The hitchhiker's guide to the Network Neutrality Bot test methodology

Simone Basso*+ Antonio Servetti+ Juan Carlos De Martin*+

simone.basso@polito.it, antonio.servetti@polito.it, demartin@polito.it

* NEXA Center for Internet & Society – DAUIN – Politecnico di Torino, Italy

+ Internet Media Group – DAUIN – Politecnico di Torino, Italy

Abstract

The Neubot project is based on an open-source computer program, the Neubot, that, downloaded and installed by Internet users, performs quality of service measurements and collects data at a central server. The raw results are published on the web under the terms and conditions of the Creative Commons Zero license. This paper is the guide for researchers and individuals that aims to study, build on and analyze Neubot methodology and results. We provide an exhaustive documentation of Neubot's HTTP test behavior, along with a discussion of the methodology. Besides that, the article shows an analysis of the Turin-area results (in the May-September time interval) and explains the rationale behind the privacy policy, which allows us to publish results as raw data.

1 Introduction

The debate on "network neutrality" is becoming a more and more relevant topic in economic, technical and political environments [1] [2] [3]. The basic question is whether network operators should be allowed to differentiate the Internet traffic that goes through their infrastructure or whether network neutrality should be explicitly safeguarded by the law, thereby enshrining what has been a characteristic of the Internet since its birth [4].

The ability to block or slow down the traffic, which is used to prevent the spreading of spam, viruses, botnets and other malwares, may also be employed by Internet Service Providers to implement other traffic policies that are very questionable [5]. Apart from the so-called "Great Firewall of China" [6] [7] and other striking blocking efforts [8] (which are beyond the scope of this paper), differentiating technologies are also employed to throttle the traffic of peer-to-peer applications and Content Providers.

The traffic of peer-to-peer applications (e.g. BitTorrent and Emule) may be slowed down to protect underprovisioned shared access networks from congestion, during peak hours or whenever the network load rises above certain threshold. More selectively, differentiation may be employed to throttle "seeding-only" peer-to-peer traffic, especially when the ISP has a transit agreement with the upstream provider and it is not willing to pay money for traffic that does not directly benefit the user. Finally, the differentiation of peer-to-peer applications (e.g. Skype) and Content Providers' traffic (e.g. YouTube and Netflix) allows to allocate and guarantee bandwidth to the so-called "Managed or Specialized Services", i.e. services like voice over IP and video on demand, bundled with the Internet-access offer and available to the ISP subscribers only [9] [10].

Technically, network differentiation is a two-step process. At the edge of the network, there is the classification function, which assigns each packet to a traffic class, using different techniques [11] and policies, possibly also accounting for the user's past behavior. Then, inside the network, packets receive the service level associated with their class. In particular, packets that belong to low priority classes may be diverted on slower and/or more congested links, resulting on a worst

average quality of service. Alternatively, they may be scheduled for forwarding after higher priority packets [12] [13] and dropped with higher probability from router queues in case of congestion [14].

As a consequence, the topic of network neutrality is tightly related to the one of network quality measuremement. A nonneutral network path is one where certain traffic classes experience, on the average, more packet losses and/or delays than others. For this reason most of the network neutrality tools available in literature perform network quality measurement, via active probing and/or capturing and analyzing the user's traffic. Then, they compare the measurement results for different protocols and/or services, seeking for significant quality differences [15] [16] [17] [18].

Despite the potential impact of traffic differentiation and the fact that there are some available tools for network quality measurement, most users, developers and network administators are still not provided with enough information. At the same time, ISPs and Content Providers have access to a wealth of data, respectively because they convey users' traffic and they serve a large number of eyeballs [19]. To start closing this information gap and empower the users, our proposal is the "Network Neutrality Bot" (Neubot) project.

The Neubot project is based on an open-source computer program, the Neubot, that, downloaded and installed by Internet users, performs quality of service measurements and collects data at a *central server* [20]. Neubot runs in the background on the user computer and periodically contacts the *central server*, which redirects it to the closest *test server*. In turn, Neubot connects to the *test server*, performs a *transmission test*, saves the results locally and uploads them to the test server as well [21]. Neubot is designed as a software architecture for network measurements and currently hosts two *transmission tests*. The *speedtest* test uses the HTTP protocol and measures round-trip time, download and upload goodput [22]. It is inspired to Speedtest.net [23] test, hence the name. The *bittorrent* test emulates the BitTorrent peer-wire protocol and measures round-trip time, download and upload goodput. The raw results are published at `http://www.neubot.org/data` under the terms and conditions of the Creative Commons Zero license.

