The NeuViz Data Visualization Tool for Visualizing Internet-Measurements Data

Giuseppe Futia[1], Enrico Zimuel[2], Simone Basso[1], Juan Carlos De Martin[1]
[1]Nexa Center for Internet & Society, Politecnico di Torino (DAUIN),
Corso Duca degli Abruzzi, 24, 10129 Torino (TO), Italy
{giuseppe.futia, simone.basso, demartin}@polito.it
[2]R&D Department, Zend Technologies Inc.,
19200 Stevens Creek Blvd, Cupertino, CA 95014, USA
enrico@zend.com

Abstract. *In this paper we present NeuViz, a data processing and visualization architecture for network measurement experiments. NeuViz has been tailored to work on the data produced by Neubot (Net Neutrality Bot), an Internet bot that performs periodic, active network performance tests. We show that NeuViz is an effective tool to navigate Neubot data to identify cases (to be investigated with more specific network tests) in which a protocol seems discriminated. Also, we suggest how the information provided by the NeuViz Web API can help to automatically detect cases in which a protocol seems discriminated, to raise warnings or trigger more specific tests.*

Keywords: data visualization, network performance, big data.

1. Introduction

The Internet is a cornerstone of our societies and has been enabling unprecedented levels of social interaction, content sharing, business creation, as well as innovation in many fields. As Frischmann argues convincingly, one of the main reasons why the Internet is so relevant for us is that the Internet is an *infrastructural resource*, i.e., a shared piece of infrastructure that is typically managed as a *commons* in a non-discriminatory way [Frischmann, 2012].

However, the Internet is not an infrastructural resource as a fact of nature, or because of an immutable, technological law; the current status of the Internet is, instead, the consequence of specific choices, both private and public, that could very well change over time. For example the policy decision of who (the State or the Internet Service Providers) should finance (and under which conditions) the so-called ‘Next Generation Networks’ (NGNs) has the potential of radically changing the landscape.

In fact, many parties (including the authors of this contribution) believe that, if States allow the Internet Service Providers (ISPs) to implement premium services to collect more money and finance NGNs, the infrastructural-resource characteristics of the Internet may become less relevant, and the Internet may lose part of its *generativity* (i.e., the property of enabling more and more people to write and distribute software and/or media content [Zittrain, 2009]).

To be fair, there is little empirical evidence supporting most policy positions on both sides of the debate. On the one hand, for instance, it is hard to prove empirically ex ante that allowing ISPs to implement premium services will reduce the generativity of the Internet. On the other hand, there is surprisingly little evidence backing the 'bandwidth hogs' argument (i.e., the argument that there is a little number of people that consume most bandwidth). The Internet policy debate, in general, is so ill informed by poor data, by missing data, and by data provided by one single stakeholder that – we agree with Palfrey and Zittrain – there is a need for more, better data to anchor the debate to solid foundations and move forward [Palfrey and Zittrain, 2011].

This is indeed starting to happen: more and more network measurement tools and visualizations, in fact, are being developed by researchers and companies worldwide. Many of such tools and visualizations are hosted by Measurement Lab [MLab], an umbrella project run by the Open Technology Institute and the PlanetLab Consortium, and supported by academic partners and companies such as Google.

In this paper, in particular, we propose NeuViz (Neubot Visualizer), an architecture that allows us to process and visualize the data collected by Neubot, the network neutrality bot [Basso et al, 2011a], one of the tools hosted by Measurement Lab. Neubot – a project of the Nexa Center for Internet & Society – is a centrally-coordinated bot that runs the in background on the user computer and periodically runs network-performance tests that currently emulate HTTP and BitTorrent, and, in future, will emulate other protocols, such as the uTorrent Transport Protocol (uTP) [Norberg, 2009].

The purpose of NeuViz is to visualize and navigate Neubot data through its Web user interface, to search for cases (to be investigated with more specific network tests) in which a protocol seems discriminated. Also, NeuViz is designed to help, in the future and with a more advanced Neubot architecture, to automatically detect cases in which a protocol seems discriminated, to raise warnings or trigger more specific tests.

Many existing visualization architectures are based on cloud services and allow one to query the data on demand using SQL-like query languages; compared to such visualization tools, NeuViz is much more optimized for the specific purpose of visualizing network measurement data. We designed, in fact, a robust, scalable backend architecture to support special-purpose, complex data analysis, in which the query (or the filtering algorithm) is executed in advance on the network-experiments dataset, and in which the result is stored in one (or more) NoSQL database(s), for fast data access.

