Sensing the Internet through crowdsourcing

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*—Portolan is a crowdsourcing-based system aimed at building an annotated graph of the Internet: the smartphones of participating volunteers are used as mobile monitors to collect measures about the network that surrounds them, then results are conveyed on a central server where they are aggregated. Thus, differently from all the other Internet monitoring systems, which are based on fixed infrastructure, Portolan rely on a multitude of mobile sensing nodes. While this paves the way to the opportunity of having detailed and geo-referenced measures, the design of the systems has to take into account additional difficulties such as scalability, coordination and smartphones' reachability. Besides describing the Portolan's architecture, this paper also shows some preliminary results that confirm the validity of the proposed approach.**

I. Introduction and motivation

The Internet evolved from a nucleus of few academic and government networks into a global communication infrastructure. This massive system is now composed of thousands of service providers belonging to different business categories, e.g. regional or international transit providers, content providers, enterprise and academic networks, access providers, and content distribution networks. Internet Service Providers (ISPs) typically operate as commercial entities and are reluctant to publicly reveal their network structure and properties. Therefore, research focusing on methods for the discovery of the Internet topology gained momentum in the last years [17], [6], [7], as a deeper understanding of the Internet graph would help researchers and practitioners to design new protocols and networks, or to improve the efficiency and scalability of existing systems.

A. Existing measurement systems

Measurement methods are typically classified as *passive* or *active*.

Passive measurement methods are commonly used to discover the topology of the Internet at the Autonomous System (AS) level. There are two main sources of AS-level topology data: BGP routing information [19] and Internet Registries. The main advantages of using BGP information, captured from path announcements, are: i) data can be easily collected at specific places and there is no need to deploy a dedicated infrastructure; ii) information is up-to-date and reflects the current state of the network. However, using BGP routing information has several drawbacks. First, the set of relationships between ASes discovered through BGP is not complete, especially for peering links. Further, the publicly available BGP paths do not cover the entire Internet because of issues such as visibility constraints, route aggregation, hidden sub-optimal paths and policy filtering [16]. Positive aspects of the approaches based on Internet registry information include ease of implementation and efficiency since, also in this case, direct exploration of the network is not required as information is already available at specific locations. In addition, other valuable information such as routing policies can be easily extracted. On the negative side registry information is not always complete (e.g. because of privacy reasons) or, in other cases, it may contain redundant or incoherent data. Examples of research efforts based on passive methods include the Routing Information Service, maintained by RIPE [20], RouteViews [21], and the Packet Clearing House [18].

Complementary to the passive approaches, active methods generally refer to techniques that inject specially crafted probes into the Internet in order to infer its topology. The most commonly used measurement methods are based on traceroute or its variations, such as Paris Traceroute [2] or MDA [25] (the latter two tools are able to discover the correct paths also in the presence of load-balancing routers). Two relevant research projects based on active measurements are CAIDA Archipelago [5] and DIMES [22]. The self-evident disadvantage of active methods is the necessity of injecting a large amount of packets into the network to operate. Nevertheless, they provide the unique opportunity of analyzing those "obscure" parts of the Internet that cannot be explored through passive methods, because, for example, ISPs are reluctant to make public information about their internal structure.

B. Motivation

Measuring the Internet, with active methods, from a set of fixed observation points (also called monitors) can still lead to unsatisfactory results: since such observation points are usually placed in proximity of the core of the Internet they are unable to provide detailed information about the fringes of the network (this is particularly true for peering links). To overcome this limitations we designed Portolan[1]: a crowdsourcing-based system where smartphones play the role of mobile measuring elements [9]. Each smartphone collects a number of local measures that are subsequently forwarded to a server, where they are assembled to generate a global map of the Internet. Mobility of nodes enables each single monitor to have different perspectives of the network, thus obtaining more detailed information. Moreover, the possibility of geo-locating

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

978-1-4244-9529-0/13/$31.00 ©2013 IEEE

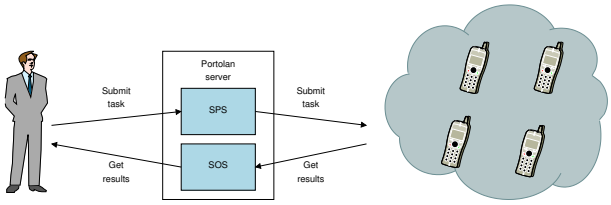


Fig. 1. Overview of the Portolan system

measures through the smartphone GPS adds new dimensions to the analysis of the Internet.

