Challenges and Issues on Collecting and Analyzing Large Volumes of Network Data Measurements

Enrico Masala[2], Antonio Servetti[2],
Simone Basso[1,2], and Juan Carlos De Martin[1,2]

[1] NEXA Center for Internet & Society
[2] Internet Media Group
Control and Computer Engineering Department
Politecnico di Torino, 10129 Torino, Italy
firstname.lastname@polito.it

Abstract. This paper presents the main challenges and issues faced when collecting and analyzing a large volume of network data measurements. We refer in particular to data collected by means of Neubot, an open source project that uses active probes on the client side to measure the evolution of key network parameters over time to better understand the performance of end-users' Internet connections. The measured data are already freely accessible and stored on Measurement Lab (M-Lab), an organization that provides dedicated resources to perform network measurements and diagnostics in the Internet. Given the ever increasing amount of data collected by the Neubot project as well as other similar projects hosted by M-Lab, it is necessary to improve the platform to efficiently handle the huge amount of data that is expected to come in the very near future, so that it can be used by researchers and end-users themselves to gain a better understanding of network behavior.

Keywords: Network Data Collection, Network Data Analysis

1 Introduction

In the study of the performance of the Internet one of the major challenges is how to collect a large number of reliable network measurements from a sufficiently large number of locations in the network. In the recent years, several independent tools have been developed to collect this type of information [1–3].

To study and monitor the performance of Internet access links, we designed Neubot [4–6], a software for distributed network measurements that runs on the user's computer and periodically monitors the performance of its connection to the Internet. The results are collected on the user device and made publicly available on a central server to allow constant monitoring of the state of the Internet by any interested parties.

Since Feb 9, 2012, Neubot has been hosted on Measurement Lab (M-Lab) [7, 8], an organization that provides dedicated server-side resources to open-source network measurement tools, including Neubot and NDT [3]. In turn, the availability of a network of servers around the world allows Neubot to effectively test the performance of the clients broadband access network, by connecting them to the closest M-Lab server available. In addition, in the near future, we will release a mobile version of Neubot that will be available for the increasing number of mobile Internet devices, which is predicted to surpass the number of desktop devices by the end of 2014 [9].

Thus, while Neubot is performing a large amount of measurements each day, the problem of measuring the network is becoming a problem of managing the available data for storage, querying and analysis purposes. For the first two challenges Neubot may take advantage of two services provided by M-Lab, namely the Google Cloud Storage [10] and the Google Big Query platform [11]. However, a more flexible solution is needed to perform a deeper analysis of the measurements and to gain clear, if possible, and real-time insight into the behavior of the Internet and of the Internet connection. of the end users.

The objective is to be able to answer questions such as the following: Is the server on the other end of the connection having a problem? Is my device or modem not properly configured? Is something wrong with the ADSL connection? Is an ISP deliberately interfering with my traffic? Only a flexible platform that can efficiently manage the processing of measures collected by millions of users that share the same (or similar) network path could answer these questions. In fact, benchmarking Internet access link performance cannot be achieved by merely running a single speed test. Speed varies with time and it is affected by a number of confounding factors (i.e., home network cross-traffic, end-host configuration, wireless connection quality) that must be isolated as much as possible by proper data processing and analysis.

The rest of this paper is organized as follows. In Section 2 we present the Neubot architecture for data collection and we describe the tests currently implemented. Information about data query and data analysis is given respectively in Section 3 and in Section 4. Finally conclusions are drawn in Section 5 and possible future work is presented.

2 Architecture for Data Collection

The architecture of Neubot consists of an *agent* that runs on the users' computer, and of a set of servers as shown in Fig. 1. The servers have different roles and may be replicated in different locations of the M-Lab network. We distinguish between the *master server*, the *test servers* and the *archive server*.

The Neubot *agent* runs in background as a system service and periodically check the master server to be informed on the next test to perform and with which test server.

