CHARACTERIZING AND IMPROVING LAST MILE PERFORMANCE USING HOME NETWORKING INFRASTRUCTURE

A Thesis
Presented to
The Academic Faculty

by

Srikanth Sundaresan

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
College of Computing

Georgia Institute of Technology
August 2014

CHARACTERIZING AND IMPROVING LAST MILE PERFORMANCE USING HOME NETWORKING INFRASTRUCTURE

Approved by:

Nick Feamster, Advisor
College of Computing
Georgia Institute of Technology

Sivakumar Raghupathy
School of Electrical Engineering
Georgia Institute of Technology

Mostafa Ammar
College of Computing
Georgia Institute of Technology

Renata Teixeira
Institut national de recherche en informatique et en automatique
INRIA Paris-Rocquencourt

Ellen Zegura
College of Computing
Georgia Institute of Technology

Date Approved: 16 May 2014

To Mom, who's been a rock, and Dad, who would have been so proud.

ACKNOWLEDGEMENTS

If, before I started, someone had told me that so many people would have such significant impact on my research (and by extension, this dissertation), I would not have believed them. After all, a PhD is supposed to be a very individual effort. While that is true to a large extent, it is safe to say that this dissertation would have been much poorer but for a variety of people; so much so that it is hard to know where to start acknowledging them.

At least my start at Georgia Tech was surer — I am lucky to have had the opportunity to work with Nick right from the beginning; he's been the best advisor I could have dared to hope for. The trust he placed on me to pick and pursue interesting problems and projects and his unstinting support through the years have made all the difference. He has also been a friend. His dedication to his work has been humbling and inspiring. Renata has been a wonderful as-good-as-co-advisor; always pushing me that much harder so that papers became that much better. The Networking group has been wonderful; it is difficult to imagine that I could have retained my sanity but for the company and the many varied and interesting debates (and beers) with Sam Burnett, Robert Lychev, Samantha Lo, Hyojoon Kim, Ilias Fountalis, Anirudh Ramachandran, Demetris Antoniades, Walter de Donato, Giuseppe Aceto, Vytautas Valancius, Murtaza Motiwala, and Amogh Dhamdhere.

I might not even have considered doing a Ph.D but for the ever-present encouragement and support of my amazing family. My parents, whose sacrifices made it possible for me to be where I am today, deserve much more than meager words. Dad would have been proud and happy. Mom's strength and fortitude has been inspiring. Aravind, as he has done always, opened my eyes to what is possible. He and Kalpana have been a constant source of encouragement (and an occasional source of funds).

It is impossible to acknowledge everyone here. It must suffice to say that this dissertation owes a lot to a lot of people.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

More than a billion people access the Internet through residential broadband connections worldwide, and this number is projected to grow further. Surprisingly, little is known about some important properties of these networks: What performance do users obtain from their ISP? What factors affect performance of broadband networks? Are users bottlenecked by their ISP or by their home network? How are applications such as the Web affected by these factors? Answering these questions is difficult; there is tremendous diversity of technologies and applications in home and broadband networks. While a lot of research has tackled these questions piecemeal, the lack of a good vantage point to obtain good measurements from these networks makes it notably difficult to do a holistic characterization of the "last mile".

In this dissertation we use the home gateway to characterize home and access networks and mitigate performance bottlenecks that are specific to such networks. The home gateway is uniquely situated; it is always on and, as the hub of the network, it can directly observe the home network, the access network, and user traffic. We present one such gateway-based platform, BISmark, that currently has nearly 200 active access points in over 20 countries. We do a holistic characterization of three important components of the last mile using the gateway as the vantage point: the access link that connects the user to the wider Internet, the home network to which devices connect, and Web performance, one of the most commonly used applications in today's Internet.

We first describe the design, development, and deployment of the BISmark platform. BISmark uses custom gateways to enable measurements and evaluate performance optimizations directly from home networks. We characterize access link performance in the US using measurements from the gateway; we evaluate existing techniques and propose new techniques that help us understand these networks better. We show how access link

technology and home networking hardware can affect performance. We then develop a new system that uses passive measurements at the gateway to localize bottlenecks to either the wireless network or the access link. We deploy this system in 64 homes worldwide and characterize the nature of bottlenecks, and the state of the wireless network in these homes — specifically we show how the wireless network is rarely the bottleneck as throughput exceeds 35 Mbits/s. Finally, we characterize bottlenecks that affect Web performance that are specific to the last mile. We show how latency in the last mile results in page load times stagnating at throughput exceeding 16 Mbits/s, and how simple techniques deployed at the gateway can mitigate these bottlenecks.

CHAPTER 1

INTRODUCTION

Of nearly two billion Internet users worldwide, about 500 million are residential broadband subscribers [79]. The United States alone has more than 245 million broadband users, and usage statistics in other regions are even more impressive: at the end of 2012, China reported more than 560 million Internet users, with a penetration rate of more than 40% [80, 81], and Africa is seeing increased penetration and plummeting costs for high-speed connectivity [82, 121]. Home broadband Internet access is also getting faster: the OECD reports that broadband speeds are increasing by about 15–20% every year. Average advertised broadband speeds are now about 16 Mbits/s in the U.S. and 37.5 Mbits/s across OECD areas [122]. Both broadband penetration and speeds are likely to increase further, with people relying on home connectivity for day-to-day and even critical activities.

As broadband becomes ubiquitous and critical to our lives — more of a utility like gas and electricity rather than a service — it is natural to ask how they perform in practice. When the UK regulator Ofcom released a damning report on the state of broadband throughput in the UK in 2010 [165], it got widespread attention. There is widespread interest amongst all the stakeholders — users, ISPs, application providers, and regulators — to know how broadband networks perform. Users want to know whether their ISP is delivering what they are paying for, and be able to troubleshoot their network when something goes wrong (which seems all too frequent). ISPs would like to know what performance they are delivering their customers, and also avoid expensive service calls when something that is out of their hands, like the home wireless network or the application provider, causes performance degradation. For application providers, every millisecond counts in the battle over consumers' attention span and advertising dollars [108]. Naturally, they would like to know how the last mile affects their application so that they can tune their services for optimum user experience. Finally, for regulators, information about end-to-end performance will help

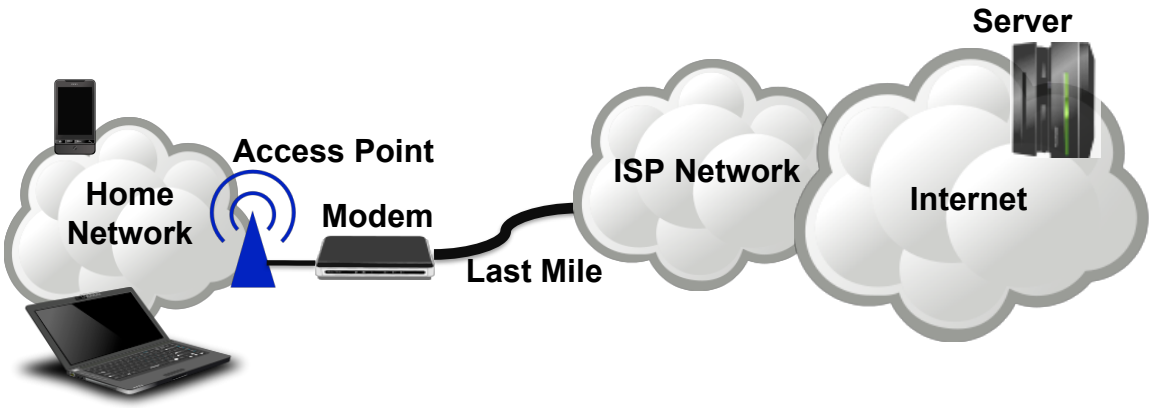


Figure 1: Example home network showing the different components in the end to end path.

shape current and future policy as they weigh how best to ensure that society benefits from the transformative power of the Internet. Surprisingly, given their importance, we do not yet have conclusive answers for the above questions. There have been attempts to answer them, but they fall short for reasons we describe next.

1.1 Last-mile performance characterization: Why is it hard?

Figure 1 shows an example broadband network and the components in the end-to-end path. The access link connects the home network to the ISP and the wider Internet. Researchers and networking practitioners have traditionally used many classes of techniques in the attempt to understand and characterize both home and access networks. We now discuss the challenges involved in performance characterization in the "last mile" and give an overview of how it has been tackled by the research community.

1.1.1 Challenges

The last mile is very heterogeneous in terms of the technologies and applications used. Common access technologies (the "last mile" in Figure 1) are either wireline technologies like DSL, cable, or fiber, or wireless technologies like WiMax. Each of these technologies have unique properties, that may affect applications like Web, video streaming, and Voice over IP (VoIP) differently, as each of these applications have different network requirements. Web browsing needs high bandwidth and low latency to DNS and content servers, gaming requires low latency, and VoIP needs low jitter. This means that the notion of performance

cannot be reduced merely to one number. Consider an example: a user runs the popular Ookla speedtest from her laptop. (Ookla is a popular way for users to test and troubleshoot their network; even many ISP support desks suggest that their clients run this test while troubleshooting.) The tool reports her throughput to be 20 Mbits/s. However, this report does not have enough context to help her. For example, it does not tell her whether the throughput it measured is that of the wireless link or the access link. She has no way of knowing whether she will actually get 20 Mbits/s when streaming a video, or browsing the Web. She cannot tell whether it is good enough to make VoIP calls or for gaming. She cannot even tell whether she will consistently get 20 Mbits/s unless she runs repeated tests over time.

The reason why she doesn't get this information is because it is impossible to get from a simple throughput test. Even application providers, with more sophisticated tools at their disposal, struggle to troubleshoot applications or localize faults in such situations. They can analyze network logs to see if a particular connection was affected (by congestion, loss, or other pathologies), but they cannot proceed much beyond that. Their view from the server stops at the home gateway; the final, critical hop between the gateway and the client is completely hidden from the server. It is not possible to say, for instance, whether a flow was being bottlenecked by the home network or the access link. Depending on the vantage point from which the measurements are conducted, getting longitudinal measurements might not be possible. Trying to understand performance requires repeated measurements over a period of time from a vantage point that is able to see the part of the network that we are interested in.

1.1.2 Techniques

Performance characterization techniques can be classified based on *which component* of the end-to-end path they characterize, *how* they do it, and from *where* they do it.

What to characterize? The question of what to characterize boils down to what performance means in the last mile. It could mean the capacity of the access link or the wireless link in the home, the end-to-end throughput for some applications, or latency for

others. Previous studies have looked into each of the above piecemeal. One class of studies [35, 53, 56, 94] characterizes access link capacity and throughput. Others have looked into performance beyond throughput; Netalyzr [116] analyzes the performance of common protocols using a browser-based tool. Another aspect of performance is how users experience it; Zhang *et al.* study network logs to analyze TCP performance and understand what affects it [169]; other studies have looked into how applications perform [40, 149]. Each of these are valid, and they give us important insight about how different components of the end-to-end path perform.

How to characterize it? Depending on what we want to characterize, there are different techniques one could use. Active techniques generate their own traffic. Examples include throughput measurement techniques, or more comprehensive tools such as Netalyzr [116]. Active techniques allow us to run focused experiments to characterize specific targets instead of relying on the user. The disadvantage with this approach is that such characterization may not translate into the actual performance users get. Sophisticated protocols such as those used for video streaming might be difficult to replicate using active measurements. Passive techniques [41, 169] that rely on analyzing user generated traffic solve this problem because they can observe and analyze the actual performance that users get. However, the effectiveness of passive techniques depend on *where* and *how* the traffic is collected. For instance, passively monitoring traffic at the server might not be enough to identify wireless problems inside the home.

Where to characterize it from? The choice of vantage point has implications on performance characterization. The vantage point could be on the server [56], or the client [35,116], or in the middle [169]. Server-side tools are typically far away from the last mile, with a limited view of the access link or the home network. The target link is typically many hops away and maybe even many ASes (Autonomous Systems) away from the server, which makes it difficult to ascertain whether we are truly characterizing the last mile without being affected by the intermediate network. Tools deployed in user clients seem at first glance to solve the problem of distance from the access link that server-side tools have. However,

while they get a direct view of the home network and are typically only one hop away from the access link, the nature of the home network renders them unreliable too. Such tools are often run from the application layer, which impedes the view of the home network. The home network itself, which is likely to be wireless, introduces a new variable due to the shared nature of the medium. Cross-traffic from other users may also confound measurements and make precise measurements difficult. On top of all this, the clients need not be always on, or they can be mobile, which reduces our ability to get repeated measurements over a period of time. Vantage points in the middle typically do passive analysis of traffic to understand protocol or application performance. T-RAT [169] analyses network logs to understand TCP performance, but while it gives us important insight (*e.g.*, how TCP is being bottlenecked), it does not tell us where the bottleneck is (*e.g.* the home, or the access link).

1.1.3 Why do we need better techniques?

The techniques described above offer a tantalizing glimpse into performance in the last mile. However, they have been largely inconclusive because the data that we can obtain from them is not good enough due the following reasons:

- Not holistic: The above techniques cannot measure all the important components in the last mile — the access link and the home network — directly. This means that it cannot account for the multiple confounding factors in the last mile, and therefore cannot give us a holistic characterization of the last mile.

- Not longitudinal: Many of the techniques are also only able to take only single (or a handful of) measurements from any single end point. They only give us a snapshot of the network; it is hard, if not impossible, to filter out noise and extrapolate much of importance from them.

- Inflexible: As described above, both active and passive techniques are valuable in giving us different views of the network; it is useful to choose one or the other depending on the use case. Most of the described techniques fall into one or the other category.

In short, and rather surprisingly given the stakes involved, our understanding of Internet performance at the edge is hobbled by the lack of a good vantage performance that allows us to characterize it. Going back to the earlier example, knowing that a user measured the capacity of their link to be 20 Mbits/s is useful, but more information is required to make it more meaningful for the user. For example, whether this is expected performance or an outlier; or that application performance more likely to be bottlenecked by their wireless network; or that high latency to their favorite websites means that they never utilize that throughput.

1.2 The home gateway: A unique vantage point

In this dissertation, we use the gateway as the vantage point to characterize last mile performance, and also improve it. The gateway as a platform for home network measurements tackles the shortcomings of other approaches because:

- It is the bridge between the home network and the access network (and the wider Internet). The home network is opaque to the outside world, and, in turn, devices inside have their view out affected by confounding factors such as other devices and cross traffic in the network. The gateway, however, can directly observe both the home and the access networks.

- It is the hub of the home network, and all traffic passes through it. It can therefore observe user behavior if required and also the performance of devices that are connected to it. It is also the natural location to deploy optimizations that would help improve performance for all devices in the home network.

- It is always on, and therefore enables longitudinal and repeatable experiments.

Thesis Statement In this dissertation, we posit that the home gateway provides a vantage point that allows us to holistically characterize performance and bottlenecks unique to home and access networks, and also mitigate them. The access point can act as a platform for home network measurements that tackles the shortcomings of approaches that do not have a clear view of the last mile. Since it acts as the hub of traffic in the home, and has

a view into the home network, the access link, and user traffic, it is able to use active or passive techniques to characterize network or application performance.

We use the gateway as the starting point for understanding properties of broadband and home networks, characterizing their performance, and diagnosing bottlenecks in such networks. We demonstrate how it allows us to characterize the last mile better. We use two deployments: BISmark, a medium-scale (order of hundreds) world-wide deployment of gateways, and FCC/SamKnows, a large-scale (order of thousands) US-wide deployment of home gateways by SamKnows in conjunction with the Federal Communications Commission.

1.3 Contributions

This dissertation provides a framework for understanding and tackling last-mile performance issues in today's Internet. It demonstrates the importance of a good vantage point for good measurements and good data; it is the recurring theme in the components that make up this dissertation. In the context of broadband networks, the ability to obtain repeatable and unclouded measurements – both active and passive, of the access link, the home network, and of application performance – from the gateway transforms our understanding of these networks, allows us to draw meaningful conclusions, and also to develop new methods and techniques for solving problems in these networks. We also underscore the importance of revisiting accepted wisdom in light of new developments, and the need to develop custom techniques to aid our pursuit. For example, our characterization of Web performance seems superficially similar to studies conducted more than a decade ago, but our measurement approach offers new insight into Web bottlenecks in broadband networks. We make the following contributions:

1. **The design and development of a testbed to experiment on home and access networks** We describe our efforts in designing, developing and deploying BISmark, a world-wide home gateway testbed to enable experimentation on home and access networks. BISmark is a programmable platform that has a presence in nearly 200 homes in more than 20 countries. We describe our design choices, and the technical and social challenges we faced during the growth of the deployment.

We demonstrate its value to the networking research community as a platform for conducting continuous and repeatable experiments that offer a unique view into such networks.

2. **Using home gateways to characterize performance, and understand and mitigate bottlenecks in the last mile** We use deployments of home gateways to characterize last mile performance. We look at three important components in the last mile: the access link, the home network, and a popular application, the Web. We first evaluate access link performance in the US using active measurements from the gateway; we evaluate existing methods and propose new ways to characterize broadband networks and how it should be presented to users. We then characterize last mile bottlenecks using passive measurements of user traffic; we develop and deploy new techniques to localize performance bottlenecks to within the home network or outside. We then use active measurements from a custom Web browser emulator to show how the last-mile, particularly the latency introduced by it, can be a critical bottleneck in Web performance. We show the limits of throughput in improving page load time, characterize the overhead of the last-mile on page load time, and propose techniques to mitigate them.

We now describe our contributions in more detail.

BISmark: A Testbed for Deploying Measurements and Applications in Broadband Access Networks BISmark (Broadband Internet Service Benchmark) is a deployment of home routers running custom software, and backend infrastructure to manage experiments and collect measurements. The project began in 2010 as an attempt to better understand the characteristics of broadband access networks. We have since deployed BISmark routers in hundreds of home networks in about thirty countries. BISmark is currently used and shared by researchers at nine institutions, including commercial Internet service providers, and has enabled studies of access link performance, Web page load times, and user behavior and activity. Research using BISmark and its data has informed both technical and policy research. Although BISmark has been successful by many measures, it

continues to face unique technical and social challenges as it matures and expands. In Chapter 4, we describe and revisit design choices we made during the course of the platform's evolution and lessons we have learned from the deployment effort thus far. We also explain how BISmark enables experimentation, and our efforts to make the testbed available to the networking community.

Broadband Internet Performance: A View From the Gateway In Chapter 5, we present the first study of network access link performance measured directly from home gateway devices. Policymakers, ISPs, and users are increasingly interested in studying the performance of Internet access links. Because of many confounding factors in a home network or on end hosts, however, thoroughly understanding access network performance requires deploying measurement infrastructure in users' homes as gateway devices. In conjunction with the Federal Communication Commission's study of broadband Internet access in the United States, we study the throughput and latency of network access links using longitudinal measurements from nearly 4,000 gateway devices across 8 ISPs from a deployment of over 4,200 devices. We study the performance users achieve and how various factors ranging from the user's choice of modem to the ISP's traffic shaping policies can affect performance. Our study yields many important findings about the characteristics of existing access networks. Our findings also provide insights into the ways that access network performance should be measured and presented to users, which can help inform ongoing broader efforts to benchmark the performance of access networks.

Where's the Fault? Characterizing Home Network Performance Problems Chapter 6 presents the design and deployment of *WTF (Where's The Fault?)*, a system that localizes performance problems in home and access networks. WTF uses timing and buffering information from passively monitored traffic at home gateways to detect both access link and wireless network bottlenecks with high accuracy and low false positive rates; we use extensive controlled experiments to validate each parameter that we measure. WTF also collects wireless metrics from the traffic to gain deeper insight into wireless performance in homes. We implemented WTF as custom firmware that runs on BISmark routers and

deployed it in 66 home networks across 15 countries. The real-world deployment sheds light on common pathologies that occur in home networks. We find that wireless bottlenecks are significantly more common than access link bottlenecks, especially as access link throughputs increase beyond 35 Mbits/s, that the 5 GHz spectrum consistently outperforms the 2.4 GHz spectrum, that many homes experience high TCP round-trip latencies between wireless clients and the access point, and that performance can vary dramatically across wireless devices, even within a single home network.

Measuring and Mitigating Web Performance Bottlenecks in Broadband Access Networks In Chapter 7, we measure Web performance bottlenecks in home broadband access networks and evaluate ways to mitigate these bottlenecks with caching within home networks. We first measure Web performance bottlenecks to nine popular Web sites from more than 5,000 broadband access networks and demonstrate that when the downstream throughput of the access link exceeds about 16 Mbits/s, latency is the main bottleneck for Web page load time. Next, we use a router-based Web measurement tool, Mirage, to deconstruct Web page load time into its constituent components (DNS lookup, TCP connection setup, object download) and show that simple latency optimizations can yield significant improvements in overall page load times. We then present a case for placing a cache in the home network and deploy three common optimizations: DNS caching, TCP connection caching, and content caching. We show that caching only DNS and TCP connections yields significant improvements in page load time, even when the user's browser is already performing similar independent optimizations.

1.4 Who should read this dissertation?

This dissertation caters to those interested in deploying large-scale measurement infrastructure, particularly at the edge of the Internet, and those interested in understanding performance issues on the edge, and how to characterize them better.

As the Internet becomes more heterogeneous, building different types of measurement infrastructure becomes more important. Due to the nature of Internet access, it is possible that much of this infrastructure may need to be deployed closer to users, or even in their

homes or devices. Those interested in building such large distributed systems will find interesting insights in Chapter 4 on how to tackle technical and, equally importantly, social issues associated with deploying such systems.

Chapters 5 and 7 describe attempts to characterize the access link and Web performance respectively, using new (and established) active techniques. It provides insights on how to design and deploy such techniques to understand last-mile performance issues, and also detailed characterization of broadband and Web performance in homes, respectively. Users and regulators will find insight into the state of broadband in the US, and how to understand it better. Application providers and ISPs can potentially improve performance for users based on our characterization of last-mile bottlenecks as well as our suggestions on how to mitigate them. Chapter 6 describes a passive technique to localize network bottlenecks to either the access link or the home network. It provides insight on bottleneck detection and localization for network researchers, particularly in resource constrained settings. It also provides a characterization of such bottlenecks in 66 homes from which ISPs and regulators can get a first-order impression of home network performance.

Those unfamiliar with the background on access and home network technologies and research literature should read Chapters 2 and 3 for context. Chapter 2 covers the common technologies and applications in the last mile. Chapter 3 provides an overview of the literature in characterization, trouble shooting, and application performance improvement in the last mile.

1.5 Bibliographic Notes

A version of the material in Chapter 4 that describes the BISmark platform was published as a paper co-authored with Sam Burnett, Nick Feamster, and Walter de Donato [157] (an earlier version was published as a poster [54], co-authored with Walter de Donato, and Nick Feamster, Renata Teixeira and Antonio Pescapé). The work on access link performance characterization, presented in Chapter 5, was published with coauthors Walter de Donato, Nick Feamster, Renata Teixeira, and Sam Crawford, and Antonio Pescapé [155, 158]. The work on characterizing and mitigating Web performance bottlenecks, presented in Chapter 7

was published as a paper coauthored with Nick Feamster, Renata Teixeira, and Nazanin Magharei [159]. Early versions were published as an extended abstract [161] and as a poster [160].

CHAPTER 2

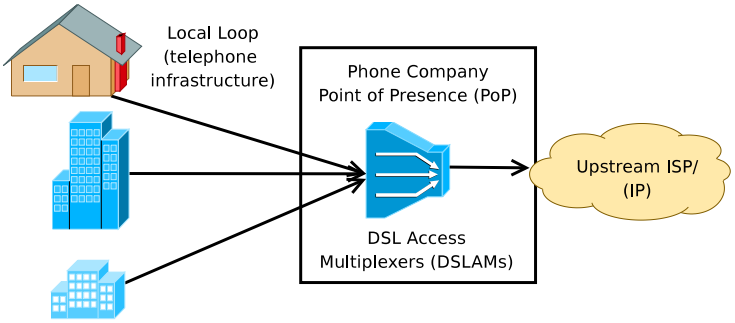
BACKGROUND

Broadband networks typically comprises of a variety of different technolgies in the end-to-end path. A typical home user might be "connected" via a wireless network; while her Internet service might be cable, DSL, fiber, or a wireless technology such as WiMax. Unsurprisingly, this heterogeneity has an impact on performance. This thesis looks at various performance issues in home and broadband networks; this chapter provides a brief background on the topics we cover in the thesis.

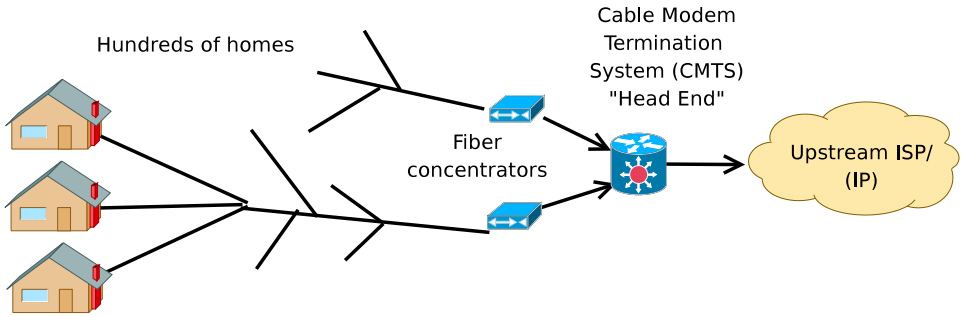
2.1 Access Networks

The two most common access technologies that we encounter in our deployments are Digital Subscriber Line (DSL) and cable. Although a few users in our deployments have fiber-to-the-node (FTTN), fiber-to-the-premises (FTTP), and WiMax, we do not have enough users to analyze these technologies. We provide an overview of how DSL and cable works, and describe how a user's choice of service plan and local configuration can affect performance.

DSL networks use telephone lines; subscribers have dedicated lines between their own DSL modems and the closest DSL Access Multiplexer (DSLAM). The DSLAM multiplexes data between the access modems and upstream networks, as shown in Figure 2a. The most common type of DSL access is asymmetric (ADSL), which provides different upload and download rates. In cable access networks, groups of users send data over a shared medium (typically coaxial cable); at a regional *headend*, a Cable Modem Termination System (CMTS) receives these signals and converts them to Ethernet, as shown in Figure 2b. The physical connection between a customer's home and the DSLAM or the CMTS is often referred to as the *local loop* or *last mile*. Users buy a service plan from a provider that typically offers some *maximum* capacity in both the upload and download directions.



(a) DSL.



(b) Cable.

Figure 2: Access network architectures.

ADSL capacity. The ITU-T standardization body establishes that the achievable rate for ADSL 1 [83] is 12 Mbps downstream and 1.8 Mbps upstream. The ADSL2+ specification [84] extends the capacity of ADSL links to at most 24 Mbps download and 3.5 Mbps upload. Although the ADSL technology is theoretically able to reach these speeds, there are many factors that limit the capacity in practice. An ADSL modem negotiates the operational rate with the DSLAM (often called the *sync rate*); this rate depends on the quality of the local loop, which is mainly determined by the distance to the DSLAM from the user's home and noise on the line. The maximum IP link capacity is lower than the sync rate because of the overhead of underlying protocols. The best service plan that an ADSL provider advertises usually represents the rate that customers can achieve if they have a good connection to the DSLAM. Providers also offer service plans with lower rates and can rate-limit a customer's traffic at the DSLAM.

Modem configuration can also affect performance. ADSL users or providers configure their modems to operate in either *fastpath* or *interleaved* mode. In *fastpath* mode, data

is exchanged between the DSL modem and the DSLAM in the same order that they are received, which minimizes latency but prevents error correction from being applied across frames. Thus, ISPs typically configure fastpath only if the line has a low bit error rate. *Interleaving* increases robustness to line noise at the cost of increased latency by splitting data from each frame into multiple segments and interleaving those segments with one another before transmitting them.

Cable capacity. In cable networks, the most widely deployed version of the standard is Data Over Cable Service Interface Specification version 2 (DOCSIS 2.0) [85], which specifies download rates up to 42.88 Mbps and upload rates up to 30.72 Mbps in the United States. The latest standard, DOCSIS 3.0, allows for hundreds of megabits per second by bundling multiple channels. Cable providers often offer service plans with lower rates. The service plan rate limit is configured at the cable modem and is typically implemented using a token bucket rate shaper. Many cable providers offer *PowerBoost*, which allows users to download (and, in some cases, upload) at rates that are higher than the contracted ones, for an initial part of a transfer. The actual rate that a cable user receives will vary with the network utilization of other users connecting to the same headend. The CMTS controls the rate at which cable modems transmit. For instance, Comcast describes that when a CMTS's port becomes congested, it ensures fairness by scheduling heavy users on a lower priority queue [18].

2.2 The Home Network

The home network resides behind the modem and the home router, which typically has multiple ethernet ports and also one or more wireless networks. The modem or the router gets a single IP address from the host ISP, typically over DHCP (Dynamic Host Configuration Protocol) [59,60]. It then provides Network Address Translation, commonly abbreviated to NAT [78]. Devices behind a NAT typically get assigned IP addresses in the private IP address range [140]. From outside, it is not possible to distinguish between traffic originating from different devices within the home network without higher layer hints.

Ethernet ports are typically at least 100 Mbits/s, which ensures that they are very

rarely the bottleneck. However, depending on the wireless technology and the quality of the wireless channel, the wireless network capacity could vary from a few Mbits/s to a few hundred Mbits/s. 802.11b provides up to 11 Mbits/s, while 802.11a and 802.11g provide up to 54 Mbits/s. 802.11n provides anywhere between 130 Mbits/s to 300 Mbits/s depending on the configuration. These bitrates are the rates at which the frames are sent; the actual throughput obtained by transport protocols such as TCP may be much lower due to protocol and medium access overhead. The bitrates are not fixed for a protocol; they are decided by bitrate adaptation algorithms depending on channel properties such as signal strength or retransmission rates. In general Lower bitrates have better error recovery properties and therefore higher probability of being successfully delivered. OpenWrt [124], an open Linux-based distribution for home wireless access points deploys the Minstrel rate adaptation algorithm, which adapts its bitrate according to retransmission rates.

