Opportunities and Challenges of Using Crowdsourced Measurements for Mobile Network Benchmarking
A Case Study on RTR Open Data

Cise Midoglu
Institute of Telecommunications
Vienna University of Technology
Gusshausstrasse 25/389, A-1040 Vienna, Austria
Email: cise.midoglu@nt.tuwien.ac.at

Philipp Svoboda
Institute of Telecommunications
Vienna University of Technology
Gusshausstrasse 25/389, A-1040 Vienna, Austria
Email: philipp.svoboda@nt.tuwien.ac.at

***Abstract*—Crowdsourcing is a novel paradigm which has applications in a plethora of disciplines including software development, public administration, and communication technologies. In the field of telecommunications, crowdsourcing makes a viable addition to state of the art benchmarking methods for mobile networks, when combined with mobile sensing approaches to employ smartphones as end nodes. In this paper, we review the opportunities and challenges of using crowdsourced measurements from smartphones for benchmarking mobile networks, and demonstrate some of these generic aspects using an open data set containing over two million entries. We show that there is a big potential to distributing performance measurements towards the peripherals of the network, but in order to achieve accurate benchmarking, the collection of data has to be complemented with appropriate signal processing. Our study also emphasizes the importance of open data and open source tooling in achieving fairness and repeatability.**

***Keywords*—Benchmarking; Crowdsourcing; Mobile networks; Performance evaluation; QoS; RTR Open Data; Smartphone**

I. Introduction

Crowdsourcing is an emerging paradigm in innovation, problem solving and knowledge acquisition where tasks are accomplished by a usually large and diverse crowd. The neologism was coined as a portmanteau of *crowd* and *outsourcing* by Howe [1] and defined as "the act of taking a job traditionally performed by a designated agent (usually an employee) and outsourcing it to an undefined, generally large group of people in the form of an open call". Crowdsourcing has been used in many fields of Information Technology (IT), commerce and industry.

Despite the increase in academic and industrial interest in crowdsourcing, there is still a high degree of diversity in the interpretation and the application of the concept. Hosseini et al. [2] analyze literature to deduce a taxonomy of crowdsourcing, defining the pillars of crodwourcing to be *the crowdsourcer*, *the crowd*, *the crowdsourced task* and *the crowdsourcing platform*, in order for particular interpretations and configurations to be precisely stated in a common framework. Dallora et al. [3] extend these to include *the call to participate*, *the reward*, *the type of process*, and *the outcome*.

In recent years, crowdsourcing applications in which mobile devices serve as the crowd, in contrast to those where the cognitive skills of humans are employed, have received incresing attention due to the explosive increase in the number of smartphones. Mobile sensing/computing exploits the pervasiveness of smartphones for data collection, computation, and processing, in which instantaneity and situation-awareness play an essential role. Equipped with a cellular/wireless connection, a multitude of sensors (location, light, movement, audio and visual), large/extendable memories, and powerful processors, smartphones make excellent end-terminals for measurement.

Some smartphone-based crowdsourcing applications are *OpenWeather* [4], where the potential of real-time temperature monitoring in densely populated areas using battery temperatures collected from mobile devices is demonstrated using an Android application, *Portolan* [5], [6], a system aimed at building an annotated graph of the Internet where the smartphones of participating volunteers are used as mobile monitors, *MCNet* [7] where Wireless Local Area Network (WLAN) performance is monitored via periodic sampling from smartphones, and *NetworkCoverage App* [8] where an Android application allowing for the measurement of cellular and WLAN network quality by executing active and passive measurements.

In this study, we systematically review the opportunities and challenges of using crowdsourced measurements from smartphones for benchmarking cellular mobile networks, and demonstrate some of these generic aspects using RTR Open Data[1], an open data set of measurements containing over two million entries. The extensive analysis of the state of the art allows for a thorough comparison among existing methods.

The paper is organized as follows: Section II describes the state of the art in benchmarking mobile networks and introduces RTR Open Data, Sections III and IV respectively present the opportunities and challenges of using crowdsourced measurements for benchmarking cellular mobile networks, and Section V concludes the paper.

II. State of the Art

Information regarding network performance is increasingly needed by almost every component of telecommunications

[1] https://www.netztest.at/en/Opendata [accessed 20.03.2016]

systems. Content providers, for instance, utilize this information to develop suitable applications, where mobile network operators might be interested in improving deployment, optimization, maintenance and benchmarking operations, and regulatory bodies focus on monitoring the networks of different operators for quality assurance and network neutrality. In this study, we focus on cellular network monitoring efforts carried out by operators or regulatory bodies for benchmarking purposes.

Network monitoring is traditionally carried out in a passive or active context. Passive measurements monitor core network interfaces and collect detailed performance statistics by merely observing existing traffic, whereas active monitoring mechanisms inject their own traffic and observe the reaction of the network. Based on this information, performance metrics such as available Downlink (DL) or Uplink (UL) bandwidth and link latency can be estimated.

Passive monitoring, as often called due to the practice being more of an observational study, entails monitoring traffic that is already on the network using interfaces to capture packets for analysis. This can be done using specialized probes or with built-in capabilities on switches or other network devices. Limitations of passive monitoring are the requirement for access privileges to all interfaces along the path of a traffic flow, along with extensive storage and processing power, which usually make it non-feasible to track a user through the network. Additionally, there is no control over the generated traffic, meaning that network issues and problems can only be observed while or after they occur (they cannot be reproduced for study).

Active measurements, on the other hand, do not rely on access to underlying core network infrastructure. The basic idea is to inject a sequence of probing packets at the sender (end device with cellular network connectivity) and to estimate respective performance metrics by analysing the output at the receiver (core network of the operator, or a designated server belonging to the regulatory body). The advantage of active probing is that arbitrary traffic can be generated. This allows for certain issues to be reproduced and desired scenarios to be simulated, as well as performing complex tasks, such as collecting measurements to verify that Quality of Service (QoS) agreements are being met. Despite the obvious disadvantage stemming from the necessity of injecting additional traffic, active methods provide the opportunity to selectively analyse the ”obscure” regions of the network which are not covered with sufficient detail while using passive methods.

