Making Internet Measurements Accessible for Multi-Disciplinary Research

An in depth look at using MLab's Glasnost data for net-neutrality research

Hadi Asghari
TU Delft
h.asghari@tudelft.nl

Milton L. Mueller
Syracuse University
mueller@syr.edu

Michel J. G. van Eeten
TU Delft
m.j.g.vaneeten@tudelft.nl

Xiang Wang
Syracuse University
xwang52@syr.edu

ABSTRACT

This paper documents the efforts of a group of social science researchers to make use of data generated by Glasnost, a test that detects whether Internet service providers are throttling specific applications. It shows how data gathering from a purely technical standpoint may not be optimal for social science purposes. The paper develops a five part framework to guide computer scientists' design of large-scale data collection efforts so that they can be useful for social science and policy work.

1. INTRODUCTION

Policy researchers in Internet-related fields, such as cyber security and network neutrality, benefit substantially from large-scale empirical data sets. Findings of projects based on global and longitudinal evidence are more reliable and insightful than those based on secondary sources and anecdotes. Luckily for policy researchers, quite a number of large data-gathering projects are actively working – for instance, those hosted by the Measurement Lab, the Internet Storm Center, and various botnet working groups are waiting to be used for research.

However in our experience, a mismatch exists between what the technical folks like to record and what the social scientists, economists and policy researchers actually need. Just as an example, consider the cases of packet dumps and detailed traces. These are often considered crown jewels by the engineers building the systems - they provide accountability, and enable answering new questions in the future. For the policy researchers, who are mostly interested in analysing much higher levels of aggregation, e.g. in months and ASNs, these detailed dumps are actually troublesome. Data that is stored in such detail requires downloading gigabytes of data just to *grep* for a few interesting bytes per test, an unnecessarily time consuming process. Social scientists also have to deal with problems of measurement error and pay attention to the statistical validity of data samples in ways that technical researchers may not fully appreciate.

In this paper we aim to shed light on this problem by describing our team's efforts to make use of the Glasnost logs to estimate Deep Packet Inspection deployment in ISPs worldwide. Glasnost is a test created by researchers at the Max Plank Institute and hosted on MLab that "enables ordinary Internet users to detect whether their ISPs are differentiating between flows of specific applications" [1], making it an ideal data source for our purpose. We discuss the various challenges involved in parsing, analysing and interpreting the logs.

The contributions of this paper are two-fold: (a) it provides a "cleaned up" version of the Glasnost data and documents the decisions we made along the way on how to use it for other teams interested doing similar work to ours; and (b) it provides a more general list of wishes and guidelines on how to store or structure Internet measurement results to make it easier for use for policy

researchers and social scientists. The significance of this exercise goes beyond the case of the Glasnost data. The same problem exists in many different datasets that we have seen. This paper provides just one in-depth example of the gap between social science and technical approaches to data-gathering. That gap is a problem for the technical people as well, for if it is not addressed their efforts to store and make accessible technical data will not have the policy impact that they deserve.

2. Deep Packet Inspection & the use of Glasnost

Our research investigates the deployment and governance of deep packet inspection (DPI), a new technology for monitoring and controlling Internet communications by network operators. DPI allows a network operator to recognize specific applications, or even specific items of content, as they flow through the network in real time, and then take actions based on that recognition. Those actions can be beneficial – e.g., recognizing and thwarting spam or malware – or politically controversial – e.g., rationing bandwidth, blocking access to censored content, or building user profiles for advertisers. With the dual potential to improve the operation and governance of the Internet and to restrict or regulate it in repressive ways, DPI use is being actively contested politically. Many operators have deployed it secretly, and some have even denied that it was being used after it was discovered. In broad terms, we wanted to find out what impact this new technological capability is having on the way the Internet is governed. That led to a set of more specific research questions:

- Which operators are using DPI to discriminate among applications and which are not?
- When they do discriminate, do they throttle or block, and do their policies affect uploads, downloads or both?
- How do operator practices respond to regulations and laws aimed at DPI or net neutrality?