There are two radio frequencies that 802.11 uses; 2.4 GHz and 5 GHz. 2.4 GHz, which is used by 802.11b, g, and n, is the most commonly used [73], although 5 GHz, used by 802.11a and n, is also used in the US. Regulatory restrictions prevent the use of 5 GHz in many countries around the world. the 2.4 GHz channel is also crowded; household appliances such as microwave ovens, cordless phones, and baby monitors, all use the same frequency, which can interfere with wireless communication. The 5 GHz channel is less used, but may have higher attenuation than 2.4 GHz.

Due to the vagaries of the wireless channel, the performance users get is harder to predict. As broadband networks get faster, it is possible that users are bottlenecked not by their ISP, but by their home network. Due to NAT, debugging performance issues in the home becomes more challenging.

2.3 Web Performance

Factors that affect page load time Web downloads begin by downloading the home page of the requested Web page. The home page object typically contains references to other objects that the browser must subsequently retrieve. Each of these objects is referenced with another URL, which the browser retrieves with additional HTTP requests. These

objects are either static, in which case the URL is in the home page itself; or dynamic, in which case the URL is determined by running active scripts (*e.g.*, Javascript) that determine which objects to download. Modern Web sites typically contain a mix of static and dynamic objects. Browsers minimize the download time of the individual objects using many optimizations such as parallel fetches, DNS caching, TCP connection reuse, and optimizations for processing scripts and rendering. Different browsers implement different sets of optimizations.

Because the time for the browser to render a page depends on the choice of browser and machine configuration, we instead define *page load time* as the time from the initial request to the time when all objects for the page have been retrieved. The retrieval time for each object has the following components: (1) the *DNS lookup time* for the domain referenced in the object URL; (2) the *TCP connection time*, which is the time to complete the TCP three-way handshake to the server; (3) the *server response time*, which is the time it takes for the server to start sending the object once it has been requested; and (4) the *object download time*, which is the time to download the object itself over TCP. Some of these factors, such as the DNS lookup and TCP connection times, are bound by latency; others, such as the object download time, are bound by both latency and throughput.

Optimizations to improve page load time A number of optimizations have been proposed and implemented in the quest to minimize page load time. Server-side optimizations include HTTP replacements [135,153] and TCP modifications [19,20,42,47,61]. Recent proposals suggest using a larger initial congestion window sizes on servers for TCP connection, so that small objects can be transferred with significantly fewer round trips [61]. Al-Fares *et al.* studied the effects of server-side optimizations, such as increasing TCP's initial congestion window (ICW) and enabling HTTP pipelining on Web page load times [11]. They found that increasing the ICW can reduce page load times by several hundred milliseconds in many cases. Since latency also affects loss receovery, there have been recent proposals to improve recovery from faults by proctively sending multiple copies of packets [68]. Although these server-side optimizations can improve page load times, they do not reduce

certain components that contribute to page load time, including DNS lookup and the TCP connection setup time.

Many client-side optimizations from the browser have also been developed. HTTP/1.1 introduced persistent connections and pipelining. Persistent connections allow the client to retrieve multiple objects over the same TCP connection (thereby amortizing the three-way handshake and TCP congestion window ramp up cost over multiple objects). Pipelining allows the client to initiate a request for the next object as soon as it sees a reference to that object in another object (rather than waiting for the object download to complete). Nielsen *et al.* showed the superior performance of HTTP/1.1 over HTTP/1.0 [120]. However, most browsers do not enable pipelining by default, and some servers do not enable persistent connections. Zhou *et al.* propose a new protocol that minimizes connection time by having DNS resolvers set up TCP connection on the client's behalf [170]. Browsers also download multiple objects in parallel and have highly optimized engines for parsing and rendering objects.

Content caching is also a common optimization. Content Distribution Networks (CDNs) are large distributed caches that are typically deployed at the edge of ISPs to reduce the latency between the end-host and the content. Ihm *et al.* characterized five years of Web traffic traces from a globally distributed proxy service; they observe that Web caches typically have a 15–25% hit rate, and these rates could almost double if caches operated on 128-byte blocks [77]. Previous studies have reported object cache hit rates in the range of 35–50%, although these cache hit rates have continued to drop over time [3,33,72,114,168]. Some browsers also support content prefetching [107]; Padmanabhan *et al.* proposed predictive content prefetching using server hints [127].

To improve cache hit ratios, Web browsers prefetch DNS records anticipating client requests; the browser parses certain downloaded pages (*e.g.*, a search result page) for domains and resolves them before the user clicks on them [57]. To reduce the time associated with DNS lookups, browsers and intermediate DNS servers employ caching and prefetching [48,49]. Jung *et al.* studied DNS performance and the effectiveness of DNS caching [92], and saw that DNS cache hit rates can be as high as 80%, even with only a few clients [91].

Feldmann *et al.* observed in Web traces from AT&T home users that 47% of objects retrieved incur more than half of the total download time from TCP connection setup [66]. Based on this observation, the study proposes a *connection cache* in the ISP network to reduce connection setup time, which reduces download times by up to 40%.

CHAPTER 3

RELATED WORK

3.1 Measuring Networks at Scale

Fixed server or gateway deployments. PlanetLab [129, 130] is similar to BISmark in the sense that it aims to be a fixed, large-scale deployment that hosts a variety of research experiments. Because BISmark is deployed in home networks on resource-constrained devices, however, it faces a unique set of challenges that PlanetLab does not face. The RIPE Atlas [142] project has deployed hundreds of probing devices around the world, but the capabilities of these devices are limited: they allow experimenters to conduct simple measurements (*e.g.*, ping, traceroute), but they do not allow heavier, or application-specific measurements. SamKnows [144] has deployed thousands of home gateway devices across various countries (*e.g.*, the US and UK), but they only support limited performance measurements and are focused in a few countries; in contrast, BISmark supports more experiments and is deployed in more countries.

Host-based deployments. Dasu [145] is host-based and allows a variety of network measurements from end hosts, but limits permissions to run certain measurements. It cannot perform direct measurements of either the access network or the home network. It Measurements also (1) cannot be continuous (*i.e.*, since hosts can be turned off, moved, etc.); (2) do not reflect the performance of a fixed network vantage point, since the measurements can reflect limitations of the host or the application associated with the measurement. The Grenouille project in France [71] uses an agent that runs from a end host inside the home network. Neti@Home [89] and BSense [21] also use this approach, although these projects have fewer users than Grenouille.

Seattle [37] allows researchers to deploy tests using a restricted version of Python; the platform provides a portable virtual machine environment but does not itself specify a set of measurements. It also is a *software* distribution, and does not provide a fixed set of

hardware resources for experimentation.

There are various options for users who want to diagnose their network. Speedtest by Ookla [123] is commonly used by ISPs and end users to diagnose performance issues. It measures latency and downstream and upstream throughput to a nearby server. Netalzyr [100] allows users to conduct a series of tests from a browser and provides a rich set of measurements about the host, home network, and the access link. The drawback with such measurement approaches is that they are not continuous, and researchers cannot run custom tests from a set of hosts—the measurements collected are fixed, and the set of hosts from which measurements are collected depend on the users who decide to run the tool.

Programming frameworks. The process of vetting BISmark experiments is manual (as it was in previous testbeds such as RON [13]), which will be a limiting factor as the testbed grows. BISmark must ultimately strike a balance between flexibility (allowing researchers to specify experiments) and a constrained programming environment (limiting researchers from specifying experiments that could interfere with a user's Internet connection). Previous work on sandboxed, programmable measurement environments, such as ScriptRoute [154], could ultimately serve as a useful environment for specifying BISmark tests.

3.2 Measuring the Last-mile

From access ISPs. Previous work characterizes access networks using passive traffic measurements from DSL provider networks in Japan [46], France [149], and Europe [110]. These studies mostly focus on traffic patterns and application usage, but they also infer the round-trip time and throughput of residential users. Without active measurements or a vantage point within the home network, however, it is not possible to measure the actual performance that users receive from their ISPs, because user traffic does not always saturate the user's access network connection. For example, Siekkinen *et al.* [149] show that applications (*e.g.*, peer-to-peer file sharing applications) often rate limit themselves, so performance observed through passive traffic analysis may reflect application rate limiting, as opposed to the performance of the access link.

From servers in the wide area. Other studies have characterized access network performance by probing access links from servers in the wide area [53,56]. Active probing from a fixed set of servers can characterize many access links because each link can be measured from the same server. Unfortunately, because the server is often located far from the access network, the measurements may be inaccurate or inconsistent. Isolating the performance of the access network from the performance of the end-to-end path can be challenging, and dynamic IP addressing can make it difficult to determine whether repeated measurements of the same IP address are in fact measuring the same access link over time. A remote server also cannot isolate confounding factors, such as whether the user's own traffic is affecting the access-link performance.

From inside home networks. The Grenouille project [71] measures the performance of access an agent that runs from a user's machine inside the home network. PeerMetric [104] measured P2P performance from about 25 end hosts. Installing software at the end-host measures the access network from the user's perspective and can also gather continuous measurements of the same access link. Han *et al.* [74] measured access network performance from a laptop that searched for open wireless networks. This approach is convenient because it does not require user intervention, but it does not scale to a large number of access networks, cannot collect continuous measurements, and offers no insights into the specifics of the home network configuration.

Other studies have performed "one-time" measurements of access-link performance. These studies typically help users troubleshoot performance problems by asking the users to run tests from a Web site and running analysis based on these tests. Netalyzr [116] measures the performance of commonly used protocols using a Java applet that is launched from the client's browser. Canadi *et al.* [35] study broadband performance world-wide using data obtained from Ookla; while Chetty *et al.* [44] combine Ookla, BISmark, and mobile measurements data to study broadband network in South Africa. While this crowd-sourced approach is scalable and offers insight into broadband performance, it lacks the depth offered by repeated measurements from the same hosts over a period of time. Network Diagnostic

Tool (NDT) [39] and Network Path and Application Diagnostics (NPAD) [111] send active probes to detect issues with client performance. Glasnost performs active measurements to determine whether the user's ISP is actively blocking BitTorrent traffic [70]. Users typically run these tools only once (or, at most, a few times), so the resulting datasets cannot capture a longitudinal view of the performance of any single access link. In addition, any technique that measures performance from a device inside the home can be affected by factors such as load on the host or features of the home network (*e.g.*, cross-traffic, wireless signal strength). Finally, none of these studies measure the access link directly from the home network gateway.

3.3 Diagnosing and Characterizing Network Performance

Measuring and diagnosing network performance issues has a long history that has spanned many types of networks. In this section, we briefly survey prior approaches and discuss why home networks require a new approach.

There have been many previous approaches to diagnosing wireless networks. One approach is to deploy passive traffic monitors throughout the network. Kanuparthy *et al.* [96] develop a tool to detect common wireless pathologies (such as low signal-to-noise ratio, congestion, and hidden terminals) by using both active probes and an additional passive monitor deployed within the network. Cooperative techniques can also diagnose certain classes of problems like hidden terminals and conflict graphs [6, 119]. Pervasive monitoring approaches work well in enterprise networks [2, 43, 137]: Mahajan *et al.* study the wireless performance in a large network by collecting traces from many vantage points and piecing them together [109]. Judd *et al.* [90] characterize the link-layer performance of 802.11 under various different cases such as clear channels and with hidden and exposed terminals. Unfortunately, it is difficult to perform this kind of extensive monitoring in many home networks, since it requires deploying equipment beyond that which a normal user is typically willing to install or have installed in their home.

Other approaches have monitored wireless networks with custom hardware [41, 105, 137–139]. RFDump [105] is a tool built on GNU Radio and USRP to monitor heterogeneous

wireless networks with devices such as Bluetooth. AirShark [138] exploits a recent 802.11 chipset to collect spectrum samples, allowing for detection of non-WiFi interference.

Several techniques for detecting bottlenecks in wide-area networks exist; these approaches typically rely on active measurements [14, 75, 76, 86, 95, 103, 141, 146, 147]. PathNeck [75, 76], for instance, is an active probing tool which can accurately locate bottleneck links in a wide-area network. Unfortunately, in home networks, active techniques have two key disadvantages: they may not accurately reflect the actual performance users experience (and even interfere with it), and additional cross-traffic can actually affect the wireless network's performance. Thus, we design a passive monitoring technique for bottleneck detection in WTF.

Zhang *et al.* develop T-RAT [169] to analyze TCP performance. T-RAT estimates TCP parameters such as maximum segment size, round-trip time, and loss to understand flow behavior. Katabi *et al.* [97], use entropy in packet interarrival time to estimate shared bottlenecks. Biaz *et al.* [22] use packet interarrival times for distinguishing between different kinds of losses. WTF is similar to some of the approaches used in these papers (*e.g.*, it uses packet interarrival time as input to a maximum likelihood detector for access link bottlenecks), but we tailor our approach so that it only relies on data that can be easily collected from a home router.

Home networks can also be subject to performance problems caused by explicit policy or configuration decisions. Netprints [5] is a diagnostic tool for home networks solves problems arising due to misconfigurations of home network devices including routers. Other work has explored the extent of performance degradations due to service discrimination [55, 70, 93, 162].

3.4 Measuring and Modeling Web Performance

There is much previous work on measuring and modeling Web performance, ranging from protocol modeling to empirical analysis to designing techniques to improve performance. Many of these previous studies were performed before the growth of broadband access networks, content distribution networks, and modern browsers. This evolution suggests

that it is time for a reappraisal of some past studies in the context of broadband access networks, although it is important to put our study in the context of previous work. Below, we compare past work in measuring and modeling Web performance to the study in this paper.

Measuring Web performance Barford *et al.* analyzed the contribution of server, client, and network delays to HTTP 1.0 transaction times [17]; the authors used virtual clients on several campus networks to request files from a virtual Web server. They found that for small files, server load is the dominant factor for performance, while for large files, the network properties dominate. Krishnamurthy and Wills analyzed the impact of variants of TCP (*i.e.*, Tahoe, Reno) and HTTP (*i.e.*, persistent, parallel, or pipelined) on Web performance; they also studied the effects of network latency and server load on observed performance [101]. These studies were performed more than ten years ago. During this time period, the Web has changed dramatically, in terms of the type of content is served, how content is hosted, and how browsers operate. Additionally, broadband Internet access has proliferated since the late 1990s. It is also much faster than it was when these original studies were conducted and is now a predominant mode of Internet access. To our knowledge, this study is the first to explore Web page load times from broadband access links, and the first to quantify the extent to which latency becomes a bottleneck on high-throughput access links.

More recent work has studied the performance of CDNs [88] and caching [63]. Akella *et al.* [7–10] studied the effects of the performance of 68 CDN sites across 17 cities, focusing on how server multihoming can improve CDN performance for clients. "WhyHigh?" identifies cases where a certain set of clients experience higher Web page load times [102]. Butkiewicz *et al.* studied how the complexity of modern Web sites may contribute to slower page load times and found that more than 60% of the Web sites they profiled retrieve content from five non-origin sources that contribute to more than 35% of the bytes downloaded [32]. Flach *et al.* [68] study the impact of loss on TCP performance and find that lossy connections can take as much as five times more time to finish than non-lossy connections; they propose

proactive loss recovery mechanisms to reduce this delay.

The online service `webpagetest.org` [167] analyzes the contribution of different network components to overall page load time via a "waterfall"; the analysis is similar to the decomposition that we perform in this paper, although the software we develop can run directly on home routers, unlike `webpagetest.org`, which runs from data centers. Keynote [98] and Compuware [52] perform end-to-end Web performance measurements from a variety of vantage points, but they do not study how individual network factors such as latency, loss, or throughput ultimately contribute to page load time.

Modeling Web performance Previous work developed models for TCP and HTTP performance based on various network properties such as latency and packet loss [12, 15, 36, 38, 126]. We present an empirical study of Web performance from thousands of access networks. This empirical approach provides new insights into how access network characteristics affect each component of page load time. WebProphet [106] and WProf [164] analyze dependencies between Web page objects to predict browser-level response time and study bottlenecks in the critical path of page load times. These tools are valuable in the quest to improve page load times by exposing inefficiencies in Web site design, but they are difficult to deploy at scale.

CHAPTER 4

BISMARK: A TESTBED FOR DEPLOYING MEASUREMENTS AND APPLICATIONS IN BROADBAND ACCESS NETWORKS

The changing nature of Internet access makes it imperative to study Internet connectivity as most users now experience it, and to develop a platform for developing, testing, and deploying new systems and services for common access network environments. To support rich and accurate Internet measurements, researchers need a testbed that represents the perspective of the growing population of Internet users. Because increasingly many users access network services from home broadband networks, the ability to perform continuous, reliable measurements from these vantage points is critical.

To address this need, we have developed BISmark, a system that allows researchers, operators, and policymakers to deploy experiments (and applications) and gather data about network availability, reachability, topology, security, and performance from globally distributed access networks. Given the long-running history of wide-area testbeds such as PlanetLab, the BISmark vision at a glance would appear to be merely an application of these concepts to home networks (albeit more modest, since BISmark doesn't even support features like virtualization). As PlanetLab operators will readily admit, however, dealing with physical infrastructure is expensive and challenging. A lot of similar projects (*e.g.*, RON [13], RoofNet [24]) do not outlive the initial research effort, yet the research community desperately needs measurement vantage points in non-academic networks.

Beyond the conventional challenges of operating a long-running service in the wide-area Internet (*e.g.*, PlanetLab), deploying such a service in *home networks* poses a unique set of challenges. First, incentives do not naturally align: whereas in PlanetLab, researchers have an incentive to host machines to gain access to the testbed, BISmark explicitly targets home users, who may not necessarily be interested in networking research. Second, unlike in universities where PlanetLab nodes are deployed, technical support is not readily

available, which makes system robustness, remote maintenance, and recovery even more important. Third, nodes must be small and easy-to-deploy; such nodes are typically resource-constrained. Finally, BISmark nodes are on the direct path of real Internet users; a malfunctioning BISmark device could disrupt a normal user's Internet connectivity, which could result in the loss of the device as the user is likely to remove the device from the network entirely and never re-install it. BISmark addresses these challenges and also achieves the following goals, many of which are common to other long-running testbeds but present unique technical challenges introduced by the constraints of home networks:

- *Continuous.* Although many tools can perform "one shot" measurements of Internet characteristics from access networks, these measurements do not reveal patterns (*e.g.*, diurnal trends) or evolution (*e.g.*, the change in performance over time, as conditions evolve). BISmark supports continuous measurement.

- *Direct.* Users can run measurement software from handsets, browsers, and host applications; and content providers (*e.g.*, Google, Amazon) can measure performance from applications. Unfortunately, these measurements are indirect; they often reflect performance limitations of the host, application, or the home wireless network. BISmark allows direct, unambiguous measurements of the edge of the network.

- *Diverse.* One lesson from our continued study of the Internet's edge is that there is no such thing as a "representative" set of vantage points. Network characteristics vary considerably by ISP, country, and service plan. Most current testbeds are deployed on academic networks (*e.g.*, PlanetLab) that do not reflect connectivity that most Internet users experience.

- *Extensible.* Researchers need to perform custom network measurements or deploy custom systems or services. A lightweight probe capable of simple measurements is not extensible enough for many applications.

- *Robust.* Because the platform lies on the critical path of a typical Internet user, an unstable device or rogue experiment can wreak havoc on a user's Internet connection,

and ultimately result in the loss of the device if the user "decommissions" it out of frustration.

- *Secure.* A BISmark router should not compromise user security or privacy. It should be safe from external attacks, and experiments should not monitor PII without the user's explicit and informed consent.

BISmark has enjoyed reasonable success during its first four years. It has enabled the publication of many studies from broadband access networks from around the world, and is now being adopted by major ISPs, policymakers, and researchers in several countries. Many research groups are either using the data we have collected or deploying their own custom experiments. Yet, enabling a broader collection of experiments and scaling BISmark beyond its current size poses new challenges. Security and robustness are becoming increasingly important, and device deployment and attrition remain challenges, particularly in certain regions. This chapter discusses the constraints we have faced (and continue to face) in the design, implementation, and deployment of BISmark, discusses lessons and things that we would have done differently (or will change in the future), and describes new challenges as the platform expands both in terms of the number of vantage points and the diversity of experiments we aim to support.

4.1 Architecture and Implementation

BISmark aims to enable a variety of research and experimentation under constraints inherent to home routers. Many of the challenges that we faced are not unique to our deployment, but they are exacerbated by operating (1) in a resource-limited setting on home routers; (2) in a setting where downtime (or general interference with users' Internet connectivity) is not acceptable.

BISmark's software fulfills four roles. First, it uniquely identifies each router and correlates it with metadata useful for conducting networking research. Second, it manages software installation and upgrades, which lets us fix bugs, issue security patches, and deploy new experiments after we have mailed the routers to participants. Third, it provides

experiments a common, easy-to-use, and efficient way to upload data to a central collection server. Finally, it enables flexible and efficient remote troubleshooting. We describe BISmark's evolution path, its components, and the various roles that the BISmark software plays.

4.1.1 Evolution Path

Like many rapidly growing systems, BISmark's software evolved organically in response to use. Several components written for an early pilot deployment persist. Although many design choices were sub-optimal in retrospect, the software has always addressed three practical constraints.

Constraint 1 () *Severely limited client resources dominate software design decisions.*

Resource limitations preclude several conveniences. For example, we cannot run heavy scripting languages like Python or Ruby; instead, we glue together standard UNIX utilities and small C programs with shell and Lua scripts.

Constraint 2 () *The basic routing functionality of BISmark routers is critical; users often place them on the home network's forwarding path.*

Therefore, the router should not noticeably affect the user's home networking experience (*e.g.*, by frequently saturating the uplink). Combined with limited client resources, this constraint requires us to thoroughly test software before deploying it, because the consequences of malfunctioning software may be the potentially terminal loss of a deployment site. (In our experience, most users simply unplug the router at the first annoyance and never plug it in again.)

Constraint 3 () *User intervention is impractical and should be as limited as possible.*

Although users expect their router to always forward traffic, they have no desire to otherwise interact with it. After installation, attempts to interact with users via the router itself are awkward and annoying (*e.g.*, captive portals) and out-of-band communication (*e.g.*, email) is unreliable.

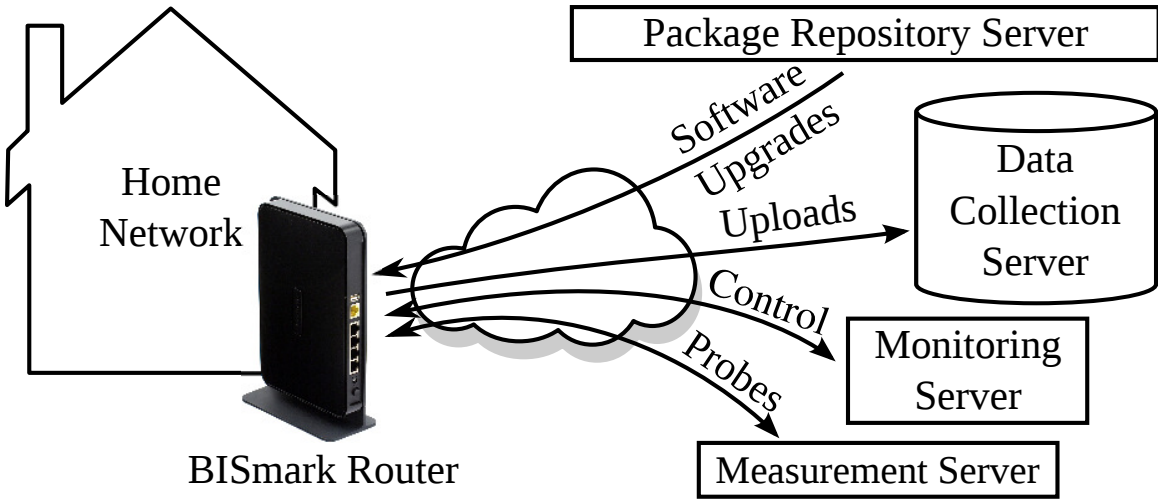


Figure 3: BISmark architecture. Both the Packages Server and Data Collection Server scale into multiple server instances. The Monitoring Server is harder to scale, but also sees less load.

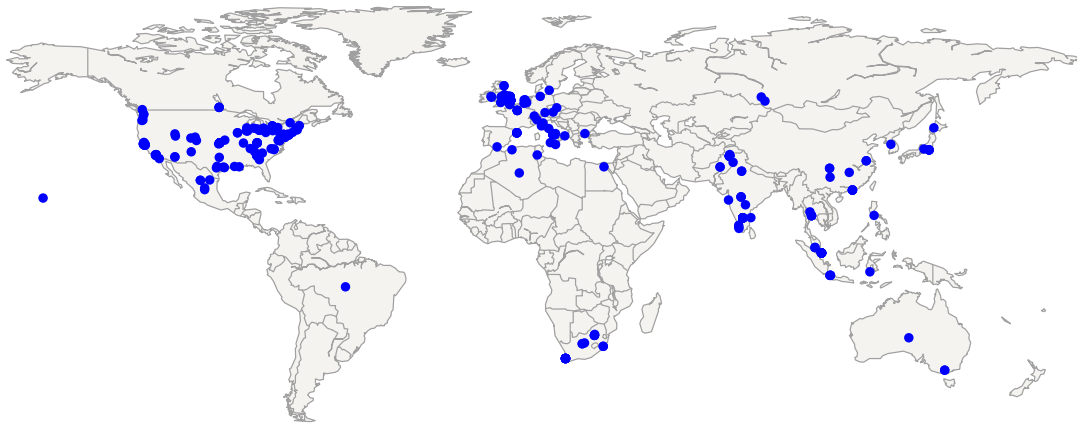


Figure 4: Locations of the 191 BISmark routers that were online in September 2013. We have focused concentrations of routers in the US, South Africa, Pakistan, and the UK.

4.1.2 System Components

Figure 3 shows BISmark's architecture. The deployment currently comprises hundreds of BISmark routers and a collection of servers that manage software, collect data, and facilitate troubleshooting.

BISmark routers. BISmark currently has a deployment of hundreds of home routers. As of January 2014, the deployment has 195 active routers in over 20 countries. Figure 4 maps

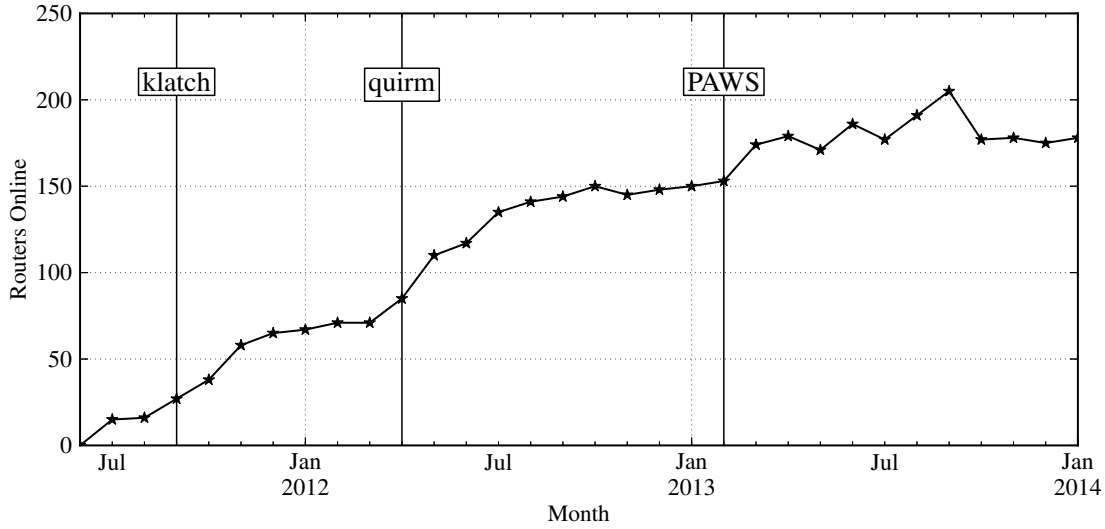


Figure 5: The number of routers online during each month. BISmark has grown to nearly 200 routers over the past two and a half years. *klatch* and *quirm* signposts indicate two firmware releases; *PAWS* indicates a new deployment in the UK. Growth falters in some later months because we focused deployment in developing countries, where router availability is inconsistent.

router locations and Figure 5 graphs the deployment's growth over the past two and a half years. We purchased, prepared, and mailed nearly half of these routers, although some of the deployment sites have arisen either organically (*e.g.*, as users "flash" their own routers with BISmark firmware) or through coordinated efforts (*e.g.*, with other organizations or research groups). The current deployment uses Netgear WNDR 3700v2 and WNDR 3800 routers, which have a 450 MHz MIPS processor, 16 MB of flash storage, 64 MB or 128 MB of RAM, 5 Gigabit Ethernet ports, and a dual-band wireless interface. This hardware is limited, even when compared to other embedded mobile devices like smartphones, yet it is powerful enough to reliably support both basic routing features and a variety of measurement experiments. Our latest build supports more hardware; we are planning to extend our deployment with the similar TP-Link WDR3600.

We replace each router's default software with a custom version of the OpenWrt Linux distribution [124]. OpenWrt has an active developer community, a simple and usable configuration GUI, and broad, mature, and consistent hardware support [125]. It frees us from maintaining our own firmware, but ties us to its release cycles, bugs and all. Despite a

few persistent problems that have we have successfully deployed three hardware revisions of Netgear routers and three firmware releases.

We have had cases of users downloading the BISmark firmware from the BISmark project page and installing it on their own hardware. This mode enables further growth, but presents us with the challenge of determining the identity of the users who install the software. We have deployed a registration page that forces users to register their device and meta-data about their Internet connection with our server on first-boot. Section 4.3.4 discusses the security implications of letting users install BISmark on their own hardware.

Management servers. To support the router deployment, BISmark has three types of servers:

1. *Package repository servers* decide which software should run on each router. Different routers can run slightly different sets of software because we've deployed several firmware versions and users have consented to run various experiments.

2. *Data collection servers* validate, store and serve data gathered by routers. We serve publicly accessible active measurements data from Amazon S3 and mirror all data in servers at our university.