A common practice is to place passive monitors in the proximity of the core network and the active probes towards the edges. The state of the art in active measurements comprise designated drive tests; the 3GPP Minimization of Drive Tests (MDT) work item finalized in March 2011 as part of Release 10 (Rel-10) is discussed in detail in [9]. The downside of conventional drive tests are: consumption of significant amount of time and human effort, geographically limited data acquisition, requirement of off-line analysis which hinders real-time monitoring, and large Operation Expenditure (OPEX).

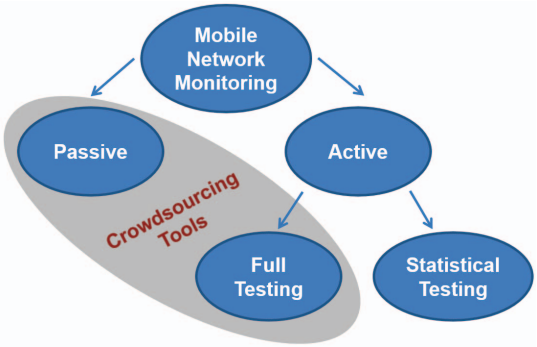

Fig. 1: Categorization of mobile network monitoring paradigms

A recent alternative to drive tests is crowdsourcing the necessary measurements using User Equipments (UEs), which carries along the paradigm of pushing the monitoring efforts towards the periphery of the network, where they are most relevant and representative of Quality of Experience (QoE). The idea is for smartphones to operate as geo-localized sensors capable of monitoring the state of the access network (passive) and serving as probes (active), where the common practice is for the devices to run a mobile application which, besides coordination with a designated server, can provide measurements of network-related properties such as available bandwidth and delay. Figure 1 indicates the location of existing crowdsourcing tools in our categorization of mobile network monitoring paradigms.

Some of the widely-deployed tools for crowdsourced benchmarking of network performance include *Speedtest*[2] by Ookla, *FLOQ*[3] by NCQA, *Kyago*[4] by Zafaco, *Traffic Monitor*[5] by RadioOpt, and *Netztest*[6] by RTR[7]. Such tools are generally for both fixed and mobile networks, hence they are available as browser tests and mobile applications (most have both Android and iOS versions). All employ a uni-modal interaction channel where tests are initiated by end users and results are collected and aggregated on a central server. Similar to conventional drive tests, their common approach in active scenarios is full (intrusive) testing, where the communication link under observation is completely used for the duration of the test (see Figure 1).

In sections III and IV, we use our analysis of RTR-Netztest to introduce some of the opportunities and challenges associated with crowdsourced mobile network benchmarking,

[2]www.speedtest.net [accessed 20.03.2016]
[3]www.floq.net [accessed 20.03.2016]
[4]www.kyago.de [accessed 20.03.2016]
[5]www.trafficmonitor.mobi [accessed 20.03.2016]
[6]https://www.netztest.at [accessed 20.03.2016]
[7]The Austrian Regulatory Authority for Broadcasting and Telecommunications (Rundfunk & Telekom Regulierungs-GmbH, RTR) is a semi-private company established under the Austrian Law, which provides operational support for the Austrian Communications Authority, the Telekom-Control-Commission and the Post-Control-Commission. https://www.rtr.at/en/rtr/RTRGmbH [accessed 20.03.2016]

TABLE I: Some of the fields in the RTR Open Data. Items marked with (*) indicate multi-valued parameters (time series)

Type of Field	Name of Field
Test Related	open_test_uuid open_uuid time_utc implausible
Position Related	lat long loc_src loc_accuracy
Device Related	client_version model platform product
Network Related	cat_technology network_type network_mcc_mnc sim_mcc_mnc
Performance Related	download_kbit upload_kbit signal_strength lte_rsrp lte_rsrq speed_curve* ping_ms speed_curve_threadwise*

due to the availability of its source code[8] and results[1] as open data. Overall, a total of 68 parameters are available in the data set for each measurement sample and data available from each measurement can be pulled by individual query in JavaScript Object Notation (JSON) format from the RTR server using a unique test identifier (*open_test_uuid*). Table I shows some of the parameters available in the data set. Complete list of available parameters, their descriptions and instructions on importing can be found in the *Open Data Interface Specification*[1].

III. Opportunities

The main opportunities of crowdsourcing mobile network benchmarking can be identified as *the power of the crowd*, *mobility and ubiquity*, *real-time operation*, *cost reduction* and *representation of realistic user performance*. Combined, these provide a viable alternative to the challenges associated with conventional drive tests.

A. "Power of the Crowd"

One reason for adopting a crowdsourcing-based approach in mobile network benchmarking, referred to as "the power of the crowds" in [6], can be advocated for by reasons of parallelization and reduction of organizational overhead. When a large task is divided into a set of small and loosely coupled microtasks, monitoring activities are parallelized and completion time can be reduced. Similarly, when the monitoring task is distributed among end nodes, each of which are charged with the completion of their own part by employing their own resources (precisely the case of mobile applications running on users' hardware), the organizational overhead is reduced both in a financial and practical sense, in comparison with the complete maintenance of a dedicated system.

In a small scale, this can be exemplified by looking at typical smartphone user behaviour: owners keep their smartphones in operational condition, they charge their own devices, they implement software updates, and they pay for maintenance and repair, precisely because they regularly use these devices. Utilization of smartphones for measurement has all the advantages of outsourcing.

B. Mobility and Ubiquity

Human mobility offers remarkable opportunities for ubiquitous sensing, and systems which envisage the use of mobile devices carried by people can exploit this to its full potential.

An important characteristic presented by smartphones is the easy geo-localization possibility through the embedded Global Positioning System (GPS) unit. The presence of a precise positioning system, combined with human mobility allows for the collection and analysis of data not only in the temporal dimension but also in the spatial dimension. This is critical for mobile network benchmarking, since the position of the end user is an important parameter affecting network performance as observed by the user. Hapsari et al. [9] identify the conventional drive test measurements where, typically, a measurement vehicle such as a van equipped with specially developed test terminals, measurement devices and a GPS receiver is used to check outdoor coverage, as being specific to the measurement route.