In this paper we present the design and implementation of Portolan and we provide some preliminary results that confirm the soundness of this approach. System description focuses on the control and coordination plane: each global measurement campaign is translated into a possibly large set of smaller tasks that are assigned to mobile devices; assignment of tasks to mobile devices is subject to efficiency and scalability constraints and must take into account some runtime properties of mobile devices, such as their position, load, and connectivity. Since these problems are common to the majority of crowdsourcing-based systems we believe that the lessons learned can be re-used as useful guidelines by other implementers, even in a different application domain.

II. System overview

The system is composed by a large number of mobile devices coordinated by a central unit. The mobile devices are responsible for carrying out the raw measurements according to their geographical and network positions, while the central unit (the Portolan server) assigns tasks to mobile devices, collects the results, and performs data processing if needed. The Portolan server receives measurement campaign specifications from human users, and translates them into smaller jobs which will be executed by mobile devices. Interfaces for tasking sensors and collecting results have been defined by using the standards specified in the Sensor Web Enablement (SWE) [4] framework.

SWE is an initiative of the Open Geospatial Consortium [15] that puts together open standards for web accessible sensors. The functionalities of the SWE framework include the discovery of sensor systems, determination of a sensor's capabilities and quality of measurement, geo-localization of observations, storage and retrieval of observations in standard encodings, tasking of sensors for the collection of new measurements. While SWE is generally used for providing access to physical sensors [3], [10], [8], [24] (such as air pollution, temperature, water levels, etc) nothing prevents its adoption in a network monitoring and analysis system. SWE includes the following specifications: *Observation & Measurements* (O&M), a set of models for encoding observations produced by sensors; *Sensor Model Language* (SensorML), a specification that provides support for the discovery of sensors and the description of their characteristics; *Sensor Observation Service* (SOS), a standard web interface for retrieving observations and information about sensors; *Sensor Planning Service* (SPS), a standard web interface for requesting new acquisitions to sensors.

Mobile devices operate as monitors and can be seen as geo-localized sensors able to measure some network-related properties. Mobile devices run an application that, besides coordination with the server, provides the measurement functionalities. More in detail, the application contains a number of measuring subsystems, one for each network property (e.g. route between two endpoints, available bandwidth, round trip time, signal strength, etc.). Each subsystem is described using the SensorML specification. The system also supports the presence of devices with heterogeneous characteristics in terms of sensing capabilities.

The overall system is shown in Figure 1. A user who wants to start a measure campaign has to prepare an XML document and submit it to the Portolan server (in particular, the document is handled by the SPS component). The XML document describes the characteristics of the campaign: type of measure, duration, involved clients, targets, etc. The SPS performs some consistency controls and then decomposes the campaign into a set of small tasks that can be assigned to clients. Mapping between tasks and clients can be done according to a number of different criteria, including their position, country, or network address. Tasks are then assigned to the clients, which perform the requested measures. Once finished, they upload the results on the Portolan server where they are transferred onto persistent storage. Finally, the user can retrieve the results by querying the SOS component of the Portolan server. Queries are expressed again in the form of XML documents and allow the user the selection of results on the base of geographical properties, time of acquisition, measure of interest, and the like.

Currently, we have a complete implementation of the client app for the Android platform while the iOS version includes, at the moment, only a subset of the measuring subsystems. The implementation of tracerouting mechanisms for an unprivileged Android environment is described in [9].

III. Task specification

Portolan, as mentioned, is currently able of executing campaigns aimed at discovering routers, round trip time, and available bandwidth of end to end path. The system is also able to measure the received signal strength of mobile networks. Since we plan to add to the system the possibility of measuring also other properties, we designed the task specification to be as general as possible. From now on the term *task* will be used to mean a measurement campaign specified by a human user of Portolan system. The term *microtask* will be used to specify the part of a task that is assigned to a single mobile device. A task is made of six sections:

- the *operation* to be performed, i.e the desired type of measurements (e.g., traceroute or bandwidth measurement);
- a *source identification* part: information needed to identify which mobile devices can perform requested measurements;
- a *destination identification* part: information needed to identify measurement targets (e.g., for traceroute, the addresses of the IP interfaces to be probed);

- the *duration*, i.e. the maximum time that should pass before a task can be considered as finished (even if the execution of all microtasks has not been completed);
- a set of *operation specific parameters*: additional information for a specific operation (e.g., hop limit for traceroute operation);
- an *urgent flag*: for tasks to which must be given precedence.

