Defining and Understanding Smartphone Churn over the Internet: a Measurement Study

Árpád Berta
University of Szeged, Hungary

Vilmos Bilicki
University of Szeged, Hungary

Márk Jelasity
University of Szeged and MTA-SZTE Research Group
on Artificial Intelligence, Hungary

***Abstract*—Smart portable devices form the largest personal computing platform so far in human history, yet the adoption of P2P techniques has been very slow. One reason is the lack of a detailed understanding of the dynamic patterns of network connectivity and battery usage. For example, we know that when a smartphone is on a charger connected to a WiFi network behind a friendly NAT device, it can act as an unrestricted P2P node. However, we do not know how to model these "good" intervals and so we do not know what P2P applications are possible at all if we restrict participation to only such intervals. This raises a problem similar to modeling churn in classical P2P research. We are not aware of any suitable and available measurement data sets or models. To address this problem, we developed a publicly available smart phone app that provides the user with information about the current network connection such as NAT type, public IP, and so on. The app also collects data about network connectivity and battery status in the background. The app has been downloaded by several hundred users from all over the world. Based on this data we identify and model the sessions during which a phone can participate in a P2P protocol. We also demonstrate through the simulation of gossip protocols that it is feasible to develop smartphone-friendly P2P applications. The raw data is also available for research purposes in an anonymized form upon request.**

Keywords—smartphones; network address translation; churn measurements

I. Introduction

Smart portable devices represent a seemingly ideal platform for P2P protocols for a wide range of applications, yet the adoption of P2P technology has been very slow. The most important reason is the ubiquitous availability of cloud technology that is also very cheap or even free for small scale deployments. At the same time, while smart devices are capable and well connected computers, energy is a scarce resource. As a result of these two factors, current trends point into the direction of offloading not only storage, but also computation and various functions [1] (more recently even P2P communication [2]) to the cloud in order to save energy.

However, mobile platforms are still well worth considering for P2P applications. At a large scale, cloud services do become expensive, which prevents experimental or non-profit applications from growing. Besides, relying on central services raises privacy and security issues. At the same time, energy is not a real issue when the device is connected to a charger. If an unlimited network connection (e.g., WiFi) is also available, the device can act as a fully capable P2P node equivalent to a desktop computer. Indeed, some attempts have been made to incorporate mobile nodes into P2P networks as fully functional nodes [3].

One of the most important open problems, however, is to understand the patterns of availability, that is, to build models of churn which reflect the intervals when a given device could potentially participate in a P2P protocol. Understanding these patters could answer the question of what applications are feasible without cloud support, and what applications are hopeless. It would also make it possible to design specific algorithms to maximize the utility of the available time of the devices.

For desktop systems thorough churn studies are available [4] but, as we will show, these are not applicable for smart devices. There are numerous data collection efforts related to energy usage [3], [5] or data traffic [6] but these are not sufficient to enable P2P research. There are generic data collecting platforms, the closest to our work is Device Analyzer [7]. However, it does not detect NAT type, which is a crucial part of P2P communication models; it only logs locally available data. Besides, the incentive model builds on the desire of users to contribute to scientific projects. As opposed to this, we wish to offer useful functionality as the main incentive for people to download and use our client [5].

Our contributions are twofold. First, we developed and deployed an Android app that collects data covering all aspects that are relevant to the design of P2P protocols over networks of smartphones, including time series of network and battery status complete with information on NAT type, network type, and network provider. We make this data available upon request in an anonymized form. This allows the research community to design and validate, for example, realistic simulation models. Second, we motivate and analyze a time-inhomogeneous Markovian model of churn that predicts the length of the next session based on the length of the previous session and the time of day. We validate our model comparing it to the trace we collected.