We evaluate our work by loading into NeuViz the results of two Neubot network tests (Speedtest and BitTorrent) collected in the January 2012 - May 2013 period. We show that NeuViz helps us to effectively navigate Neubot data to identify cases in which a protocol seems discriminated. Also, we suggest that the information provided by the NeuViz Web API can help to automatically detect cases in which a protocol seems discriminated.

The rest of this paper is organized as follows. In Section 2 we describe related network measurement tools and visualizations. In Section 3 we describe Neubot and the Neubot data that we used in this paper. In Section 4 we describe the NeuViz architecture. In Section 5 we describe our implementation choices. In Section 6 we describe what we learnt from browsing Neubot data with NeuViz. In Section 7 we draw the conclusions, and we describe future developments.

2. Related Work

In this section we mention the related tools and visualizations. Some of the tools that we mention (including Neubot) are hosted by Measurement Lab (M-Lab) [Dovrolis et al, 2010], a distributed server platform that also provides advanced services (e.g., the possibility of querying the hosted-tools data using BigQuery, a RESTful service to query big datasets using an SQL-like query language [BigQuery], and the possibility of measuring TCP state variables by using the instrumented Web100 TCP/IP Linux stack [Mathis et al, 2003]).

2.1 Network-Measurement Tools

In this section we mention four tools similar to Neubot: Glasnost, the Network Diagnostic Tool, SpeedTest.net, and Grenouille.

Glasnost is a client-server browser-based Java applet developed by the Max Planck Institute for Software Systems and maintained by the Measurement Lab community. Glasnost compares a certain protocol flow (e.g., BitTorrent, Emule) with a reference flow to detect traffic shaping and its cause (e.g., the port number, the payload). Glasnost flags a network path as shaped if repeated tests show that (i) the path is non-noisy and (ii) the application-level speed of the protocol flow is 20% (or more) lower than the one of the reference flow [Dischinger et al, 2010].

The Network Diagnostic Tool (NDT) is network-measurement Java applet that measures the download and upload speed between the user computer and a Measurement Lab server [Carlson, 2003]. During the measurement, the server uses the modified Web100 Linux TCP/IP stack to expose the state variables of TCP during the transfer. In addition to the Java applet a NDT command-line application is also available.

The well-known SpeedTest.net web site [SpeedTest] provides a network-measurement, flash-based test that relies on many parallel HTTP connections to estimate the download and upload broadband speed of the user's connection, using a methodology that is documented, e.g., in “Understanding Broadband Speed Measurements” [Bauer et al, 2010].

Grenouille is a network measurement tool that measures the round trip time, the download speed, and the upload speed [Grenouille].

Differently from Glasnost, Speedtest.net, and NDT (which run on-demand tests), Neubot and Grenouille run tests in the background; however, Neubot uses diverse protocols, while Grenouille focuses on the performance only.

2.2 Network-Measurement Visualizations

In this section we mention six visualizations similar to NeuViz: the visualizations of the Syracuse University School of Information studies, the world map created by Open Knowledge Foundation, the two tools proposed by Measurement Lab, the visualization of data collected by SpeedTests.net, and the visualization of the data collected by Grenouille.

The Syracuse University School of Information Studies developed three visualizations of the data collected by Glasnost [SyracuseVis]: an interactive table that shows which ISPs seem to shape (or block) BitTorrent; a visualization that displays the “top throttlers” ISPs from 2009 to 2012; a visualization that shows alleged BitTorrent shaping (or blocking) in selected countries.

Michael Bauer, data wrangler at the Open Knowledge Foundation, created a visualization of Glasnost data as well, which shows on the world map the percentage of tests that Glasnost detected as shaped [OkfnVis]. The user can filter the dataset to show only the results that are related to a single protocol emulated by Glasnost, e.g., HTTP, BitTorrent, eMule.

The Measurement Lab team developed a visualization of NDT data that shows many indexes (e.g., the number of tests, the download and the upload speed, the round trip time) on the world map [MLabVis]. Such visualization allows one to aggregate the data by ISP and by geographical dimension (country, region/state, city), and it also allows one to compare the performance of multiple ISPs at different geographical levels.

Dominic Hamon, a software engineer at Google and Measurement Lab, developed visualizations (and a video) that show, on the world map, a point indicating the latitude and the longitude of each client that runs a test towards a Measurement Lab server, using NDT data and BigQuery [BigQueryVis].

Visualizations of the data collected by SpeedTest.net can be browsed online and downloaded from the NetIndex.com website [NetIndex].

