Smartphone-based crowdsourcing for network monitoring: opportunities, challenges, and a case study

Adriano Faggiani*†, Enrico Gregori†, Luciano Lenzini*, Valerio Luconi*, Alessio Vecchio*
*Dip. di Ingegneria dell'Informazione, University of Pisa, Pisa, Italy
firstname.lastname@iet.unipi.it
†Istituto di Informatica e Telematica (IIT), Italian National Research Council (CNR), Pisa, Italy
firstname.lastname@iit.cnr.it

***Abstract*—Smartphone-based crowdsourcing fosters the rise of radically novel systems and applications in the context of network monitoring. This paper discusses the most significant opportunities offered by this approach, and the major challenges that have to be faced. Our experience in building a smartphone-based crowdsourcing system, Portolan, is also included to provide a practical background to the discussion and to demonstrate the possible benefits.**

I. Introduction

Measuring and monitoring large-scale networks is a challenging task: due to their always growing size and complexity, collecting information about these, potentially huge, artifacts requires an unprecedented technical effort and significant economical resources. Nevertheless, a deeper understanding of the structure and performance of both wired and wireless networks may be extremely useful for commercial entities, research institutions and even single individuals. For instance, measuring the Internet provides the opportunity to understand the topological properties of the network, associate performance indexes to the graph, and study its evolution. For research purposes, having an annotated graph of the Internet is useful for building accurate models and in turn for the design of next generation protocols and networked applications. The same information is extremely valuable for carriers and service providers, as it enables the definition of more accurate business strategies and tuning of existing systems. Unfortunately, given its current size, measuring the Internet from a limited number of observation points is not sufficient to capture all the details, especially at the fringes of the network where the majority of hosts are located. Similar considerations apply to wireless networks. Cellular network operators may be interested in gathering information about the quality of signal coverage to improve the placement of cellular antennas, but building detailed maps that span whole countries is clearly a daunting task. Mapping the availability of free WiFi access points is useful for end users, or for companies that may want to build a business in providing such information, but also in this case collecting the required information is not practically feasible with a proprietary workforce.

All the previous examples share some common necessities: monitoring activities provide more accurate information when carried out where the end users are; the required amount of work is, in many cases, beyond the possibilities of a single institution; geographic distribution of monitors enables fine-grained measurements. In the last years, crowdsourcing gained momentum as a viable strategy for solving very large-scale problems with the help of the masses (a brief introduction to crowdsourcing is provided in Table I). By outsourcing a task to the crowd, cost-effective network monitoring can be performed at societal scale, using a possibly large number of end users' devices scattered over a wide geographic area. In the rest of the paper we present the opportunities offered by smartphone-based crowdsourcing for monitoring large-scale networks, and the major challenges that have to be faced when adopting such approach. Our experience in building a smartphone-based crowdsourcing system is also reported to provide a practical background to some of the introduced concepts.

II. Opportunities

Here, we advocate the adoption of smartphones as a computing and networking platform for the construction of mobile crowdsourcing applications, highlighting those positive aspects that are peculiar of network monitoring scenarios.

Why crowdsourcing?

The first reason for adopting a crowdsourcing-based solution for network monitoring lies in the power of crowds: by dividing a large task into a set of small and loosely coupled microtasks, the execution of monitoring activities can be parallelized with evident benefits in terms of completion time. Moreover, since monitoring activities are executed on users' hardware, the organization that coordinates the monitoring activity is relieved from the economical and practical burden of managing a dedicated system.

Some network monitoring tools inject packets into the network to infer the properties of interest, e.g. the maximum throughput or the round trip time. If a large amount of measures have to be carried out using a limited number of observation points, the obtained measures could suffer from the observer effect: the measured system is unintentionally altered by the measuring instruments (for instance, if an observation point injects a large number of probes, the network in

TABLE I: Crowdsourcing: main concepts and further reading

The term crowdsourcing has been introduced to denote the process of solving problems with the help of the masses. Differently from outsourcing, where the people hired for performing a service are external to the hiring company but their identity is still relevant, in crowdsourcing the call is open and directed to an undefined audience. Most of crowdsourcing systems are based on the Web, as it provides efficient and inexpensive collaboration tools. Notable examples of crowdsourcing systems include:

1) Amazon's Mechanical Turk[a]: users operate as computing elements to perform those tasks that cannot be easily automated through machines, such as identifying elements within images or transcribing audio clips.
2) OpenStreetMap[b]: users contribute GPS tracks, collected with their personal devices, to build and update a free and open map of the world.
3) InnoCentive[c]: a platform for corporate research where scientific problems are posted by companies to be studied by crowds of possible solvers.
4) The road traffic overlays that can be displayed on Google and Apple maps[d]: a large number of devices report their GPS location and speed information, after being anonymized and encrypted, to the companies' servers where these data are aggregated and then again delivered to map applications on mobile terminals.