Some of the previous sections are made of a single value, such as *operation*, *duration*, and *urgent flag*, while others are composite. The *source identification* section allows the human operator to specify the country of the mobile terminal, a geographical area (where the terminal must be located), the autonomous system where the mobile device resides, the network type (e.g., wifi or cellular), the mobile network operator that provides connectivity, and the mobile device ID (e.g., IMEI or MEID).

A microtask is assigned to a mobile device that satisfies all the specified requirements. The only mandatory field of this section is the country field, since currently mobile devices coordination is made on a geographic base with granularity at country level. All other fields are optional.

The *destination identification* section is operation dependent, as target specification depends on the type of requested measurement. For the *traceroute* operation targets are IP addresses, and they can be identified in two ways: explicitly or implicitly. In implicit mode, targets are identified by specifying the destination country, the geographic area of interest, and the ASes the targets belong to. None of the three fields is mandatory. If more fields are present, the target addresses are those that satisfy all the conditions.

The content of the *operation specific parameters* depends on the selected operation. Some of the parameters that can be configured for the traceroute operation include the hop limit (the maximum distance that can be reached before stopping the measurement) and a black list (a traceroute operation is stopped when one of the specified IP addresses is discovered).

IV. Architecture and implementation

Mobility of devices and their access to the Internet via cellular networks adds complexity to the design of an efficient measurement infrastructure. First, mobile devices are not always available: they might reside in an area without connectivity, they might be switched off or, simply, the Portolan application might be closed by the user. Second, smartphones often reside in private networks and thus they are identified by a private IP address. Access to the Internet is achieved via a gateway running a NAT server which maps multiple private IP addresses to one public IP address [23]. Therefore, a mobile device is unreachable from the outside unless a persistent connection is maintained between the server and the device. Such a connection would cause consumption of resources both on the mobile device, in terms of energy, and on the server, in terms of OS resources. Finally, since the system leverages the crowdsourcing principles, the server infrastructure of the Portolan system has to be scalable, to handle a possibly large number of mobile devices without performance degradation.

A. Portolan architecture

In order to fulfill the previous requirements and overcome the connectivity limitations, we designed a system architecture that extends the basic model shown in Figure 1. The architecture of the implemented system is depicted in Figure 2 and comprises a central unit that includes a SPS, a SOS and a *Proxy Assigner*. Other decentralized units named *Proxies* have been introduced to achieve scalability.

The SPS and the SOS act as a front end, hiding the system's complexity from the human user. The SPS is responsible for receiving the specifications of measurement campaigns (tasks) from human users and translates them into sets of microtasks that will be executed by mobile devices. The SOS receives from mobile devices the results of microtask execution and makes them available for visualization or further processing.

Each Proxy handles the subset of mobile devices located in a given geographic area. Currently, granularity is set at the country level, thus each Proxy is responsible for handling the mobile devices residing within one or more countries (e.g. a Proxy may control one country with a large number of mobile devices, or a few countries if they contain a small number of devices). Therefore, scalability is achieved by splitting the computational and communication load over multiple distributed and decentralized units. Each Proxy receives from the SPS the microtasks addressed to its controlled countries and assigns them to the mobile devices. Proxies have been organized by country for two main reasons: i) since it is a static property of clients (it does not change with time like, for instance, clients' position) it eases the implementation of the system and does not require communication between clients and server; ii) it reflects the "local" philosophy of Portolan (clients are responsible for measuring the network that surrounds them). In case of excessive load for a single Proxy, other Proxies can be arranged in cascade also using other properties as assignment criteria (e.g. clients' IDs).