Data collected by the Grenouille tool can be browsed online through the visualization available at the Grenouille website [Grenouille].

Similarly to the NDT visualizations NeuViz is based on the world map; however, NeuViz is optimized for complex data analysis and uses precomputed data, while the NDT visualizations are more interactive and fetch the data from BigQuery on demand. Also, the aim of NeuViz is similar to the aim of the Glasnost visualizations; both, in fact, intend to make access networks more transparent by, respectively, showing anomalies and alleged shaping.

3. Neubot and Neubot data

In this section we describe Neubot and the Neubot data that we use in this paper.

3.1 Description of Neubot

Neubot is a free-software Internet bot that performs active, lightweight network-performance tests [De Martin and Glorioso, 2008; Basso et al, 2010; Basso et al, 2011a]. Once installed on the user's computer, Neubot runs in the background and every 30 minutes performs active transmission tests with servers hosted by Measurement Lab. To coordinate the botnet composed of all the Neubot instances worldwide, there is the so-called *Master Server*, which suggests each Neubot the next test to run as well as the default test parameters. Currently, the Master Server does not optimize the suggestions returned to each Neubot; however, as we will show, the information returned by NeuViz could help the Master Server to implement more dynamic policies.

Neubot implements three network performance tests: Speedtest, BitTorrent, and RawTest. Speedtest measures the network performance using the HTTP protocol, BitTorrent measures the network performance using the BitTorrent protocol, and the RawTest test measures raw, TCP-level performance (hence the name of the test). In this paper we only describe the Speedtest and the BitTorrent tests, because we are mainly interested to use NeuViz to find cases in which a protocol seems discriminated.

3.1.1 The Speedtest Test

Speedtest is an HTTP-based test – originally inspired to the test of SpeedTest.net, hence the test name – that downloads and uploads data using a single HTTP connection [Basso et al, 2011b]. The test measures the download and the upload speed at the application level. Also, the test estimates the base Round Trip Time (RTT) using as a proxy the time that the connect system call takes to complete (later indicated as connect time). The test transfers a number of bytes that guarantees that each phase of the test (download, upload) lasts for about five seconds.

3.1.2 The BitTorrent Test

The BitTorrent test is, in principle, similar to the Speedtest test, except that it uses the BitTorrent peer-wire protocol [Cohen, 2009] instead of the HTTP protocol. As Speedtest does, the BitTorrent test transfers a number of bytes that guarantees that each phase of the test (download, upload) lasts for about five seconds.

However, while Speedtest makes a single GET request for a large-enough amount of data, BitTorrent – to better emulate the BitTorrent protocol – downloads many small chunks in a request-response fashion and, to approximate a continuous transfer, makes many back-to-back requests at the beginning of the test.

3.2 Data Preprocessing and Publishing

Measurement Lab (which hosts Neubot on its servers) periodically collects the Neubot experiments results saved on its servers and publishes such results on the Web [MLabData] under the terms and conditions of the Creative Commons Zero 1.0 Universal license [CC0]. We mirrored the data provided by Measurement Lab, and we converted such data to CSV format, generating CSV files that contain one month of data each. To prepare this paper, we imported into NeuViz the CSV files from January 2012 to May 2013 (reading 5,383,376 test, from 4,037 Neubot clients worldwide, for a total of 1.5 GB) [NeubotData].

Each CSV file contains the following fields (the type is indicated in parentheses): client address (str); connect time, in second (float); download speed, in byte/s (float); Neubot version (str); operating system platform (str); server address (str); test name (str: "speedtest" or "bittorrent"); timestamp of the test, i.e., the number of seconds since 1970-01-01 00:00 UTC (int); upload speed, in byte/s (float); unique identifier of the Neubot instance (str).

4. Description of the NeuViz Architecture

Fig. 1 shows the NeuViz architecture, which is a pipeline that processes data provided by *Producers*, and which organizes the data such that *Consumers* can visualize (or further process) such data. The pipeline is composed of a *Backend* and a *Frontend*: the Backend receives data from many Producers and processes such data to allow for efficient visualization; the Frontend is a Web interface that visualizes the data. In the middle there is a *Web API*.