The ideas behind crowdsourcing encompass a wide range of applications and are characterized by different business models. For instance, just limiting the analysis to the few examples above, in some cases users are rewarded with money (1 and 3), whereas in other cases the model relies on the contributions of non-professional users (2 and 4). The amount of work to be carried out by the single worker can be significant (as in 3) or very small (1, 2 and 4), thus relying on the aggregate power of large numbers. In the last years smartphones and other portable devices fueled the rise of *crowdsensing*, where participants contribute data produced by the sensors these devices are equipped with (images, audio, pollution level, etc).

Further reading about crowdsourcing:

- J. Howe. *Crowdsourcing: Why the Power of the Crowd Is Driving the Future of Business*. Random House, 2008.
- D.C. Brabham. Crowdsourcing as a Model for Problem Solving: an Introduction and Cases. *Convergence: The International Journal of Research into new Media Technologies*, 14(1), 2008.
- A. Doan, R. Ramakrishnan, A.Y. Halevy. *Crowdsourcing systems on the World-Wide Web*. Communications of the ACM, 54(4), 2011.
- R.K. Ganti, Y. Fan, L. Hui. Mobile crowdsensing: current state and future challenges. IEEE Communications Magazine, 49(11), 2011.

[a] http://www.mturk.com
[b] http://www.openstreetmap.org
[c] http://www.innocentive.com
[d] http://support.apple.com/kb/HT5467, http://googleblog.blogspot.com/2009/08/bright-side-of-sitting-in-traffic.html

its proximity may become congested because of the observer itself). In those circumstances the crowdsourcing approach, based on a large number of small tasks, could reduce such biasing (each node would be responsible of few measures).

Important benefits directly come from shifting network monitoring towards the end systems: networked services and applications can be observed where they are used, and this paves the way to an evaluation of performance metrics from the end user perspective. In [3], the authors demonstrated the effectiveness of monitoring service-level network events through crowdsourcing. In particular, they showed that locally detected events, if properly correlated, can be considered as the symptoms of a widespread network problem.

Why smartphones?

A prerequisite for every successful crowdsourcing-based system is having a large user base, as the power of crowdsourcing directly comes from the number of participants. In the last years, the sales of PCs have been surpassed by the sales of smartphones, whose number is currently in the order of one billion and it is expected to double in the next three years. Thus, although the computing power and networking capabilities of smartphones are considerably below the levels of commodity PCs, it must be noted that not only their absolute values are sufficient for non elementary applications, but also that their aggregate computing and networking power is practically unlimited.

Smartphones are characterized by high mobility, as they are always carried by their owners. As a consequence, they "visit" a possibly large number of wireless networks (both WiFi and cellular) and access the Internet from different entry points. Furthermore, differently from other portable devices (e.g. laptops), smartphones are always on. This eases automation of tasks, such as performing specific actions when entering or leaving a network, or collecting measurements at scheduled times.

At the same time, smartphones present an important characteristic that is unavailable on PCs: they can be easily geolocalized through the embedded GPS unit. The presence of a precise positioning system enables the analysis of networked systems along additional dimensions, since monitoring can take into account the position of end users (which is one of the relevant parameters affecting networking performance). Moreover, whereas in previous work correlation of local events has been based on time and network locality, we believe that the use of smartphones, thanks to their positioning ability, provides the opportunity for more sophisticated studies based on geographical locality as well.

Another aspect where smartphones are more mature than PCs concerns the mechanisms for software installation and upgrade. Distribution of software through the common app stores is extremely simple, and end users are accustomed to the guided process for application download and installation. Moreover, the built-in mechanisms for software upgrade ease and accelerate the development of network monitoring applications: software implementing core functionalities can be subsequently improved with new features or bug fixes. While application stores are now available also on PCs, their use is not so consolidated throughout the user base.

Another positive aspect that originates from the combination of mobility and crowdsourcing is the detection of miscalibrated sensors through rendezvous. As discussed in [9], significant correlation among sensor readings is determined by space-time proximity in the physical world, and we believe that this concept can be extended to network proximity as well. These principles make possible to combine the data generated by nearby smartphones (since, in the end, they operate as sensors), to increase the quality of information. For example, two smartphones located in the same network should provide similar results; if not, one of them is probably operating in the wrong way.

III. Challenges

Architectures based on crowdsourcing are generally more complex with respect to systems where design, implementation, and execution are centralized and/or controlled more strictly. Other difficulties arise from adopting smartphones as the reference platform, e.g. because of their limited resources. Here we discuss the most significant challenges introduced by these factors.

Scarce resources

On smartphones, resources are not as abundant as on desktop- or server-class computing systems. The most severe limitations are a consequence of being battery-operated. Designing an application for a smartphone requires the programmer to be aware of the impact on the device lifetime of every architectural choice or implementation decision. Although this poses a limit on the amount of work that can be assigned to a single device, to not deplete its battery and annoy the user, in crowdsourcing the aggregate power is the result of executing a large amount of small tasks. Therefore, if the application activities are highly parallelizable and parcelable these obstacles can be avoided through a proper division of workload. In general, the activities related to network monitoring are not computationally intensive, thus the energy and processing limitations can be non-stringent if proper architectural countermeasures are put into practice. For instance processing and aggregation of collected data, which are typically heavy activities, can be in some cases off-loaded to the server side.

