Measuring Broadband Performance using M-Lab: Why Averages Tell a Poor Tale

Xiaohong Deng, Jordan Hamilton, Jason Thorne, Vijay Sivaraman
The University of New South Wales
Email: {xiaohong.deng, j.thorne, vijay}@unsw.edu.au, jordan.hamilton@student.unsw.edu.au

***Abstract*—Broadband network performance is multi-faceted: it varies by ISP, by content source, by household connection, and by time-of-day. Daily or monthly averages, as published by content providers such as Netflix and Google, do not convey the full picture. In this paper we leverage M-Lab, the world's largest open measurement platform, to characterize broadband performance across Australian households. Our study delves into millions of data samples collected from 96,882 households over four months, and looks beyond averages to make several interesting observations: 1) There is considerable variation amongst households, in terms of their broadband speeds and variability of network performance within a day and across days, and this information is lost when data is averaged across houses; 2) The fluctuations (even for a specific house) are significant, and can exhibit unexpected patterns, such as wide variations from one day to the next, and some clusters of outliers at certain times of the day. 3) By our experimental results, we conclude that neither aggregating by household nor aggregating by day or by hour is a sound measurement strategy. Moreover, our study sheds new perspectives on broadband evaluation by using M-Lab data, and can inspire future study into the underlying reasons of performance variation.**

Keywords. Measurement, Bandwidth, M-Lab, Big data

I. Introduction

The task of network measurement and monitoring has become increasingly essential to provide for customer satisfaction, but is extremely challenging when new technologies emerge and produce dramatic growth in size and complexity of computer network systems. An important aspect of way the Internet is changing is that there are more bandwidth hungry applications. Peer to Peer (P2P) traffic used to be dominant, but since a few years ago video streaming has become and will continue to be the dominant traffic type on the Internet. Netflix and YouTube make up half of the peak-time Internet traffic in North America, accounting for 34.9 percent and 14.04 percent of downstream traffic, according to [1]. Globally, it is predicted IP video traffic will be 79 percent of all consumer Internet traffic in 2018 up from 66 percent in 2013 [2]. This phenomenon challenges the providers ability to measure and estimate resource requirements. In addition, study in network performance measurement is essential to network performance tuning and bandwidth planning.

Network performance data is mostly maintained by ISPs and is generally out of public reach. This invisibility of network performance measurement has raised people's concern on various issues, one of which is around network neutrality. Besides ISPs, Cloud providers such as Amazon Web Services (AWS) and video streaming providers, such as Youtube and Netflix are also greatly concerned about bandwidth and the users' experience. However, the Internet is a complex ecosystem, consisting of operators, cloud providers, content providers and other service providers. When the performance degrades, customers don't know which party to blame. The giants such as Google and Netflix have decided to publish the SP Speed Index periodically,to inform their customers the speed they can expect for video streaming. On their website Google state "There are many factors that influence your video streaming quality, including your choice of ISP. Learn how your ISP performs and understand your options".

In 2008, Vint Cerf initiated a conversation with researchers about challenges in the effective study of broadband networks. A result of this was the establishment of the Measurement Lab (M-Lab) consortium. M-Lab has built a platform on which test servers are well distributed across continents and ISPs, so that researchers and interested parties can design, implement and deploy new Internet measurement tools and collect data under an open license. In order to further enhance Internet transparency, M-Lab makes all the measurement data generated by a number of tools publicly accessible via various means. For example, a SQL-like query tool called BigQuery for querying structured data, and a cloud storage tool called Gsutil for downloading raw data from the cloud.

The amount of data accumulated from such systems is massive. Dramatic progress in computer systems in the last decade has made storage and accessibility of terabytes of big data possible, however, the methodology of how to make sense of such data is still far beyond clear. Fortunately, there are some visualization and statistical tools out there to help visualize and understand the data. The major challenge lies in how to combine tools and domain knowledge to find patterns and make sense of the data. In our work, we address this challenge by interpreting certain behavioral phenomenon suggested by outliers and spikes in visualized raw data.

In section III we analyze why the TCP/IP protocols' compensation ability is making network measurement challenging and why the widely used averaging method is exacerbating the situation. In section IV we show how we use our domain knowledge and statistics to sample the data for our study. We sample two households from the set of 96,882 Australian households and retrieve two months performance data of the sampled households. In section V, we visualize and analyze sampled data in different ways to identify outliers and spikes.