3. *Monitoring servers* track availability and can initiate SSH connections to routers for troubleshooting.

Measurement servers. BISmark uses a fixed set of measurement servers against which it conducts standard performance measurements (*e.g.*, throughput). The validity of these experiments in many cases depends on having measurement servers that are geographically close to the deployed routers. We have been fortunate to obtain access to the globally distributed Measurement Lab (MLab) infrastructure [113]. Measurements are scheduled on the measurement servers by a central scheduler. This is to prevent overloading of servers by several concurrent requests from BISmark routers. We note that this infrastructure is used for intensive active tests such as throughput measurements. Other experiments that do not rely on a low-latency, globally distributed infrastructure do not use these servers.

4.1.3 Naming

We assign each router a unique *router identifier* which we use for data analysis, troubleshooting, and inventory; and we correlate them with all measurements we collect from the router, participant-provided details about the upstream ISP's advertised performance, the router's geographic location, and the participant's name and mailing address (used to ship the router). We do not disclose personal information except when required by law enforcement [26] (a scenario that we have not yet encountered).

Router identifiers must satisfy two requirements: (1) they must be unique across all routers and (2) they must be persistent across reboots and reflashes. Common identifiers such as manually assigned hostnames, dynamically generated tokens, or public IP addresses do not satisfy these requirements. Instead, we use the routers' MAC address, which is both unique and unchanging. We chose the interface whose address corresponds to the one printed on the back of the router, which simplifies technical support and inventory.

Unfortunately, MAC addresses pose a security risk because attackers could use them to geographically locate a router. By default, routers broadcast their MAC address to WiFi devices in the vicinity, including smartphones and collectors for Google's Street View and similar data collection projects. An attacker with access to both the router's MAC address and these databases could geolocate a router [166]. This vulnerability highlights a broader set of tradeoffs BISmark makes between privacy and transparency; Section 4.3 discusses these tradeoffs.

4.1.4 Troubleshooting

Every BISmark router runs an SSH server, which has been useful for testing and troubleshooting during development of the platform and associated experiments. Remote access has let us quickly fix critical problems that would have otherwise taken a lengthy packaging and software update cycle to fix.

Most routers are behind NATs. BISmark routers do not expose an SSH server on their WAN interface because of security concerns, and the fact that over 60% of routers in our deployment are obscured by at least one layer of Network Address Translation (NAT), which

renders the SSH server on WAN useless. Instead, BISmark routers poll the *monitoring server* by sending small UDP probes ("heartbeats") once per minute. If the server wishes to initiate an SSH session with a router, it responds with a UDP response to that effect; the router then opens an SSH tunnel forwarding a port on the monitoring server to the router's local SSH server. Administrators on the server then initiate SSH sessions to the forwarded server port. The DNS time-to-live on the monitoring server is 15 minutes, which allows us to quickly migrate the monitoring server in an emergency.

4.1.5 Software Upgrades

After we have deployed a router, we must be able to manage its software packages. Throughout the lifetime of the deployment, we have issued many bug and security fixes as package upgrades, deployed new measurement experiments by installing new packages, and rolled back faulty experiments via package removal. OpenWrt's built-in *opkg* package manager is limited and lacks several features and safeguards necessary for managing software on a large deployment of routers in the homes of non-technical users.

Software upgrades must be automatic, but without the risk of accidental installation, upgrade, or removal of packages are high. Our user base is not expected to manage the system package updates, it must be automatic. However, a single faulty or buggy package could cripple the entire deployment. Our management tools, on top of *opkg* automatically check our repositories for new packages and package updates twice a day, and install/upgrade these packages. We impose restrictions to guard against accidental installation, upgrade, and removal of packages, and also installs the correct version of packages depending on the version of the BISmark firmware installed on the router.

4.1.6 Data Collection

Experiments generate data of a wide variety of sizes and rates and upload it to data collection servers for analysis. For example, some experiments generate files as small as 45 bytes each every five seconds, while others generate files as large as 200 KB and as infrequently as once a day.

We initially used off-the-shelf file synchronization tools to upload this data, but resource,

flexibility, and reliability constraints motivated us to develop custom software. Bandwidth is scarce and may be capped; this constraints deployed experiments to minimize the data collected. Routers upload files as soon as possible because they store these files on a volatile RAM disk; both excessive wear and extreme scarcity prevent them from storing frequently generated data on persistent flash storage. To minimize the risk of data loss when they lose power or reboot, routers do not batch files for more efficient transmission.

4.2 Experimentation on BISmark

This section describes research projects that have used BISmark. We first describe the modes of collaboration that we have used since opening BISmark to external researchers in mid-2013. Because the platform is both resource constrained and on many users' critical path to the Internet, experiments on BISmark must cope with harsher conditions than most existing testbeds that support long-running deployments (*e.g.*, PlanetLab). For example, experiments must deal with nodes of highly variable connectivity and availability. Figure 6 plots the 95^{th} percentile of throughputs of homes in the deployment; we see the gamut from basic broadband (about 1 Mbps) to fiber speeds (100 Mbps). Figure 7 shows the fraction of time a router is available and online during its lifetime; about 50% of the routers are available more than 90% of the time, but a significant fraction of routers have much patchier availability.

4.2.1 Modes of Collaboration

We have been advertising BISmark to collaborators and encouraging them to run experiments on the deployment. Most of this recruiting has been through word-of-mouth, as we build confidence that we can support a larger group of researchers. In many of these cases, we have informally adopted a PlanetLab-like incentives model by asking the researchers to spearhead their own small deployment of BISmark routers in an ISP or region of interest. In certain research projects, the researchers want to do this anyhow because they have a specific group of users that they want to study. We have two modes of collaboration, which we outline below. In both cases, researchers must be comfortable with OpenWrt and embedded platform development. The data from certain active measurements are also made

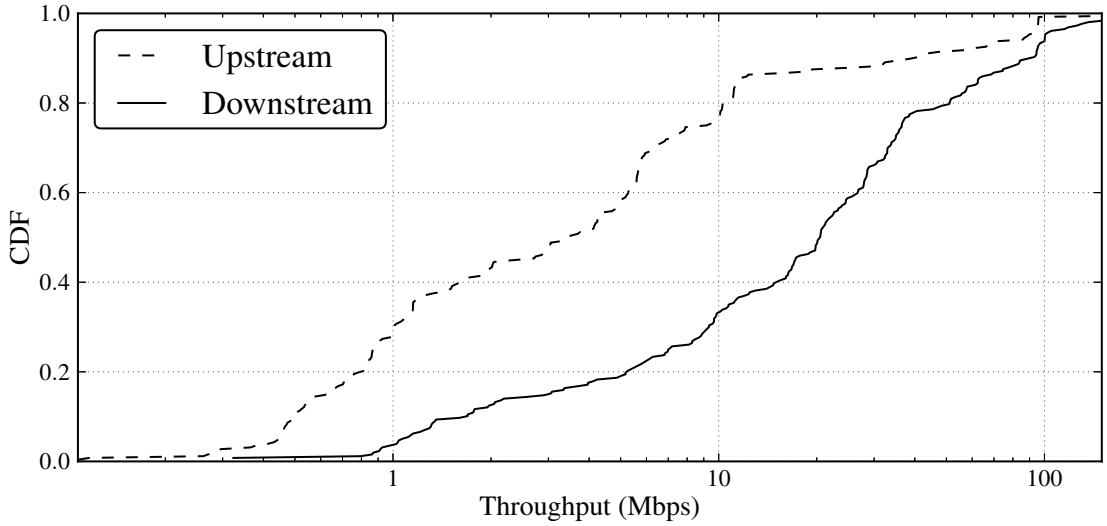


Figure 6: Downstream and upstream throughputs for routers.

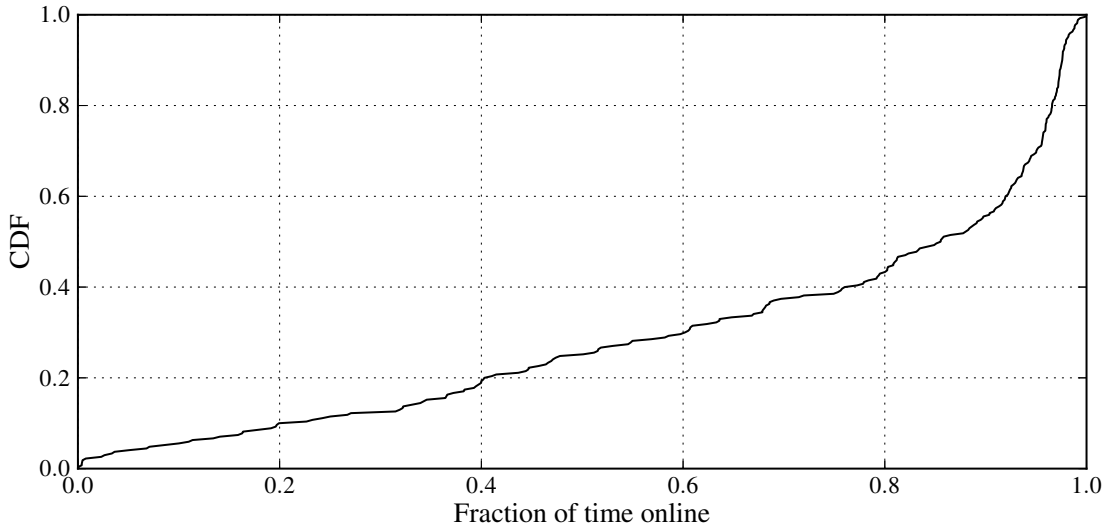


Figure 7: Distribution of the fraction of time each router is online during its deployment. We only include routers that have been online for at least a month.

public for anyone to use.

Public deployment. Collaborators run experiments on the main deployment of routers, which we manage. We control access and schedule the experiment to run in conjunction with other experiments that are already running on the deployment. This mode works well for researchers who want to run light-weight experiments from the variety of vantage points

that our deployment offers. We have enabled research from the University of Southern California in this way.

Private deployments. Researchers (or, in some cases, operators) purchase and deploy their own routers, while we provide the client software and manage back-end services. In these cases, the researchers retain a high degree of access to their routers, thereby giving them an incentive to keep their deployment running. This mode is best for researchers who want to run complex or time-consuming experiments in a small geographic area. For example, University of Cambridge has deployed more than 20 BISmark routers in under-privileged communities in Nottingham to study the mechanics of Internet sharing in such communities. We have also engaged with several ISPs who have wanted to run their own autonomous deployments.

4.2.2 Research Projects

BISmark offers the ability to study poorly understood or understudied aspects of home networks, including access link performance, application characterization, user behavior patterns, security, and wireless performance. Table 1 summarizes several experiments we are coordinating on the deployment. In many cases, we are leading (or have led) the study ourselves; more recently, we have been collaborating with the researchers who are leading the study. The latter projects are works-in-progress. The following sections describe both sets of projects in more detail. Our discussion of experiments that have been run on BISmark is not exhaustive but is intended to highlight both the capabilities and shortcomings of the platform.

Performance Characterization BISmark's placement as the hub of the home network lets it gauge performance of both local devices and upstream connectivity without confounding factors from the rest of the network.

- *Broadband performance in the US and abroad (Chapter 5).* The home access point is ideally suited for measuring access link characteristics. We characterized access link performance and the effects of access technology and customer premise equipment in

Table 1: Summary of various experiments (and publications) that BISmark has enabled to date. "WiP" denotes work in progress.

Project	**Institution(s)**	**Description**	**Publications**
Performance Characterization			
Broadband performance	Georgia Tech, University of Napoli, Université Pierre et Marie Curie, FCC/SamKnows, Research ICT Africa, National University of Sciences and Technology	Study factors affecting broadband performance in the US and in developing countries.	[45, 155], WiP
Web performance	Georgia Tech, UPMC	Characterizing and mitigating last-mile bottlenecks affecting Web performance	[159]
Home wireless performance	Georgia Tech	Studying bottlenecks and wireless pathologies in home networks	WiP
Usage and Home Network Characterization			
Home Network Characterization	Georgia Tech	Understand usage and connectivity	[73]
Home Constant Guard	Comcast	Expand Constant Guard to provide information about devices infected in home networks.	WiP
PAWS	University of Cambridge	Internet sharing in underserved communities	WiP
Topology and Connectivity Characterization			
Google Cache Measurements	University of Southern California	Study effects of Google's cache deployment on performance of Web services.	[34],WiP
Network outages and DHCP	University of Maryland	Study effects of outages on IP address allocation worldwide.	WiP
OONI/Censorship	NUST, University of Napoli	Study the extent and practice of censorship in various countries (initial focus on Pakistan)	WiP

the United States [155] using data from BISmark and the similar FCC/SamKnows deployment. Research ICT Africa (RIA) reproduced our study in South Africa [45] and expanded it to include mobile devices and 3G dongles.

- *Home network bottlenecks (Chapter 6).* We developed and deployed techniques that isolate the source of performance bottlenecks to either the access link or the wireless network, as well as tools that help us understand the nature of wireless pathologies. The home access point sits between two common sources of performance issues—the access link and the wireless network—and is therefore ideally suited for identifying and isolating problems between these locations.

- *Application performance (Chapter 7).* Because hardware limitations can prevent us from running full applications (*e.g.*, Web browsers), we often aim to emulate applications' network behavior with active measurements. Our work measuring network bottlenecks in Web performance used this technique [159]. We measured only one aspect of Web performance—the impact of the last mile. Although BISmark was suitable for this experiment, we did not (and arguably cannot) measure other aspects of Web performance, such as user perception, or the effect of object ordering or scripting on performance.

Lessons and caveats. Experiments that measure the access link by sending active probe traffic (*e.g.*, throughput tests) must not degrade performance of the home network while doing so. For users with bandwidth caps, probe traffic and data traffic (from uploading measurements to the server) should not constitute a significant fraction of the cap without the user's knowledge or consent. Some measurements such as TCP throughput require server deployments with low latency. Fortunately, Measurement Lab's global server infrastructure has helped: BISmark nodes automatically select the nearest MLab node for throughput measurements; Figure 8 shows that over 80% of nodes are within 100 ms of a measurement server. We tune TCP parameters like receive and send windows, and use multiple parallel threads to mitigate the effect of latency on throughput estimates. Applications (or their

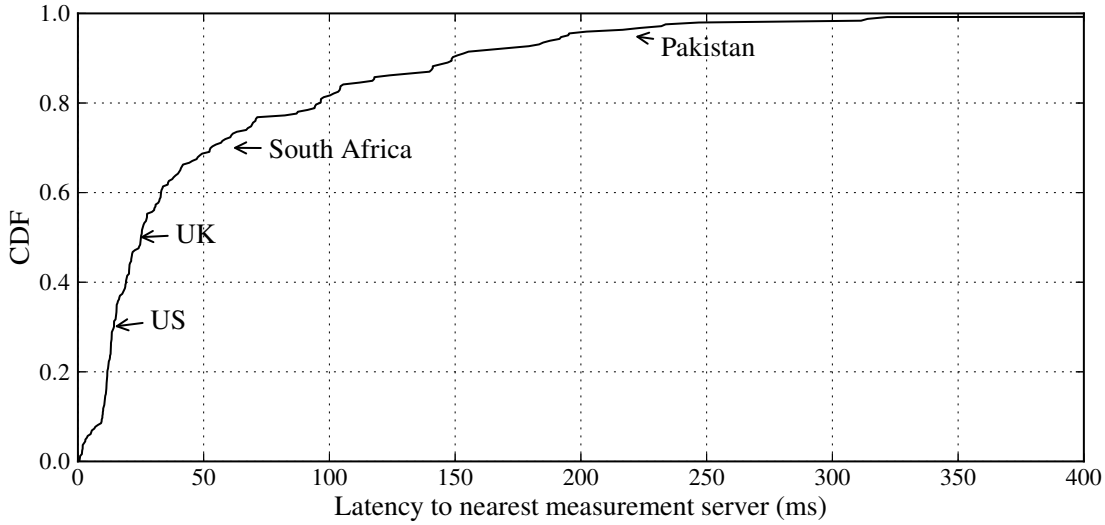


Figure 8: Distribution of latencies to the nearest measurement server from each BISmark router, with annotations of the median latency from several countries with many routers. Latencies from Pakistan are very high because the nearest server is in Europe.

emulations) must be light enough to run on the router; this might preclude certain types of applications. Therefore, while the deployment can support continuous measurements, experiments must be lightweight to exploit this capability.

Usage and Home Network Characterization Several projects leverage BISmark's view of the home network behind the NAT.

- *Home network availability, usage, and infrastructure.* This study looks at the kinds of devices home users use to access the network, how they access the network, and their usage patterns [73]. The study found interesting behavioral patterns (*e.g.*, users in developing countries turn off their home routers when not using it) and usage patterns (*e.g.*, most traffic is exchanged with a few domains).

- *Home network security.* The ability to isolate traffic from different devices behind the NAT can be used to improve security. Comcast offers a security solution called Constant Guard, which captures DNS lookups to suspicious domains to inform a user when devices in their home may be compromised. Researchers at Georgia Tech are extending BISmark to let Constant Guard identifying particular infected devices and

redirect some or all flows from suspected devices through Comcast security middleboxes via a virtual private network [25].

- *Internet usage in underprivileged communities.* The PAWS project [128] distributes BISmark routers augmented with extra measurement tools to broadband customers who volunteer to share their high-speed broadband Internet connection for free with fellow citizens. The project studies how underprivileged communities share Internet access.

Lessons and caveats. BISmark's view into the home network brings with it a new set of concerns. Resource constraints limit the amount of data that can be collected and processed on the device. Another significant concern is user privacy. For any measurement that studies user behavior, we must obtain informed consent from the user, which can be a slow and cumbersome process. We have conducted our own experiments that have required institutional review board (IRB) approval, but interesting concerns arise when BISmark serves as a host platform for experiments run from other universities that are sometimes in other countries. Even when we have permission to collect personal information, we design our experiments to collect only information necessary to answer targeted questions.

Connectivity Characterization BISmark's worldwide presence lends itself to measuring aspects of Internet connectivity.

- *Measuring Internet topology and connectivity.* We have looked at Internet availability in developed and developing countries [73], and researchers at USC are using BISmark to study the effects of Google's expanding cache deployment on the end-to-end performance of various Web services. Researchers at the University of Maryland are analyzing BISmark's UDP "heartbeat" logs (Chapter 4.1.4) to understand the effects of network outages on DHCP address allocations. Recent work explores correlated latency spikes in ISPs [143] and the extent to which interconnectivity (or lack thereof) at Internet exchange points contributes to latency inflation and degraded application performance.

- *Global measurements of censorship.* BISmark routers represent a unique opportunity to collect detailed, longitudinal data about how countries engage in censorship. Researchers in Pakistan have deployed BISmark routers in several homes to measure this phenomena; routers in other countries could also potentially collect similar measurements. Researchers at Georgia Tech are replicating OONI [67] on BISmark.

Lessons and caveats. BISmark is well-suited for connectivity measurements because of its geographical footprint, availability, and therefore its ability to run periodic measurements (time scale of minutes) over a long period of time (months, or even years). Though BISmark will likely always have fewer deployment sites than platforms such as Dasu, it can perform more kinds of measurements and continuously. Experiments that measure censorship have additional ethical concerns because in some countries, testing sites for censorship is illegal or even place the household at risk. In these cases, we must obtain informed consent, which may not be possible for users who flashed their own hardware or don't speak English.

4.3 Lessons

This section summarizes lessons we have learned during BISmark's development.

4.3.1 Recruiting Users

Convincing users to deploy BISmark routers in their homes, particularly while deploying custom hardware, is not easy. Prior to our current deployment of commodity routers, we conducted a year-long pilot study with the NOX Box, a small form-factor PC originally conceived to run the NOX OpenFlow controller on Debian Linux. We assembled the hardware from an ALIX 2D13 6-inch by 6-inch board with a 500 MHz AMD Geode processor, 256 MB of RAM, 2 GB of flash memory, three Ethernet ports, and a wireless card. Although the NOX Box's relatively unconstrained hardware and full-featured Linux distribution were a boon for rapid development, our pilot revealed several practical problems with deploying custom hardware in the field.

Lesson 1 () *Form factor matters. Users often trust commodity hardware over custom hardware simply because it is in a recognizable form.*

(a) NOX Box (b) Netgear WNDR 3800

Figure 9: We used the NOX Box for our pilot deployment and the Netgear WNDR3700/WNDR3800 for the second deployment. Unlike the NOX Box, the Netgear router looks like standard home networking equipment.

Figure 9 compares the NOX Box hardware from our pilot phase to the commodity Netgear hardware from our current deployment. The NOX Box doesn't look like a typical home router: it lacks familiar branding, has few status indicator lights, lacks labels for both the status indicators and Ethernet ports (*i.e.*, to distinguish WAN and LAN ports), and has a metal rather than plastic enclosure. These factors bred an inherent distrust of the NOX Box. We found people were generally more willing to deploy commodity hardware, even after they learned we had replaced the software on both devices.

Lesson 2 () *Users often blame BISmark for problems in their home network—deserved or not. Many users react by removing the router permanently from their network.*

Even with commodity hardware, users have heightened awareness of the BISmark router, particularly the experimental nature of the device, and therefore suspect it first when problems arise with their home network. In some cases, BISmark is indeed at fault. For example, a firmware bug causes unstable wireless connectivity on some devices, notably Apple MacBooks. In other cases, the router uncovered buffering problems elsewhere in the home

network, temporarily degrading network conditions in the process. Many times, users misconfigured the router themselves (*e.g.*, by changing firewall settings) or incorrectly blamed BISmark for upstream ISP outages or problems with end hosts (*e.g.*, older devices that lack support for WPA2.)

Regardless of the cause of these problems, many users "solve" them by removing the BISmark router from the network. This has consequences in terms of money (the router likely will not be turned on again) and time (in flashing, packaging, and shipping the router to the user).

Lesson 3 () *Home users and researchers have vastly divergent incentives. Home users want a working network, and researchers want to gather data and information. Care and effort must be invested to align these incentives.*

It is critical that our deployment strategy allows us to finance and maintain a large number of routers. PlanetLab's incentive structure (*i.e.*, hosting infrastructure for the right to run deployment-wide experiments) does not work in our case, because many of the most interesting vantage points will be hosted by users who are not networking researchers and have no interest in conducting their own experiments. We use two deployment strategies:

Free (or subsidized) router distribution. Our initial strategy has been to ship routers to acquaintances, friends of friends, and through targeted advertising in venues such as NANOG and Internet2. It is difficult and time consuming to track routers in such cases, particularly when users turn them off. Due to the cost and effort involved, individual shipments only work at relatively small scales. About 50% of routers we distributed have either never been turned on or have since been decommissioned by their users.

Federated distribution. We are now attempting a federated deployment model to expand our geographical footprint. We work with a local contact who buys or receives a shipment of routers from us, recruits volunteers and follows up with them to ensure that routers stay up. This approach worked well for a deployment of approximately 15 routers in South Africa, 20 routers in the UK, and 10 routers in Italy and Pakistan. We are now attempting similar approaches in Tunisia, and Cyprus.

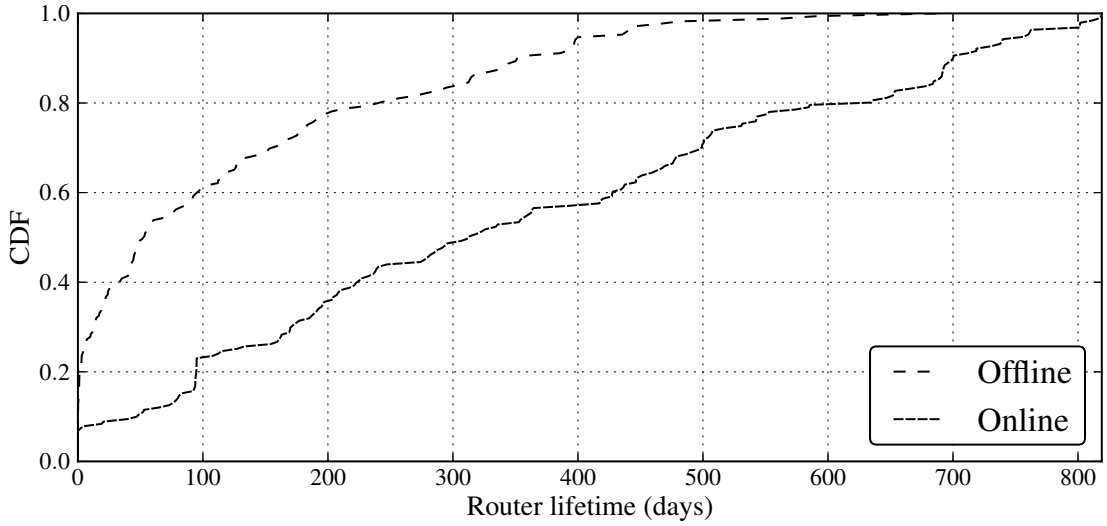


Figure 10: Distribution of router lifetime. Lifetime is the difference between the first and last time we saw the router online. For 191 routers currently online, it is how long they have been online to date. 169 are now offline, while 31 never turned on at all.

4.3.2 Sustaining the Deployment

Even after deploying BISmark routers in homes, it is a struggle to keep them online. Figure 10 shows attrition of the deployment. Nearly 25% of all routers go offline within three months, while another 25% have remained online for more than a year. We have learned many lessons, both about how to deploy reliable router software and how to keep users involved when unreliable software disrupts the user experience.

Lesson 4 () *Users must be engaged to help keep routers online. Engagement can come in a variety of forms, and may be as simple as helping them better understand their network using the data we collect.*

If users disconnect their routers, we stand to lose both the device hardware and the data. We attempt to keep users engaged by providing useful tools like the Network Dashboard [118] to visualize ISP performance. We conduct our development and data collection in the open; users can track BISmark development online [28, 134].

Lesson 5 () *Upgrading critical software in the field is risky, but the ability to upgrade other software is essential for sustainable deployment.*

The ability to upgrade non-critical software after deployment has enabled us to pursue "good-enough" software development by deploying systems that are not fully ready. In certain critical cases (*e.g.*, wireless bugs) we cannot update the software because there is a chance, however small, that a critical functionality (the wireless network, in this case) could break. Our approach has been to minimize such cases by using a well-tested base platform that can maintain basic functionality even when higher-level client and backend software malfunctions.

Lesson 6 () *Every home network has unique conditions, and usage patterns; therefore comprehensive testing before deployment is nearly impossible; bugs arise in practice.*

We aim to ensure that BISmark is foremost a stable access point, and that our custom software and experiments do not degrade user experience. The BISmark gateway is on the critical path of Internet access for at least one wired or wireless device in 92% of the homes; a malfunctioning gateway will disrupt network connectivity for those devices and, in the worst case, even completely take those devices offline. Most people have no desire to troubleshoot home networks and will readily disconnect their BISmark router if it stops working as intended.

We have one window of opportunity per user to ensure that a router is installed and working correctly. Unlike most distributed systems (*e.g.*, PlanetLab), BISmark hardware typically resides in homes of technically inexperienced users. Homes lack the support infrastructure typically available in data centers, like standardized power and network monitoring, which makes troubleshooting opaque: routers either work or not, but when they break, it is difficult to ascertain the cause. Although we have SSH access to each router to fix software problems, this access is useless in identifying power failures and network outages.

Lesson 7 () *Community support is crucial; we rely heavily on commodity hardware manufacturers and open source software developers to build reliable, usable home router hardware and software and test it in real home networks.*

Commodity hardware solves many problems we faced with custom hardware because equipment manufacturers (*e.g.*, Netgear) design hardware specifically to deploy it in the homes of

non-technical users, which exactly matches BISmark's deployment scenario. Along with the aforementioned problems with the NOX Box's appearance, commodity hardware addresses many of the NOX Box's reliability, usability, and practical problems: (a) flash storage cards failed after only 3–4 months in the field, far sooner than expected for our workload; (b) we assembled each NOX Box from components, a laborious process; and (c) the component cost for each device was approximately $250 USD, or 2–3 times the cost of a commodity wireless router. Similarly, OpenWrt's global community ensures that it is much more comprehensively tested on a variety of home routers than Debian.

Lesson 8 () *Users may want to customize router settings, but doing so may introduce security vulnerabilities.*

BISmark routers have a flexible administrative interface to help users configure the router; this is a potential security vulnerability. The router firewall in one household was accidentally disabled, opening its DNS resolver to the Internet. Attackers eventually recruited the router for amplification attacks over a period of many months, which we only discovered when the ISP notified the user of the problem. Although the disabled firewall was the culprit in this case, it led to a wholesale audit of the deployment and a spirited email exchange with the affected user. It is still unclear exactly how the firewall was disabled.

4.3.3 Experimentation

Designing and deploying measurements on BISmark has highlighted several nuances of supporting experimentation in production home networks.

Lesson 9 () *It is difficult to reconcile the need for open data with that of user privacy.*

To encourage open data, we publish measurements collected from BISmark, but only if doing so doesn't threaten user privacy. Sometimes this decision isn't obvious. For example, our original policy was to not publish routers' IP addresses in active performance measurements, but MLab's policy does not consider IP addresses as PII—we were bound by that policy when we started using the MLab platform. It is also unclear when active data measurements can yield insight into user behavior; for example, patterns in router availability

or throughput and latency measurements could indicate when users are home and using the network.

Lesson 10 () *Vetting experiments is challenging, and a poorly designed (or controlled) experiment can cripple a user's Internet connection.*

Enabling a wide range of experiments introduces management and security concerns, specifically with reviewing code, controlling access, and ensuring that experiments do not disrupt user experience by making the device unstable or consuming too much network resources.

One household had comparatively slow upstream connectivity (512 Kbps upload) and an old modem with a large buffer, where even short throughput tests can induce bufferbloat pathologies [69]. Although the household's typical workload did not stress the network often enough to expose bufferbloat in their typical usage, BISmark's periodic throughput tests saturated the buffer and rendered the Internet unusable for the duration of the test (a few seconds). The degradation was bad enough for the user to complain and stop using the device after a few weeks.