This paper is the guide for researchers and individuals that aims to study, build on and analyze Neubot methodology and results. The major contribution of this work is the exhaustive documentation of the *speedtest* test behavior, along with a discussion of the methodology. Besides that, the article shows a qualitative analysis of the Turin-area results, in the May-September time interval. And explains the rationale behind the privacy policy, which allows to publish results as raw data.

The rest of this paper is organized as follows. In section 2 we describe related research efforts. Section 3 documents and discusses in detail the *speedtest* transmission test. Section 4 is dedicated to the presentation of the measurements in the Turin area. In section 5 we explain the rationale behind the privacy policy. Finally, in section 6 we draw the conclusions.

2 Related work

Early network-neutrality literature and tools focus on detecting port blocking and RST injection. The seminal work of Beverly, Bauer and Berger aims to build a map of port blocking policies, with emphasis on VPN, email and file sharing ports [24]. Electronic Frontier Foundation's Switzerland is a semi peer-to-peer prototypal application designed to detect the modification or injection of packets with the help of a central server [25]. The article of Dischinger et al. describes BTTest, an easy-to-use tool to detect BitTorrent RST injection practices, which has been used to characterize the RST policy of many providers [26]. NNSquad Network Measurement Agent (NNMA) is a prototypal piece of software that detects certain spoofed RST segments [27]. Weaver, Sommer and Paxson's paper provides a rich set of heuristics to detect RST injections and identify the generating device with a certain degree of confidence [28].

More recent works measure various facets of quality and compare those measurements to assess the degree of neutrality. NANO (Network Access Neutrality Observatory) is an agent that passively collects performance data from users' computers and uploads them to a central server. In turn, the server relies on stratification to group clients in clusters where the performance difference

depends on the ISP policy and not on confounding factors [29] [18]. NetPolice (formerly NVLens) is a protocol-aware traceroute-like tool to (i) probe backbone ISP paths with traffic that emulates different applications protocols and (ii) compare the loss rate [30] [17]. DiffProbe is an active probing method to detect whether an ISP is performing delay or loss discrimination comparing the delays and losses experienced by two flows [16]. Glasnost – the evolution of BTTest – compares the throughput of two traffic flows, identical in all aspects other than their packet payloads. It runs the two flows multiple times back-to-back, filters the results to eliminate noise and reports (non)discrimination [15]. WindRider is a mobile application that performs active goodput tests using different ports. It also performs passive tests such as measuring the delay to load web pages. In both cases the results are uploaded to central servers [31].

Other research and nonresearch efforts just focus on quality. BISMark is a OpenWRT-based home router that performs periodic active measurement of the access link performances [32]. Grenouille periodically performs ping, download and upload tests with nearby servers and collects results at a central server [33]. Pathload2 is a tool that measures the available bandwidth of the user's broadband connection [34].

Completely different from the above projects and tools is Measurement-Lab (M-Lab). Which is a distributed server platform – founded by the New America Foundation's Open Technology Institute, the PlanetLab Consortium, Google Inc. and academic researchers – where researchers can deploy the server of their active transmission test tools [35].

3 The *speedtest* transmission test

The *speedtest* test takes place between the Neubot and the test server indicated by the master server. Before the actual test, there is the negotiation of the test parameters and the test itself may be delayed to control the load. Then there is the actual test, discussed below. After that, the Neubot sends the test server its results and vice versa, so there is complete information. For a more detailed explanation of the test phases we refer the reader to our previous work [21].

The actual *speedtest* test uses the HTTP protocol and measures round-trip time, download and upload goodput. The round-trip time is estimated using two distinct techniques: the measurement of the time required to connect and the measurement of the request-response latency. The goodput is the application-level throughput [36], calculated dividing the amount of sent (or received) bytes over the elapsed time.

In the following subsections we provide more details regarding the measurements. Then we discuss the limits of our methodology.

3.1 Time required to connect

The time required to connect is the time it takes for the `connect(2)` system call to complete. This approxymates the round-trip time because this system call immediately sends a `SYN` segment and returns when it receives the `SYN|ACK` segment.