II. Collecting the Data

We developed an Android app[1] that informs the user about the current network environment of the phone: private and public IP, NAT (network address translation) type, and other details.[2]

We did not advertise our app actively. Most of the users in our survey installed our app voluntarily because it provided utility for them. About 30 users were local students recruited

[1] What is My IP & NAT – STUNner, http://futurict.szte.hu/mobilecloud/en/what-is-my-ip-nat-stunner/

[2] The type of the NAT is detected with the help of the STUN protocol using a public STUN server. We used the the implementation http://jstun.javawi.de.

978-1-4799-6201-3/14/$31.00 ©2014 IEEE

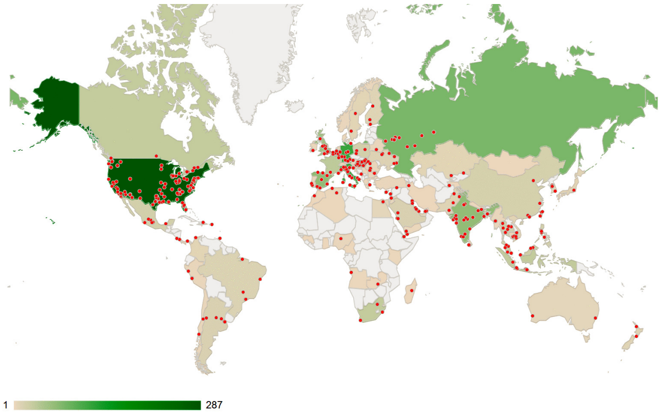

Fig. 1. Locations of the contributions to our data set. The coloring indicates the number of different network providers we collected NAT data about.

for the survey. They installed the app but they received no further instructions.

The data is collected by a background service that can be disabled by the user at any time. This background service listens to various events broadcast by Android that are related to the network interface and the status of the battery. In particular, it listens to TELEPHONYMANAGER, WIFIMANAGER, and BATTERYMANAGER. When such an event arrives, or when the user explicitly runs the app, it collects the status of the network and the battery and logs this information. The app also collects information periodically every 30 minutes. The network properties we collect include network type (WiFi/cellular), carrier, signal strength, bandwidth, public and private IP, NAT type, etc. About the battery we store the temperature, voltage, load percentage, health, charging status (from AC/USB/WiFi), etc.

The data is timestamped using the UTC real-time clock of the phone, along with time zone information so that the local time can be calculated. The data is periodically uploaded to our server in an anonymized form. Further details can be found in the technical documentation of our app. The raw data trace is publicly available.[3]

The devices are identified by a 128-bit random number that is generated during the installation of the application. So far there have been 622 installations to different mobile phones. We collected data in 91 countries about 1425 different networks.[4] The geographic distribution is illustrated in Figure 1.

III. PATTERNS OF AVAILABILITY

We restrict our data set to those continuous measurements that cover at least one day without interruptions. This is to reduce the bias introduced by short measurement intervals due the diurnal pattern in the data.

A. When is a Phone Available?

We model each phone as a series of alternating *available* and *unavailable* sessions. Intuitively, the available sessions are those during which the phone can participate in a P2P protocol.

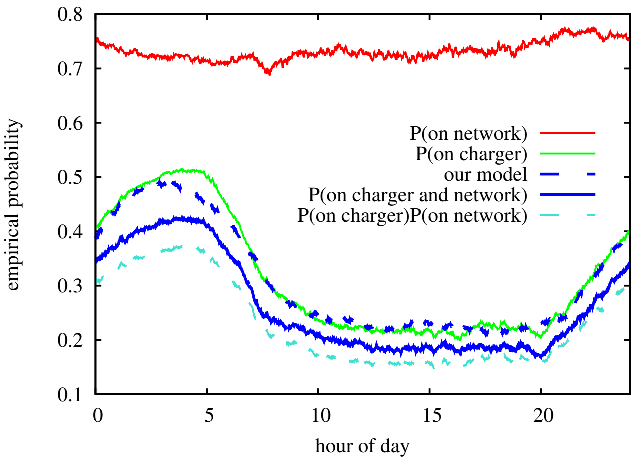

Fig. 2. Diurnal pattern of availability. The plot shows the proportion (empirical probability) of different types of phones, as well as the prediction of our churn model, as a function of the hour.