One way to get this information is to ask the operators, but as a research method that is not really viable. There are thousands of different ASNs in the world and it would be costly and time consuming to survey all of them. Aside from that, the operators have nothing to gain by telling researchers about their technical policies and much to lose. And how would one know they were telling the truth? So an alternative strategy is for network users to run tests that reveal what is actually happening to their traffic.

A network test known as *Glasnost* was developed by researchers in Germany to detect blocking or throttling of BitTorrent and other peer to peer (P2P) file sharing protocols. The detailed workings of the Glasnost test are described in Dischinger, Marcon, et al [1]. Thanks to an initiative known as the Measurement Lab (MLab), the Glasnost test was placed on a global platform so that end users all over the world could run the test and the results would be stored and made available to researchers. This way of crowdsourcing the generation of network performance data provided the researchers with data for the last three quarters of 2008, all of 2009 – 2011, and the first quarter of 2012. An Internet user who runs the Glasnost test can see whether BitTorrent (or other protocols) is completely blocked, slowed down, or running unhindered. The results also allow us to infer whether the BitTorrent manipulation took place through DPI or through simpler port blocking.

The Glasnost client transfers data between the client and the Glasnost measurement servers, using both a random bit-stream and the BitTorrent protocol, on different TCP ports. It subsequently compares the transfer speed among these different application flows. Based on a set of criteria it determines whether the ISP is slowing down or blocking BitTorrent, and if so, whether discrimination is based on TCP port numbers or actual recognition of the BitTorrent protocol on any port. The latter, we infer, is only possible if DPI is being used. The servers log all test information along with the client's IP address and the time of the test.

These empirical tests can often produce surprising and unexpected results. For example, Canada's telecommunications law and CRTC regulations make it illegal for operators to discriminate between and meddle with traffic. Additionally, Canada adopted a comprehensive set of principles governing "Internet Traffic Management Practices (ITMPs)" in 2009. Yet in Canada, use of DPI to throttle BitTorrent was either

unaffected or increased after the ITMP decision. In contrast, the US telecommunications law does not classify ISPs as common carriers, and the Federal Communications Commission's attempt to regulate network management practices was struck down by the courts in 2010. Yet network tests using Glasnost revealed that in the US, the use of DPI to throttle BitTorrent virtually ceased after August 2008, when the FCC signalled its displeasure with it [2]. One would certainly not expect this to be the result, and without empirical tests, one would have no way of knowing it. This is the power of using measurement datasets in such studies; when such datasets are not accessible to policy researchers it severely limits the accuracy and relevance of their work.

3. MAKING MEASUREMENTS ACCESSIBLE

3.1 ISSUES WITH THE GLASNOST DATA

The Glasnost test was an invaluable tool for our research. Although not specifically designed to detect DPI per se, by testing both the BitTorrent port and the neutral port it made reliable inferences about DPI use possible. Additionally, decisions by Google and the New America foundation and PlanetLab to support the infrastructure for conducting the tests and storing the data logs were crucially important public benefits.

But running the tests and storing the logs is really only the first and second steps in a four-step process. Transforming these logs into a dataset useable in statistical tools is the third; the fourth is experimenting with models, adding independent variables, uncovering patterns, and validating and interpreting the results. Ideally, policy researchers would like to focus on this final step. In practice however making sense of the raw data (the third step) turns out to be more challenging than expected and perhaps necessary. Just to give some examples, one could name scant documentation, issues with false positives and negatives, anomalous results, corner cases, aborted tests, and various issues of statistical validity.

If these issues are not handled carefully, misunderstandings and backlash over publishing of the data can pursue. This was exemplified by a *New York Times* article regarding the Glasnost test published on November 13th 2011 titled "*Putting the Brakes on Web-Surfing Speeds" [3]*. The article stated that *"throttling was detected in 18% of tests on Verizon's landline network"* and reported similar percentages for other ISPs, some of which do not in fact employ application-specific throttling mechanisms. Anger ensued from the ISPs and serious questions about the validity of Glasnost were asked by researchers and regulators. The Max Plank Institute, host of the Glasnost project, published an explanation on November 28th in effect retracting the NYT published figures. The explanation states that *"when inferring whether an ISP is deploying traffic shaping, it is important to view those percentages in the context of the potential measurement errors, all of which were not emphasized sufficiently … for example, for US ISPs like AT&T and Verizon the percentages of tests that were reported to have detected traffic shaping fall within the range of false positives"* [4].The point here is to highlight the importance all steps have when using measurement data for a work with policy implications.