Neubot uses nonblocking I/O and `select(2)` to dispatch I/O readiness events. Therefore the time required to connect is actually computed performing the difference between the time when the socket becomes writable and the time when `connect(2)` reports that the connection attempt is either complete or `EINPROGRESS`.

Since the test uses two (or four) connections, the result is the average. The current implementation (0.4.1) serializes multiple concurrent connection attempts, so each `connect()` is measured independently of the others. This was not the case before Neubot 0.3.7: in old versions multiple attempts were overlapped. So the average is more noisy in old Neubot versions.

3.2 Request-response latency

The request-response latency is computed performing the difference between the time when the response is received and the time when the request is sent. This is a reasonable overestimation of

the round-trip time because Neubot sends small `HEAD` requests and the server responds with very small bodyless responses. Indeed, per `RFC 2616` the response to an `HEAD` request must not contain the body and must consist of the headers only [37].

While the test uses two (or four) connections, only one connection is employed to perform this measurement. The request-response latency is measured back-to-back for ten times. And the result is the average.

3.3 Goodput

The goodput is calculated dividing the amount of bytes sent (or received) over the elapsed time. The download estimation is performed at the receiver, while the upload estimation is performed at the sender. The amount of bytes sent (or received) is initially small and may be adapted after the test, if needed. The elapsed time is the difference between the time when the last byte of the response is received and the time when the first byte of the request is sent. The test is repeated if the elapsed time is less than the target test duration. Before repeating the test, the amount of bytes to send (or receive) is adapted, so that subsequent tests would run for more than the target test duration (under current conditions).

Since the first public release (Neubot 0.3.0) the test has employed two concurrent connections. This was motivated by the fact that we started with just one server, located in Turin, and we wanted to double the range of bandwidth-delay products we could serve. There was also a period (Neubot 0.3.5 – 0.3.6) when Neubot employed four connections, but we decided to step back because the goal was to measure the quality not to maximize the goodput (more on this later).

The target test duration changed significantly since the first public release. One key point is to perform *quick* measurements, because Neubot is a background tool and should not consume too much network resources. For this reason, version 0.3.0 estimated the goodput out of one-second transfers. Unfortunately that was not enough and caused too much variation, even if the mean was consistent with other independent measurements [22]. To counter this problem, Neubot 0.3.5 and Neubot 0.3.6 raised the number of concurrent connections, from two to four. But the variation was still high, so Neubot 0.3.7 returned to the two-connections model but increased the target to five seconds.

The algorithm to adapt the number of bytes to send (or receive) after a test changed since the first public release too. At the beginning the strategy was to double the amount of bytes when the duration was less than the target. That was acceptable for short transfers but is too aggressive for five-seconds transfers. So, starting from Neubot 0.3.7 the amount of bytes is still doubled, but when the duration is less than one second only. Otherwise, it is scaled so that the next test would run for at least five seconds (under current conditions).

Starting from Neubot 0.3.7, the receive buffer is set to the fixed value of 256 KiB, because we have observed that different operating systems scale differently the receive buffer during slow start and we wanted to eliminate a possibile confounding factor. We have chosen 256 KiB because it is the maximum-possible socket buffer value on BSD (to increase it more one needs root privileges to raise the limit via `sysctl`).

3.4 Discussion

The "time required to connect" approximation of the RTT is acceptable if neither the `SYN` nor the `SYN|ACK` segments are lost. And if the two operating systems are not heavily loaded, so that the network delay is at least one order of magnitude more than the segment processing delay. The "request-response latency" is less precise because there is always some application delay involved and because there is application overhead caused by HTTP headers. But it allows to collect more than one sample per connection. Of course, both estimates are performed at the beginning of the connection so they may not reflect the actual RTT during the data transfer.

The upload goodput estimate is measured at the sender. This is done for implementation simplicity. But it would be more robust to measure it at the receiver. Because measuring at the

receiver smooths possible peaks, such as the ones caused by drop tail in the home router. And because, more generally, the measurement at the receiver conveys more information.

The methodology always yields an underestimate of the actual goodput. Because the measurement starts when the requester sends the request. Therefore the elapsed time includes at least one idle round-trip time: the one when the connection is waiting for the response to arrive. The impact of that was higher with one-second measurements than it is with five second. Assuming a round-trip of 100 ms the idle time was 10% (or more), while now it is 2% (or more).

