Best Practices for Collecting Speed Test Data

Kyle MacMillan, Tarun Mangla, Marc Richardson, Nick Feamster
Department of Computer Science, University of Chicago, USA
{macmillan,tmangla,mtrichardson,feamster}@uchicago.edu

Abstract

In an effort to expand Internet access, local and federal policymakers have sought to use speed test data to determine where to allocate funding. However, drawing accurate and meaningful conclusions about Internet Service Provider (ISP) network performance from speed test data requires careful consideration of the test conditions under which the data was collected. This paper provides a set of recommendations for gathering and analysing a single speed test measurement. Our recommendations are based on our own analysis of speed test data collected from in-lab, controlled experiments, as well as from a six month long deployment across 77 households in Chicago. Based on the results of our analysis, we suggest that the client-server latency, the client device, and the access medium (wireless vs. wired) be logged for each test. These three metadata can be used later to assess the accuracy or utility of the speed test. We also propose that running "paired tests" (running two different speed tests in succession) can help mitigate the effects of test conditions for which we can't measure, such as server load or upstream congestion. Such paired tests are most beneficial for households subscribed to download speeds greater than 500 Mbps.

1 Introduction

Consumer speed test data has become an attractive resource for federal and local officials to evaluate the last-mile Internet Service Provider (ISP) performance. In particular, these policymakers have sought to use speed test data to identify neighborhoods and regions that qualify for funding from the Broadband Equity, Access, and Deployment (BEAD) program, which provides $42.5 billion to expand Internet access in underserved areas. The National Telecommunications and Information Administration (NTIA) [21] maintains a National Broadband Availability Map to identify underserved neighborhoods and uses speed test data among other sources. In addition, state and local officials across the country are urging consumers to participate in speed test crowd-sourcing initiatives to help to further identify which areas meet the federal funding criteria.

However, there are multiple challenge to evaluating Internet speed using crowdsourced data. First, there are no standardized protocols to measure Internet speed. As a result, there exist multiple tools that each have there own testing protocol and produce different results under similar test conditions. Second, there is no guarantee that a speed test will measure the ISP access link. Speed tests measure the bottleneck link, wherever along the end-to-end path (see Figure 1) that link may be. The bottleneck can shift to the WiFi network inside the user's home but can also appear outside the ISP altogether, occurring at an interconnect or closer to the test server. Thus, a test result indicating poor performance may not be attributable to the ISP.

Given these challenges, drawing accurate conclusions from speed test data is a path fraught with pitfalls. Nevertheless, speed test data remains an invaluable tool in measuring Internet performance and will continue to be used in high stakes decision making. Therefore, we present a set of recommendations for both gathering and analyzing individual speed test results. In particular, we discuss factors that should be controlled for at *test-time* (when the test is conducted) or *post-hoc* during data analysis[1].

Our recommendations are based on speed test measurement data collected from both in-lab experiments conducted under controlled network conditions and a six-month long deployment across 77 households in Chicago. While our recommendations are applicable across speed test tools, they are drawn from analyses of two popular speed test tools, Ookla Speedtest (Ookla) [26] and Measurement Lab's (M-Lab's) Network Diagnostic Test version 7 (NDT7) [18]. We chose to analyze Ookla and NDT7 because they are used extensively—Ookla and NDT7 report a daily average of over 10 million [25] and 6 million tests [16], respectively. As a result, Ookla and M-Lab have collectively amassed billions of speed tests [6, 25] and have compiled data sets that have become universal resources for analyzing broadband Internet performance [3, 8, 22, 30].

[1] There are other important issues including metrics and methods used to aggregate data from multiple measurements which we relegate to future work

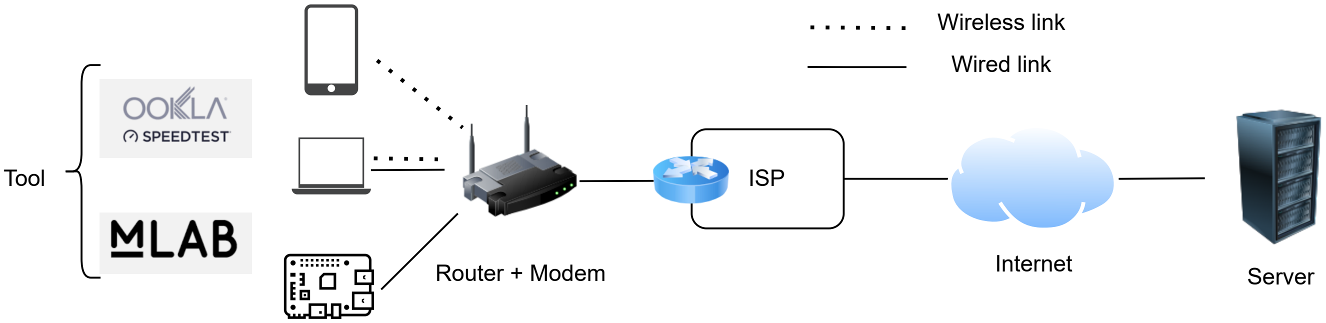


Figure 1: *End-to-end path for a speed test.*

Our key recommendations are summarized below:

- Record the latency between the client and server at test time. High latency can prevent the tool from saturating the link, thus underestimating the speed. This can occur above 100 ms and 400 ms for NDT7 and Ookla, respectively.
- Record whether the client connected via WiFi or wired connection. For users subscribed to speeds above 50 Mbps, connecting via WiFi will often cause the tool to underestimate speed, almost always at speeds above 800 Mbps. However, the tool is less likely to underestimate for speeds < 50 Mbps.
- Record the client device. The client device's hardware may become the bottleneck at sufficiently high speeds (1Gbps).
- To mitigate the effects of non-ISP unobservable bottlenecks, such as server and congested transit ISPs, that appear especially in high-speed connections, consider running two different tools in succession and taking the max of the two results.

2 Related Work

Speed test design. There are two primary ways to measure throughput: (1) packet probing and (2) flooding. Most packet probing techniques send a series of packets and infer metrics like available bandwidth or link capacity based on the inter-arrival packet delay [9, 12, 14, 15, 27]. More recently, Ahmed et al. [1] estimate bandwidth bottlenecks by probing the network using recursive in-band packet trains. However, these techniques can be inaccurate especially for high speed networks due to their sensitivity to packet loss, queuing policy etc. As a result, most commercial speed tests, including ones offered by both ISPs [2, 7] and non-ISP entities [18, 23, 26], are flooding-based tools that work by saturating the bottleneck link through active measurements.

Evaluating speed tests. Feamster and Livingood [10] discuss considerations with using flooding-based tools to measure speed. They do not, however, conduct empirical experiments to characterize NDT7 and Ookla performance. Similarly, Bauer et al. [4] explain how differences in speed test design and execution contribute to differences in test results. Bauer et al.'s work differs from ours in several ways. First, both Ookla and NDT have seen major design changes in the 12 years since that study. Both tools have updated their flooding and sampling mechanisms, and NDT's latest version (NDT7) uses TCP BBR instead of TCP Reno. Second, they only analyze public NDT data and do not study both Ookla and NDT in controlled lab settings, nor did they conduct paired measurements in the wide area that allows direct comparison of Ookla and NDT, as we do. Complimentary to our comparative analysis is work by Clark et al. [6] that provides recommendations on how to use aggregated NDT data, including considering the self-selection bias and other end-user bottlenecks like slow WiFi and outdated modems.

Residential broadband. Goga et al. [11] evaluate the accuracy of various speed test tools in residential networks, yet tools have changed and speeds on residential networks have increased more than 20× since this study ten years ago. Sundaresan et al. [29] studied network access link performance in residential networks more than ten years ago. Whereas our work is more focused on characterizing speed test tools, this work examined network performance differences across ISPs, looking at latency, packet loss, and jitter in addition to throughput. Canadi et al. [5] use publicly available Ookla data to analyze broadband performance in 35 metropolitan regions. Finally, the Federal Communications Commission (FCC) conducts the Measuring Broadband America project (MBA) [20], an ongoing study of fixed broadband performance in the United States. The FCC uses SamKnows whiteboxes [28] to collect a suite of network QoS metrics, including throughput, latency, and packet loss. Because the MBA project maps broadband Internet performance across different ISPs, they use a single speed test—a proprietary test developed by SamKnows—and do not consider Ookla or NDT7.

3 Factors impacting speed test outcome

An Internet speed test measures the achievable throughput (or data transfer rate) of a network path between the client and the test server (see Figure 1). To do so, most existing speed test tools send as much traffic along the network path as possible, a technique typically referred to as "flooding". The sending rate is controlled by the Transmission Control Protocol's congestion control algorithm, which adapts the sending rate during the test based on its estimation of the available network bandwidth. The tool will then estimate the speed based on the throughput achieved during the test. There are a number of factors that can impact the final speed that the tool reports, including test protocol and elements on the network path, as explained below:

Client platform. The client platform, including the client's device and software, could impact the result of a speed test. In most cases, speed tests are initiated by consumers on devices such as laptops and smartphones. In some cases, speed tests can also be automatically conducted using wrapper software (e.g., Murakami [17], Netrics [13]) installed on dedicated hardware. As for test software, the measurements can be run using a browser or native application, depending on user preference, client device, and the speed test tool implementation.

