for ad hoc communication is WiFi Direct. However, Bluetooth is currently preferable to WiFi Direct because the latter is only supported by devices running Android version 4.0 or above (although some Android 2.3 devices through proprietary operating system extensions developed by OEMs have this feature). Moreover, currently WiFi Direct poses several technical problems due to the lack of user transparent authentication modes. Finally, the dimensions of the area chosen for the measurements is such that using WiFi Direct for opportunistic message transfers would have created a full mesh network within the anchor zone, preventing us from analyzing the effects of user mobility patterns on FC performance.

Whenever a smartphone running the Floaty app is in the AZ, it generates a new message every T minutes, which is transferred to all hosts in range running the same app. In our experiments we chose T equal to 15 minutes. In order to univocally identify it, every floating content is tagged with an identifier of the phone which generated it (its MAC address and Bluetooth name), with the time at which it has been generated, and with a sequence number.

Every D seconds, Floaty retrieves the list of available Bluetooth peers. For each peer, it checks whether it is running the app as well (the presence of the application running on the device is determined by a Universally Unique Identifier (UUID), broadcasted within the scan response data). Then the Floaty app chooses a peer among those running the app, and it tries to establish a connection to it. If it succeeds, each of the two peers sends all the messages it is currently storing. Each peer then retains only the messages it did not already have. After this, the connection is closed. Due to Bluetooth limitations, Floaty cannot handle simultaneous connections. In our experiments we chose D equal to 1 minute. Since scanning and connecting/disconnecting from a Bluetooth peer takes several seconds, it is important to optimize the choice of the peers, to improve content diffusion in dynamic settings. To this end, the application maintains a list of content exchanges, and prioritizes connections to those peers to which it did not send a given content before. We assumed the AZ to be defined as the area covered by the signal of at least one of a set of reference WiFi access points (APs). In order to understand whether it is in the AZ or not, every P seconds each app retrieves the list of available APs, and it checks the presence of at least one reference AP. To mitigate the effects of instantaneous fluctuations in APs signal intensity on the localization process, the application makes use of a temporal hysteresis. Whenever it finds no reference APs in n consecutive scans, the app assumes to be out of the AZ, and it deletes all stored messages. Similarly, the app considers to be in the AZ whenever it finds at least one reference AP in m consecutive scans. In our tests we verified experimentally that the values $n = m = 2$ and $P = 30$s are sufficient to avoid localization flickering at AZ borders, and to strike a good balance between localization accuracy on one side, and the delay with which an ingress/egress into/from the AZ is registered by the app on the other. The resulting AZ is generally not circular, and it may have holes due to shadow zones.

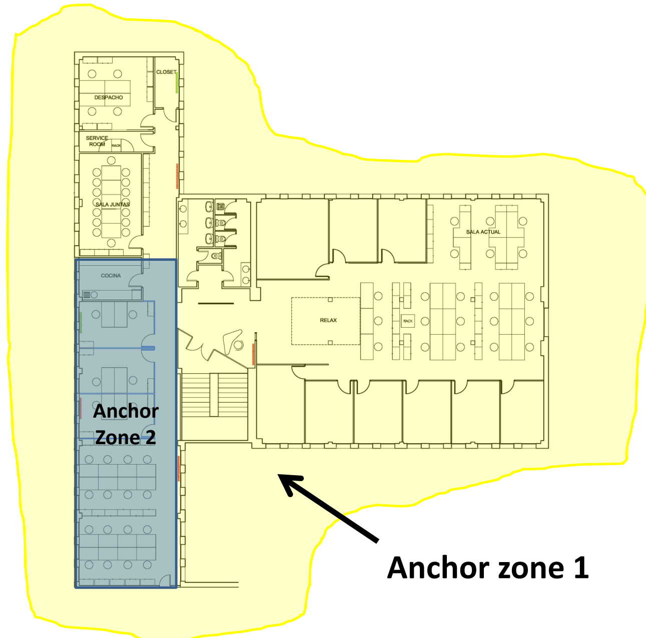


Fig. 2. Planimetry of the site of the experiments.