3.2 THE MISMATCH BETWEEN WHAT COMPUTER SCIENTISTS AND POLICY RESEARCHERS SEEK IN DATA

To understand the gap between how measurement data is stored and how it later needs to be used, one has to take into consideration the different requirements and challenges that computer scientists and social scientists face.

Social scientists using econometric techniques are interested in having well understood dependant variables as a starting point. This can be combined with independent variables and, using a variety of statistical methods, the econometrician can explore and evaluate various hypothesis and causal models, and interpret the results. Key values include: *longitudinal*, *reliable*, and *aggregated* data (aggregated at the level of countries, organisations, or whatever other independent variable is considered to be interesting from a social science standpoint).

Computer scientists often need to remain as close as possible to the raw data. This allows for accountability and validation so that for instance if an ISP denies deploying DPI they can be presented with the packet dumps of the test. It also enables asking new questions

to the data and mining it in previously unthought-of ways (to a certain extent of course). In the case of Glasnost, the tendency to stay close to the raw data went so far that the test verdicts - results shown to the user - were not even stored. The idea was that one could rerun all calculations again using the raw data should the need arise. In another dataset that we use, the IP addresses of millions of bots are extracted per day, but not stored for similar reasons.

This difference of viewpoints manifests itself in the size of datasets: whereas social scientists end up working with datasets of kilobytes and megabytes, it is quite normal for raw logs to be gigabytes or terabytes in size. Even though cases where all the data is logged are the better ones, in many cases historical data is not kept at all, as many security systems are used in real-time. (We are often reminded to feel free to build a system that logs that particular data; the only problem is that this cannot be used to look into the past).

Technologies such as Hadoop - advocated as a powerful distributed processing solution for big data, cannot bridge the gap mentioned here, as they do not address the underlying issues.

3.3 KEY IDEAS FOR THE IDEAL DATASET

Thinking about these issues would be better done when building the test infrastructure in the first place, rather than as an afterthought. So what would an ideal data source look like? In our experience based on working with several large measurement datasets in the past few years (not just Glasnost), it should meet the following five criteria:

1. ***Provide access to filtered and aggregated versions of the data:*** Usually quantitative policy research involves working with data aggregated at the level of country or organisation and over periods of weeks, months, quarters and years. Giving researchers the ability to download these more limited versions of the data is actually helpful. Not only does it reduce download and processing times, but it can also solve the problems of privacy in certain cases and enable opening up the data for more people.
2. ***Consistency of data over time***. The consistency of a measurement instrument over time is very important in social science. Measurement researchers often play around with various parameters in their systems to see which creates the best results. However from a social science standpoint one cannot simply stick the results derived from different measuring instruments together – the changes in the instrument will yield different measurements and thus disrupt longitudinal analysis. Maintenance of the testing infrastructure is also critical so that there are no gaps in the recorded data.
3. ***Organize data collection to promote statistical validity.*** The number of tests and, when crowdsourcing is involved, their distribution over various parts of the population, is extremely important for statistical validity. If we lack enough tests from a particular ASN, country, or use case to draw statistically valid inferences, we simply cannot use the data. A good dataset will thus be based on collection practices that incentivise users to perform particular measurements so that we have an adequate sample size for each observation unit.
4. ***Individual measurements need to have verdicts and interpretations that are as clear as possible***. It is very hard to make decisions on how to interpret individual measurements correctly, for a test created by another person, if the verdict is ambiguous. It forces scientists from other disciplines to study the technical details of a system they have not implemented and for which they might lack important knowledge about went into building the system. Anomalies and corner cases make the process even harder. *(Identifying false positives/negatives also falls into this category)*
5. ***Having a support infrastructure to facilitate understanding and working with the data***. As M-Lab has proven to us, having an active mailing list, access to the test authors, and providing parsing scripts are all very valuable when it becomes necessary to delve deep into the raw data, even in cases of scarcity of documentation and other issues outlined.