Access medium. The client device connects to the home router via wireless or wired link. With that said, most devices typically connect via WiFi. The WiFi access medium, as we will show later, may introduce a new speed bottleneck into the network path. As a result, conducting a speed test over WiFi can produce lower results than if the test had been conducted using a wired connection.

Test server and End-to-end network path Before data transfer begins, the speed test tool chooses a test server from its network of test servers. Each tool's network of test servers is unique and is characterized by the tool's server inclusion policy. For instance, any network can operate an Ookla server, though each server needs to pass some quality checks and can be removed if it is empirically determined to under-report throughput. On the other hand, NDT7 servers are operated by a managed infrastructure, owned and operated by a single organization (Measurement Lab). Ookla servers are sometimes "on net" (within the same ISP as the client), although that is neither a requirement nor a guarantee on the other hand, because NDT7 servers are operated in data centers, they are typically "off net". Servers that are off net result in end-to-end paths that may traverse multiple networks, including transit networks and interconnection points that may introduce bottlenecks.

Test protocol. The test protocol is an important aspect of the test and consists of the following key functions:

- **Server selection.** Before beginning the test, the tool must determine the test server to be used. Existing tools typically choose a server based on the client-server latency and client geolocation. For example, Ookla begins by choosing the 10 closest servers to the client, where the client's location can be determined either by IP geolocation or GPS. Ookla then pings of each of these servers and picks the server with the lowest latency.
- **Flooding mechanism.** Having picked the server, the tool begins to transfer data. With the goal of measuring the maximum data transfer rate, the tool will attempt to saturate the bottleneck link. The tools can differ in their saturation mechanisms. We find that Ookla adapts both the number of open TCP connections and the test length in response to changes in the measured throughput over the course of the test; whereas NDT7 opens only a single TCP connection, and the test itself always runs for ten seconds. With that said, the latest versions of both Ookla and NDT7 use TCP websockets.
- **Sampling and aggregation.** Over the course of the test, the tool will periodically sample its sending rate. Once the data transfer is complete, the tool will aggregate this sample to determine the final reported speed. As a result, the chosen sampling and aggregation methods can have varying impact on the reported speed. While NDT7 reports the average throughput over the entire test (bytes transferred / test time), Ookla discards lower speed samples. This decision affects every test because of how the tools increase their sending rates. Each TCP connection begins with a period called "slow start", in which the client and server transfer data at a rate that is slower than the steady state transfer

rate. Futhermore, any transient decrease in sending rate that occurs during the test, triggered by, for example, short-term packet loss, will be captured by the speed reported by NDT7 but not by Ookla. Although the effects of sampling and aggregation difference are comparatively small, they nevertheless impact the reported speed.

Among these factors, the client platform and access medium are dependent on the end-user and their network setup. The choice of the speed test tool determines the testing protocol. The test server and the corresponding client-server network path is dependent on both the test tool and the client location. Given the number of non-ISP related factors can impact a speed test, a blind use of speed measurement data can lead to incorrect conclusions about ISP performance. For example, a low speed result that was caused by poor WiFi, could be incorrectly attributed to the ISP underperforming. Therefore, in this paper, we provide recommendations on how to best gather (and analyze) the speed test measurements to reduce the likelihood of a non-ISP factor impacting speed test outcomes.

4 Recommendations

In this section, we provide our key recommendations. Wherever needed, we support our measurements with data collected either from experiments conducted in-lab under controlled network conditions or from a six-month deployment gathering measurement data from 67 households in Chicago.

4.1 Log the client-server latency.

Recommendation: Record the latency between the client and server at test time. High latency can prevent the tool from saturating the link, thus underestimating the speed. This can occur above 100 ms and 400 ms for NDT7 and Ookla, respectively.

The measured speed is a function of the network conditions along the end-to-end path. Although many of these conditions, such as the load on each link, are unknown, there are conditions that are measurable and possible to control for. One such factor is the latency between the client and the server. Intuitively, higher latency can delay the feedback process in TCP's congestion control and thus it may take longer for the test to saturate the link. To understand the threshold at which the latency begins to impact the speed test, we conduct a set of controlled experiments to precisely control the network conditions along the network path. To do so, we host both the test client and test server in-lab, allowing us to fix the link capacity while varying the latency between the client and server. We then conduct speed test measurements and log their accuracy. Here, accuracy is defined as the ratio of reported speed metric and the link capacity. The accuracy is always between 0 and 1; value of 1 indicates the tool reported the capacity accurately and lower values indicate under estimation of link capacity.

