UNDERSTANDING THE PERFORMANCE OF BROADBAND NETWORKS THROUGH THE STATISTICAL ANALYSIS OF SPEED TESTS

by

RUBÉN GARCÍA GARCÍA

Ingeniero de Telecomunicación, and
M.S. Telecommunication Networks,
Ramon Llull University (2009)

SUBMITTED TO THE ENGINEERING SYSTEMS DIVISION
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE IN TECHNOLOGY AND POLICY

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2011

© Rubén García García, MMXI. All rights reserved.

The author hereby grants to MIT permission to reproduce and distribute publicly paper and electronic copies of this thesis document in whole or in part.

Author ............................................................... Engineering Systems Division

Certified by ........ ........................ .................

David D. Clark
Senior Research Scientist, Computer Science and Artificial Intelligence Laboratory
Thesis Supervisor

Accepted by .............. ........................ .................

Dava J. Newman
Professor of Aeronautics and Astronautics and Engineering Systems
Director, Technology and Policy Program
Understanding the Performance of Broadband Networks through the
Statistical Analysis of Speed Tests

by

Rubén García García

Submitted to the Engineering Systems Division
on May 6, 2011, in partial fulfillment of the
requirements for the degree of
Master of Science in Technology and Policy

Abstract

In this thesis we performed the statistical analysis of a dataset containing raw data captured through user-initiated Internet speed tests. The NDT dataset represents the largest publicly-available source of raw measurements ever collected on the Internet. Using the statistical package R, we analyzed the dataset and quantified the impact that several factors have on speed measurements. Those factors include the distance between client and server (round-trip time), the receive window, buffer sizes along the path, the effect that users who run the test multiple times have on aggregate variables, and the effect of test durations in the outcome. Further, we used the information to obtain a clearer picture on congestion, an important phenomenon which is not widely understood, answering questions such as the proportion of tests that encounter congestion, how many times they encountered congestion and what sort of congestion it was. Thanks to the analysis of the data, we have been able to reflect on the current landscape of broadband Internet in the United States, characterizing the service offered by the top six Internet Service Providers from different angles. The current thesis describes a procedure that allows us to analyze the network capacity, by excluding from the analysis tests that are limited by causes alien to the network itself, such as limitations on the receive window imposed by the operating system. Finally, some recommendations are offered that we believe would make the NDT test a more useful tool to gather information about the performance of broadband networks, and we come up with policy recommendations for the different stakeholders in the broadband arena.

Thesis Supervisor: David D. Clark
Title: Senior Research Scientist, Computer Science and Artificial Intelligence Laboratory
Acknowledgments

Writing a thesis is a long journey. In my case, the journey started back in February 2010, at the beginning of my second semester at MIT. I walked into Dr. David Clark’s office at the 8th floor of the Stata Center, ready to find out more about what his research group, the Advanced Network Architecture (ANA) group, does, and to talk about my own research interests. We sat there and talked for a while and at some point he opened his laptop and pulled a TCP packet trace and started to examine it and to explain to me what everything in the graph meant, and to explain all the anomalies in the graph. After a few minutes distilling the information in the plot, he turned to me and said: “I’m just doing this to see if you have any sympathy for what I’m doing”, at which I laughed and nodded, “I do”. After that initial meeting, he introduced me to Steve Bauer, researcher at the ANA, with whom I met to discuss possible areas where I could work. Those were my first steps as a member of the ANA, a 16-month voyage that finishes as I turn in my thesis in May 2011.

David Clark is the first person I would like to thank. He has been an endless source of inspiration, of advice and guidance throughout this process. I am grateful to him for finding time in his extremely busy schedule, filled with travel and meetings, to talk to me and discuss my progress. He always had good insight to offer on my findings, and always came up with further questions to ask. I also want to thank him for his patience in explaining concepts to me and helping me define my thesis question.

I would also like to thank Steve Bauer for his constant supervision and help, for always being there ready to answer my questions and to contribute ideas for my analysis, and for caring so much about the progress of my work.

I also want to express my gratitude to other members of the ANA group who have made my stay more pleasant and fun, as well as helped me out with my research and thesis writing. To Steve Woodrow for always being ready to help, for the many interesting conversations, and for spreading his passion for the subject among the rest of us. To Josephine Wolff for making the long evenings at the office much more fun (as well as less productive ;-)). To Jesse Sowell for sharing his extensive knowledge with me. To Mikko Heikkinen for the fun conversations (work-related and not), and for bringing some more European flavor to the office. To Shirley Hung for making the ANA much more social, fun, and close together, and for baking those delicious treats for us! To Chintan Vaishnav for spreading kindness around and for having such a lovely baby. To Arthur Berger for his help and great input to my analyses. And to Susan Perez for being so diligent with her job and always ready to help.

Doing research that ultimately leads to writing a thesis is a process with its ups and downs, good moments when things work and results arrive, and bad moments when nothing seems to make sense and you feel you’ve lost direction. And, especially for the latter, I have to deeply thank my friends in the Technology and Policy Program (TPP) for their support, for being so inspiring and talented, and for always putting a smile in my face. The list is long, so I will only name a few and extend my gratitude to all the rest: my roommates Andy Whitaker, Arjun Gupta and Alex Loijos, and fellow TPPers Pam DeAmicis, Rachna Pande, Paco Alonso, William Li, Georgie Campbell, Kris Jensen, Stephen Zoepf, Roedolph Opperman, An Vu, Melissa Zgola, Katharine Chu, Judy Wang, Pernilla Regårdh, Caroline Brun, Eriko Nishimura, Shreya Dave, Mike Kearney, Suhail Ahmad, Ece Gülşen, Hiro Ono, Anna Delgado, Toni Tugores, Mauricio Gómez, Amanda Cuellar, Anjuli Figueroa and Sarah Wood.

Also my deepest gratitude to my friends from Spain@MIT. Being president of Spain@MIT
is one of the things I’m most proud of during my stay at MIT. We have achieved great things this year, and most importantly, it has been a great experience to meet such nice, fun, and supportive people and spend time with you, and to corroborate that there is plenty of brilliant and inspiring people in Spain. Again, the list is long and I hope I don’t forget anyone (I apologize if I do): Roberto Sanchís, Fernando de Sisternes, María José Nieves, Patricia Gómez, Maite Peña, María de Soria, Javi Hernández, Isa Garós, Jorge Cañizales, Jorge Aguilera, Enrique Lizárraga, Gabriel Torres, Diego Féijer, Michelle Valverde, Mariana Rodríguez, Laura Escudero, Lucinio Muñoz, María Fraile, Rafa Marañón, Amparo Cañaveras, and Miguel Ángel Sánchez.

And last but not least, many many thanks to my parents, Justo And Merche. Thank you for everything you’ve done for me. Thanks for always supporting me, encouraging me to think big and to fight for my dreams. Thanks for always believing in me, even when I didn’t.

NOTE: This work has been done thanks to the support of the “la Caixa” Fellowship for Graduate Studies in the United States.
# Contents

1 Introduction .......................................................... 13  
1.1 Motivation and objectives ............................................. 13  
1.2 Related work .......................................................... 15  
1.3 Major findings ........................................................ 17  

2 Background ................................................................. 21  
2.1 Brief overview of TCP flow control and Internet speed tests .......... 21  
  2.1.1 Brief introduction to TCP flow control .......................... 21  
  2.1.2 Internet speed tests .............................................. 23  
2.2 The Network Diagnostic Tool .......................................... 25  
  2.2.1 What is the Network Diagnostic Tool (NDT)? .................. 25  
  2.2.2 The NDT infrastructure ......................................... 28  
2.3 Web100 Project ......................................................... 30  

3 Initial Analysis of the NDT Dataset .................................. 33  
3.1 Data Cleaning .......................................................... 33  
3.2 Initial Transformations ............................................... 34  
3.3 Characteristics of the Data Sample ................................ 35  

4 What the data tells us and how a reckless interpretation of the data can be catastrophic .............................................. 39  
4.1 Speeds per country – or the effect of Round-Trip Time on speed .... 39  
4.2 Analysis of speeds by Internet Service Provider and by Autonomous System .............................................. 41  
4.3 The big buffer problem ............................................... 48  
4.4 The effect of the receive window on speed ......................... 50
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.5 Study of congestion signals</td>
<td>54</td>
</tr>
<tr>
<td>4.6 The effect of test durations</td>
<td>67</td>
</tr>
<tr>
<td>4.7 Measuring speeds not limited by the Operating System</td>
<td>71</td>
</tr>
<tr>
<td>5 Policy implications of these analyses</td>
<td>77</td>
</tr>
<tr>
<td>5.1 Relevance of broadband measurements</td>
<td>77</td>
</tr>
<tr>
<td>5.2 Possible enhancements in NDT testing procedure</td>
<td>78</td>
</tr>
<tr>
<td>5.3 Policy recommendations</td>
<td>80</td>
</tr>
<tr>
<td>6 Conclusions and future work</td>
<td>83</td>
</tr>
<tr>
<td>6.1 Concluding remarks and contributions from the current work</td>
<td>83</td>
</tr>
<tr>
<td>6.2 Future lines of action</td>
<td>85</td>
</tr>
<tr>
<td>A Data cleaning</td>
<td>89</td>
</tr>
<tr>
<td>A.1 Adding 'whois' information to each test</td>
<td>89</td>
</tr>
<tr>
<td>A.2 Grouping tests by geographic location</td>
<td>90</td>
</tr>
<tr>
<td>A.3 Division of the dataset for specific uses</td>
<td>91</td>
</tr>
<tr>
<td>A.4 Aggregation of the tests by IP</td>
<td>93</td>
</tr>
</tbody>
</table>
List of Figures

3-1 Distribution of tests among the top countries present in the NDT dataset . 37
3-2 Proportion of repeating users and repeated tests, out of the total tests recorded in the NDT dataset. ................................................. 38

4-1 Average speed per country, top countries ........................................ 41
4-2 Distribution of speeds and average round-trip time per country, top countries. Blue dots represent the average speeds of each country. ......................... 42
4-3 Distribution of speeds and average round-trip time per ISP, top ISPs. Blue dots represent the average speeds of each provider. .......................... 44
4-4 Average speeds and average round-trip times per autonomous system, colored by Internet Service Provider. The size of the bubbles is proportional to the number of tests registered from that AS. ........................................... 46
4-5 Average speeds and average round-trip times per autonomous system, colored by Internet Service Provider. Tests are aggregated by IP address of the client. The size of the bubbles is proportional to the number of tests registered from that AS. ........................................... 47
4-6 Distribution of minimum round-trip times per ISP. ............................. 49
4-7 Distribution of average round-trip times per ISP. ............................... 50
4-8 Histogram of the maximum receive window sizes advertised for each test. 52
4-9 Scatterplot of speed vs round-trip time per test, colored by maximum receive window advertised during the test. ............................... 54
4-10 Distribution of speeds per ISP, for tests with round-trip times between 85 and 90ms. ................................................................. 55
4-11 Time series of NDT tests by the day, divided into tests limited by the congestion window, the receive window, or both. .............................. 57
4-12 Time series of NDT tests by the hour, divided into tests limited by the congestion window, the receive window, or both. ............................................. 58
4-13 Time series analysis of speeds recorded for Comcast customers not bound by the receive window, on a 24-hour basis. ............................................. 59
4-14 Time series analysis of speeds recorded for Comcast customers not bound by the receive window, except for the user who ran most tests, on a 24-hour basis. 60
4-15 Histogram of the number of congestion signals per test, tests in the United States. ................................................................. 63
4-16 Scatterplot of speed vs round-trip time per test, colored by number of congestion signals encountered. ................................................ 65
4-17 Scatterplot of speed vs round-trip time per test, colored by number of congestion signals encountered, and divided in two plots based on whether the test encountered congestion at all or not. ................................. 66
4-18 Average speed recorded per test, plotted according to the date that the test was performed, and colored by the duration of the test. ..................... 69
4-19 Average speed recorded per test, plotted according to the duration of the test, and colored by the number of congestion events encountered. .............. 70
4-20 Histogram of the number of congestion signals per test, for a given test duration. Each of the sub-plots contains the histogram of tests for a certain duration, specified at the heading of the sub-plot. .............................. 72
4-21 Distribution of speeds and average round-trip time per ISP, top ISPs. The subfigures compare tests not “OS limited” (tests limited by the receive window 5% or less of the connection, and with a WinScaleRcvd option greater than 3), with the distribution for all tests. Blue dots represent the average speeds of each provider. .......................................................... 74
4-22 Distribution of speeds and average round-trip time per ISP, top ISPs. The subfigures compare tests not “OS limited” (tests limited by the receive window 5% or less of the connection, and with a WinScaleRcvd option greater than 3), with tests “OS limited”. Blue dots represent the average speeds of each provider. .......................................................... 75
List of Tables

3.1 Statistics on the prevalence of certain TCP configuration parameters. The “Tests” column corresponds to the number of tests recorded out of the total 381,240, and the “Users” column corresponds to the number of unique users, out of the total 198,894. .................................................. 38

4.1 95% confidence intervals for the average speeds of each of the six Internet Service Providers studied. ................................................................. 45

4.2 T-test values for the true difference in average speeds between ISPs, with a 95% confidence interval. ................................................................. 45

4.3 Summary statistics for the MaxRwinRcvd variable, dataset containing global tests. ................................................................. 53

4.4 Summary statistics for the CongestionSignals variable, tests in the United States. ................................................................. 62

4.5 Number of tests featuring the different congestion variables. Tests in the United States that experienced any congestion signal. .......................... 64
Chapter 1

Introduction

1.1 Motivation and objectives

The Internet and broadband networks have acquired a significant relevance over the past two decades, and have become mainstream over the past few years, providing people with new possibilities never imagined before and affecting their lives in many aspects, from the way they communicate with each other, to their purchasing habits and satisfying their information needs. However, even today after the Internet has been around since the 1960s, there is no standardized and accurate methodology to characterize the speed of broadband networks.

This lack of appropriate measuring methodologies has led to uncertainty in the policy arena, in questions such as what the appropriate way to measure broadband speed is, or if speed is the right way to characterize the service at all. There has also been regulatory concern that customers may have been getting less speed than they should (as advertised by the Internet Service Providers), but often the data used to make those statements has been misused or misunderstood.

Speed measurements play a critical role in the Internet and telecommunications policy arena because those numbers are oftentimes used to characterize the service and everything that this characterization brings along: from investment in infrastructure to fining an operator for offering less speed than advertised. Since there is no standard way to measure speed in broadband networks, each party (cable providers, DSL providers, Internet measurement companies) is free to use different algorithms and parameters that may better serve their needs and, therefore, policy actions will be different depending on which measurements they
are based on.

The current time is very important for the future definition of the broadband space: the FCC recently released its National Broadband Plan that calls for universal service, and is in the process of launching a large-scale, country-wide study that aims to provide a clear picture of the current state of broadband connections in the United States. A British company that conducted a similar study in the United Kingdom, SamKnows, is in charge of deploying measurement boxes across the country and conducting tests that will inform policymakers. SamKnows is in ongoing conversations with the FCC, Internet Service Providers and academics about which tests should be run on those measurement boxes and how those tests should be designed.

