Preliminary Longitudinal Study of Internet Responsiveness

Matt Mathis, Aug 2021

This is a very preliminary longitudinal study of Internet responsiveness based on data collected by Measurement Lab (MLab). It is quite limited in a number of ways, none-the-less it shows improvement of Internet Responsiveness over several years. Part of the motivation for this work is to test some prototype algorithms to measure responsiveness, and explore some of the issues that might need to be addressed in a well specified Internet responsiveness metric.

This study is based on a speculative approximation of a future formal "Responsiveness Under Working Conditions" metric, patterned after a measurement tool that Apple is already shipping with their developer tools for iPhone and iOS devices.[Apple] Although there is a partial draft specification [Draft] all implementations long precede any formal specification and validation.

Responsiveness is tentatively defined as the rate for round trips in the network, typically as Rounds per Minute (RPM). Since queues in the network are known to be a major contributor to poor responsiveness [BufferBloat], the metric of interest is actually "Responsiveness Under Working Conditions" which requires an authentic load on the measurement stream, and potentially additional network cross traffic to expose the behavior of the queue management at network bottlenecks.

It is expected that we will ultimately define parallel responsiveness metrics at multiple levels in the stack. For this paper we estimate counting transport layer (TCP) rounds: data segments, trigger ACKs which in turn trigger additional segments. The Apple tool measures application rounds, by using application layer ping messages to measure the Round Trip Time of application to application messages. In addition to network delays, application messages must traverse all of the queues and buffering in the operating system and the application itself.

The metric presented in this note is computed from the final value of the Smoothed RTT estimator maintained by TCP. The final value, at the end of a test, is not an ideal method for estimating responsiveness over the lifetime of the connection, but it is good enough to get some insight into both the evolution of the Internet and considerations for specifying future formal responsiveness metrics.

The smoothed RTT estimator was introduced by Van Jacobson in 1988, and has since been standardized[SRTT]. One of the strengths of using a metric based on SRTT is that it has changed relatively little over the 30+ years since it was first developed, making it more likely that metrics derived from SRTT values archived on different vintage systems can be compared directly.

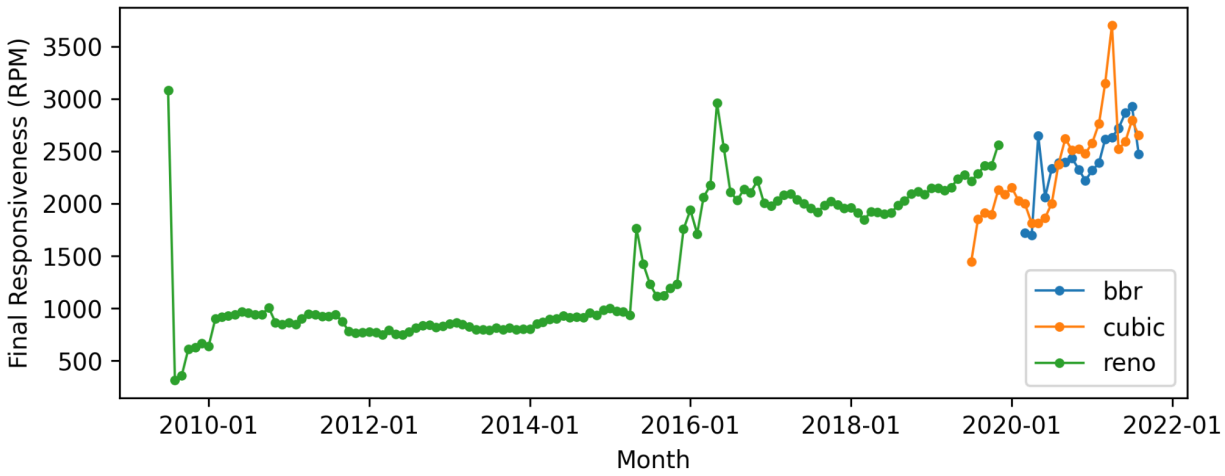
The MLab archive includes measurements from more than 4 billion download tests. Every one of them includes periodic TCP state snapshots via Web100 or TCPinfo. All of the snapshots are preserved in the raw data in Google Cloud Storage (GCS) (generally at 10 mS intervals). However this data was downsampled as it was parsed into BigQuery. TCPinfo data from the new platform was thinned to 100 mS samples, but Web100 data only presents the final snapshot in BigQuery. We have the capability to reprocess all of the raw data, for example to make different decisions about downsampling, or even to compute a responsiveness metric in the parser, before the data is down sampled at all.