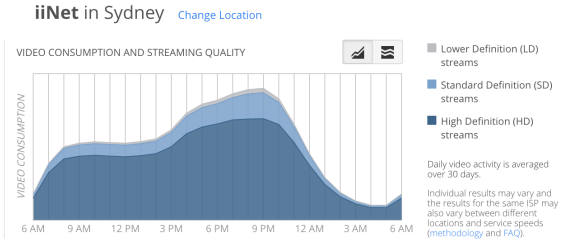

Fig. 1. Google's ISP Video Quality Report

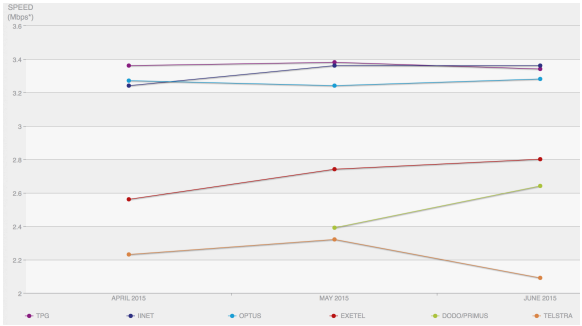

Fig. 2. Netflix's ISP index

We then discuss how our approach gives a better understanding of the performance data than The Simple Aggregated Method (SAM) mentioned previously. We conclude in section VI why averages produced by SAM tell a poor tale given the results we obtained.

II. Related Work

For the last few years, video streaming providers have been releasing ISP performance data to better inform customers of quality of service. Also, since the M-Lab platform was established in 2008, new tools have been designed to examine different aspects of network performance. In the following two subsections, we discuss what may be improved in the provision of performance information to customers.

A. Content providers' measurement data

Figure 1 from Google shows video streaming quality aggregated by ISP. For each ISP, it shows volume of Lower Definition (LD) streams, Standard Definition (SD) streams, High Definition (HD) streams over twenty-four hours.

Figure 2 from Netflix shows a list of Australian ISP's average speed of accessing Netflix's video across months. Each ISP has three data points: average speed of April, May and June and a straight line connects two consecutive months to show the trend across month for each ISP. Yet, to what degree those graphs aggregated by ISP, by hour and even by month are informative to users is uncertain. For example, in Netflix's chart, it shows TPG has a higher average speed than IINET in April. This suggests that if one house were subscribed to TPG, it would have a better Netflix experience than if subscribed to IINET. This is not necessarily the case.

B. M-Lab related work

Valuable research has been done upon M-Lab platform. In [3], authors have developed a tool and deployed its service on world wide located M-Lab servers to enable any Internet user to detect whether their ISPs are differentiating their P2P flows. The tool discovered a number of ISPs performing traffic shaping on P2P traffic as people suspected. Similarly, a tool called DiffProbe [4] and another one named ShaperProbe [5] were proposed and deployed on M-Lab infrastructure respectively for the detection of ISP Service Discrimination against different traffic types by employing the active probing method.

These are undoubtedly valuable research areas. Yet, they were mainly focused on developing new tools and generating more measurement data from different measurement angles. Little attention has been paid to mining m-Lab's extensive data sets and as a result enlightening information may have been neglected.

Empowered with advanced data mining and statistical tools, our study aims at providing a practical use case of mining m-Lab data from a novel perspective. We aim to find out inaccuracy of SAM in determining practical network performance expectation, and investigate better alternatives and their effectiveness by comparing results.

III. Why Estimating Network Performance is Difficult and Averaging Makes it Worse?

By definition, the goal of performance evaluation is to determine the effectiveness and correctness of a computer network system. The task of diagnosing and measuring performance is often complicated by the complex nature of the network, that is; degradation in performance often occurs due to a synergy of faults at various links of the network.

A. TCP re-transmitting masks the source of the problem

There is a lack of transparency to users of network flaws due to the TCP adaptive re-transmission algorithm retransmitting lost packets. Longer communication is the only obvious symptom to the user. This property of TCP/IP makes it difficult to tackle the source of network problems.

