Measurements with the masses

Michael Faath, Rolf Winter
University of Applied Sciences Augsburg
michael.faath@hs-augsburg.de
rolf.winter@hs-augsburg.de

ABSTRACT

The network measurement community has produced an impressive set of network measurement tools to which a number are added each year. A few prominent examples have seen wide-spread deployment covering large parts of the Internet but there is arguably one network segment which is extremely difficult to cover: host-to-host measurements with measurement instrumentation on "regular" end-devices. The problems in this space are manifold, but the main challenge is non-technical: incentivizing "regular" Internet citizens to take part in an experiment. In this paper, we look at the challenges, examples of measurements which involved end-users and conclude with potential actions which would make measurements involving end-users easier (but by no means easy).

1. INTRODUCTION

A large set of network experiments have been carried out in the past on networks which are open for experimentation, where a large set of end-devices are under the control of an experimenter but which are neither attached to "regular" Internet access networks nor are these devices "common" end-user devices. A prominent example of such a network is PlanetLab[1]. For many experiments such a network is not only sufficient but ideal because of the tight control and root access on these devices. For other experiments, these environments are not suitable because the Internet experience for real Internet users cannot adequately be captured.

There are many factors which influence the user-perceived Internet quality and which therefore also influences network measurements. For example the type of connection used (like WiFi, 3G/4G, DSL or cable) or cross traffic generated by other users in e.g. a home network or applications generating traffic in the background (like updates from an application or the system) influence a user's experience. Also the workload and system limitations of the end device, or the workload of the requested service or the used path in the Internet all influence the end users' Internet experience, albeit not all of these factors are actually due to characteristics related to the network.

These conditions of course can be simulated in a controlled environment once their parameters are known, but having an experimentation platform with real end-users allows to analyze the actual end-user experience with all foreseen and unforeseen problems. It also allows to assess how the Internet performs from a regular Internet user's point of view and measurements include all potential network segments starting from the home and access network. On the other hand, involving real end-users in network measurements is challenging to say the least and the reasons are manifold —many of which are addressed in the following.

2. CHALLENGES

The following list of challenges is certainly not complete nor is the ordering to be understood as a prioritization. Also, for some measurements a certain challenge might not apply.

2.1 Access-rights on end-devices

A number of measurement tools need super-user access on an end-device in order to have access to low-level system functionality. This is difficult to explain to regular end-users since it would potentially allow an application to see all traffic sent by a device and access all data and functions available on the device without further consent, which is unusual for typical applications installed by end-users. Usually, the operating system issues a type of warning or notification to the end-user when an application requires such privileges. This would require immense trust in the application developer which cannot generally be expected.

In addition, on some mobile operating systems an app developer has to specify which capabilities are accessed and the user has to grant those when installing the app or once the app needs to use them. If a network measurement app for a mobile platform makes excessive use of the available capabilities for the sake of better measurement results, a user might be reluctant to install this app altogether. At least a detailed explanation of why a capability is needed has to be provided, but it is questionable if a user will even read such information. Therefore, the less privileges an application needs, the higher the chances an application will be installed by a user.

[1] `https://www.planet-lab.org/`

2.2 Cross-platform development

A large set of operating systems are being used by end-users. Of course there are certain mainstream platforms which would already cover a significant potential user-base, but many of these platforms cover a narrow device spectrum such as Android for example. Therefore, to not only increase the potential user-base but to also cover a larger set of device classes and potential network segments, a measurement tool should be developed cross-platform to the largest extent possible.

A simple answer seems to be to use a language like Java, but there are often drawbacks such as access to low-level (non-root) features that are available to languages such as C++ and low-level access is often required for network measurements. This aspiration to develop cross-platform however comes with significant development costs. It means that a GUI version needs to be developed with multiple versions of the GUI for different screen resolutions and input methods (e.g desktop vs. mobile). It also means that the measurement code might differ from platform to platform. For example Mac OS allows opening non-privileged ICMP sockets which can be used to implement a non-root ping which is not trivially possible on other platforms. In addition, certain APIs/frameworks/libraries like PCAP might not be available on all desired platforms. In summary, cross-platform development is a very costly and difficult process.