A more challenging constrain is introduced by communication which, on smartphones, is bandwidth-limited and relatively expensive, especially when access to the network is achieved via cellular connection. From this point of view it is important to notice that network monitoring can rely on passive or active techniques. With passive techniques, limited bandwidth and communication costs are not a problem, since measurements are carried out without sending packets and traffic is generated only to forward collected data to the centralized repository. With active techniques, i.e. those methods that require sending packets, the amount of generated traffic is the main concern to be considered, when evaluating the feasibility of smartphone-based crowdsourcing solutions.

Finally, mobile operating systems can be not particularly suitable for implementing network monitoring mechanisms. In fact, such operating systems are generally optimized to ease the production of application-level software, whereas providing support to low-level networking mechanisms and tools is not their main goal. Moreover, in some cases, mobile operating systems are “closed” environments, where the programmer is confined to a sandbox for security reasons.

Control

The control-plane of crowdsourcing-based applications is complex and subject to additional difficulties with respect to classical applications (even when these latter are distributed). First, crowdsourcing-based systems include humans in the control loop: the position of the device depends on the will of its owner; the device is turned on/off according to non controllable patterns; participation to the network monitoring activities is under control of an external entity. Second, the execution platform is not under a single administrative domain: smartphones belong to their owners and thus it is not possible to exert strict control. For instance, the version of the operating system available on a device is a parameter that cannot be controlled, differently from the case where infrastructure is dedicated and not opportunistically enrolled and aggregated. Finally, the system must be organized to handle large amounts of data. While this may be inherently tied to the size of the problem (if a large network is observed, the amount of produced data can be considerable, independently from the adopted approach) some additional difficulties are introduced by the previous elements. For example, in crowdsourcing-based solutions there is a fraction of duplicate/redundant information: the same monitoring task may have to be assigned to several smartphones, just to be sure that at least one of the involved devices completes the requested duty.

Motivation and incentives

Motivating users to participate in a crowdsourcing application is of paramount importance, as success strongly depends on their volunteer contribution. Previous literature discussed the role of motivation as a key ingredient to reach and maintain a critical mass of users [12]: since contributors carry out small tasks, in many cases they do not directly benefit from their work. Methods to motivate users include money (e.g. Amazon Mechanical Turk (Table I)), altruism (e.g. SETI@Home [1]), entertainment (e.g. ESP Game [16]), and implicit work (users contribute when performing other activities, e.g. reCAPTCHA [17]). In the context of network monitoring, almost all these reasons still hold: a large number of users participated in the network monitoring activity described in [3] as they thought that the problem being solved was socially important; paying may be an option if the cost of enrolling users is economically convenient with respect to acquiring and managing a proprietary network monitoring infrastructure.

Security and privacy

Two different meanings of security have to be considered: protection of the network monitoring system from malicious users, and protection of users' devices. As far as system security is concerned, note that the whole crowdsourcing philosophy originates from the assumption that people are fundamentally honest and that they provide accurate results. This can be an insurmountable problem if network monitoring activities are mission critical, for instance when the safety of people depends on network operation (e.g. military networks).

[1] http://setiathome.berkeley.edu

TABLE II: Crowdsourcing-based systems in networking

Internet characterization and detection of network events

- DIMES is a distributed measurement infrastructure based on the contribution of volunteers and aimed at collecting information about the structure of the Internet [15]. DIMES demonstrates that location diversity of vantage points and a large mass of participants provide significant benefits.
- Dasu is a measurement experimentation platform for the edges of the Internet [14]. To achieve large-scale deployment, it has been implemented as a plug-in for a popular P2P client (the usage pattern of P2P applications is characterized by long sessions, thus it provides wide time windows for collecting data and running experiments). A simple script-like language supports the definition of actions to be executed on remote hosts.
- NEWS (Network Early Warning System) follows an edge-based approach for detecting service-level network events in wired networks through crowdsourcing [1]. Local events, initially detected on end systems, are then corroborated by other possible events occurring in the same region and at the same time. The system has been implemented as a BitTorrent plug-in, to facilitate its adoption through the large user base.

Fingerprinting and georeferencing wireless networks

- The major producers of mobile operating systems use a crowdsourcing-based approach to enhance the localization capabilities of smartphones (in addition to GPS-based localization). With network-based localization, when an application needs to determine the user's location, the OS sends information concerning the currently visible WiFi access points and cellular towers to a server, that compares the received data against a database where network information is mapped to positions; the server then replies with the estimated location. Building such databases requires a huge effort, thus the crowdsourcing approach has been followed: smartphones periodically send to the servers geo-tagged information (anonymized and encrypted) about visible WiFi access points and cellular towers in the nearby; on the server-side this information is used to augment the databases and can be used for future network-based localizations.
- Zee is a crowdsourcing-based system able to annotate the map of an indoor space with the radio frequency fingerprint of the installed WiFi network [13]. Sensors commonly available on users' smartphones (accelerometer, gyroscope, compass) are used to infer their location over time without previous knowledge (such as initial position or device placement).