For example, a decrease in download speed of a certain application running on an Internet user's host, or more specifically, when an user is watching video from Netflix, there can be a number of reasons why the quality is switched from HD to SD. It may be due to that host reaching its bandwidth capacity by running multiple bandwidth hungry applications simultaneously, or that a burst of traffic provokes congestion on the access and/or the backbone networks at that period of time, or that routing errors that need time to converge are triggered by faulty routing configuration, BGP routing attacks or anomalous Autonomous Domains (ASes) changes, or that the server is just reaching its capacity limits.

The task of tackling network performance faults is extremely difficult in the first place, and SAM method and reporting results from SAM only makes the problem worse. SAM may cause a great deal of valuable information to be lost,

inluding the most significant information that the network flaw (bad performance) ever happened, as when averaging "good" and "bad" measurements, you end up with something in middle that's acceptable. Consider the following example: for one household, at a given period of time, the user experiences a dramatic decrease in download speed, but this household has a reasonably high bandwidth capacity in general. If we were to aggregate speed data by time, we would lose the essential detail at that given period of time, network faults occur and did impact greatly on the user's experience.

If we were to further compare this household with another household which has a low bandwidth but never experienced such notable degradation of performance in the same period. The average number would suggest the network is constantly performing better for the former household than the later household, which is the opposite of the experience in this scenario.

B. RTT masks the scale of the problem

Another property of the TCP protocol is that its compensation cost (re-transmission time) is proportional to the round trip time (RTT) of a path. Consequently, if a problem occurs somewhere in the network, it may be noticeble on a large RTT path, but not noticed on a small enough RTT path. Consider the case of two hosts retrieving video content at the same time from a server that is performing poorly. Host A has a much larger RTT (accessing the server via a longer path) than host B (located closer to the server, or near the content cache). The host A would experience bad quality while host B would still obtain an acceptable quality of service. The SAM would suggest an issue with host A's network rather than the server.

The ability to cover network faults on a shorter RTT path makes the true network performance even more vague using SAM.

C. Why isn't averaging representative?

Average are not good enough in many scenarios. Averaging doesn't deal well with largely varying samples. Consider the four data sets shown in the Figure 3. They have exactly the same average, however they have very different properties statistically. The first data set is more similar to the third one both of which seem linearly distributed, while a polynomial model fits the second and the last one is the most skewed and represents a fixed value. If people were to use average representing those four data sets, consider how misleading it would be.

Another example that shows issues with applying SAM on network measurement. Let's say, we have two customers one has a bandwidth capacity of 8Mbps while another has 2Mbps from the same ISP, applying SAM on them involves creating data for a non-existing client with 5Mbps bandwidth, and the SAM report including trend, percentage and histogram plots are based on this non-existing client. This reporting is not representative.

However, data aggregation has been widely employed in network and IT data, because computing and storage resources

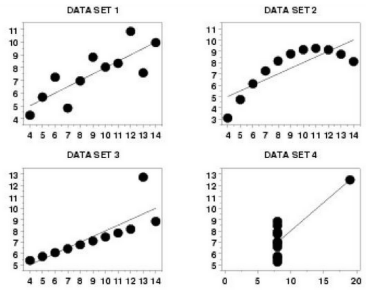

Fig. 3. Four data sets having the same average

were limited in the past. Averaging, among other possible aggregating methods, is the most prevalent technique used by tool vendors. Data aggregation is often employed by 1)keeping raw data for 24 hours, 2)aggregating once per hour for data less than 7 days old, and aggregating once per 24 hours for data greater than 7 days old. According to [6], quantifying Information loss through data aggregation has showed that usage of aggregated data, particularly those data sets with a greater than 2.5 hour aggregation for capacity analysis will have lost a significant amount of its original behavior. Quantitatively 50 percent of IT type metrics lose their distribution coherence (relative to the raw data) after 2.5 hours of aggregation and 85 percent lose their distribution coherence after 12 hours of aggregation, as per [6].

In the light of the significant amount of information loss with a greater than 2.5 hour aggregation, and that average performs badly at skewed data, thus it is better to analyze the original measurement data without aggregation.