4.3.4 Security

BISmark routers should not compromise either home network security the integrity of the platform. Although we try to minimize the possibility of security vulnerabilities by adopting industry-standard software and protocols where ever possible, some attacks against BISmark's backend infrastructure are still possible.

Lesson 11 () *Users have physical access to hardware and can modify firmware; this imposes new security challenges.*

BISmark's backend is subject to two broad security threats. The first is denial-of-service attacks, where malicious users could attempt to exhaust server resources for processing legitimate routers or measurements. Attackers could impersonate other users or even mount Sybil attacks to create many fake routers. Several backend components employ rate limiting, but these limits generally only protect against errant behavior of non-malicious clients. Thus far, we have deliberately chosen *not* to fix this class of vulnerabilities. To drive adoption,

we initially allowed users to install BISmark on their own hardware; without registering or authenticating their router. We now require registration and authentication.

Other attacks could contribute malicious data to influence conclusions. Mitigating such attacks requires instrumenting routers with Trusted Platform Modules running signed executables to generate signed measurement data. Attackers have physical access to router hardware and the software source code, so we rely on social measures: we try to deploy to trusted users and assume that they won't collude. Because anyone can install BISmark on their own hardware, we treat measurements from such routers with greater suspicion.

4.4 Takeaways

Although we did not initially plan to build (and maintain) such a large testbed, we realized the need for it 2009 when we began a study of access network performance. We recognized the variety of uses for a programmable testbed in home networks, and we also discovered that other researchers and operators share our curiosity. As BISmark continues to expand in terms of size and the diversity of experiments that it hosts, we will need to continually re-evaluate many of our design decisions. We believe our experiences thus far offer a unique perspective in comparison to existing long-running testbeds and useful lessons for others who perform research in home networks.

4.5 Acknowledgments

BISmark is not a solo effort. There have been many contributors to the project. Developers, volunteers, and supporters (with infrastructure); this project would not be where it is now, without their singular contributions. Sam Burnett, Thomas Copeland, Walter de Donato, Hyojoon Kim, Abhishek Jain, Aman Jain, Guilherme Martins, Brian Poole, Alfred Roberts, Dave Taht, and Stephen Woodrow were all instrumental in developing BISmark.

CHAPTER 5

BROADBAND INTERNET PERFORMANCE: A VIEW FROM THE GATEWAY

As Broadband Internet penetration increases, and vital services move online, benchmarking the performance of these networks becomes critical. Accordingly, the Federal Communication Commission (FCC) is actively developing performance-testing metrics for access providers [29, 65, 163]. These efforts affect every stakeholder in the end-to-end path. Users would like to know whether their performance matches what their ISP promises, ISPs would like to ensure that they do meet their promises, application providers, whose business might depend on delivering good performance, and regulators, as they plan for tomorrow's Internet.

Benchmarking home Internet performance is not simple. One-time "speed tests" such as the ones provided by Ookla are the most commonly used and understood metric of Internet performance. There exist countless other tools to measure performance [39, 111, 116, 148]. However, throughput by itself does not mean much; gamers might prefer lower latency to higher throughput, for example. Previous work has studied various aspects of access networks, particularly dowload and upload throughput [56,99]; others have uncovered previously unknown prolems such as buffering [99], and that DSL links often have high latency [110]. These studies have shed some light on access-link performance, but they have typically have two drawbacks: a) they usually run one-time measurements. Without continual measurements of the same access link, these tools cannot establish a baseline performance level or observe how performance varies over time, and b) they either run these measurements from an end-host inside the home (from the "inside out") or from a server on the wide-area Internet (from the "outside in"). Because these tools run from end-hosts, they cannot analyze the effects of confounding factors such as home network cross-traffic, the wireless network, or end-host configuration.

In this chapter, we present the case for measuring the characterizing broadband Internet performane from the home gateway. The home gateway connects the home network to the user's modem; taking measurements from this vantage point allows us to control the effects of many confounding factors, such as the home wireless network and load on the measurement host (Section 5.1). The home gateway is always on; it can conduct unobstructed measurements of the ISP's network and account for confounding factors in the home network. The drawback to measuring access performance from the gateway, of course, is that deploying gateways in many homes is incredibly difficult and expensive. We were fortunate to be able to take advantage of the broad deployment by the FCC as part of the ongoing broadband study. We also used an early stage deployment of BISmark in Atlanta, GA, for this study.

We perform our measurements using two complementary deployments; the first is a large FCC-sponsored study, operated by SamKnows, that, at the time of this study, had installed gateways in over 4,200 homes across the United States, across many different ISPs. The second, BISmark, at the time of the study, had been deployed in 16 homes across three ISPs in Atlanta. The SamKnows deployment provides a large user base, as well as diversity in ISPs, service plans, and geographical locations. We designed the BISmark deployment to allow us to access the gateway remotely and run repeated experiments to investigate the effect of factors that we could not study in a larger "production" deployment. Both deployments run a comprehensive suite of measurement tools that periodically measure throughput, latency, packet loss, and jitter.

We characterize access network throughput (Section 5.2) and latency (Section 7.2.3) from the SamKnows and BISmark deployments. We explain how our throughput measurements differ from common "speed tests" and also propose several different latency metrics. When our measurements cannot fully explain the observed behavior, we model the access link and verify our hypotheses using controlled experiments. We find that the most significant sources of throughput variability are the access technology, ISPs' traffic shaping policies, and congestion during peak hours. On the other hand, latency is mostly affected by the quality of the access link, modem buffering, and cross-traffic within the home.

This study offers many insights into both access network performance and the appropriate measurement methods for benchmarking home broadband performance. Our study has three high-level lessons, which we expand on in Section 5.4:

- ISPs use different policies and traffic shaping behavior that can make it difficult to compare measurements across ISPs.
- There is no "best" ISP for all users. Different users may prefer different ISPs depending on their usage profiles and how those ISPs perform along performance dimensions that matter to them.
- A user's home network equipment and infrastructure can significantly affect performance.

As the first in-depth analysis of home access network performance, this study offers insights for users, ISPs, and policymakers. Users and ISPs can better understand the performance of the access link, as measured directly from the gateway; ultimately, such a deployment could help an ISP differentiate performance problems within the home from those on the access link. Our study also informs policy by illustrating that a diverse set of network metrics ultimately affect the performance that a user experiences. The need for a benchmark is clear, and the results from this study can serve as a principled foundation for such an effort.

5.1 Measurement Infrastructure

We describe the measurement infrastructure that we deployed and the datasets that we collected. We first motivate the need for deploying measurement infrastructure directly at the gateway; then, we describe the *SamKnows* and *BISmark* (Broadband Internet Service benchMark) gateway deployments.

5.1.1 Why a Gateway?

We briefly discussed the advantages of using gateway devices over the other techniques in Chapters 1 and 3. To recap:

Table 2: Confounding factors and how we address them.

Factor	How we address it
Wireless Effects	Use a wired connection to modem.
Cross Traffic	Measure cross traffic and avoid it/account for it.
Load on gateway	Use a well-provisioned gateway.
Location of server	Choose a nearby server.
End-to-end path	Focus on characterizing the last mile.
Gateway configuration	Test configuration in practice and controlled settings.

- *Direct measurement* of the ISP's access link: the gateway sits behind the modem; between the access link and all other devices at the home network as shown in Figure11. This allows us to isolate the effect of confounding factors such as wireless effects and cross traffic.
- *Continual/longitudinal measurements*, which allow us to meaningfully characterize performance of ISPs for individual users.
- *The ability to instrument a single home with different hardware and configurations*, which allows us to explore the effects of multiple factors on performance. In some deployments, we were even able to swap modems to study their effect on performance, holding all other conditions about the network setup equal.

Table 2 summarizes the challenges involved in conducting such a study, and how deploying gateways solves them. We now describe the two gateway deployments in our study.

5.1.2 Gateway Deployments

The FCC/SamKnows gateway deployment collected data from over 4,200 users across different ISPs in the United States, as of January 2011. This deployment currently has over 10,000 users. Our goal in using the measurements from this deployment is to achieve *breadth*: we aim to classify a large set of users across a diverse set of ISPs and geographical locations. The second, the BISmark deployment, collects measurements from a smaller, focused group of users from different ISPs and service plans in Atlanta. Our goal with the measurements from this deployment is to achieve *depth*: this platform allows us to take measurements with detailed knowledge of how every gateway is deployed; we can also take

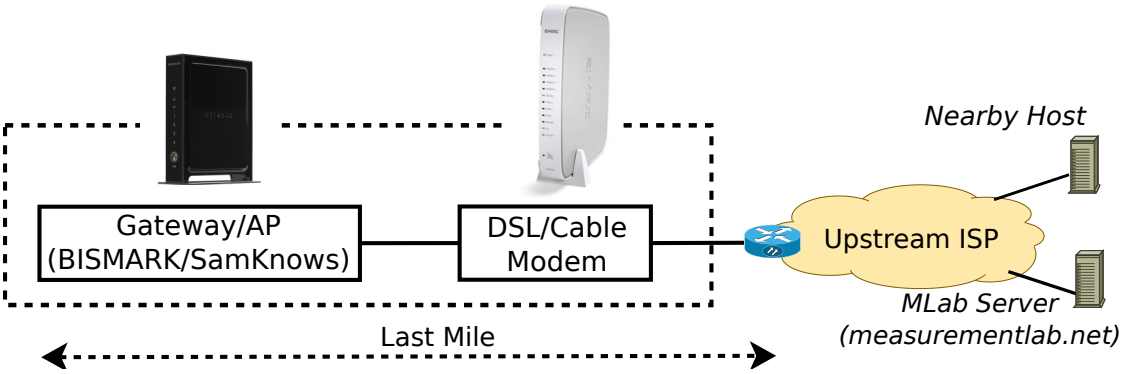


Figure 11: Our gateway device sits directly behind the modem in the home network. They take measurements both to the *last mile router* (first non-NAT IP hop on the path) and to wide area hosts.

repeated measurements and conduct specific experiments from the same deployment with different settings and configurations.

Gateway deployments entail significant challenges concerning the resource constraints of the gateway platform and the need to remotely maintain and manage the devices (especially because these devices are deployed in homes of "real users"); we omit discussion of these logistical challenges due to lack of space and instead focus on the details of the platforms and the measurements we collect.

SamKnows SamKnows specializes in performance evaluation of access networks; it has studied access ISP performance in the United Kingdom and has now contracted with the FCC for a similar study in the United States. SamKnows deployed gateways in each participant's home either directly behind the home user's router or behind the home wireless router; the devices can be updated and managed remotely. The gateway is a Netgear WNR3500L RangeMax Wireless-N Gigabit router with a 480 MHz MIPS processor, 8 MB of flash storage, and 64 MB of RAM. We use active measurement data from the SamKnows study from December 14, 2010 to January 14, 2011. This dataset comprises measurements from 4,200 devices that are deployed across sixteen different ISPs and hundreds of cities in the United States. The volunteers for the study were recruited through `http://www.testmyisp.com`. Figure 12 shows a map of the deployment.

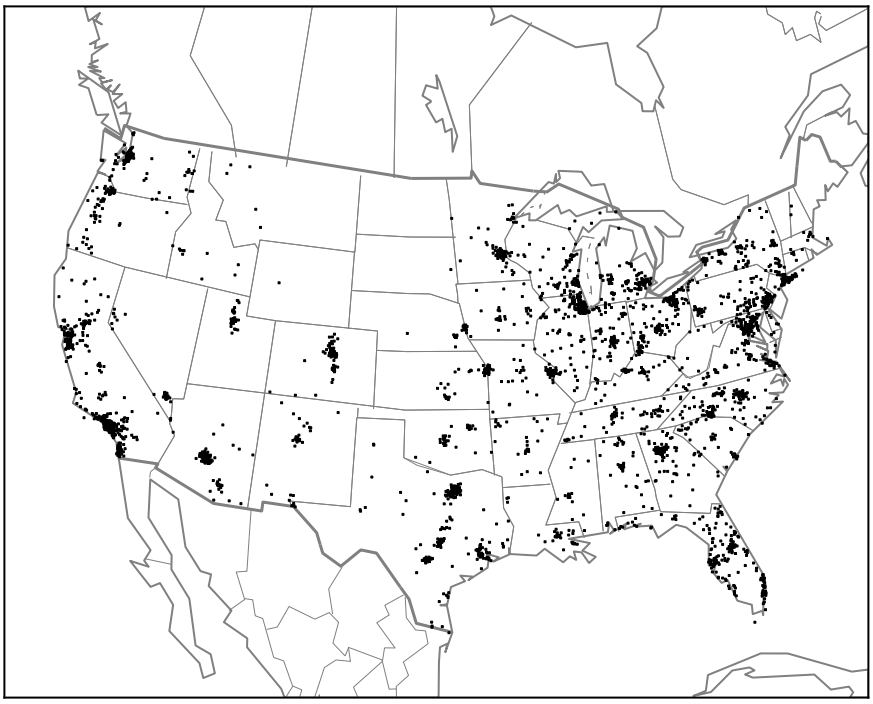


Figure 12: SamKnows deployment: 4,200 devices, 16 ISPs.

Table 3 lists the ISPs that we study, the number of gateways deployed in them, and the number of gateways that report more than 100 throughput measurements. Gateways are rolled out in phases. These devices perform measurements less aggressively when users are sending a lot of traffic. Therefore, not all gateways report data for the entire duration of the study. When we report averages and 95^{th} percentile values for some metric, we only consider gateways that have reported more than 100 measurements for that metric. We also only consider the eight ISPs with the most gateways.

Table 4 shows the active measurements that we use from the SamKnows deployment; some of these (*e.g.*, *last mile latency*) were inspired from our experience running them on BISmark. The gateways conduct upstream and downstream measurements to servers hosted at Measurement Lab [113] about once every two hours.

There are many ways to measure throughput, though there is no standard method.

Bauer *et al.* list several notions of "broadband speed": *capacity* is the total carrying capacity of the link; and the *bulk transfer capacity* is the amount of data that can be transferred along a path with a congestion-aware protocol like TCP. In Section 5.2.1, we evaluate several methods for measuring these metrics.

The SamKnows gateways measure bulk transfer capacity using an HTTP client that spawns three parallel threads; this approach increases the likelihood of saturating the access link. The software first executes a "warmup" transfer until throughput is steady to ensure that the throughput measurements are not affected by TCP slow start. The download tests that follows use the same TCP connection to exploit the "warmed up" session. The tests last for about 30 seconds; the software reports snapshots of how many bytes were transferred for every five-second interval.

The gateways also measure different aspects of latency: (1) end-to-end latency; (2) latency to the first IP hop inside the ISP (*last mile latency*); and (3) latency coinciding with an upload or download (*latency under load*). They measure end-to-end latency in two ways: (1) Using a UDP client that sends about six hundred packets an hour to the servers and measures latency and packet loss, and (2) using ICMP ping to the same set of servers at the rate of five packets per hour. To measure latency under load, the gateway measures end-to-end latency during both the upload and the download measurements. They also measure jitter based on RFC 5481 [115] and the time to download the home page of ten popular websites. Before any test begins, the measurement software checks whether cross traffic on the outgoing interface exceeds 64 Kbits/s down or 32 Kbits/s up; if traffic exceeds this threshold, it aborts the test.

BISmark We described the BISmark deployment in detail in Chapter 4. In this study, we used a pilot version of BISmark with the NOX Box hardware. We had 16 users in Atlanta, GA, USA, spread across three ISPs (AT&T, Comcast, and Clear) at the time of this study; we use data from AT&T and Comcast in this study (Table 3). The AT&T users formed the most diverse set of users in the deployment, with five distinct service plans. We use data from the same period as the SamKnows study.

Table 3: The SamKnows and BISmark deployments. *Active* deployments are those that report more than 100 download throughput measurements over the course of our study.

ISP	**Technology**	**SamKnows**		BISmark
		Total	**Active**	Total
Comcast	Cable	864	560	4
AT&T	DSL/FTTN	787	335	10
TimeWarner	Cable	690	381	-
Verizon	DSL/FTTP	551	256	-
Cox	Cable	381	161	-
Qwest	DSL/FTTN	265	117	-
Charter	Cable	187	51	-
Cablevision	Cable	104	53	-

Table 4: Active measurements periodically collected by the SamKnows and BISmark deployments.

Parameter	**Type**	**Prot.**	**Freq.**	**Comments**
SamKnows: 4,200 devices, 16 ISPs				
Latency	End-to-end	UDP	600 pkts/hr	MLab
	End-to-end	ICMP	5 pkts/hr	MLab
	Last-mile	ICMP	5 pkts/hr	First IP hop
	Upstream load	ICMP	2 hours	During upload
	Downstream load	ICMP	2 hours	During download
Loss	End-to-end	UDP	600 pkts/hr	MLab
Downstream Throughput	Multi-threaded HTTP	TCP	2 hours	MLab, idle link
Upstream Throughput	Multi-threaded HTTP	TCP	2 hours	MLab, idle link
Jitter	Bi-directional	UDP	1 hour	500pkts/30sec
Web GET	HTTP	TCP	1 hour	Alexa sites
BISmark: 17 devices, 3 ISPs				
Latency	End-to-end	ICMP	5 min	Host
	Last-mile	ICMP	5 min	First IP hop
	Upstream load	ICMP	30 min	During upload
	Downstream load	ICMP	30 min	During download
Packet loss	End-to-end	UDP	15 min	D-ITG
Jitter	End-to-end	UDP	15 min	D-ITG
Downstream Throughput	Single-thread HTTP	TCP	30 min	curl get to Host
	Passive throughput	N/A	30 min	/proc/net/dev
	Capacity	UDP	12 hrs	ShaperProbe
Upstream Throughput	Single-thread HTTP	TCP	30 min	curl put to Host
	Passive throughput	N/A	30 min	/proc/net/dev
	Capacity	UDP	12 hrs	ShaperProbe

Table 4 lists the measurements that BISmark collects. We collect throughput, latency, packet loss, and jitter measurements.

BISmark measures *bulk transfer capacity* by performing an HTTP download and upload for 15 seconds using a single-threaded TCP connection once every 30 minutes, regardless of cross traffic. We do this to have more readings, and to account for cross-traffic, we count bytes transferred by reading directly from `/proc/net/dev`, and compute the "passive throughput" as the byte count after the HTTP transfer minus the byte count before the transfer, divided by the transfer time. This yields the combined throughput of the HTTP transfer and the cross traffic. To measure *capacity*, we run ShaperProbe [148] once every twelve hours to measure UDP capacity. The gateways measure end-to-end latency to a nearby wide-area host, last-mile latency, and latency-under load to the last-mile router. They also measure packet loss and jitter using the D-ITG tool [30]. The gateways perform each measurement at the frequency presented in Table 4 regardless of cross traffic. All measurements are synchronized to avoid overlapping towards the same measurement server.

5.2 Understanding Throughput

We study throughput measurements from both the SamKnows and BISmark deployments. We first explore how different mechanisms for measuring throughput can generate different results and offer guidelines on how to interpret them. We then investigate the throughput users achieve on different access links, the consistency of throughput obtained by users, and the factors that affect it. Finally, we explore the effects of ISP traffic shaping and the implications it holds for throughput measurement.

5.2.1 Interpreting Throughput Measurements

Users of access networks are often interested in the throughput that they receive on uploads or downloads, yet the notion of "throughput" can vary depending on how, when, and who is measuring it. For example, a sample run of `www.speedtest.net` in an author's home, where the service plan was 6Mbits/s down and 512Kbits/s up, reported a downlink speed of 4.4 Mbits/s and an uplink speed of 140 Kbits/s. Netalyzr reported 4.8 Mbits/s and

430 Kbits/s. Long-term measurements (from the SamKnows gateway deployed in that author's home) paint a different picture: the user achieves 5.6 Mbits/s down and 452 Kbits/s up. Both `www.speedtest.net` and Netalyzr measurements reflect transient network conditions, as well as other confounding factors. Users cannot complain to their ISPs based solely on these measurements. Although measuring throughput may seem straightforward, our results in this section demonstrate the extent to which different measurement methods can produce different results and, hence, may result in different conclusions about the ISP's performance.

We compare several methods for measuring upstream and downstream throughput from Table 4. We normalize the values of throughput by the service plan rates advertised by the ISP so that we can compare throughput across access links where users have different service plans.

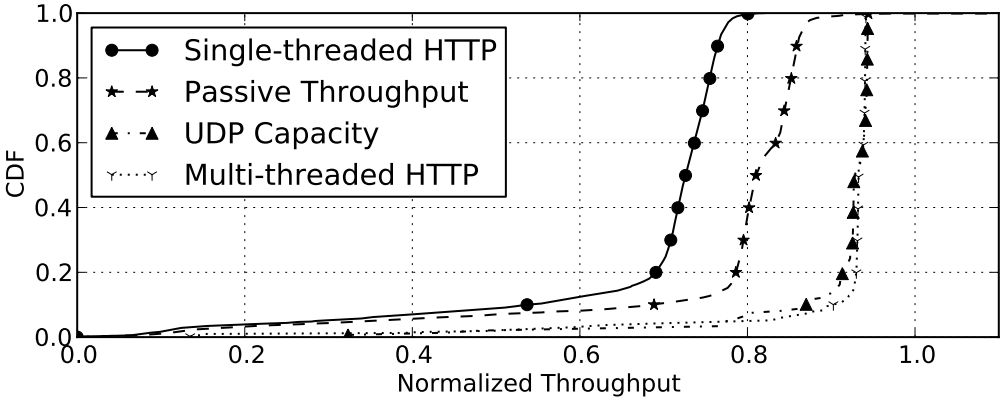


Figure 13: Comparison of various methods of measuring throughput. (SamKnows and BISmark)

Throughput measurement techniques—even commonly accepted ones—can yield variable results. We perform comparisons of throughput measurement techniques in two locations that have deployed both the SamKnows and BISmark gateways (we are restricted to two due to the logistical difficulty in deploying both gateways in the same location). In both cases, the ISP is AT&T, but the service plans are different (6 Mbits/s down and 512 Kbits/s up; and 3 Mbit/s down and 384 Kbits/s up). Figure 13 shows a CDF of the normalized throughput reported by the four methods we presented in Table 4. Each data point in

the distribution represents a single throughput measurement by a client. A value of 1.0 on the x-axis indicates that the throughput matches the ISP's advertised rate. None of the four methods achieve that value. This could be due to many factors: the sync rate of the modem to the DSLAM; layer-2 framing overhead on the line; or overhead from the measurement techniques themselves. The throughput achieved by multiple parallel TCP sessions comes closer to achieving the advertised throughput. UDP measurements (obtained from ShaperProbe) also produce consistent measurements of throughput that are closer to the multi-threaded TCP measurement. A single-threaded TCP session may not be able to achieve the same throughput, but accounting for cross traffic with passive measurements can provide a better estimate of the actual achieved throughput.

The behavior of single-threaded TCP measurements varies for different access links. We compare the passive throughput for two BISmark users with the *same ISP and service plan* (AT&T; 3 Mbits/s down, 384 Kbits/s up) who live only a few blocks apart. Figure 14 shows that User 2 consistently sees nearly 20% more throughput—much closer to the advertised rate—than User 1. One possible explanation for this difference is the loss rates experienced by these two users; User 1 suffers more loss than User 2 (0.78% vs. 0.20% on the downlink and 0.24% vs. 0.06% on the uplink). Their baseline latencies differ by about 16 milliseconds (8 ms vs. 24 ms). We confirmed from the respective modem portals that User 1 has interleaving disabled and that User 2 has interleaving enabled. Therefore, User 2 is able to recover from noisy access links that cause packet corruption or losses. Single-threaded downloads are more adversely affected by the loss rate on the access link than multi-threaded downloads (even when accounting for cross traffic); reducing the loss rate (*e.g.*, by interleaving) can improve the performance of a single-threaded download. For the rest of the paper, we consider only multi-threaded TCP throughput.

Takeaway: Different throughput measurement techniques capture different aspects of throughput. A single-threaded TCP session is sensitive to packet loss. Augmenting this measurement with passive usage measurements improves its accuracy. Multi-threaded TCP and the UDP capacity measurements measure the access link capacity more accurately and are more robust to loss.

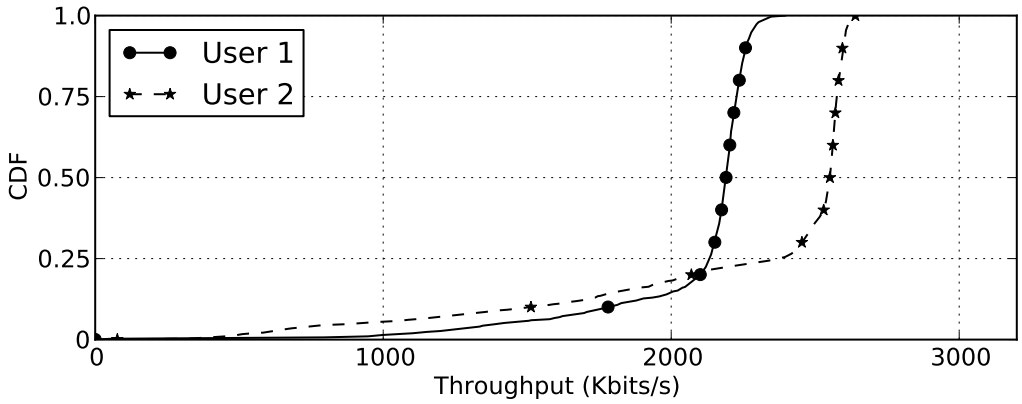


Figure 14: Users with the same service plan but different loss profiles see different performance. User 1 has higher loss and sees lower performance. (BISmark)

5.2.2 Throughput Performance

We investigate the throughput obtained by users in the SamKnows deployment. We then study the consistency of their performance.

What performance do users achieve? Figure 15 shows the average download and upload speeds obtained by each user in the SamKnows dataset. Each point in the scatter plot shows the average performance obtained by a single user in the deployment. Clusters of points in the plot reveal common service plans of different ISPs, identified in the plot by labels. In general, these results agree with the findings from both Netalyzr [99] and Dischinger *et al.* [56], although our dataset also contains Verizon FiOS (FTTP) users that clearly stand out, as well as other more recent service offerings (*e.g.*, AT&T U-Verse). Although the statistics do show some noticeable clusters around various service plans, there appears to be considerable variation even within a single service plan. We seek to understand and characterize both the performance variations and their causes. We do not yet have access to the service plan information of each user, so we focus on how and why throughput performance varies, rather than whether the measured values actually match the rate corresponding to the service plan.

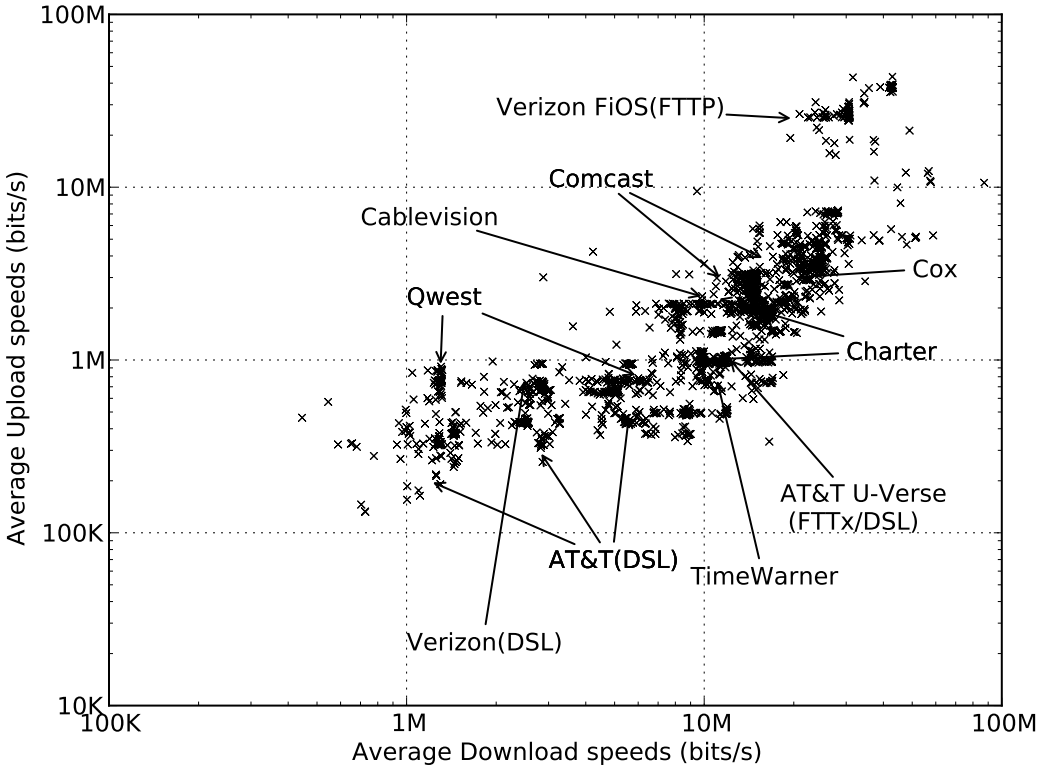
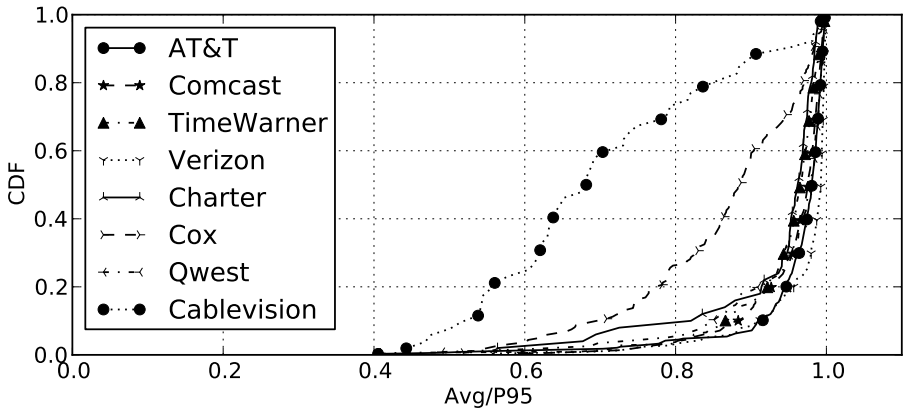


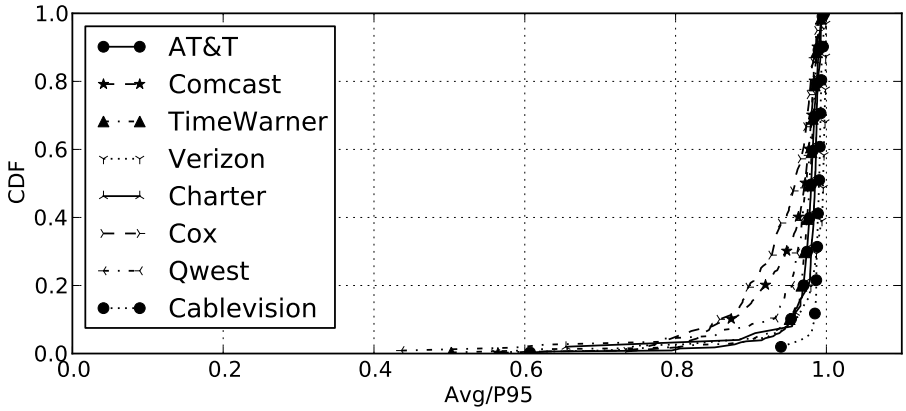
Figure 15: Average download rate versus the average upload rate obtained by individual users in the dataset. (SamKnows)

Do users achieve consistent performance? We analyze how consistently users in the SamKnows achieve their peak performance deployment using the $Avg/P95$ metric, which we define as the ratio of the average upload or download throughput obtained by a user to the 95^{th} percentile of the upload or download throughput value obtained by the same user. Higher values for these ratios reflect that users' upload and download rates that are more consistently close to the highest rates that they achieve; lower values indicate that user performance fluctuates.