Figure 2 shows how the accuracy of the measured download speed decreases as the round-trip time between the client and the server increases. The "Method" indicates what the measured speed would be using different sampling and aggregation techniques. "Reported" indicates the speed reported by the tool, while *Avg* shows the speed as calculated by taking the total number of bytes transferred over the total transmission time. Unsurprisingly, the overall trend is that both Ookla and NDT7 report lower speeds as the latency increases. However, Ookla is not affected by the increase in latency until it exceeds 400 ms. On the other hand, NDT7's accuracy decreases to 90% of the link capacity when the RTT is 100 ms and further to 83% at 200 ms.

The differences between the tools can be explained by the differences in test protocol and TCP's congestion control. Under high latency, TCP takes longer time to saturate the link due to delayed feedback. Ookla is able to mitigate latency effect to a large extent as it uses an adaptive test length. It runs the test for longer duration when the variance in measured throughput is high and thus can saturate the link even under high latency. Moreover, its sampling strategy discards the TCP slow start phase. NDT7, on the other hand, uses a fixed test length and reports average throughput across the entire duration of the test. This leads to Ookla reporting more accurate throughput values than NDT7.

Nevertheless, the latency effects are potentially significant because they imply that if a client test selects a path to a server with high latency, then the reported throughput could be significantly lower. We argue that it would be prudent to discard any speed tests for which the client-server round-trip time exceeds 400 ms for Ookla tests and 100 ms for NDT7 tests. Tests that are conducted when the latency exceeds these thresholds finish before the sending rate has increased enough to saturate the bottleneck link, leading to the lower than expected speed. It could also happen that a significant number of tests from a specific region exceed the latency threshold. We argue that such data is indicative of either poor test server infrastructure or overall network infrastructure in the area. However, the solution in both cases is not to increase the ISP access speeds, but to improve the server footprint in the former case and improve the network infrastructure in the latter case by increased peering and improving the middle-mile connectivity.

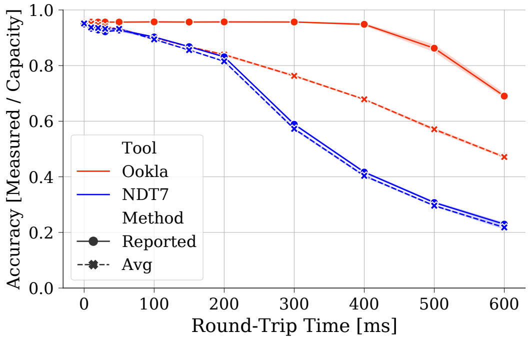


Figure 2: *Download accuracy vs. round-trip latency. Shaded region represents a 95% confidence interval for* $n = 10$ *tests. The reported method shows the speeds reported by the tool. The average method is the average data transfer rate during the test.*

4.2 Do NOT use WiFi for high-speed links

Recommendation: Record whether the client connected via WiFi or wired connection. For users subscribed to speeds above 50 Mbps, connecting via WiFi will often cause the tool to underestimate speed, almost always at speeds above 800 Mbps. However, the tool is less likely to underestimate for speeds < 50 Mbps.

When conducting a speed test, users access the Internet via a wireless (e.g. WiFi) or wired (e.g. Ethernet) connection, the choice of which, may skew the measurement. Because wireless transmission is more susceptible to interference and signal attenuation, wireless connections can have lower throughput limits than wired connections. As a result, running a speed test over WiFi may shift the bottleneck link from the access link to the wireless connection. Ideally this shift is inhibited by exclusively conducting speed test over a wired connection. However, encouraging a wired-only approach across all speed tiers may be excessive.

As part of our deployment, we conducted an experiment to determine how often and under what speeds does the wireless hop becomes the bottleneck during a speed test in practice. We placed Raspberry Pis (RPi) in 67 households across Chicago from November 2021 to April 2022[2]. Each RPi is connected to the home network via wired connection and conducts daily NDT7 speed tests. In addition, study participants installed a browser plugin on their laptop that would conduct an NDT7 test between a user's laptop and the RPi. This test indicates the maximum achievable throughput over the wireless connection at that point in time[3]. Note, the plugin runs speed test only when there is no browser network activity to avoid ruining the user experience.