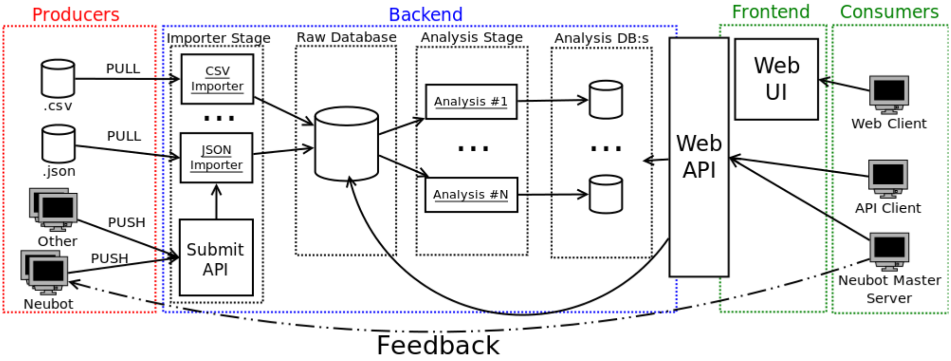

Fig. 1: NeuViz Architecture

4.1 The Producers

As a first approximation a Producer is a static dataset. For example, in this paper we used Neubot data expressed in CSV format and in the future we may want to import datasets from other projects (e.g., SpeedTest.net) and encoded in other formats (e.g., JSON).

NeuViz also includes a Submit API, which allows network-experiment tools (e.g., Neubot and possibly other network-measurement tools) to push the result of their experiments just after the experiments are run. We added the Submit API because we want to create a feedback loop in which data posted by Neubot is processed by NeuViz and consumed by the Master Server to provide better suggestions to Neubot instances.

4.2 The Backend

The Backend receives network-experiments data from many sources and organizes such data for an efficient visualization. As Fig. 1 shows, the Backend is composed of two processing stages, each followed by a database stage. The first processing stage is the *Importer Stage*, which receives data from many sources, normalizes the data, and writes the data into the *Raw Database*. The second processing stage is the *Analysis Stage*, which reads data from the Raw Database, analyzes the data to compute aggregate metrics, and saves the aggregate metrics into one or more *Analysis Databases*.

In the following sections we discuss the stages of the Backend, starting from the Importer Stage.

4.2.1 The Importer Stage

The Importer Stage organizes data coming from many sources (and possibly represented using different formats) into a single database. There is one Importer Module for each network measurement tool and data format. To make an example, if we want to use NeuViz to visualize SpeedTest.net data (expressed in CSV format) and Neubot data (expressed in CSV and JSON format), we need to write three Importer Modules: one for the SpeedTest.net data and two for the Neubot data (the former for the CSV and the latter for the JSON format).

The Submit API design reflects the fact that there is an Importer Module for each network measurement tool and data format. The basic API request to store the result of a new experiment, in fact, is a POST request to this URI: “/neuviz/1.0./import/<tool>/<params>”, where <tool> is the name of the tool that produced the piece of data (e.g., “neubot”), and where the Content-Type HTTP header must reflect the data type (e.g., “application/json”). The problem of whether (and how) to authenticate the measurement tool submitting the data is not discussed in this paper.

The Importer Stage does not reduce all the input data to the same schema (be it a real SQL schema or not), because such transformation is not practical. The input data schema, in fact, depends on which metrics the specific network experiment measures; therefore, this stage just enriches the data with geographical information (if needed), converts the data into a common, database-dependent format (e.g., JSON), and writes the data into the Raw Database.

4.2.2 The Raw Database

The Raw Database receives heterogeneous data organized in a uniform format (e.g., JSON) by the Importer Stage. As said before, it is not practical to reduce all the input data to the same schema, suggesting that the Raw Database could be easily implemented using NoSQL (e.g., MongoDB [Mongo]).

A possibly-conflicting requirement for the Raw Database is that, in addition to being able to store heterogeneous data, the Raw Database shall also be scalable-enough to handle continuous streams of data posted on the Submit API by, at least, Neubot and possibly by other network measurement tools.

4.2.3 The Analysis Stage

The Analysis Stage is a collection of Analysis Modules that periodically fetch data from the Raw Database and process it to produce the aggregate data needed for the visualizations. To start off we plan to implement two different visualizations: one that shows a given performance metric (e.g., the median download speed) on the world map and that allows the user to zoom and see the same performance metric on a smaller geographic scale (i.e., country, province, city); the other that shows a given performance metric in function of the time.

As far as functional requirements are concerned, the Analysis Stage needs to process data in a scalable way, because we need to process multiple times the raw data stored in the Raw Database. Also, the Analysis Stage should minimize the computational cost of adding the results of new experiments to NeuViz.

4.2.4 The Analysis Databases

The Analysis Databases are a number of (conceptually-separated) databases that store data which is ready to be visualized on the NeuViz Frontend with minimal computational cost. We want, in fact, to allow the user to visualize and browse the data as seamlessly as possible.