At this stage, the competition between providers to root for their preferred parameters is fierce. Managing to get SamKnows to pick a parameter that favors them would make them look better in the eyes of both the FCC and the general public. This is not a public debate however; it has not been published nor expressed in front of a big audience, but my research team, the Advanced Network Architecture Group in the Computer Science and Artificial Intelligence Laboratory at MIT is participating in the conversations with the FCC, SamKnows and the other ISPs.

Prior to the measurements being conducted by SamKnows and the FCC, Richard Carlson (an engineer at the Internet2 consortium) built an open source measurement tool to help any Internet user learn more about the performance of their connections[1]. That tool, called the Network Diagnostic Tool (NDT) is based on a client-server application, where users run the speed test on demand and all the data gathered is made publicly available. Unlike the SamKnows tests, any user can run the NDT test anytime to gather valuable information about their connection, and the overall result is a large dataset rich in information waiting to be analyzed and understood. In an effort backed by academic institutions and industry giants such as Google or Amazon, terabytes of data are being made available and a web interface is being provided for the average user to visualize the data.

But obtaining numbers without understanding them is dangerous. As Vint Cerf, considered one of the fathers of the Internet and Chief Internet Evangelist at Google puts it, "it is not a trivial thing to understand the meaning of that data. [...] The purpose of data gathering is not numbers, is insight"[2]. With such a vast quantity and depth of data we can obtain a good picture of the broadband landscape, but "researchers who analyze this data
need to make their conclusions more visible to the public”. And that is exactly the goal of this thesis. We aim to shed light onto the data and advocate for the right ways to look at it and interpret it and, at the same time, to highlight potential mistakes that could be made when analyzing the data if certain factors are ignored.

This thesis will start by providing some background information on Internet speed tests and the mechanisms that TCP uses to control the data flow, and explaining what the Network Diagnostic Tool is and the variables it uses, in Chapter 2. Next, Chapter 3 goes over the initial processing of the NDT dataset: the filtering of the erroneous data and the different transformations and divisions performed on the dataset, as well as some descriptive statistics of the data sample. That leads to Chapter 4, where different variables in the dataset are analyzed with the goal of establishing an analysis methodology and obtaining insight from the data. From those analyses, we extract some suggestions to improve the NDT test and discuss some derived policy recommendations, in Chapter 5, and finish by recapitulating the conclusions and contributions derived from our work, and by outlining possible future work to expand the current research, in Chapter 6. Appendix A is an extension to Chapter 3 and contains more details on the data cleaning procedure. Finally, a comprehensive list of the Web100 variables contained in the dataset we analyzed can be found in http://www.web100.org/download/kernel/2.5.30/tcp-kis.txt, and the source code that we used for all of the analyses described in this thesis, written for the R statistical package, can be found online in http://hdl.handle.net/1721.1/62812.

1.2 Related work

The field of broadband modeling and analysis is rich in research. Many authors over the years have tried to understand the operation of communications networks and to model their behavior, with the goals of increasing the predictability of connections and enhancing their performance. In [3], a landmark of broadband measurements, the author designs a measurement framework that aims to measure the behavior along a large number of Internet paths only setting up measurement points in a small number of sites, namely \(O(N^2)\) paths for a framework consisting of \(N\) sites. Thanks to that framework, the author analyzes routing, outages, packet loss, and variation in delays, among other parameters, finding a wide range of behaviors, including occasional violations of some common assumptions.
such as in-order packet delivery or FIFO bottleneck queueing, concluding that one must be cautious when depicting any aspect of packet dynamics as "typical". The main contribution of the mentioned PhD thesis is the development of a methodology that might contribute towards the development of a "measurement infrastructure" that would serve to analyze general Internet behavior. Many of the subsequent essays on broadband measurements build on the fundamentals developed in [3].

Over the years, researchers have worked on enhancing measurement methodologies, with the aim of being able to obtain a more generally valid picture of Internet performance. In [4], the authors develop a passive measurement framework which collects TCP/IP packet headers on a single unspecified measurement point and analyzes them to unveil the factors that limit the throughput of TCP connections. They build a model that computes a value they call limitation score, and they compute how big of a role different factors such as buffer sizes, path capacity, or receive window, play in limiting TCP speeds. Similarly, in [5] the authors investigate the question of how the network needs to be sampled in order to obtain reliable estimates of important characteristics such as round-trip time (RTT) and loss rates, which then they use to develop a model that predicts TCP throughput using those parameters.

Speed is the parameter that draws most of the attention and literature. In [6], the authors build a model to predict TCP throughput based on parameters such as initial congestion window size, packet loss, and round-trip times, among others. They also study loss recovery mechanisms such as fast retransmits and timeouts. Similarly, in [7], the authors develop prediction algorithms for the throughput of large TCP transfers based on characteristics of the network such as RTT, loss rate or available bandwidth, and they test their models empirically in a testbed. A new element they add to their algorithms is the use of history-based predictors such as moving average and Holt-Winters, which they find to be quite accurate using a history of limited and sporadic samples. In a related article [8], the authors review several of the tools and methodologies for estimating bandwidth employed back then. Other studies have focused on studying different characteristics of TCP connections, such as loss detection and recovery [9], or the size of the initial congestion window [10].

The current thesis builds on the previous literature on network measurements but from a different angle. Instead of building a theoretical model for predicting the behavior of Internet traffic, our goal is to perform the statistical analysis of already existent Internet
traffic measurements to (1) better understand the performance of broadband networks and (2) to develop a methodology for future analysis of Internet data that can serve as input to both customers and policymakers. As opposed to the aforementioned literature, we have a very rich collection of data from speed measurements run at the users’ discretion, and therefore the development of non-invasive network sampling methodologies is not an objective of the current work. Instead, the current thesis builds on the work of [11] where the authors studied speed measurements conducted by different entities and highlighted how different test methodologies and conditions can lead to very disparate test outcomes. In the current thesis we will continue to show which factors influence measurements and to what degree, and advocate for best practices in the collection, analysis and communication of Internet traffic data.

1.3 Major findings

This section aims to foreshadow some of the main results and findings that we came across while working on this thesis, and to point to the sections where those results are explained in greater detail, for a more comprehensive explanation.

Through our analyses of the speed tests contained in the NDT dataset, we have been able to characterize many aspects of the behavior and operation of broadband networks, including the following:

- We have shown how important the effect of round-trip time on speed is, being able to distort the rankings of speeds per country and make leading countries such as Korea or Japan fall behind speeds measured in the United States, only because NDT test servers are located in the United States and not in Asia (see section 4.1).

- We have measured the speed distributions for the top 6 Internet Service Providers in the United States, with a 95% confidence interval, and measured the statistical range for the true differences in average speeds between providers (see section 4.2).

- We have shown how the so-called “heavy users” (users that run the test a large number of times) can distort speed averages and medians, and we advocate for a procedure to average speed tests by client IP in order to report only one data point per
client and obtain a better picture of the service enjoyed by the users at large (see section 4.2).

- We have shown how some ISPs (in our data, Verizon) might be using erroneous buffering practices, the so-called “big buffer” problem, which makes the average round-trip times for their clients’ connections increase, and consequently their speeds drop (see section 4.3).

- We have shown how the receive window is one of the main factors limiting the speeds that Internet users experience, which in most cases means that the Operating System in the receiving end is doing a poor job managing the receive window. We have proved how the recorded tests fit the theoretical model of the maximum achievable throughput given a certain receive window size and round-trip time, and how the most common receive window size, by far, is still 64 KB, which is small for today’s connections and a huge limitation in the speeds enjoyed by customers (see section 4.4).

- We have shown how only a small proportion of the tests (around 15% across the day, increasing up to 25% during nighttime) exhibit a healthy TCP behavior in which the flow of data is controlled by the congestion window, whereas the majority are partially or totally limited by the receive window. For a small subset of tests with similar cable connections that were bound by the congestion window, we found a time-of-the-day effect on speeds where speeds are higher during the night and lower during the day. We have also characterized congestion by quantifying the number of congestion events encountered and dissecting the types of those congestion events. We show that Fast Retransmit is by far the most common event, and that 97.71% of the tests that encounter congestion experienced at least a Fast Retransmit event. We have also characterized the effect that an increase in congestion events has in the speeds measured (see section 4.5).

- We have highlighted the importance of the duration of a speed test for its result. We designed an experiment in which one client ran speed tests of different durations to an NDT server that we controlled during a week. Our data shows that increasing test duration from 5 to 30 seconds made median speeds decline over 34% (from 23.1 to 15.2 Mpbs) (see section 4.6).
• We have designed a procedure to measure the capacity of a broadband connection itself, removing the limitations caused by the operating system in the client side. Our results using this methodology show that the service level offered by ISPs is higher than generally believed, and close to their advertised speeds (see section 4.7).

Derived from these findings, we have proposed some enhancements to the NDT test to make it more relevant (section 5.2), and some policy recommendations for the different stakeholders in the broadband arena (section 5.3).
Chapter 2

Background

2.1 Brief overview of TCP flow control and Internet speed tests

2.1.1 Brief introduction to TCP flow control

The Transmission Control Protocol, TCP, is the dominant transport protocol used by Internet applications—including the world-wide web, peer-to-peer file sharing, and media streaming[12, 13, 14, 15]. One of the key features of TCP is that it offers reliable data transfer, by detecting the loss of packets and retransmitting them. TCP carries out the loss detection by assigning sequence numbers to each packet it sends, and by having the receiver acknowledge the received packets. TCP uses a windowing system where only a certain number of packets can be sent until a previously sent packet is acknowledged by the receiver. When a loss is detected, the sender retransmits the lost packet. For that purpose, TCP uses primarily two detection mechanisms: retransmission-timeouts (RTOs) and fast-retransmit/recovery (FR/R)[9].

The retransmission timeout is a timer that starts after one packet is sent, and is reset when an acknowledgement for the packet is received. If the timer expires before the packet is acknowledged, the sender resends the packet, assuming it got lost. Fast-retransmit/recovery operates based on duplicate acknowledgements. When the receiving side of a TCP connection receives a packet with a higher sequence number than the one it is expecting, it sends a duplicate acknowledgement asking for the packet it is missing. If the sending side receives three duplicate acknowledgements asking for the same packet number, it assumes the packet
is lost and resends it. FR/R represents a speed improvement over RTO, since generally the three duplicate ACKs arrive much earlier than the retransmission timer would have expired, and hence the sender is able to retransmit the packet earlier, therefore disrupting the flow of data for a shorter interval.

Speeds on a TCP connection are not constant from beginning to end. The TCP protocol uses different mechanisms to control the amount of information that is sent at different times of the connection in order to fully utilize the available capacity, while avoiding causing congestion in the network. The two main variables that affect the behavior of the sending side, and therefore limit transmission rates, are the congestion window and the receive window (given that the sender has enough information to send, otherwise a sender stall will happen).

The receive window is set by the receiving end and reflects on the buffer space that the receiver has available. Its value is sent back to the sending side in order to inform of how much data the receiver is able to handle without overflowing its buffer, and the sender will act accordingly and not send more data than it is allowed to. However, the correct functioning of a TCP connection does not only depend on the amount of buffer space available at the receiving side: there may be other bottlenecks along the path between sender and receiver, and the TCP sender needs to take those bottlenecks into account. That is the purpose of the congestion window, explained below. Since the congestion window reflects on the limits of the network, and the receive window reflects on the limits of the receiving side, the sender will pick the more restrictive window of the two when determining the amount of data it can send.

The congestion window is a state variable kept at the sending side of a TCP connection and it reflects on the network (it’s capacity and level of utilization), telling the sender the amount of packets it can send without overwhelming the network[16]. Transmission starts with a congestion window of 1: the sender sends one packet and then waits to receive acknowledgement from the receiver. Every time the process is repeated successfully, the congestion window is doubled in size and transmission speed ramps up exponentially. This initial phase is called slow start for historical reasons ("slow" as opposed to starting the transmission by sending a full window of packets at once). Slow start and its expo-

\[1\text{Note: the congestion window is defined in terms of bytes, but it is conceptually easier to explain and think about it in terms of packets. The TCP parameter that relates packets and bytes is the Maximum Segment Size (MSS).}\]
nential growth of the congestion window last until the connection encounters congestion (for instance, a buffer in an intermediate router gets full) and a packet is dropped. When the sending side realizes that a packet got dropped, this is a signal that it is sending data faster than the network can absorb, and it reacts by reducing the size of the congestion window by half in order to reduce the amount of new data sent into the network and help alleviate congestion. At that point, the window starts to grow linearly instead of exponentially, adding the equivalent of 1 packet to the size of the congestion window every time the source successfully sends a full window of packets and receives acknowledgements for them. This process of linear growth and reduction by half when congestion is encountered is called additive increase/multiplicative decrease (AIMD), and is repeated successively until the end of the transmission, unless more severe problems are encountered, which may cause the congestion window to drop to 1 and to start the slow start phase again.

With the introduction of higher bandwidth networks, some researchers are proposing to increase the initial congestion window size to at least 10 segments, so that the connection throughput ramps up faster[10]. This is especially important for short-lived connections, such as web browsing, where a small volume of information is transmitted and the connection remains in the slow start phase for its entire duration.

2.1.2 Internet speed tests

What is an Internet speed test?

Internet speed test is a generic way to refer to the different methodologies used by different sites or organizations to quantify how fast an Internet connection is going. There is no standardized, universally accepted way to measure speed, since different techniques may be relevant for different Internet uses (eg. web browsing vs. gaming vs. file sharing) and since the conditions in which a test is performed are also variable and relevant to the measurement (eg. distance of the user to the DSL central, operating system in the user end, or load level in the network).

On a high level, speed tests could be classified by two criteria: specificity of the test, and point of measurement:

- Speed measurements can be conducted over traffic generated specifically for the test, or over existing, unrelated, data on connections.
The Internet user can go to a specific website to measure the speed of her connection. Examples of such services are Speedtest.net or the Network Diagnostic Tool (NDT) offered at Broadband.gov. These tests are characterized because they are triggered at user request, and because data traffic is generated and transmitted specifically for the purpose of measuring the speed.

The provider (content provider or Internet Service Provider) can run speed measurements for their own assessments over traffic flowing through their servers. Examples are Akamai or Youtube, who measure speeds using their own traffic from and towards the clients. In these cases, the measurements are done at the provider's discretion, and are computed over the content traffic sent and received by the provider's servers, rather than traffic generated specifically for the test as was the previous case.

By point of measurement, we can differentiate between server-side and client-side tests.

- Server-side measurements are those where the measurements are taken at the server, where I define server as the content provider, speed test website, or in general the entity responsible for the speed test. Recording the variables at the server will mean that in download tests we will be measuring at the sending side while in upload tests we will be measuring at the receiving side. This has implications in that certain variables only make sense and should be taken into account when observed at the sending side since, for instance, the receiving side does not see acknowledgements or congestion signals.