Speed Tier (Mbps)	Number of Households	Mean Frequency of Bottlenecks
<50	2	0.01
50 - 100	11	0.16
100 - 200	3	0.41
200 - 400	6	0.72
400 - 800	9	0.84
>800	6	0.98

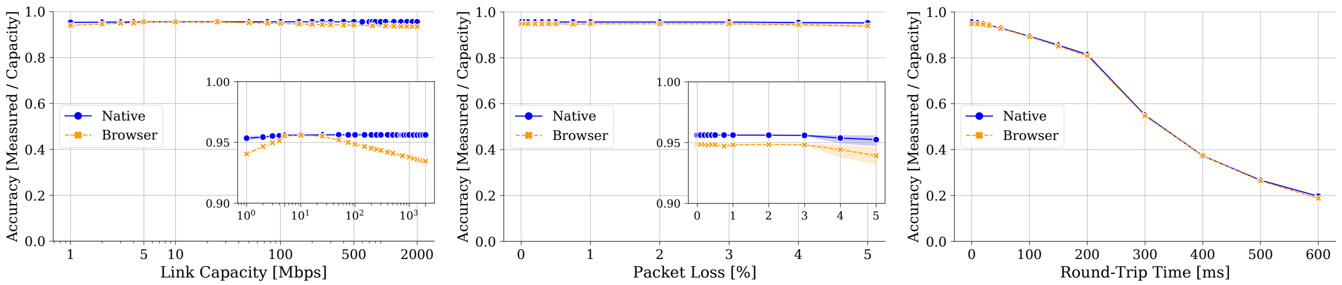
Table 1: *Average frequency of WiFi bottlenecks by speed tier. There are at least 30 tests for each household.*

We then compare the result of this (WiFi) speed test with a speed test conducted between the RPi and an M-Lab hosted NDT7 server that occurred within 6 hours of the laptop to device test. Although ideally these two tests would be conducted in succession, our experimental setup did not easily facilitate this approach. In comparing these two results, we posit that if the WiFi test result is lower than the wired test result, then the WiFi link would have been the bottleneck had the user conducted a speed test between their laptop and the M-Lab NDT7 server. This framework allows us to determine how often the WiFi link is the bottleneck. We further consider the effect of WiFi by grouping the devices into different speed tiers. We determine the speed tier based on the results of speed tests conducted by that participant over the course of the study.

Table 1 shows the frequency of speed tests for which the WiFi is the bottleneck, as determined by our experiment setup. We only include households for which there are more than 30 pairs of WiFi and wired tests that fulfill our requirement that they occurred within 6 hours of each other. Our results suggest that the wireless hop becomes the bottleneck in the vast majority of speed tests for households subscribed to speeds greater than 200Mbps. For households subscribed to plans above 800Mbps, the WiFi will almost always be the bottleneck link that is measured. Conversely, households with speed plans lower than 100Mbps are far less likely to experience WiFi bottlenecks. Of particular interest is that the 2 households subscribed to speed tiers lower than 50Mbps experience a WiFi bottleneck in only 1% of tests. Although the number of households is too low to draw broad conclusions, this result suggests that, for users subscribed to these speeds,

[2] For more details about the deployment, please refer to our past work [19]

[3] We assume that the user laptop is connected to Internet through WiFi which is often the case these days.



(a) *Link Capacity*

(b) *Packet Loss*

(c) *Latency*

Figure 3: *NDT7 accuracy on different client types (browser vs. native) for upload tests.*

it is possible to measure the access link throughput despite the user connecting over WiFi.

4.3 Log the client hardware and software

Recommendation: Record the client device. The client device's hardware may become the bottleneck at sufficiently high speeds (1Gbps).

Speed bottlenecks are not limited to links along the end-to-end path. The endpoints of a connection, in our case the client and the server, can sometimes be the limiting factor. As such, it's important to know what device the client is using (e.g. laptop or phone), as well as whether they are using the native or browser test application.

The client device is only likely to become the bottleneck at high speeds. For example, the RPis used in our deployment have a network interface card (NIC) that supports up to 1 Gbps throughput. Thus, the RPi device becomes the bottleneck for households subscribed to plans exceeding 1 Gbps. The specific limits, however, can vary based on the device hardware, and hence it is important to log the client hardware.

Past work has found that the client type (browser vs native) may also impact the accuracy of the speed test [10]. However, we observe minimal effects in our in-lab measurements. We compare the accuracy of the NDT7 native client and NDT7 browser client under different network conditions, varying the link capacity, packet loss, and client to server latency. There is no readily available way to conduct the same comparison for Ookla because it is not open-source, and we cannot configure the test server in Ookla 's current browser client.