The methodology allows to compare connections with similar round-trip time ranges. But the comparison is more problematic when the round-trip times significantly differ. The problem is that the target duration is expressed in seconds, while TCP behavior is typically modeled in terms of "rounds", where the round duration is one round-trip time [38]. So, connections with very different round-trips run the test for the same number of seconds but certainly not for the same number of rounds.

One-second tests experienced more variation because the percentage of rounds spent in the slow start was higher. In this state, fast recovery is fragile and TCP performances are more loss-sensitive. So, while the goodput measured by a short test may not be a good throughput approximation, the variation is still interesting, because it conveys information on the average loss ratio. In turn, this information can be used to compare the quality of different protocols or different access networks.

The concurrent-connections model has the advantage that the measured goodput is closer to the throughput. The available bandwidth is used more efficiently, because each connection has a smaller window, so it takes less round-trip times to recover after a loss. And because a connection may grow and take more bandwidth when another experiences a loss. As a consequence, measuring a congested network with more concurrent connections yields a better goodput. Which is not desirable, because worst access links should be penalized more, not less.

The advantage of fixed-size receive buffers is that all operating systems should have the same behavior. Moreover, knowing in advance the buffer size simplifies results analysis, because it is easier to classify buffer-limited TCP connections. On the other hand, the problem with this strategy is that most high-speed high-delay connections are likely buffer limited. Especially in countries like Korea and Japan where it is common to have 100 Mbit/s access links. This is not desirable because better access links should be premiated, not penalized.

Finally, the methodology does not account for many confounding factors that may reduce the goodput. It assumes that the user is not heavily using her computer and/or her access link. That no other users are heavily using/sharing the same access link. That the local wireless connection (if any) does not experience many losses. That there is no congestion in the backbone. That the test server is not overloaded. And possibly there are other hidden assumptions and unaccounted confounding factors.

4 Measurements in the Turin area

This section presents an analysis of a subset of the data collected in the period from 30rd May to 13 September 2011.

Table 1 shows the number of tests, the number of Neubots and the date of the first and last test for the whole dataset and for the subset of Neubots geolocated in the Turin area. We use MaxMind `<http://www.maxmind.com/>` free tools and databases for geolocation and for the identification of the Provider. We focus on the Turin area because our test server is located in the Torino-Piemonte Internet eXchange (TopIX), so we show measurements performed by "nearby" (in terms of RTT) Neubots. Of course, the analysis performed on the Turin area can be repeated for other regions and countries to build a per-geographic-location per-provider results map.

Table 2 shows the number of measurements and the number of Neubots geolocated in the Turin area for each Autonomous System (which is a Provider's network). In this paper we show the results of the four commercial providers with more tests and more users in the Turin area: Vodafone, Infostrada, Fastweb and Telecom Italia. GARR, the Italian universities network, is

	Whole dataset	Turin area
Number of tests	947729	80196
Number of Neubots	1054	166
First test	30-05-2011	30-05-2011
Last test	13-09-2011	13-09-2011

Table 1: This table shows the number of tests, the number of Neubots and the date of the first and last test for the whole dataset and for the subset of Neubots geolocated in the Turin area.

Autonomous System	Measurements	Neubots
AS2594 CSI Piemonte	77	1
AS44957 OPITEL AS number	45	2
AS8968 BT Italia S.p.A.	1234	2
AS35612 NGI Spa	11	3
AS35719 TEX97 S.p.a	573	3
AS24608 H3G S.p.A.	18	4
AS16232 TIM (Telecom Italia Mobile) Autonomous System	90	4
AS8612 Tiscali Italia S.P.A.	555	7
AS137 GARR Italian academic and research network	26200	25
AS30722 Vodafone N.V.	1162	8
AS1267 Infostrada S.p.A.	794	24
AS12874 Fastweb SpA	24735	44
AS3269 Telecom Italia S.p.a.	24702	65

Table 2: This table shows the number of measurements and the number of Neubots geolocated in the Turin area for each Autonomous System (which is a Provider's network). The four bottom Autonomous Systems are the ones plotted in Figure 1.

not included, even if it has many tests and users, because its performances are not comparable with the ones of commercial providers, having much more access link capacity. Of course, with the public availability of the results, one can build its own qualitative analysis with more, less or different providers.