As opposed to desktop systems, in the case of smartphones the battery state is more important for determining availability than network connectivity. P2P applications crucially rely on bi-directional communication that costs energy, a precious resource on mobile devices. As shown in Figure 2, phones have network connectivity around 75% of the time. Recall that this is a statistic over traces that are at least a day long, which indicates excellent connectivity in general. This is complicated somewhat by taking NAT types into account, as described in Section V. Still, the major problem is energy, not communication.

Based on these observations, we say that the peer is available if it has network connectivity and if it is on a charger. In this work we do not differentiate between network types (WiFi or 3G) and charging types (USB or AC), although this finer grain analysis is also possible based on our data. Figure 2 illustrates that being on a charger is more decisive. It is interesting to observe that being on a charger and on a network are not independent properties. As seen in the figure, the observed probability of the co-occurrence of these two properties is higher than predicted by an independence assumption. In other words, being on a charger increases the probability of the phone being connected to the Internet.

B. Modeling Availability

We model a user j as a series of alternating available and unavailable sessions denoted as

$$\ldots, a_{i-1,j}, u_{i,j}, a_{i+1,j}, u_{i+2,j}, \ldots \qquad (1)$$

where $a_{i,j}$ and $u_{i,j}$ denote the length of the ith available or unavailable session of user j, respectively. Let $t_{i,j}$ denote the starting time of the ith session of user j. Note that $t_{i+1,j} = t_{i,j} + x_{i,j}$ (where $x = a$ or $x = u$, depending on i).

We want to stochastically model this series based on our measurement data so that we could generate traces for the purposes of simulation. More precisely, for a given i and j, we wish to learn the distribution of $a_{i,j}$ (or $u_{i,j}$, respectively). In

[3] STUNner-1 trace at http://futurict.szte.hu/mobilecloud/en/what-is-my-ip-nat-stunner/trace/

[4] Based on the AS number and the city determined by the public IP address using the service provided by Telize: http://www.telize.com/geoip/

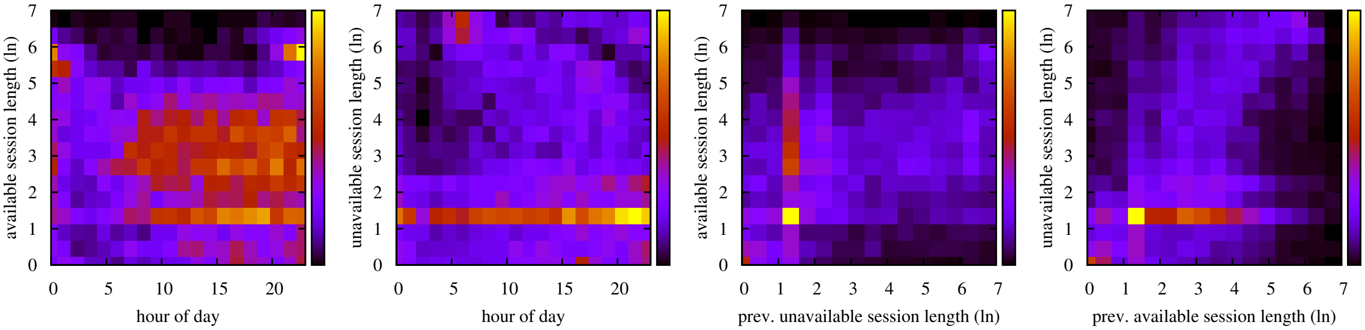

Fig. 3. Conditional distributions of the logarithm of available and unavailable session lengths with the hour of day and previous session length as conditions. In the heatmaps warmer (lighter) colors indicate higher values. The original session lengths are measured in minutes.

the general case, this probability distribution can be formulated as the conditional distribution

$$P(a_{i,j}|t_{i,j}, u_{i-1,j}, t_{i-1,j}, a_{i-2,j}, t_{i-2,j}, \ldots). \qquad (2)$$

A similar formula for $u_{i,j}$ can be given.