The *master* server acts as a coordinator. Once contacted, it can implement different policies for the coordination of the Neubot agents. For example, for certain kind of tests that aim at measuring the speed of the user's connection to the Internet (e.g., *speedtest*), it can select the test server that is closer to the agent, for another kind of tests, that do not have particular requirements (e.g., *raw*), it can select the test server with the lowest load.

The *test* servers implement one or more transmission tests. First, a negotiation phase assigns a temporary unique identifier to each connecting agent that wants to perform a test, and uses this identifier to manage the queue of incoming test requests, i.e., to schedule each single test when the right conditions are met. Second, a test phase implements the actual transmission test to estimate selected characteristics of the network between the agent and the test server. The measured performance metrics depend on the type of test and are described later. Once the test is completed, the agent uploads the results to the test server, and saves a copy locally.

The *archive* server once a day collects the test results from each test server and makes them available on the web.

M-Lab support The test servers are hosted on the M-Lab network [8] in 114 different servers operating across 32 geographically distributed sites around the globe (as of April 2013).

Each Neubot server instance is allocated dedicated resources on the M-Lab platform to facilitate accurate measurements and perform high-bandwidth measurements at large scale (e.g., each partner research institution or company that hosts M-Lab servers is typically required to provide at least 1 Gbit/s upstream capacity [12]).

The archive server is hosted on the Google Cloud Storage service (similar to Amazon S3) [10]. Currently, Neubot data is uploaded to this public repository

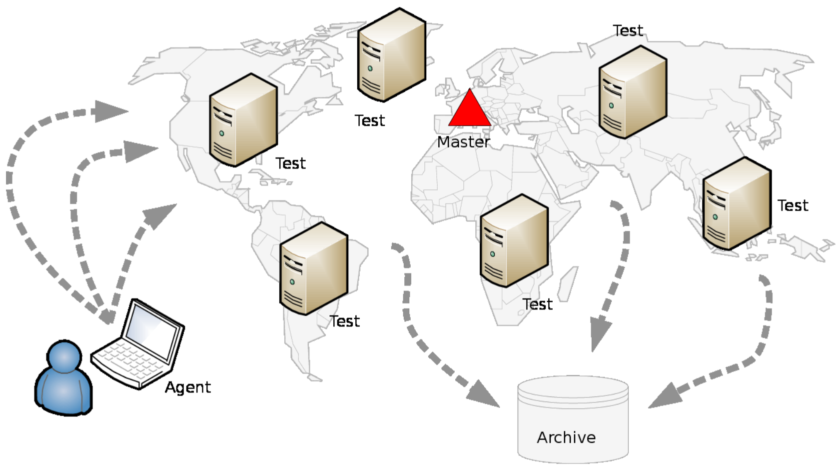

Fig. 1. Neubot architecture. Agents on the users computer are coordinated by the Neubot master server and connect to the preferred test server to perform the actual tests. Results are collected on the archive platform.