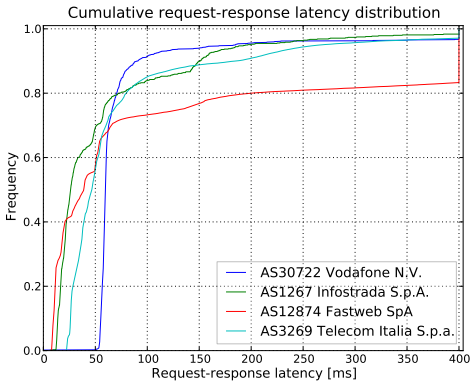
Results do not represent users' broadband connection speed: instead, they represent the average quality experienced by users that perform the same test with different Internet Service Providers and different broadband connections. They are more valuable as a tool for differential analysis than as a tool for measuring the speed magnitude.

The test methodology depends on the version of Neubot and is explained in section 3.3. In addition, the methodology does not account for confounding factors such as poor wireless connectivity and the activity of other users that share the same home network, as discussed in Section 3.4.

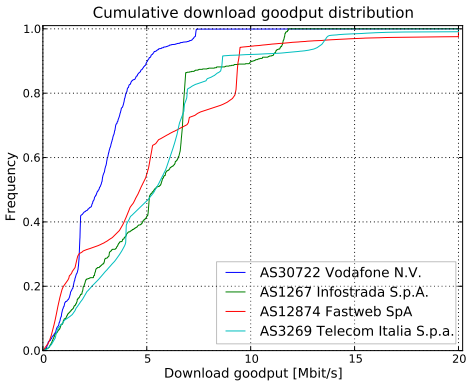
Figure 1a shows the cumulative distribution of the request-response latency for the tests performed in the Turin area. The distribution is plotted in the range 0-400 ms and greater values are assigned to the last bin. We use request-response latency because time required to connect is buggy in old versions of Neubot, as explained in Section 3.1. About 20% of Fastweb tests have latency higher than 400 ms: this is caused by 20 Neubots that consistently report very high latencies[1].

Figure 1b shows the cumulative distribution of the download goodput for the tests performed in the Turin area. The distribution is plotted in the range 0-20 Mbit/s and greater values are assigned to the last bin. The goodput is calculated at the receiver, dividing the number of bytes transmitted over the time elapsed from sending the request until receiving the whole response. If we except Vodafone, which has a single 7-Mbit/s-download–512-Kbit/s-upload commercial offer, results are collected by users with different broadband speeds and contracts. It would be interesting to classify the results depending on the highest measured speed and plot the contribution of each class separately.

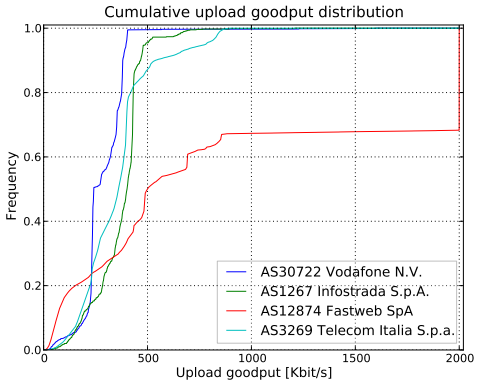
[1]Incidentally, this seems not to be caused by the fiber vs. ADSL split of Fastweb customers. There are many Neubots with less than 1 Mbit/s upload both in the more-than-400-ms camp and in the other camp.



(a) RTT



(b) Download goodput



(c) Upload goodput

Figure 1: This figure shows the cumulative distribution of request-response latency, download and upload goodput for the tests performed in the Turin area. The distribution is plotted in a given range and greater values are assigned to the last bin. The test methodology changes slightly depending on the version of Neubot and does not account for confounding factors such as poor wireless connectivity and the activity of other users that share the same home network (see Section 3). In addition, one must consider that the maximum goodput that a user can reach depends, of course, on the maximum speed of the access link, as indicated in the contract.

Figure 1c shows the cumulative distribution of the upload goodput for the tests performed in the Turin area. The distribution is plotted in the range 0-2000 Kbit/s and greater values are assigned to the last bin. The goodput is calculated at the sender, dividing the number of bytes transmitted over the time elapsed from sending the request until receiving the whole response. Vodafone apart, results are collected by users with different broadband speeds. For example, 35% of Fastweb tests are above 2000 Kbit/s, clearly due to Fiber-To-The-Premises connections.