Figure 16a shows the CDF of the Avg/$P95$ metric for each user; Figure 16b shows the same metric for uploads. Most users obtain throughput that is close to their 95^{th} percentile value. Users of certain ISPs (*e.g.*, Cox, Cablevision) experience average download throughput that is significantly less than their 95^{th} percentile. (Both ISPs have more than 50 active users in our data set; see Table 3). Upload throughput performance is more



(a) Download throughput is mostly consistent, with some exceptions.

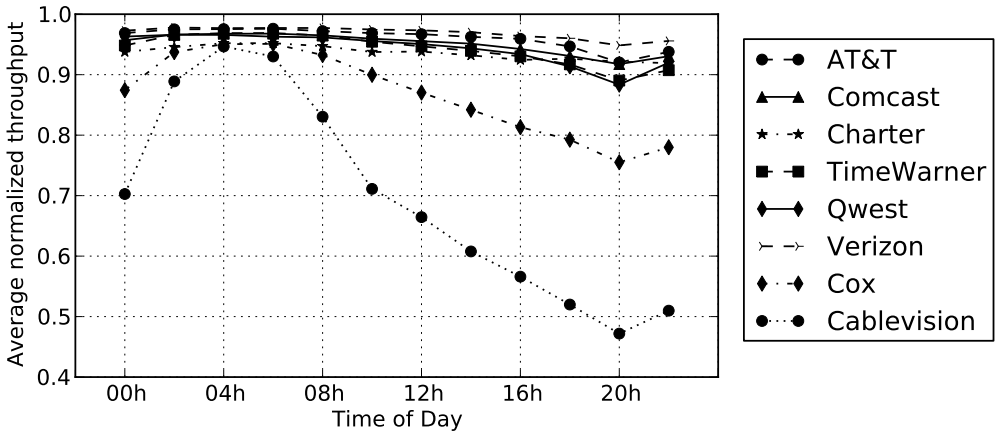


(b) Upload throughput is consistent across ISPs.

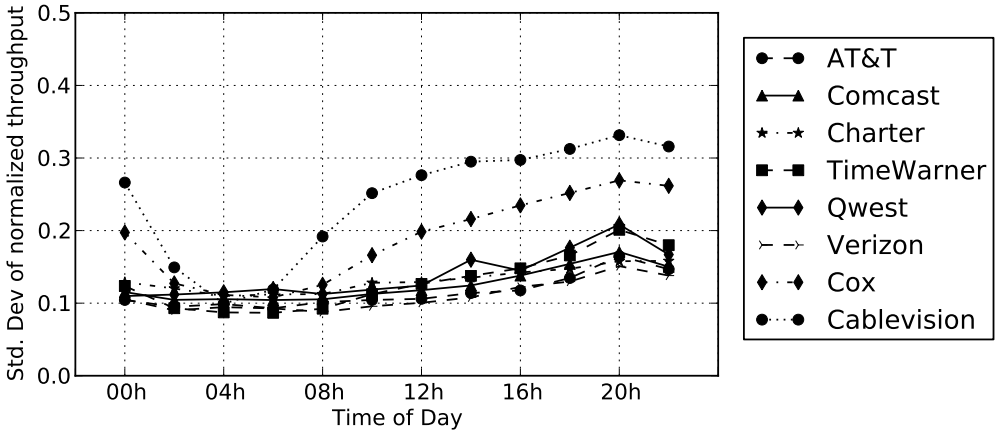
Figure 16: Consistency of throughput performance: The average throughput of each user is normalized by the 95^{th} percentile value obtained by that user. (SamKnows)

consistent across ISPs. The big difference between download rates and upload rates for popular service plans could account for the fact that upstream rates are more consistent than downstream rates. We also studied the Median/$P95$ performance; which is similar to Avg/$P95$, and so we do not show them. Our results suggest that upload and download throughput are more consistent than they were when Dischinger *et al.* performed a similar study few years ago [56], especially for some cable providers.

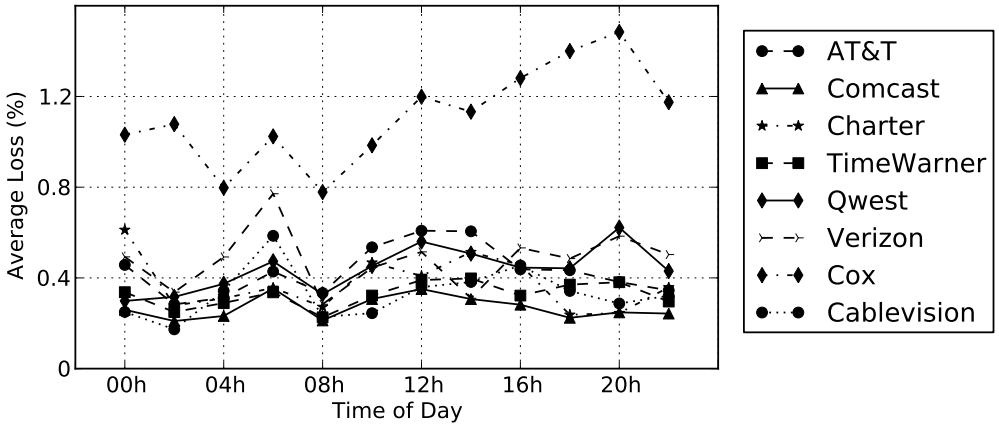
Why is performance sometimes inconsistent? One possible explanation for inconsistent download performance is that the access link may exhibit different performance characteristics depending on time of day. Figure 17a shows the Avg/$P95$ metric across the



(a) The biggest difference between peak and worst performance is about 40%.



(b) The standard deviation of throughput measurements increases during peak hours, most significantly for ISPs that see lower throughputs at peak hours.



(c) Loss increases during peak hours for Cox. Other ISPs do not see this effect as much.

Figure 17: Time of day is significant: The average download throughput for Cablevision and Cox users drops significantly during the evening peak time. Throughput is also significantly more variable during peak time. (SamKnows)

time of day. We obtain the average measurement reported by each user at that particular time of day and normalize it with the 95^{th} percentile value of that user over all reports. Cablevision users see, on average, a 40% drop in performance from early morning and evening time (when users are likely to be home). For Cox, this number is about 20%. As the figure shows, this effect exists for other ISPs to a lesser extent, confirming prior findings [56]. Because we do not know the service plan for each user, we cannot say whether the decrease in performance for Cox and Cablevision represents a drop below the service plans for those users (*e.g.*, these users might see rates higher than their plan during off-peak hours). Figure 17b shows how the standard deviation of normalized throughput varies depending on the time of day. Performance variability increases for *all* ISPs during peak hours. Figure 17c shows the loss behavior for different times of day; although most ISPs do not see an increase in loss rates during peak hours, Cox does. This behavior suggests that some access ISPs may be under-provisioned; those ISPs for which users experience poor performance during peak hours may be experiencing congestion, or they may be explicitly throttling user traffic during peak hours.

Takeaway: Although there is no significant decrease in performance during peak hours, there is significant variation. A one-time "speed test" measurement taken at the wrong time could likely report misleading numbers that do not have much bearing on the long-term performance.

5.2.3 Effect of Traffic Shaping on Throughput

ISPs shape traffic in different ways, which makes it difficult to compare measurements across ISPs, and sometimes even across users within the same ISP. We study the effect of PowerBoost (Chapter 2.1) across different ISPs, time, and users. We also explore how Comcast implements PowerBoost.

Which ISPs use PowerBoost, and how does it vary across ISPs? The SamKnows deployment performs throughput measurements once every two hours; each measurement lasts 30 seconds, and each report is divided into six snapshots at roughly 5-second intervals for the duration of the 30-second test (Section 5.1). This measurement approach allows us

to see the progress of each throughput measurement over time; if PowerBoost is applied, then the throughput during the last snapshot will be less than the throughput during the first. For each report, we normalize the throughput in each period by the throughput reported for the first period. Without PowerBoost, we would expect that the normalized ratio would be close to one for all intervals. On the other hand, with PowerBoost, we expect the throughput in the last five seconds to be less than the throughput in the first five seconds (assuming that PowerBoost lasts less than 30 seconds, the duration of the test). Figure 18 shows the average progression of throughput over all users in an ISP: the average normalized throughput decreases steadily. We conclude that most cable ISPs provide some level of PowerBoost for less than 30 seconds, at a rate of about 50% more than the normal rate. Cablevision's line is flat; this suggests that either it does not provide PowerBoost, or it lasts well over 30 seconds consistently, in which case the throughput test would see only the PowerBoost effect. The gradual decrease, rather than an abrupt decrease, could be because PowerBoost durations vary across users or that the ISP changes PowerBoost parameters based on network state. From a similar analysis for uploads (not shown), we saw that only Comcast and Cox seem to provide PowerBoost for uploads; we observed a decrease in throughput of about 20%. Dischinger *et al.* [56] also reported PowerBoost effects, and we also see that it is widespread among cable ISPs. For the DSL ISPs (not shown), the lines are flat.

Takeaway: Many cable ISPs implement PowerBoost, which could distort speedtest-like measurements. While some people may be only interested in short-term burst rates, others may be more interested in long-term rates. Any throughput benchmark should aim to characterize both burst rates and steady-state throughput rates.

Do different users see different PowerBoost effects? Using BISmark, we study Comcast's use of PowerBoost in depth. According to Comcast [50], their implementation of PowerBoost provides higher throughput for the first 10 MBytes of a download and the first 5 MBytes of an upload. We measure the shaped throughput for download and upload at the receiver using `tcpdump`. Because our tests are intrusive, we conducted them only a

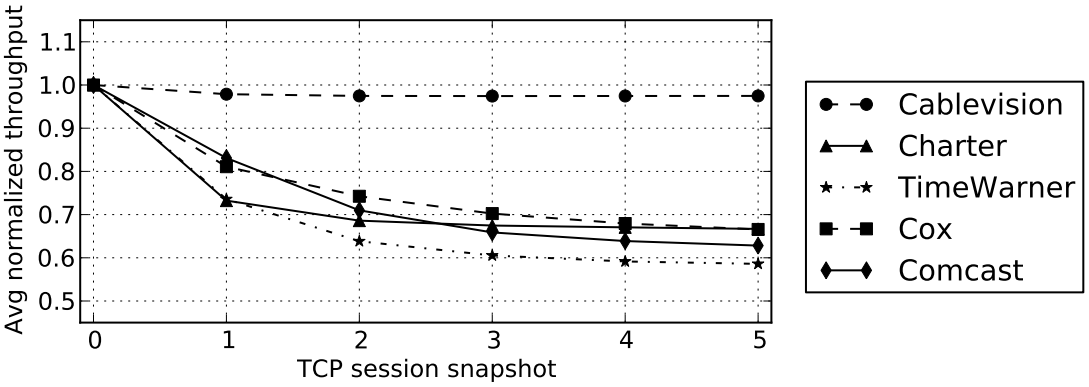
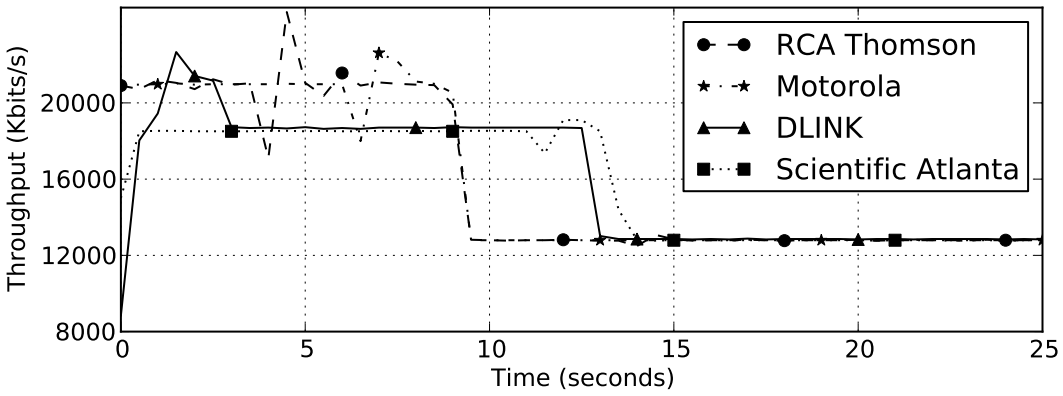


Figure 18: The average throughput obtained during the course of the measurement goes down significantly for the ISPs that enable PowerBoost. (SamKnows)

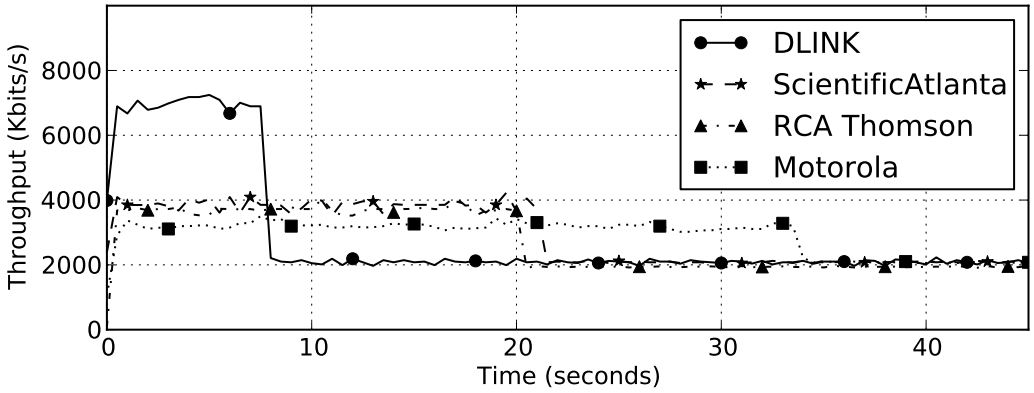
few times; however the results do not vary with choice of traffic generators or ports.

Figure 19 shows the observed throughput for four users for both download and uploads. All four users see PowerBoost effects, but, surprisingly, we see many different profiles even in such a small subset of users. Figure 19a shows download profiles for each user (identified by the modem they use; while the modem doesn't have an effect on burst rates, it does have an effect on buffering latencies as we show in Section 7.2.3). The user with a D-LINK modem sees a peak rate of about 21 Mbits/s for 3 seconds, 18.5 Mbits/s for a further ten seconds, and a steady-state rate of 12.5 Mbits/s. The Motorola user sees a peak rate of 21 Mbits/s for about 8 seconds. The PowerBoost technology [51] provides token buckets working on both packet and data rates; it also allows for dynamic bucket sizes. The D-LINK profile can be modeled as a cascaded filter with rates of 18.5 Mbits/s and 12.5 Mbits/s, and buffer sizes of 10MBytes and 1Mbyte respectively, with the line capacity being 21Mbits/s. We see varying profiles for uploads as well, although we only see evidence of single token buckets (Figure 19b). The D-LINK user sees about 7 Mbits/s for 8 seconds, Scientific Atlanta and Thomson users see about 4 Mbits/s for 20 seconds, and the Motorola user sees about 3.5Mbits/s for nearly 35 seconds. Because our results do not vary with respect to the packet size, we conclude that Comcast does not currently apply buckets based on packet rates.

Takeaway: Depending on how throughput measurements are conducted and how long



(a) PowerBoost download behavior for 4 users.



(b) PowerBoost upload behavior for 4 users.

Figure 19: The level and duration of the burstiness is different for users with different modems, suggesting different shaping mechanisms or parameters. (BISmark)

they last, the measurements across users may vary considerably. Specifically, any speedtest measurement that lasts less than 35 seconds will *only* capture the effects of PowerBoost in some cases, and any short-term throughput measurement may be biased by PowerBoost rates.

5.3 Understanding Latency

We show how latency can drastically affect performance, even on ISP service plans with high throughput. We then study how various factors ranging from the user's modem to ISP traffic shaping policies can affect latency.

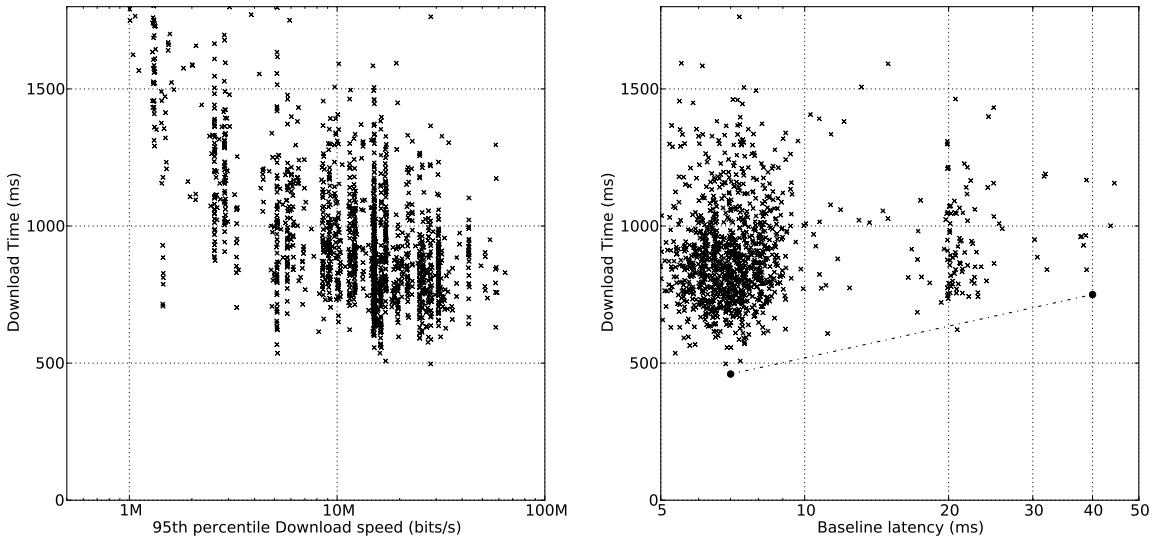
5.3.1 How (and Why) to Measure Latency

Latency affects the performance that users experience. It not only affects the throughput that users achieve, it also affects *perceived* performance: on a connection with high latency, various operations ranging from resolving DNS queries to rendering content may simply take longer.

Although latency appears to be a straightforward characteristic to measure, arriving at the appropriate metric is a subtle challenge because our goal is to isolate the performance of the access link from the performance of the end-to-end path. End-to-end latency between endpoints is a common metric in network measurement, but it reflects the delay that a user experiences along a wide-area path. We use two metrics that are more appropriate for access networks.

The first metric is the *last-mile latency*, which is the latency to the first hop inside the ISP's network. This metric captures the latency of the access link, which could affect gaming or short downloads. We measure last-mile latency in both of our deployments. As we show in this section, the last-mile latency is often a dominant factor in determining the end-user performance. The second metric we define is *latency under load*, which is the latency that a user experiences during an upload or download (*i.e.*, when the link is saturated in either direction). For BISmark, we measure the last-mile latency under load; on the SamKnows platform, we measure latency under load on the end-to-end path.

To investigate the effect of latency on performance, we measured how the time to download popular Web pages varies for users with different throughput and latency. Figure 20 shows the download time for `www.facebook.com` and how it varies by both the user's throughput and baseline last-mile latency. Figure 20a plots the 95^{th} percentile of each user's downstream throughput versus the average time it takes to download all objects from `www.facebook.com`. The average size of the download is 125 KByte. As expected, the download times decrease as throughput increases; interestingly, there is negligible improvement beyond a rate of 6 Mbits/s. Figure 20b plots download time against the baseline latency for all users whose downstream throughput (95^{th} percentile) exceeds 6 Mbits/s. Minimum download times increase by about 50% when baseline latencies increase from



(a) Fetch time stabilizes above 6 Mbits/s.

(b) Latency affects fetch times.

Figure 20: Effect of downstream throughput and baseline latency on fetch time from `facebook.com`. (SamKnows)

10 ms to 40 ms. The fact that this effect is so pronounced, even for small downloads, underscores importance of baseline latency.

We investigate the effects of cable and DSL access-link technologies on last-mile latency, packet loss, and jitter. We also explore how different DSL modem configurations, such as whether the modem has interleaving enabled, affects last-mile latency and loss. Finally, we study the effect of modem hardware on performance. Specifically, we investigate how oversized modem buffers that has recently received much attention from both operators and users [69]—affects interactivity and throughput.

5.3.2 Last-Mile Latency

We obtain the last-mile latency by running `traceroute` to a wide-area destination and extracting the first IP address along the path that is not a NAT address. Note that we are measuring the latency to the first network-layer hop, which may not in fact be the DSLAM or the CMTS, since some ISPs have layer-two DSLAMs that are not visible in traceroute. This should not be problematic, since the latency between hops inside an ISP is typically much smaller than the last-mile latency.

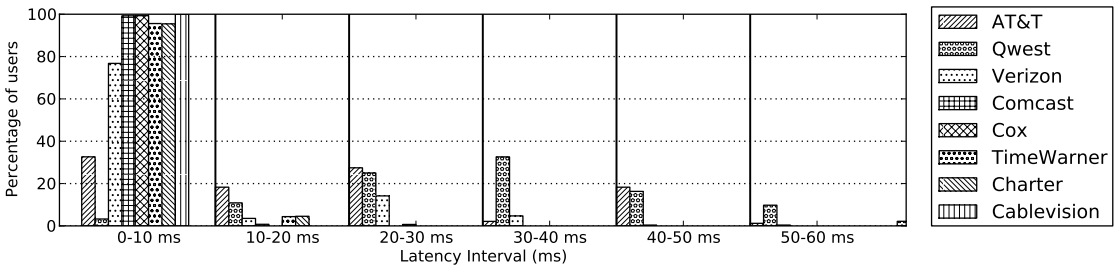


Figure 21: The baseline last mile latency for each user is computed as the 10^{th} percentile of the last mile latency. Most users see latencies less than 10 ms, but there are a significant number of users with the last mile latency greater than 10 ms. (SamKnows)

Table 5: Last-mile latency and variation is significant; Variation in loss is high, suggesting bursty losses. (SamKnows)

ISP	**Last mile latency**		**Loss**	
	Average	Std. dev	Avg(%)	Std. dev
AT&T	25.23	33.47	0.48%	3.59
Comcast	10.36	14.49	0.27%	2.79
TimeWarner	11.87	25.18	0.33%	3.09
Verizon	12.41	20.60	0.51%	4.07
Charter	11.87	11.80	0.43%	3.29
Cox	13.88	28.02	1.11%	8.35
Qwest	39.42	32.27	0.33%	3.38
Cablevision	10.21	7.52	0.33%	3.14

How does access technology affect last-mile latency? Table 5 shows the average last-mile latency experienced by users in the ISPs included in our study. Last-mile latency is generally quite high, varying from about 10 ms to nearly 40 ms (ranging from $40-80\%$ of the end-to-end path latency). Variance is also high. One might expect that variance would be lower for DSL, since it is not a shared medium like cable. Surprisingly, the opposite is true: AT&T and Verizon have high variance compared to the mean. Qwest also has high variance, though it is a smaller fraction of the mean. To understand this variance, we divide different users in each ISP according to their baseline latency, as shown in Figure 21 Most users of cable ISPs are in the 0–10 ms interval. On the other hand, a significant proportion of DSL users have baseline last-mile latencies more than 20 ms, with some users seeing last-mile latencies as high as 50 to 60 ms. Based on discussions with network operators, we believe DSL companies may be enabling an interleaved local loop for these users.

Table 5 shows loss rates for users across ISPs. The average loss is small, but variance

Table 6: Downstream jitter is quite low, however upstream jitter is significant. (SamKnows)

ISP	Downstream		Upstream	
	Average	Std. dev	Average	Std. dev
AT&T	1.85	7.63	3.02	12.92
Comcast	1.15	6.37	3.24	6.60
TimeWarner	1.68	3.35	3.67	12.52
Verizon	1.71	5.01	1.97	4.82
Charter	1.17	1.66	2.66	7.48
Cox	1.18	1.89	4.27	7.10
Qwest	3.04	12.59	2.16	10.95
Cablevision	1.69	3.52	2.25	1.18

is high for all ISPs, suggesting bursty loss. Jitter has similar characteristics, as shown in Table 6; while the average jitter is low, the variation is high, especially on the upstream, also suggesting burstiness.

How does interleaving affect last-mile latency? ISPs enable interleaving for three main reasons: (1) the user is far from the DSLAM; (2) the user has a poor quality link to the DSLAM; or (3) the user subscribes to "triple play" services. An interleaved last-mile data path increases robustness to line noise at the cost of higher latency. The cost varies between two to four times the baseline latency.

Takeaway: Cable providers in general have lower last-mile latency and jitter. Baseline latencies for DSL users may vary significantly based on physical factors such as distance to the DSLAM or line quality.

5.3.3 Latency Under Load

We turn our attention to a problem that has gathered much interest recently because of its performance implications: modem buffering under load conditions [69]. We confirm that excessive buffering is a widespread problem afflicting most ISPs (and the equipment they provide). We profile different modems to study how the problem affects each of them. We also see the possible effect of ISP policies such as active queue and buffer management on latency and loss. Finally we explore exploiting shaping mechanisms such as PowerBoost might help mitigate the problem.

Problem: Oversized buffers. Buffers on DSL and cable modems are too large. Buffers do perform an important role: they absorb bursty traffic and enable smooth outflow at the configured rate [99]. Buffering only affects latency during periods when the access link is loaded, but during such periods, packets can see substantial delays as they queue up in the buffer. The capacity of the uplink also affects the latency introduced by buffering. Given a fixed buffer size, queuing delay will be lower for access links with higher capacities because the draining rate for such buffers is higher. We study the effect of buffering on access links by measuring latency when the access link is saturated, under the assumption that the last-mile is the bottleneck. We also present a simple model for modem buffering and use emulation to verify its accuracy.

How widespread are oversized buffers? Figure 22 shows the average ratios of latency under load to baseline latency for each user across different ISPs for the SamKnows data. The histogram shows the latencies when the uplink and the downlink are saturated separately. This figure confirms that oversized buffers affect users across all ISPs, though in differing intensity. The factor of increase when the uplink is saturated is much higher than when the downlink is saturated. One plausible explanation is that the downlink usually has more capacity than the uplink, so buffering on the ISP side is lower. The home network (at least 10 Mbits/s) is also probably better provisioned than the downlink, so there is minimal buffering in the modem for downstream traffic. The high variability in the latency under load can be partly explained by the variety in service plans; for instance, AT&T offers plans ranging from 768 Kbits/s to 6 Mbits/s for DSL and up to 18 Mbits/s for UVerse and from 128 Kbits/s to more than 1 Mbit/s for upstream. In contrast, Comcast offers fewer service plans, which makes it easier to design a device that works well for all service plans.