- Client-side tests are those where the measurements are taken at the Internet user's computer or home network, either through an instrumented browser or through measurement boxes installed in the user's local network. An example of client-side test that runs through a signed Java applet in the user's Web browser is Netalyzr\[17\], and an example of client-side test that uses measurement boxes is SamKnows, a British company that deployed 2,128 measurement units in residential customers' houses in the UK in September 2008\[18\], and that is now collaborating with the Federal Communications Commission (FCC) to conduct a similar study in the United States\[19\].
What determines the result of a speed test?

Many factors can influence the outcome of a speed test. These factors can be technical - related to how the test is implemented, or “alien” - external factors beyond control of the test design. However, both types of factors are not mutually exclusive: in the end, technical factors will determine which external factors limit the performance of the test.

An example of technical factor is the number of TCP connections opened to run the test: Speedtest.com opens up to 8 parallel HTTP threads during the test, while NDT runs on a single TCP connection[1]. This is relevant because, despite the fact that TCP is designed to occupy the available bandwidth with one connection, it has been shown that multiple connections are more effective doing so, leading to higher speeds than a single connection[16]. Moreover, web browsing generally runs over multiple HTTP connections like the Speedtest.com test does, while file transfers generally run over a single connection, like NDT, so each test may be more significant for a different Internet use.

Alien factors may include distance between the user and the test server, or time of the day when the test is run. Bigger distances means higher round-trip times (RTT) between client and server, which in turn translate into lower speeds, while tests performed during peak hours may encounter more congestion than tests performed during the early morning. All these different factors, whether they impact speeds and to what degree, will be analyzed in Chapter 4.

2.2 The Network Diagnostic Tool

For the analyses conducted during this thesis, we have used a publicly available dataset containing a very comprehensive set of variables measured during speed tests that users around the globe did using the Network Diagnostic Tool (NDT). For simplicity, from now on we will refer to this dataset as the “NDT dataset”.

2.2.1 What is the Network Diagnostic Tool (NDT)?

The Network Diagnostic Tool is a client/server application that performs a download and upload speed test at the user’s request. The NDT is one of the tools deployed at the Measurement Lab (M-Lab), which is an open, distributed server platform where researchers can deploy the tools they develop for measuring different aspects of the Internet. M-Lab
aims to enhance the level of understanding on how the Internet operates and to increase awareness on Internet performance, both in a consumer level and also to inform the public policy discussion[20].

Measurement Lab is an initiative founded and promoted by[21]:

- New America Foundation’s Open Technology Initiative
- PlanetLab Consortium
- Google
- Academic researchers:
  - Georgia Institute of Technology
  - Internet2
  - Max Planck Institute
  - Northwestern University
  - Pittsburgh Supercomputing Center
  - Princeton University

And supported by:

- Servers hosting and connectivity
  - Google
  - Voxel
  - Helenic Telecommunication & Post Comission (EETT)
  - Australia’s Academic and Research Network (aarnet)

- Client support
  - uTorrent
  - FCC
  - SamKnows

- DNS hosting and server selection
Princeton University

- Data hosting
  - Amazon
  - Google

The above sponsor companies offer the platform so that researchers can build tools to measure different aspects of Internet traffic. One of those tools is the Network Diagnostic Tool, developed by Richard Carlson, an engineer at Internet2. NDT attempts to measure the performance of TCP over a path and diagnose the causes for any possible errors or low performance in the connection[1]. Studies have shown that the majority of network performance problems occur in the user’s equipment or premises rather than in the access network or backbone. The NDT test aims to identify certain conditions that impact network performance, such as duplex mismatch conditions, incorrectly set TCP buffers, or problems with the local network infrastructure[22]. For that purpose, NDT relies on a server application that consists of an instrumented TCP stack and a packet trace capture. On the client side, NDT uses a java application loaded in the user’s browser, which, at the user’s request, attempts to send traffic as fast as possible for an interval of 10 seconds to measure upload speed. Then the test is repeated in the opposite direction, and the server sends packets as fast as possible for another 10 seconds to measure download speed. During the upload and download tests, NDT records the values of a set of variables every 5 milliseconds. Those are the Web100 variables, a set of parameters defined by The Web100 Project to diagnose TCP performance issues. The Web100 variables will be explained in greater depth in section 2.3 and a comprehensive list of the variables can be found in http://www.web100.org/download/kernel/2.5.30/tcp-kis.txt. The values of those variables paired with a TCPdump log of the connection represent a very detailed and comprehensive picture of the test, providing the researcher with an amount of data unparalleled by other similar tests[11].

In March 2010, the FCC launched its Consumer Broadband Test, which aims to inform consumers of the performance of their Internet connection, as part of its Broadband.gov website. The Consumer Broadband Test randomly assigns users to one of two measurement tests—NDT or a speed test from Ookla. The fact of NDT being included in the Broadband.gov website gave the tool an increased relevance, because of the endorsement
that represents having the FCC rely on its test methodology, and because of the enormous increase in the number of tests that this represented.

Moreover, recently other institutions and companies started to adopt NDT as their reference tool to assess broadband performance. Examples of this are BitTorrent, which built the NDT test in one of their applications[23, 24], or EETT (Greece’s telecommunications regulator), which plans to use data collected through the M-Lab platform to provide information to consumers so that they will be able to compare broadband providers and their connections’ performance across several dimensions[20, 25]. These examples highlight the importance of having a good, open tool to analyze broadband performance such as NDT, since most countries will not be able to afford paying an external company to deploy thousands of measurement boxes across the country, like is the case of SamKnows in the United States.

2.2.2 The NDT infrastructure

The NDT test is hosted in a network of servers provided primarily by Google and maintained by PlanetLab. The network is composed of 48 servers in 16 locations, although two of the cities have two locations each. The list of locations is the following[21]:

- Atlanta (US)
- Chicago (US)
- Dallas (US)
- Los Angeles (US)
- Miami (US)
- Mountain View (US)
- New York (US) - 2 locations
- Seattle (US)
- Amsterdam (Netherlands) - 2 locations
- Athens (Greece)
• Hamburg (Germany)
• London (United Kingdom)
• Paris (France)
• Sydney (Australia)

Note how that means there are 9 test locations in the US, 6 in the European Union, and only one in the rest of the world. Furthermore, in the dataset that we analyzed there are no tests recorded from either the Sydney or Hamburg sites. This will have implications for the measurements, as we will see later on in chapter 4.

The system uses the DONAR DNS service to assign the client to its closest server. DONAR is a distributed system that provides name resolution and server selection for replication services. Each server uses Linux as its operating system, and uses the Web100 instruments in the Linux TCP/IP stack to expose over a hundred network related variables. Servers implement the TCP Selective Acknowledgement (SACK) option and use the New Reno modification to TCP's Fast Recovery Algorithm, which is a modification aimed at improving the performance of Fast Retransmit and Fast Recovery algorithms when partial acknowledgements are in effect[26]. Each site is composed of three identical servers for redundancy and load balancing. Each of those enterprise-class servers feature the following hardware[27]:

• 8x CPU cores
• 8GB RAM
• Dual 100+GB disks
• 1 Gbps Ethernet Network Adapter
• Built-in remote diagnostic, configuration and power reset interface (such as Dell DRAC, HP iLO or OpenIPMI, Intel AMT, etc.)

Those technical specifications and the level of redundancy built in the sites, coupled with the fact that any site is required to have a minimum of 1Gbps connectivity to an upstream provider with no filtering or firewall, seem to show that the infrastructure is well provisioned and will not constitute a limitation or bottleneck for the tests. In fact, the people responsible for maintaining the platform claim that currently only 1% of its resources are in use[27].
2.3 Web100 Project

Despite the fact that the Internet infrastructure has evolved greatly over the last decades, the growth in speeds experienced by the customers has been slower, and many users still enjoy far slower speeds than their connections would support. Furthermore, the difference in speeds experienced by the average user as compared to those experienced by “Internet wizards” has increased over the years, giving us an idea that the default TCP settings may not be optimal for high capacity networks such as the ones we enjoy nowadays, and that some adjusting by a technical expert leads to a greatly enhanced performance[28]. However, understanding what is causing a slow connection and which parameters should be adjusted is a difficult task even for those “Internet wizards”. To facilitate that task, The Web100 Project (http://www.web100.org) designed an architecture that provides for per-connection instrumentation of the TCP stack, with the goal of exposing protocol events and connection variables that are otherwise hidden by TCP, which by design hides the complexity of lower layers from higher layers[29]. The aim is to reduce the effort required by programmers and network administrators to fix minor bugs that impede higher performance.

The Web100 Project specified over 125 variables, also called instruments, which are precisely defined and are coded directly into the operating system kernel, in what is called the TCP Kernel Instrument Set (TCP-KIS). These instruments expose many of the variables of the TCP connection, such as the receive window, bytes sent, or the number of congestion signals received, among many others. For a complete list and explanation of the Web100 variables, please see http://www.web100.org/download/kernel/2.5.30/tcp-kis.txt. The Web100 library provides the necessary functions to access the values of those variables, so that researchers can analyze them and network diagnostic applications can be built on top. The latter use is what enabled the development of tools like the Network Diagnostic Tool. NDT is an application which uses an instrumented TCP stack in its servers that exposes and records the values of the Web100 variables during its TCP connections with clients. The values of those variables over a seven-month period compose the NDT dataset that is analyzed throughout the present thesis. Web100 is a passive Internet measurement tool, since it does not inject traffic into the network; it relies on reading the TCP variables. However, NDT relies on Web100 but it generates traffic specifically for the purpose of measuring the network performance, and hence is an active measurement tool.
Each of the aforementioned instruments is a variable in a “stats” structure that is linked through the kernel socket structure, making use of the Linux /proc interface to expose those instruments outside of the kernel[28]. A portable API was defined for accessing the instruments, keeping performance in mind. On a 2 GHz Mobile Pentium 4, taking a snapshot of 140 variables takes about 30μs, which repeated every 5ms does not represent a burden for the CPU[29].

The Web100 Project, whose development ended in 2003, was funded by the National Science Foundation and by Cisco Systems and has recently received new funding by the NSF for a new three-year project called “Web10Gig” which will build on the Web100 Project and correct weaknesses in Web100’s installation and eliminate barriers to its wide use, to enable measurement tools to operate underneath all types of network-based research[30]. The original Web100 Project was developed by a partnership of:

- Pittsburgh Supercomputing Center (PSC)
- National Center for Atmospheric Research (NCAR)
- National Center for Supercomputing Applications at the University of Illinois, Urbana-Champaign (NCSA)

The new Web10Gig Project is again a partnership of the PSC and the NCSA.
Chapter 3

Initial Analysis of the NDT Dataset

3.1 Data Cleaning

Prior to starting our analysis of the NDT dataset, it is important to remove certain rows that contain erroneous data. Otherwise, our estimates and conclusions about the data would be flawed.

Data can be erroneous for different reasons, but in our case those factors essentially boil down to two:

1. A technical failure in the test: for example, the test fails to end after 10 seconds and continues to send packets, or the server fails to record the values of the variables, among other possibilities. It is oftentimes difficult to predict or explain why in certain occasions software failed, but that seems to be the most likely explanation to certain pieces of erroneous data in the NDT dataset.

2. Human intervention: the user who runs the NDT speed test can take actions that will lead to an erroneous or incomplete test. Primarily, we have found numerous evidence of tests shorter than 10 seconds, which suggests that the users may have closed the test before it had finished.

Stemming from those two factors, we have identified two types of erroneous data that should be filtered out of the dataset before proceeding to analyze the data:

1. Tests with duration smaller than 10 or larger than 11.5 seconds. Since the NDT test is designed to last 10 seconds, test durations that deviate from that number mean that
something went wrong with the test, and it would be incorrect to include their values in the data aggregation. Moreover, since the test encountered some kind of problem, it could be possible that the values of other variables were incorrect as well. We admit a range of up to 1.5 seconds more than the stated test duration because ending the test is not something instantaneous, and the application needs to wait until the test is complete before finishing, that is, waiting until the packets sent before the timer reached 10 seconds do the round trip and their values are recorded.

2. Tests with $\text{CountRTT} = 0$. $\text{CountRTT}$ is a variable that records the number of round trip time samples included in the test data. We have encountered instances of tests where the value of $\text{CountRTT}$ is zero, which in normal operation of the test would be an impossible value. Moreover, since $\text{CountRTT}$ is used as a denominator when computing averages of other variables, it having a value of zero is problematic.

To see how the filtering is performed, please see the source code file “merge_filter_NDT4.R”, contained in the source code available in http://hdl.handle.net/1721.1/62812.

3.2 Initial Transformations

The original source of data used for the experiments in this thesis is the dataset containing the records of approximately 400,000 speed tests carried out using the Network Diagnostic Tool, hosted in Amazon Web Services. However, for the purpose of our analyses, the original raw data needed some processing. In order to not interrupt the flow of this document with a long section on the technicalities of how the transformations were performed, only a summary will be shown below. For a more detailed explanation, please refer to Appendix A.

The main transformations performed to the original data are the following:

Adding ‘whois’ information to each test. The original dataset contains a wealth of technical measurements of speed tests; however, it does not contain geographic information per se. It does contain the IP addresses of both client and server, and thanks to that information we have used an online lookup service to add information on the client’s Autonomous System, country and Internet registry. Internet registries are organizations that manage and allocate IP addresses and Autonomous System numbers within a particular region of the
world. The world is divided into five regional Internet registries: AfriNIC, APNIC, ARIN, LACNIC, and RIPE NCC[31].

Grouping tests by geographic location. First, we grouped all 27 countries belonging to the European Union under a single country code "EU". Then we ranked all countries present in the dataset by number of tests recorded, and selected the top 10 countries, which are the countries we will analyze throughout the current thesis. Finally, we grouped all other countries outside of the top 10 under a single country code "OT".

Division of the dataset for specific uses. The original, unprocessed dataset containing the records of 398,529 speed tests, which we called lastline, was combined with the location information contained in the dataset whois, resulting in the dataset filtWhois, which was then filtered as described in section 3.1 to result in the dataset filtLine, which is the main structure we will use throughout our analyses. Further subsets of filtLine include a set containing only tests that encountered congestion (congLine) and a dataset containing only tests in the United States (USLine).

Aggregation of tests by IP. There are many users who ran the test more than once, and some of those ran it a very significant number of times, enough to bias the results of the population. In some cases, we will be interested in studying the performance experienced by the population as a whole. In those cases, we are interested in seeing only one record per user. Therefore, for those cases we propose to average the results of the multiple tests run by a single user.