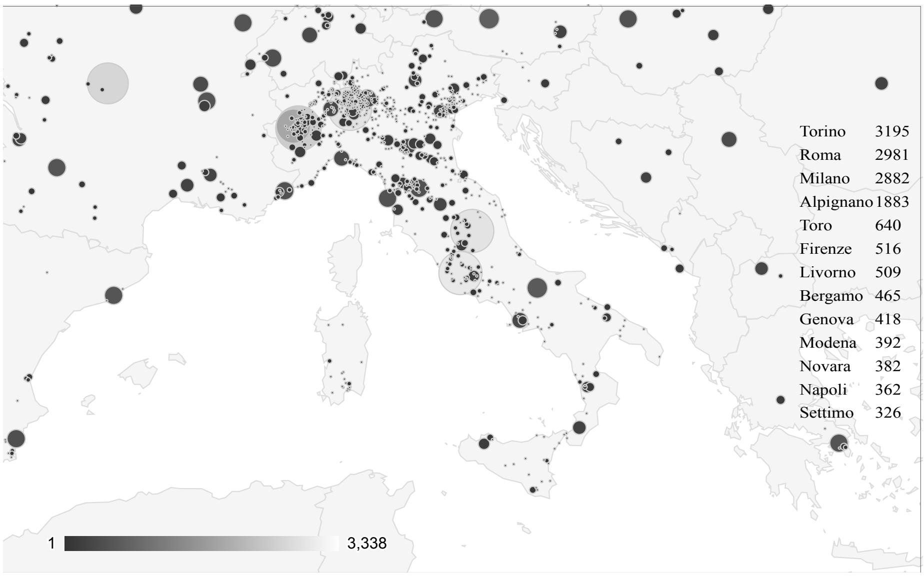

Fig. 2. Geographical distribution of the Neubot clients in Italy on the basis of their IP address. The number of test performed from each site is presented in the table on the right and corresponds to the color and size of the circles in the figure. Figure refers to tests run in April 2013.

in batches once a day. Raw data are organized into tarballs that contain all the data collected during a single day on a single M-Lab server.

In a month about 500'000 tests are run and the collected data sums to about half a gigabyte (gzip compressed). Fig. 2 shows the location of the clients and the number of tests run on April 2013 in Italy. A more extended representation covering all the 10'000+ different IPs and the 2'000+ different locations from which Neubot test have been performed (e.g., Singapore, Rome, Cape Town, Toronto, etc.) can be accessed on the web at the following link http://bit.ly/neubot_gmap. The amount of data collected by Neubot could considerably increase if the project becomes more widely known as, for instance, is the case of the Network Diagnostic Tool (NDT) [3], another tool available on M-Lab, that collects up to 1 TB of compressed data each month.

2.1 Implemented tests

Neubot implements three active network tests: *bittorrent*, *raw* and *speedtest*. Neubot schedules one of these tests at random every half an hour. In addition the user has the possibility of running tests on demand.

The speedtest test emulates HTTP and estimates the round-trip time, the download and the upload bandwidth. The bittorrent test emulates BitTorrent peer-wire protocol and estimates the round-trip time, the download and the upload bandwidth. The raw test performs a raw 10-second TCP download to estimate the download bandwidth, and it collects statistics about the TCP sender by using Web100 [13], which is installed in all M-Lab servers. Test results are saved in JSON format and range from 470 to 30,000 bytes per test (the latter being the raw test that collects data periodically every 10 seconds).

Among the data collected for each speedtest test there are:

uuid Random unique identifier of the Neubot instance, useful to perform time series analysis.

timestamp Time when the test was performed, expressed as number of seconds elapsed since midnight of January 1, 1970.

real_address Neubot's IP address, as seen by the server. It is either an IPv4 or an IPv6 address.

remote_address The server's IP address. It is either an IPv4 or an IPv6 address.

connect_time Round-trip time (RTT) estimated by measuring the time that the connect system call takes to complete, measured in seconds.

download_speed Download speed measured by dividing the number of received bytes by the download time, measured in bytes per second.

upload_speed Upload speed measured by dividing the number of sent bytes by the upload time, measured in bytes per second.

latency RTT estimated by measuring the average time elapsed between sending a small request and receiving a small response, measured in seconds.

2.2 Expected evolution

A significant effort is currently being devoted to port the Neubot client to a mobile platform, i.e., Android. This will greatly extend the amount of collected data since the availability as an app will increase the chance that the application is installed on mobile devices. Similar applications already exists, such as speedtest.net, which had between 10 and 50 millions of downloads from the Android market (i.e. between 1/100 and 1/20 of the number of Android users [14]). If 1 out of 1,000 Android users installs the Neubot application, between 1 and 10 million tests could be performed each day, considering that Neubot run tests periodically. Moreover, new types of tests such as HTTP-based ones will be soon implemented, which will further increase the amount of collected data. To analyze such large amount of data, which will be 100 to 1,000 times the current amount, new analysis techniques will be necessary. The potential issues arising from this scenario are described in the remaining part of this paper.