Assessing the quality of experience

- Quadrant of Euphoria [2] is a platform aimed at facilitating the evaluation of quality of experience in multimedia and network studies. Crowdsourcing is used to enroll the large number of users needed to evaluate the quality of experience with adequate confidence. Consistency rules are adopted to eliminate those users who provide dishonest or incoherent answers.
- In [10] the authors used crowdsourcing to quantify the impact of network problems on the quality of experience for YouTube video streaming. The study demonstrates that a careful design of experiments, consistency controls, and the use of gold standard data (questions whereof the results are known and interspersed within normal tasks) can increase significantly the trustworthiness of the crowdsourcing process.

Network neutrality

- Glasnost [5] is a system that enables Internet users to detect if their ISPs are differentiating a class of traffic from another. Results coming from different users are aggregated to infer ISP behavior.

However, in other scenarios network monitoring is a best-effort activity, and a limited amount of inaccuracies is tolerable. In this case some practices may mitigate the effects of harmful users: for instance a user that consistently provides measures that are not compatible with the ones provided by other users in the nearby may be banned from the system. Recognition of outlier values depends on the measured network properties, but in general the techniques used to recognize miscalibrated sensors can be reused for such purpose.

On the other hand, security of users' devices is not specifically threatened by the crowdsourcing model. The application running on smartphones is subject to the usual controls operated by app stores, thus integrity of devices is preserved as long as code sources are reasonably trusted. Preserving users' privacy is instead more complex, as it happens with applications characterized by user-generated content. Anonymization techniques can help to decouple users' identity from generated data ([8], [4]).

IV. Application scenarios and related work

In this section, we provide some network monitoring scenarios that could benefit from a smartphone-based crowdsourcing approach. Table II summarizes existing crowdsourcing systems, based on both smartphones and ordinary PCs, that are related to networking.

Fingerprinting and georeferencing wireless networks

Fingerprinting and georeferencing wireless network is useful for both cellular and WiFi infrastructure. In fact, areas where cellular signal is very weak or where coverage is not available are still a reality. The problem of monitoring such large networks can be successfully approached with crowdsourcing: an app installed on users' smartphones periodically samples the signal strength and other potentially useful information such as cell ID and network type (GPRS, UMTS, HSPA, etc.). Then, acquired information is georeferenced and forwarded to a server, where data is processed and aggregated. Users are self-incentived to participate in such effort, as they could directly benefit from the results (e.g. to select the carrier that provides best coverage in the area where they live in). Similarly, building a map of WiFi access points can be useful to implement localization systems (indoor and outdoor). Some notable examples are summarized in Table II.

Internet characterization and detection of network events

Most of Internet stakeholders are commercial entities and are therefore reluctant to publicly reveal their network structure. For these reasons, in the last years, a significant amount of research has been devoted to the study of methods for the discovery of the Internet topology. Some passive measurement techniques discover the topology of the Internet at the Autonomous System (AS) level of abstraction by using Border Gateway Protocol (BGP) routing information. However, because of problems such as route aggregation, visibility constraints and hidden sub-optimal paths, the BGP-inferred topology is by nature incomplete. Active techniques, on the contrary, infer the topology of the Internet by relying on tools such as traceroute, and comprise a set of monitors distributed throughout the globe from where traceroute operations are launched. Despite the self-evident disadvantage coming from the necessity of injecting traffic, active methods provide the opportunity to analyze selectively those regions of the network that are not covered with sufficient detail when using passive methods. Table II reports some existing systems based on crowdsourcing, where users' PCs are involved in the monitoring process. We believe that active methods can be pushed further, using smartphones as sources of traceroute probes: smartphones act as network monitors with limited capabilities, but differently from the past they are able to provide different views of the network thanks to their mobility. In fact, during its lifetime, a mobile device may connect to the Internet through access points managed by different ISPs and via cellular connection, obtaining independent measures even when probing the same target. Users can find motivations to participate because of the scientific relevance of the end goal.

Geolocalization of hosts

Localization of Internet hosts is important for both research and industrial reasons. Examples include a detailed understanding of the relationship between topology and geography, and providing services based on location. Currently, two main approaches are followed: in passive methods, geolocation of IP addresses is achieved with the help of administrative registries, where organizations are associated with a position (this technique proved to be rather coarse-grained, especially for very large organizations); in active methods (such as [11]), position of hosts is calculated via trilateration, where the distances from a set of landmarks are inferred by measuring the respective communication delays.

Since geolocation of smartphones is easily determined via GPS, they could be involved as a large set of landmarks in trilateration measures. Motivating users to participate is, in this case, not so trivial, as no direct personal benefit emerges from their contribution. To the best of our knowledge, no geolocalization systems based on crowdsourcing and/or smartphones are currently available.