Delivery of microtasks to mobile devices is based on a polling mechanism: each mobile device, at regular intervals, sends a request message to the Proxy it has been assigned to, and obtains a microtask, if any. The use of polling, where communication is initiated by clients, solves both the problem of availability of clients and the difficulties introduced by NAT translation: unavailable devices simply do not poll the Proxy (they are not considered as participating to the Portolan system), and there is no need to reach the mobile devices from the outside of their private networks, as it's up to them to connect to the Proxy for receiving microtasks. At each poll the mobile device sends to the Proxy a message containing information about its geo and network position. The Proxy uses such information to decide whether the device is suitable for executing a microtask (i.e. if it matches the source identification criteria of some microtask). If so, the Proxy assigns the microtask to the mobile device. Otherwise, the Proxy communicates to the mobile device that there are no microtasks available at the moment. Polling is usually considered as a solution characterized by possible inefficiencies, since the polling component consumes resources even when there is nothing to do. However, it must be noted that in this case

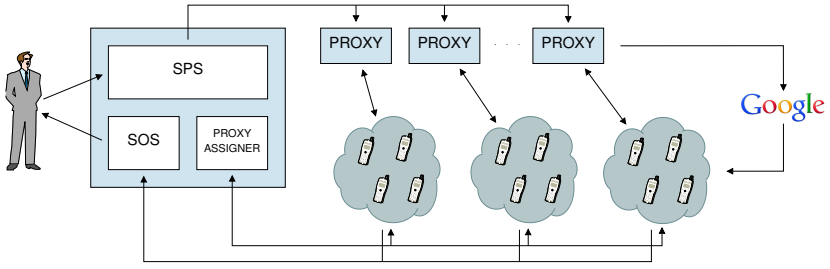


Fig. 2. Portolan system architecture

this kind of solution is somehow unavoidable: assignment of microtasks to mobile devices depends on their current position and network address; but these properties are dynamical and are known only on the client side. Thus, the mobile device is forced to transfer this information on the server side where it is used to select the possible microtask.

The Proxy Assigner module is responsible for assigning mobile devices to a given Proxy. When the app running on the mobile devices is started, it connects to the Proxy Assigner whose URL is fixed and well-known. The app sends the code of the country where it resides and the Proxy Assigner replies with the URL of the Proxy controlling that country.

B. Enhancements

A task may be flagged as urgent, and in such case the system must give priority to its execution. In order to speed up the delivery of such tasks, we introduced a mechanism that enables out-of-band communication between the server and the clients. For the Android-based clients we use the *Google Cloud Messaging (GCM)* service. GCM allows an application running on the fixed network (e.g., the Portolan server) to send small text messages to an app running on Android devices, even if the app is not currently in execution. Use of the GCM service requires the Portolan server to store an API key and the registration ID that corresponds to the application running on the client side.

When a Proxy receives an urgent task from the SPS, it sends via GCM service a short message to the involved mobile devices, notifying the arrival of an urgent task. All mobile devices that receive the message, after a short random amount of time, send a request to the Proxy without waiting the end of the polling interval. The Proxy then responds to poll requests as previously described. This mechanism allows to keep a low polling frequency when the tasks to execute are not urgent, and, on the other hand, when an urgent task is submitted, to assign the generated microtasks as quickly as possible.

The GCM service is also used to dynamically tune polling intervals. When a Proxy is polled by a large number of mobile devices, keeping a high poll rate could lead to congestion and decreased performance. Instead, when a small number of mobile devices are polling for microtasks, it would be desirable to increase the poll frequency to keep the system responsive. This is made possible through the GCM service: when the number of mobile devices grows over or decreases below a certain threshold, the Proxy sends a message via GCM to communicate the new poll rate.

The GCM service has not been used also for assigning microtasks to mobile devices, instead of the polling based solution, because matching between clients and microtasks depends on runtime and dynamical properties, as previously highlighted.

A similar service, Push Notification, is provided by Apple on iOS systems.

C. Implementation

An implementation of the SPS and SOS components, in form of web services, is provided by the 52North consortium [1]. The SPS has been extended with a plugin implementing the task management and subdivision functionality. The SOS has been customized to handle the Portolan data, with new XML documents describing the Portolan sensors (i.e., mobile devices) and the Portolan measurement capabilities. On the contrary, the Proxy Assigner and the Proxy components have been implemented from scratch as web services using the Java Servlet technology. This ensures ease of communication with mobile clients since the HTTP protocol is extensively supported by all mobile OSes.