In order to measure the performance of the floating content service, the application logs all events related to message transfer between hosts, associating them with a timestamp, with a duration of the transfer, and with the list of transferred messages. The application also logs all events relating to AZ ingress and egress, as well as all failed message transfers. These logs are periodically uploaded to a server: whenever the application has some logs to send, it regularly checks for connectivity to the internet (via WiFi or 3G) and, whenever available, it sends the results to the server.

The application does not require interaction with the smartphone's owner. For more details on the app and its implementation, please refer to [9].

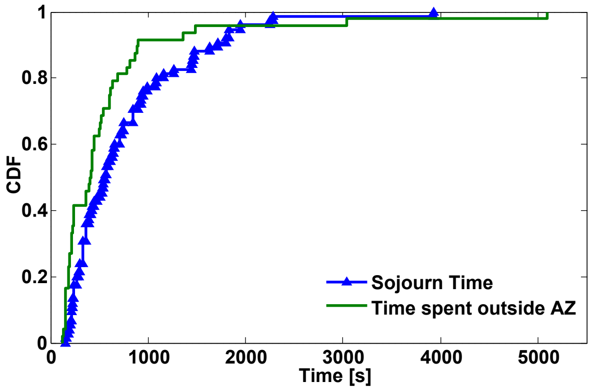


Fig. 3. CDF of sojourn time, and of the time spent outside of the anchor zone.

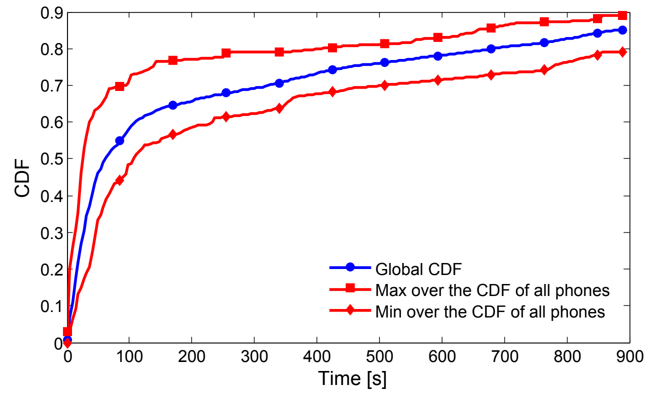


Fig. 4. CDF of the time to get the content (values higher that $900s$ and undelivered contents are not represented in the figure).

B. Impact on utilization of system resources

In a first set of tests, we characterized our app in terms of system resources utilization. Our initial goal indeed was to assess the memory and CPU utilization, as well as energy consumption (and therefore impact on the smartphone autonomy) of our app, while content is being generated and replicated. We installed the Floaty app on 12 smartphones of different brands and with different Android versions. The AZ in this case covered the entire INI premises at the second floor of an office building (Anchor Zone 1 in Fig. 2), for a total area of 800 m^2. This AZ coincided with the area served by all of INI's WiFi access points. Each phone running the Floaty app generated a new message every 15 min, and every minute each instance of the Floaty app searched for Bluetooth hosts running the same app, with which messages could be exchanged. In order to maximize the reproducibility of our tests, no application was active on smartphones other than Floaty, and all screens were kept off. Every test lasted 4 hours. As a baseline, we also measured resources utilization on the same pool of smartphones when the Floaty application is turned off.

We first analyzed the increase in battery consumption due to the Floaty app, while content was generated and exchanged in the AZ. We found that, over all phones, the decrease in battery charge due only to the Floaty app was between 5% and 8% per hour, with an additional 1% consumed by the OS (i.e., the same smartphones, running no app, consume around 1% per hour). Over the same set of tests, we verified that the average total smartphone CPU utilization never goes above 45% and that the memory used by the app never goes above 46 MB. We observed that both these quantities grow with increasing number of message transfers, as it may happen, for instance in settings with higher density of users. Overall, this initial set of tests allowed us to verify that the Floaty app implies a resource utilization level which does not impair the global performance of the smartphone or substantially decrease its battery lifetime.