Scripts and aggregate data used to produce plots in Figure 1 are available on the Neubot website for other researchers and institution to carry out independent analysis. The whole dataset of raw Neubot results is available online as well. Whenever the user provided the permission to publish (see Section 5), the result retains the original Internet address. Otherwise, the Autonomous System name and estimated geographic location are provided.

5 Privacy policy

The result of a transmission test is a tuple that contains performance metrics, such as goodput and latency, information on the user's computer load, and the user's Internet address. Neubot needs to

collect test results on its servers to be able to study them and publish aggregate data. In addition, the project would like to publish the raw results, to allow other individuals and institutions to reuse them for research purposes.

Both the collection and the publication of the results, and, in particular, of Internet addresses, are instrumental to Neubot goals. The projects needs to collect data to accomplish its goal of studying the quality and neutrality of users' Internet connections. In particular, without the Internet address it is not possible to associate the results to an Internet Service Provider and a geographic location. The publication of raw results is very relevant as well, albeit it is not so critical as the collection. The availability of public data allows independent researchers and institutions to study, question and analyze the project methodology and research results. This will benefit Neubot developers and users, as well as the Internet community at large – eventually leading to a better shared understanding of the 'Net. And, again, no serious study of the data can be carried out without knowing the original Internet address.

However, Internet addresses cannot be collected and published as easily as goodput and latency measurements. Indeed, according to many opinions, including the ones of Ohm [39] and of the Article 29 Working Party[2] [40] [41], Internet addresses are personal data and they must be treated under the provisions of Italian and European Privacy Laws (Decree 196/03: "Codice in materia di protezione dei dati personali") [42]. This means that, in general, the user's *informed consent* is required for each purpose – collection and publication in this case – of the personal data treatment. There are exceptions to this regime, which allow public institutions, such as the Polytechnic University of Turin, to collect personal data without asking for the permission to do so [43]. However, Neubot does not take advantage of this possibility for two reasons. First, data collected in this way cannot be published freely on the web. Second, providing users a privacy policy and asking for their explicit permission is more transparent and fair.

According to the Law, Neubot installer shows users the privacy policy, informs on personal data treatment procedures and explains users how to exercise their rights. Then, asks users to: (i) assert that they have read the policy; (ii) provide the permission to collect their Internet address for research purposes; (iii) provide the permission to publish their Internet address for reusing them for research purposes. The permission to collect is mandatory, as explained above: if the user does not provide it, the installation aborts. The permission to publish is optional. If it is not provided, the user Internet address will not be published. And the project will publish instead the Internet Service Provider name and the (estimated) geographic location[3]

6 Conclusion and future work

In this work we present an exhaustive documentation of Neubot's *speedtest* test behavior. The test estimates round-trip time, download and upload goodput. Round-trip time estimation is performed (i) measuring the time it takes for the `connect(2)` system call to complete and (ii) measuring the request-response application-level delay for small requests and responses. The goodput is calculated dividing the amount of bytes sent (or received) over the elapsed time. The test uses two concurrent HTTP connections, but used four connections in the 0.3.5-0.3.6 version range. The download estimation is performed at the receiver, the upload estimation at the sender. Starting from version 0.3.7 the receive socket buffer is fixed at 256 KiB.

The major contribution of this paper is the discussion of the test's methodology. Which does not account for confounding factors such as poor wireless connectivity and the activity of other users that share the same home network. Considering all the caveats, results are still a valuable tool for differential analysis of providers' performances. But they are not an effective measure of the real broadband connection speed.

Future versions of Neubot should dedicate one test to estimate the quality of the network, using

[2] A body composed of the data protection authorithy of each EU Member State, the European Data Protection Supervisor and the Europen Commission.

[3] We use MaxMind `<http://www.maxmind.com/>` free services to map the Internet address to the Internet Service Provider name and to the estimated geographic location.

just one connection, and another test to approximate the real broadband connection speed, using as many connections as needed. Moreover, given that Neubot has been accepted by Measurement Lab distributed server platform, the fixed buffer limitation should be relaxed (possibly adding another confounding factor), to be able to test high-speed high-latency networks. Finally, the test should be modified to run for a target number of RTTs, not seconds, to allow for comparison of connections with very different RTTs.