Particular attention has been dedicated to the translation process from a task to microtasks. This operation, as mentioned, is performed by the SPS module, after having checked the consistency of the task parameters submitted by human users. The task sections involved in the translation process are the *destination identification* and the *operation specific parameters*. The other sections are simply copied from the task to each generated microtask. The translation process is divided in two phases: i) the identification of a set of measurement targets based on the destination identification and operation specific parameters; ii) the division of the targets in blocks with smaller size. Notice that the target identification phase is dependent on the measurement type (identified by the operation parameter), therefore separate logic (i.e. a separate Java class) has to be defined and implemented for each type of measure.

For example, the traceroute operation translation process operates as follows. The measurement targets are IP addresses. If targets are specified explicitly, the system just splits them in small sets that a mobile device can probe in a reasonable amount of time and with limited battery consumption (our tests show that a mobile device can execute one hundred traceroutes in approximately 10-15 minutes with less than 1% battery consumption [9]). If targets are implicitly specified, the system performs the translation of the destination country, geographic

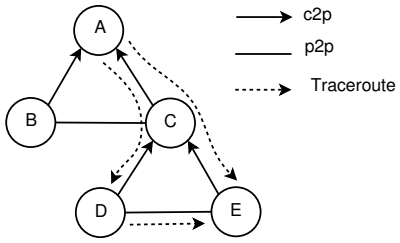


Fig. 3. Probing the Internet: top-down vs bottom to bottom

area and AS list into a set of IP addresses. The IP addresses are retrieved from a database created by merging MaxMind *GeoLite Country* and *GeoLite City* databases for IP addresses geolocation [13] and Isolario *Prefix_AS* dataset, which pairs IP subnet prefixes to AS numbers [12]. The retrieved set is then divided into smaller sets as described above.

V. Sensing the Internet

We carried out a measurement campaign to validate the system functionality, and to assess the soundness of an approach based on local measures. In particular, the experiment was aimed at discovering the public peering links between a specific AS (source) and a set of candidate peers (targets).

Interconnections between ASes are driven by economic relationships, which are classified into customer-to-provider (c2p), provider-to-customer (p2c), peer-to-peer (p2p) and sibling-to-sibling (s2s) [11]. In c2p and p2c, an AS (customer) pays another AS (provider) to obtain connectivity to the rest of the Internet. In s2s agreements, a pair of ASes (siblings) provide each other with connectivity to the rest of the Internet. Finally, in p2p, a pair of ASes (peers) agree to exchange traffic between their respective customers. Two ways of peering are possible: i) private peering, the two ASes connect their networks directly with a physical link and split the costs of operating this link; ii) public peering, both ASes connect a physical link to the same Internet eXchange Point (IXP). An IXP is a physical location that allows multiple ASes to exchange their traffic through its switching infrastructure. AS members of an IXP typically share the IXP operation costs [14, Chap. 5].

The most of the unknown links belong to the p2p category. Unfortunately, these links cannot be discovered when the monitoring system is in proximity of the top of the AS hierarchy. Consider for instance the scenario depicted in Figure 3: if the traceroute probes are sent from the autonomous system A towards the autonomous systems D and E, the p2p link between D and E cannot be discovered. To discover such link, the traceroute operation has to be started from one of the two ASes (i.e., D or E) and it must be directed towards the other (from D to E as shown in the figure, or vice-versa). The same happens also for the p2p link between B and C. This peculiarities of the graph of the Internet at the AS level motivates the Portolan approach: instead of probing the Internet from the top to the bottom, the graph can be discovered by using a large number of horizontal measures.

A. Experiment setup

The test was carried out using a single mobile device. However, in the near future, we plan to conduct further experiments by recruiting a larger number of volunteers, in order to exhaustively test the crowdsourcing approach.

The selected source AS has been *Registro.it*, which is the organization responsible for assigning the country code top level domain (ccTLD) for Italy. Its AS number is 2597 (from now on we will refer to it as AS2597). AS2597 is connected to the Milan Internet eXchange[2] (MIX), an IXP that provides connectivity services to Italian and international ISPs and carriers. The targets of traceroute have been chosen from the address spaces of 79 Italian ASes connected to the MIX. The list of ASes connected to the MIX is publicly available on the MIX website. The measurements have been performed by an Android device connected to a Wi-Fi network belonging to AS2597.