Clearly, this expression has too many parameters to approximate so we introduce a time-inhomogeneous Markovian model that considers only the length of the previous session and the starting time of the session:

$$P(a_{i,j}|t_{i,j}, u_{i-1,j}), \text{ and } P(u_{i,j}|t_{i,j}, a_{i-1,j}). \qquad (3)$$

In principle all users j will have their own specific versions of these two distributions. However, since we have been collecting data for only a month at the time of writing, we do not have sufficient data for most of the users. For this reason, in this work we assume that all the users have the same distributions. In other words, we create a model of an average user. Note that the data will make it possible in the future to identify user types and thus to create more specific models involving mixtures of users.

To motivate why we need to keep the previous session length and the starting time as conditions, Figure 3 illustrates the distributions $P(\ln a_{i,j}|t_{i,j})$, $P(\ln u_{i,j}|t_{i,j})$, $P(\ln a_{i,j}|u_{i-1,j})$, and $P(\ln u_{i,j}|a_{i-1,j})$, respectively. To be more precise, instead of absolute time we plot the dependence on the hour within a day as the condition, since the dependence on time is mainly due to the diurnal pattern of phone usage behavior. Note also that we work with the logarithm of the session lengths. This is because we observed that the interesting patterns of the distribution are more apparent on the log scale. On the linear scale the distribution appears to be a simple heavy tailed distribution without any apparent structure. For this reason we model the distribution of the logarithm of the lengths and then take the exponential to generate actual session lengths during simulation.

The distributions clearly show complex patterns. For example, it is clear that after 8pm many long available sessions start that last approximately until the morning. Similarly, in the morning at around 6-7am many long unavailable sessions start. This has to do with the fact that many phones are left on a charger during the night. Figure 2 also supports this interpretation. Also, most of the sessions are rather short, only a few minutes long. After inspecting the data in more detail, we hypothesize that this behavior is mostly the result of a weak unreliable WiFi or mobile network signal that causes quickly alternating short sessions, as implied by the rightmost two plots in Figure 3.

In our model, instead of introducing a parametric approximation of (3), we kept the original data (session lengths classified by starting time and previous session length) and resampled these classes when generating the next session length. In order to have enough data in each class, we reduced the resolution of the time and session length parameters that condition the distributions. As of time, we differentiate between 8 different intervals dividing the 24 hours into 3 hour intervals starting at midnight. As of previous session length, we define three different intervals over the logarithm of the lengths heuristically based on the observed distributions in Figure 3. These three intervals are $[0 : 2)$, $[2, 5)$ and $[5, 9]$. These two low resolution variables define $8 \cdot 3 = 24$ classes for both available and unavailable sessions.

Figure 2 shows the observed proportion of available phones when using the model. We modeled a network of 1000 nodes for 1000 days and calculated the statistics based on this. Note that the model does not control the proportion of available nodes directly; it is an emergent property that is suitable for validating the model. We can see that our predictions are slightly higher than the actual observed proportion. This is due to the fact that currently we are not always able to account for intervals when the phone is switched off and therefore the lengths of unavailable sessions are slightly underestimated. By taking into account only long continuous measurement intervals we minimize this effect, but in next versions of the client this aspect will be improved.

IV. An Example Application

Here we illustrate the application of our churn model[5] via the simulation of a simple push-pull gossip broadcast protocol. Our motivation is to shed light on the importance of a realistic churn model in understanding the behavior of fully distributed protocols. As we will see, in our model gossip behaves in a radically different way than in simpler churn models.