How does modem buffering affect latency under load? To study the effects of modem buffers on latency under load, we conduct tests on AT&T and Comcast modems using BISmark. We ran tests on the best AT&T DSL (6 Mbits/s down; 512 Kbits/s up) and Comcast (12.5 Mbits/s down; 2 Mbits/s up) plans. We perform the following experiment: we start ICMP `ping` (at the rate of 10 pkts/s for Comcast and 2 pkts/s for AT&T as

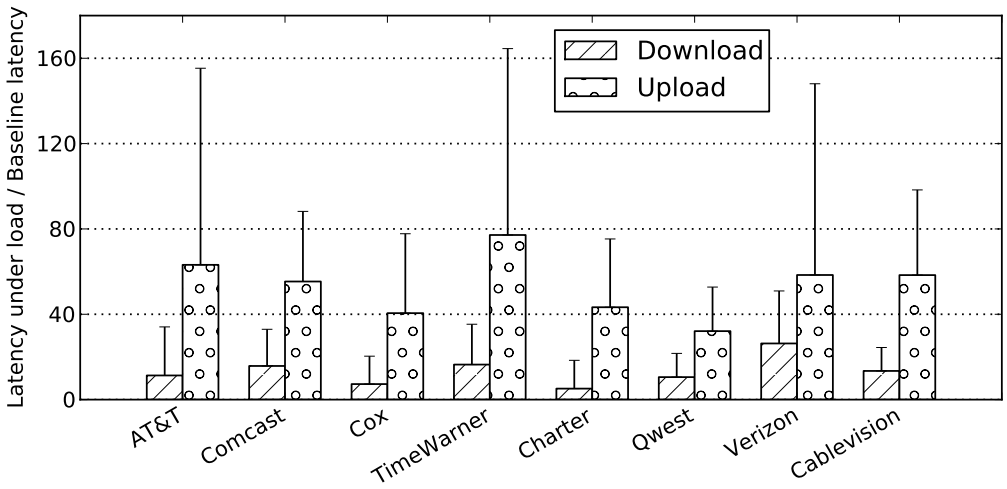
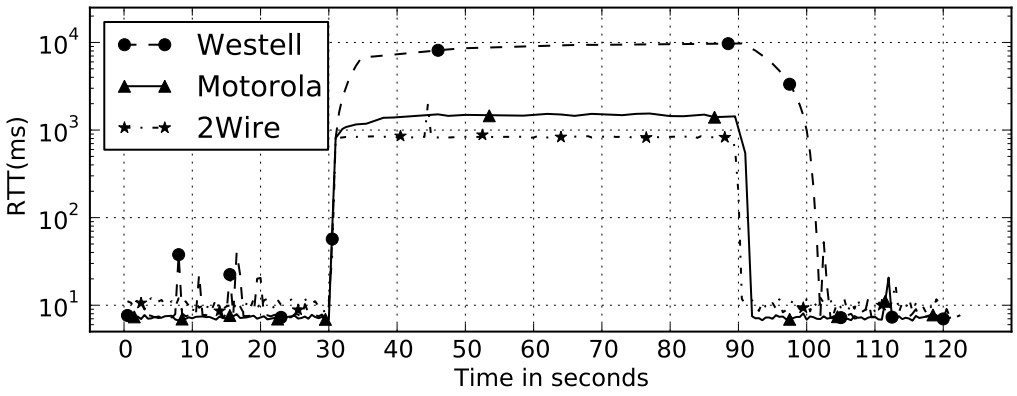


Figure 22: Latency under load: the factor by which baseline latency goes up when the upstream or the downstream is busy. The high ratios translate to significant real latencies, often in the order of seconds. (SamKnows)

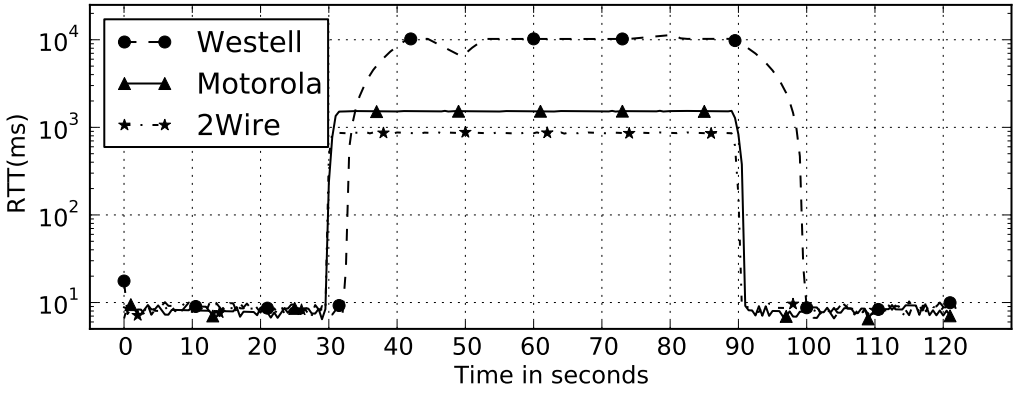
some modems were blocking higher rates) to the last mile hop. After 30 seconds, we flood the uplink (at 1 Mbits/s for AT&T and at 10 Mbits/s for Comcast using `iperf` UDP). After 60 seconds, we stop `iperf`, but let `ping` continue for another 30 seconds. The `ping` measurements 30 seconds on either side of the `iperf` test establishes baseline latency. The Motorola and the 2Wire modems were brand new, while the Westell modem is about 5 years old, and was in place at the home where we conducted the experiment. We also saw the same Westell modem in two other homes in the BISmark deployment.

Figure 23a shows the latency under load for the three modems. In all cases, the latency increases dramatically at the start of the flooding and plateaus when the buffer is saturated. The delay experienced by packets at this stage indicates the size of the buffer, since we know the uplink draining rate. Surprisingly, we see more than an order of magnitude of difference between modems. The 2Wire modem has the lowest worst case latency, of 800 ms. Motorola's is about 1600 ms, while the Westell has a worst case latency of more than 10 seconds. Because modems are usually the same across service plans, we expect that this problem may be even worse for users with slower plans.

To model the effects of modem buffering, we emulated this setup in Emulab [62] with a 2 end-host, 1-router graph. We configured a token bucket filter using `tc`. We compute the



(a) *Empirical measurements* of modem buffering. Different modems have different buffer sizes, leading to wide disparities in observed latencies when the upstream link is busy. (BISmark)



(b) *Emulated modems* with token bucket filters. We see similar latency progression. Emulated buffer sizes have minimal effect on throughput.

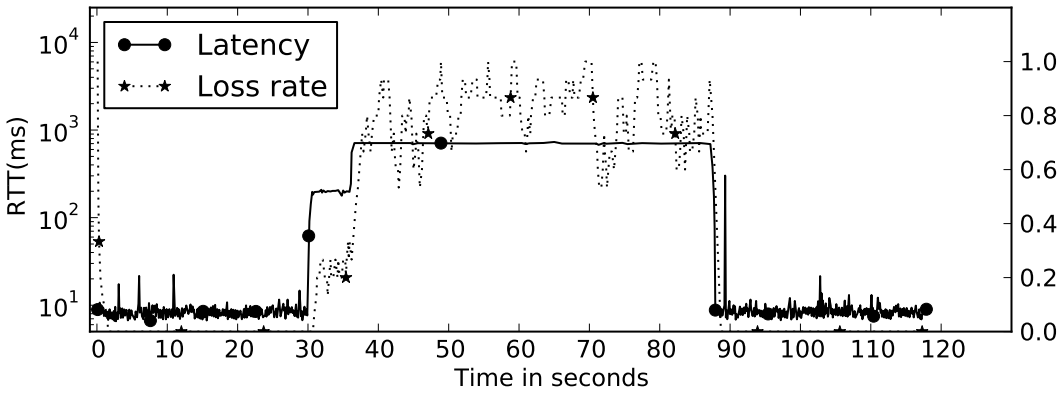
Figure 23: Buffering in AT&T modems. There is little benefit to the buffering seen in most modems.

buffer size to be: 512 Kbits/s×max(latency of modem), which yields a size of 640 Kbytes for Westell, 100 Kbytes for Motorola, and 55 Kbytes for 2Wire. This simple setup almost perfectly captures the latency profile that the actual modems exhibit. Figure 23b shows the emulated latencies. Interestingly, we observed little difference in throughput for the three buffer sizes. We also emulated other buffer sizes. For a 512 Kbits/s uplink, we observed that the modem buffers exceeding 20 KBytes do little for throughput, but cause a linear increase in latency under load. Thus, the buffer sizes in all three modems are too large for the uplink.

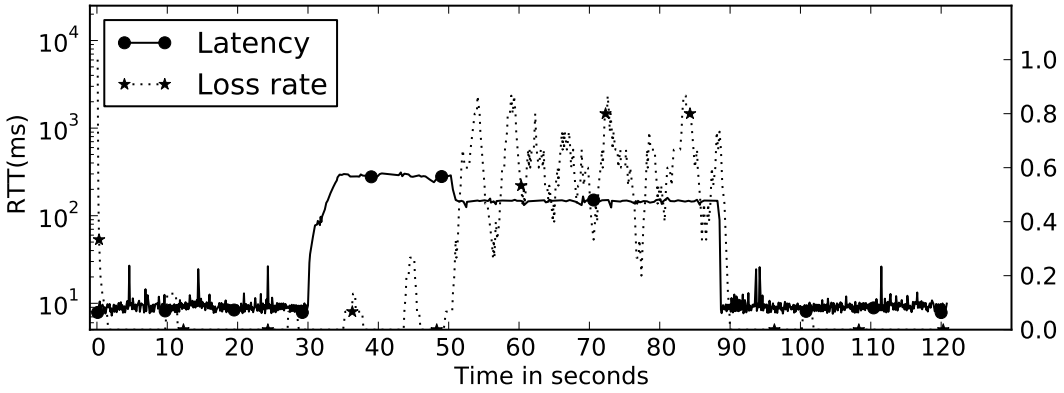
How does PowerBoost traffic shaping affect latency under load? To understand latency under load for cable users, we study the Comcast users from BISmark. All of the modems we study have buffers that induce less than one second of delay, but these users see surprising latency under load profiles due to traffic shaping. Figures 24a and 24b show the latency under load for two Comcast users. The other two Comcast users (with the Scientific Atlanta and the Motorola modems) had latency profiles similar to the user with the Thomson modem, so we do not show them. The difference in the two latency profiles is interesting; the D-LINK user sees a jump in latency when the flooding begins and about 8 seconds later, *another increase* in latency. The Thomson user sees an initial increase in latency when flooding starts but then a *decrease* in latency after about 20 seconds. The first effect is consistent with buffering and PowerBoost. Packets see lower latencies during PowerBoost because, for a fixed buffer, the latency is inversely proportional to the draining rate. The increase in latency due to PowerBoost (from 200 ms to 700 ms) is proportional to the decrease in the draining rate (from 7 Mbits/s to 2 Mbits/s, as shown in Figure 19b). The decrease in latency for the Thomson user cannot be explained in the same way. Figure 24 shows the average loss rates alongside the latencies for the two users; interestingly for the user with the Thomson modem, the loss rate is low for about 20 seconds after the link is saturated, but there is a sharp hike in loss corresponding to the drop in latency. This behavior may correspond to dynamic buffer sizing, as discussed in Section 5.2.3.

5.4 Takeaways

We conclude with some high-level lessons and suggestions for future research directions. One significant takeaway for users, policymakers, ISPs, and researchers is that *continual measurements, directly from home network gateways are crucial for understanding the details of home access network performance.* Existing "speed test" downloads and end-to-end latency measurements do not often reflect access network performance over an extended period of time, and they neglect various confounding factors on the host and within the home. Our ability to execute measurements directly, both from a small set of gateways where we can control the network conditions and measurements (BISmark) and a larger,



(a) Comcast user with D-LINK modem.



(b) Comcast user with RCA Thomson modem.

Figure 24: Possible effect of active buffer management: Loss rates increase when the latency drops. (BISmark)

more representative set of gateways across the United States (SamKnows), yields several lessons:

Lesson 1 (One Measurement Does Not Fit All) *Different ISPs use different policies and traffic shaping behaviors that make it difficult to compare measurements across ISPs.*

There is no single number that characterizes performance, or even throughput. Certain ISP practices such as PowerBoost can distort benchmarking measurements; ISPs might even design their networks so that widely used performance tests yield good performance. Developing a benchmarking suite for ISP performance that users can understand (*e.g.*, in terms of the applications they use) is critical; the measurements we develop in this paper may be a good starting point for that. Along these lines, more work is needed to understand

the performance of specific applications, such as how Web or video streaming performance is affected by the last mile. NetFlix has a study on ISP streaming performance [117]. This thesis studies Web performance bottlenecks in Chapter 7.

Lesson 2 (One ISP Does Not Fit All) *There is no "best" ISP for all users. Different users may prefer different ISPs depending on their usage profiles and how those ISPs perform along performance dimensions that matter to them.*

Different ISPs may be "better" along different performance dimensions, and the service plan that a user buys is only part of the picture. For example, we saw that, above a certain throughput, latency is the dominant factor in determining Web page loading time. Similarly, a gamer might be interested in low latency or jitter, while an avid file swapper may be more interested in high throughput. An imminent technical and usability challenge is to summarize access network performance data so that users can make informed choices about the service plans that are most appropriate for them (akin to a "performance nutrition label" [16]). Our work proposes some first steps in this direction [156].

Lesson 3 (The Home Network Matters) *A user's home network infrastructure can significantly affect performance.*

Modems can introduce latency variations that are orders of magnitude more than the variations introduced by the ISP. Other effects inside the home such as the wireless network, may also ultimately affect the user's experience. Chapter 6 studies the extent to which the wireless network could be a bottleneck in homes, and also takes a first look at the state of wireless networks in homes.

CHAPTER 6

WHERE'S THE FAULT? CHARACTERIZING HOME NETWORK PERFORMANCE PROBLEMS

The home network, dominated by the home wireless network, is a critical part of the end-to-end path. This part of the network functions almost independently of the access network. We saw in Chapter 5 how external factors such as the ISP and congested access links can affect performance. Other factors such as routing problems and poor interdomain connectivity, are also important contributors to performance degradations. *Inside the home*, though, the performance of the home network depends on factors such as cross traffic from devices within the network, or interference from nearby hosts, which can result in poor wireless connections. Even mundane things such as poor placement of an access point can result in a good access link connection performing poorly as far as the user is concerned. Unfortunately, performance measurements of the access link tell us nothing about the performance users actually get; a poor wireless network can dominate user experience. To compound the matter further, neither users nor ISPs currently have a reliable way to determine whether the problem lies within the home network or with the access ISP. This ambiguity is both frustrating and costly: our discussions with several large access ISPs reveal that the cost of service calls range from $9–25 per call, and as many as 75% of service calls from customers are usually caused by problems that have nothing to do with the ISP.

In this chapter, we develop an algorithm and tool that determines whether network performance problems lie inside or outside the home network. Our tool, *WTF (Where's The Fault?)*, detects performance bottlenecks in the last mile. WTF localizes the source of throughput bottlenecks in the end-to-end path to the wireless network or the access link (or beyond). We develop and deploy WTF on home access points, where it can directly observe the both the access link and the home wireless network. We characterize performance bottlenecks in the home, and the wireless network using measurements from WTF. This

chapter gives us an insight into the state of home networks.

We develop and deploy WTF as a tool that runs on the BISmark gateways. This gives us several benefits: a) it allows us to continuously measure the performance characteristics of real home networks for real home network traffic, b) it uses the vantage point offered by the gateway sitting between the wireless network and the access link - two obvious sources of problems, and c) it allows us to piggyback on the global deployment of BISmark nodes. This choice, while having the advantages listed, also made designing WTF interesting and challenging. Although we are now able to collect measurements on a low-cost device that users are familiar with it introduces a unique set of challenges because the device is so resource constrained. This environment makes it difficult to apply existing bottleneck detection and wireless analysis tools, since they typically require additional affordances (*e.g.*, multiple wireless vantage points, significant trace collection). WTF bases its detectors on network properties that can be easily measured from resource-constrained home gateways, which allows us both to design an accurate tool and to implement a longitudinal measurement study. Although WTF does not determine *why* a particular bottleneck or problem exists (*e.g.*, it cannot determine whether a wireless problem results from poor device placement, non-WiFi interference, or other causes), it takes an important first step in helping users and ISPs determine *where* the problem exists, at least to the granularity of whether the problem is inside or outside the home.

We also deployed WTF in 66 homes in 15 countries and measured the extent of wireless and access network performance problems that users experience in these networks; we report on a period covering one month in 2013. Our study yields some interesting findings: First, access link speeds greater than about 35 Mbits/s are more likely to be bottlenecked by the wireless link; as access link capacity increases, the effects of wireless performance play a greater role in the TCP throughput that users observe. Second, the 5 GHz wireless band consistently outperforms the 2.4 GHz band, likely because it has less contention and interference. Third, TCP round-trip latencies between a home wireless access point and devices in the home can be high; in many cases, the round-trip latency introduced by the wireless network is a significant fraction of the end-to-end round-trip latency. Finally,

performance varies across devices, even within a single home.

This chapter discusses two contributions: (1) the design, development, and validation of WTF, a tool that both accurately detects home access link and wireless network bottlenecks and is lightweight enough to run on a home gateway; (2) a detailed characterization of the nature and extent of performance problems that commonly arise in many home networks. The Federal Communications Commission is planning a wider deployment of WTF, and we plan to release WTF to the community by the summer of 2014. Our results lend insight into home networks that we believe have potentially important ramifications for ISPs, content providers, and users. In particular, our results suggest that it is worth spending effort to improve home wireless network performance, in addition to the extensive attempts to optimize latency in other parts of the network and end hosts.

6.1 Detection Algorithm

We are interested in localizing the source of bottlenecks to the access link, the wireless network, or neither. Due to the constrained nature of access links and wireless networks compared to backbone networks, throughput bottlenecks are more likely to occur in one of these two links. However, it is also possible that there is no throughput bottleneck at all; high latency or loss in the end-to-end path, or simply lack of application demand could cause low throughput.

6.1.1 Design

There are a variety of techniques one could potentially use to determine the source of bottlenecks. Active measurements are one. Our gateways are capable of running active measurements of the access link. Such measurements could reveal a lot about the access and the wireless links. However, they do not tell us anything about the performance clients in the home network get. Since we do not have control over clients, we can only run basic active probes like pings to them. The varying nature of wireless networks also make the quality of active measurements suspect; measurements taken while the device is idle may not correspond to the actual performance the user gets when using the device. Conversely, active measurements conducted while users are using the network could adversely affect the

performance they get. We therefore focus on passive traffic so that we can detect bottlenecks independent of the actual throughput of the access link or the wireless link, or the quality of the wireless link.

We exploit a fundamental property of bottleneck links to guide the decision process: *packets buffer at the head of the bottleneck queue.* This property manifests itself in two ways from the point of view of the access point depending on the location of the bottleneck.

- **The bottleneck link is upstream of the gateway (*i.e.*, the access link is the bottleneck)**: In this case, the impact of packet buffering is seen as smoothed packet arrivals — TCP's natural variation caused by congestion control is not seen downstream of the bottleneck link. Packet arrivals are not smoothed out if the access link is not bottlenecked.

- **The bottleneck link is downstream of the gateway (*i.e.*, the wireless network is the bottleneck link)**: In this case, buffer buildup occurs on the gateway, which is at the head of the wireless link. We therefore see increased RTTs between the gateway and the client. This metric is useful because the home network path is short; it is very likely there is only a single hop between the gateway and the client. This means that we can assume that the baseline latency between the client and the access point is low (of the order of 1 ms). Local wireless effects such as loss or contention can cause a slight increase in this latency; however, buffering at the head of the link causes a significant increase in this RTT, as we show later in this section.

- **Neither the access link nor the wireless link is the bottleneck**: This could happen due to many reasons: high latency or loss in the path could prevent either link from being saturated, the end hosts could be the bottleneck, or the application demand is not sufficient to saturate the bottleneck link. This case is essentially a negation of the previous two cases. If we see that the packets are not smoothed out, and that there is no buffering in the wireless link, it then means that the bottleneck link is not saturated.

We can use the above intuition to detect three distinct scenarios: (1) the access link is the

bottleneck, (2) the home wireless link is the bottleneck, and (3) neither the access link nor the wireless link is the bottleneck. We now expand on this intuition and describe how we can build a threshold-based system, and in Section 6.1.3 we describe how we pick thresholds that can detect these cases with high accuracy.

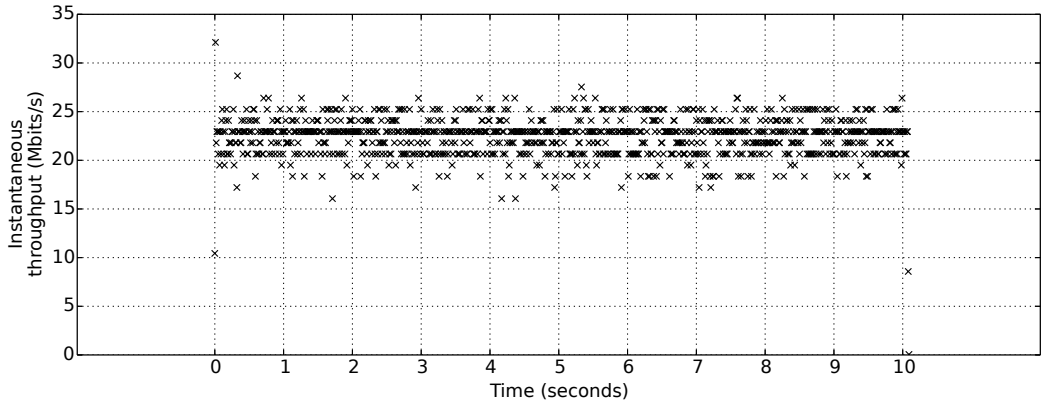
Access link bottleneck We now describe how we use the coefficient of variation of packet interarrival times to detect access link bottlenecks.

We use the intuition that bottleneck links smooth packet arrival rates. Because a bottleneck link services packets at a rate slower than they arrive, queues build up at the link, and the link paces packets at an even rate. Packets upstream of the bottleneck will arrive according to the natural variation induced by TCP congestion control, but packets are more evenly spaced downstream of the bottleneck link. We assume that the most likely bottleneck upstream of the home network is the access link, so *all* flows are buffered, which allows us to use the overall packet distribution for detection.

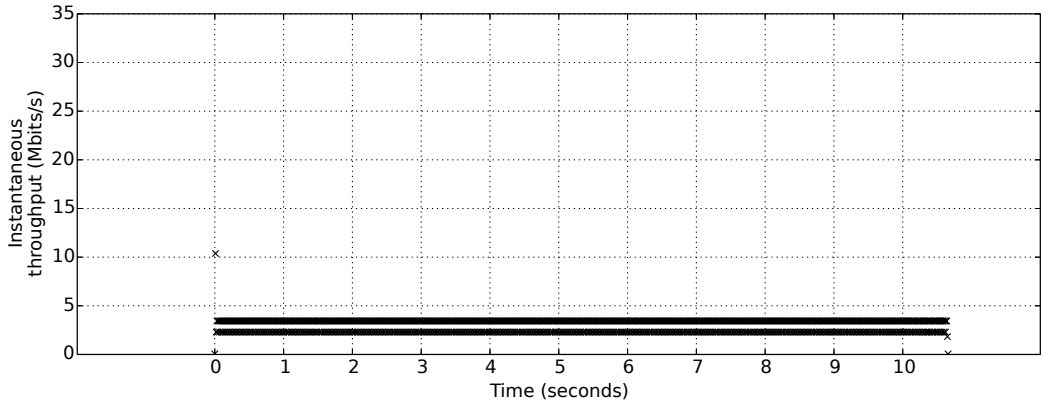
We expect to see high variance in packet interarrival times before the bottleneck link due to congestion control, but significantly lower variance after the bottleneck link itself because the buffer smoothes packet arrivals. Figure 25 shows this effect: It shows the instantaneous TCP throughput at a granularity of 10 ms, as measured from the gateway. In Figure 25a, the access link throughput is 100 Mbits/s; the wireless link is the bottleneck because the maximum TCP throughput it can support is about 21 Mbits/s. In Figure 25b, we shape the access link to 3 Mbits/s, significantly lower than the wireless capacity. In this case, throughput is less variable. Indeed, the coefficient of variation for packet interarrival times, c_v, when the access link is the bottleneck for this example is 0.05; in contrast, when it is *not* the bottleneck, c_v is 0.88.

Wireless bottleneck We describe how we can use the RTT between the gateway and end hosts to determine whether the wireless link is the bottleneck or not.

We use the intuition that TCP round-trip time from gateway to client is high if the wireless link is the bottleneck. The gateway is at the head of the wireless link. Queues build up at the head of a bottleneck link; we use this intuition to detect whether the wireless link is



(a) **Access link is not the bottleneck.** Instantaneous throughput at the WAN interface varies at short time scales due to high variance in packet inter-arrival times.



(b) **Access link is the bottleneck.** Instantaneous throughput at the WAN interface is steady, due relatively uniform packet interarrival times caused by upstream shaping.

Figure 25: Behavior of packet inter-arrival times.

the bottleneck link. Since we cannot view the wireless buffer directly without instrumenting the driver, we look at the impact of buffering on TCP flows. We run `tcptrace` on the traces we collect on the gateway to obtain the RTT of TCP flows between the gateway clients in the local network. If the wireless link is not the bottleneck, the RTT is expected to be low, as the packet will dispatched without delay. Even though the wireless link is not work-conserving, the delays caused by access control are low compared to buffering delays.

Figure 26 illustrates this effect with an example. We run two tests in a setting where the wireless link capacity is about 40 Mbits/s (obtained by repeated measurements). In the first case, the access link is throttled to 30 Mbits/s, so it is always the bottleneck. In the second case, the access link is throttled to 70 Mbits/s so that the wireless link becomes the bottleneck. We see that there is a significant disparity in the TCP RTT in these two

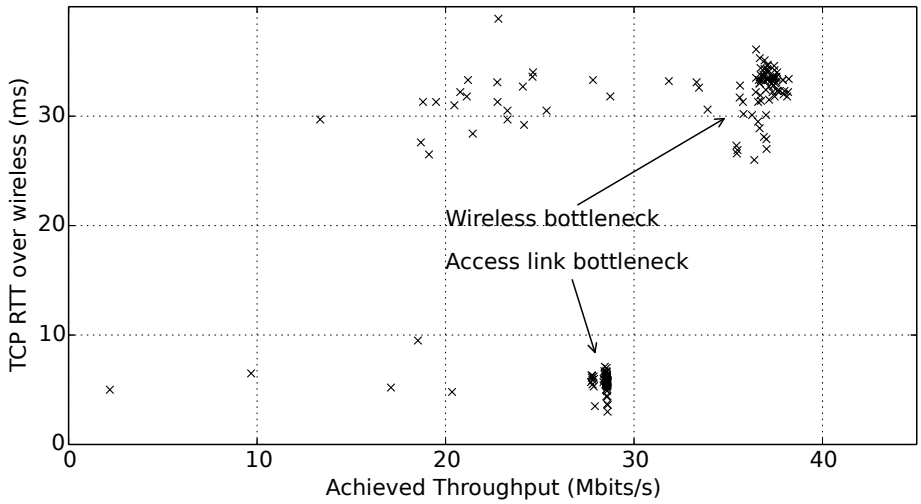


Figure 26: TCP RTT between client and gateway. RTT is significantly higher when the wireless link is the bottleneck; this is caused by buffering. link and the wireless link throughput decrease.

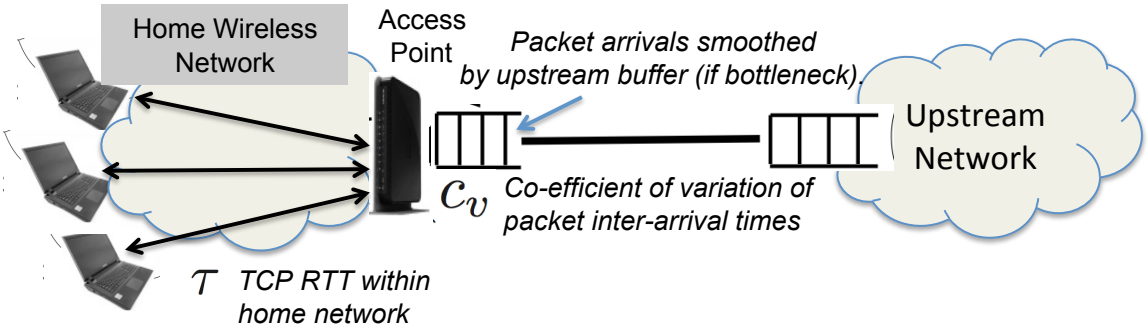


Figure 27: WTF runs on the gateway between the home network and the access link, thus offering a unique vantage point for observing pathologies on either side.

cases; when the wireless link is the bottleneck, the RTT is about 25—30 ms, while when the access link is the bottleneck, the RTT is about 5 ms. This effect does not depend on the achieved throughput; it depends solely on the occurrence of buffering.

6.1.2 Limitations

Using the bottleneck link buffering to localize bottlenecks to the home network or the access network relies on a few assumptions, and also has a few limitations. First, it assumes that the most likely throughput bottlenecks are in the last mile; if a bottleneck link lies beyond the ISP, then it becomes less certain how the detector works — cross traffic traversing

Table 7: The random variables that WTF measures and the roles that they play in helping localize faults to either the home network or the access link. For each random variable that we observe and measure, we design a maximum likelihood estimator to detect whether or not the pathology exists.

Parameter	Description
Access Link Bottleneck (B)	
c_v	Coefficient of variation of packet inter-arrival time
Wireless Bottleneck (W)	
τ	TCP RTT between the AP and the client

intermediate links might cause smoothed packets to become more variable. A limitation of using buffering artifacts is also that the *cause* of the performance degradation is impossible to identify. Later in this chapter, we look at how the wireless potentially affects TCP performance, and the the potential causes of wireless bottlenecks with measurements of wireless properties such as frame bitrates and retransmission, and also TCP properties such wide-area latency to glean more insight into the cause of bottlenecks, but we leave root-cause analysis for future work. The technique works for downstream traffic; it is not straightforward to apply it to upstream traffic. There are two reasons: the first is that when packets arrive from a wireless network, inherent variability (due to the non-work conserving nature of wireless links) may disturb packet smoothing effects; the second is that the increase in latency due to buffering at the access point (when the gateway is the bottleneck) might be harder to extract given the varying nature of latencies to wide-area servers.

6.1.3 Detector Design

We use the intuition described in Section 6.1.1 to identify two features that can be easily measured from the gateway to localize performance bottlenecks. WTF performs two independent detections:

- **Determine whether the access link is a bottleneck.** WTF determines whether the access link is bottlenecked by computing the coefficient of variation of packet interarrival time, c_v, and comparing it against a threshold.

- **Determine whether the wireless link is a bottleneck.** WTF determines whether the wireless link is bottlenecked by estimating the TCP round-trip time (RTT) between the gateway and end hosts in the home network and comparing it against a threshold.

If neither of the above thresholds are breached, then we deem it to be either a bottleneck elsewhere or not enough demand.

Selecting detection thresholds: Maximum likelihood estimation. For each parameter that we evaluate, we design a maximum likelihood detector that treats the observed values of the parameter as a random variable to determine whether it is more likely or not that the pathology has occurred.