3.3 Characteristics of the Data Sample

In this section we will look into the dataset of NDT tests and generate some basic statistics to have an initial idea of the information contained in the dataset. Before performing the initial filtering and transformations described in sections 3.1 and 3.2 respectively, the raw numbers of the dataset are as follows:

- Total number of tests: 398,520
- Variables recorded for each test: 121
• Total countries represented: 178

• Range of observations per country: 106,260 (Brazil)-1 (multiple countries)

• Autonomous Systems (ASes) represented: 5,897

• Range of observations per AS: 20,896 (AS #28,573)-1 (multiple ASes)

After filtering the dataset and grouping the tests by country under the codes ‘EU’ and ‘OT’ as explained in section 3.2, we can obtain some other statistics on the distribution of tests:

• Total number of tests: 381,240

• Total countries represented: 11 (grouping the EU countries, top 10 countries, and all other under ‘OT’) 

• Range of observations per country: 112,911 (EU)-3,493 (Japan)

• Autonomous Systems (ASes) represented: 5,897

• Range of observations per AS: 20,896 (AS #28,573)-1 (multiple ASes)

Figure 3-1 shows the distribution of tests among the top countries represented in the NDT dataset. Not surprisingly, the European Union and the United States are among the most represented regions. What is more surprising is that Brazil ranks second, only after the European Union, and that Ukraine ranks 9th with more tests than South Korea and Japan. Also, as seen in the chart, the distribution of tests is not very uniform since the top three countries (EU, Brazil and US) represent nearly 75% of the tests, the individual countries ranked 4 to 10 by number of tests represent roughly 1/8 of the total tests, and the all other countries in the dataset combined (‘OT’) make up for another 1/8 of the sample.

In figure 3-2(a) we can see the distribution of users who ran the test only once as compared to those who ran it more than once. Of the total 198,894 users who ran the test, 136,608 (68.68%) ran it only once, and 62,286 (31.32%) ran it more than once. However, if we add the number of tests in the dataset that were run by “repeating users”, they add up to 244,632, or 64.17% of the total tests in the dataset, as shown in figure 3-2(b). This will
be significant, as we will see later on, in that users who repeat the test multiple times may be biasing the outcome of our analysis.

Finally, we will look into how prevalent certain configurations such as ECN (Explicit Congestion Notification) or SACK (Selective ACKnowledgement) are. Table 3.1 shows some basic statistics for both the \textit{filtLine} and the \textit{agLine} datasets, that is the dataset containing all tests and the dataset of tests aggregated per unique user. The main takeaways are: (1) SACK is commonplace, almost all the tests and unique users use it, (2) Timestamps were enabled only in a very small fraction of the tests, (3) ECN was not active in any test, and (4) the MaxMSS commonly used sizes correspond to two of the big sizes: 1452 and 1460 (the biggest possible size).
Figure 3-2: Proportion of repeating users and repeated tests, out of the total tests recorded in the NDT dataset.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Tests</th>
<th>Users</th>
</tr>
</thead>
<tbody>
<tr>
<td>SACKEnabled</td>
<td>378,281 (99.22%)</td>
<td>197,302 (99.2%)</td>
</tr>
<tr>
<td>TimestampsEnabled</td>
<td>22,936 (6.02%)</td>
<td>9,862 (4.96%)</td>
</tr>
<tr>
<td>ECNEnabled</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

MaxMSS size

<table>
<thead>
<tr>
<th></th>
<th>Tests</th>
<th>Users</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt; 1440:</td>
<td>39,338 (10.32%)</td>
<td>20,575 (10.34%)</td>
</tr>
<tr>
<td>&gt; 1440, !={1452,1460}:</td>
<td>88,674 (23.26%)</td>
<td>46,710 (23.48%)</td>
</tr>
<tr>
<td>1452:</td>
<td>103,600 (27.17%)</td>
<td>57,333 (28.83%)</td>
</tr>
<tr>
<td>1460:</td>
<td>149,628 (39.25%)</td>
<td>74,276 (37.35%)</td>
</tr>
</tbody>
</table>

Table 3.1: Statistics on the prevalence of certain TCP configuration parameters. The "Tests" column corresponds to the number of tests recorded out of the total 381,240, and the "Users" column corresponds to the number of unique users, out of the total 198,894.
Chapter 4

What the data tells us and how a reckless interpretation of the data can be catastrophic

In this chapter, we will proceed to analyze the NDT dataset, looking to draw conclusions on the performance of broadband networks. The NDT dataset is a rich source of data, with over 100 variables recorded for every test, and tests recorded all around the world. Performing statistical analysis of the dataset can bring to light some interesting relationships, but at the same time can lead to erroneous interpretations of the data, caused by the so-called "lurking" variables. "Lurking" variables are factors which one doesn’t take into account while analyzing the data, but that have an impact on the outcome. For instance, not stratifying the data correctly, or not realizing the effect that a third variable has in the relationship between two variables can lead to the wrong conclusions.

The following sections contain the analyses of different variables of the dataset, and show examples of both correct and incorrect readings of the data.

4.1 Speeds per country – or the effect of Round-Trip Time on speed

As the NDT dataset contains records of tests performed worldwide, a clear analysis one can perform is to compare the Internet access speeds in different countries. For the sake
of simplicity and not to overwhelm the reader with large amounts of information, we have chosen to limit the analysis to the 10 most represented countries in the dataset, and to combine all other countries under the category “Other”, as described in section 3.2.

This case is a good first example on how a reckless analysis can lead to the wrong conclusions. Figure 4-1 shows a barplot of the average speed recorded in the tests belonging to each of the 11 regions or countries studied. At first glance, it is surprising to find the United States leading all countries, with an average speed that almost doubles Korea’s and more than doubles that of Japan, when those two countries are generally considered to have the fastest networks[32, 33]. If our analysis ended here, we would be getting a result that would be misleading to policymakers in the involved countries, making them think that the speed in their networks is either higher or lower than expected.

However, if we delve deeper into the reasons why the data may be showing this results, we may recall that speed is greatly affected by distance. Considering that the NDT servers that run the speed tests contained in our dataset are only located in the United States and the European Union (see section 2.2.2), it is conceivable that the distance between client and server could play an important role. To study whether distance is indeed causing the aforementioned surprising results, we will include the average round-trip time between clients and servers in our plot.

Figure 4-2 shows the same plot of average speeds by country that we had previously seen, but at the same time plots the average round-trip time. Also, boxplots have been used instead of bars, to show the distribution of speeds. As we can appreciate neatly in this graph, round-trip time and speed have a clear inverse relationship. This effect is especially important in small transfers such as a typical Web page, even more so than the available bandwidth of the path[8].

The main takeaway from this analysis is the importance that distance has on speed measurements, and that using a summary variable such as average speed without understanding the details on how the measurements were taken can be misleading. The same principle shown in the previous graphs (that countries further from the test servers will be penalized) also applies, maybe at a smaller scale, to measurements within the United States. For instance, if two US Internet users are measuring their speeds and one is twice as far from the server as the other, his speed measurement will be greatly penalized because of that factor.
4.2 Analysis of speeds by Internet Service Provider and by Autonomous System

An important regulatory question that has been debated recently within the Federal Communications Commission is whether customers have been receiving an appropriate level of service from their providers. In [34], the FCC reaches the conclusion that ISPs are providing their users with less speed than advertised, based on data provided by comScore[35]. However, [36, 37] argue that speeds reported by comScore are low by a large margin. The main reasons for comScore underreporting speeds are the fact that the measurements are based on a single TCP connection[38], which will tend to understate the achievable speed of the broadband service and, primarily, mistakes when identifying the service tier that users
have purchased[11]. ComScore oftentimes fails to assign customers to their purchased speed tier when those users are in cable networks which employ Powerboost. ComScore assumes that measured speeds will never be above a 10% over the Powerboost level, but our research group has packet traces that prove otherwise[39]. The aim of this section is to analyze our dataset of tests classifying them by Internet provider and see which conclusions we can reach.

In order to better characterize the ISPs, we have selected the top 6 ISPs in the United States by number of tests in the NDT dataset, to make sure that the number of tests is significant to draw meaningful conclusions. The dataset we used is *USLine*, and the selected ISPs are:
• Comcast: 16,516 tests
• Road Runner: 8,475 tests
• AT&T: 8,365 tests
• Verizon: 6,005 tests
• Cox: 2,998 tests
• Charter: 2,700 tests

For a first analysis, figure 4-3 shows the quartiles of the distribution of speeds by provider, their mean value, and the average round-trip time experienced by tests from each provider. Things that call our attention from the plot are the fact that Verizon shows a very spread distribution, with a decent average speed but a surprisingly low median speed, as compared to the other providers. The wide distribution of speeds shown in Verizon tests can be explained because Verizon users include those with (slow) DSL connections and those with (fast) FiOS fiber-to-the-home/premises connections. Most of the users recorded probably have DSL connections, but the FiOS customers drag up the average speeds and make the distribution spread. It is also important to note that the low median speed comes coupled with a really high RTT, almost 2.5 times as high as that from Comcast or Cox. As we saw in section 4.1, round-trip times have a very significant effect in measured speeds. We could attribute that high RTT to the possibility that the Verizon Autonomous Systems (ASes) from which the test was run were far from the NDT servers, or to erroneous buffering practices, which we will analyze in detail in section 4.3.

Another interesting thing to note are the low speeds shown by AT&T. The explanation for that is that AT&T offers DSL connections to its customers, hence the median and average speeds of roughly two to three megabits per second. Finally, another observation are the average and median speeds shown by Comcast customers, a bit over 10 Mbps and close to the 12 Mbps that the ISP sells in its most popular speed tier. However, it should be noted that Comcast, like other cable providers, uses Powerboost to give its users a boost in speed during the first few seconds of a connection. Since NDT tests are designed to last 10 seconds, the effect of Powerboost can significantly bias the results of the speed tests, as we will see in section 4.6. We should also note that it is unknown to us how many of those customers
may have purchased a higher speed tier from Comcast, advertised at 15 Mbps. Table 4.1 reflects on the same data as figure 4-3, but gives us the 95% confidence intervals of the speed averages. Table 4.2 shows the true difference in average speeds between consecutively ranked ISPs. It has been computed using the t-test, and all the differences in speeds are certain with very small p-values (<7e-6).

In order to delve deeper into the analysis and understand better where these distributions of speeds come from, the next step is to analyze the speeds and RTTs by Autonomous System for each provider. For the sake of having representative data points, we have chosen to focus only on ASes with more than 200 tests. Figure 4-4 shows a bubblechart where each of the
<table>
<thead>
<tr>
<th>ISP</th>
<th>Low</th>
<th>High</th>
</tr>
</thead>
<tbody>
<tr>
<td>Comcast</td>
<td>10.078</td>
<td>10.284</td>
</tr>
<tr>
<td>Cox</td>
<td>9.181</td>
<td>9.718</td>
</tr>
<tr>
<td>Verizon</td>
<td>7.808</td>
<td>8.258</td>
</tr>
<tr>
<td>Road Runner</td>
<td>7.314</td>
<td>7.568</td>
</tr>
<tr>
<td>Charter</td>
<td>5.679</td>
<td>6.038</td>
</tr>
<tr>
<td>AT&amp;T</td>
<td>3.282</td>
<td>3.441</td>
</tr>
</tbody>
</table>

Table 4.1: 95% confidence intervals for the average speeds of each of the six Internet Service Providers studied.

<table>
<thead>
<tr>
<th>ISP Comparison</th>
<th>Low</th>
<th>High</th>
</tr>
</thead>
<tbody>
<tr>
<td>Comcast vs. Cox</td>
<td>0.444</td>
<td>1.019</td>
</tr>
<tr>
<td>Cox vs. Verizon</td>
<td>1.066</td>
<td>1.766</td>
</tr>
<tr>
<td>Verizon vs. Road Runner</td>
<td>0.334</td>
<td>0.851</td>
</tr>
<tr>
<td>Road Runner vs. Charter</td>
<td>1.362</td>
<td>1.802</td>
</tr>
<tr>
<td>Charter vs. AT&amp;T</td>
<td>2.301</td>
<td>2.693</td>
</tr>
<tr>
<td>AT&amp;T</td>
<td>3.282</td>
<td>3.441</td>
</tr>
</tbody>
</table>

Table 4.2: T-test values for the true difference in average speeds between ISPs, with a 95% confidence interval.

bubbles represents one of the autonomous systems present in the dataset. The bubbles are colored by ISP, and their sizes are proportional to the number of tests recorded in each AS. We see how AT&T and Verizon congregate most of their customers into one big AS each, while Comcast and Road Runner have their customers split into many smaller ASes. However we do observe how most of Comcast’s and Road Runner’s ASes form a cloud with similar speeds and RTT. The plot shows clearly how Verizon’s AS features a very high RTT, and how AT&T’s main AS shows relatively slow speeds, corresponding to their DSL connections.

Also important is noting how a few users who perform the test a significant number of times can distort a picture like 4-4. Following, we will use the dataset containing tests in the US aggregated by IP, agUSLine, so that we can observe the speeds and round-trip times of the user base, without being affected by a few heavy users. The ISPs studied are the same, but once the tests have been aggregated by IP the number of tests for each ISP is the following:
Figure 4-4: Average speeds and average round-trip times per autonomous system, colored by Internet Service Provider. The size of the bubbles is proportional to the number of tests registered from that AS.

- Comcast, 7,646 tests
- Road Runner, 4,309 tests
- AT&T, 4,264 tests
- Verizon, 3,094 tests
- Cox, 1,721 tests
- Charter, 1,308 tests

In figure 4-5 we can see the same type of graph as in 4-4, but with tests averaged by
IP. That means that when a user recorded more than one speed test, the values of all his tests have been averaged and, therefore, in the resulting dataset there is only one test record per user. This way we aim to obtain a better view of the speeds and round-trip times that users experience. Notice how some of the bubbles present in the previous plot now have disappeared. That means that most (or all) of the tests recorded in certain ASes were recorded by the same user. Also note how the bubbles are now even more lined up than in the previous picture, showing clearly the decline of speeds with round-trip times.

Figure 4-5: Average speeds and average round-trip times per autonomous system, colored by Internet Service Provider. Tests are aggregated by IP address of the client. The size of the bubbles is proportional to the number of tests registered from that AS.
4.3 The big buffer problem

Another topic that has been discussed recently relates to the sizing of buffers in Internet routers. Buffers are necessary to store packets as they arrive to the intermediate routers in the Internet backbone, so that packets can be retained while the router is processing other packets, instead of being dropped. This responds to the bursty nature of the connections, where a router may receive a burst of packets that it stores and processes, emptying the buffer as the rate of arrival of new packets decreases. The question is how to correctly size buffers in order to achieve optimal throughput. On one extreme, no buffering would mean that the router wouldn’t be able to absorb bursts of packets, and therefore this would be an extremely conservative approach in that it would continually tell the TCP sender to slow down because packets got dropped. On the other extreme, “infinite buffers” would mean that no packets would ever be dropped, and the TCP sender would continue to ever increase its sending rate and overwhelm the network with data, and packets would pile up in the buffers and experience enormous delays. Hence, the optimal point that does not drop too many packets but at the same time does not introduce too much delay lays somewhere in the middle.

Traditional literature states that the rule-of-thumb for sizing the buffers of Internet routers is the so-called bandwidth-delay product, which says that if we want the core routers to have 100% utilization, the buffer size should be greater than or equal to \(2T \times C\), where \(2T\) is the effective round-trip time (also denoted as RTT), and \(C\) is the capacity of the bottleneck link\(^{[40]}\). Later, it was shown that the buffer size calculated with the previous formula can be divided by \(\sqrt{N}\) without sacrificing throughput, where \(N\) is the number of flows sharing the bottleneck\(^{[11]}\). More recent literature has shown that the buffer sizes can be further reduced, if we are willing to sacrifice a small amount of link capacity. In \(^{[42]}\), the author argues that as long as the access network is much slower than the backbone, which is true today and likely to remain true in the future, buffers of \(O(\log W)\), where \(W\) is the congestion window size of each flow, are sufficient for high throughput. Some researchers believe that some ISP’s networks may have inappropriately large buffers, which would increase the round-trip time experienced by user connections therefore making their speeds decrease. This phenomenon is called the “big buffer” problem.