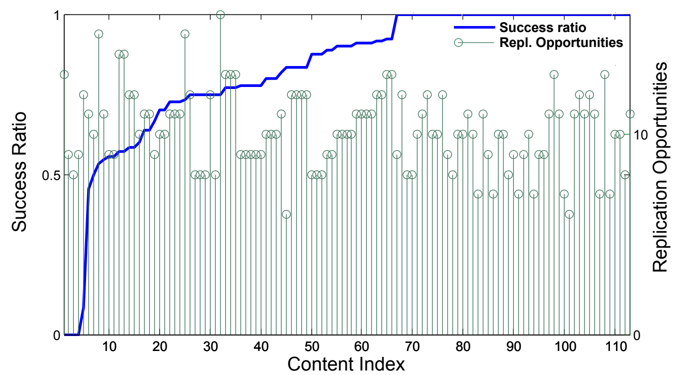


Fig. 5. Content success ratio and number of entries in AZ for each content.

IV. Performance assessment of the FC app

In order to experimentally assess the performance of the FC service in an office setting, we ran the Floaty app on 8 smartphones, and the AZ covered only a portion of the INI premises (Anchor Zone 2 in Fig. 2). The AZ coincided with the area served by just one of INI's WiFi access points. The choice of a smaller AZ with respect to the previous set of tests was made to allow frequent ingresses and egresses of users from the AZ during the experiment. Whenever the app is in the AZ, it generates a new content every 15 min, and every minute each instance of the Floaty app searches for Bluetooth hosts running the same app, with which contents can be exchanged. Conversely, when the app is running outside of the AZ, it does not generate floating contents nor it replicates any content.

Every test lasted 4 hours during office working hours, so that most of the smartphone owners are within the INI premises. As expected, in the chosen application context, users exhibit the typical mobility pattern of office employees, spending long periods of time at their desk, but frequently walking within the premises for meetings, to interact with colleagues, as well as for coffee and lunch breaks, and always carrying their smartphone with them. Several of the owners of the smartphones used in our tests have their seat in the area of the AZ. In order to characterize the mobility pattern of users, in Fig. 3 we plotted the CDF of the smartphone sojourn time within the AZ, as well as of the time spent outside of the anchor zone. We can see how the average amount of time spent in the AZ, as well as the average time spent out of the AZ are of the order of $15-20$ minutes, indicating a rather dynamic setting. On one side, this dynamism might make the spreading of the content faster and more efficient. Nonetheless, it may also make it easier for the content to stop floating and disappear from the AZ.

One factor which contributes to the speed of the content diffusion process is the time necessary for a successful transfer of the content from one host to another. In our tests, 10% of transfers required less than 2 seconds, while for 80% of the contents the transfer time has been within 8 seconds. Given the mobility features of the hosts in the considered setting, these values of transfer time are low enough to have an overall negligeable impact on the performance of the FC service. Indeed, as we have seen, average sojourn times in the AZ are significantly longer than 8 seconds. As we will show, this gives each floating content enough time to reach the vast majority of the hosts in the AZ within its floating time.

An important performance metric in several FC applications is the time taken by a fresh user entering the AZ to get the floating content or, for a user already in the AZ when the content was generated, the time the content takes to reach the user. Especially in dynamic settings, for applications such as emergency warnings, or situated introductions, receiving a new content in a reasonably short time (of the order of few minutes) is essential for an acceptable application performance. In Fig. 4 we plot the CDF of the time to get the content. These plots have been scaled to consider as infinite the time for getting those contents which are not delivered to a given user within the content lifetime (that is, 900 seconds). Therefore these plots also indicate how successfully the content has been transferred during its whole floating time. In our experiments, more than 60% of the contents has been received within two minutes, and around ten minutes have been sufficient to get 80% of the contents in the AZ. These values show that in these settings the performance of the FC service has proven adequate for the majority of the FC applications proposed so far.