The plots below are very preliminary. The data was not filtered or groomed in any way. Every NDT connection that reached the TCP established state is included in the plot, even if it is otherwise not a meaningful measurement. Appendix A has instructions for reproducing these plots.

Observations

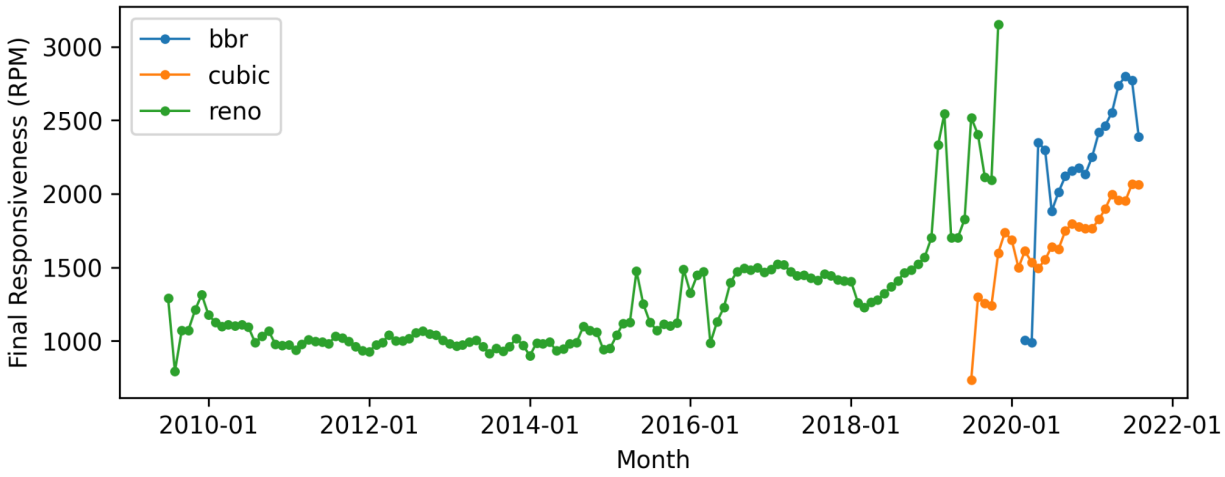
For figure 1-3 there are discontinuities in the data that correspond to MLab measurement platform upgrades. We partitioned the data by TCP Congestion Control and implicitly platform version, because we expect CC to affect network queue occupancy and thus responsiveness. The transition from Reno to Cubic reflects a fleet-wide server upgrade that was rolled out over approximately half a year. Since CUBIC maximizes throughput by maintaining larger network queues, it causes substantially lower responsiveness.

Figure 1: Mean Responsiveness circa NYC



The still progressing transition from CUBIC to BBR reflects a much more gradual client transition to a complete reimplementation of NDT, which uses the BBR congestion control algorithem. BBR is intended to more accurately model the network capacity. In the NYC data, they don't appear to be materially different, however for the entire US (Figure 2) they are clearly different - BBR maintains smaller queues and better responsiveness. This is what we would expect, based on BBR's design.

Figure 2: Mean US Responsiveness



Note that tool discontinuities are a potential problem for any long duration longitudinal data set. An ideal, state-of-the-art measurement tool in 2009 is unlikely to be good enough (and certainly not state-of-the-art) by 2021.

Since this data has not been filtered or groomed in any way it contains measurements that should properly be discarded: connections that did not send sufficient data (not "working conditions") tests to non-local servers (e.g. global VPNs) and even MLab's own OAM traffic.