Maps of measurement locations in Austria where the RTR-Netztest tool was used by a mobile device, provided in the form of open data[9] can be used to exemplify the potential ubiquity of crowdsourced measurements. Figure 2 shows how wide the coverage of the RTR-Netztest application in Austria is, albeit being relatively new (little over 3 years), as well as how test density coincides with population density[10], indicating a realistic representation of consumer location.

A crowdsourcing system based on a smartphone application allows users to contribute with measurements from almost everywhere: indoor, outdoor, urban, suburban, rural locations, roads, highways and train tracks. The advantage over conventional drive tests is that areas that are hard to reach via measurement vehicles optimized for highways (such as densely populated urban areas, indoor locations and train tracks) can also be covered without any need to customize implementation.

The particular benefit of potentially having a multitude of simultaneous measurement results from different localities in the same time period is the possibility of detecting network-wide problems. In [10] authors demonstrate the effectiveness

[8]https://github.com/alladin-IT/open-rmbt [accessed 20.03.2016]

[9]https://www.netztest.at/de/Karte [accessed 20.03.2016]

[10]official statistics on population density per region can be retrieved in the form of open data from https://www.data.gv.at/katalog/dataset/land-noe-bevolkerung-nach-gemeinden-volkszahlungen [accessed 20.03.2016] and http://www.statistik.at/web_de/statistiken/menschen_und_gesellschaft/bevoelkerung/index.html [accessed 20.03.2016]

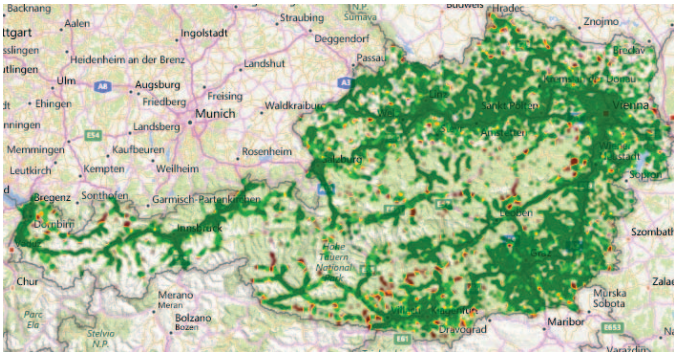

Fig. 2: RTR-Netztest measurement locations for the last year; single operator, all cellular technologies, heatmap according to 80% percentile of DL data rate

of monitoring service-level network events through crowdsourcing and in particular show that locally detected events, if properly correlated, can be considered as the symptoms of a widespread network problem. This approach can be extended to the temporal dimension as well, and can serve as a metric for comparison between congested and non-congested states (time of day effects in network performance).

C. Real-Time Operation

Conventional drive tests consume significant time and human efforts to obtain reliable data, leading to delays in the detection of network problems and the deployment of countermeasures [9]. In this regard, crowdsourcing has the potential benefit of providing timely measurement results which might be critical for identifying network issues.

Using mobile applications on smartphones, the results from each device can immediately be available to the server after the completion of a single measurement, whereas in conventional drive tests each device typically runs a complete set of measurements before results can be obtained, and these might not even be uploaded to the server until the drive is complete (for instance, once a day). Distributing measurements to a higher number of nodes facilitates modularity in operation, decreases overall time consumption due to idle periods between measurements, supports online analysis, and enhances the situation awareness of the benchmarking body.

The RTR-Netztest has three main stages within one measurement: DL test, UL test, and ping test. The DL and UL tests have a fixed nominal duration of 7 seconds, and ping test consists of the exchange of 10 Transmission Control Protocol (TCP) pings. As shown by Table II, although an intrusive full testing method is employed by the RTR-Netztest, test duration is neglectable for the use case of a few measurements by the same user within a day. Even periodic testing would be possible with the current paradigm, which yields an overall duration in the range of 30 seconds. The possibility of reduction in duration through the use of non-intrusive statistical methods is discussed in Section IV, along with the potential of running performance measurements continuously in the background.

TABLE II: Duration of a measurement by the RTR-Netztest; minimum, maximum and median values aggregated over a total of 245 measurements in 4G, overall durations include server selection, pre-loading and intermittent waits

Measurement Phase	Duration (s)	
DL Test	Max	7.0
	Min	7.0
	Median	7.0
UL Test	Max	7.3
	Min	5.7
	Median	7.0
Ping Test	Max	1.6
	Min	1.1
	Median	1.3
Overall	Max	32.4
	Min	26.1
	Median	27.2

TABLE III: Qualitative analysis of potential cost items for crowdsourcing and drive tests (P: personnel, HW: hardware, SW: software, T: testing, PP: post-processing)

Category	Potential Cost Items	
	Drive Tests	Crowdsourcing
Development	P, HW, SW, T(HW+SW)	P, SW, T(SW)
Installation	P, HW, SW	SW
Maintenance	P, HW, SW	P, SW
Operation	P	-
Other	PP	PP

D. Cost Reduction

One of the biggest benefits of crowdsourcing performance measurements is the reduction in cost compared to conducting drive tests. In comparison to systems where design, deployment and operation are centralized, crowdsourcing allows outsourcing some of the major cost items to the end nodes.

Table III illustrates a qualitative analysis of the potential cost items for crowdsourcing in comparison to drive tests[11]. In general, smartphones are more mature than PCs or other devices in software installation and upgrade. Distribution of software is extremely simple (cost item for crowdsourcing in installation category only comprises software upload on app store) and end users are accustomed to the guided process of application download and installation. Implementation of new features and bug fixes are also made simpler, thereby reducing the relevant cost item for crowdsourcing in the maintenance category.

The substantial cost reduction achievable by crowdsourcing is mainly due to the utilization of smartphone owners' own resources: employment of personal hardware as measurement equipment, installation and operation carried out in owners' own time.

E. Representation of Realistic User Performance

One of the most important benefits coming from shifting network monitoring towards the end systems is that services and applications can be observed where they are used, and

[11]It should be noted that software cost for drive tests in the development category might also include the purchase of and integration efforts towards expensive third-party software.