3 Data Querying

The first objective of a data collection effort such as the described one is to be able to look into the data themselves to gain some understanding about both the data collection process and the statistics extracted from the data. Although currently the amount of data is somewhat limited, i.e., in the order of few gigabytes, it is expected that this amount will increase rapidly as the software used for data collection will be ported to mobile platforms. In such conditions it is very important to be able to monitor some key parameters of the system, such as the number of unique users, how they move, how frequently they change IP (or provider), etc.

The above kind of monitoring, however, cannot be performed on the test servers just after the completion of a network experiment. To avoid to perturb and/or invalidate the network experiments, in fact, Measurement Lab servers are dedicated to the measurement task only. Therefore, the collected data must be moved to a different location (or network), where they are permanently stored, and when they are post-processed to facilitate querying and analysis.

In particular, the current implementation of Measurement Lab moves test results from each test server to a public repository, i.e., the *archive* server, once a day. The repository is hosted by the Google Cloud Storage service. In this way the compressed archives of the test results can be accessed through a web interface or by means of the Google Cloud Storage command line utility.

The designers of Measurement Lab decided to use compressed archives to efficiently collect and transfer the data in a public location, but the whole range of data are only useful to those planning to run an extensive and detailed analysis; for most researchers, instead, aggregated information may be sufficient. Moreover most people cannot afford either the time required to download or the amount of terabytes necessary to store such large amount of data for personal processing. Also, a large amount of processing power is required to manage this volume of data.

Of course, researchers could selectively download only the tarballs to which they are interested, but this is unpractical, because one typically does not know in advance what tarballs contain the information to which he/she is interested. Instead, to help to analyze the collected data, Measurement Lab is currently being enhanced to export results of the network experiments via Google BigQuery, a web service that allows to run interactive analysis (e.g., SQL queries) of large datasets without downloading it [11]. In fact, NDT data is already available through BigQuery, and Neubot data will be available through BigQuery soon.

Interested parties will then be able to query the measurement data in a matter of seconds – even with complex queries – in order to gain a better understanding into Internet operation and performance. For example, the real-time processing of millions of measurements may allow to identify Internet congestion, traffic shaping, or network outages on a world scale.

4 Data Analytics

The large amount of measurement data already available allows to employ data mining techniques to discover correlations among data that would otherwise be difficult to observe. In this section we present an overview of the type of analysis useful for the purpose of analyzing the network behavior both as a whole and with respect to the experience of each single user. Examples of possible analyses are given by showing some small but representative subsets of data.

4.1 Dimensions of the Analysis

First, note that the data presents several dimensions that require careful analysis. The most important ones are highlighted in the following.

time : clients run tests periodically when connected to the Internet. The evolution over time of the measured parameters need to be considered to better understand the situation of the connection and relate it to other clients in similar conditions at the same time instant.

location : the information about the approximate physical location of the client will play a role in understanding if any unusual value detected in the Internet access parameters is due to the location (e.g., scarcity of provisioned resources in a certain area) or not (e.g., limitations imposed by the provider). This can be detected by analyzing the behavior of the parameters for other clients in the same conditions.

network metrics : the values of the network metrics themselves need to be analyzed, since the active measurements are very informative but they can be influenced by the concurrent usage of other network-based applications.

connectivity type : the widespread use of wireless technology for Internet access will require to consider the parameters differently depending on the connection type, in order not to mix the data coming from wired connection with wireless ones, since the constraints and business models behind the provisioning of such connections are widely different.

Note that some types of data may not be readily available and they potentially need to be inferred by others. For instance, recognizing the connection type is not easy when only network-level metrics are available.

4.2 Challenges of the Analysis and Potential Solutions

Data analytics approaches need to be used both to recognize and cluster together similar behaviors with particular attention to the dimensions mentioned before. A possible analysis is described here to give hints about the type and complexity of the data processing needed.