[5] http://rgai.inf.u-szeged.hu/download/p2p14/src.zip

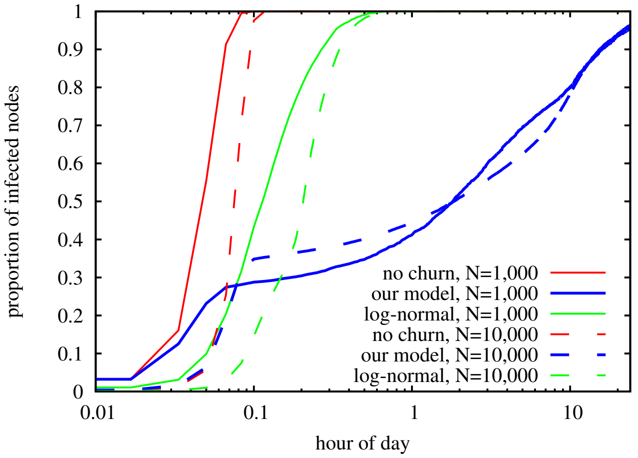

Fig. 4. Push-pull gossip broadcast under different churn models.

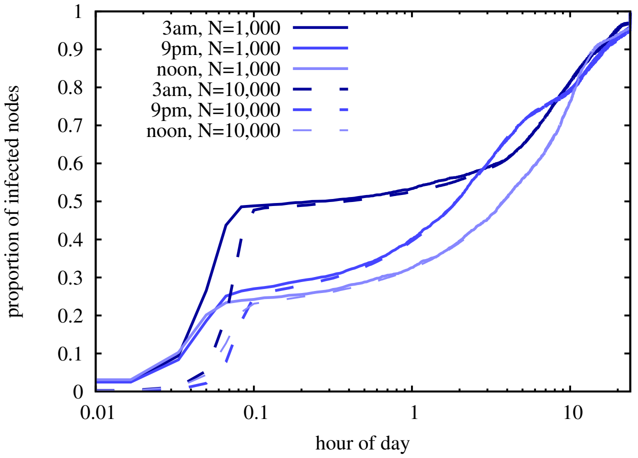

Fig. 5. Push-pull gossip broadcast started at different hours, using our churn model.

The protocol we simulate is a classical push-pull gossip broadcast protocol over a static overlay network. That is, each node has a fixed set of neighbors throughout the simulation. Each node i contacts one random available neighbor j (if there is one available) in each round and if i has the update, it sends it to j, and if j has the update, it sends it to i. The round length is one minute. Initially, one random node has the update. We make sure this initial node is available (online) at the start of the broadcast. If not otherwise stated, the plots show the average of 1000 runs and are based on a random 20-out overlay topology (each node has 20 random out-neighbors).

In the first set of experiments, we compare three churn models: no churn, log-normal churn (a classical model generally used in P2P simulations [4]), and our model. The parameters of log-normal churn were set so that the average and the variance of the available session lengths is the same as in our model, and we also made sure that the average proportion of available nodes (that is kept constant) matches the daily average of our model as well. The log-normal model works by drawing the available session lengths from the log-normal distribution, and in addition if the proportion of available nodes drops below the fixed threshold, unavailable nodes are made available at random.

Figure 4 shows the results for two different network sizes. It is striking how different the dynamics are from the more homogeneous log-normal model. The reason will be clearer considering our next set of experiments, where we started the broadcast at specific hours. Figure 5 shows the results. It is clear that the broadcast reaches the nodes that are available at the start of the broadcast in a few rounds (see also Figure 2). Then progress slows down since new nodes join slowly. However, clearly, spreading speed also depends on the distribution of session lengths in the respective time intervals during spreading (see Figure 3).

We also include a validation experiment, during which we simulate churn based on the exact trace we collected. That is, we select those users for which we have at least a 3-day long continuous measurement interval. We found 140 such users. In our simulation we simulated the exact availability of these users as recorded in our data. In Figure 6 we compare this trace-based simulation to the churn models we examined in Figure 4. Here we used a 10-out random topology, and the plot shows the average of 3 days where spreading is started at midnight. The trace-based simulation closely follows our time-inhomogeneous Markovian model, which supports the validity of the approach.