Network neutrality

Some ISPs differentiate traffic on the base of applications. For instance, peer-to-peer traffic may be subject to a different policy with respect to HTTP traffic, because of its high bandwidth requirements. In other situations traffic is discriminated on the base of routing information (e.g. source or destination AS). This may be done without informing the user, or violating the service level agreements where these differentiations are, generally, not explicitly stated. Similarly, some wireless carriers could be tempted to reduce the connection quality because of competing interests (e.g. VoIP applications).

Smartphone-based crowdsourcing is particularly suitable for detecting violations of network neutrality: a degree of redundancy in collected measures is mandatory to cope with fluctuations originated by congestions and other time-dependent factors; the availability of a large number of network monitors enables analyses from different network positions; through smartphones it is possible to combine in a single platform the neutrality evaluation of both ISPs and wireless carriers.

Users, in this case, are strongly motivated to participate: the results would allow them to detect those ISPs or carriers that are not operating in accordance with contractual specifications. Currently, as far as we know, there are no systems based on smartphones for detecting net neutrality.

V. A case study: Portolan

Starting from these ideas we designed and built Portolan, a crowdsourcing-based system that uses smartphones as mobile measuring elements. Users who participate in the Portolan activities collect a small amount of local measures, then the contributions of a large number of volunteers are assembled to build a detailed map of the network. Currently, Portolan is able to build signal coverage maps and to produce the graph of the Internet at the AS level. Nevertheless, the system has been designed in a modular and flexible way to operate as a general purpose measuring tool (other network properties can be studied by just adding few modules).

The Portolan server coordinates the activity of a large number of clients (Figure 1). The planner of a campaign submits measurement specifications (tasks) to the server. Measurement campaigns are specified through XML documents and possible parameters include the type of measure, duration, involved clients, targets, etc. The server performs some consistency controls and then translates campaigns into a set of small and loosely coupled activities (microtasks) that can be executed by mobile devices. Assignment of microtasks to smartphones takes into account not only their sensing capabilities, but also some time changing properties such as their position or network address. Smartphones run an app that, besides coordination with the server, provides the measurement functionalities, implemented by a number of subsystems. Some measuring subsystems operate actively (e.g. traceroute, RTT, maximum throughput), while others are just passive (e.g. collection of received signal strength). The Android-based client app is available for free on Google Play, the one based on iOS is currently available for internal use only.

Scalability is achieved through peripheral units named *Proxies*. Each Proxy handles a subset of the enrolled mobile devices. Subdivision is performed on a geographical basis, with granularity set at the country level. Each Proxy receives from the server the microtasks addressed to its controlled region and assigns them to the mobile devices. Proxies have been organized by country for two main reasons: *i*) since it is a quasi-static property of clients it eases the implementation of the system; *ii*) it reflects the "local" philosophy of Portolan (clients are responsible of measuring the network that surrounds them). The Proxy Assigner, a module executed on the central unit, manages the assignment of mobile devices to Proxies.

Interaction between smartphones and server

In Portolan, microtasks queued on the server are delivered to mobile devices through a polling mechanism: smartphones, at regular intervals, send a request to the Proxy they have been

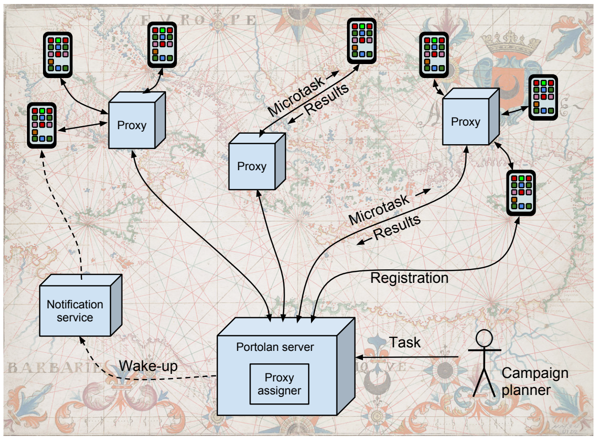

Fig. 1: The Portolan system

assigned to. Requests contain information about smartphones' geo and network position (with geo position we mean the lat-lon coordinates of the device, with network position we mean the IP address of the device and the ID of the cell it belongs to). Proxies use such information to decide whether the devices are suitable for executing microtasks or not. It must be noted that the polling-based solution is somehow unavoidable, since assignment of microtasks to mobile devices depends on dynamical properties known only on the client side. As a positive side effect, since communication is initiated by clients, the difficulties introduced by the possible presence of NAT are avoided (usually telecommunication operators assign smartphones a private address).

To speed up the execution of urgent tasks, we introduced a mechanism that enables out-of-band communication between server and clients. For the Android-based clients we use a notification service provided by Google, the *Google Cloud Messaging* (GCM) service. GCM allows an application running on the fixed network (the Portolan server) to send small messages to an app running on Android devices. The notification service can also be used to dynamically tune polling intervals, in case of significant variations in the number of enrolled devices. During registration each device obtains a pseudo-randomly generated ID, which is subsequently used to identify the single smartphone, or to contact it selectively. On the server-side, the ID is not associated to any personal information to preserve users' privacy and anonymity (more advanced techniques, like [4] or [8], will be added if needed).