An important aim of the Neubot and similar projects [15, 16] is to understand if events are local to the user or they happen as the result of more general conditions, for instance network congestion. In fact, in the first case the user would probably like to be informed to better understand if the behavior is normal and it conforms to his/her expectations, whereas in the latter it is highly probable that some activity between the user and the Internet backbone is happening (e.g., traffic shaping).

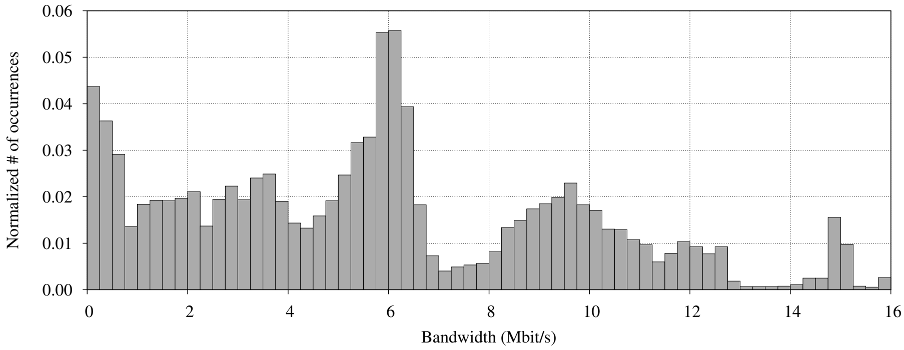

Fig. 3. Normalized number of occurrences of download speed values. The plot considers only clients connected using a given Internet operator.

To detect this condition, a possible approach is described in the following. First, some statistical performance analysis should be done to know if the experienced performance is typical or it happens only sporadically. This type of processing may be demanding in terms of computational resources, especially if the data need to be clustered. For instance, data can be separated by operator on the basis of the client IP address, and within each subset a number of clusters need to be identified, which correspond to the typical connection speeds offered to the users. An example of such a situation is shown in Fig. 3. The data refers to clients whose IP belong to a given network operator. Data suggest that there are some download speeds more common than others: in fact they correspond to the typical commercial DSL offers of that operator. However, due to the number of operators in the world, it would be infeasible to analyze the data manually to find those clusters, hence an automatic approach is needed.

Also, note that the dimensions mentioned before can influence the position of the clusters, for instance the average value of the cluster depends on many factors, such as time and location. Moreover, intercorrelation between those average values may be present and they need to be searched since they can be a very valuable indication to researchers to understand network dynamics.

4.3 Case Study

Once the previous type of analysis is available, users whose measurements significantly differ from the expected behavior can be alerted in real-time. To show the possibilities offered by such approach we present a simple analysis conducted on a limited amount of data. Fig. 4 shows the download speed of all tests run in April 2013 from clients connected to the Internet using a given operator, as grey points. The data also show the time of the day at which the test was run. The density of grey points is correlated to the probability that certain download speed values are measured in the tests. A specific user is also plotted in the graph

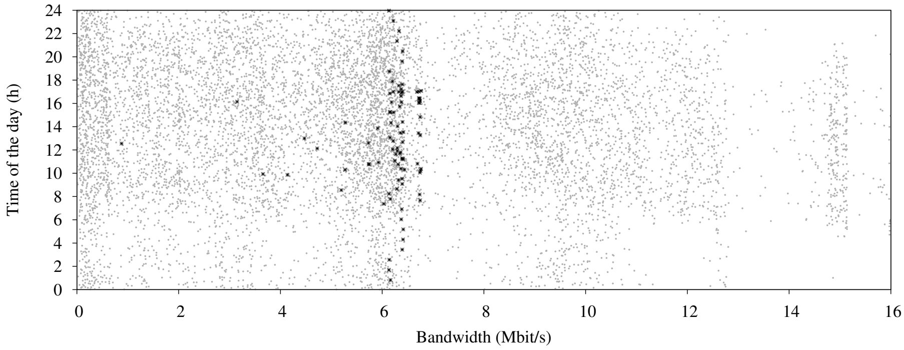

Fig. 4. Download speed of a specific user (black asterisks) over all the tests (grey dots) performed by users using the same operator for Internet connection.