For example, to determine whether the access link is the bottleneck, we calculate c_v, the coefficient of variation (the standard deviation divided by the mean) of packet interarrival times on the WAN side of the gateway. Our detector is based on a decision rule that determines whether the "access link bottleneck" event, B, occurs given a particular observed value of c_v during a particular time period. We first compute the conditional probabilities $f(c_v|B)$ and $f(c_v|\overline{B})$ in our controlled setting, where we use our ability to control the throughput of the upstream link to introduce a bottleneck on the access link. We then define our decision rule in terms of the likelihood ratio:

$$\Lambda = \frac{f(c_v = v|B)}{f(c_v = v|\overline{B})}$$

where c_v is the coefficient of variation of packet interarrival time for packets over the observation window. When Λ is greater than some threshold γ, the detector says that the access link is the bottleneck (*i.e.*, it is more likely than not, given the observation of c_v, that the prior is the event B). We can tune the detector by varying the value of the detection threshold, γ; higher values will result in higher detection rates, but also higher false positive rates. Given Λ, we can thus determine the probabilities of a false positive and detection for different values of γ.

These ranges of false positives and detection are commonly known as a receiver operating characteristic (ROC) for a decision rule. We develop a maximum likelihood detector for

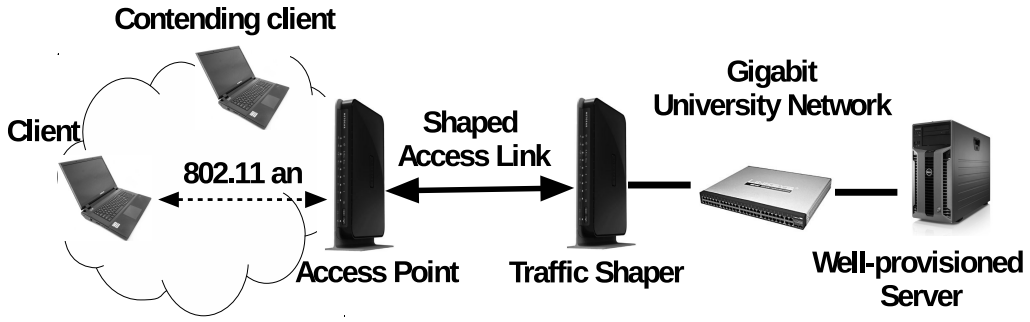


Figure 28: Controlled experiment setup.

each parameter (c_v, and τ) as detectors for the corresponding pathologies B and W, as outlined in Table 7.

Controlled Experiments We built a testbed to run controlled experiments to evaluate each of WTF's detectors; Figure 28 shows this testbed. The testbed has an gateway, its LAN, a network shaper upstream of the gateway, a well provisioned university network, and servers in the university network. The gateway is a Netgear WNDR3800 gateway running OpenWrt. To change the downstream throughput of the emulated access link, we use `tc` and `netem` on a second WNDR3800 gateway running OpenWrt. We run our throughput tests against servers in the same well provisioned university network to avoid potential wide-area effects.

We use this testbed to explore WTF's behavior for a variety of scenarios. For each maximum likelihood estimator we select an appropriate threshold based on its receiver operating characteristic (ROC) that yields a high detection rate and a low false positive rate. We run two sets of experiments using the testbed, for the two pathologies we are trying to detect. For the access link bottleneck scenario, we use the traffic shaper to shape the link to different throughput levels while keeping the wireless link constant. In this case, identifying the ground truth is straightforward, as we know the capacities of both the wireless link and the shaped access link.

For the wireless pathologies, introducing pathological cases and determining ground

truth is more difficult. Rather than directly controlling wireless throughput, we must directly subject the network to certain conditions and then observe the achieved TCP throughput. However, since we know the access link throughput (which we shape), we label the wireless as the bottleneck (event W) if the achieved throughput is less than 90% of the access link capacity. To introduce wireless pathologies, we run two sets of experiments: (1) reduce capacity by degrading channel quality: we do this by positioning the host at different distances from the gateway, and with multiple obstructions. (2) reduce the available capacity of the channel by creating contention with another host that sends constant UDP traffic, with the first host close to the gateway. We test each scenario with different access link throughputs.

For each experiment, we run a TCP throughput test using `iperf`. To minimize interference that we do not introduce ourselves, we use the 5 GHz spectrum, which is less congested than the 2.4 GHz range in our testbed. In our repeated controlled experiments, we found that the wireless channel in our testbed delivers a TCP throughput of about 80 Mbits/s on 802.11n. (We also verify our results with 802.11a which has a maximum TCP throughput of about 21 Mbits/s.) We then extract the random variables that we describe in Table 7 and apply maximum likelihood detection to determine the most effective thresholds for detecting these wireless pathologies. We generate over 1800 samples with 6 different emulated access link throughputs varying from 10 Mbits/s to over 100 Mbits/s and many different wireless conditions with throughput varying from about 20 Mbits/s to the maximum supported (80 Mbits/s).

Choosing a threshold for c_v. Based on observed c_v, we can determine whether it is more likely or not that the access link is the bottleneck. We develop a maximum likelihood detector based on the two different conditional probability distributions, $f(c_v|B)$ and $f(c_v|\overline{B})$ to determine the threshold. We first evaluate the detection accuracy of the algorithm for different values of the detection threshold for c_v. Figure 29 shows the receiver operating characteristic for this detector. When the threshold is low (close to zero), it will always identify the access link as not the bottleneck, and when it is high (close to one), it

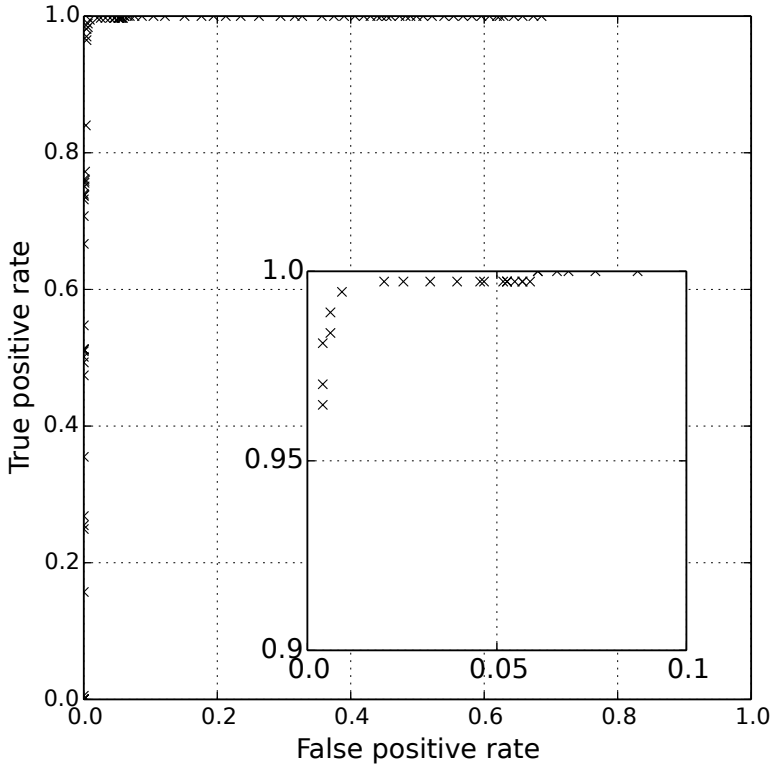


Figure 29: Receiver operating characteristic for access link bottleneck detection using the coefficient of variation of packet interarrival time. The inset zooms into the region of interest — high true positive rate ($> 95\%$) and low false positive rate ($< 5\%$). There is a range of values for which we see good performance, so the threshold is robust.

will always identify the access link as the bottleneck. Our results indicate that detection accuracy remains high for a wide range of threshold settings, particularly between 0.65 and 0.85. Detection accuracy is very high in this range, with a true positive rate more than 95% and a false positive rate less than 5%. The range of good thresholds reinforces our confidence it its robustness as a detection metric. We use a threshold of $c_v < 0.8$ to declare the access link the bottleneck.

Choosing a threshold for τ. We designed a maximum likelihood detector based on the distributions $f(\tau|W)$ and $f(\tau|\overline{W})$; For the wireless bottleneck case, we introduce different kinds of events. In the first case, we configure the setup so that the wireless TCP throughput is the maximum it can support (80 Mbits/s). In the second case, we move the client farther and introduce obstacles so that the wireless throughput is reduced to different levels. In

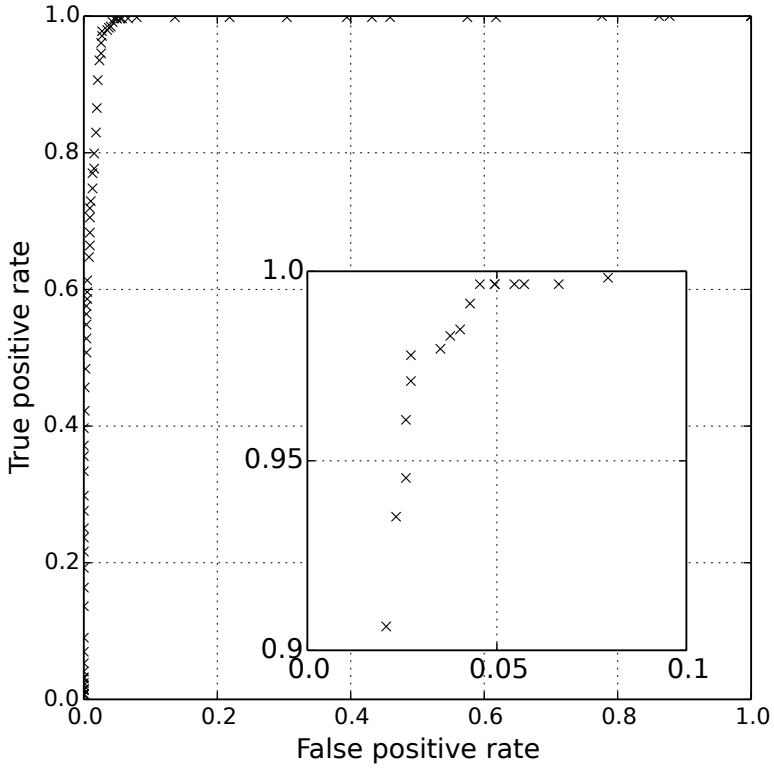


Figure 30: Receiver operating characteristic for wireless bottleneck detection using the TCP RTT between the gateway and the client. Similar to the access link bottleneck ROC, the inset zooms into the region of interest — high true positive rate ($> 95\%$) and low false positive rate ($< 5\%$). There is a range of values for which we see good performance, so the threshold is robust.

the third case, we introduce contention from a competing client sending UDP traffic at 10, 30, and 50 Mbits/s. We test for different access link speeds so that the wireless is either the bottleneck or not. Figure 30 shows the corresponding ROC. We again see high true positive and low false positive rates for a range of threshold values. We choose a threshold of $\tau > 15$ ms yields a detection rate of greater than 95% and a low false positive rate of less than 5

Putting it together We combine the access link and the wireless link bottleneck detectors using a simple algorithm. Figure 31 shows the algorithm. Both the detectors are simple threshold based; therefore there are four scenarios. When either the access link threshold or the wireless threshold is breached, we deem the corresponding link to be the bottleneck.

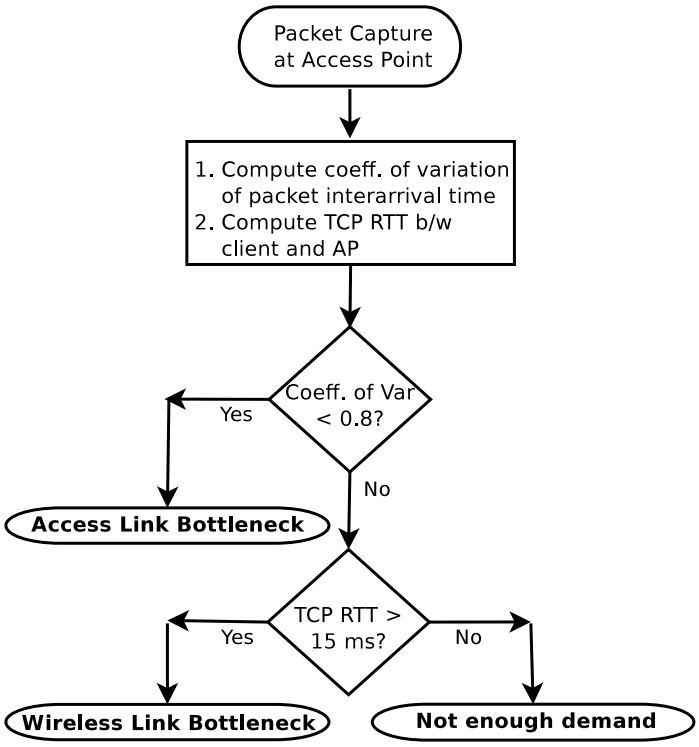


Figure 31: Combining the two bottleneck detectors to create a single combined detection algorithm for access link bottlenecks (event B) and wireless bottlenecks (event W).

When neither thresholds are breached, we deem the bottleneck to be elsewhere, or the application demand to be insufficient. In our experiments, we model this by introducing latency or loss in the path so that TCP throughput is less than the access link and the wireless throughput — neither are the bottlenecks. We test for the case where we detect "other" bottleneck as the output when neither the c_v nor the τ thresholds are breached. We saw that with the thresholds we developed independently above, this case is detected with a true positive rate of 97% and a false positive rate of 2%.

This leaves one case - when both the thresholds are breached. Unless the access link and the wireless link throughputs are closely matched, this is very unlikely to happen, because there can only be one throughput bottleneck in the end-to-end path. Indeed, from our experiments, we see that this case occurs less than 2% of the time. This also increases our confidence in the robustness of the thresholds we choose.

6.2 System Design and Deployment

We describe a prototype of WTF that we have deployed in 66 homes. Running on commodity gateways in so many homes posed several technical challenges: Although commodity gateways offer a low cost and familiar form factor, they have limited computation and storage capacity. Moreover, keeping users engaged requires that WTF be unobtrusive and respect user privacy. We detail the data WTF collects and how the overall system design addresses these challenges.

6.2.1 Measurements

WTF uses passive measurements, which (unlike active measurements) do not risk introducing contention that could affect the very conditions and performance that we seek to characterize. Further, passive measurements more accurately reflect the actual performance that users experience. To facilitate deployment across a large number of homes, WTF collects traces from only a single vantage point; this approach allows us to run WTF within the context of any existing home network, without deploying additional (or customized) hardware.

There are many ways to collect the data used in the detection algorithm that we described in Section 6.1. To facilitate deployment, WTF collects only measurements that were easily accessible from a resource-constrained home gateway. Additionally, we designed WTF's data collection to be as lightweight and concise as possible, to facilitate fast and unobtrusive uploads to a central analysis server. WTF collects the following measurements:

- `pcap` *traces of connections.* We collect packet traces with `tcpdump` from both the WAN and the wireless interfaces (each gateway has two). Packet traces from the WAN interface provide information about TCP connections and IP packets flowing through the gateway. The wireless interfaces (in monitor mode) capture radiotap headers [136], which, for each frame, include: the source and destination stations, the bitrate used, and whether the frame was retransmitted (but not how many times it was retransmitted). The server computes bitrates and retransmission rates independently for each device.

- *ARP information.* This data provides the device MAC ID-to-IP address mapping for end points in the home network.
- *Connection tracking information from Network Address Translator (NAT) module.* To obtain information about the end point of TCP connections inside the home, we collect a snapshot of the `conntrack` file that maps WAN ports to LAN IP addresses and ports.

For a further characterization of the wireless network in homes, WTF collects per-client 802.11 information from radiotap headers. This data gives us the per-frame bitrates and the retranmission bits.

6.2.2 Design and Implementation

We use Netgear's WNDR3700/3800 platform, which has an Atheros chipset with a 450 MHz processor, one 802.11gn radio, and one 802.11an radio. The 3800 has 128 Mbytes of RAM, and the 3700 has 64 Mbytes of RAM. The devices run OpenWrt, with the ath9k wireless driver. The driver uses the Minstrel rate adaptation algorithm, with the default setting to a maximum bitrate of 130 Mbits/s.

Due to the resource limitations on the gateway, we perform data collection and some amount of limited processing locally but push most processing and analysis to a central server. WTF first processes the WAN `pcap` traces to extract timestamps of arriving packets and information about individual flows such as RTT on either side of the gateway, and the number of packets in each connection (using `tcptrace` [1]). Performing the trace at the access point allows us WTF to clearly identify the latencies between the gateway and each respective endpoint. WTF also processes the radiotap traces to obtain the source and destination MAC addresses and the frame control bits for each frame.

To respect user privacy, WTF anonymizes all IP addresses and MAC addresses completely using SHA-256 and a per-gateway secret salt. The gateway discards all private information and uploads the pre-processed to the server, at which point it deletes the local copy of the data. The data is stored in a database where the diagnosis and longitudinal analysis portions of WTF reside. *All aspects of this study have been reviewed and approved*

Table 8: We deployed WTF in 66 households in 15 countries across four continents.

Total # of homes	66
Duration	Mar 6 – Apr 6, 2013
Total # of countries	15
2.4 GHz	
Active devices	192
Devices per home	2.9
5 GHz	
Active devices	66
Devices per home	1

by our university institutional review board (IRB).

WTF considers only the instances where traffic exceeds 100 packets per second, to ensure a reliable computation of c_v. Before computing c_v for an interval, WTF also discards outlier samples for cases where the packet inter-arrival time exceeds the average plus two standard deviations.

Continuous data collection and analysis would impose a significant burden on commodity gateways. Apart from CPU intensive tasks such as monitoring traffic on multiple interfaces, the gateway must also collect a significant amount of data. To minimize the CPU load and the amount of data uploaded, the current implementation of WTF collects data once every 5 minutes on average for 15 seconds per iteration. Sampling provides insight into the overall nature of each home network and facilitates rapid development and deployment, but it does not allow us to obtain fine-grained characteristics (*e.g.*, conditions that vary with high frequency).

6.3 Understanding last mile bottlenecks

To understand where performance problems tend to occur in real home networks, we deployed WTF in 66 homes. Table 8 summarizes our deployment and the characteristics of the home networks in this deployment. Our results lead to the following findings, which we highlight in respective subsections: (1) wireless network bottlenecks are common as access link throughput exceeds about 35 Mbits/s; (2) TCP latencies on the wireless network inside a home can be a significant fraction of overall round-trip latency. We now explore each of these results in more detail.

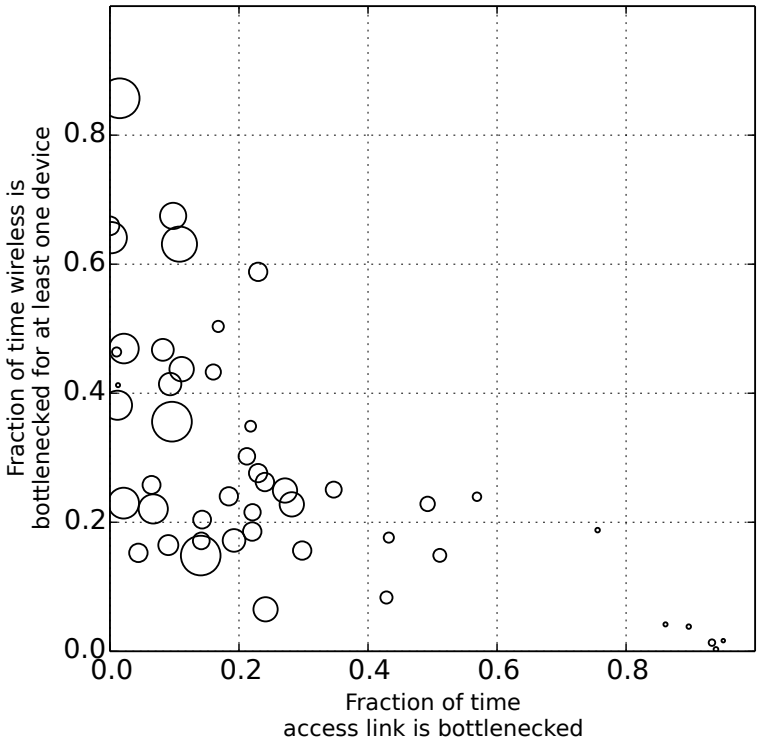


Figure 32: Prevalence of bottlenecks home networks. Each circle represents one home (circle area is proportional to downstream throughput). Poor wireless connectivity is much more common than are access link bottlenecks.

6.3.1 Wireless Bottlenecks Are Common

We study the relative frequency of the two types of pathologies that our detectors detect, based on the threshold settings that we derive from our controlled experiments. Good wireless performance with low access link utilization suggests either a lightly used network (and the possibility of even downgrading the service plan without adverse effects), or significant pathologies in the wide area — high latency or lossy paths. Figure 32 plots the fraction of time the access link is bottlenecked versus the fraction of time that at least one active wireless device is experiencing a bottleneck. Each circle represents a single home network; the area of the circle is proportional to the downstream throughput of the access link for that home. The results show that *a significant number of homes in our deployment have wireless problems, and access links exceeding about 35 Mbits/s are likely never bottlenecked by the access network*. The quality of the wireless links also varies; users with access link

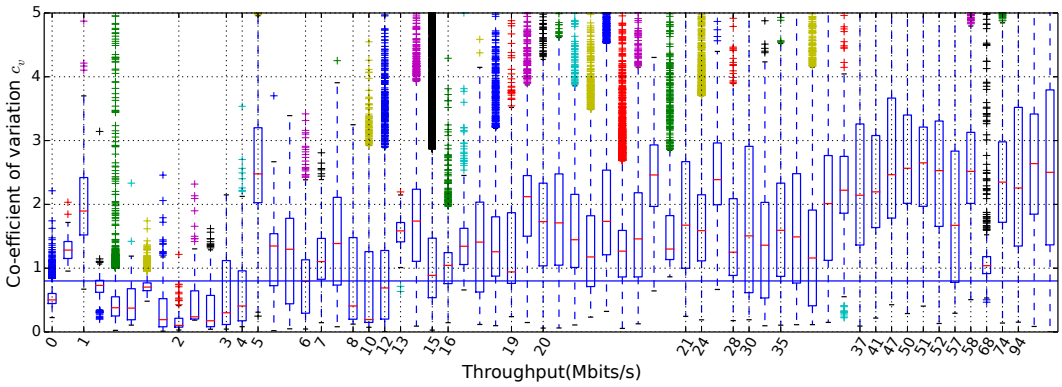


Figure 33: c_v values for all home networks in our study; values below the horizontal line indicate consistent access link bottlenecks. The horizontal line shows the threshold for c_v of 0.8, below which we declare the access link to be the bottleneck. None of the home networks whose access links have downstream throughput greater 35 Mbits/s experience a significant access link bottleneck.

throughput between 15 and 25 Mbits/s had wireless bottlenecks ranging from 15 — 58% of the time. Since by default the devices are configured with 802.11n which can support throughputs much higher than that (even with 802.11g, the maximum TCP throughput is greater than 25 Mbits/s), this suggests significant wireless performance issues in homes.

We now look at the nature of these access link bottlenecks in more detail. Figure 33 shows the distribution of the coefficient of variation for packet inter-arrival time, c_v, for homes in our deployment. The box plot shows the inter-quartile range of c_v when traffic on the access link exceeds 100 packets per second (*i.e.*, when the network is not idle). We observe that *none of the homes with downstream throughput greater than 35 Mbits/s experience a significant access link bottleneck* (which we define as having the 25^{th} percentile value of c_v falling below the bottleneck detection threshold). We also observe two other features: First, c_v generally increases as access link speed increases. This result makes sense: high downstream throughput reduces the likelihood of the access link being bottlenecked with traffic and increases the likelihood of the wireless being the bottleneck. Second, we observe large variations in c_v, even among access links of similar throughputs. This variation results from the diversity of wireless conditions and usage patterns across households. Home networks with higher access link throughput also tend to have higher c_v values, since it is less likely that the access link is a bottleneck in those cases.

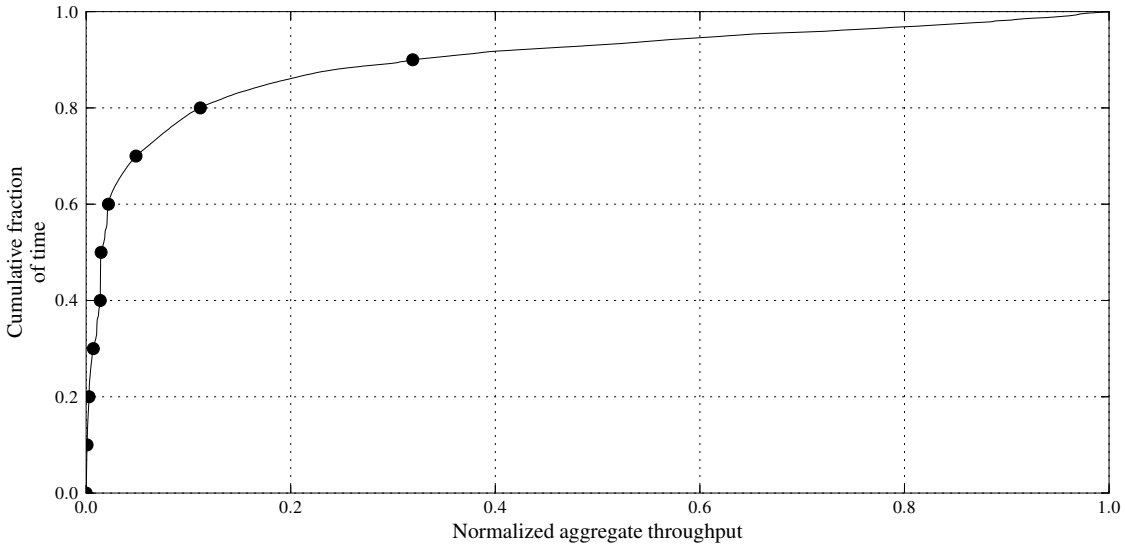


Figure 34: The fraction of time that the collection of active flows receive a particular ratio of flow throughput to access link throughput.

6.3.2 Correlating TCP & Wireless Performance

We explore the achieved throughput of user traffic and the contributions of the home wireless network to this performance. Then, we study RTTs of user traffic and how the poor wireless network performance can result in higher LAN RTTs. This finding is relevant in light of the many recent efforts by service providers to reduce latency to end-to-end services with myriad optimizations and careful placement of content.

Metrics Used We use the passive traffic traces to extract both TCP-level performance metrics and wireless performance metrics.

- *TCP performance metrics.* The access point runs `tcptrace`, which processes the pcap traces to provide TCP statistics. We study the *average download TCP throughput* achieved during the captured lifetime of the flow. We use this metric to compute the *aggregate throughput* at every one-second interval by summing the average throughput of all active flows downloading traffic through a given access point during that interval. For reference, we compare the aggregate throughput with the access link capacity, which we measure using BISmark's active measurements. BISmark performs

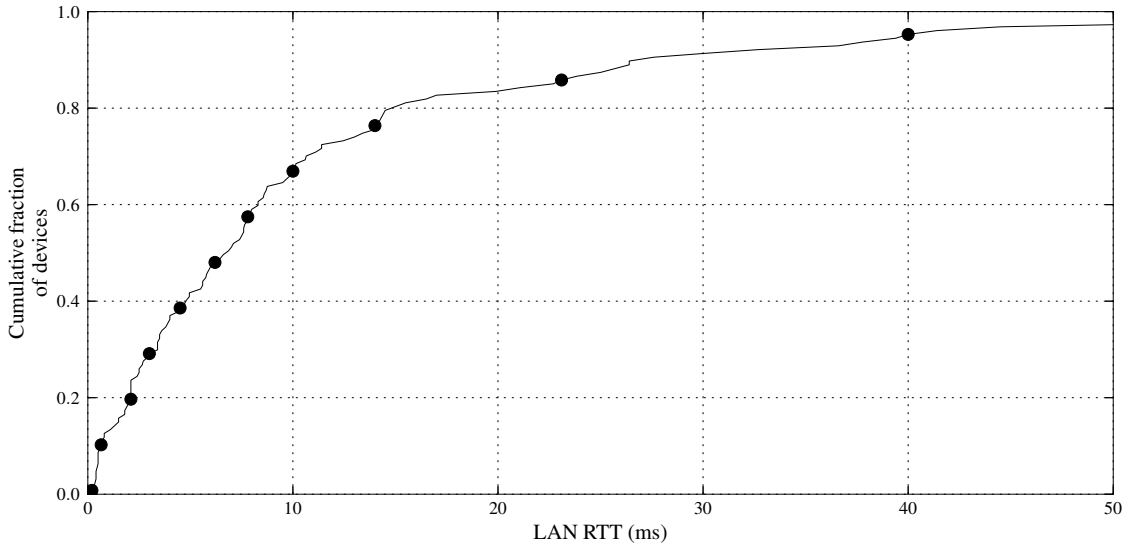
a multi-threaded TCP throughput test approximately every two hours. We define the *access link capacity* as the 95^{th} percentile of the multi-threaded throughput test measurements. We also study the round-trip time (RTT) of TCP connections, which we compute as the difference between the time of the data packet and its corresponding acknowledgment (`tcptrace`'s analysis algorithm already handles many corner cases, such as delayed acknowledgments). Running `tcptrace` at the access point allows us to measure both the RTT between the access point and home devices (the *LAN RTT*) and the RTT between the access point and destinations in the wide-area (the *WAN RTT*).

- *Wireless performance metrics.* We use the bitrate and retransmission rate as our indicators of wireless performance problems, since both metrics can be easily obtained from packet headers. IEEE 802.11 bitrate adaptation techniques adjust the transmission bitrate as wireless channel conditions change. Although SNR also correlates with the performance of user traffic [23,87], we focus on retransmission rate since this is the metric that Minstrel uses in its bitrate adaptation algorithm [112]. Although these techniques usually adapt rates even under benign conditions to determine the channel quality, rate adaptation is typically more frequent when the channel quality is poor, because wireless senders typically reduce the bitrate in response to bit errors [112]. Thus, we also use the *normalized bitrate*, which is the average wireless bitrate computed over one second intervals, normalized by the maximum bitrate supported by that channel, as an indicator of a poor wireless channel. Normalized bitrate tends to be low when the wireless channel quality is poor. When bitrate adaptation does not adjust the bitrate (*e.g.*, due to varying channel conditions or contention), the normalized bitrate might not indicate channel quality, but in these cases retransmission rates are still high. We also compute *retransmission rates* as the number of frames with the retransmit bit set over one second intervals.