There are two more issues in policy work, but these are broader problems that need to be handled collectively by

the measurement community and not for one particular measurement.

One is that ***historical ASN & country level lookups*** are needed, as the country and ASN that an IP belonged to at the time of the measurement is important. Telling us who holds the IP address now, which is what current infrastructure does, is of no use for longitudinal data sets. Our team's solution is described in section 4.

Another common problem is that of ***aggregating ASNs into operators***, i.e. the actual business and organisational entities. As far as we are aware, the only way to do this is via manual mapping of the ASNs to the list of ISPs in different countries.

4. Processing the Glasnost data

In this section of the paper we will describe the steps involved in processing the glasnost logs and turning them into a dataset suitable for policy research. Our initial impression, based on what we had read about the test, and our prior experience with processing large datasets was that this should be relatively straightforward. However, this turned out not to be the case. The details are given and in the process the Glasnost data is benchmarked against the ideal dataset guidelines.

4.1 Level of aggregation

The Glasnost data is stored at the level of individual tests on Google Storage. For each test, a server log and a packet dump are stored. Only the log is of use in our work. All the logs for a single day and server are grouped into one compressed file and accessible via the *gsutil* tool.[1]

To give an indication of the extraneous data that has to be downloaded: for February 2012 alone, 115 GB of compressed data has to be downloaded. This takes several hours to download on a campus gigabit connection (the speed limit is due to the way gsutil functions apparently). After extraction and removal of the packet-dumps, we are left with 33 GB of uncompressed data and around eighty thousand test logs. Out of each log, only a few lines are actually needed (some header data and a few summary lines at the end of each log), which when extracted leave us with under 40 MB of useful data. Thus, the actual data needed to process the logs is less than 0.1% of the total downloaded data.

For each test, we were interested in knowing whether it was indicative of BitTorrent throttling or blocking or not. Although these results are calculated and shown to the user when they conduct the test, they are not stored in the logs. This made it necessary to parse and analyse the logs to re-calculate the result. The format of the log files changes after May 2010, with the newer logs allowing more protocols to be tested and adding a summary line to the end of each log. Based on our request to Google, parsing and analysis scripts were provided for these newer logs. For the first batch of log files, we were forced to write our own parser based on the Glasnost server code. Due to anomalies in the logs and lack of documentation, this took a considerable amount of time.

4.2 ASN & country-code lookups

Before going into details about interpreting the tests, there are some general steps that need to be taken: mapping each test client's IP address to its country and autonomous system. For the first purpose, we make use of the MaxMind GeoIP country database [5] which provides good accuracy and historical versions going back for several years. For the second purpose, we use PyASN [6], a Python library developed by ourselves that performs fast IP to ASN lookups using historical RViews BGP data. These methods were developed for previous research at Delft University of Technology using other large measurement datasets [7].

4.3 Consistency over time

The Glasnost data has several breaks in the logs. First, the log formats change: the 2008 data is available only as a CSV file by direct request from the authors; January 2009 to April 2010 tests use what we call the version 1 logs; and May 2010 onwards use the version 2 logs. There were good reasons for changing the formats, and the change did not cause any statistical problems, but it did add another task to the research process.

The second break, which has statistical relevance, is due to changes in the default Glasnost test parameters. The measurement duration, repetitions, and directions were

[1] `./gsutil cp gs://m-lab/glasnost/YYYY/MM/* .`

changed during the test life. Prior to August 2009, the measurements were repeated twice and conducted in both directions. During August to October of that same year, the Glasnost project experimented with setting the test to use different repetitions, duplex and durations, to see which measurements would yield the fewest false positives and false negatives, yet not be so long as to bore the users into stopping the test. These changes create very visible jumps in the results' percentages. (See Figure 1)

TABLE 1 – GLASNOST LOG DIFFERENCES OVER TIME

Period	Log format	Protocol	Measure. repetitions	Test direction
04/2008–01/2009	CSV	BT	2	Both
02/2009-07/2009	V1	BT	2	Both
08/2009-10/2009	V1	BT	2 to 5	Single & Both
11/2009-04/2010	V1	BT	3	Both
05/2010–02/2012	V2	Many (incl. BT)	3	Both

The third break which is that for several periods in 2010 and 2011, the longest of which spans several weeks (12-Oct-2010 and 25-Nov- 2010), no test logs exist. This was due to an unfortunate rsync problem between the M-Lab servers and Google Storage, the result of which has been the loss of data.