using black asterisks. Most of its points are close to the center of one of the clusters, i.e., around 750 kbit/s. However, some of them are quite far. This is the type of situations which are deemed interesting, both from the research and the user's point of view. Researchers may be interested in understanding why this happens, while affected users could simply be warned about the unusual values measured by the tool, so they can decide if it is important to further investigate the situation or not. Note that this simple example consider only one user to simplify visualization, but algorithms should be able to consider more users at a time to investigate if the behavior is typical or due to some specific, isolated, reasons. Moreover, data analysis and clustering must also be run in real-time since the unusual behavior may be due only to transient reasons that however affect many users, e.g., congestion in the network. This is important since it allows to distinguish typical behaviors and patterns from transient conditions.

In any case, a scalable approach should be used so that even complex algorithms such as machine learning or data mining ones can be efficiently run on large set of data. Moreover, results should be obtained quickly enough to be useful to the users of the system, e.g., informing them about the characteristics of the detected situation. A possible approach could be to employ libraries such as the one of the Mahout project [17] that promises to provide scalable algorithms for these purposes.

5 Conclusion and Future Work

This work presented the main challenges and issues faced when collecting and analyzing a large amount of network data measurements. The data collection architecture of the Neubot project has been discussed including potential evolutions. The possibility to collect huge amount of data has been addressed from the point of view of both querying data and analyzing it with more complex algorithms, potentially in real-time. The algorithms that are expected to be used on such data have been discussed including their scalability implications and how to efficiently address them. We believe that the ability to process such huge amount of data with complex algorithms in real time can greatly contribute to gain more understanding in network dynamics by researchers interested in the area as well as by end-users interested in knowing additional information about the conditions of the network to which they are connected.

References

1. Sundaresan, S., de Donato, W., Feamster, N., Teixeira, R., Crawford, S., Pescapè, A.: Broadband internet performance: a view from the gateway. ACM SIGCOMM Computer Communication Review **41**(4) (2011) 134–145
2. Speedtest.net: The global broadband speed test. http://speedtest.net
3. Carlson, R.: Network diagnostic tool. http://e2epi.internet2.edu/ndt/
4. Neubot: The network neutrality bot. http://neubot.org/
5. Basso, S., Servetti, A., J.C. De Martin: Rationale, design, and implementation of the network neutrality bot. In: Proc. of AICA, L'Aquila, Italy (September 2010)
6. Basso, S., Servetti, A., J.C. De Martin: The network neutrality bot architecture: a preliminary approach for self-monitoring of Internet access QoS. In: Proc. of IEEE 16th International Symposium on Computers and Communications (ISCC), Corfu, Greece (July 2011)
7. Measurement Lab. http://www.measurementlab.net/
8. Dovrolis, C., Gummadi, K., Kuzmanovic, A., Meinrath, S.D., Tech, G.: Measurement lab: Overview and an invitation to the research community. ACM SIGCOMM Computer Communication Review **40**(3) (2010) 53–56
9. Meeker, M.: Internet trends. D10 Conference, Rancho Palos Verdes, CA, USA (May 2012)
10. Google: Cloud storage. https://developers.google.com/storage/
11. Google: Big query. https://developers.google.com/bigquery/
12. Measurement Lab: Host an M-Lab site! http://www.measurementlab.net/host_site
13. Web 100 project. http://www.web100.org/
14. Asymco: When will Android reach one billion users? http://www.asymco.com/2012/02/29/when-will-android-reach-one-billion-users/ (February 2012)
15. Spring, N. and Wetherall, D. and Anderson, T.: Reverse engineering the Internet. ACM SIGCOMM Computer Communication Review **34**(1) (2004) 8
16. Kanuparthy, P., Dovrolis, C.: Shaperprobe: end-to-end detection of isp traffic shaping using active methods. In: Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference, ACM (2011) 473–482
17. Apache community: Apache mahout project, http://mahout.apache.org (2013)