VI. Results

In order to demonstrate the effectiveness of the crowdsourcing approach adopted in Portolan, we carried out two parallel experimentations: *i*) a received signal strength (RSS) measurement campaign (cellular network), and *ii*) an Internet mapping campaign. To this purpose, the Portolan app has been distributed to approximately 100 students and faculty members of Italian universities.

Mapping the signal strength of mobile operators

In six months, more than 800000 RSS samples have been collected. A map that displays the collected data is publicly available at the Portolan's website [2]. Figure 2 shows the detailed view of signal quality in the center of Pisa.

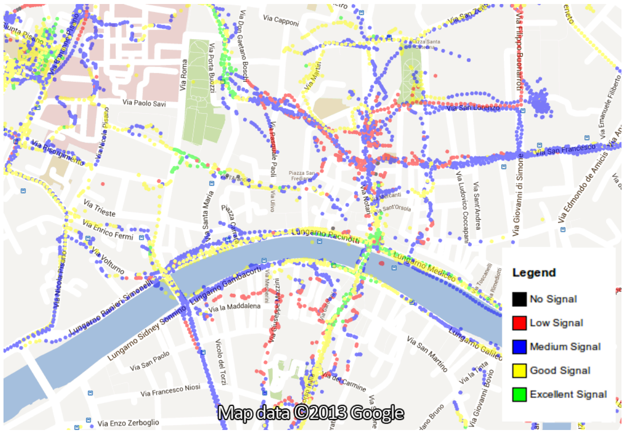

Fig. 2: Signal coverage map (Urban map ©Google. Google and the Google logo are registered trademarks of Google Inc., used with permission).

Some interesting considerations can be drawn through a qualitative analysis of the map. First, it is evident that signal variations can be significant even when users change their position by few dozens meters. We believe that fine-grained information like this can be extremely valuable for telecom operators, as it would allow them a fine tuning of wireless access infrastructure. Second, RSS in a given point changes dramatically with the different operators; as previously mentioned, this makes a detailed map useful for the end users who can select the operator that provides the best quality in those places where they spend the most of their time. Third, in (few) cases the map provides interesting information related to the loss of signal quality when users enter a building: sometimes the GPS unit is able to locate the terminal even when the user is indoor and it is then possible to roughly evaluate the reduction of signal quality due to the building.

To avoid the production of redundant information, the logic running on smartphones generates a new sample only when the position of the device and/or the signal strength change.

Building a graph of the Internet

The Internet mapping campaign was aimed at producing the topology graph of the Italian network structure at the AS-level, as seen by three major Italian ISPs. For this purpose we submitted a traceroute task to Portolan, specifying that smartphones allowed to perform measurements should be located in Italy, under the control of one of the three ISPs. The traceroute target was a large set of IP addresses within the Italian address space. The campaign required the execution of approximately 435 thousands traceroute operations. At the end of the campaign, Portolan had discovered 1117 links at the AS abstraction level. These results have been compared with an analogous dataset produced by CAIDA, a collaborative effort supported by commercial, government, and research organizations aimed at studying the Internet infrastructure [3]

[2] http://portolan.iet.unipi.it

[3] http://www.caida.org

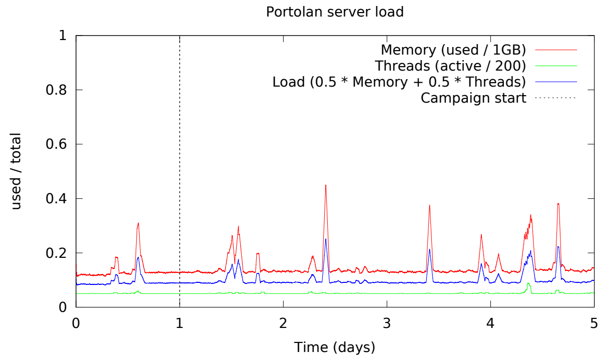

Fig. 3: Load during a campaign

(*IPv4 Routed /24 AS Links* dataset, January 2013). 244 links out of 1117 are unknown to CAIDA (21.8%).

At this time, the "inverse" comparison cannot be performed, because CAIDA's dataset is not limited to Italy. Nevertheless, a preliminary comparison can be derived by restricting CAIDA's dataset: we selected only the links corresponding to ASes located in Italy and involving at least one of the three considered ISPs. We also removed from the Portolan's results all the links where either the source or the destination AS is not located in Italy (the campaign has been started with both sources and targets in Italy, but a large fraction of the AS paths discovered included also some ASes located outside Italy). Of the 333 links discovered by Portolan, 195 were unknown to CAIDA; of the 241 links found by CAIDA, 103 have not been discovered by Portolan. If we consider the union of the two result sets (436 links) as the ground truth (the real set of links is not known), then we can say that Portolan discovered 76.4% of all the links, whereas only 55.3% were known to CAIDA.