4.4 TURNING LOGS TO VERDICTS

Our need was to have a simple verdict for each test run: does it indicate the presence of application based throttling or blocking (hence, DPI), or not?

This turns out to be very involved, for three reasons: (1) the test results that are shown to the end user are not stored in the logs; detailed information about each measurement flow is what is recorded; this necessitates reanalysis of the measurements; (2) the Glasnost server code that performs the analysis of the measurements does not calculate a *combined* judgement, but rather reports individual results for upload, download, throttling and blocking; the end-user herself deduces what this means, which in our case has to be done programmatically; (3) the data contains quite a number of corner cases and anomalies which further complicate matters, as we shall see.

We will have to start by giving a crash course on how the Glasnost test works. The client and server run a series of measurements in which they transfer bytes of data using different protocols (the application being tested versus a random bit-stream) and different TCP ports (the specific port assigned to the tested application versus a neutral port). The speed of each measurement is recorded, in addition to whether it was interrupted for some reason. Now, if we compare the speeds of these flows (listed in Table 2) and for instance discover that the speed of the application being tested is much slower than the speed of the control flow then we can conclude that the ISP is performing application-based throttling, as presumably the client's link and the server's link speeds remain the same. There is one caveat however: the Internet routes traffic on a 'best-effort' basis, which means that some speed fluctuations are to be expected. For this reason, certain thresholds have to be met: first, the flow speeds have to differ more than 20%; second, to make sure that the connection is not too noisy, each measurement is repeated three times, and these have to yield similar results (i.e., measurements of the same flow should have less than 20% difference). The choice of these thresholds is explained in the aforementioned paper by Dischinger and his colleagues. The simple comparison cases are illustrated in Table 3.

TABLE 2 - GLASNOST MEASUREMENT FLOWS

Flow #	Direction	Port	Protocol
0	upstream	app-port	application
1	upstream	app-port	control-flow
2	upstream	neutral-port	application
3	upstream	neutral-port	control-flow
4	downstream	app-port	application
5	downstream	app-port	control-flow
6	downstream	neutral-port	application
7	downstream	neutral-port	control-flow

The cases get more complicated when considering *noisy flows*, and also a condition known as *broken measurement*. A broken measurement is when either the transfer does not start (duration is zero) or it starts but no data is transferred (bytes transferred is zero). This is

indicative of some form of network blocking and if it happens more than once on a particular flow, we consider that flow as failed and cannot use it for speed comparisons (all is not lost, however, for depending on which port the flow occurs on, some judgements can be made.)

TABLE 3 – DETECTION OF THROTTLING - SIMPLIFIED

Interim	Comparison	Meaning
AD1	*app-flow on app-port* << *control-flow on app-port*	App-based throttling
AD2	app-flow on neutral-port << control-flow on neutral-port	App-based throttling
PD1	app-flow on app-port << app-flow on neutral-port	Port-based throttling
PD2	control-flow on app-port << control-flow on neutral-port	Port-based throttling

If we take into account the noisy and failed flows, we end up with the Table 4 to detect app-based throttling. A similar table can be drawn for port-based throttling, and both sub-verdicts can be made for both the upstream and downstream directions.