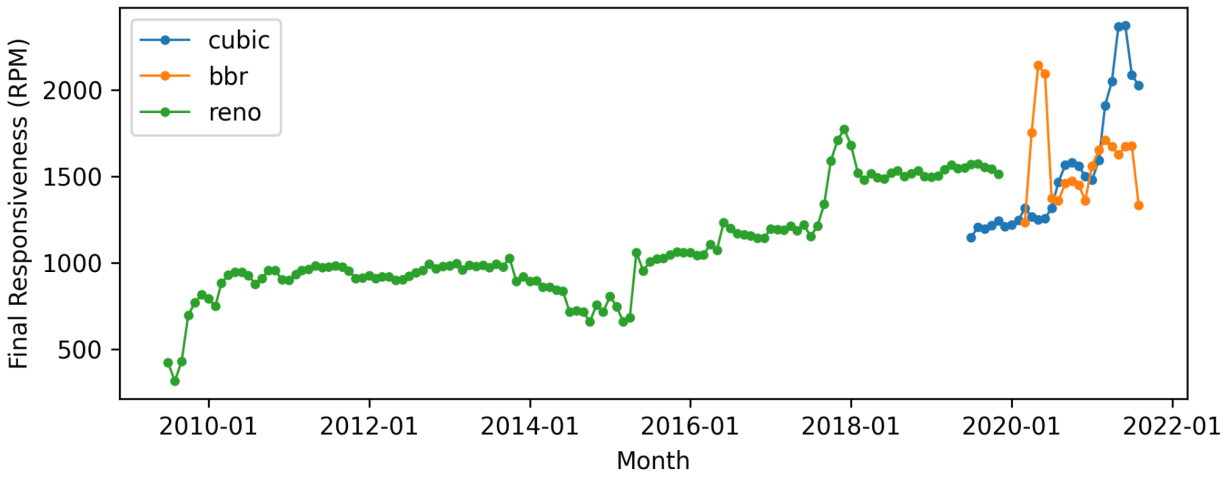
The current MLab Extended Views [Extended] do not support filters that would be appropriate for grooming the data for this experiment. We will update this report once MLab has a more general filter framework for grooming the data.

Even without grooming, adjacent months typically exhibit very similar responsiveness, suggesting that the measurements are quite stable and do not have much noise. Nearly all outliers involve more than one month, suggesting that they are caused by real changes in the network or environment. Although most of the outliers are parallel between NYC and all of the US, there is one substantial event circa April of 2016 where NYC and all US move in opposite directions -- we have not explored this event.

We selected NYC because there is a huge population in a relatively small geographical area. Most MLab test traffic goes to the closest server. The vernoi map for NYC is bound by Washington DC, Tronto and Montreal, which makes it one of the geographically smallest regions in the entire fleet. This was intended to minimize the extent to which the client geographical distribution blurs other effects such as changes to queue management.

It was very unexpected that the NYC and all US data are so similar. This is probably an indication that the vast majority of tests are from metropolitan areas, each of which has their own servers, and people in the huge geographical expenses of the American west don't run many tests.

Figure 3: Mean German Responsiveness

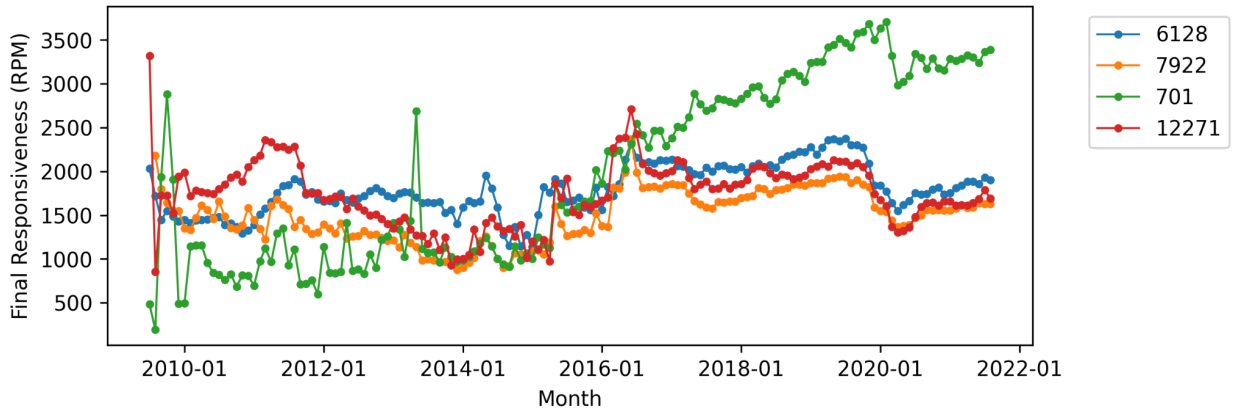


In Figure 3 we show the mean final responsiveness for Germany. We chose this example because the server locations in and around Germany have been stable since the beginning of Measurement Lab, so we expect this data to reflect progresive improvement in traffic management and perhaps topology. This is not true for all of Europe. MLab implemented a substantial buildout in 2019, which added many new server locations. In some countries most users will select newer, closer, servers, which will intrinsically be more responsive.

We have not looked at the Southern Hemisphere data at all, because MLab's footprint was initially too sparse and has changed too much over the years to easily make meaningful comparisons. The changing server locations cause radically different geographical path lengths, making it difficult to understand changes to the network.