Fig. 5 shows, for each content generated during our tests, the success ratio (the fraction of users inside the AZ who got that content during its floating time), and the number of "replication opportunities" during the content floating period (given by the number of hosts present in the AZ at the time of the content generation, plus all those who entered the AZ during its lifetime). In order to characterize the empirical distribution of the success ratio, contents have been ordered according to increasing values of this parameter. We can see that about 80% of the contents have a success probability greater than 75%. In order to check to which extent the differences in content success ratio among hosts might be due to mobility patterns and to fluctuations in the population of hosts in the AZ over time, in Fig. 5 we also show, for each

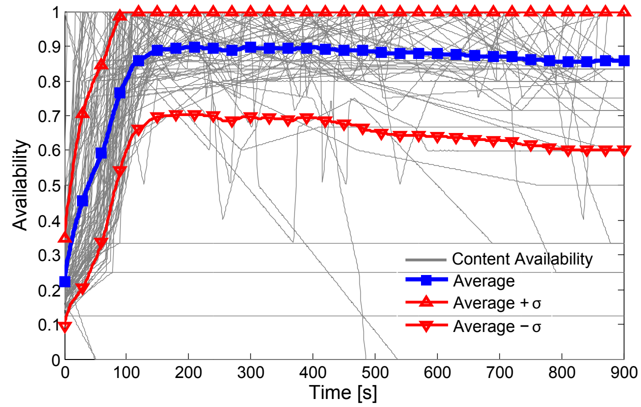


Fig. 6. Content availability over the content lifetime.

content, the number of replications opportunities during its floating lifetime. These plots allows to reject the hypothesis of a correlation between the values of success ratio of a given content measured in our experiments, and replication opportunities of that content. A system parameter which has a direct impact on the probability for a user in the AZ to receive a content in a given time interval is the *availability* of that content during the time interval, i.e., the percentage of hosts in the AZ which, in that time interval, have the content. This is confirmed by Fig. 6, which shows the evolution of average content availability over the floating lifetime. As expected, the variation of content availability over time is clearly correlated with the time taken to get the content seen in Fig. 4, and the time it takes for the average availability to reach its "steady state" value is close to the average time it takes to get a content in the AZ. In Fig. 6 we have also plotted per-content availability over time, and the standard deviation, in order to give an idea of the variability of availability across contents, mainly due to the effect of mobility patterns. We can see that only 3.5% of the contents stops fluctuating before 15 minutes, and half of these contents disappear within the first minute after its generation. In other terms, the vast majority of the contents floated successfully for the whole duration of their lifetime.

V. Conclusions

In this work, we analyzed the performance of floating content service by developing and deploying an Android mobile application in an office environment. To the best of our knowledge, this is the first ever experimental evaluation of floating content service in an office setting. Our results confirm the suitability of floating content as a communication service for context-aware applications which make use of information with restricted geographic and temporal scope.

References

[1] A. Zimmermann, A. Lorenz, and R. Oppermann, "An operational definition of context," in *CONTEXT'07*, 2007, pp. 558–571.

[2] E. Hyytiä, J. Virtamo, P. Lassila, J. Kangasharju, and J. Ott, "When does content float? characterizing availability of anchored information in opportunistic content sharing," in *INFOCOM*, Shanghai, China, Apr. 2011, pp. 3123–3131.

[3] J. Virtamo, E. Hyyti, and P. Lassila, "Criticality condition for information floating with random walk of nodes," *Performance Evaluation*, vol. 70, no. 2, pp. 114 – 123, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0166531612001204

[4] S. Ali, G. Rizzo, B. Rengarajan, and M. Ajmone Marsan, "A simple approximate analysis of floating content for context-aware applications," in *Proceedings of MobiHoc'13*, 2013.

[5] S. Ali, G. Rizzo, M. Ajmone Marsan, and V. Mancuso, "Impact of mobility on the performance of context-aware applications using floating content," in *ICCASA 2013*, November 2013.

[6] M. Desta, E. Hyytia, J. Ott, and J. Kangasharju, "Characterizing content sharing properties for mobile users in open city squares," in *WONS 2013*, April 2013.