The aim of this section is to investigate our data to obtain a better picture of the round-
trip times that the users experienced, find differences between providers, and see if the data shows evidence of such “big buffer” problem. For that purpose, we have analyzed the dataset of tests recorded in the United States, USLine, and divided it by Internet Service Provider using the same list of the six major providers by number of tests that we previously used in section 4.2. Figure 4-6 shows the distributions of minimum round-trip times recorded for tests for each of the six ISPs. Note how the median minimum RTT for all six ISPs is very small and similar, ranging from 28ms to 39ms, and the distributions are very similar as well. First, since the minimum RTTs are very similar, that means that under optimal conditions (no load in the network), users take on average the same time to reach the servers. Hence, we can argue that there is no significant difference in terms of the physical characteristics of the user base for each provider, and on average their distances to the test servers are comparable.

![Figure 4-6: Distribution of minimum round-trip times per ISP.](image)

Now look at figure 4-7, which shows the distributions of average round-trip times, and observe how the distributions are now significantly different. In fact, the more spread dis-
tribution (Verizon’s) is almost three times as spread as the most concentrated one (Cox’s), and the range of median round-trip times now goes from 58ms to 132ms. This seems to hint that the conditions that TCP connections encounter when the network is loaded are more diminished for certain providers than others. A possible explanation for that is the aforementioned “big buffer” problem. For instance, Verizon and AT&T could have large buffers on their networks that would be causing the packets to wait in queue for a longer time. Instead, for other providers with smaller buffers, some packets may be dropped instead of waiting in line and that would trigger the congestion avoidance mechanisms. Another possible explanation is that the access network on certain providers (such as Verizon) is more congested that in other providers, and therefore packets encounter more traffic and queues.

4.4 The effect of the receive window on speed

Despite the fact that congestion is the first thing that usually comes to mind when thinking about factors that degrade broadband performance, the truth is that some TCP parameters, such as the receive window, play a role as important or even more so[11]. The TCP Receive
Window, or Rwin, advertised by the receiving side of a TCP connection corresponds to the amount of data that a computer can accept without acknowledging the sender\cite{43}. Therefore, independently of other factors such as round-trip time and congestion, windowing does play an important role in controlling the throughput of a connection, and an adequate tuning of the size of the receive window is critical to achieve a good throughput\cite{44}. This becomes problematic when some operating systems, especially older versions, don’t do an optimal job managing the receive window and cause TCP to advertise unnecessarily small windows, for example in the case where the TCPs do not use receiver window scaling\cite{4}.

First of all, let us get a better picture of the ranges of receive windows present in our dataset. Since the receive window is a parameter used by the receiving side to control the amount of information that it allows the sender to transmit, it is normal that during a connection its value will fluctuate. That is why we will focus on the maximum window that the receiving side advertised during a connection, recorded in the variable \textit{MaxRwinRcvd}. Figure 4-8 shows a histogram of the variable \textit{MaxRwinRcvd} for the global NDT dataset. As expected, we can see spikes in the most common window sizes, that is 16 KB, 64 KB, 256 KB and 512 KB (16,384 bytes, 65,536 bytes, 262,144 bytes and 524,288 bytes respectively). It is also interesting to notice how the 64 KB window is by far the most popular, with approximately half the tests in the dataset. It is worth mentioning that the plot has been trimmed on the horizontal axis in order to be able to zoom in the relevant range, but window values span a much wider range than that displayed, although for an insignificant number of tests.

For a comprehensive picture of the ranges of window sizes and the distribution among tests in the dataset, table 4.3 shows the main summary statistics of the \textit{MaxRwinRcvd} variable. Interesting takeaways are (1) that the median window size falls on 64 KB, as expected since half the tests show that maximum window size; (2) that the mean of the maximum window size here is not a representative measurement, since it falls well above the 3rd quartile measurement, showing how outlier values have a great effect in mean values; and (3), that the maximum receive window advertised is 1 GB, which happens to be the maximum window size that TCP allows under its current specifications. There are 28 tests that at some point achieved a receive window of 1 GB.

After understanding the distribution of maximum receive window in the NDT dataset,
the goal is to characterize how much it affects the performance of TCP connection. Similarly to what we did in section 4.5, to observe the effect of the receive window on speeds we also need to keep an eye on the round-trip time that each test experienced since it plays a fundamental role. Figure 4-9 shows a scatterplot of the nearly 400,000 tests in the NDT dataset showing their average speed and round-trip times, colored by the maximum receive window they used during the connection. Even though the cloud of points is pretty disperse, we can clearly observe three thick lines of dots, corresponding to the maximum achievable throughput for each receive window. For the 512 KB window, there are not enough dots to form a distinguishable curve, but we can see how the purple dots cover the top-left part of the plot, and stop at a certain point, not reaching the corner.

The theoretical limit for throughput given a round-trip time and a maximum window
Table 4.3: Summary statistics for the MaxRwinRcvd variable, dataset containing global tests.

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min</td>
<td>49 bytes(^1)</td>
</tr>
<tr>
<td>1st Quartile</td>
<td>63.81 KB</td>
</tr>
<tr>
<td>Median</td>
<td>64.00 KB</td>
</tr>
<tr>
<td>Mean</td>
<td>400.78 KB</td>
</tr>
<tr>
<td>3rd Quartile</td>
<td>179.98 KB</td>
</tr>
<tr>
<td>Max</td>
<td>1 GB</td>
</tr>
</tbody>
</table>

size is given by the formula $\text{Throughput} \leq \frac{\text{RWIN}}{\text{RTT}}$ \[^{[43]}\]. This limit, proportional to the receive window size and decaying with round-trip time, could be seen in the previous plot. However, to re-confirm this theoretical model, we will pick a subset of tests with a given delay and Rwin characteristics and we will verify if they adhere to the model. The range of RTT values selected is 85-90ms, since that range is a pretty standard round-trip time range for tests in the United States, and it features a high enough number of tests. We have chosen to use the dataset with tests in the United States so that we can establish a cross-ISP comparison, using the same six top providers that we previously used in section 4.2.

Figure 4-10 shows the distribution of speeds for the six aforementioned providers, for tests with average round-trip times between 85 and 90ms. Also represented in the graph are the theoretical lines of maximum speeds achievable by tests with such delays, and with maximum receive windows of 16 KB and 64 KB. The lines for common bigger receive windows (256 KB and 512 KB) are not plotted because they do not fit in the speed range depicted in the graph. Observe how the median speed for each provider lies very close to the theoretical speed limit for the 64 KB window. That confirms our model, since the majority of the tests feature a 64 KB window, as we have seen in figure 4-8.

\[^{1}\text{Note that a value of 49 bytes as the maximum receive window advertised on the receiving side throughout the whole test does not make logical sense. This number hints that something was wrong with the test that featured that value. This is a good example of how in “real” tests, sometimes we can measure values that do not conform to the models or to the theoretical behaviors we expected.}\]
Figure 4-9: Scatterplot of speed vs round-trip time per test, colored by maximum receive window advertised during the test.

4.5 Study of congestion signals

One very relevant factor affecting today’s broadband performance is congestion. However, as important as it is, it is not a very well-understood or well-characterized phenomenon. NDT records which tests encounter congestion, how many times, and the type of congestion encountered. First of all, it is important to understand that the fact that a TCP connection encounters congestion is not bad on itself; on the contrary, it is the sign that the connection is working well. As we explained in section 2.1.1, TCP is designed to use all the capacity available in the network. So, even if a TCP flow is the only flow along a path, TCP will
grow the congestion window exponentially during the slow start period until it hits the limit (be it link bandwidth, buffer limit in an intermediate router, etc.) and then a packet will be dropped. A packet dropped will trigger duplicate acknowledgements from the receiving side, to inform the sender that it did not receive a packet correctly. Duplicate acknowledgements are one form of congestion signal, and they will cause the sender to reduce the congestion window by half and resend the lost packet. As we mentioned, this is the sign of a correctly-working TCP connection. If a connection never encounters congestion, it means that there is something else that is keeping it from reaching the limits of the network. This “something else”, is generally the receive window, as we have seen in section 4.4.

In a correctly working TCP connection, there is a rule that states that knowing the round-trip time and loss rate (frequency in which packets get dropped), we can know the
throughput of the connection, following the formula

\[ B(p) = \frac{1}{RTT} \sqrt{\frac{3}{2bp} + o(1/\sqrt{p})} \]

where \( B \) is the throughput, \( b \) is the number of packets that are acknowledged by a single received ACK (typically 2), and \( p \) is the probability that a packet is lost[6].

First of all, we can start by studying the proportions of tests whose performance was restricted by the network (i.e. limited by the congestion window), tests that were bounded by the receiver (i.e. receive window limited), and tests that were limited by both the receive and the congestion windows. For that purpose, we have analyzed the dataset with tests in the United States so that we can later look into specific ISPs like we did in the previous sections. We selected the tests which encountered congestion (i.e. had their CongestionSignals variable different from zero), resulting in the dataset congUSLine. We have also factored in the receive window and the parameter WinScaleRcvd, in order to tease out those tests that were not limited by the receive window or by a restriction in how much they can grow it. We will consider that a test was not limited by the receive window when the parameter SndLimTimeRwin was less than 5% of the duration, and the parameter WinScaleRcvd was greater or equal to 3. What those values mean, and the parameter WinScaleRcvd and the impact it has in the tests are described in greater detail in section 4.7.

Figure 4-11(a) shows a time series of the number of speed tests recorded in the NDT dataset, divided between those that encountered congestion and weren’t limited by the receive window (pink – these are the a priori well-functioning tests, bound in their speeds by the congestion window), tests that encountered congestion but were also limited by the receive window (green) and those that did not receive a single congestion signal and were therefore only limited by the receiver (blue). A first observation is that the tests only limited by the network (pink) are a minority. In order to better appreciate the magnitudes and evolution over time, figure 4-11(b) reflects on the same data, but normalized on a daily basis. We can observe how, on average, between 30 and 40 percent of the tests did not encounter congestion and were presumably limited by the receive window all the time, which is something undesirable. With regards to this category of tests, there doesn’t seem to be an evolution or pattern over the roughly seven months contained in the dataset. Regarding tests that were only limited by the congestion window, we do observe a change starting in...
mid-July, when the NDT test became more popular and the number of users running the test increased. Before mid-July, tests in this "good" category oscillated between 3 and 10% approximately, whereas after mid-July there is a big spike that lasts for approximately a month and a half in which they get to represent over 30% of the tests, with a maximum of around 55%, and starting at the end of August the level stabilized again but at a higher level than before, around 15%.

![Figure 4-11: Time series of NDT tests by the day, divided into tests limited by the congestion window, the receive window, or both.](image)

Another interesting thing to analyze is whether there is any temporal pattern if we group the tests on a 24-hour basis. Figure 4-12(a) shows the number of tests recorded by hour of the day, again divided into tests limited by the network (congestion window), tests limited by the receiving side (receive window) and tests limited by both. Note the huge variation in number of tests recorded depending on the hour of the day. The lowest hours in number of tests run from 6 AM to 12 PM GMT, which corresponds to 1 AM to 7 AM EST. Keep in mind that the dataset contains tests across all the United States and its four different timezones, and this same valley hours would be equivalent to 10 PM to 4 AM PST. On a normalized scale by the hour, as shown in figure 4-12(b), we again see how the percent of tests which did not encounter congestion (blue) is pretty stable across the day, at around...
a 35%, and how the tests that were only bound by the congestion window (pink) represent around a 15% of the tests on average across the day, except for a peak of three hours (from 8 to 11 AM GMT -2 to 5 AM EST) where these tests represent around a 25% of the total. Also note how in the hours immediately surrounding this peak the level of congestion window limited tests is somewhat higher than during the rest of the day. This corresponds to nighttime, and seems to shows us that during those times, the number of tests where TCP operates in its plenitude is larger.

![Figure 4-12: Time series of NDT tests by the hour, divided into tests limited by the congestion window, the receive window, or both.](image)

Next, we aim to study whether there is a time-of-the-day effect on speeds. That is, whether tests at certain hours of the day encounter the network more loaded and enjoy slower speeds, or whether this effect is not present in our data. For that purpose, it is important to analyze a homogeneous set of tests, since there are many factors that can alter speeds and mislead us when studying the presence of a time-of-the-day effect. If a test is limited in its speed by the receive window, we would not necessarily expect to find variation across the day, so first of all we will retain only tests that are not receive window limited and that have a WinScaleRcvd value greater than 3, as we have done before. One other factor that can alter average speeds recorded is the different populations present at different times.
of the day. For instance, if for some reason DSL users ran their speed tests in the morning while cable users ran them in the evening, we would find lower speeds in the morning, but this result would not imply anything about congestion levels in the network and its effects. So, to avoid population change effects, we have further filtered our dataset to keep only Comcast customers. We chose Comcast since it is the ISP with a larger number of tests recorded, and it only offers cable Internet (Verizon offers cable and DSL, for instance).

![Figure 4-13](image)

Figure 4-13: Time series analysis of speeds recorded for Comcast customers not bound by the receive window, on a 24-hour basis.

After performing the filtering described above, we are left with a dataset of 2,982 tests. Figure 4-13 shows the scatterplot for all test speeds recorded, plus the hourly averages, with the hours listed in GMT. We can observe that in hours between 6 GMT and 16 GMT speeds are higher. Those times correspond to 1 EST and 11 EST, or 22 PST to 8 PST. This seems
to show a time-of-the-day effect in this specific dataset, since nighttime hours show higher speed averages than daytime hours. It should be noted that the dataset contains tests from all of the United States, so we presume time-of-the-day effects in the East Coast and West Coast could somehow cancel out because of the time difference.

![Time series analysis of speeds recorded for Comcast customers not bound by the receive window, except for the user who ran most tests, on a 24-hour basis.](image)

**Figure 4-14**: Time series analysis of speeds recorded for Comcast customers not bound by the receive window, except for the user who ran most tests, on a 24-hour basis.

In the figure 4-13 we could observe another pattern, which is a stable line of tests at approximately 15 Mbps across the day. Studying this pattern, we realized that it is caused by a single user who ran the test repeatedly (probably using a script that ran it automatically). This user represents 1,820 tests of the total 2,982 in the dataset analyzed, and therefore we would expect this user to bias the hourly averages. Figure 4-14 shows the same data as before with one change: the single user who ran the most tests has been
removed. We observe how average speeds fall across the day, since that user was dragging averages closer to his own 15 Mbps recordings, and we also observe how the time-of-the-day effect here is slightly more noticeable now, since the user that we removed showed constant speeds that contributed to stabilize the speed average across the day. It should be noted, however, that between 9 and 12 GMT there are less than 20 data points per hour, and therefore the statistics computed here lack the minimum level of significance. This would be an interesting experiment to repeat with a larger data sample. Nevertheless, this is quite an unexpected picture that we do not yet fully comprehend. Being such a uniform set of users (all have Comcast cable connections and are only network limited), we expected to see a concentration of points around the 12 Mbps level or, in any case, not such a big variance in speeds.