4.3 The Web API

The Web API connects the Backend and the Frontend. The Frontend, in fact, uses the Web API to retrieve the data that should be visualized by a Web client through the Neuviz Web interface. However also other clients can access the Web API to extract information from the collected data.

The Web API typically returns the Analysis Database data, because NeuViz is optimized to store and quickly return the results of the data analyses. However, in cases in which the cost of processing the Raw Database data on the fly is negligible, the Web API will access directly the Raw Database data and will compute the result on the fly. This is represented in Fig. 1 by an arrow that goes from the Web API to the Raw Database.

In this paper we do not discuss whether and how the access to the API should be restricted. This will possibly be the subject of a future work.

4.4 The Frontend and the Consumers

The Frontend is a Web interface that visualizes the data stored in the Backend.

The typical (and default) Consumer is of course a Web client that uses the NeuViz Web interface, but also other clients can consume the available data. In particular an interesting, already-planned reuse of the Web API is the following: we plan to modify the Master Server to retrieve data from the Web API, process the data, and adapt accordingly the suggestions the Master Server provides to Neubot instances (e.g., if there are few Neubot instances in a specific geographical area, the Master Server suggests to perform tests more frequently in that area).

5. Implementation Choices

In this section we describe the implementation of the first NeuViz prototype [NeuVizGit], and we explain our implementation choices.

5.1 The Importer Stage

We implemented the Importer Stage step using a Python command-line script that accepts in input a CSV file. In our tests we imported and normalized 1.5 GB of Neubot data (using CSV files), from January 2012 to May 2013, and we stored the data into a MongoDB database. We run the code on a laptop with an Intel Core i7 CPU at 2.0 Ghz, with 8 GB of RAM, and a 256-GB SSD, running GNU/Linux 3.5.0. The Python code is designed to execute both on a common computer and in a cloud environment, if needed: to this end we divided the Importer and the Analysis code into a *map* step and a *reduce* step.

We also used the GeoLite Free Database to retrieve geo-information from the client IP address, using MongoDB to store the geographic information [GeoLite]. As explained in the GeoLite website, when the database is not up-to-date, the geolocation loses 1.5% of accuracy each month because IP addresses are re-assigned. To minimize the damages caused by out-of-date GeoLite databases, we never used databases older than two months.

5.2 The Raw Database

We implemented the The Raw Database using MongoDB, a NoSQL database very often deployed in big data scenarios [Moniruzzaman and Akhter, 2013]. We exploited the indexes feature of MongoDB to speed up the query execution, processing about 5.3 million of samples in less than 60 minutes.

As noted above, the code is written in a way that potentially allows us to use MapReduce techniques on cloud services [MapReduce], should we need to do that. However, especially during the development of the initial prototype, we didn’t used MapReduce, because a single NoSQL database allowed us to perform queries on demand and retrieve data immediately (which is not, of course, possible in a cloud-based MapReduce scenario).

5.3 The Analysis Stage

We implemented a prototypal Analysis Module, written in Python, to retrieve and process data from the MongoDB database and create our world map visualization, and we are also working on another Analysis Module that will generate data for the visualization that shows a given performance metric in function of the time.

The Analysis Stage that we implemented outputs a JSON file in which the information is aggregated at the geographical level (countries, and cities), at the temporal level (hour of the day), and at the business level (ISP). Therefore, the Web interface receives in input, for BitTorrent and Speedtest, the median value of the upload speed, of the download speed, and of the connection time of a specific country or city, and their ISPs, in a precise hour of the day. We decided to use the median, which is a common index used to analyze network traffic, to avoid the risk that few outliers could dominate our index.

We also computed the number of Neubot instances (per country, city, ISP) as well as the number of Neubot tests (per country, city, ISP). Since the IP address can vary over time, we identified a Neubot instance by using the (Neubot ID, IP address) tuple. The number of Neubot instances and the number of tests can be used to understand the geographical distribution of Neubot clients and the network traffic produced by each Neubot.

5.4 The Analysis Databases

We generated a JSON file for each month of the Analysis Stage. The collection of these files can be considered to be the Analysis Databases. However, these JSON files can also be stored in a MongoDB to retrieve the data according with different parameters or different search query. Data could also be stored in the cloud when scalability needs occur.