In order to state that AS2597 is connected to an AS with a public peering link via the MIX IXP, the traceroute must contain the following three consecutive hops:

1) one IP address belonging to AS2597 address space,
2) one IP address belonging to MIX IXP address space (its AS number is 16004),
3) one IP address belonging to the target AS address space.

Vice-versa, if an address belonging to the MIX is not found, this means that the two ASes are connected with a direct link (which could be a private peering link, a p2c/c2p link or a s2s link) without passing through the IXP.

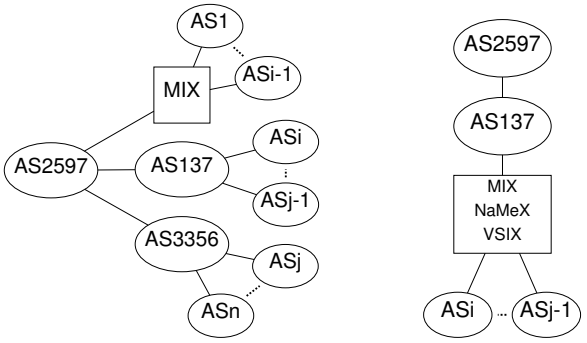
The Portolan app for the Android OS implements the traceroute tool, but provides as result only the list of traversed IP interfaces and not the ASes they belong to. Therefore, on the server side, we used the Isolario [12] *Prefix_AS* dataset to map IP subnet prefixes to AS numbers. However, such dataset contains some unreliable entries, where a single IP subnet prefix is mapped to multiple ASes. In such cases we used the Whois protocol to retrieve the correct AS an IP address belongs to.

B. Validation

The topology discovered through the experiment is shown in Figure 4(a). We discovered that the probed ASes could be reached both by public peering links via MIX IXP and via two AS2597 providers: Consortium GARR (AS137), which is the Italian network that connects universities and research institutions, and Level3 (AS3356), which is a big tier-1 ISP.

Measurements showed that within the set of probed ASes, 22 of them are reached with a link via MIX IXP. Only a single AS could not be reached, as no route to its networks was available in AS2597 routing tables (as it has been confirmed by the AS2597 network administrator).

By comparing the links found by our measurement system with the list of links provided by AS2597 network administrator, we can state that the Portolan measurement system has been able to find *all the existing links*.



(a) Paths from AS2597 to target ASes

(b) AS137 links indirectly discovered

Fig. 4. AS2597 links

C. Evaluation

The validated results have been compared with publicly available datasets provided by the following research projects: i) CAIDA's *AS Rank* dataset, which combines topological data obtained by the CAIDA's Archipelago Measurement Infrastructure with BGP routing data collected by the Route Views Project and RIPE NCC RIS; ii) Isolario *Topology* dataset, which combines BGP routing data collected by RouteViews Project, RIPE NCC RIS and PCH; iii) DIMES *ASEdges* dataset. Our measurements campaign was carried out in July 2012, so we considered the closest in time releases of the three datasets: i) June 2012 CAIDA's *ASRank* dataset[3]; ii) July 2012 Isolario *Topology* dataset[4]; iii) April 2012 DIMES *ASEdges* dataset[5].

By comparing the sets of AS2597 links provided by Isolario, CAIDA and DIMES with the AS2597 links detected by our probes, we found that the Portolan system could discover all the known links and, in addition, *16 previously unknown links* (22 against 6).

As previously mentioned, a subset of the target ASes were reached via AS137. Figure 4(b) shows that these target ASes are linked to GARR by the MIX IXP and two other IXPs: VSIX Neutral Access Point[6] (VSIX NAP) and Nautilus Mediterranean eXchange[7] (NaMeX). Hence, a number of links of AS137 was indirectly found by our measurements. By comparing the AS137 link set with the three chosen datasets we found that the Portolan system could discover all the Isolario, CAIDA and DIMES links and, in addition, *24 previously unknown links* (33 against 9).

D. Execution cost

We then conducted a rough evaluation of the execution complexity of the above task. The list of IP addresses to be used as targets has been built by selecting a single IP address for each /24 network of each target AS. The task submitted to the system was translated into 1556 microtasks, each containing 100 IP addresses belonging to a single AS. We noticed that 1407 microtasks belonged to 8 large ASes, whereas the remaining 149 microtasks belonged to 71 ASes. Thus, the vast majority of the target ASes owned less than 10% of the IP addresses to be probed.