TABLE 4 - DETECTION OF APP-BASED THROTTLING - DETAILED

AD1-comparison	AD2-comparison	App-based throttling?
af-ap or cf-ap is noisy or broken	af-np or cf-np is noisy or broken	too noisy to tell
af-ap << cf-ap	af-np << cf-np	Y
af-ap << cf-ap	noisy or broken	Y
af-ap << cf-ap	af-np >= cf-np	Y, with warning*
noisy or broken	af-np << cf-np	Y
af-ap >= cf-ap	af-np << cf-np	Y, with warning*
af-ap >= cf-ap	af-np >= cf-np	N
af-ap >= cf-ap	noisy or broken	N
noisy or broken	af-np >= cf-np	N

** this is the strangediff warning explained later*
af-ap = app-flow on app-port, etc.

The next step in the analysis is deciding on the meaning of failed flows. A flow fails when either a TCP-RST packet is seen during the measurement that was neither sent by the server or the client, or when no data is transferred at all (the measurement is broken) for at least two out of the three repetitions. Different combinations of flows might fail, and the implication will differ based on the combination. Should the *control-flow* on the *neutral port* fail, this is for sure indicative of a noisy measurement or some other form of malfunction, as there is never any basis for this flow to be blocked! The remaining combinations are listed in Table 5.

TABLE 5 - COMBINATIONS OF FAILED FLOWS

af-ap	af-np	cf-ap	Block mechanism
blocked	blocked	blocked	dpi-based & port-based
blocked	blocked	ok	dpi-based
blocked	ok	blocked	port-based
blocked	ok	ok	dpi-based **?**
ok	blocked	blocked	noisy (undefined combi.)
ok	blocked	ok	noisy (undefined combi.) **?**
ok	ok	blocked	noisy (undefined combi.)
ok	ok	ok	----

At this stage we have created several sub-verdicts that need to be combined into a final grand verdict for the test at hand. This is done in Table 6. The verdict DPI means that the test indicates DPI-based throttling or blocking is being performed by the ISP; PORT indicates PORT-based throttling or blocking is occurring; UNDEF means that the result is undefined -- due to noise we cannot deduce anything from this test; OK means that the test could not detect any form of traffic manipulation; and finally, OK1/2 cases are where the result is most likely OK but could also be other things.

TABLE 6 - CALCULATING A TESTS FINAL VERDICT

App-based throttling	Port-based throttling	Block mechanism	Test verdict	N
throt	*	*	**DPI**	12
ok	*	dpi	**DPI**	3
noisy	*	dpi	**DPI**	3
ok/noisy	throt	port/noisy/---	**PORT**	6
ok/noisy	ok/noisy	port	**PORT**	4
ok/noisy	ok/noisy	noisy	**UNDEF**	4
noisy	noisy	---	**UNDEF**	1
ok	noisy	---	**OK1/2**	1
noisy	ok	---	**OK1/2**	1
ok	ok	---	**OK**	1
				36

Table 6 shows results for a single direction. The verdict for both directions is calculated similarly. The exception is for unidirectional tests, where if the overall verdict is OK, it is changed to UNDEF, as we do not have any information on the other direction and all verdicts are possible outcomes.

Please note that these are all based on completed tests. In reality, the majority of the Glasnost tests never finish

and are aborted mid-way, and for such tests the analysis step is skipped all together.

The originally provided Glasnost analysis scripts only conduct simple comparisons and the more involved steps had to be developed by us. It required lengthy, demanding work. Thus we score Glasnost very low with regards to clarity of verdicts and ease of interpreting individual test results.

4.5 Strange and anomalous cases

In the Glasnost data there are quite a number of logs that we can only classify as odd – they might be indicative of corner cases, misconfigurations of the user's machine, or unstable network conditions. These are listed and explained in Table 7. We flag these cases in our database and in most cases take no further action.