2.3 Incentivizing the user

A potential end-user of a network experimentation platform needs some form of motivation to participate. Such platforms can be divided coarsely into two groups: measurements as the main cause of the application and measurements as a by-product of the application.

In the first case, the motivation for a user can e.g. be to use the platform for troubleshooting a problem he or she is experiencing while using an Internet service. This has the interesting side-effect of executing measurements under conditions which a user considers as problematic and the application can try to find the root cause for the perceived problems and provide possible solutions to the user. Other users might use the platform to have a look at measurement results for their own device or network out of curiosity (e.g. performance comparisons of operators). This might only be of interest to users with a technical or scientific background, but it can be an important factor if the gathered data can be visualized in an appealing and informative or even educational fashion for non tech-savvy users.

In the second case—measurements as by-product—the user installs an application which offers some service or information other than measurements, for example a game. By using the app the user grants the permission to also do network measurements, therefore the experimenter incentivizes the participant by providing another benefit.

In both cases it can be helpful in terms of motivation to provide global statistics and insights gained with the experiments. These statistics can e.g. be used to "gamify" the application[1]. A number of games and platforms e.g. motivate users by providing a form of score-board by introducing a kind of points system or "achievements" depending on the rate of participation so he or she can compare himself or herself to other users.

End-user participation is arguably the most challenging part about deploying an end-user-based measurement application. There are successful projects that have involved a significant number of end-users participating in projects without providing a direct benefit, i.e. the users support the cause of the project alone. BOINC[2] or Folding@home[3] are prominent example of this, but replicating this for network measurements is certainly difficult since many of these projects have significant, humanitarian goals.

2.4 Potential interference

Devices of end-users are obviously used and potentially heavily so. Therefore, the measurement application has to make sure that the measurement results are accurate and not an artifact of other factors such as device usage or system limitations or at least label the results accordingly. Other factors can be cross traffic generated by other users in the network, or other applications and services running on the same device either in the background or in the foreground, for example when the user is browsing in the Internet. But not only network utilization needs to be taken into consideration, it is also possible that a device has very limited processing power or memory and a measurement cannot be executed in time or at all because of that.

As described in [2] cross traffic in the local network can easily be detected if the Internet gateway device offers traffic counters for the WAN interface queryable by a uPnP request. Monitoring the processor and memory utilization on the other hand is more difficult if multiple platforms should be supported because the retrieval of such information is different on many operating systems.

Generally, the control over or the monitoring of the local network and device are important when the accuracy of measurement results are to be ensured. Therefore, additional instrumentation might be needed for a measurement application on end-devices.

2.5 Availability

End user devices do not necessarily run all the time. They might be turned off, especially at night. Even if using dedicated probes like RIPE Atlas[4] deployed in a home network it might not be possible to execute measurements all the time. Some users turn off the Internet gateway device automatically at night or if not used for a longer period of time. If the device uses a mobile connection like WiFi or 3G/4G it is also possible that an Internet connection is unavailable

[2] https://boinc.berkeley.edu/
[3] https://folding.stanford.edu/
[4] https://atlas.ripe.net/

for shorter periods of time. The measurement application should be prepared to be shut down unexpectedly and without warning, also measurement results—if they can be gathered at all—need to be cached until Internet connectivity is available again.

Another aspect which has to be taken into consideration is what happens if the network interface or type of connection changes while doing a measurement. This can be the case if the device is a mobile device like a smartphone or notebook and switches between LAN, WiFi or 3G/4G. If a user uses a mobile data plan with limited traffic it is also important to inform the user about traffic consumption and offer an option to prevent measurements to be executed when using those connections. Also, if a series of measurements are executed before and after a connection has been throttled by an operator due to exceeding the data plan's traffic budget, the results might be affected by that.