IV. Sampling and Measuring Methodology

In [7], the author examines a number of empirical strategies and show general principles for sound Internet Measurement:

1) Examining outliers (unusually low or high values) and spikes (values that repeat a great deal) represent *corner cases* at the extremes of measurement where problems often manifest. Meanwhile, these *corner cases* or anomalies are easy to spot without a great deal of effort.
2) Employing self-consistency checks, for example, plotting additional properties of the measured phenomenon to see if they agree or disagree with behaviour reflected in the initial measurement.
3) Measuring facets of the same phenomenon different ways and comparing them.
4) Ensuring reproducible analysis so that any human made mistakes and biases will be eliminated.

We aim to in line these strategies in our experimental work.

A. What to measure from M-Lab data?

In general, there are various aspects to measure: productivity, throughput, responsiveness, delay, round trip time, packet loss rate, etc. When it comes to traffic measurement, we have a variety traffic types. HTTP, Video Streaming, Voice/Audio Over IP, P2P, and old fashioned FTP and Telnet.

M-Lab provides a set of tools targeting different aspects and different traffic types. In our experimental work, we focus on the Network Diagnostic Tool (NDT) data for several reasons. First, NDT is a highly reliable test tool working in client/server mode that provides network performance testing by using the well-defined NDT Protocol. Second, NDT data contains rich meta information for each test, such as clients' public IP, servers' IP, test time and etc. Third, NDT data is well formatted and easily accessible via BigQuery, a SQL-like query, based on the processing power of Google's infrastructure.

The amount of data generated by this tool is huge. The data collected in March 2010 alone has almost 26 billions rows. The BigQuery tool facilitates analysis of large data sets on this scale. BigQuery is accessible by a web UI or a command-line tool or BigQuery REST API. We use BigQuery via web UI to analyze and retrieve NDT data. In the following section, we discuss how to sample relevant data sets theoretically and how to employ the BigQuery tool to analyse and retrieve large data sets practically.

B. How to sample data from NDT?

Vern Paxson [7] suggests to analyse large data sets by working initially on small subsets and assess variability across different subsets.

In reality, the entire population is generally unknown in many scenarios. In the case of network measurement, the entire population consists of test results from very single host in the network at every predefined measurement period. Although many m-Lab measurement tools, including NDT, are designed in a way that clients issue periodical tests automatically. The automatic scheduling of these tasks is unreliable. For example, the host machines may go off-line, some users may un-install the test client or disable periodical tests manually, or the client program may crush and not restart until next reboot of the machine. It is also not practical to enforce that every host install the test client. The previous points result in holes in the data. Some times test data may be skewed at some networks but not others. It's also likely that more samples cluster at a certain time instead of being evenly distributed across a time period.

Ideally, it's better to choose houses that have evenly distributed test data to statistically trust the samples, because It requires a sufficient sample size to estimate the population's property. Larger sample sizes generally lead to increased precision when estimating unknown data sets. One way of sampling among others is called expedience which means to include those items readily available or convenient to collect so that sufficient sample size criteria can be satisfied. Therefore, it is natural to sample from hosts that have done tests most frequently, so that we can have more evenly distributed test data across hours, days and months for comparing in different facets and analysis.

1) Identify hosts having done tests most frequently: BigQuery contains M-Lab logs generated since January 2009 by three M-Lab tools. BigQuery tables are updated every day with data from M-Lab logs collected the day before, as per BigQuery. We use the BigQuery sentence via Web UI to find the hosts that are geographically located in Australia and have done tests most frequently from February to June 2015, for the purpose of studying Australia's Broadband performance. The query returns a number of rows each of which contains a pair of client IP and the number of tests associated with that IP address. The returned rows are sorted in descending order by number of tests per query. The result of this query is shown in Table I.

TABLE I
TOP 5 FREQUENT HOSTS

Row	household IP	Number of Tests
1	60.242.16.174	1177
2	220.244.174.201	753
3	116.240.255.19	691
4*	220.253.71.79	444
5*	14.201.217.164	164

When we further investigate the hosts on the list, we find some of them only ran tests intensively within a month and have zero records in other months, such as the top house associated with IP: 60.242.16.174. We ruled out houses such as this one, so we can have data even distributed across months. That way, we can compare and infer network phenomena across months. The two houses (row 4 and row 5 with star marks) are the two from which we retrieve raw measurement data.

Beside this selection criteria for the sake of a evenly distributed sample data, the choice of sampling is purely random and without any human interference to avoid introducing human bias in data. BigQuery query we used to find the most frequent houses can be found in the appendix.