TABLE IV: Comparison of parameters that can be read from different mobile OSs

Parameter	Android	iOS	Windows
Location	+	+	+
Location Source	+	+	+
Connection Type	+	+	+
IP address	+	+	+
Server	+	+	+
Name of ISP	+	+	+
Device ID	+	+	+
Network Operator	+	+	-
SIM Operator	+	-	-
Cell ID	+	-	-
LAC	+	-	-
IMSI, IMEI	+	-	-
Signal Strength	+	-	-

this paves the way for the evaluation of performance metrics from the end user perspective. It is most accurate to query for QoE and/or to verify that QoS agreements are being met on the nodes where they are relevant.

Additionally, most of the available crowdsourcing tools employ a multi-threaded TCP implementation in estimating available bandwidth, which coincides with studies showing the dominance of Hypertext Transfer Protocol (HTTP), a TCP-based application among the service mix in mobile networks, and could be considered as a good representative of real end user experience.

IV. Challenges

In this chapter we review some of the generic challenges inherent to crowdsourcing mobile network performance measurements via smartphones, many of them are demonstrated by the study of RTR Open Data.

A. End Device Related Issues

1) Availability of Parameters: There are big differences among the wide range of smartphone models in terms of the parameter set that can be reported by the mobile Operating System (OS), as well as the restrictions of reading cellular network-related information, such as mobile technology, signal strength, cell ID, from smartphones in general. Table IV lists some of the basic parameters and whether they can be read by an application on a smartphone running Anroid, iOS and Windows mobile OS.

A generic challenge is that mobile OSs, which are generally optimized for easy production of application-level software, do not provide support to low-level networking mechanisms and tools as a main goal; they are restricted environments. For security reasons, programmers of mobile applications are confined to a sandbox (the available classes of a relevant Application Programming Interface (API), such as *android.net* or *android.telephony* for Android systems).

Generally speaking, Android is the most ”open” system in terms of reporting device and network related information. For instance cell ID, Location Area Code (LAC), signal strength metrics Received Signal Strength Indicator (RSSI), Reference Signal Received Power (RSRP) and Reference Signal Received Quality (RSRQ), and mobile technology, as well as device metrics International Mobile Subscriber Identity (IMSI) and International Mobile Equipment Identity (IMEI) can be read out by virtually any Android device, while this is not possible on iOS devices. However, the Android APIs are also somewhat vague in certain cases, especially with regards to reading signal strength[12].

There is also some information which is only available from rooted devices, such as non-aggregated signal strength for Multiple Input Multiple Output (MIMO) (i.e. reading the received signal strength from each antenna separately), which, for a more detailed analysis of the MIMO performance of the network could have been useful, but cannot be implemented in a widely deployed mobile application since users cannot be asked to root their phones.

The differences in the availability of parameters among mobile OSs pose a great challenge in the design of crowdsourcing-based mobile network benchmarking systems: in order to exploit the potentials listed in the previous section, as many end users should be involved as possible, but they cannot be forced to use a certain OS. Therefore, data collected from smartphones with different properties (OS type and version, application version, etc.) has to be aggregated in the post-processing stage. The merging of datasets with different parameters and different levels of granularity requires what we call *baselining*, and is still an open question.

2) Unknown Properties: One of the most adverse effects of crowdsourcing is the distribution of network performance measurements to essentially unknown end nodes. The User Equipment (UE)s that perform measurements are not under the control of a central authority but rather that of their owners. This leads to the *unknown ground truth* problem in the post-processing stage. For instance, some of the properties of the smartphone-based measurements that are not directly transparent to the central server of the RTR-Netztest are: battery status, Central Processing Unit (CPU) load, type and number of different applications running simultaneously with the measurement, and indoor/outdoor distinction, which is not straightforward from location.

The latter is a rather serious problem, since there is no way for a benchmarking application on a mobile phone to detect the details of the current position of the phone (e.g. right in front of a building or in its basement) based only on latitude and longitude. Although there are some preliminary ideas about using additional sensors for the classification of indoor/outdoor environments [11], there is no extensive scientific study on the validity of results. The evaluation of radio quality in the post-processing stage must take into account the many factors that affect signal strength in indoor environments.

[12]There are multiple ways to read signal strength in Android systems, some differentiated for mobile technologies, but the documentation is not always clear about what is actually being read from the chipset and there is a difference in the old and new APIs regarding the implementation of certain functions

3) Personal Usage/Preferences: Designing a mobile benchmarking application for smartphones must take into account that the download and installation of the application is a decision that the smartphone owners make. Owners' control over their smartphones entail personal preferences such as the activation/deactivation of GPS, WLAN, and possible mobile technology locks. Such user-defined parameters affect measurement results.

For instance, disabled GPS might cause location information to be acquired from the network which has around $600-1000$ m accuracy as opposed to $8-10$ m for GPS (source of location information is indicated by the *location_src* field in the RTR Open Data, see Table I). Similarly, activation of WLAN might cause a technology change from cellular to WLAN during measurement, depending on the availability of Access Point (AP)s in the vicinity.

Intervention into such preferences are possible but requires that more permissions be granted by the user to the application during the installation stage, and users might be reluctant to approve a long list. Most of the benchmarking tools mentioned above keep the list of permissions that need to be approved by the user to a minimum, and therefore can control less parameters compared to the scenario where infrastructure is dedicated.

B. Resource Consumption

The challenges regarding the scarcity of resources touched upon in [6] regarding smartphone-based crowdsourcing applications, namely *battery consumption* and *data volume* are critical for cellular network benchmarking.

The crowdsourcing tools mentioned in Section II employ full testing (require continuous data transmission during each measurement phase), which results in the consumption of data volumes that increase with overall measurement duration. However, there is no extensive analysis of these tools in literature in terms of battery power and data volume consumption. As a preliminary analysis, we provide the results of a measurement campaign in Table V which include the volume consumption per tool.

Table V shows the huge amount of data volume consumed by the tools, as well as the mishap of the fixed-duration paradigm in comparing different technologies which have different data rates. It is possible to see that with fixed duration, the higher the maximum achievable data rate in a technology is, the more data volume a measurement in that technology will consume (compare 3G and 4G for instance). We therefore propose the adoption of an *adaptive* scheme in determining test duration for full testing, using either the a priori information regarding network technology at the beginning of the test, or identifying the peak data rate by considering the completion time of the TCP ramp-up phase. This approach is discussed in subsection *Measurement Algorithms*.