TABLE 7 – WARNING FLAGS FOR VARIOUS ODD MEASUREMENTS

Flag	Explanation	%*
btfaster	The BitTorrent flows are significantly faster than the random flows!! Why would an ISP do this?	14%
cfnpfail	The random bit-stream on the neutral port is blocked! On what basis should this happen?	6%
strangediff	In app-based throttling, one of AD1 or AD2 indicates throttling while the other indicates its absence, which doesn't make sense! (Same for port-based)	9%
portchange	The application specific port was changed in the middle of the test (this sometimes happens). If the final verdict was PORT, we change it to UNDEF.	1%

** percentage of completed tests with this flag set*

A point that we briefly made in the previous section was that a majority of the tests (60%) are aborted midway by impatient users. In our own tests that we performed at the Syracuse campus we however discovered a strange caveat: the Glasnost server was logging our tests as aborted after the first flow, whereas we had patiently waited for its completion. Syracuse campus uses a DPI device to block all P2P traffic, both based on application signature and TCP port number. In correspondence, the Glasnost authors clarified that when the port is outright blocked and the TCP handshake cannot complete, the server assumes the user has aborted test after a certain timeout. It is unclear to us however what the difference is between this form of aborted test and instances where the test completes but certain flows fail (as indicated in Table 5), and our follow up questions to the test authors yielded no satisfactory explanation. What worries us most is that 9,000 of the tests that are thrown out as anomalies both in our scripts and in the original Glasnost analysis scripts follow this pattern (aborted after one flow with and a runtime of approx. 50 seconds). These would in fact make some of the interesting cases indicative of ISP manipulation. Further research is recommended on this issue.

4.6 Support infrastructure

Overall, the M-Lab team provide a good support infrastructure for making use of the Glasnost data. Despite the difficulties, complexities and unhandled cases explained in the previous sections, the following positive aspects facilitated our work:

- Support from Google / M-Lab
- An active mailing list
- Parsing scripts that acted as a foundation to build on and test against
- Access to the test authors who clarified issues
- The original paper outlining the test design

Over time, the quality of the infrastructure improved, indicating a learning process for all parties. In terms of documentation, the Glasnost paper served as a very good starting point but often more was desirable.

4.7 Issues with sample size

Although we cannot really blame any crowd-sourced measurement project for inadequate sample size in specific observation groups, methods can be thought of to remediate such a problem.

Two specific problems of sample size exist in the Glasnost data. First, although the test has supported testing many different protocols since May 2010, BitTorrent still makes up the vast majority of the tests (see Figure 1). The reason is clear: BitTorrent is selected as the default choice of the test interface. This is obviously a pity and could be remediated by the Glasnost test site setting different defaults for different users visiting the site, based on what tests are lacking for that user's particular ISP.

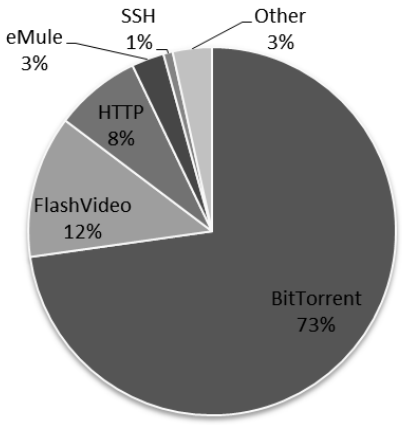

FIGURE 1 - COMPLETED TESTS BY PROTOCOL (N=109,000)

A serious sampling problem also exists for ASNs with 10 or less tests per time period. In such ASNs, extreme variation in the results – at times oscillating from 0% to 50% - is seen. This is easy to understand, as due to the low number of tests one positive result added or removed makes a large difference. The remedy again would be similar but at a broader level: the M-Lab website should recommend users to run the Glasnost test when the total number of tests from that ASN is low. Another idea might be for Glasnost (or its future descendants) to become integrated with the likes of the *SamKnows* project.

4.8 The "cleaned" dataset

At this point, after several months of work to download, parse, fully understand and analyse the raw Glasnost logs, we have our "cleaned" dataset - one that is ready for econometric analysis. In the ideal situation, this would have been our starting point!

This dataset contains approximately 326,000 completed tests for the period 2009 – 2011. These tests were run by users of approximately 1,400 ASNs located in approximately 100 countries[2]. In the last quarter of 2011 and first two months of 2012, we have about 32,000 tests with only 35 countries and 353 ASNs with adequate sample size. The completed tests account for around 40% of the total Glasnost logs; in other words, 60% of the tests had to be thrown out as invalid.