The execution of a single traceroute operation took from 5 to 10 seconds, depending on the number of hops. Thus, each microtask was completed in approximately 10-15 minutes. In the worst case the entire task execution would have required 16 days of non stop execution on a single device. However, we included a couple of optimizations for speeding up the process: i) every time a public peering link between the source AS and a target AS, say X, was found, all the microtasks having X as destination were aborted and canceled; ii) microtasks have been grouped by target AS number, placed in separate queues, and scheduled for execution using a round-robin policy (this allowed to give the same priority to each target AS, making the previous optimization more effective). In this way, the experimentation required the execution of 135 microtasks. Since microtasks have been run for 6 hours/day by a single mobile device, the entire campaign required approximately 6 days of execution.

Finally, we considered the overhead of the Portolan app on the mobile device in terms of battery consumption and traffic. As previously said, the execution of a microtask with 100 targets determines a battery consumption not greater than 1%. We then calculated that a single traceroute generates approximately 3.4 KB of outgoing data and 4.4 KB of ingoing data. Hence, per microtask, approximately 340 KB are sent and 440 KB are received. The average traffic rate is less than 1KB/s. Since our main concern is to not degrade the performance of mobile devices, we decided to limit the load to 3 microtask per day on the app version distributed to volunteers. The drawback of this limit is the slowing down of task execution. For this experiment and with such limit the task execution would have required 48 days on a single device. However, since our system is based on a crowdsourcing approach, a task should be hopefully distributed over a large number of devices. For example, with 30 devices the previous task would have been accomplished in less than 2 days.

VI. Sensing the received signal strength

Portolan has been designed as a versatile measuring tool, using crowdsourcing to face large-scale problems. The connection quality of mobile networks is of paramount importance for both end users, who can select the operator that provides the best coverage, and for network operators, which can use detailed coverage maps to improve the offered service. Moreover, mobile networks provide access to the Internet to millions of users when using mobile devices, thus studying the correlation between signal quality and other parameters (e.g. bandwidth) can be extremely interesting. Given the size of the regions covered by mobile networks, mapping the signal strength through crowdsourcing is particularly favorable; thus, we performed some preliminary experiments using Portolan to build a coverage signal map of mobile networks.

A received signal strength measurement task has been submitted to Portolan, specifying an area surrounding the

[2]http://www.mix-it.net
[3]http://as-rank.caida.org/
[4]http://www.isolario.it/index.php?page=data_interface
[5]http://www.netdimes.org/PublicData/csv/ASEdges4_2012.csv.gz
[6]http://www.vsix.it/
[7]http://www.namex.it/

Fig. 5. Map of received signal strength

Faculty of Engineering of the University of Pisa. A small set of users, with the Portolan client installed on their smartphones, has been enrolled by the system when located in the area of interest. The results, consisting of lat-lon coordinates, timestamp, and registered signal strength have been aggregated by the Portolan server. An auxiliary Web application extracts the results from the Portolan server and displays them on top of a map, using different colors depending on the value of signal strength. Figure 5 shows the result of the collection campaign.

VII. Conclusions

As far as implementation is concerned, the use of the SWE framework provided a useful conceptual background and eased the formalization of several properties and activities, such as sensor capabilities, data representation, and tasking. In few occasions the framework has been perceived as restrictive with respect to our needs, but we have always been able to find reasonable solutions.

From a more general point of view, the implementation of the Portolan system proved the soundness of the "local" sensing approach: the amount of new links discovered by using even a single smartphone demonstrates the effectiveness of our crowdsourcing-based solution for the analysis of the Internet. Moreover, to the best of our knowledge, for the first time the idea of associating geographical coordinates to Internet measures has been put into practice. We believe that, in the near future, this will be of great help in providing fine-grained characterization of network-related properties. For instance, geolocalized measures of available bandwidth could be used to understand if the connection bottleneck is due to the connection provided by the mobile operator and in such case which are the areas where the problem is more evident.

Flexibility and versatility of the proposed approach are demonstrated by the experiments dedicated to mapping the received signal strength: the conceptual framework behind Portolan and its implementation allowed us to extend it with different measurement functionalities with limited effort.