However, a more general question is how to perform active benchmarking in *reactive* networks without the need for full testing. The necessity to optimize the sharing of a common transmission medium between different users requires cellular networks to react to the traffic pattern injected by each user, and this causes each measurement to alter the state of the network in a unique way. Today's cellular networks are neither stateless nor memoryless, but rather reactive depending on the probing patterns used for benchmarking. Therefore, the probing patterns selected for performance evaluation, as well as the post-processing, must account for the reactiveness of the network.

TABLE V: Average reported values and data volume consumed per test from different tools; 100 measurements conducted with a Samsung S5 device on technology lock in the same location using an unlimited SIM card from a prominent mobile operator in Austria; tools are anonymized

Technology	Parameter	T1	T2	T3	T4	T5
3G	Downlink (Mbit/s)	11,9	15,8	18,3	24,1	18,4
	Uplink (Mbit/s)	2,0	2,7	2,6	4,2	2,2
	Delay (ms)	84,1	81,9	38,3	31,9	65,0
	Volume (MB)	20	27	58	35	32
4G	Downlink (Mbit/s)	68,9	63,2	61,1	65,9	58,4
	Uplink (Mbit/s)	22,1	45,7	44,2	34,7	44,3
	Delay (ms)	39	81,9	60,9	21,1	42,8
	Volume (MB)	65	70	85	56	37

Recent publications have started to consider reactive networks and propose strategies for fair network benchmarking. They currently focus on delay and are based on extensive measurements scanning all possible input traffic patterns in order to capture reactiveness [12]. This corresponds to a search among different methodologies, where the self-injected probing traffic usually takes the form of packet pair [13], packet train [14], or packet chirps. Some approaches directly calculate the available bandwidth based on a statistical model of the network path (gap model), while others estimate the available bandwidth by iteratively probing a path with different rates (rate model) [15]. The approaches based on the gap model are evolved from the packet-pair method.

Some of the existing tools that incorporate statistical testing are *Delphi* [16], *Pathload* [17], *PathChirp* [18], *Spruce* [15], *IGI/PTR* [19], *WBest* [20], and *Assolo* [21]. However, none of these tools are in widespread use, due to the lack of an end user available format such as a mobile application, and some are not yet implemented.

Implementation of these tools on smartphones could pose a challenge, as these platforms don't allow raw access to the network stack (see Subsection A) and it may be challenging to get accurate timing for the network packets [22]. However, in order to facilitate the acquisition of a large number of measurements and to enable continuous monitoring, it is obligatory to develop a tool which implements statistical probing and can be adopted by a large group of end users.

C. Privacy versus Reliability

Data quality is a major concern in crowdsourcing-based mobile benchmarking, since, by definition, all individuals with a smartphone can contribute directly to the process. A means of increasing the confidence in the results is to uniquely identify each measurement, as well as each end node.

We define an end node by the tuple (<device, OS, application version, home network, roaming network, technology>) where the device can be identified through IMEI or the IMSI number, and a way to identify the network is to check the Internet Protocol (IP) address or the Mobile Country Code (MCC) and Mobile Network Code (MNC) as reported by the device. Unique end node identification is required for building a reputation-trust mechanism, which enhances the reliability of the collected data. Using this information, post-processing stages equipped with filtering algorithms can remove the samples coming from end nodes which have been established as unreliable. This increases the overall sanity of the system by allowing for the omission of faulty nodes.

With regard to crowdsourcing, two aspects of privacy are discussed in literature: protecting the crowdsourcing platform and the network from malicious end nodes, and protecting the end nodes. It is acknowledged that the practice of mobile application based crowdsourcing does not increase the risk to end nodes, since applications running on smartphones are subject to usual controls operated by application stores and the integrity of the devices are preserved as long as code sources are reasonably trusted

However, presenting measurement results as an open data set raises challenges. Although it might be possible and relevant to collect certain information such as IMEI, IMSI, or the full IP address for purposes of traceable end node identification, it might not be ethically acceptable to display these if any piece of the information can be traced back to the owners of the end devices. For this reason, none of the existing tools allow for the sharing of user sensitive data. For instance, in RTR Open Data (see Table I), the `open_uuid` field which roughly identifies end nodes is only traceable within a day, IP addresses are available only after anonymization, and the boolean `implausible` field is used in a measurement-oriented way (instead of marking unreliable end nodes, which possibly generate multiple unreliable samples, the samples themselves are marked without any mapping to the end node).

D. Sample Size and Incentive Mechanisms

Although crowdsourcing has the potential to incorporate a multitude of measurements with diverse profiles (different devices, different locations, different times), it is not guaranteed that the samples are distributed adequately in time and space. Figure 3 shows the non-uniform daily pattern of the number of tests started using RTR-Netztest with respect to the time of day. It is seen that the main trend is towards conducting the measurements in the afternoon and the evening, with the peak around 6pm.

On one hand, naturally arriving measurements might be too scarce for benchmarking, especially if a particular scenario is considered. From a total of 2.131.728 samples between 2013 and 2015, only 837.617 (around 39%) are cellular tests conducted in Austria, and 464.752 of those tests (around 22%) are in 3G, for instance. Considering that multiple technologies, such as Universal Mobile Telecommunications System

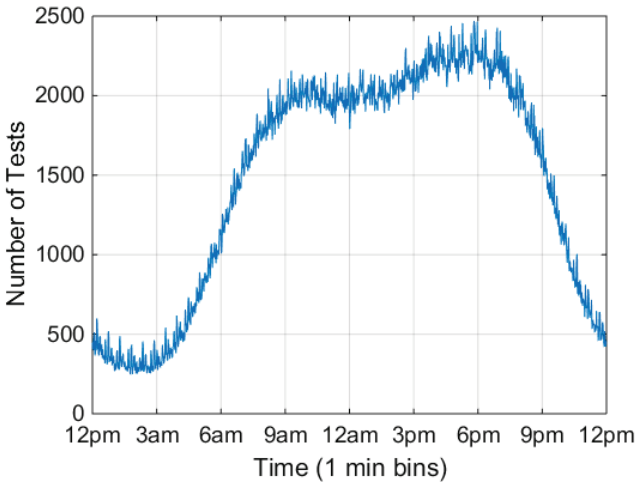

Fig. 3: Number of tests started using RTR-Netztest versus time of day (bin size: 1 minute); aggregate over all samples between 2013-2015