5.5 The Web API

To access the NeuViz API, the user sends the following HTTP/1.1 request: GET /neuviz/1.0/<viz>/<params>, where <viz> is the name of the visualization, and <params> is a placeholder for (possibly-empty) parameters. The returned JSON contains a recursive set of dictionaries that represent the geographical dimension (country, city), the time dimension (hour of the day) and the business dimension (ISP). The leaves are dictionaries that contain the following hour-wide median statistics for the Speedtest and the BitTorrent tests: download speed, upload speed, connection time, number of Neubot instances, number of tests. The geographical (country, city), the time (hour of the day), and the business (ISP) dimensions of data is shown in Fig. 2.

5.6 The Frontend

The Web interface, written using D3.js [D3], allows the user to explore different network measurement performances at different geographic dimensions (country, cities, and ISPs). For simplicity, and since it does not seem to cause any performance issue, we currently use the Web interface to compute some statistics, e.g., the difference between the median Speedtest download speed and the median BitTorrent download speed that we use in Section 6.2 to compare the performance of BitTorrent and Speedtest.

6. Results

In this section we report what we learnt from using NeuViz to browse Neubot data, both in terms of number of tests and in terms of performance.

6.1 Number of Neubot Tests

Fig. 3 shows the visualization of the number of tests per country and per hour. The alpha channel of the country color indicates the median number of tests per country. The visualization, in particular, shows the median number of tests performed between 9:00 PM and 10:00 PM (local time) in April 2013. The selected country is Canada, in which the median number of tests performed is indicated by the number in the bottom right corner (1084).

By selecting other countries in the visualization, we have seen that the countries with more median tests per hour between 9:00 PM and 10:00 PM in April 2013 are: the US (4223); Italy (2866); Germany (2285); and Canada (1084). Other countries have less tests per hour.

The availability of the number of tests per country is interesting because, by knowing the number of tests per country, the Master Server could maximize the test coverage; e.g., it can increment the frequency of testing on countries where there are few Neubot users.

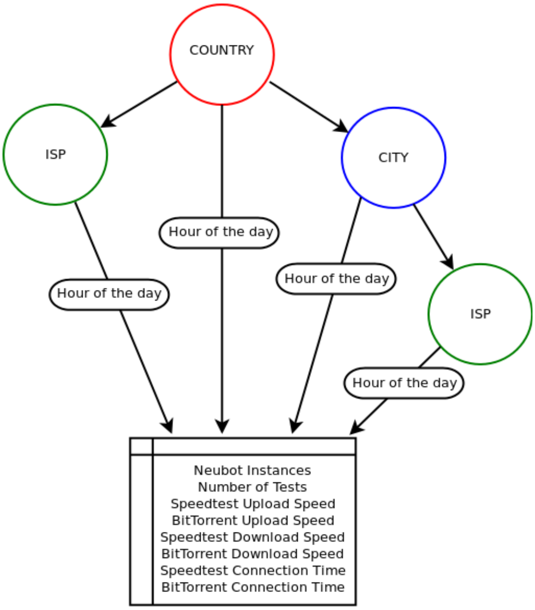

Fig 2: Representation of geographical (country, city), time (hour of the day), and business dimensions (ISP) of JSON file.

6.2 Comparison of Speedtest and BitTorrent performance

Before studying the visualization that shows the difference between the Speedtest and the BitTorrent test download and upload speeds, we checked whether the Speedtest and the BitTorrent connect times were 'comparable'. To this end we arbitrarily define ‘comparable’ two median connect times whose difference is smaller than five millisecond, in our experience a reasonable threshold for this kind of analyses.