So far, we have considered that a test encountered congestion as soon as it the sender received at least one congestion signal but this division neglects the number of congestion events that tests encountered, and that is an interesting aspect to study too. Congestion events include fast retransmits, timeouts, sender stalls, and ECN (Explicit Congestion Notification), among others. The number of congestion events that a speed test encounters are going to affect its outcome, although under normal working conditions, tests should be receiving congestion signals at periodic intervals as part of the congestion window additive increase/multiplicative decrease mechanism described in section 2.1.1. In order to have an idea of how much congestion tests encounter, table 4.4 shows the quartile statistics and mean value for the CongestionSignals variable, which counts the number of congestion events experienced by a test. Figure 4-15 shows the complete distribution of number of tests by number of congestion signals. Note how even though some tests feature a very large number of congestion signals, three fourths of the tests experience congestion four or less times during the test.

The next logical step is to dissect the data and understand the different types of congestion. The NDT dataset contains different variables that record certain types of congestion signals and the mechanisms that TCP offers to deal with them. The most relevant variables, and the ones that we are going to study going forward are the following:

- **FastRetran**, indicates the use of the Fast Retransmit algorithm. When a segment
<table>
<thead>
<tr>
<th>Statistic</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min</td>
<td>0.000</td>
</tr>
<tr>
<td>1st Quartile</td>
<td>0.000</td>
</tr>
<tr>
<td>Median</td>
<td>1.000</td>
</tr>
<tr>
<td>Mean</td>
<td>4.016</td>
</tr>
<tr>
<td>3rd Quartile</td>
<td>4.000</td>
</tr>
<tr>
<td>Max</td>
<td>3458.000</td>
</tr>
</tbody>
</table>

Table 4.4: Summary statistics for the CongestionSignals variable, tests in the United States.

gets lost and doesn’t reach its intended recipient, the TCP sender generally waits for a specified timeout value after which it will assume the packet is lost and will retransmit it. However, under the Fast Retransmit algorithm, if the sender receives three duplicate acknowledgements with the same acknowledgement number (for a total of four acknowledgements with the same number), it will assume that the packet was lost and will retransmit it without waiting for the timeout counter to expire.

- **Timeouts.** Counts the number of times that the retransmit timeout has expired, making the TCP sender retransmit the lost packet.

- **SubsequentTimeouts.** Counts the number of times that the retransmission timeout has expired right after a previous timeout. That means, after a timeout and retransmitting a packet, that packet got lost again and the timeout counter expired again.

- **AbruptTimeouts.** The number of timeouts that occurred without any immediately preceding duplicate acknowledgments or other indications of congestion. Abrupt Timeouts indicate that the path lost an entire window of data or acknowledgments.

- **SendStall.** Contains the number of times that the sender didn’t transmit any information when it could have. A common cause for those is that the sender did not have enough information ready to send, so it waited to receive more data from the application layer in order to assemble a segment and transmit it. In a designed experiment like the NDT speed test, sender stalls should be almost non-existent, since the design of the test is such that the sender is supposed to always have information to transmit, as the goal of the test is to send as much data as possible over 10 seconds.

- **OtherReductions.** Includes any other reductions of the congestion window caused
It is important to understand how common each of those congestion signals is, in order to get a better idea of what Internet congestion looks like and what kinds of losses are impacting broadband performance. Table 4.5 shows the count of tests within the dataset featuring each of the aforementioned congestion signals, as well as tests showing more than one of those signals. The main takeaway from looking at the table is that when it comes to congestion, fast retransmits are the norm. We see that a total of 44,908 tests (or 97.71% of the 45,962 tests that encountered congestion) went through the Fast Retransmit algorithm at some point. This is a sign of a good health in TCP connections: under ideal functioning, a TCP connection in its congestion avoidance phase (once slow start is over) is characterized
by the repetitive sawtooth pattern of increasing the congestion window until a packet is
dropped, at which point using the Fast Retransmit algorithm the sender receives three
duplicate acknowledgements, realizes that the packet has been dropped, and resends it and
reduces the congestion window by half. This comes to say that when no other circumstances
apply, a “healthy” TCP connection would encounter Fast Retransmit congestion signals at
regular intervals, and this is what the data seems to show. It should also be noted that of
the tests that recorded fast retransmit congestion signals, approximately one third (precisely
15,561) only show fast retransmits, and no other signals of congestion.

It is also noteworthy that timeouts are common but only in combination with other
congestion signals, primarily fast retransmits, but timeout signals in their different forms
alone are rare. Also worth noting the minuscule number of sender stalls recorded (6 total),
which is a sign that the testing infrastructure was well-built in this regard, since the senders
always had information to transmit during the test. That is important since frequent sender
stalls would have introduced a source of bias in the measurements, probably leading to lower
speed measurements.

<table>
<thead>
<tr>
<th>Congestion variables</th>
<th># tests</th>
<th>% of total</th>
</tr>
</thead>
<tbody>
<tr>
<td>FastRetran (alone)</td>
<td>15,561</td>
<td>33.86</td>
</tr>
<tr>
<td>Timeouts (alone)</td>
<td>7</td>
<td>0.02</td>
</tr>
<tr>
<td>SubsequentTimeouts (alone)</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>AbruptTimeouts (alone)</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>SendStall (alone)</td>
<td>3</td>
<td>0.01</td>
</tr>
<tr>
<td>OtherReductions (alone)</td>
<td>1</td>
<td>0.00</td>
</tr>
<tr>
<td>Combination of timeout variables (alone)</td>
<td>614</td>
<td>1.34</td>
</tr>
<tr>
<td>FastRetran (+ others)</td>
<td>44,908</td>
<td>97.71</td>
</tr>
<tr>
<td>SendStall (+ others)</td>
<td>6</td>
<td>0.01</td>
</tr>
<tr>
<td>Any timeout variable (+ others)</td>
<td>12,285</td>
<td>26.73</td>
</tr>
<tr>
<td>Any timeout variable + Fast Retrans</td>
<td>11,236</td>
<td>24.45</td>
</tr>
</tbody>
</table>

TOTAL NUMBER OF TESTS 45,962 100.00

Table 4.5: Number of tests featuring the different congestion variables. Tests in the United
States that experienced any congestion signal.

Finally, it is important to analyze whether we can characterize the effect that congestion
events have on throughput. For that purpose, we believe the correct way to look at the data
is considering its round-trip time too, since RTT plays a fundamental role in the speeds
of connections, as we saw in section 4.1. In [6], the authors build a mathematical model that claims that for a given RTT, the more packets dropped the lower the speed achieved. We wanted to validate that using our dataset. Figure 4-16 reflects that statement clearly. We can observe the curves that form the tests showing how speed declines exponentially with delay. At the same time, we see how for a given RTT, as the number of congestion signals goes up the speeds go down. Take for example the vertical line for an RTT of 50ms. Starting from the bottom, the line crosses a cloud of dark red dots (tests with over 30 congestion signals), followed by a red cloud, then orange, yellow, light green, dark green and finally purple, which represents the tests that did not encounter congestion.
course, it is not a completely continuous picture and there is a certain overlap of points with different colors, but the overall trend is clear. The smaller purple/green curve that crosses the orange cloud corresponds to the tests that encountered 0, 1 or 2 congestion events, but had a smaller receive window, while the main purple curve corresponds to the most common receive window used (64 KB) and the outermost purple curve corresponds to a bigger window, as we will see next. The effect of receive window in speed is explained in section 4.4. What is very clear from this plot is that tests with a significant number of congestion events show a poor performance.

Figure 4-17: Scatterplot of speed vs round-trip time per test, colored by number of congestion signals encountered, and divided in two plots based on whether the test encountered congestion at all or not.

Figure 4-16 contains a lot of information, since it shows the nearly 400,000 tests contained in the NDT dataset. It is worth giving it a more detailed look. Figure 4-17(a) shows only the tests that did not encounter congestion (CongestionSignals = 0), colored by the maximum size of the receive window recorded during the test. In this figure showing tests that were presumably receive window limited all of the time, we can again see the huge effect that the receive window size has on speeds. Figure 4-17(b) shows the rest of the tests (tests that did encounter congestion), colored in the same scale used before, which reflects the number
of congestion events that the tests encountered. Tests with zero congestion events is the largest single category, as we saw in figure 4-15, so removing the dots for tests with zero congestion signals (plotted in purple in figure 4-16) from the plots allows us to (1) observe the curves corresponding to the different receive windows for tests that were only receive window limited, and (2) see more clearly the “rainbow” formed by tests featuring congestion events and how performance is degraded by congestion.

4.6 The effect of test durations

One parameter of key significance for a speed test is its duration. Duration is a design parameter chosen by the designer of the test, and generally speaking there are no restrictions on its value other than practicality (i.e. a web speed test cannot last two hours, most users would not let it finish). However, duration does have a big impact on the outcome of the test. On one extreme, if we measured speed over a very short interval of time, we may be measuring the speed at which an individual packet is sent, which for instance is something around 38 mbps in DOCSIS 2.0[11]. Short bursts of packets can also be sent closer together than would be implied by the configured speed advertised by a provider[45]. On the other extreme, one could compute speed based on the whole duration of a TCP connection, which would include inactivity periods in the middle. Clearly both cases would result in very disparate measurements, and would carry little meaning for characterizing the speeds that the customer enjoys.

Many cable broadband networks feature a characteristic called Powerboost, which offers higher speed during the first few seconds of the connection than during the rest of the connection, while in DSL this does not happen. Shorter tests will reflect this increased speed whereas longer tests will minimize its effects, and therefore cable operators have an incentive to push for shorter tests. Cable providers argue that short tests measure the speed that is most relevant to the customers. Studies show that 90% of all websites are smaller than 663.19 KB[46], and that is a small enough volume that it can be perfectly loaded within the initial high-speed interval that cable networks offer. Therefore, even if the Powerboost speed is only available to the customer for a few seconds at the beginning of the connection, that speed covers almost the totality of the user connections. On the other hand, DSL operators have an incentive to push for longer tests, since as test durations increase, the
effect of Powerboost in the average speed results is reduced. They argue that long durations are the only ones that allow a test to measure the sustained speed that a user will experience in his connection. That is relevant for long TCP connections where a large amount of data is transmitted. For instance, in large file transfers the amount of data transferred greatly exceeds the Powerboost capacity. And, to a certain degree, sustained speed is important for video streaming (eg. an average Netflix two-hour movie encoded at 2000Kbps is equivalent to roughly 1.8GB of data[47]), although Powerboost plays a role during the buffering period as well. Moreover, long-term speed is the only way to assess if an ISP is offering the speed it promises in its advertising.

In general, when one thinks of factors that influence the results of a speed test, variables that come to mind are usually round-trip time or network load. It would be very rare to think about duration as having a significant effect but, as we will show next, empirical evidence proves that it does. To prove it, we set up an experiment under controlled characteristics to obtain some valuable information. The main traits of the experiment are the following:

- Tests ran from one pre-determined client to a server running the NDT platform that we installed in our lab.

- The client was a subscriber of a Comcast cable modem connection, with an advertised download speed of 12Mbps, an upload of 2Mbps, and Powerboost to 15Mbps.

- The client ran multiple tests per hour, with durations of 5, 10, 15, 20, 25, and 30 seconds.

- Since both client and server are controlled by us, we ensured that receive window limitation is small to non-existing. Therefore tests are congestion window limited most of the time.

- Since client and server are always the same, round-trip times are fairly constant between 15 and 20 ms. Hence, RTT does not play a big role in the measurements, with few exceptions.

Figure 4-18 shows the result of those tests, where average speeds are reported over a nine-day period using the same measurement algorithm, colored by the duration of the test. As we can observe, speed drops rapidly as test duration increases. The median speed for
Figure 4-18: Average speed recorded per test, plotted according to the date that the test was performed, and colored by the duration of the test.

5-second tests is approximately 23.1 Mbps, while that of a 10-second test is 19.7 Mbps, on the order of a 15% decrease. Subsequent five-second increases in test duration have smaller but still significant effects, such as roughly an 11% additional drop in speed when increasing duration from 10 to 15 seconds. Overall, when increasing the duration from 5 seconds to 30 seconds, the median speeds measured decline from approximately 23.1 megabits per second to 15.2, for a drop of over 34%.

Another interesting thing to notice in the aforementioned plot is the variability in 5-second tests. While tests in all other durations are fairly constant, with a few exceptions, 5-second tests do not form a line; instead, a cloud of points appear scattered between 20 and 26 Mbps. The reason for this high variance is the fact that for such short durations, the outcome of the test depends heavily on how long the test stayed in the slow start phase, that is, when it first encountered congestion, if it did at all.
Since, as we mentioned before, RTT is fairly constant, we can presume that in this case a relevant factor impacting the performance measured in the tests is congestion. Figure 4-19 shows a scatterplot of tests characterized by their speed and duration, and colored by the number of congestion signals encountered during the test. The plot continues with the trend seen in figure 4-18, where shorter tests reflect higher speeds. It is apparent how for a given duration, tests that encounter more congestion events show slower speeds, as we had previously seen in section 4.5. It is also apparent how tests with longer durations seem to encounter more congestion events, which is also something we expected. Note the high variation in speeds in 5-second tests that did not encounter congestion signals (red dots). The RTT of all those tests is roughly the same and, since they did not encounter congestion, that means they did not exit slow start during the 5 seconds of the test. Hence, the variation in speeds that we observe could be a time-of-the-day effect.

Figure 4-19: Average speed recorded per test, plotted according to the duration of the test, and colored by the number of congestion events encountered.
Theory says that, all else equal, and particularly for tests that by design are not receive window limited such as the ones we are currently analyzing, for the same speeds and round-trip times, given the duration of the test we can determine how many congestion events the test encountered. That is almost like saying that the number of congestion events per second for those tests is constant. We wanted to explore whether that statement is reflected in our experimental dataset. Figure 4-20 shows the histograms of number of congestion signals per test, divided into sub-plots by the duration of the test, which is specified at the heading of each sub-plot. Roughly, we do observe how the distribution of number of congestion signals increases as duration increases. Furthermore, it seems to increase proportionally: 5-second tests show between 0 and 5 congestion signals, 10-second tests show between 5 and 10, and for the subsequent groups the proportion seems to continue, where an addition of five seconds on the duration of the test seems to increase the center of the congestion signals distribution around 5 units. This is true except for the last group, the 30-second tests, where the majority of energy of the distribution falls between 30 and 35 congestion signals, where we would expect it to be between 25 and 30. We also observe how, the longer the duration, the more spread the distribution.

The facts that congestion signals increase proportionally with duration, and that we know that RTT is fairly constant, lead us to the conclusion that those two factors do not play an important role on average in the outcome of these tests, and seem to support our hypothesis that the main reason why speeds decrease with the duration of the test is that the effect of Powerboost dissipates.