The better results obtained by Portolan can be explained by its philosophy based on short range traceroute measurements, which are launched by many observation points densely located in the area of interest [7]. On the contrary, with traditional methods traceroute measurements are started from a limited number of observation points towards destinations spread all over the world.

Resource consumption

In a first version of the system, RSS measurements were remotely triggered by the server through assignment of microtasks. The size of these microtasks was limited to 30 minutes of active GPS time or 10000 samples, to avoid depleting the user's battery or abusing of his/her connection. Such values approximately correspond to 5% consumption of a typical 3000 *mAh* battery and 1MB of traffic (to transfer the samples to the server). Despite the maximum number of 1 microtask per day (for each device), some users complained about an excessive battery drain. Thus, in the current version of the Portolan app, RSS measurements are not remotely started anymore. Now the collection of signal quality samples may take place in two different ways. The user can manually start/stop the collection process, because he/she is interested in a specific location or path (in this case the operation, and the related energy costs, are under the responsibility of the user). The second possibility makes use of the passive location provider provided by Android: since the most of energy consumption is due to the GPS unit, the Portolan app does not actively starts it. Instead if another app, such as a street navigation app, starts the GPS, the positioning information is also reused by the Portolan to geolocalize the signal measurements. In both cases the results are forwarded to the server.

For the Internet mapping campaign the size of microtasks has been set to 100 traceroutes: preliminary evaluations showed that this workload corresponds to approximately 1% of energy consumption in a typical battery and less than 1MB of exchanged traffic. Initially, the number of microtasks was limited to three per day. Then, we noticed that a fraction of such microtasks could not be completed, simply because the device moved to an area without connectivity during task execution. Thus, we replaced the limit on the number of microtasks with a limit on the the global amount of exchanged traffic (2MB per day), while partially completed microtasks can be reassigned to be completed (also by other devices in similar conditions).

We also performed an evaluation of the load on the server-side. All server components (Proxies, Portolan server) are implemented as Java Servlets and are executed on Apache Tomcat. The load on the server, calculated on the base of the number of active threads and used memory, has been monitored for five days using JMeter. We started a traceroute campaign at the beginning of the second day, to use the load of the first day as a baseline. During the campaign, the number of participants fluctuated between 100 and 120. As shown in Figure 3, the average load during the campaign is practically equal to the load exerted on the system during idle periods (the load, averaged on 24h, was equal to 9.3%, 9.9%, 9.8%, 9.7%, 10.8% during the five days). The load spikes visible on the graph are not caused by the core Portolan infrastructure, they are generated by the web application that renders the map of signal strength. Such application (Geoserver), at the moment, is executed on the same Tomcat instance, but in a production environment it can be moved to another machine if needed.

VII. Conclusion

Fine-grained monitoring is still an elusive goal because of the always growing size and complexity of today's networks. Moreover, the increased pervasivity of networked applications acts as a centrifugal force that pushes monitoring to the periphery of the network. In this scenario, crowdsourcing is a possible answer to the raw power needs, whereas the smartphone platform helps to incorporate ubiquity and mobility into the mix. In this paper we identified the major opportunities and challenges of the smartphone-based crowdsourcing approach, and we pointed out some relevant network monitoring applications that could benefit from this model. General concepts have been coupled with a concise description of Portolan's architecture and implementation, to contextualize the discussion within a practical framework (some details have not been here included for the sake of clarity and brevity, and the reader is forwarded to [7] and [6] for further information). The positive experimental results obtained with Portolan confirm the soundness of smartphone-based crowdsourcing.

References

[1] Zachary S. Bischof, John S. Otto, Mario A. Sánchez, John P. Rula, David R. Choffnes, and Fabián E. Bustamante. Crowdsourcing ISP characterization to the network edge. In *Proceedings of the first ACM SIGCOMM workshop on measurements up the stack (W-MUST 11)*, pages 61–66. ACM, 2011.

[2] Kuan-Ta Chen, Chi-Jui Chang, Chen-Chi Wu, Yu-Chun Chang, and Chin-Laung Lei. Quadrant of euphoria: a crowdsourcing platform for qoe assessment. *Network, IEEE*, 24(2):28–35, 2010.

[3] David R. Choffnes, Fabián E. Bustamante, and Zihui Ge. Crowdsourcing service-level network event monitoring. *SIGCOMM Comput. Commun. Rev.*, 41(4):387–398, August 2011.

[4] Cory Cornelius, Apu Kapadia, David Kotz, Dan Peebles, Minho Shin, and Nikos Triandopoulos. AnonySense: privacy-aware people-centric sensing. In *Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services*, MobiSys '08, pages 211–224. ACM, 2008.

[5] Marcel Dischinger, Massimiliano Marcon, Saikat Guha, Krishna P. Gummadi, Ratul Mahajan, and Stefan Saroiu. Glasnost: enabling end users to detect traffic differentiation. In *Proceedings of the 7th USENIX conference on Networked systems design and implementation*, NSDI'10, pages 27–27, Berkeley, CA, USA, 2010. USENIX Association.