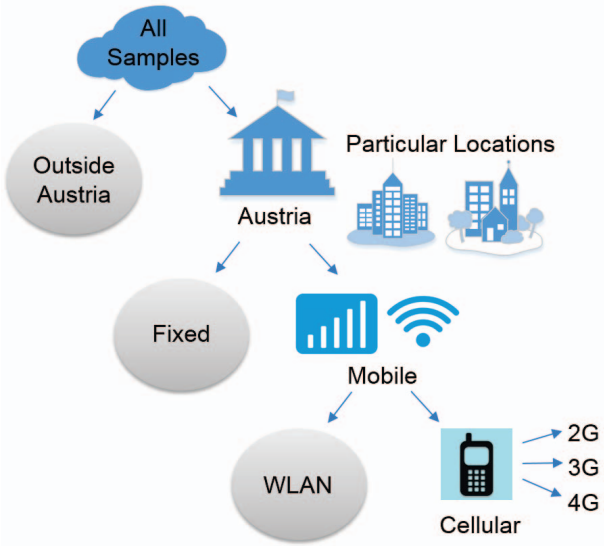

Fig. 4: Tree representation of different categories in the RTR Open Data, sample size decreases with each step

(UMTS), High-Speed Packet Access (HSPA) and Evolved High Speed Packet Access (HSPA+) are all categorized as 3G, one can imagine a three-fold reduction in sample size for a specific technology, and see that even smaller numbers are obtained if only a single operator is considered. Figure 4 represents how the seemingly large total number of tests in RTR Open Data can divide into a rather small number in this way.

On the other hand, crowdsourcing is prone to suffering from the existence of redundant information. Considering the current configuration of the RTR-Netztest, for instance, the server has no way of ”refusing” measurements (it may only choose to permit the client at a later time, if the number of active measurement is too high) and this leads to the

accumulation of potentially large amounts of measurement results, which might actually be more than needed from a device, an area, or a time period. In this aspect, the marking of useful samples is key.

For research purposes, we encourage the use of multiple fields in the data to distinguish between samples suitable for different types of analysis, and indicate those that might need to be complemented with additional information. Technology change during test, for instance, is not a cause for complete dismissal, but must be treated with care and not go undetected, where complete lack of location information might constitute a reason for discarding a sample.

Another challenging issue, directly related to optimizing the number of samples, is incentive. In the particular case of mobile network benchmarking, the challenge stems from the fact that there is no reward system associated with conducting a measurement, leading to users running these tools on their mobile devices under unknown motivation. A typical problem could be that most measurements are made by users who employ one of these tools for troubleshooting when they have bad network quality. However, it is not possible to do reliable and fair benchmarking among different networks if measurements only come from the problematic areas or temporalities.

Similarly, as seen from Figure 3, there are certain daily trends in the absence of incentive, and these have to be considered if a specific pattern needs to be established (e.g. peak number of tests to be shifted to 3am). [6] lists some of the methods to motivate users as money, altruism (e.g. thinking that the problem being solved is socially important), entertainment, and implicit work. Further, it is suggested that users would be self-incentivized to participate in efforts to fingerprint and georeference wireless networks since they could afterwards be able to select the carrier that provides the best coverage in the area where they live in; but this has not yet been verified by extensive social studies.

The role of motivation is very important to reach and maintain a critical mass of users. The question of how to leverage appropriate human resources to extend the capabilities of crowdsourced benchmarking should be explored further, and especially, incentive mechanisms should guarantee that participants can obtain more rewards if they complete tasks at a higher quality (such as turning on their GPS to provide more accurate location data).

E. Measurement Algorithms

Table V shows a large variance in the values reported by different benchmarking tools, although measurements were performed with the same device in the same technology and the same operator. This raises serious questions in terms of accuracy, and exposes one of the biggest challenges inherent to crowdsourcing: how to select, measure, and evaluate the appropriate metrics for benchmarking, when there is a huge diversity in the specific conditions of each end node?

Among the most common metrics used for network performance evaluation by these tools are DL data rate, UL data rate and delay. For the sake of brevity, we will confine our review to the evaluation of DL data rate here. The most common practice is to estimate available bandwidth by calculating average DL data rate achieved by an end node during a time interval. For instance, RTR-Netztest opens multiple TCP streams during the DL test (maximum number is 3 in the current version), within which the client simultaneously requests and the server continuously sends data streams consisting of fixed-sized chunks. After a fixed amount of time (nominally 7 seconds) the server stops sending on all connections, and the available bandwidth is estimated by adding the number of bytes downloaded by each thread and averaging over the fixed test duration [13].

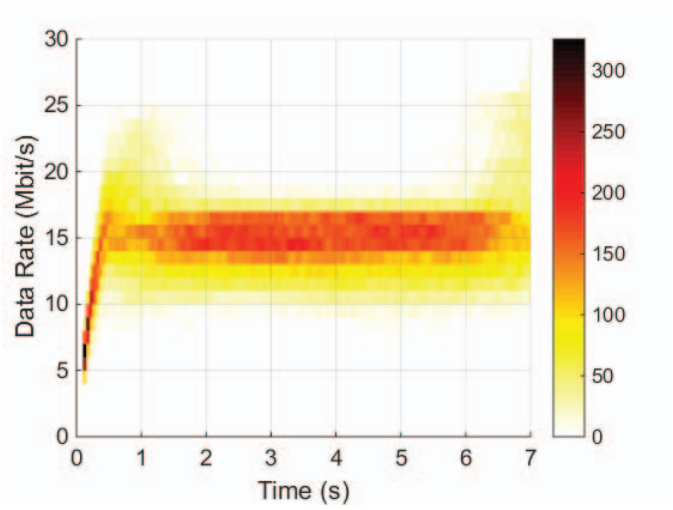

Fig. 5: Histogram of the DL data rate time series for 900 threads from 300 measurements conducted in 4G; bin size 1Mbit/s-50ms

In order to evaluate this method, we first look at the behaviour of the threads in time. Figure 5 shows a histogram of the DL data rate, reported as a time series (*speed_curve_threadwise* item in the open data, see Table I) by 900 threads belonging to 300 measurements we have made with an LG-D390n device running an Android 4.4.4 OS using an unlimited SIM card in 4G, after smoothing and resampling. It is possible to observe the characteristic TCP behaviour of an initial ramp-up phase followed by a somewhat stable phase in all of the curves. This indicates that averaging over the whole time series includes a period of time when the data rate is lower than the available bandwidth due to the transport layer protocol, hence yielding an underestimate.