The article presents the Neubot measurements in the Turin-area, in the May-September time interval. Measurements have been divided per provider and the four commercial providers with more tests and Neubots have been analyzed. The analysis consists of the cumulative distribution of request-response latency, download and upload goodput. In the latency distribution, it would be interesting to investigate why a significant fraction of Fastweb users systematically experience very high latencies. Vodafone apart, the download and upload goodput distributions convey contributions from users with different broadband connections and contracts. It would be interesting to classify the users depending on their broadband connection and then plot the performances of each class separately, possibly asking users to indicate their broadband connection type.

Of course, the public availability of the raw data – made possible by the project privacy policy – allows independent researchers and institutions to review Neubot methodology and perform alternative analysis.

Acknowledgment

We would like to thank Att. Monica Alessia Senor, Dr. Eleonora Bassi and Dr. Federico Morando for their support and for making suggestions to improve the quality and readability of this paper.

We would also like to thank Mr. Matt Mathis, who prodded us to relax the fixed-buffer limitation, making the case of high-bandwidth high-delay networks in Korea and Japan.

References

[1] B. Van Schewick, "Towards an economic framework for network neutrality regulation," *J. on Telecomm. & High Tech. L.*, vol. 5, p. 329, 2006.

[2] J. Crowcroft, "Net neutrality: the technical side of the debate: a white paper," *ACM SIGCOMM Computer Communication Review*, vol. 37, no. 1, pp. 49–56, 2007.

[3] J. Palfrey. A Citizens' Choice Framework for Net Neutrality. [Online]. Available: http://blogs.law.harvard.edu/palfrey/2010/11/03/a-citizens-choice-framework-for-net-neutrality/

[4] M. Lemley and L. Lessig, "End of End-to-End: Preserving the Architecture of the Internet in the Broadband Era," *Ucla L. Rev.*, vol. 48, p. 925, 2000.

[5] S. Jordan, "Some traffic management practices are unreasonable," in *Computer Communications and Networks, 2009. ICCCN 2009. Proceedings of 18th Internatonal Conference on.* IEEE, 2009, pp. 1–6.

[6] R. Clayton, S. Murdoch, and R. Watson, "Ignoring the great firewall of china," in *Privacy Enhancing Technologies*. Springer, 2006, pp. 20–35.

[7] J. Zittrain and B. Edelman, "Internet filtering in China," *Internet Computing, IEEE*, vol. 7, no. 2, pp. 70–77, 2003.

[8] Internet blocking booklet. [Online]. Available: http://www.edri.org/internet-blocking-brochure

[9] Netflix to FCC: scary loophole in net neutrality rules. [Online]. Available: http://arstechnica.com/tech-policy/news/2010/01/netflix-to-fcc-dont-make-managed-services-a-net-neutrality-loophole.ars

[10] How will we know when the Internet is dead? [Online]. Available: http://arstechnica.com/tech-policy/news/2010/11/are-you-on-the-internet-or-something-else.ars

[11] H. Kim, K. Claffy, M. Fomenkov, D. Barman, M. Faloutsos, and K. Lee, "Internet traffic classification demystified: myths, caveats, and the best practices," in *Proceedings of the 2008 ACM CoNEXT conference*. ACM, 2008, p. 11.

[12] S. Floyd and V. Jacobson, "Link-sharing and resource management models for packet networks," *IEEE/ACM Transactions on Networking (TON)*, vol. 3, no. 4, pp. 365–386, 1995.

[13] OpenBSD. Configuring Queueing. [Online]. Available: http://www.openbsd.org/faq/pf/queueing.html#altq

[14] C. Systems. Weighted Random Early Detection (WRED). [Online]. Available: http://www.cisco.com/en/US/docs/ios/11_2/feature/guide/wred_gs.html

[15] M. Dischinger, M. Marcon, S. Guha, K. Gummadi, R. Mahajan, and S. Saroiu, "Glasnost: Enabling end users to detect traffic differentiation," in *Proceedings of the 7th USENIX conference on Networked systems design and implementation*. USENIX Association, 2010, pp. 27–27.

[16] P. Kanuparthy and C. Dovrolis, "Diffprobe: detecting ISP service discrimination," in *INFOCOM, 2010 Proceedings IEEE*. IEEE, 2010, pp. 1–9.

[17] Y. Zhang, Z. Mao, and M. Zhang, "Detecting traffic differentiation in backbone ISPs with NetPolice," in *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*. ACM, 2009, pp. 103–115.