2) Retrieving measurement data: The next step is using BigQuery retrieving March and April 2015's measurement data for two households sampled from previous subsection *House 1* associated with IP address *220.253.71.79* and *House 2* associated with IP address *14.201.217.164*. The query via BigQuery UI returns a number of records. Each record reflects a test data that consist of year, month, day, hour and downloading speed. BigQuery query we used to retrieve that measurement data can be found at the appendix.

V. EXPERIMENTAL RESULTS: VISUALIZING DATA WITHOUT AGGREGATING

Taking into consideration of the measuring methodology stated in the previous section, we visualise the raw performance data in different ways without aggregating in order to facilitate spotting outliers and spikes.

For *House 1* and *House 2*, we plot every single test's speed value against each day, arranging them in two facets representing two months (March and April 2015), as shown in Figure 4 and Figure 5, respectively. To make the visualization detectable, we categorize each speed data into one of the four speed categories: "Low", "Medium", "High" and "Very High" and assign four different colors: "red", "green", "blue" and

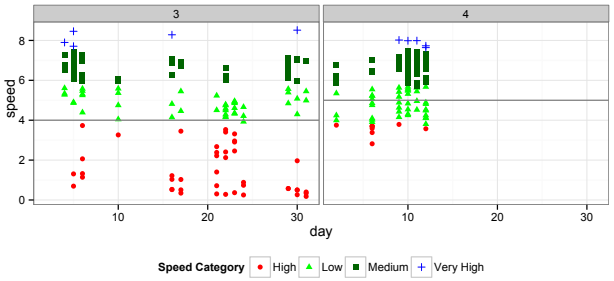

Fig. 4. *House 1*'s performance in March and April and its monthly Averages

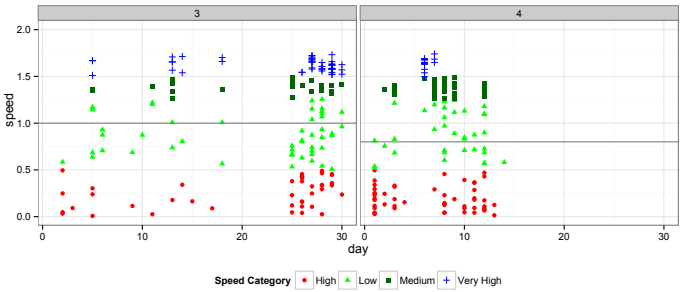

Fig. 5. *House 2*' s performance in March and April and its monthly Averages

"cyan" to each of the categories respectively. Hence, in all the plots we are able to distinguish speed categories by color and potentially catch changes in categories across time. For instance, the number of speeds in "Low" category increases or decreases from one month (day or hour) to the next. By comparing Figure 4 and Figure 5, we can see both houses have a lot of variation in speeds, but the variation is different for two houses. From March to April, *House 1* has considerably more "Very High" and "Low" speeds than *House 2* which means it is different to *House 1* in that it has more outliers than *House 2*.

The horizontal lines represent the average speeds of the month in each facet in both Figure 4 and Figure 5. In Figure 4, for *House 1* the SAM (average lines in this plot) suggests the performance of March is worse than April, as the average is lower for March. In the same figure, we can also infer that the average speed increased from March to April because a lot less data points fall into the "Low" category in April than in March. However we can not conclude that user's quality of service is noticeably increased from March to April.

To achieve better accuracy, we add more resolution to the facets and x axis. We use two methods to include all month, day and hour information in one plot.

(A) plot each facet as a pair of unique month and hour combination against day (x axis).

(B) plot each facet as a pair of unique month and day combination against hour (x axis)

We apply both methods on both *House 1* and *House 2*, to see within a month if the outliers and spikes are more consistent in particular hours or particular days of that month.