Note that a future formal responsiveness metric will need to directly address issues of CDN locations, peering and topology, especially since some of the stated queueing targets are well below the speed of light delays over many paths.

Figure 4: Responsiveness of the 4 ISPs with the largest test volumes circa LGA



In Figure 4 we present the mean final responsiveness for the 4 largest ISPs in the NYC area. This is the same data as Figure 1, except sliced by client ISP, instead of server Congestion control. The transition between the Web100 (Reno) and TCP_info (CUBIC and BBR), is clearly apparent across the beginning of 2020, with declining responsiveness likely due to more aggressive congestion control causing higher buffer occupancy.

The differences between the ISPs reflect differences in technology, and possibly other network properties such as topolog and peering. One would guess from this data that one ISP is investing in responsiveness (or probably some related property such as Video Conference quality) and the other 3, not so much.

At some level, we expect this. Some ISPs are paying attention to the consequences of excess buffers and Bufferbloat, and others are not.

Appendix A describes how to reconstruct these graphs, for your locale.

Future Work

As mentioned earlier, the raw MLab data archived in GCS includes periodic TCP state snapshots for every NDT test. We have preliminary running code to estimate the actual number of Round Trips needed to complete NDT tests from the entire SRTT time series for each NDT test. We can do this now for tests run on the new platform and in the future for all of the historical data once we reparse all of the raw Web100 data to import periodic snapshots into BigQuery.

Today most web pages are assembled from many http objects, typically fetched over the network without experiencing any indications of congestion - no lost packets or ECN marks. As a consequence every TCP connection is still in slow start when it runs out of data. We posit that in such an environment, loading a web page requires the number of round trips determined by the critical path diagram for the entire web page (both within TCP connections and as embedded links invoke additional http fetches). The page load time should be predicted by the number of rounds needed to load the page, divided by the rate that the network delivers rounds, aka its responsiveness. We plan to confirm this conjecture.

Note that page load time depends on the average responsiveness: outliers (bad RTTs) contribute "linearly" to the total time, unlike jitter metrics where exceeding deadlines cause disproportionate degradation to the user experience.

MLab also has full TCP header captures from the majority of NDT tests. We hope to calibrate responsiveness metrics by reconstructing protocol critical path timing diagrams and counting round trips directly for some small number of tests. These timing diagrams will be close proxies for the actual ground truth of historical measurements.

In addition to validating historical tests, we plan to implement a version of the responsiveness metric that runs in real time on the NDT server itself, and reports the responsiveness alongside the speed measurement in the same test.

There is an important corner case that we need to consider: We are aware that there are a substantial number of clients that run loss-less because there are oversized buffers at the bottleneck [Bufferbloat]. These clients typically exhibit a large bottleneck queue that is still growing at the end of the test. (e.g. Typically a 5 second queue for a 10 second test). For these clients, the final SRTT (and for that matter any summary of the SRTT time series) is likely to depend on the test duration (and possibly the maximum value of the receiver window). In these situations the observed responsiveness fails to actually characterize the network. How should we properly include such paths in any aggregate responsiveness metric?

Conclusion

The goal of our work is to measure responsiveness for multiple audiences: a consumer grade tool to help ordinary people understand important differences between providers, and engineering grade tools to help the providers to build better networks. All without trying to explain bufferbloat.

We have a lot of work to do. As intended, this study asks far more new questions than it answers.

The good news is that in at least some parts of the internet are more responsive now than they were a few years ago.

References

[Apple] WWDC21 video "Reduce network delays for your app"
[Draft] NDT Extended Views Responsiveness under working conditions
[BufferBloat] J. Gettys Bufferbloat: Dark buffers in the internet
[Extended] www.measurementlab.net/tests/ndt/#extended-views
[SRTT] RFC 6298, V. Paxson, "Computing TCP's Retransmission Timer"

Appendix A

These graphs are very easy to adapt to other regions or to address slightly different questions. They are also very fast: once the BQ table has been cached, the edit/rerun cycle is under a minute. Visit the colab (resembles Jupyter) at bit.ly/2WYQVix or the full URL: https://colab.research.google.com/drive/13ohipjKoUxh2YfmxoOwXb5l3cAnn5m8d#scrollTo=zrLPeHQ56_ud.

There are full instructions at the top of the notebook. You will need to follow the first few steps of the MLab BigQuery QuickStart page . You need a Google domain account that is subscribed to discuss@measurementlab.net and enrolled in Google Cloud Console (Free access is sufficient, but even for free access you need to present a credit card)

All of the graphs in this paper can be reconstructed by simple edits to the colab.