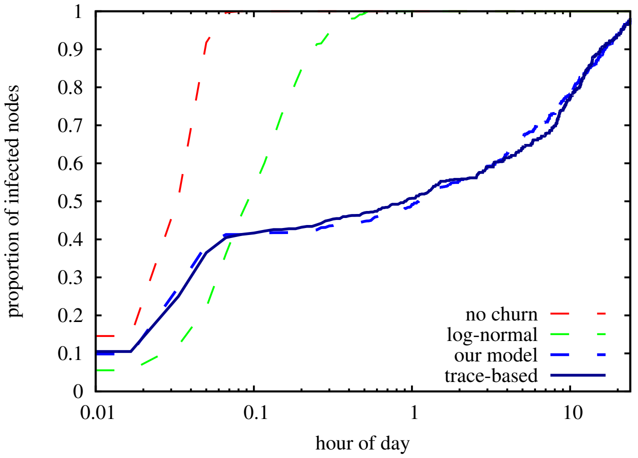

Fig. 6. Validation experiments. Push-pull gossip broadcast under different churn models including a trace-based simulation (N=140 for all churn models).

V. NAT Type Distribution

Since one of the unique aspects of the data we collected is that it covers NAT type as well, we present briefly some interesting statistics about NATs here. Although in many cases NAT devices can be dealt with via low level "hole punching" solutions [8], they can also represent design constraints at higher levels due to the potentially high cost of (let alone the complete lack of) such solutions [9].

The results of our measurements show similar trends to that of available measurements [10] in the predominantly desktop P2P ecosystem as shown in Figure 7. The chart is based only on successfully identified "classical" NAT types (in order to allow comparison with earlier desktop data), 10% of the

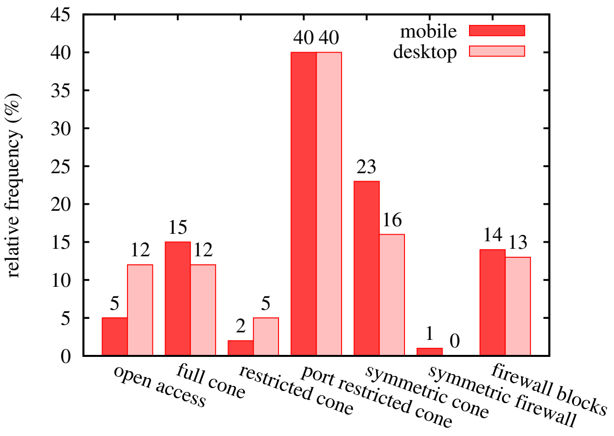

Fig. 7. Relative frequency of NAT types in use aggregated over time.

identification attempts were unsuccessful. We used only those continuously measured time series that cover at least one day for a phone without interruptions to restrict bias due to diurnal patterns in data. The most significant difference is the very low percentage of open access peers and a higher percentage of the symmetric cone NAT type. That is, smartphone users are somewhat more restricted than desktop users.

In P2P applications we are interested in the probability of a successful connection establishment. In general we find that if we take a random pair of nodes that have NAT types from the distribution we observed, the probability that the pair can communicate without a relay (but possibly needing hole punching techniques) is at least about 60% using standard techniques [11]. However, this figure largely depends on the methods and utilities that are applied for hole punching, and sophisticated state-of-the-art technology can improve this significantly [8].

VI. Conclusions

In this paper we argued that it is important to understand availability patterns of smartphones in order to assess the feasibility of P2P techniques. We considered a smartphone available when it is on a charger and connected to a network at the same time.

In order to be able to model the availability patterns of devices, we implemented an Android app to collect data. We proposed a time-inhomogeneous Markovian model based on the collected data in which the conditional probability distributions of session lengths are captured by a set of the actual observations in the data that we resample when creating synthetic traces of users to model churn.