Future work will focus on recruiting a significant user base and in running large scale measurement campaigns.

References

[1] 52North. http://52North.org/.

[2] Brice Augustin, Timur Friedman, and Renata Teixeira. Multipath tracing with Paris traceroute. In *Proceedings of the Fifth IEEE/IFIP Workshop on End-to-End Monitoring Techniques and Services (E2EMON 07)*, pages 1–8. IEEE, 2007.

[3] R. Aversa, M. Avvenuti, A. Cuomo, B. Di Martino, G. Di Modica, S. Distefano, A. Puliafito, M. Rak, O. Tomarchio, A. Vecchio, S. Venticinque, and U. Villano. The cloud@home project: Towards a new enhanced computing paradigm. *Lecture Notes in Computer Science*, 6586 LNCS:555–562, 2011.

[4] A. Bröring, J. Echterhoff, S. Jirka, I. Simonis, T. Everding, C. Stasch, S. Liang, and R. Lemmens. New generation sensor web enablement. *Sensors*, 11(3):2652–2699, 2011.

[5] The Cooperative Association for Internet Data Analysis (CAIDA). http://www.caida.org/.

[6] H. Chang, R. Govindan, S. Jamin, S. J. Shenker, and W. Willinger. Towards capturing representative AS-level Internet topologies. *Computer Networks*, 44(6):737–755, 2004.

[7] R. Cohen and D. Raz. The Internet Dark Matter: on the Missing Links in the AS Connectivity Map. In *IEEE INFOCOM*, 2006.

[8] A. Cuomo, G. Di Modica, S. Distefano, M. Rak, and A. Vecchio. The cloud@home architecture: Building a cloud infrastructure from volunteered resources. In *Proceedings of the 1st International Conference on Cloud Computing and Services Science (CLOSER 2011)*, pages 424–430, 2011.

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

[10] T. Foerster, B. Schaeffer, J. Brauner, and S. Jirka. Integrating ogc web processing services into geospatial mass-market applications. In *Proceedings of the International Conference on Advanced Geographic Information Systems and Web Services, GEOWS 2009*, pages 98–103, 2009.

[11] Lixin Gao. On inferring autonomous system relationships in the internet. *IEEE/ACM Trans. Netw.*, 9(6):733–745, December 2001.

[12] Isolario project. http://www.isolario.it/.

[13] MaxMind, Inc. http://www.maxmind.com/.

[14] W.B. Norton. *The Internet Peering Playbook: Connecting to the Core of the Internet*. DrPeering Press, 2011.

[15] Open Geospatial Consortium Inc. http://www.opengeospatial.org/.

[16] Ricardo Oliveira, Dan Pei, Walter Willinger, Beichuan Zhang, and Lixia Zhang. The (in)completeness of the observed internet as-level structure. *IEEE/ACM Trans. Netw.*, 18(1):109–122, February 2010.

[17] Ricardo V. Oliveira, Dan Pei, Walter Willinger, Beichuan Zhang, and Lixia Zhang. In search of the elusive ground truth: the internet's AS-level connectivity structure. *SIGMETRICS Perform. Eval. Rev.*, 36:217–228, June 2008.

[18] Packet Clearing House. http://www.pch.net/.

[19] Y. Rekhter, T. Li, and S. Hares. A Border Gateway Protocol 4 (BGP-4). RFC 4271 (Draft Standard), January 2006. Updated by RFCs 6286, 6608.

[20] Réseaux IP Européens Network Coordination Center (RIPE NCC). http://www.ripe.net/.

[21] University of Oregon RouteViews Project. http://www.routeviews.org/.

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

[23] P. Srisuresh and K. Egevang. Traditional IP Network Address Translator (Traditional NAT). RFC 3022 (Informational), January 2001.

[24] C. Stasch, T. Foerster, C. Autermann, and E. Pebesma. Spatio-temporal aggregation of european air quality observations in the sensor web. *Computers and Geosciences*, 47:111–118, 2012.

[25] Darryl Veitch, Brice Augustin, Renata Teixeira, and Timur Friedman. Failure control in multipath route tracing. In *INFOCOM*, pages 1395–1403. IEEE, 2009.