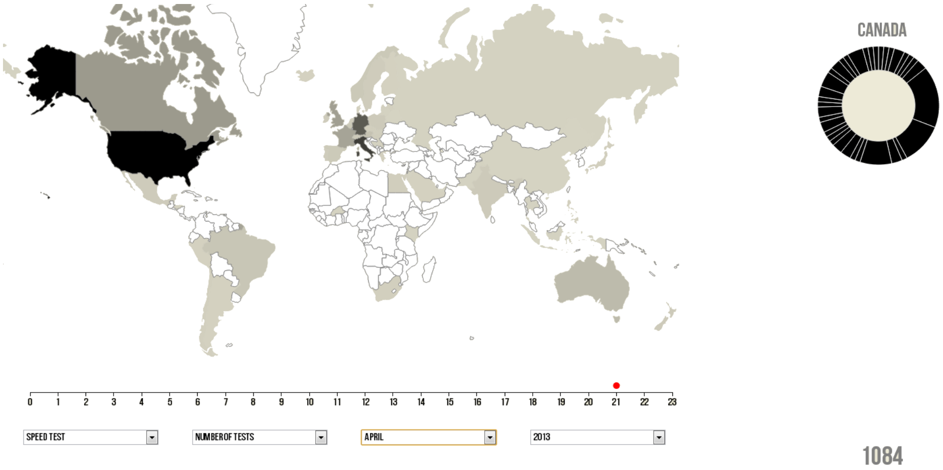

Fig 3: NeuViz interface of the worldwide map for Neubot data of April 2013

The visualization of the difference between the median BitTorrent connect time and the median Speedtest connect time shows, surprisingly, that in Italy such difference is always positive and often greater than five millisecond (i.e., the Speedtest connect time is typically lower). Italy is the only country in which, for 2013 data, we noticed this behavior.

Also we noticed interesting things from the comparison of the median upload speed in countries in which the median connect times are comparable. We noticed, in fact, that in 2013 the median upload difference between Speedtest and BitTorrent in Canada was very often positive, while the same difference was very often negative in the US (see Fig. 4).

Moreover, when comparing the download speeds in countries in which the connect times are comparable, we also noticed that the US Speedtest download speed is always lower (in median) than the BitTorrent one for every hour of the day and for every month of 2013. Interestingly, instead, the download speeds are comparable in Italy, in which – as we have seen – there is a connect time bias in favor of Speedtest.

The above observations lead us to speculate that: (a) BitTorrent is slightly faster than Speedtest; (b) in Italy the two tests are comparable because of the connect-time bias that we observed; (c) the BitTorrent upload speed seems to be discriminated in Canada. Of course, these are only hypotheses that need to be verified (or contradicted) by more detailed experiments.

6.3 Concluding Remarks

Despite being still in beta stage, NeuViz allowed us to discover the three diverse network anomalies we described in Sect. 6.2. In the future, a more advanced Master Server could learn, from the NeuViz API, about similar anomalies and ask Neubot instances that are near the anomalies to gather more information needed to investigate the anomalies (e.g., one could capture packets to gather RTT samples useful to understand whether there is a connect-time bias).

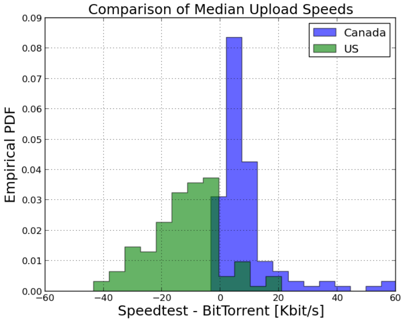

Fig 4: the Empirical Probability Density Function (PDF) of the difference of the median upload speed of US and Canada

7. Conclusion and Future Work

In this paper we described NeuViz, an architecture that allows us to process and visualize the data collected by Neubot, the active, network-measurement tool developed by the Nexa Center for Internet & Society. The purpose of NeuViz is to visualize and navigate Neubot data through its Web user interface, to search for cases (to be investigated with more specific network tests) in which a protocol seems discriminated.

Differently from other visualization architectures NeuViz is much less flexible and much more optimized, on purpose. NeuViz, in fact, executes the queries in advance and the result is stored into one or more NoSQL databases (using MongoDB), for fast data access. The Backend of NeuViz, written in Python, is structured to ease the task of porting it to a cloud-based MapReduce solution, for future scalability. The Web interface Frontend of NeuViz shows a world-map-based visualization of Neubot results implemented using the D3.js library.

To evaluate NeuViz we loaded one-year-and-a-half records collected by two network tests periodically run by Neubot, called Speedtest (based on HTTP) and BitTorrent. We showed that NeuViz effectively helped us to identify cases (to be investigated with more specific network tests) in which a protocol seems discriminated. In our discussion we also suggested how the Web API of NeuViz can help to automatically detect cases in which a protocol seems discriminated, to raise warnings or trigger more specific tests (by cooperating with the Master Server of Neubot). As part of our future work we plan to extend NeuViz to automatically raise warnings and to cooperate with the Master Server of Neubot to trigger more-specific network experiments.

Acknowledgments

The first prototype of the NeuViz project has been developed as final project of the BigDive course 2013 [BigDive]. We would like to thank Christian Racca of the TOP-IX Consortium and all the staff and teachers of the BigDive course for their support during the development of this project.

References

[Basso et al, 2010] Basso S., Servetti A., De Martin J. C., Rationale, Design, and Implementation of the Network Neutrality Bot, in Proc. of Congresso Nazionale AICA 2010, L'Aquila, 2010.