[7] *Floaty app*, https://play.google.com/store/apps/details?id=com.vittorio.

[8] P. Serrano, A. De La Oliva, P. Patras, V. Mancuso, and A. Banchs, "Greening wireless communications: Status and future directions," *Comput. Commun.*, vol. 35, no. 14, pp. 1651–1661, Aug. 2012.

[9] V. Cozzolino, "Design and implementation of an android context-aware application based on floating content," Master's thesis, Universita' Federico II, Napoli, 2013. [Online]. Available: http://eprints.networks.imdea.org/id/eprint/607

Biographies

SHAHZAD ALI received his M.Sc. in Telematics Engineering in 2011 from University Carlos III of Madrid, Spain and Masters in Computer Science in 2009 from COMSATS Institute of Information Technology, Pakistan. Currently he is pursuing his Ph.D in Telematics Engineering from University Carlos III of Madrid, Spain. Since October 2010, he has been with IMDEA Networks Institute, Madrid, Spain. His research interests include performance analysis of context-aware applications, wireless sensor networks, vehicular ad hoc networks, and opportunistic networks.

GIANLUCA RIZZO received the degree in Electronic Engineering from Politecnico di Torino, Italy, in 2001. From September 2001 to December 2003, he has been a researcher in Telecom Italia Lab, Torino, Italy. From January 2004, to October 2008, he has been at EPFL Lausanne, where in 2008 he received his PhD in Computer Science. From 2009 to 2013 he has been Staff Researcher at IMDEA Networks Institute in Madrid, Spain. Since April 2013 he is Senior researcher at HES SO Valais, Switzerland. He was the recipient of a 2010 Marie-Curie Amarout Europe Programme fellowship. He is also the recipient of the best paper award at the 23rd International Teletraffic Congress (ITC), 2011, and at the 11th IEEE International Symposium on Network Computing and Applications (NCA 2012). His research interests are in performance evaluation of Computer Networks, and particularly on Network Calculus, and in Green Networking.

VINCENZO MANCUSO obtained his master degree in Electronics from University of Palermo, Italy, in 2001, and a PhD in Electronics, Computer Science and Telecommunications from the same University in 2005. After the PhD, he has collaborated with University of Roma "Tor Vergata" and University of Palermo. He has been visiting scholar at the ECE Department of Rice University, Houston, Texas, and postdoc in the MAESTRO team at INRIA Sophia Antipolis, France. Since September 2010, Vincenzo is with IMDEA Networks Institute, working on analytical and experimental projects on wireless MAC optimization (802.11 and 802.16/LTE) and energy efficient network protocols for green wired (802.3az) and wireless networks (LTE, 802.11, and heterogeneous networks).

VITTORIO COZZOLINO obtained his master degree in Computer Science from University of Naples Federico II, Italy, in 2014. He was with IMDEA Networks Institute from March, 2013 to September, 2013. Since November 2013 he is working as a Software Developer for an IT company located in Turin, Italy, named AgileLab where he is exploiting the potentials of BigData analysis infrastructures tools and also developing mobile Android applications and desktop business-level solutions.

MARCO AJMONE MARSAN earned his graduate degree in Electrical Engineering from the Politecnico di Torino in 1974 and completed his M.Sc. in Electrical Engineering at the University of California at Los Angeles, USA in 1978. In 2002, he was awarded a "Honoris Causa" Ph.D. in Telecommunication Networks from the Budapest University of Technology and Economics. He holds a double appointment as Full Professor at the Department of Electronics and Telecommunications of the Politecnico di Torino, Italy, and Research Professor at IMDEA Networks Institute, Spain. He is also involved in several national and international scientific groups: He was Chair of the Italian Group of Telecommunication Professors (GTTI); the Italian Delegate in the ICT Committee and in the ERC Committee of the EC's 7th Framework Programme. He is a Fellow of the IEEE and he is listed by Thomson-ISI amongst the highly-cited researchers in Computer Science.