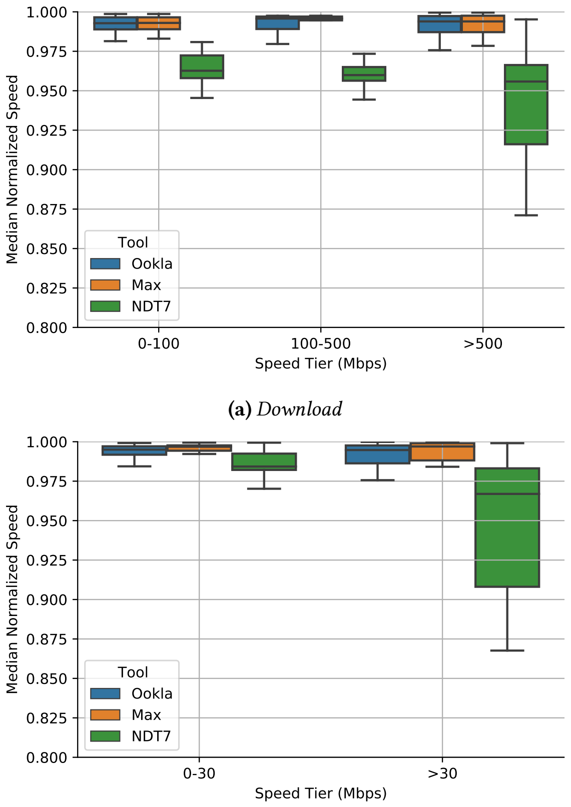
Figure 3 shows how the accuracy of the NDT7 browser and native client vary under different network conditions. We do not observe significant differences between tests conducted over browser and native client. As packet loss and latency between the client and server is induced, the difference in median accuracy between the browser and the native client is within 1%. Although there is a small dip in accuracy as the link capacity is increased, the difference is less than 2%. The difference in the results from past measurements might be due to upgrades in the browser and testing client. Nevertheless, browser-based clients may still become a bottleneck at higher speeds. For instance, Ookla suggests using their native clients instead of browser clients for testing connections with speed greater than 3 Gbps [24]. Thus, it is good practice to record the client application type used to conduct the test.

4.4 Run paired tests for higher confidence

Recommendation: To mitigate the effects of non-ISP unobservable bottlenecks, such as server and congested transit ISPs, that appear especially in high-speed connections, consider running two different tools in succession and taking the max of the two results.

While some bottlenecks are observable (or easy to infer), the others are challenging to locate or unobservable (e.g., test server, transit ISPs). To account for unobservable factors and increase the likelihood of measuring the ISP access link, we propose running paired tests, i.e., two different speed test tools in succession. The "paired test" idea relies on the path diversity achieved by using two different speed test tools, as different tools have different server infrastructure. Running the same tool twice does not yield the same advantage as the two successive runs could measure the same network path containing a non-ISP bottleneck link.

To make the idea more clear, we provide an example of running two successive speed tests using Tool A and Tool B. If Tool A and Tool B report similar throughput values, it is highly likely that they are measuring the same bottleneck link. Moreover, the bottleneck lies in the common network path between Tool A and Tool B. On the other hand, if Tool A reports a relatively low speed while Tool B reports a high



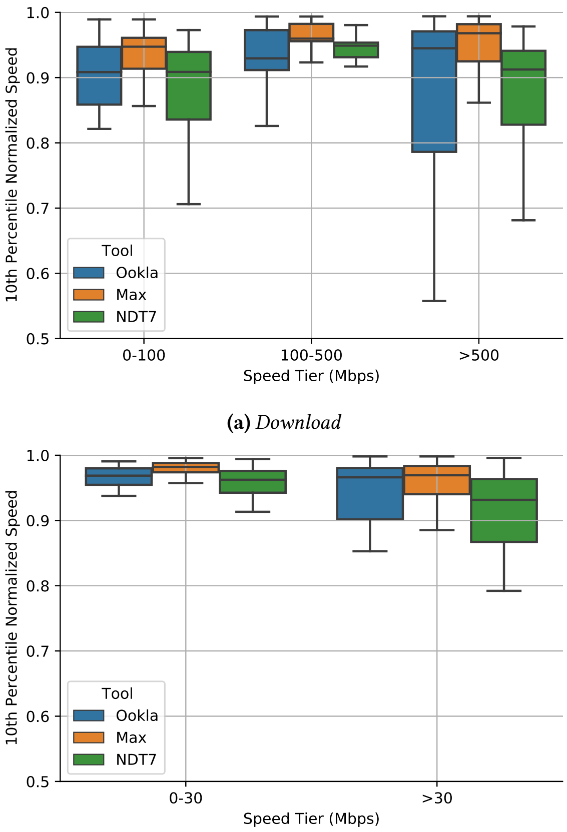
(a) *Download*

(b) *Upload*

Figure 4: *The distribution of median speeds across households from Ookla, NDT7, and paired tests by speed tier. Each point in the plot represents the median speed from a single household.*

speed, then the bottleneck link for tool A was likely upstream, as otherwise tool B would also have reported a low speed. Therefore, we propose running two successive tests using two different tools and take the maximum of the two reported speed values. The "paired tests" methods is not a full-proof method to measure the ISP access link but it only increase the confidence. It could very well happen that the both the tests measure a different upstream bottleneck.