We validated this model in multiple ways (see Figures 2 and 6). We found that the model captures observed availability as well as the behavior of push-pull gossip broadcast. The model can be enhanced further in the future, as more data becomes available, by explicitly modeling different user types as well as incorporating NAT types that are also captured by our data. This would open up the possibility for very large scale realistic simulations.

While for many P2P applications (especially for those that execute distributed data processing in the background) this model of availability is sufficient, for others (especially for highly interactive ones) the requirement of being on a charger is too restrictive. Users typically use their devices when those are not on chargers. Indeed, we have seen that availability peaks in the middle of the night, when users are probably asleep and leave their devices on the charger for the night. However, this is not our main point. Since our data is available upon request, it becomes possible for the community to explore many different definitions of availability, as well as to explore applications that can tolerate or even exploit these availability patterns.

Acknowledgments

M. Jelasity was supported by the Bolyai Scholarship of the Hungarian Academy of Sciences. This work was partially supported by the European Union and the European Social Fund through project FuturICT.hu (grant no.: TAMOP-4.2.2.C-11/1/KONV-2012-0013).

References

[1] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, “A survey of mobile cloud computing: architecture, applications, and approaches,” *Wireless Communications and Mobile Computing*, vol. 13, no. 18, pp. 1587–1611, 2013.

[2] S. Kosta, V. C. Perta, J. Stefa, P. Hui, and A. Mei, “Clone2clone (c2c): Peer-to-peer networking of smartphones on the cloud,” in *Proceedings of the 5th USENIX Workshop on Hot Topics in Cloud Computing*. San Jose, CA: USENIX, 2013.

[3] R. Jimenez, G. Kreitz, B. Knutsson, M. Isaksson, and S. Haridi, “Integrating smartphones in spotify's peer-assisted music streaming service,” KTH, Stockholm, Sweden, Tech. Rep. diva-134609, 2013. [Online]. Available: http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-134609

[4] D. Stutzbach and R. Rejaie, “Understanding churn in peer-to-peer networks,” in *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement (IMC'06)*. New York, NY, USA: ACM, 2006, pp. 189–202.

[5] D. Ferreira, V. Kostakos, and A. K. Dey, “Lessons learned from large-scale user studies: Using android market as a source of data,” *International Journal of Mobile Human Computer Interaction*, vol. 4, no. 3, 2012.

[6] Y. Jin, N. Duffield, A. Gerber, P. Haffner, W.-L. Hsu, G. Jacobson, S. Sen, S. Venkataraman, and Z.-L. Zhang, “Characterizing data usage patterns in a large cellular network,” in *Proceedings of the 2012 ACM SIGCOMM Workshop on Cellular Networks: Operations, Challenges, and Future Design (CellNet'12)*. New York, NY, USA: ACM, 2012, pp. 7–12.

[7] D. T. Wagner, A. Rice, and A. R. Beresford, “Device analyzer: Large-scale mobile data collection,” in *Workshop on Big Data Analytics*, 2013.

[8] R. Roverso, S. El-Ansary, and S. Haridi, “NATCracker: NAT combinations matter,” in *Proceedings of 18th Internatonal Conference on Computer Communications and Networks (ICCCN)*, August 2009, pp. 1–7.

[9] R. Roverso, J. Dowling, and M. Jelasity, “Through the wormhole: Low cost, fresh peer sampling for the internet,” in *Proceedings of the 13th IEEE International Conference on Peer-to-Peer Computing (P2P 2013)*. IEEE, 2013.

[10] L. D'Acunto, J. A. Pouwelse, and H. J. Sips, “A measurement of NAT and firewall characteristics in peer-to-peer systems,” in *Proc. 15-th ASCI Conference*. Advanced School for Computing and Imaging (ASCI), June 2009, pp. 1–5.

[11] Y. Takeda, “Symmetric NAT traversal using STUN,” Internet Engineering Task Force, Internet Draft, 2003. [Online]. Available: http://tools.ietf.org/id/draft-takeda-symmetric-nat-traversal-00.txt