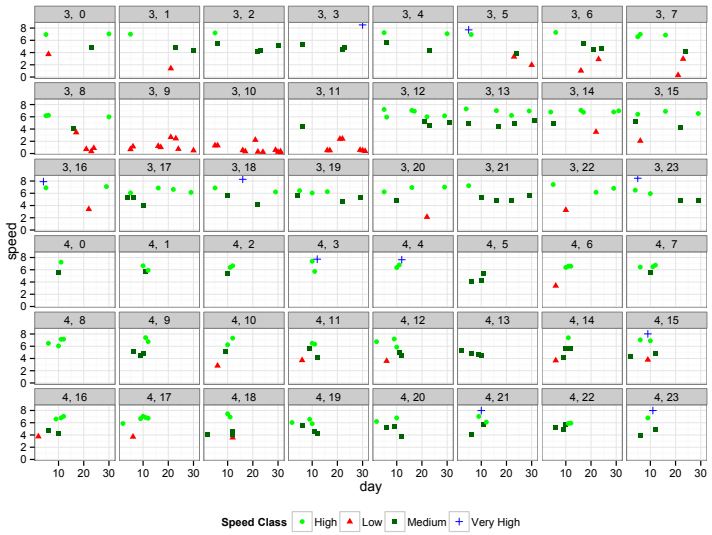

Fig. 6. *House 1*'s performance in March and April faceted by (Month, Hour)

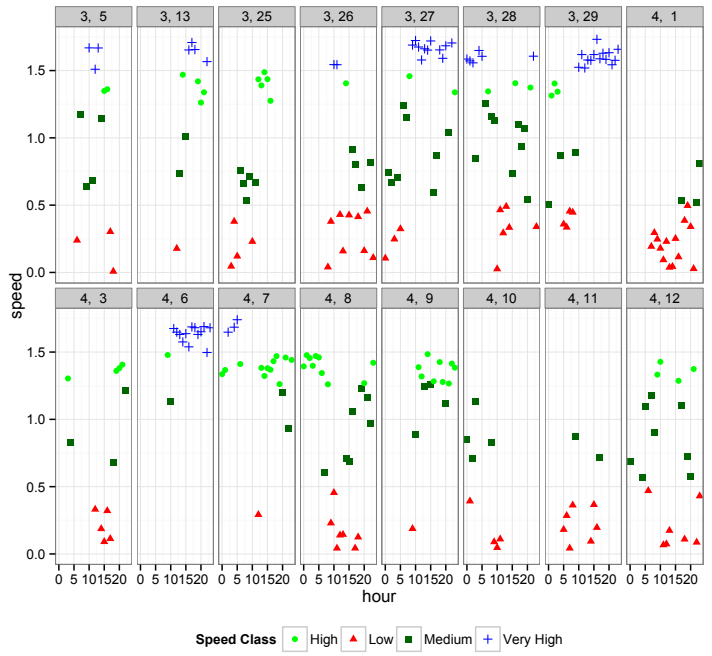

Fig. 7. *House 2*'s performance in March and April faceted by (Month, Day)

As shown in the Figure 6, for *House 1*, the variation within most hours in both March and April seem negligible which suggests consistent performance across hours in both months. However, a significant number of "Low" samples repetitively appear at 9 am, 10 am and 11 am in March. Together with these two clues, this plot is revealing the fact that March's performance looks worse because in a number of days at the same hours: 9am, 10am, 11am, the performance was bad. The "Low" in these three hours appears to be low values and occurring repetitively across a number of days, which suggest

they are the outliers and spikes that we aim to find. Hence, unexpectedly, we observe low outliers and spikes during non peak hours, which caused March's low average performance. Since the *House 1* may not even be using the Internet during these three hours (9am - 11am), this does not agree with "bad performance" for March as suggested by the SAM method.

We plot the un-aggregated speed data from *House 2* by method B and show the result in Figure 7. We find that March's average looks better than April is because across days there are seven days in March that show a significant amount of "Very high" speed in Figure 7, while there are only two days we can observe "Very high" in April. Unlike *House 1*, these outliers are clustered by day instead of by hour. Apparently these "Very High" outliers are the reason why March outperforms April in average. However, in the mean time, there are also noticeably a lot more "Low" samples in March than in April, which disagrees that March's performance is simply better than April.

For the purpose of reproducible analysis of the same and different data sets, the R code of how we employ a package called "ggplot2" to plot all the figures illustrated in this section is shown in the appendix.

VI. Conclusion

Our results show considerable variation in broadband speeds between households within a day and across days. We conclude that aggregating performance data by household is not an accurate indication of quality of service, since by aggregating two households, we will have lost both information that each household has a lot of variation and that the variation pattern for each household is different.