To examine in more detail, we look at the aggregated time series for DL data rate (data rate aggregated over all DL threads can be extracted via the *speed_curve* item in the open data, see Table I), and try to identify the stable period through breakpoint detection. As an alternative to taking the overall average of data rate, which includes the TCP ramp-up phase and therefore yields a lower estimate, we calculate a ”good average” by considering the stable period after the breakpoint,

[13]https://www.netztest.at/doc/ [accessed 20.03.2016]

Fig. 6: Aggregated time series for DL data rate for single sample, alternative calculation results reported on plot

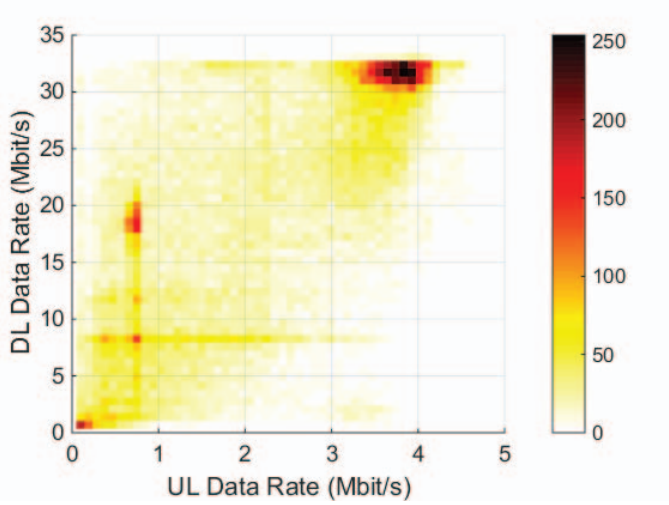

Fig. 7: Density map of DL data rate versus UL data rate for RTR Open Data samples from a single operator in 3G for the year 2014; 50x50 bins

in which the device is using all available network resources. This is a potentially better representation of network quality, since there is no external bottleneck.

Figure 6 shows the aggregated time series for DL data rate for a single sample we randomly selected from the RTR Open Data ($open\_test\_uuid = O8295114b-8150-4fe6-a9d8-1ea63fd3e0e3$), with a Cumulative Sum (CUMSUM) average applied to the smoothed and resampled curve. The position of the breakpoint is determined by configuring the convergence criteria (currently allowing up to 15% difference from overall average). It is seen that the "good average" value can be up to 3% higher than the reported value.

We have employed a moving average filter in smoothing and applied CUMSUM for simplicity of analysis here, but the extension of this algorithm to more sophisticated causal filters could also allow for the online detection of the stable period. If the stable period can be detected in real-time, only a few seconds would be enough to calculate the peak achievable data rate. This would enable an adaptive selection of test duration: as soon as the peak is identified within an acceptable error range, the test could be stopped. Adaptive test durations could thus decrease the consumption of valuable smartphone resources.

F. Inference of Network Performance from User Performance

Another big challenge for benchmarking today's cellular networks is that, most of the customers are tariff-limited. Tariffs might appear as a limit on the data rate achievable by the customer, as well as a limit on the total volume that a customer can consume per month, the latter of which might be very hard to detect.

The problem in terms of fair benchmarking is that limited users do not fully represent network quality. Figure 7 presents a density plot of samples from RTR Open Data with respect to DL and acUL data rate, in order to give an idea about which user groups (differentiated by device, technology, and tariff) exist in the dataset. It can be seen that the data rates achieved by end users span a very large range, even within the scope of a single generation of mobile technologies, in a non-uniform manner indicated by the clusters around certain values.

It is clear that, in the existence of limited users, if benchmarking algorithms simply calculate an average or quantiles over all samples, they would not precisely be measuring the maximum achievable performance for a particular network (which is a general benchmarking objective). Rather, they would get a vague sense of average user performance, including many diverse groups which are not weighted properly, and cannot differentiate between tariff options that customers willingly select and undesired network problems coming from the operator side.

A detailed analysis of the user clusters in the crowdsourced data set is required for a better understanding of network performance. Possible approaches include: discarding all measurement results coming from limited users (very undesirable due to the potential massive decrease in sample size); identification and tracking of cluster centers for identifying the *changes* in network performance (minimal approach); or finding the correlation between the performance of cluster centers and the overall network, in order to reach repeatable translation algorithms between user performance and network performance (ideal).

Deriving this translation metric is of key importance for establishing crowdsourcing as a viable alternative to the conventional methods for mobile network benchmarking.

V. Conclusion

In this paper, we reviewed the opportunities and challenges of using smartphone-based crowdsourcing to complement/replace traditional mechanisms for mobile network benchmarking, such as conventional drive tests. Some of the opportunities were identified as *"the power of the crowd"*,

mobility and ubiquity, *real-time operation*, *cost reduction* and *representation of realistic user experience*, where the challenges were identified as *end device related issues*, *resource consumption*, *privacy versus reliability*, *sample size and incentive mechanisms*, *measurement algorithms*, and *inference of network performance from user performance*.

It is seen that there is a big potential to distributing network performance measurements towards the end nodes, but in order to achieve accurate and fair benchmarking, the post-processing stages have to be designed with care, paying special attention to the measurement and calculation of relevant performance metrics without external bottlenecks which obfuscate the effects of the core network, allowing for the translation of raw values collected from end nodes into realistic estimates of network performance.