There is also a trade-off in using paired tests as it may ignore transient congestion in the ISP link. More specifically, it could happen that the last-mile is the bottleneck for both the tests but due to dynamic link conditions the test reports two different throughput values. Paired tests, in this case, would ignore the lower throughput value. While this could be considered a limitation of using paired tests, we argue that a single measurement (paired or single) is anyways not enough to characterize a dynamic link. Ultimately, multiple



(a) *Download*

(b) *Upload*

Figure 5: *The distribution of 10th percentile speeds across households from Ookla, NDT7, and paired tests by speed tier. Each point in the plot represents the median speed form a single household.*

measurements are needed over time and paired tests can also detect persistent congestion events at the last-mile ISP link.

Advantage of using Paired Test: We now empirically analyze the advantage of using paired tests using the data from our wide-area deployment. We consider the speed test data collected from our deployment of RPis. Each RPi would conduct at least daily paired Ookla and NDT7 tests at random times of day. The RPi is connected directly to the router to avoid any WiFi bottlenecks. Over the course of the deployment, the median number of paired download test across households was 354, with a minimum of 50 and a maximum of 2,429. For each paired test, we compute the maximum of the two tests. We compare the distribution of individual tools with the maximum results of a paired test.

To understand the magnitude of the benefit, we first normalize each speed test result. Normalizing allows us to compare results across different households, among which there are different speed tiers. Ideally we could normalize against each result against the offered speed. However, because we do not have access to this information, we instead define the *nominal speed* for a given household to be the 90th percentile. Using the nominal speed, we compute the normalized speed for each test i as follows:

$$\hat{S}_i = \frac{S_i}{S_{90th}} \tag{1}$$

where S_i is the speed reported by test i and S_{90th} is the 90th percentile result across all speed tests from that particular household. In addition to using the nominal speed to normalize test results, we use it when assigning households to different speed tiers.

Having normalized each test, we study how the distribution of paired test results differs from the distribution of Ookla and NDT7 results. Although there are several ways to characterize the distribution, we focus on the average (median) and the tail (10th percentile) of the distribution. Using this framework, Figure 5 shows the 10th percentile speed and Figure 4 shows the median speed across households from the distributions of Ookla, NDT7, and paired tests.

Looking first at the median download results (Figure 4a, there is not a significant advantage to using paired tests as opposed to conducting only Ookla tests. Indeed, the median household speed using paired tests is within 1% of the median household speed using Ookla across all speed tiers. It is likely that congestion in an upstream link is a rare enough event that its occurrence is not reflected in the median speed. Instead, its impact is more apparent when looking at the tail of the distribution.

Looking at Figure 5a we see that the 10th percentile for paired tests is strictly greater than both NDT7 and Ookla. This indicates that neither tool always reports a higher speed. Moreover, we observe that the benefit of using paired tests is greatest for users subscribed to download speeds above 500Mbps. For these users, the median 10th percentile speed of paired tests is 2% and 5% higher than that of Ookla and NDT7, respectively. The difference is more striking for the 25th percentile household, in which the paired tests are 14% and 10% greater than that of Ookla and NDT7. As for upload results, the differences are less pronounced. This is unsurprising, as upload speeds are typically much lower than download, so there is less likely to be upstream congestion.

5 Conclusion

The paper presents a best practice guide to accurately measure ISP access speeds using consumer speed tests. We categorize the factors that can impact accuracy of tests as *observable* and *unobservable* factors. We quantify the impact of observable factors under different conditions using data collected from both in-lab experiments and wide-area deployment. For unobservable factors, we propose taking paired measurements as it can increase the confidence in a test measuring ISP access link. Future work will consider methods and metrics around aggregation of speed data across users and times.

References

[1] Adnan Ahmed, Ricky Mok, and Zubair Shafiq. "Flowtrace: A framework for active bandwidth measurements using in-band packet trains". In: *Passive and Active Network Measurement*. Springer. 2020.