In addition, as shown in our results, unexpected patterns occur even within a house, such as wide variations from one day to the next, or showing clustered outliers in certain hours of the day. We also conclude that aggregating by day or by hour is not a good measurement method, because by aggregating hours and/or days for one household, we will have lost the information for which hours and/or days are problematic within a house.

References

[1] John W. Dower *2014 global Internet phenomena report* 1991.

[2] E. H. Norman *Cisco Visual Networking Index: Forecast and Methodology, 20132018* 1940: International Secretariat, Institute of Pacific Relations.

[3] Bob Tadashi Wakabayashi *Glasnost: Enabling End Users to Detect Traffic Differentiation* Proceedings of the 7th USENIX conference on Networked systems design and implementation Pages 27-27,2010, USENIX Association Berkeley, CA, USA

[4] Partha Kanuparthy *DiffProbe- Detecting ISP Service Discrimination* , INFOCOM'10 Proceedings of the 29th conference on Information communications, 2010, NJ, USA,

[5] Partha Kanuparthy *ShaperProbe- End-to-end Detection of ISP Traffic Shaping using Active Methods* Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference, New York, NY, USA

[6] Mazda A. Marvasti *Quantifying Information Loss Through Data Aggregation*, 2011, VMware, Inc. 3.

[7] Vern Paxson *Strategies for Sound Internet Measuremen*, October, 2004, IMC04, Taormina, Sicily, Italy.

Appendix

Query the most Frequent hosts

```
SELECT  remote_ip,
COUNT(*) AS num_tests
FROM (
SELECT  test_id,
web100_log_entry.connection_spec.remote_ip
AS remote_ip
FROM
[plx.google:m_lab.2015_02.all],
 <specify other month here>
WHERE
connection_spec.client_geolocation.country_code
== "AU"
   ..
AND <specify other ceritiers  that validate the test>
   ..
GROUP BY  test_id,  remote_ip )
GROUP BY remote_ip  ORDER BY
num_tests DESC;
```

Retrieving measurement data of House 1

```
SELECT
  YEAR(FORMAT_UTC_USEC(web100_log_entry.log_time*1000000)),
  MONTH(FORMAT_UTC_USEC(web100_log_entry.log_time*1000000)),
  DAY(FORMAT_UTC_USEC(web100_log_entry.log_time*1000000)),
  HOUR(FORMAT_UTC_USEC(web100_log_entry.log_time*1000000)),
  web100_log_entry.connection_spec.remote_ip,
  web100_log_entry.connection_spec.local_ip,
  8*web100_log_entry.snap.HCThruOctetsAcked/(web100_log_entry.snap.SndLimTimeRwin
   +web100_log_entry.snap.SndLimTimeCwnd +web100_log_entry.snap.SndLimTimeSnd)
FROM
  [plx.google:m_lab.2015_03.all], [plx.google:m_lab.2015_04.all]
WHERE
   web100_log_entry.connection_spec.remote_ip =  220.253.71.79 "
___AND_<other_conditions_that_a_valid_test_shall_conform_with>
_
```

Plotting the month charts

```
#Visualizing example code
#User defined colours for four catagries
#of Class
cc = read.csv("C1-Speed-March-April.csv")
cc$color[cc$range=='LOW'] = "red"
cc$color[cc$range=='MEDIUM'] = "green"
cc$color[cc$range=='HIGH'] = "blue"
cc$color[cc$range=='VERY_HIGH'] = "cyan"
col.list <- c("red","green","blue","cyan")
palette(col.list)

ave = data.frame(AVE = c(AVE1,AVE2),
month= c(3,4))

ave$month = factor(ave$month)

#Plot two months speed samples in two
#facets, together with average line of
#each month
a = ggplot(cc, aes(x=day, y = speed,
color = color, shape = color))
+ geom_point()
+ scale_fill_discrete(name="Speed_Class",
labels=c("Low", "Medium", "High",
"Very_High"))
+ xlab("day") +facet_wrap(~month)
+ theme_bw()
+ scale_color_manual
(values=c("red", "green",
"darkgreen", "blue"),
name="Speed_Category",
labels=c("High", "Low", "Medium",
"Very_High"))
+ scale_shape_discrete
(name="Speed_Category",
labels=c("High", "Low", "Medium",
"Very_High"))+geom_hline
(data = ave, aes(yintercept=AVE),
colour="grey50")
+theme(legend.position="bottom"))
```