Another problem that influences the availability of devices is the prevalence of NAT. When designing the measurement infrastructure, the presence of NAT needs to be taken into account. E.g. end-to-end measurements, i.e. measurements between two clients running the application can be difficult when both are behind a NAT and a NAT traversal mechanism should be provided.

Availability is a problem that affects many aspects of a measurement application and the whole platform used to coordinate measurements and to collect measurement results.

2.6 User privacy

Very early in the process of developing a network measurement platform one has to carefully consider data privacy issues. In particular, this includes which data to gather, identifying, discarding or anonymizing personally identifiable information, and how to store results in a law-abiding way [3].

Many users might have very understandable resentments to install a measurement application because network measurements can be misinterpreted as some kind of surveillance—in particular after Snowden. This is another social, but even more so a legal issue. Also, legislation regarding data privacy is different across countries and therefore a good guiding principle is to collect as little information as possible and to inform the user well what kind of information is being collected and why. In particular, it is necessary to make it clear that no personally identifiable data is being gathered and to provide an explanation of the experiments for a non-technical audience.

Because legal issues are possible, a lawyer and data protection professional needs to be involved. An EULA needs to be carefully formulated which should be stable, because changing EULAs are not only an annoyance but (rightfully) a cause for suspicion by users. This ties down the measurement tool significantly and future extensions will be difficult. Therefore, this is a key step and often the most difficult for the experimenter itself.

2.7 Measurement coordination

For certain measurements, in order to make them comparable, to achieve a certain order of measurements from different vantage points, or to make sure that end-devices not synchronize measurements, a coordination of the execution times is needed. User devices are likely not time-synchronized using e.g. an NTP server, so the offset to a known time reference should be calculated to store the correct execution time of a measurement (using UTC as time-zone). A re-calculation of the offset needs to be done on a regular basis in case the measurement application runs over longer periods of time due to potential system clock drift.

As mentioned, coordination is also important to make sure that e.g. measuring against one central end-point does not overload this end-point due to synchronized measurements. For one, this influences measurement results, but this might also be identified as a distributed denial of service attack, which could lead to widespread filtering of the apps traffic. To handle this, measurements could for example be executed with a random offset on each device.

The need for coordination differs depending on the run-time characteristics of measurements: continuously, repeated, or one-shot. In the first case, a measurement runs as long as the measurement application is executed, for example a passive observation of network traffic or a continuous ping. To the second type belong measurements which are executed regularly, influenced by a schedule provided by the platform operator. The last case only executes measurements once, possibly triggered by an event (like the user telling the application to measure now or after automatically detecting a problem in the network). But many measurements will certainly need some form of coordination.

2.8 Operations

The goal of any measurement platform involving end-users is certainly good coverage, which includes geography, ASes, access network types, device types and demographics. This ultimately means that the operational system will eventually need to handle a large user base which translates to a lot of coordination, failure, updates, a huge amount of data including a huge amount of noise.

The potential scale of such a system is difficult to handle and difficult to operate as a research group both financially and in terms of developer resources. There is little return on investment as an academic on the tedious maintenance and operational aspects of such a system, in particular given all the challenges described so far and the marginal outlook on success.

Success in that respect is a blessing and curse at the same time. Unless the application has a one-shot measurement nature, data retention, archiving and other data management task will increasingly become a problem and need to be planned well ahead. Updating the app, a kill-switch for network measurements and other vital operational mechanisms need to be thoroughly tested before going live.

3. CONCLUSION

There already have been a number of successful measurement platforms or applications involving end-users. Out of these Dasu[2] (see also references therein for a good overview of other tools and platforms) particularly stands out as it has solved the problem of incentivizing the end-user and has activated over 100,000 end-users as participants. Dasu can be installed as a stand-alone application (where support for the Dasu-cause and altruism are the main drivers for installing it) but it was first deployed as a plugin for a popular BitTorrent client leveraging the popularity and the deployed base of the application. Measurements here were a by-product, but Dasu also performed active measurements to characterize the user's ISP.