Figure 2 shows how the test verdict percentages change over time. When we look at the data at the global world-wide level, it seems relatively stable. However, when we open up the data and look at it at the level of individual countries or ASNs, we see large changes in these patterns depending on specific circumstances. Overall, around a third of the completed tests indicate noisy as the result of at least one of their sub-verdicts, although a lower limit in the end stays undefined.

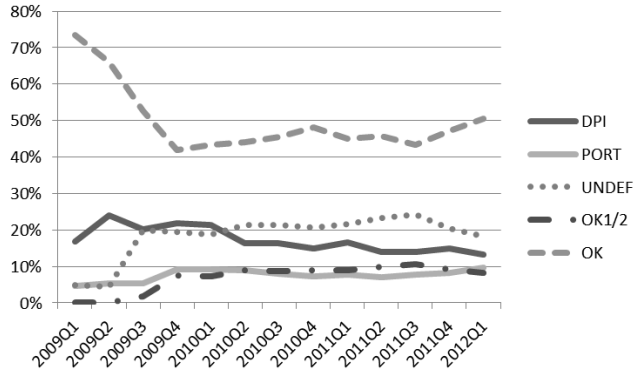

FIGURE 2 - TEST VERDICT PERCENTAGES OVER TIME

As the ending note, our team is publishing our processing and analysis scripts, along with parts of the processed data and some analysis on our project website [8] for other researchers to use.

5. Discussion and Conclusion

This paper developed a five part framework to guide computer scientists' design of large-scale data collection efforts so that they can be useful for social science and policy work. An ideal dataset would:

1. Provide access to filtered and aggregated versions of the data
2. Provide consistent measurements over time so as to enable longitudinal analysis
3. Organize data collection to support statistical validity
4. Enable clear verdicts and interpretations based on the measurements
5. Develop a support infrastructure.

The weakest point regarding the Glasnost data has been criterion 4 (clear verdicts). It took an enormous amount of work to decide whether a particular test indicates traffic manipulation or not. Based on the Glasnost project's purpose and description, one would expect this

[2] Counting countries with >100 tests and ASNs with >20 test

result to be prepared and included with the data. The time spent on understanding the measurement in this case is a major distraction from the policy research intended to be done. Glasnost was also weak on criterion 1, as a large amount of unneeded data had to be downloaded. Breaks in the continuity of the measurements over time were also limiting factors (criterion 2). Glasnost's handling of statistical validity issues (criterion 3) is mixed. Google and the New America Foundation made good efforts to publicize the MLab initiative generally. The number of tests conducted directly responds to these publicity efforts. When those efforts flag the number of users conducting tests dwindles. This dwindling interest is very noticeable in the last three quarters (Q3-4 2011 and Q1 2012), especially in Asia.

The strongest point regarding the Glasnost data is the MLab support infrastructure (criterion 5), to which Google seems to have made a serious corporate commitment. This made it possible to take steps to address the other problems and provided a clear safety valve for resolving issues as they came up.

Perhaps the most important benefit of the Glasnost test is the fact that it actually exists. Glasnost is the only tool available that allows researchers to know how often DPI is being used to detect and manipulate specific applications and by whom it is being used. It is a unique data source that provides answers to questions that otherwise could not be answered.

REFERENCES

[1] M. Dischinger, *et al.*, "Glasnost: Enabling End Users to Detect Traffic Differentiation," 2010.

[2] M. Mueller and H. Asghari, "Deep Packet Inspection and Bandwidth Management: Battles over BitTorrent in Canada and the United States," 2012.

[3 K. O'Brien, "Putting the Brakes on Web-Surfing Speeds," New York Times (Business Day section) November 14, 2011. Accessed at:] http://www.nytimes.com/2011/11/14/technology/putting-the-brakes-on-web-surfing-speeds.html

[4] http://broadband.mpi-sws.org/transparency/results/

[5] http://www.maxmind.com/app/country

[6] http://code.google.com/p/pyasn/

[7] M. Van Eeten, *et al.*, "The Role of Internet Service Providers in Botnet Mitigation: An empirical Analysis Based on Spam Data," 2010.

[8] http://deeppacket.info

[9] http://deeppacket.info