[Basso et al, 2011a] Basso S., Servetti A., De Martin J. C., The network neutrality bot architecture: A preliminary approach for self-monitoring of Internet access QoS, in Proc. of the Sixteenth IEEE Symposium on Computers and Communications, Corfu, Greece, 2011.

[Basso et al, 2011b] Basso S., Servetti A., De Martin J. C., The hitchhiker's guide to the Network Neutrality Bot test methodology, in Proc. of Congresso Nazionale AICA 2011, Torino, 2011.

[Bauer et al, 2010] Bauer S., Clark D., Lehr W., Understanding broadband speed measurements, in Proc. of Telecommunications Policy Research Conference, 2010.

[BigDive] Big Dive course website, from http://www.bigdive.eu.

[BigQuery] Google BigQuery, from http://developers.google.com/bigquery/.

[BigQueryVis] Hamon D., Visualizing M-Lab data with BigQuery, from http://dmadev.com/2012/11/19/.

[Carlson, 2003] Carlson R., Developing the Web100 Based Network Diagnostic Tool (NDT), In Proc of the Passive and Active Measurement Conference, 2003.

[CC0] Creative Commons Zero 1.0 Universal License, from http://creativecommons.org/publicdomain/zero/1.0/.

[Cohen, 2009] Cohen B., The BitTorrent Protocol Specification, from http://www.bittorrent.org/beps/bep_0003.html.

[D3] D3.js – Data Driven Documents, from http://d3js.org/.

[De Martin and Glorioso, 2008] De Martin J.C., Glorioso A., The Neubot project: A collaborative approach to measuring internet neutrality, in Proc. of the IEEE International Symposium on Technology and Society, Fredericton, Canada, 2008.

[Dischinger et al, 2010] Dischinger M., Marcon M., Guha S., Gummadi K. P., Mahajan R., Saroiu S., Glasnost: Enabling End Users to Detect Traffic Differentiation, in Proc. of USENIX Symposium on Networked Systems Design and Implementation, 2010.

[Dovrolis et al, 2010] Dovrolis C., Gummadi K. P., Kuzmanovic A., Meinrath S., Measurement Lab: Overview and an Invitation to the Research Community, ACM SIGCOMM Computer Communication Review, 40, 3, 2010, 53–56.

[Frischmann, 2012] Frischmann B. M., Infrastructure: The Social Value of Shared Resources, Oxford University Press, 2012.

[GeoLite] GeoLite Free Database, from http://dev.maxmind.com/geoip/legacy/geolite/.

[Grenouille] Grenoulile.com website, from http://grenouille.com/.

[MapReduce] Amazon Elastic MapReduce service, from http://aws.amazon.com/elasticmapreduce/.

[Mathis et al, 2003] Mathis M., Heffner J., Reddy R., Web100: Extended TCP Instrumentation for Research, Education and Diagnosis, ACM SIGCOMM Computer Communication Review, 33, 3, 2003, 69–79.

[MLab] Measurement Lab website, from http://www.measurementlab.net/.

[MLabData] Measurement Lab data, from http://measurementlab.net/data.

[MLabVis] Broadband performance using NDT data, from http://goo.gl/m9WbS (google.com/publicdata/explore/...).

[Moniruzzaman and Akhter, 2013] Moniruzzaman A. B. M., Akhter H. S., NoSQL Database: New Era of Databases for Big data Analytics - Classification, Characteristics and Comparison, International Journal of Database Theory and Application, Vol. 6, No.4, 2013.

[Mongo] MongoDB, from http://www.mongodb.org/.

[NetIndex] Net Index by Ookla company, from http://www.netindex.com/.

[NeubotData] Neubot Measurement Lab data mirror, from http://data.neubot.org/mlab_mirror/.

[NeuVizGit] NeuViz GitHub repository, from https://github.com/neubot/neuviz.

[Norberg, 2009] Norberg A., uTorrent transport protocol, from http://www.bittorrent.org/beps/bep_0029.html.

[OkfnVis] Network neutrality map using Glasnost data, from http://netneutralitymap.org/.

[Palfrey and Zittrain, 2011] Palfrey J., Zittrain J., Better Data for a Better Internet, Science, 334, 6060, 2011, 1210-1211.

[SpeedTest] SpeedTest.net website, from http://www.speedtest.net/

[SyracuseVis] Deep packet inspection stats using Glasnost data, from http://dpi.ischool.syr.edu/MLab-Data.html.

[Zittrain, 2009] Zittrain J., The future of the Internet--and how to stop it., Yale University Press, 2009.