4.7 Measuring speeds not limited by the Operating System

One very relevant question for both policy and commercial purposes is being able to measure the service level offered by providers. As we saw in section 4.4, a parameter that depends on the Operating System (OS) and has nothing to do with the provider’s network has a huge impact on speeds. Therefore, it would be unfair to blame the providers for low speeds caused by a deficient management of the receive window. Nevertheless, we realize this is a very important issue, and we have made it a priority of our research to come up with a methodology that allows us to measure the actual capacity provided by the ISPs, teasing out tests that encounter other limitations not caused by the network itself.
We have found that older operating systems such as Windows XP don’t do a good job managing the receive window; they advertise smaller sizes than would be feasible, causing the connection speeds to drop. This issue has been solved in newer versions of Windows such as Vista and 7, and similarly it is not an issue in OSes like Linux and Mac OS X. However, back in 2009 when our data was captured, Windows XP represented a very significant proportion of the world OSes, and it is still the dominant OS even as of April 2011[48].

Our goal was to identify which Web100 variables we could use to discern the tests that had been limited by cause of their operating system’s management of the receive window. The first variable that comes to mind is SndLimTimeRwin, which indicates the time that the sender has been limited by the receive window of the receiving side. Since we are looking

Figure 4-20: Histogram of the number of congestion signals per test, for a given test duration. Each of the sub-plots contains the histogram of tests for a certain duration, specified at the heading of the sub-plot.
at download tests, the receiving side is in fact the client, which is what we are interested in measuring. NDT servers run Linux, and therefore if we looked at upload tests, receive window limitation would not be an issue.

But $SndLimTimeRwin$ is not the only signal we have to detect what we can call "OS limited tests". An important TCP parameter is the TCP Window Scale Option, which is one of the TCP extensions for high performance added in 1992. The Window Scale Option is a three-byte field sent in a SYN segment during the establishment of a TCP connection which contains the scale factor to be applied to its receive window\[49\]. When using the Window Scale Option, advertised receive window sizes grow dramatically, from the previous 64KB to up to 1GB. Therefore, this becomes a critical parameter in that a receiving side not able to advertise big windows will not be able to achieve high download speeds. The TCP Window Scale Option needs to be negotiated by both parties, and in our NDT dataset is recorded in the variable $WinScaleRcvd$. This variable contains the number of positions that the original window can be left shifted to obtain the new size, up to a maximum value of 14. The TCP Window Scale Option is only enabled by default in Windows Vista, Windows Server 2008 and newer, and is enabled by default in all versions of Mac OS X and in Linux kernels from 2.6.8 (August 2004) and newer\[50\].

In order to filter out the "OS limited tests", we have come up with the cutoffs of being receive window limited more than 5% of the duration of the test, or using a Window Scale Option smaller or equal to 3. After filtering those tests out, we believe we are left with a dataset that does not contain "OS limited tests", so that we can observe the real network limitations. Note that the not limited dataset only contains 9,636 tests of the total 71,618 tests recorded in the United States, which reinforces our previous statement that most tests are still receive window limited. Figure 4-21(a) shows the distribution of speeds for the top ISPs in the United States, for tests "not OS limited". For comparison purposes, figure 4-21(b) shows the same distribution of speeds for all tests, including those "OS limited" and not (recall this is the same figure we saw in section 4.2).

Note how speeds have significantly gone up. This comes back to our point from section 4.4 on how significant the impact of receive window on speeds is. We see how tests where the OS manages the receive window effectively show higher speeds, with more than noticeable increases in providers such as Comcast (from roughly over 10 Mbps to 15 Mbps), Cox
Figure 4-21: Distribution of speeds and average round-trip time per ISP, top ISPs. The subfigures compare tests not “OS limited” (tests limited by the receive window 5% or less of the connection, and with a WinScaleRcvd option greater than 3), with the distribution for all tests. Blue dots represent the average speeds of each provider.

(from 8 to 12 Mbps), or Verizon (from roughly 3 to over 6 Mbps). Furthermore, here we are comparing a subset of “not OS limited” tests to the total US dataset, which includes the tests in the subset. To appreciate clearly how significant this effect is, we ought to compare “OS limited” tests with “not OS limited” tests. Figure 4-22 shows the comparison of distributions of speeds between limited and not limited tests.

In this case, we see even larger differences in Comcast, now jumping from 9 to 15 Mbps, and to a smaller degree in Cox, which now shows median speeds of approximately 7.5 Mbps instead of 8 Mbps in the plot showing all tests. For the other ISPs, the difference in speeds between the dataset with all tests and the dataset with only limited tests is negligible. This is explained by the observation we made earlier, that only 9,636 tests belong to the “not OS limited” category. This leaves 61,982 tests in the limited dataset, a number very close to the total 71,618 total US tests. Considering that 2,990 of the 9,636 tests that are “not OS limited” belong to Comcast, this means the number of non-limited tests in the other ISPs is relatively small, and does not affect much the distribution of speeds of the combined dataset.
Figure 4-22: Distribution of speeds and average round-trip time per ISP, top ISPs. The subfigures compare tests not “OS limited” (tests limited by the receive window 5% or less of the connection, and with a WinScaleRcvd option greater than 3), with tests “OS limited”. Blue dots represent the average speeds of each provider.

The main takeaways from this section are:

1. We have come up with a methodology that we believe accurately measures the speeds that customers obtain when only limited by the network and not by their own operating system. This way of filtering tests opens the door to perform many other analyses to our data in order to obtain valuable insight on the network infrastructure itself, eliminating the bias introduced by client-side operating system limitations.

2. Speeds shown in the previous plots for “non-OS limited tests” are relatively high, and reasonably correspond to the service levels we would expect from ISPs (with the exception of Verizon that mixes DSL and FiOS customers). This conclusion is significantly disparate from previous claims by the FCC[34] and a widely accepted vision that ISPs are offering much less speed than they advertise.
Chapter 5

Policy implications of these analyses

5.1 Relevance of broadband measurements

Broadband measurements play an important role in the Internet arena. Speed is the main variable used to characterize Internet connections; it is used for both commercial and policy purposes. From a commercial point of view, it is the number that providers use to advertise their services and to differentiate themselves from their competitors, by claiming to offer a better price-to-speed ratio. In the policy arena, speed numbers are oftentimes stated as goals for new policies. Policymakers need to base their decisions in facts and numbers that represent the state of the connections; those numbers are the different measurements that certain companies and research institutions take on the network. However, there is a concern that policymakers may be using the wrong measurements to make their decisions, and this could have bad consequences. Examples have already been mentioned throughout the current thesis, and they include:

1. **Investment in infrastructure**: sometimes decisions on where to invest in building or renewing the telecommunications infrastructure are based on plain summary statistics such as speed averages measured, or even worse, on speeds advertised by the ISPs. It is important to understand where the numbers come from and what factors have influenced them. An example is the case of Ireland where Forfás, Ireland’s policy advisory board for enterprise, trade, science, technology and innovation, based its decisions on investment in telecommunications infrastructure on the speeds advertised by providers[51].
2. **Universal service:** the FCC's National Broadband Plan states the goal "to provide 100 Mbps connections to 100 million people by 2020"[52]. However, it does not specify how those requirements will be measured, and indeed it makes a call to develop methodologies for accurate measurements.

3. **Level of service provided by the ISPs:** The FCC accused the ISPs of providing much less speed than advertised[34], but issues were raised later on the quality of the data that the FCC used to make its claim[38]. This case highlights the necessity to have reliable ways to characterize the service level offered by ISPs.

   Hence, having different entities using different methodologies to measure speeds, or using those speed quotes without understanding the circumstances under which they were taken, may have very significant effects on both market competition and government regulation and investment in the broadband networks arena.

### 5.2 Possible enhancements in NDT testing procedure

The Network Diagnostic Tool represents a big improvement in the broadband measurement space, both because it provides all the source code and raw data collected openly and because of the large number of tests collected. It is an unprecedented data collection that allows us to have a better insight on how our Internet operates. However, there are some things that could be done in a different way and result in a more comprehensive and useful set of data to analyze. As part of the current thesis' work, we wanted to make some recommendations in changes to NDT that we believe would be beneficial.

**Number of TCP connections.** NDT runs the speed test over one TCP connection, whereas other tests such as Ookla/Speedtest.net allows to specify the number of connections used, and can use up to eight parallel HTTP threads for the measurement[11]. As we have mentioned previously, both numbers of connections make sense depending on the type of service we are trying to characterize: for instance, one TCP connection will be representative of FTP transfers, while multiple TCP connections will be representative of web browsing. Moreover, it has been shown that in high bandwidth-delay environments, a single TCP connection will only be able to achieve a throughput close to link speed if the loss rate is extremely low, making it impossible to fill the pipe with a single TCP connection[16], and it
would be useful to have empirical data to analyze that phenomenon. Therefore, we propose that the NDT test should allow to select the number of TCP connections used for the speed test.

**Test duration.** As seen in section 4.6, the duration of a test has a big impact on its outcome in terms of speed. Short tests tend to reflect higher speeds since they are measuring the Powerboost interval in cable connections, while longer tests dilute the effect of Powerboost and show lower speeds, corresponding to the sustained, long-term speed that the connection offers. Both are relevant and both are worth being measured, that's why we suggest that NDT offers different test durations. One possibility to gather a good amount of data causing the least hassle to the user would be to run one 30-second test and report on the speeds at different points of the test. With a long duration test like that, researchers would be able to obtain the speeds at any time point they wish, such as at the 10, 15, 20, 25 and 30-second marks. For reporting to the consumers, probably the 10 and 30 seconds results would be enough.

**Geographic location of the client.** As the test is designed right now, no geographic information of the client is recorded. For our analysis, we used online lookup services to obtain more information, using the client IP addresses. This, however, is of limited effectiveness since the free online lookup services are not accurate enough. Online geolocation services are very accurate on a country level (around 99% accuracy), but when trying to identify regions or cities, their accuracy drops to levels between 50 and 80%[53, 54, 55]. To avoid error, we only considered their information on a country level, but it would be interesting to have more fine-grain information on the location of the client. It would also be interesting to add time-zone information to the dataset, so that time-of-the-day analyses would be more accurate.

**Selection of NDT sever.** The current implementation of the NDT test uses the DONAR DNS system to choose the server that is closest to the client by default, based on geographic location and platform load distribution[27]. We believe this is an appropriate design for most of the cases and is the least burdensome for users. However, we believe that including an option allowing the client to choose the destination NDT server would give much more flexibility to the test. This would open doors to perform different types of analyses;
analyses that may be less relevant for the average user of the test, but that could be of great interest to providers as well as researchers and other network professionals. For instance, by being able to select a test server in Europe while running the test from the US, one could study the performance of trans-continental TCP connections. This, coupled with the next recommendation, would give us the possibility to analyze cross-country connectivity levels.

**Increased number and distribution of servers.** One big limitation in the current M-Lab/NDT platform is the number and geographic distribution of test servers. As seen in section 4.1, round-trip times play a very important role in the outcomes of the tests, and as explained in section 2.2, M-Lab servers are concentrated in the United States and Europe, plus one location in Australia. If we aim to use the measurements collected through the NDT speed test for policy purposes, with the current infrastructure those results would only be valid in regions with nearby servers. In order to make the measurements universally valid the number of M-Lab servers hosting the NDT test needs to increase dramatically, and the servers need to be uniformly distributed across the globe, as Vint Cerf requested in his speech for the New America Foundation[2].

### 5.3 Policy recommendations

After performing our analysis of the data and understanding the role that the different stakeholders play, the main recommendations we came up with are the following:

**Increase transparency.** There is no set of test conditions (duration, distance from server, etc.) that is equally valid for all the users under all circumstances. Therefore it becomes critical to do an exercise of disclosure of the test conditions prevalent in a certain test. This applies to Internet measurement companies such as comScore or SamKnows. Those companies should be as transparent as possible with the testing methodology they use, and disclose as well the external relevant factors that may affect the test outcome (e.g. is there just one central server or are there many servers geographically distributed?). It also applies to policymakers, who should try to inform of the sources of the data they use for their decisions, and the technical details of the data collection as well.
Mistrust summary statistics. Throughout chapter 4 we have seen how a single summary statistic, such as average speed, has little meaning if it is not put in context. Run a speed test to a server located next door and then run the same test to a server located thousands of miles away, the speed measured won’t seem to belong to the same connection. Hence, our recommendation for policymakers is to not rely on a summary statistic without understanding the context that comes with it. It is paramount to know how the measurements were taken. This follows from the first recommendation, since for policymakers to consider the other factors in a measurement, Internet measurement companies need to disclose those first.

Advertise short and long run speeds. Since both short-term and sustained speed matter for different users, why pick one? We suggest measuring both types of speeds, and presenting this information to the consumer, whether it is an individual residential consumer or a policymaker. This way, different users will be able to pick based on their different needs.

While reporting two different speeds may be confusing to consumers at first, it is a much more informative approach that would give more complete information both to consumers and policymakers. In addition, a similar approach has proven to be effective in the case of cars’ gasoline consumption: manufacturers offer numbers of the fuel efficiency both in the city and on highways. Therefore, we believe Internet providers should be compelled by regulation to advertise real, measured, short and long-term speeds to their customers, instead of the current "up to" speeds.

Independent, open and verifiable measurements. We have mentioned how the competition is fierce between the different providers to push for the test parameters that would favor them over their competitors. On that regard, this thesis proposes an independent, third-party assessment of the speeds, which would assure a common measurement procedure across providers. For that purpose, it is important the role that academic and research institutions like MIT play. By having an independent party design the test methodology, pick the short and long test durations, and use the same publicly-available algorithm for all providers, we can avoid each party picking the numbers that favors them over the rest and we can ensure that anyone can verify the procedure. This would give us a fair way to compare across the different actors. We want to lead by example, and so we are making
available all the source code used for the analyses performed in this thesis so that anyone who is interested can review our approach and verify our methodology. The source code can be found in http://hdl.handle.net/1721.1/62812.
Chapter 6

Conclusions and future work

6.1 Concluding remarks and contributions from the current work

As important as gathering data is being able to understand what the data means. In the past, data analysis was characterized by a lack of data and it focused on how to generalize and extrapolate based on the available information, obtaining results that would be valid for the general population. In the recent years the paradigm has shifted, and nowadays the situation is diametrically opposite: in many disciplines there is a significant overabundance of data, and the focus now is on which of that data is relevant to include and which should be left out, and what is the methodology to follow for its analysis.

The overarching goal of the current thesis has been to lay the foundations for the study and interpretation of a rich dataset such as the NDT dataset. We hope to have developed a set of guidelines on the correct and incorrect ways of looking into this data, and we make our source code publicly available at http://hdl.handle.net/1721.1/62812, so that others can see our approach and build on top of it. We believe that the current work has contributed to further the understanding of broadband performance in different ways:

**Identifying and measuring the impact that different factors have on speeds.**

Through our analysis, we have singled out and studied the relevance of factors such as distance to the measurement point (round-trip time), users running the test repeatedly, erroneous buffering practices, network congestion, receive window, or test duration. We advocate for a more complete reporting of measurements, instead of simply reporting a
summary statistic.