Although crowdsourcing-based systems carry a great potential to render some of the traditional benchmarking practices obsolete, they are not yet completely mature. Open areas for research include the identification of appropriate benchmarking metrics, minimization of resource consumption (including the implementation of non-intrusive and/or faster converging measurement algorithms), design of appropriate incentive mechanisms, detection and analysis of the user clusters in mobile networks, and accurate representation of network performance.

ACKNOWLEDGEMENTS

This work was supported by the Austrian Research Promotion Agency (FFG) Bridge Project 850742: Mc.Hypa-Miner (Methodical Solution for Cooperative Hybrid Performance Analytics in Mobile Networks).

The authors would like to thank Dipl.-Ing. Dietmar Zlabinger and Dipl.-Ing. Dubravko Jagar from the RTR-Netztest Technical Team and Leonhard Wimmer from SPECURE GmbH for their continuous support regarding the RTR-Netztest, and Vaclav Raida and Martin Horak for their assistance in the analysis of RTR Open Data during their bachelor studies.

REFERENCES

[1] J. Howe. (2006, June) Crowdsourcing: A definition. [Online]. Available: http://crowdsourcing.com/cs/2006/06/crowdsourcing_a.html

[2] M. Hosseini, K. Phalp, J. Taylor, and R. Ali, "The four pillars of crowdsourcing: A reference model," *2014 IEEE Eighth International Conference on Research Challenges in Information Science (RCIS)*, May 2014.

[3] A. L. D. Moraes, F. Fonseca, M. G. P. Esteves, D. Schneider, and J. M. de Souza, "A meta-model for crowdsourcing platforms in data collection and participatory sensing," *Proceedings of the 2014 IEEE 18th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, May 2014.

[4] A. Overeem, J. C. R. Robinson, H. Leijnse, G. J. Steeneveld, B. K. P. Horn, and R. Uijlenhoet, "Crowdsourcing urban air temperatures from smartphone battery temperatures," *Geophysical Research Letters*, vol. 40, no. 15, pp. 4081–4085, Aug 2013.

[5] E. Gregori, L. Lenzini, V. Luconi, and A. Vecchio, "Sensing the internet through crowdsourcing," *2013 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, Mar 2013.

[6] A. Faggiani, E. Gregori, L. Lenzini, V. Luconi, and A. Vecchio, "Smartphone-based crowdsourcing for network monitoring: Opportunities, challenges, and a case study," *IEEE Communications Magazine*, vol. 52, no. 1, pp. 106–113, Jan 2014.

[7] S. Rosen, S.-j. Lee, J. Lee, P. Congdon, Z. Mao, and K. Burden, "MCnet: Crowdsourcing wireless performance measurements through the eyes of mobile devices," *IEEE Communications Magazine*, vol. 52, no. 10, pp. 86–91, Oct 2014.

[8] F. Kaup, F. Jomrich, and D. Hausheer, "Demonstration of network coverage - a mobile network performance measurement app," *International Conference on Networked Systems (NetSys)*, 2015.

[9] W. A. Hapsari, A. Umesh, M. Iwamura, M. Tomala, B. Gyula, and B. Sebire, "Minimization of drive tests solution in 3GPP," *IEEE Communications Magazine*, vol. 50, no. 6, pp. 28–36, Jun 2012.

[10] D. R. Choffnes, F. E. Bustamante, and Z. Ge, "Crowdsourcing service-level network event monitoring," *SIGCOMM Comput. Commun. Rev.*, vol. 40, no. 4, p. 387, Aug 2010.

[11] M. Stahr. (2015, Feb) Determine the environment of smartphones by an indoor/outdoor classifier. [Online]. Available: https://www.radioopt.com/determine-the-environment-of-smartphones-by-an-indoor-outdoor-classifier/

[12] M. Laner, J. Fabini, P. Svoboda, and M. Rupp, "End-to-end delay in mobile network: Does the traffic pattern matter?" *Proceedings of the Tenth International Symposium on Wireless Communication Systems (ISWCS 2013)*, pp. 1–5, Aug 2013.

[13] S. Keshav, "A control-theoretic approach to flow control," *SIGCOMM Comput. Commun. Rev.*, vol. 21, no. 4, pp. 3–15, Aug 1991.

[14] R. Jain and S. Routhier, "Packet trains–measurements and a new model for computer network traffic," *IEEE Journal on Selected Areas in Communications*, vol. 4, no. 6, p. 986995, Sep 1986.

[15] J. Strauss, D. Katabi, and K. F., "A measurement study of available bandwidth estimation tools," *Proceedings of the 3rd ACM SIGCOMM Conference on Internet Measurement. ACM.*, pp. 39–44, 2003.

[16] V. Ribeiro, M. Coates, R. Riedi, S. Sarvotham, B. Hendricks, and R. Baraniuk, "Multifractal cross-traffic estimation," *Proceedings of ITC Specialist Seminar on IP Traffic Measurement*, 2000.

[17] M. Jain and C. Dovrolis, "Pathload: A measurement tool for end-to-end available bandwidth," *Proceedings of Passive and Active Measurements (PAM) Workshop*, pp. 14–25, 2002.

[18] L. Cottrell, "pathchirp: Efficient available bandwidth estimation for network paths," Apr 2003.

[19] N. Hu and P. Steenkiste, "Evaluation and characterization of available bandwidth probing techniques," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 6, p. 879894, Aug 2003.

[20] M. Li, M. Claypool, and R. Kinicki, "Wbest: A bandwidth estimation tool for IEEE 802.11 wireless networks," *2008 33rd IEEE Conference on Local Computer Networks (LCN)*, Oct 2008.

[21] E. Goldoni, G. Rossi, and A. Torelli, "Assolo, a new method for available bandwidth estimation," *2009 Fourth International Conference on Internet Monitoring and Protection*, 2009.

[22] W. Li, R. K. P. Mok, D. Wu, and R. K. C. Chang, "On the accuracy of smartphone-based mobile network measurement," *2015 IEEE Conference on Computer Communications (INFOCOM)*, Apr 2015.