We faced many of the challenges above while building an end-user-based network measurement platform ourselves called GLIMPSE[5], which is currently in a closed beta phase, for now only deployed on embedded, dedicated probes. The development process was challenging and very time consuming and the most challenging bit is still ahead—convincing end-users to install the app. The value proposition is simple: identifying potential network issues. Other than that, the app offers the implemented network measurements as built-in tools for the user to trigger measurements him or herself, targeting "advanced" users. That leaves the cause of the platform for the altruistic, cause-driven users: we build an application to "measure the Internet", where measuring the Internet means our measurement coordinator triggers measurements, or series of measurements to capture certain Internet-characteristics as part of a purpose-driven measurement campaign. A user basically donates a certain amount of bandwidth for these measurements per month.

The real question is whether there is something that can be done to make end-user based measurements easier, measurements more expressive and accelerate the development of measurement tools. We believe there are things that can be done, non of which are particularly easy but all well worth pursuing. We see these mainly in two areas: protocols and standards, libraries and APIs.

Available protocols and standards constrain what can be measured in many respects. It is quite difficult to do very trivial measurements today. E.g. measuring the return path from a destination back to the source typically requires control over the destination. Allowing more measurements to be taken is of course a careful consideration between the value a new measurement gives and the potential dangers a new measurement introduces such as potential misuse for Denial of Service attacks. There have been efforts in the past to introduce new protocols such as the IP Measurement Protocol (IPMP)[6] but introducing new measurement capabilities is quite difficult. Instead of introducing new protocols, it would be possible to extend protocols such as ICMP using new types but that as well has not happened recently. At least type ranges have been set aside for experimentation by RFC4727. In essence, standardizing protocols or protocol extensions for use in end-user measurements is very difficult to say the least but would massively help to gather a better understanding of the Internet and to better troubleshoot problems experienced by end-users. On the positive side, the IETF is in the process of developing a standard for measurement coordination in the LMAP working group, which could at least harmonize this aspect of network measurement coordination.

The second area really concerns the development cost and the time spent developing and testing the measurement application. The development burden could be significantly reduced if libraries would be actively maintained and contributed to for cross-platform development of network measurements. Many of the challenges described above resulted in significant effort working around platform-specific differences. But many of the measurements such as a simple UDP-based ping would certainly be useful in many network measurement contexts. Building up a library and agreeing on APIs would greatly speed up the development process and lower the development costs for future tools to come.

There is probably little that can be done about the operational complexity of a measurement platform involving end-users. Also, the legal side is something that every tool needs to cover individually and no simple recipe exists that is suitable for a wide range of network measurements and experiments. The two areas described above also will need broad consensus as these protocols or protocol extensions will likely have to be implemented by operating system vendors and large support by developers and implementers. It is also worth asking whether certain OS features can be adoted by other operating systems. E.g. non-privileged ICMP sockets as provided by OS X are very handy and do not seem like a overly dangerous security hazard. But this—just as the other things mentioned— requires significant support, interest and time.

[5] `https://www.measure-it.net/`

[6] `https://tools.ietf.org/html/draft-mcgregor-ipmp`

4. REFERENCES

[1] K. Knaving and S. Bjørk, "Designing for fun and play: Exploring possibilities in design for gamification," in *Proceedings of the First International Conference on Gameful Design, Research, and Applications*, Gamification '13, (New York, NY, USA), pp. 131–134, ACM, 2013.

[2] M. Sanchez, J. Otto, Z. Bischof, D. Choffnes, F. Bustamante, B. Krishnamurthy, and W. Willinger, "A measurement experimentation platform at the internet's edge," *Networking, IEEE/ACM Transactions on*, vol. PP, no. 99, pp. 1–1, 2014.

[3] M. Allman and V. Paxson, "Issues and etiquette concerning use of shared measurement data," in *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement*, IMC '07, (New York, NY, USA), pp. 135–140, ACM, 2007.