**Procedure to measure network speeds.** In section 4.7 we have outlined a procedure to measure the speeds achievable when the connection is only constrained by the network, removing constrains imposed by the operating system on the client side. We achieve this by filtering out tests that have been receive window limited more than 5% of the time, or whose variable WinScaleRcvd has a value of 3 or less, which impedes the receiver to advertise large enough window sizes. We believe this is a fair procedure to quantify the speeds offered by ISPs, and therefore is very relevant in the broadband policy sphere. We have also pointed to cases in which it is appropriate to aggregate the tests that a same user has done, in order to avoid large users generating biases in the results.

**Measuring the service provided by ISPs.** This relates to the previous point on our proposed procedure to measure network speeds, since we have used our new procedure to filter out tests that are limited by the client operating system and have divided the remaining tests based on their ISP. Previously, the FCC had claimed that ISPs were offering much less speed than advertised[34], and this idea was broadly accepted. However, our analysis’ results are contrary to that statement, showing that the speeds offered on average by the providers are in fact close to what would be expected by their advertisements. Moreover, the speeds we have measured with our proposed methodology are similar to the preliminary results obtained by the FCC through the measurement boxes deployed by SamKnows, which don’t suffer from OS limitations.

**Enhancements to the NDT testing procedure.** While the Network Diagnostic Tool is a well-thought and built tool that is helping gather an unprecedented amount of data rich in technical information about the state of our connections, we believe certain changes would make it an even better tool, a tool with more impact in worldwide competition and regulation, as we have pointed out in section 5.2.

**Policy recommendations.** One of the main deliverables of the current work is to make concrete recommendations for the different players in the broadband field. In section 5.3, among other recommendations, we ask for policymakers to pass regulation that asks ISPs to advertise both short and long-term measured speeds, as well as for the establishment of
an independent, standardized, measurement procedure that could reflect on the providers under equal conditions.

**Contribution to the education on the topic.** On the task of increasing awareness and understanding on how the Internet operates, the current work has served as input to design a team project for the class “ESD.864J - Modeling and Assessment for Policy”, taught at the Massachusetts Institute of Technology. The framework and tools developed as part of the current thesis (including data, source code and plots and statistics) have been used to build a policy-related research question in which students (mostly first-year students in the masters in Technology and Policy at MIT) need to combine data from the NDT dataset with other sources such as the Ookla/Speedtest.net or the NTIA dataset and come up with policy relevant questions that they try to answer.

### 6.2 Future lines of action

The present document represents an introductory approach to the analysis of large datasets on technical measurements in broadband networks. While we have conducted a good number of analyses, described in chapter 4, and made some contributions to the field, summarized in section 6.1, there is some work that we couldn’t include in the current thesis due to time constraints. Following are some of the ideas for further research that builds on top of the foundations developed in this thesis, and that we believe are worth pursuing and we expect will bring interesting contributions to the topic:

**Further analyze “non OS limited” tests.** In section 4.7 we outline a methodology to filter out tests which were limited by the operating system on the client side, which managed the receive window poorly. The remaining tests, we believe, are a good way to analyze the broadband network itself, clean of client-side distortions. We did an initial analysis of speeds per ISP, for tests “not OS limited”, but this filtering methodology opens the door to further analyses that could help us understand how the network operates, and have a better picture of congestion. In our dataset, after removing the “OS limited” tests, we are left with a relatively small subset of 9,636 tests (in the United States), which is still significant but does not allow for much more subsetting. However, we propose applying this methodology to the BigQuery database (see the line of action below), in order to have a much larger
Extend the analysis to **BigQuery**. For our study, we have used the data on NDT tests hosted in Amazon Webservices. The dataset contains approximately 400,000 tests spanning the period between February 2009 and October 2009. This is a very large dataset, containing one of the most extensive samples on the topic, but it is still several orders of magnitude smaller than the full collection of NDT data available. As of July 2010, Google was hosting in its storage servers over 97 Terabytes of data corresponding to over 22 million NDT speed tests. That is an amount of data that would not be manageable for an on-site analysis like the one conducted for this thesis, and consequently Google is offering a web service that allows to run interactive analysis of huge datasets. The service is called BigQuery, and can execute a query against the whole dataset of NDT tests in approximately 40 seconds[21].

Obviously, such a large number of tests (on the order of 55 times larger than the dataset analyzed for this thesis), would allow for more accurate and generalizable analyses of the data. For instance, it would allow to further subset the data into different categories based on the values of many variables, and still keep a significant number of tests in the subset so that the outcome of the analysis can be meaningful.

In order to be able to use the existing code written for the R statistical package with the BigQuery webservice, a collaborator at the Advanced Network Architecture group wrote a library for R called **RBigQuery**, which serves as an interface between R and BigQuery. Thanks to that interface, and with some learning of the basic query syntax and variable names for BigQuery, the analyses conducted in this thesis could be repeated for the whole NDT dataset hosted in Google storage.

**Build a testbed for further experiments.** During our analysis of the data, we have come across some results that we didn't manage to understand or to find a logical explanation for them. Sometimes we expected to see some trend in a plot, and the trend was not there. In such a large, uncontrolled study such as the NDT speed test, there are many variables that can influence the results, and sometimes it is difficult to tease them out. Hence, we propose building tailor-made experiments where the researcher knows and controls the circumstances under which the tests are recorded, and therefore the variable or variables of interest can be observed more clearly, without external interference. As an example, in the current thesis
we already built one custom experiment to test the influence of test duration on speeds (see section 4.6). In that experiment, with a client and server that we controlled, we eliminated the variability introduced by differences in round-trip times or by external limitations such as the receive window. With those variables out of the game, the effect of duration could be explored more thoroughly. We suggest that for future work in the area, similar experiments where other variables are explored would be valuable, for instance to understand congestion better and find patterns that would allow us to distinguish between self congestion and shared congestion.

**Analyze all 5 millisecond logs in each test.** As explained in section 2.2, during an NDT speed test the values of a large set of variables are collected and saved every 5 milliseconds. For our analysis, we have only used the last log line for every test, since the last line contains many significant values such as minimum and maximum values, counts, totals, etc. Nevertheless, studying all the log lines of a test would give us some insight that we cannot obtain from just looking at the last log line. For instance, we could characterize the evolution of the receive window throughout a connection, the evolution of round-trip times, bytes sent, or understand better the effects of congestion by plotting congestion signals with speeds and receive window over time of the connection. This type of analysis would require the use of raw data not present in the NDT dataset that we used for our research, and therefore such data should be obtained from Amazon Webservices or, even better, processed through BigQuery.
Appendix A

Data cleaning

This Appendix is an extension to section 3.2, and it elaborates on the methodology used to filter and subset the initial dataset. The main transformations performed to the original data are the following:

A.1 Adding 'whois' information to each test

The dataset containing the original unprocessed data is loaded to the variable lastline, and contains the observations of 114 variables through 398,529 speed tests. Those variables, described in detail in section 2.3 contain a wide variety of technical information about the tests, such as round-trip times, bytes retransmitted, or congestion signals, to name a few. However, for our analysis it will be important to map the client and server of the tests to geographic locations, so that we can do per country analyses, or factor in other variables such as distance between locations.

For that purpose, we use the only location information contained in the original data: the IP addresses of both client and server. Initially we focus on the client side, since we will classify the tests based on the location of the user who runs the test, not of the test server. In order to obtain geographic location information from the client IP address, we have used an online bulk lookup service, in particular the Team Cymru – IP to ASN Mapping service, available at http://www.team-cymru.org/Services/ip-to-asn.html. For each IP address, the lookup service returns the following variables:

- AsDescription: name of the Autonomous System (AS) where the user initiated the
test.

- **AsNum**: Autonomous System number.

- **CountryCode**: two-letter country code where the Autonomous System is located (eg. US, CN).

- **Registry**: Internet registry that controls that Autonomous System (eg. ARIN, APNIC).

From the aforementioned online lookup service, we have obtained the *whois* dataset, which links the IP addresses provided to its corresponding *AsDescription*, *AsNum*, *CountryCode* and *Registry* values. We have proceeded to merge the NDT dataset *lastline* with the *whois* dataset, to obtain a dataset which we have named *filtWhois*.

### A.2 Grouping tests by geographic location

After combining the *lastline* and *whois* datasets, we are able to characterize the geographic location of each of the tests. Thanks to that, we will be performing multiple comparisons across countries. However, since 178 countries are represented in the dataset, that number makes it unpractical to establish cross-country comparisons without performing some grouping first.

The criteria used for classification has been:

**First**, countries belonging to the European Union have been grouped together. For that purpose, the country codes of tests belonging to the list of European Union country codes\(^1\) listed below have been replaced for the code ‘EU’.

\[
\]

**Second**, countries have been ranked by number of tests and the top 10 countries have been selected. The remaining countries have been combined under the group ‘Other’, and

\(^1\text{List of European Union country codes obtained from Eurostat (http://epp.eurostat.ec.europa.eu/statistics_explained/index.php/Glossary:Country_codes =) with the addition of the two alternative codes for Greece – GR, and the United Kingdom – GB.}\)
their country code has been replaced by 'OT'. The top 10 countries by number of tests plus the group 'Other', in decreasing order, are the following:

1. European Union ('EU')
2. Brazil ('BR')
3. United States ('US')
4. Other ('OT')
5. Russia ('RU')
6. India ('IN')
7. Canada ('CA')
8. China ('CN')
9. Ukraine ('UA')
10. Korea ('KR')
11. Japan ('JP')

A.3 Division of the dataset for specific uses

On top of the previous modifications to the initial dataset (adding location information and grouping by country), we will be interested in dividing the original dataset into subsets containing test with a certain trait that makes them interesting for analysis. In general, through our analyses we will start with the original raw NDT dataset and filter it or combine it with other information, leading to new derived datasets. Since it can become tricky to follow all the nomenclature and remember the main traits of each dataset used throughout the analyses, the following list will serve as a reference:

- **lastline**: original, unprocessed, dataset containing the records of 398,529 speed tests performed using the Network Diagnostic Tool. For each test, the values of 114 variables are recorded.

- **whois**: dataset containing 199,468 rows corresponding to the number of unique IP addresses of the users that have taken the test (some users took the test more than once). For each IP address, the fields *AsDescription, AsNum, CountryCode* and *Registry* are saved, as described above.
- **filtWhois**: dataset that results from merging *lastline* and *whois* by the IP address field of the user. Results in a dataset of 118 variables for each of the 398,529 tests recorded, result of adding the 114 variables in *lastline* and the four variables in *whois*.

- **filtLine**: primary dataset used throughout the analyses. It is a filtered version of *filtWhois*. The filtering, as described in section 3.1, consists in removing tests with erroneous durations or values for the variable *CountRTT*. Those “erroneous” tests amount to 17,289, so after removing them the resulting dataset contains 381,240 observations. Other than the filtering, *filtLine* also adds three new variables, bringing the total to 121. The new variables are:
  
  - **AvgRTT**: Average round-trip time during the lapse of a test. Computed by dividing the field *SumRTT* (sum of all the round-trip times during the test) by *CountRTT* (number of round trips in the test):
    
    \[
    \text{filtLine} \cdot \text{AvgRTT} = \text{filtLine} \cdot \text{SumRTT} / \text{filtLine} \cdot \text{CountRTT}
    \]

  - **Speed**: Average effective speed of transmission experienced during the test, based on the total number of bytes transmitted from the server, minus the bytes retransmitted, over the duration of the test. Computed according to the following formula:
    
    \[
    \text{filtLine} \cdot \text{Speed} = (\text{filtLine} \cdot \text{DataBytesOut} - \text{filtLine} \cdot \text{BytesRetrans}) \times 8
    
    / (\text{filtLine} \cdot \text{Duration} / 1e6)
    \]

  - **PercentRetrans**: Percentage of the total bytes sent that are retransmissions of bytes previously sent. Computed as:
    
    \[
    \text{filtLine} \cdot \text{PercentRetrans} = \text{filtLine} \cdot \text{BytesRetrans} \times 100 / \text{filtLine} \cdot \text{DataBytesOut}
    \]

- **congLine**: Dataset derived from * filtLine* containing only tests that encountered congestion, as determined by the variable *CongestionSignals*, and that kept transmitting after the congestion signal (i.e. it had at least one round trip transmission after the congestion signal).

- **USLine**: Subset of * filtLine* containing only tests originated in the United States. For some of our analyses, rather than cross-country comparisons we will focus on a US-specific analysis, comparing Internet Service Providers, Autonomous Systems, etc. For
those analyses, *USLine* will be the dataset used. Moreover, *USLine* includes two new variables that will be used for time series analysis of the data. Those variables are:

- *dateTime*: Contains the date and time that the test was performed. Obtained through a conversion of the variable *StartTimeSec*, which contains the Unix time when the test was run, that is the number of seconds from January 1st, 1970, until the time the test was run.

- *wday*: Contains the day of the week when the test was run. This will be used to look for patterns across different days of the week.

### A.4 Aggregation of the tests by IP

As mentioned before, the base dataset *filtLine* contains records of 381,240 speed tests, performed by 198,936 different users. That means that some users ran the test more than once. In order to perform an accurate statistical analysis of the data, it is important to understand the distribution of tests per user and how that distribution affects the outcome.

In some cases, our interest will lay on the performance experienced by the population as a whole, and therefore having a few users repeat the test many times would present a biased outcome: those users would be dragging the averages to their values. For those circumstances, the solution we propose is to average the results of the multiple tests run by a single user.

On the other hand, in some cases we will be interested in seeing how performance varies across time. In those cases, averaging the results on a per-user basis would eliminate that variation across time, so for those cases we will use the normal, non-averaged, dataset.

For the cases where we are interested in studying the aggregated version of the dataset, we will use a new collection of datasets that derive from the initial datasets described in the previous section, where all the entries containing the same value in the field *IpAddress* have been combined using a SQL statement. For numerical values, this combination process generally means averaging the values, except in select cases where the name of the variable indicates the median has been taken instead.

The list of aggregated datasets is the following:

- *agLine*: Dataset derived from *filtLine* (the standard, filtered dataset) where the rows
have been grouped by the field RemAddress (the IP address of the client).

- **agUSLine**: Dataset derived from USLine (the filtered dataset only containing tests in the United States) where the rows have been grouped by the field RemAddress (the IP address of the client).

Subsequently, another level of aggregation has been performed on the aforementioned datasets, in order to group test records by AS number, so that comparative studies can be performed on a per-AS level. Median values for certain variables, as well as a variable counting the total number of tests from each AS (NumTests), have been added. Those datasets are:

- **ASLine**: Dataset derived from agLine where the rows have been grouped by the field AsNum.
- **ASUSLine**: Dataset derived from agUSLine where the rows have been grouped by the field AsNum.
Bibliography


[54] IP2Location. IP2Location Application FAQs - What is the level of accuracy. [Online].

http://whatismyipaddress.com/geolocation-accuracy