[6] Adriano Faggiani, Enrico Gregori, Luciano Lenzini, Simone Mainardi, and Alessio Vecchio. On the feasibility of measuring the Internet through smartphone-based crowdsourcing. In *Proceedings of the 10th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt)*, pages 318–323, May 2012.

[7] Enrico Gregori, Luciano Lenzini, Valerio Luconi, and Alessio Vecchio. Sensing the Internet through crowdsourcing. In *Proceedings of the Second IEEE PerCom Workshop on the Impact of Human Mobility in Pervasive Systems and Applications (PerMoby)*, March 2013.

[8] Marco Gruteser and Dirk Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In *Proceedings of the 1st International Conference on Mobile Systems, Applications and Services*, MobiSys '03, pages 31–42. ACM, 2003.

[9] R.J. Honicky. Understanding and using rendezvous to enhance mobile crowdsourcing applications. *Computer*, 44(6):22 –28, June 2011.

[10] T. Hossfeld, M. Seufert, M. Hirth, T. Zinner, P. Tran-Gia, and R. Schatz. Quantification of YouTube QoE via Crowdsourcing. In *IEEE International Symposium on Multimedia*, pages 494–499, 2011.

[11] S. Laki, P. Matray, P. Haga, T. Sebok, I. Csabai, and G. Vattay. Spotter: a model based active geolocation service. In *Proceeding of IEEE International Conference on Computer Communications (INFOCOM)*, pages 3173 –3181, April 2011.

[12] Alexander J. Quinn and Benjamin B. Bederson. Human computation: a survey and taxonomy of a growing field. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 1403–1412. ACM, 2011.

[13] Anshul Rai, Krishna Kant Chintalapudi, Venkata N. Padmanabhan, and Rijurekha Sen. Zee: zero-effort crowdsourcing for indoor localization. In *Proceedings of the 18th annual international conference on Mobile computing and networking*, Mobicom '12, pages 293–304, New York, NY, USA, 2012. ACM.

[14] Mario A. Sánchez, John S. Otto, Zachary S. Bischof, David R. Choffnes, Fabián E. Bustamante, Balachander Krishnamurthy, and Walter Willinger. Dasu: pushing experiments to the internet's edge. In *Proceedings of the 10th USENIX conference on Networked Systems Design and Implementation*, nsdi'13, pages 487–500, Berkeley, CA, USA, 2013. USENIX Association.

[15] Yuval Shavitt and Eran Shir. DIMES: let the Internet measure itself. *SIGCOMM Comput. Commun. Rev.*, 35:71–74, October 2005.

[16] Luis von Ahn and Laura Dabbish. Labeling images with a computer game. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '04, pages 319–326. ACM, 2004.

[17] Luis von Ahn, Benjamin Maurer, Colin McMillen, David Abraham, and Manuel Blum. reCAPTCHA: Human-Based Character Recognition via Web Security Measures. *Science*, 321(5895):1465–1468, 2008.

Adriano Faggiani Adriano Faggiani received his B.Sc. and M.Sc. in Computer Engineering from the University of Pisa, respectively in 2009 and 2012. Since 2012 he his a PhD student at the University of Pisa and a research assistant at the Institute of Informatics and Telematics of the Italian National Research Council (CNR) in Pisa, Italy. His research interest is in Internet mapping, monitoring and analysis.

Enrico Gregori Enrico Gregori received the laurea degree in electronic engineering from the University of Pisa in 1980. He has contributed to several national and international projects on computer networking. He has authored more than 100 papers in the area of computer networks. His current research interests include Internet measurements and data analysis, ad-hoc networks, sensor networks, wireless LANs, and quality of service in packet-switching networks. He is IEEE member.

Luciano Lenzini Luciano Lenzini holds a degree in Physics from the University of Pisa, Italy. He joined CNUCE, an institute of the Italian National Research Council (CNR) in 1970. In 1994 he joined the Department of Information Engineering of the University of Pisa as a Full Professor. His current research interests include the design and performance evaluation of architectures and protocols for multi-hop wireless networks, and the evolution and structure of Internet.

Valerio Luconi Valerio Luconi received the Master's degree in computer engineering from the University of Pisa, Pisa, Italy in 2012. He is a Ph.D. student in computer engineering at the University of Pisa, Pisa, Italy, under the supervision of Prof. Luciano Lenzini (Department of Information Engineering, University of Pisa) and Enrico Gregori (Institute of Informatics and Telematics, CNR, Pisa). His research interests include Internet topology, network measurement and network monitoring.

Alessio Vecchio Alessio Vecchio received the Laurea degree in computer engineering and the Ph.D. degree in information engineering from the University of Pisa, Italy. Since 2006, he has been a Researcher with the Department of Information Engineering, University of Pisa. His research interests include pervasive and ubiquitous systems, sensor networks, e-health, and bioinformatics. He is IEEE member.