[18] M. Tariq, M. Motiwala, N. Feamster, and M. Ammar, "Detecting network neutrality violations with causal inference," in *Proceedings of the 5th international conference on Emerging networking experiments and technologies*. ACM, 2009, pp. 289–300.

[19] Youtube Video Speed History. [Online]. Available: http://www.youtube.com/my_speed

[20] J. De Martin and A. Glorioso, "The Neubot project: A collaborative approach to measuring internet neutrality," in *Technology and Society, 2008. ISTAS 2008. IEEE International Symposium on*. IEEE, 2008, pp. 1–4.

[21] S. Basso, A. Servetti, and J. De Martin, "Rationale, Design, and Implementation of the Network Neutrality Bot." [Online]. Available: http://www.neubot.org/neubotfiles/aica2010-neubot-paper.pdf

[22] ——, "The network neutrality bot architecture: a preliminary approach for self-monitoring of Internet access QoS."

[23] Speedtest.net - The Global Broadband Speed Test. [Online]. Available: http://speedtest.net/

[24] R. Beverly, S. Bauer, and A. Berger, "The internet is not a big truck: toward quantifying network neutrality," *Passive and Active Network Measurement*, pp. 135–144, 2007.

[25] Switzerland Network Testing Tool — Electronic Frontier Foundation. [Online]. Available: https://www.eff.org/pages/switzerland-network-testing-tool

[26] M. Dischinger, A. Mislove, A. Haeberlen, and K. Gummadi, "Detecting bittorrent blocking," in *Proceedings of the 8th ACM SIGCOMM conference on Internet measurement*. ACM, 2008, pp. 3–8.

[27] NNMA – NNSquad Network Measurement Agent. [Online]. Available: http://www.nnsquad.org/agent

[28] N. Weaver, R. Sommer, and V. Paxson, "Detecting forged TCP reset packets," in *In Proc. of NDSS*. Citeseer, 2009.

[29] M. Tariq, M. Motiwala, and N. Feamster, "NANO: Network Access Neutrality Observatory," in *Proceedings of ACM HotNets*. Citeseer, 2008.

[30] Y. Zhang, Z. Mao, and M. Zhang, "Ascertaining the Reality of Network Neutrality Violation in Backbone ISPs," in *Proc. of ACM HotNets-VII Workshop*, 2008.

[31] WindRider – A Mobile Network Neutrality Monitoring System. [Online]. Available: http://www.cs.northwestern.edu/ ict992/mobile.htm

[32] BISMark – Monitor and Manage Your Home Network.

[33] Grenouille.com - la météo du net depuis 2000. [Online]. Available: http://www.grenouille.com/

[34] R. Prasad, C. Dovrolis, M. Murray, and K. Claffy, "Bandwidth estimation: metrics, measurement techniques, and tools," *Network, IEEE*, vol. 17, no. 6, pp. 27–35, 2003.

[35] M-Lab — Welcome to Measurement Lab. [Online]. Available: http://measurementlab.net/

[36] Goodput – Wikipedia, the free encyclopedia. [Online]. Available: http://en.wikipedia.org/wiki/Goodput

[37] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee, "RFC2616: Hypertext Transfer Protocol–HTTP/1.1," *RFC Editor United States*, 1999.

[38] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP throughput: A simple model and its empirical validation," in *ACM SIGCOMM Computer Communication Review*, vol. 28, no. 4. ACM, 1998, pp. 303–314.

[39] P. Ohm, "Broken promises of privacy: Responding to the surprising failure of anonymization."

[40] IP Addresses Are Personal Data, E.U. Regulator Says. [Online]. Available: http://www.washingtonpost.com/wpdyn/content/article/2008/01/21/AR2008012101340.html

[41] Opinion 1/2008 on data protection issues related to search engines. [Online]. Available: http://ec.europa.eu/justice/policies/privacy/docs/wpdocs/2008/wp148_en.pdf

[42] Codice in materia di protezione dei dati personali. [Online]. Available: http://www.garanteprivacy.it/garante/doc.jsp?ID=1311248

[43] Codice di deontologia e di buona condotta per i trattamenti di dati personali per scopi statistici e scientifici. [Online]. Available: http://www.garanteprivacy.it/garante/doc.jsp?ID=1556635