[2] AT&T. *AT&T Internet Speed Test*. URL: https://www.att.com/support/speedtest/.

[3] New York State Office of the Attorney General. *New York Internet Health Test*. URL: https://ag.ny.gov/SpeedTest.

[4] Steven Bauer, David D Clark, and William Lehr. "Understanding broadband speed measurements". In: TPRC. 2010.

[5] Igor Canadi, Paul Barford, and Joel Sommers. "Revisiting broadband performance". In: *Proceedings of the 2012 Internet Measurement Conference*. ACM, 2012.

[6] David D. Clark and Sara Wedeman. *Measurement, Meaning and Purpose: Exploring the M-Lab NDT Dataset*. SSRN Scholarly Paper. Rochester, NY, Aug. 2021.

[7] Comcast. *Xfinity Speed Test*. URL: https://speedtest.xfinity.com/.

[8] Federal Communications Commission. *FTC Takes Action Against Frontier for Lying about Internet Speeds and Ripping Off Customers Who Paid High-Speed Prices for Slow Service*. May 2022. URL: https://www.ftc.gov/news-events/news/press-releases/2022/05/ftc-takes-action-against-frontier-lying-about-internet-speeds-ripping-customers-who-paid-high-speed.

[9] Constantinos Dovrolis, Parameswaran Ramanathan, and David Moore. "What do packet dispersion techniques measure?" In: *Proceedings IEEE INFOCOM*. 2001.

[10] Nick Feamster and Jason Livingood. "Measuring internet speed: current challenges and future recommendations". In: *Communications of the ACM* (2020).

[11] Oana Goga and Renata Teixeira. "Speed measurements of residential internet access". In: *Passive and Active Network Measurement*. Springer. 2012.

[12] Ningning Hu and Peter Steenkiste. "Evaluation and characterization of available bandwidth probing techniques". In: *IEEE Journal on Selected Areas in Communications* (2003).

[13] Internet Equity Initiative. *Netrics: Automating Speed Test Measurements*. URL: https://github.com/chicago-cdac/nm-exp-active-netrics.

[14] Manish Jain and Constantinos Dovrolis. "Pathload: A measurement tool for end-to-end available bandwidth". In: *Passive and Active Measurements (PAM) Workshop*. 2002.

[15] Srinivasan Keshav. "A control-theoretic approach to flow control". In: *Communications architecture & protocols*. 1991.

[16] Measurement Lab. *BigQuery QuickStart*. URL: https://www.measurementlab.net/data/docs/bq/quickstart/.

[17] Measurement Lab. *Murakami: Automating Speed Test Measurements*. URL: https://github.com/m-lab/murakami.

[18] Measurement Lab. *Speed test by Measurement Lab*. URL: https://speed.measurementlab.net/.

[19] Kyle MacMillan, Tarun Mangla, James Saxon, Nicole P Marwell, and Nick Feamster. "A Comparative Analysis of Ookla Speedtest and Measurement Labs Network Diagnostic Test (NDT7)". In: *arXiv preprint arXiv:2205.12376* (2022).

[20] *Measuring Broadband America*. Apr. 2022. URL: https://www.fcc.gov/general/measuring-broadband-america.

[21] *National Broadband Availability Map*. URL: https://www.ntia.doc.gov/category/national-broadband-availability-map.

[22] Battle For the Net. *Internet Health Test based on Measurement Lab NDT*. URL: https://www.battleforthenet.com/internethealthtest/.

[23] Netflix. *fast.com*. URL: https://fast.com/.

[24] Ookla. *10 Gbps? There's a Speedtest for That*. URL: https://www.ookla.com/articles/10-gbps-speedtest.

[25] Ookla. *About Ookla SpeedTests*. URL: https://www.speedtest.net/about.

[26] Ookla. *Speedtest by Ookla - The Global Broadband Speed Test*. URL: https://www.speedtest.net/.

[27] Vinay Joseph Ribeiro, Rudolf H Riedi, Richard G Baraniuk, Jiri Navratil, and Les Cottrell. "pathchirp: Efficient available bandwidth estimation for network paths". In: *Passive and active measurement workshop*. 2003.

[28] SamKnows. URL: https://www.samknows.com/.

[29] Srikanth Sundaresan, Walter De Donato, Nick Feamster, Renata Teixeira, Sam Crawford, and Antonio Pescapè. "Broadband internet performance: a view from the gateway". In: *ACM SIGCOMM computer communication review* (2011).

[30] Pennsylvania State University and Measurement Lab. *Broadband Availability and Access in Rural Pennsylvania*. URL: https://www.rural.pa.gov/publications/broadband.cfm.