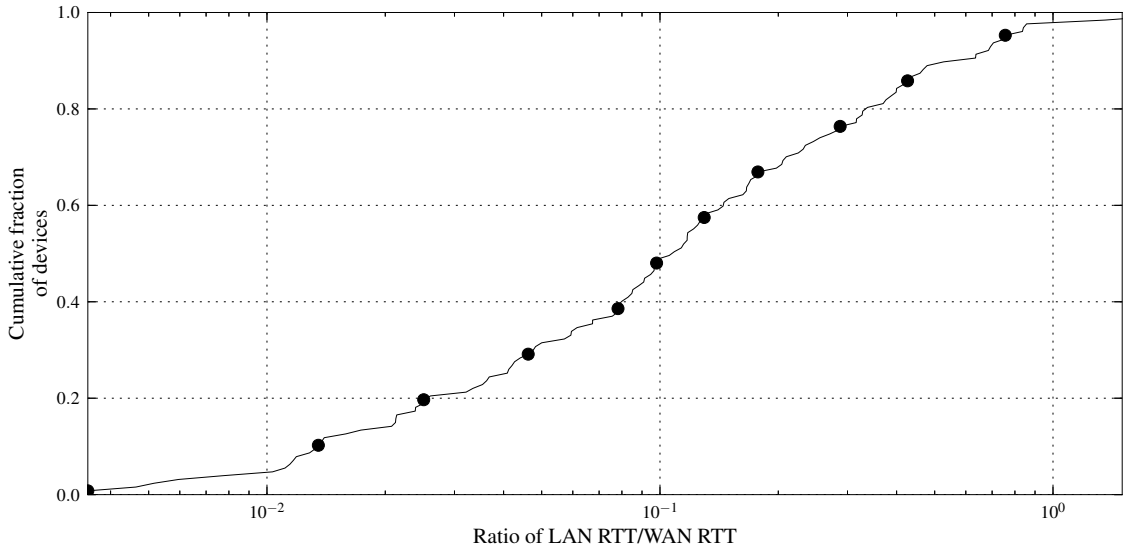
User traffic rarely achieves the full access-link throughput. Figure 34 shows the fraction of time that the sum of TCP throughput for all flows in a home (the "aggregate

throughput") achieves a particular value relative to the access link throughput, as measured with BISmark's active throughput test. The results show that the user traffic rarely saturates the available access link throughput. Of course, the TCP traffic might not saturate the access link throughput for many reasons: for example, user traffic demand may be insufficient (in fact, previous studies have shown this phenomena is often the case [149]), or flows may be short enough that they fail to saturate the access link. Unfortunately, we have only the flow statistics exported by `tcptrace`, so we cannot run a tool like T-RAT [169] to identify with certainty when the application was limiting TCP throughput. Nevertheless, it is remarkable that the access link is so underutilized so often. We suspect that one reason for lower utilization of the access link throughput may be wireless bottlenecks in the home network. The rest of this section explores this possibility.

Achieved throughput often correlates with wireless performance metrics. To explore the relationship between the throughput that active flows in the home network achieve and the access link throughput, we measure how the aggregate throughput correlates with the bitrate and the retransmission rate. We normalize the aggregate throughput by the access link capacity (*normalized throughput*) and correlate this value with each of the wireless performance metrics. For the set of all flows, TCP performance does not correlate with either of the wireless performance metrics: the correlation coefficient between retransmission rate and normalized throughput is -0.01; for bitrate, the correlation coefficient is -0.02. On the other hand, when we explore the correlation for the subset of flows whose normalized throughput is greater than 0.1, correlation is stronger (below 0.1, we reason that throughput may be low simply due to lack of demand). This correlation grows as the access link throughput increases. In Figure 36 we show how the correlation coefficient between aggregate throughput and retransmission rate varies as we only consider users with access link throughput above a certain value; we see as this value increases, the correlation becomes stronger. This makes sense: wireless is more likely to introduce a bottleneck as access link throughput increases. The coefficient for bitrate follows a similar trend, however, it starts weakening as access link throughput exceeds about 60 Mbps. We believe that this is because



(a) Distribution of TCP round-trip time between the access point and client across all devices in our study.



(b) The distribution of the median ratio of the LAN TCP round-trip time to the WAN TCP round-trip time across all flows for that device, across all devices.

Figure 35: Round-trip latency of flows.

of the default setting of the access points, which supports a maximum bitrate of 130 Mbps which translates to a TCP throughput of about 80 Mbps under excellent conditions; actual throughput will likely be less.

The latency inside a home network is often a significant contributor to overall round-trip time. The TCP round-trip time between the wireless access point and a

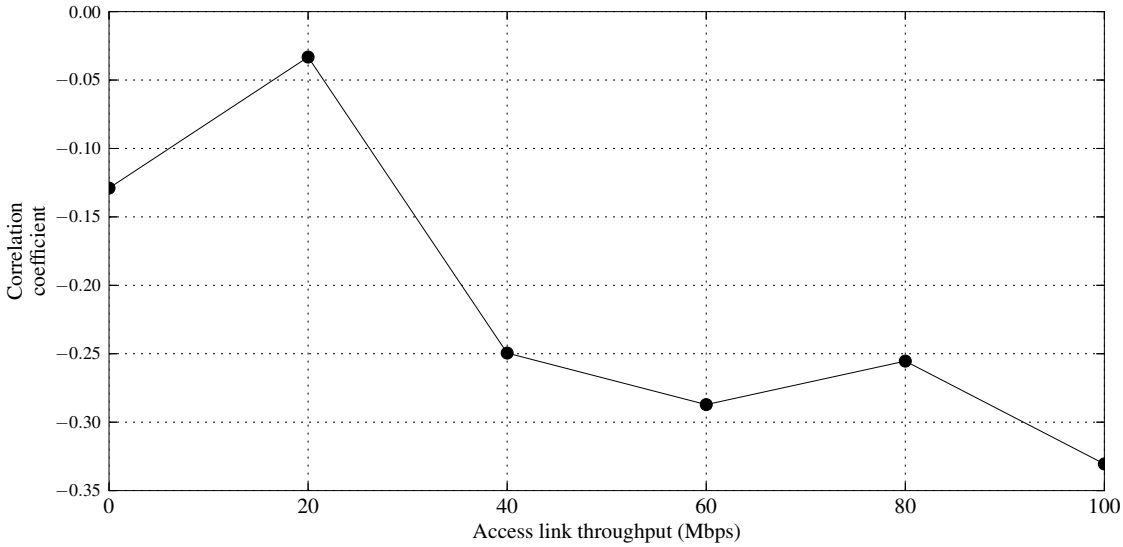


Figure 36: Coefficient of correlation of wireless retransmission rate to normalized throughput.

wireless client should be on the order of one millisecond. As this RTT increases, it not only signifies that the wireless link is bottlenecked due to buffering or medium access delays, but it can have an adverse impact on performance, especially for applications that are latency sensitive. Figure 35a plots the distribution of the median LAN RTT across all devices in our study. The median device on the local wireless network sees a median wireless latency of about 8 ms, but nearly 30% of the devices experience local TCP round-trip latencies greater than 15 ms.

We also analyze the performance of the home network relative to the wide-area network performance; we compare the round-trip times between the devices and the access point to the round-trip times from the access point to the wide-area destination for each flow. We define the *median latency ratio* for a device as the median ratio of the LAN RTT to the WAN RTT across all flows for that device. Figure 35b shows the distribution of the median latency ratio across all devices. The result shows that 30% of devices have a median latency ratio greater than 0.2, meaning that for those devices, at least half of the flows have end-to-end latencies where the home wireless network contributes more than 20% of the overall end-to-end latency.

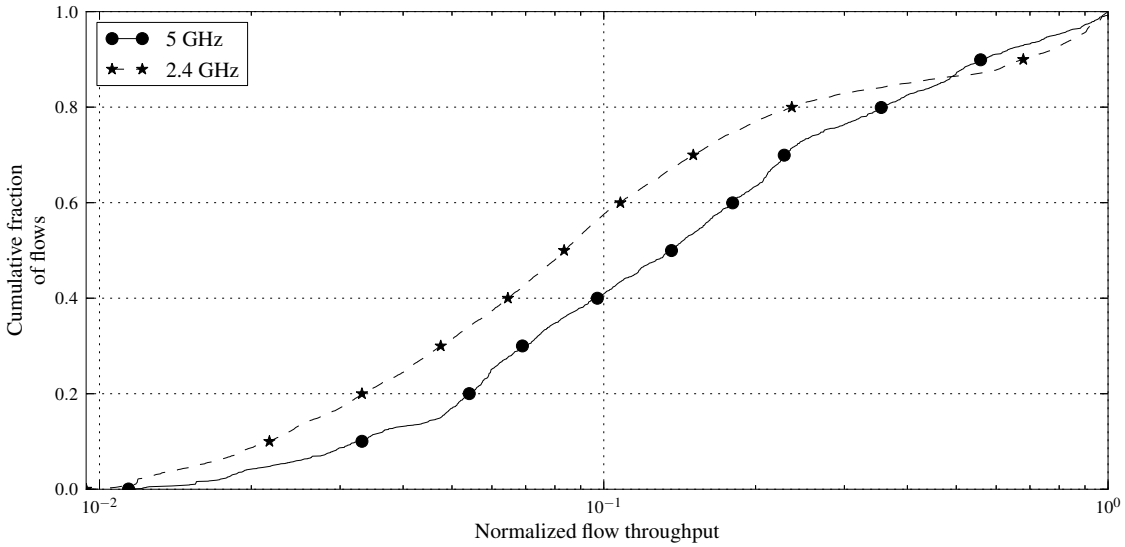
6.3.3 Wireless Performance

We now characterize wireless performance in our deployment. Our preliminary findings include: (1) the 5 GHz wireless band consistently achieves better performance than the 2.4 GHz band; (2) the performance of a home wireless network varies across individual wireless devices within the same home; and (3) simultaneous communication occurs infrequently.

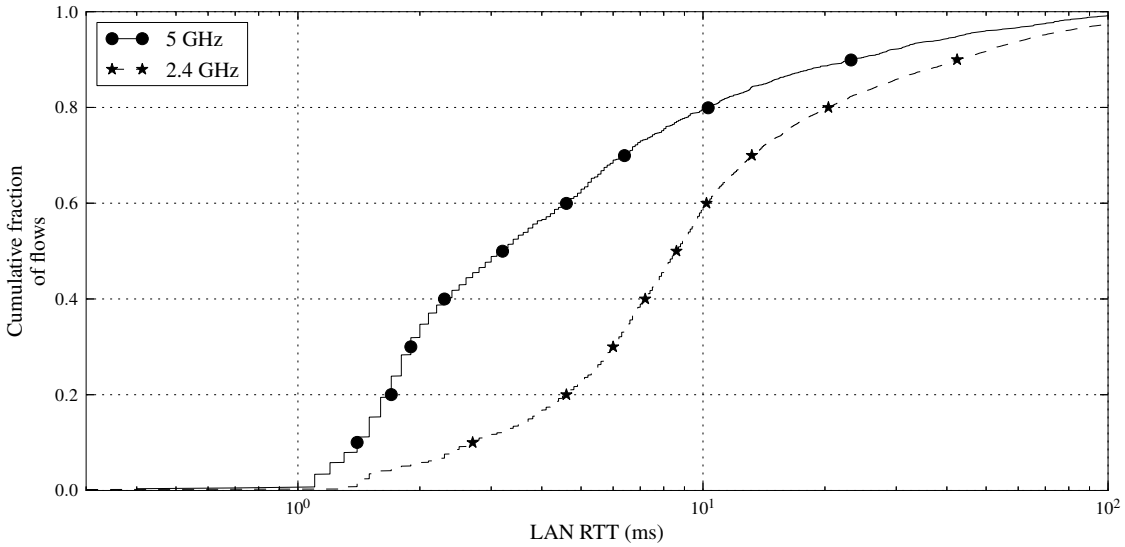
The 5 GHz band performs better than the 2.4 GHz band. We analyze the performance that devices in home wireless networks achieve and how performance varies depending on whether devices are on the 2.4 GHz band or the 5 GHz band. Our hypothesis was that devices on the 5 GHz band would perform better because there are generally fewer devices (and surrounding access points) in the 5 GHz band, and that the 5 GHz band also has less non-WiFi interference (*e.g.*, microwaves, baby monitors).

Figure 37a shows the impact of spectrum on flow throughput for flows that have throughput greater than 1 Mbps. We present the normalized flow throughput to eliminate any bias related to the access link capacity. We see that flows to devices on the 5 GHz spectrum have higher normalized throughput than those on 2.4 GHz. Similarly, we see in Figure 37b that the LAN RTT for flows in 2.4 GHz are much higher than for flows in 5 GHz. The distribution of normalized flow throughput in each spectrum is similar between the 2.4 GHz and 5 GHz when we consider flows whose normalized throughput is greater than 0.1. We are investigating this phenomenon, but these could include cases where we suspect that there is not enough application demand. Even in those cases, however, the LAN RTTs are smaller for devices connected over 5 GHz.

Figure 38 plots the CDF of the median bitrate for all devices in all homes, for both the 2.4 GHz band and the 5 GHz bands. Only 30% of 2.4 GHz devices see median bitrates above 65 Mbps; in contrast, more than 50% of devices in the 5 GHz spectrum see bitrates greater than 100 Mbps. It is worth noting here that the wireless bitrates do not correspond to the actual throughput. Even under perfect conditions, a wireless bitrate of 130 Mbps corresponds to an actual TCP throughput of about 80 Mbps. The bitrate values thus reflect



(a) Flows in the 5 GHz band achieve higher throughput.



(b) Flows in the 2.4 GHz band experience higher LAN RTT.

Figure 37: Characteristics of flows in the 5 GHz vs. the 2.4 GHz spectrum.

a very loose upper bound on the achievable end-to-end throughput.

Figure 39 shows the median bitrate per device for each home network, normalized by the maximum supported bitrate of the corresponding wireless protocol (between 65 Mbps and 300 Mbps for 802.11n, and 54 Mbps for 802.11a/g). Many devices, especially those in the 2.4 GHz range, often operate close to the maximum bitrate supported by the protocol, more so than 5 GHz devices. However we also see that the maximum bitrates of 5 GHz

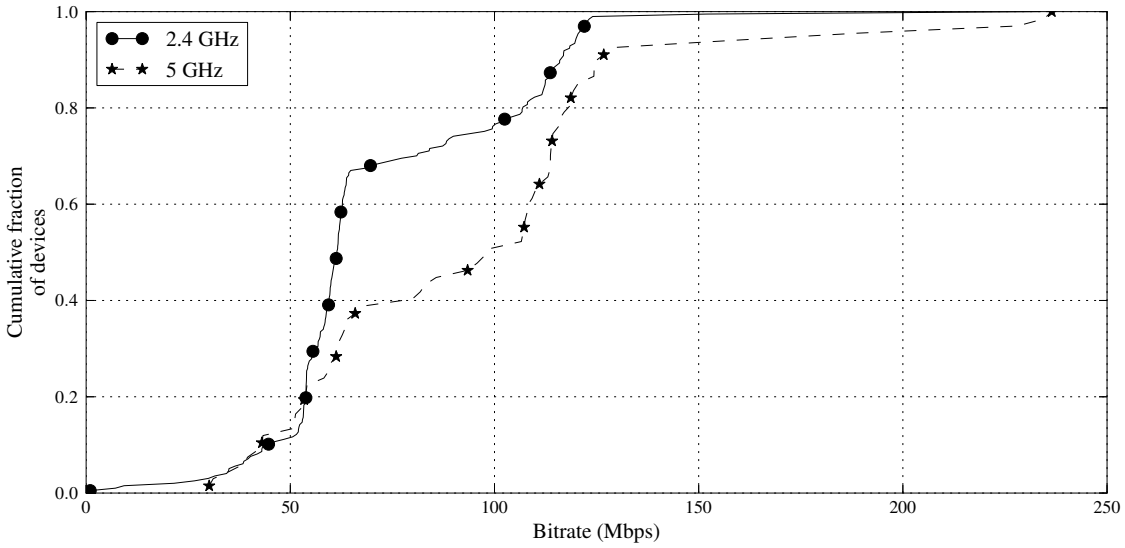


Figure 38: Distribution of wireless bitrates for devices in both the 2.4 GHz and 5 GHz spectrums, for all devices in the deployment.

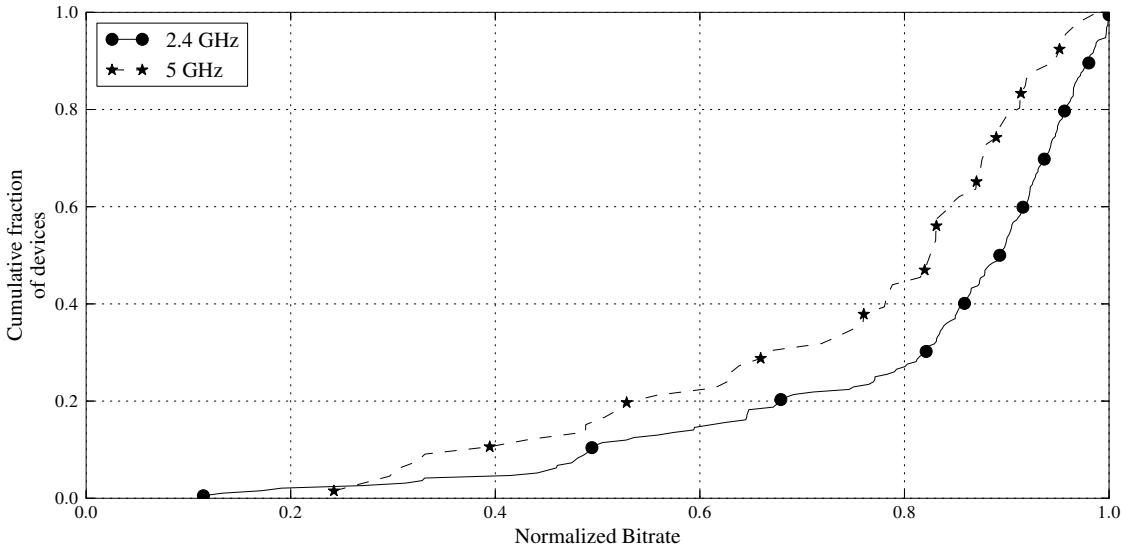


Figure 39: Distribution of median normalized bitrates, for devices in both the 2.4 GHz and 5 GHz spectrums. Devices do not achieve maximum bitrate, especially in the 2.4 GHz range, and about 50% of the devices experience poor wireless channels at least half of the time.

devices are higher. This discrepancy can be explained by the fact that many devices in the 2.4 GHz channel could be small mobile devices with single antennas that restrict their maximum bitrates to 65 Mbps. Also, attenuation is higher on 5 GHz, which could lead to more active bitrate adaptation.

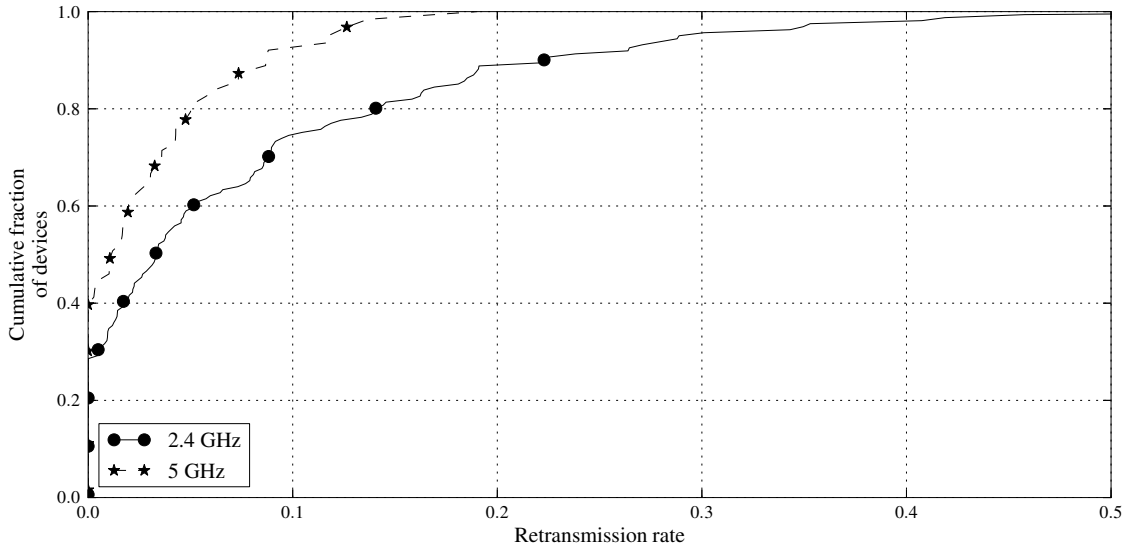


Figure 40: Distribution of median retransmission rates, for devices in both the 2.4 GHz and 5 GHz spectrums. Retransmissions are higher in the 2.4 GHz spectrum, where nearly 30% of devices see a median retransmission rate greater than 10%.

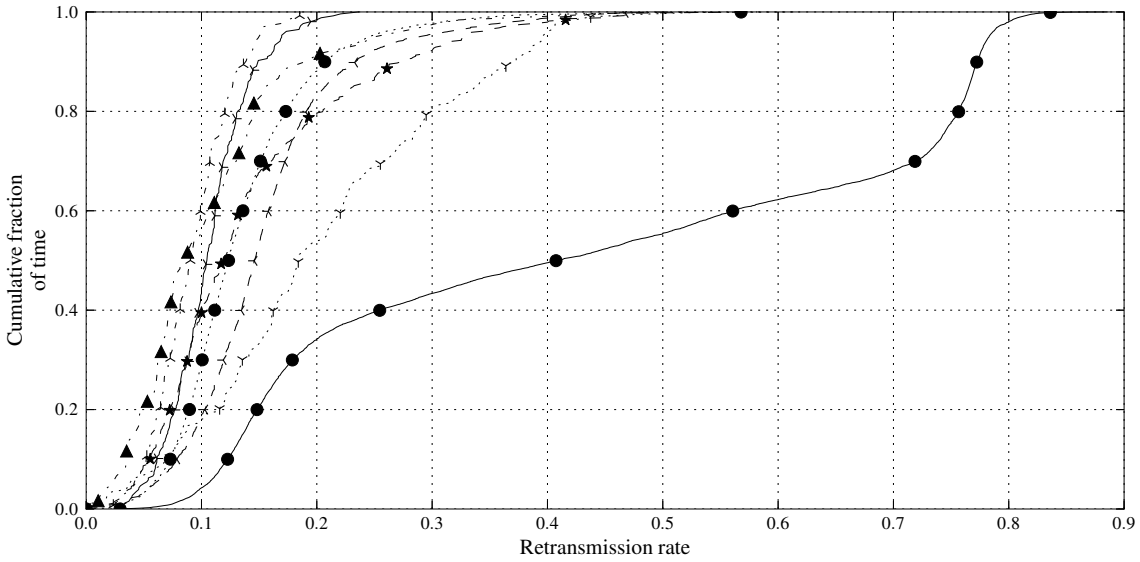


Figure 41: The retransmission rates between the access point and clients in a single home network. In this home retransmission rates are high. Interestingly, one device has a significantly higher retransmission rate.

Figure 40 shows the retransmission rates for all devices across all homes; the result shows similar trends with respect to the 2.4 GHz and 5 GHz ranges: retransmissions are common in the 2.4 GHz band, with about 20% of devices having retransmission rates above 10%.

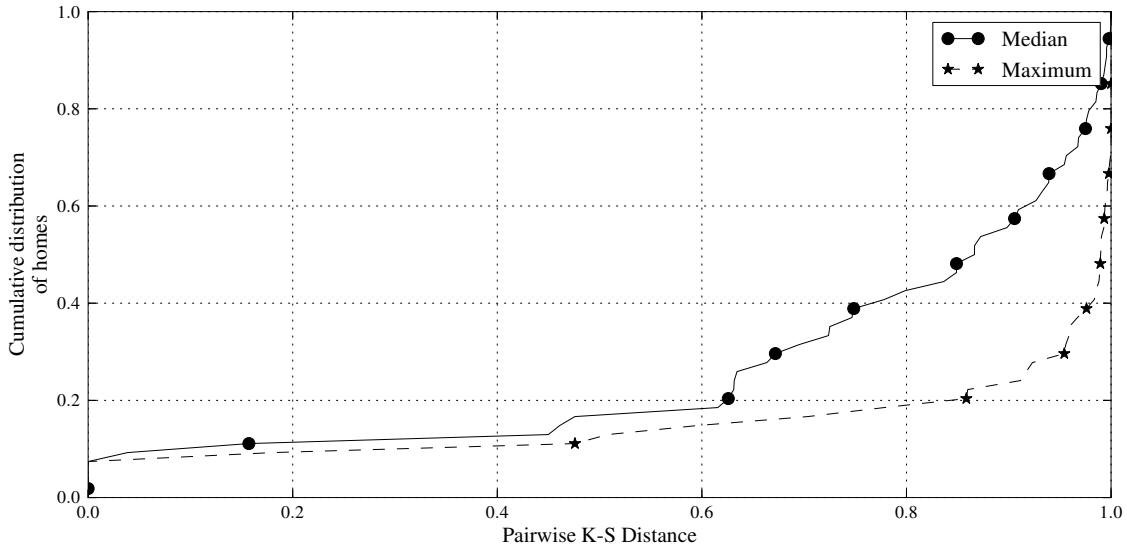


Figure 42: Average K-S distance for distributions of raw bitrates between pairwise devices within a home network, for all home networks.

Within a single home network, individual devices can experience very different wireless performance. We also studied the performance of individual devices in a home network and the extent to which wireless performance varies across devices in the same home network. We found many cases where the median wireless retransmission rates for a device was high. For the devices in the home shown in Figure 41, nearly all of the devices have median retransmission rates greater than 10%. Interestingly, one device experiences a high retransmission rates nearly all of the time, suggesting a persistent problem that may result from device placement, interactions between the access point and that device's driver, or some other cause.

To study how wireless performance varies across devices in a single home, we measure the K-S distance of the distributions of raw wireless bitrates between each pair of devices in each home. Figure 42 plots the median and the maximum pairwise K-S distance in each home. We find that more than 80% of homes have at least one pair of devices with a K-S distance of more than 0.6, indicating that most homes have at least one poorly performing device (due to either poor placement, poor hardware, or poor drivers). Future work could involve investigating the disparate performance across devices further and determining whether the variability in device performance is caused by any single factor.

Simultaneous communication is infrequent. Most of the homes in our deployment had more than one active device during our study. Interestingly, however, these devices often were not highly active at the same time. We measured one-second intervals and observed the number of times that multiple devices were sending at least 25 packets within the one-second interval. To our surprise, simultaneous communication was rare: for 85% of the one-second intervals on the 2.4 GHz band and 93% of the intervals on the 5 GHz band, we observed at most one device sending at least 25 packets in the interval. This observation relates to wireless contention, and it may imply that certain types of wireless contention are infrequent; we intend to explore this phenomena in more detail in future work.

6.4 Takeaways

We introduced WTF, a tool that runs on the gateway in a user's home network, that can provide visibility about whether performance problems exist inside the home network or elsewhere. One of the more significant challenges in executing this study involved designing a measurement tool that could operate within the tight constraints of a commodity home router and draw reasonable inferences from a single vantage point without a second monitor radio. Our results from 64 homes suggest when downstream throughput of a user's access link exceeds about 15 Mbits/s, the underlying cause of poor performance is more likely to be a poorly performing wireless network (when the downstream throughput exceeds 35 Mbits/s, the access network is never the problem).

We also found that TCP flows in home networks achieve only a small fraction of the available access link throughput, that wireless characteristics have a greater effect on the performance of user traffic as access link throughput increases, that the 5 GHz channel exhibits better performance than the 2.4 GHz band, and that distinct devices within the same home can see very different wireless performance.

This study opens several avenues for future work. First, although WTF can tell a user that their home wireless network is performing poorly, it does not offer any insights into the underlying causes. There is an acute need for methods that explain *why* various wireless performance problems exist in addition to where they are. Second, a follow-up to WTF could

attribute problems that home network users experience to a more complete and more specific set of causes: for example, end hosts and applications can sometimes introduce performance problems. A more complete diagnosis tool might also identify whether problems truly lie in the access ISP or further afield in the wide area.

CHAPTER 7

MEASURING AND MITIGATING WEB PERFORMANCE BOTTLENECKS IN BROADBAND ACCESS NETWORKS

As downstream throughput continues to increase, one might expect the Web to get faster at home, as well. Meanwhile, Internet service providers and application providers are increasingly cognizant of the importance of reducing Web page load times; even seemingly small differences in latency can introduce significant effects on usability (and revenue). The Bing search engine experiences reduced revenue of 1.2% with just a 500-millisecond delay [152], and a 400-millisecond delay resulted in a 0.74% decrease in searches on the Google search engine [31]. Forrester research found that most users expected online shopping sites to load in two seconds or fewer [108]. Content providers struggle to mitigate any network performance bottleneck that can slow down Web page loads in access networks by even tens of milliseconds. Thus, it is crucial to understand both how network properties of access networks such as latency can introduce bottlenecks in Web page load times and the extent to which various optimizations can help further mitigate these bottlenecks.

Towards this goal, in this chapter, we use measurements from a router-based Web performance measurement tool, Mirage, to analyze Web page load times to nine popular Web sites. This tool has been deployed in over 5,000 home networks as part of the FCC/SamKnows deployment in the US. We also deploy the tool in our own smaller deployment, BISmark. We examine how access network latency and throughput can introduce performance bottlenecks and evaluate how to mitigate these bottlenecks by deploying various caching optimizations on the router in the home network. Next, we demonstrate that caching on a home router can improve page load time, even if the router does not cache any content (*i.e.*, even if it only caches DNS records and TCP connections), and even if the end host or Web browser is already independently performing similar optimizations. Finally, we show how prefetching DNS records and TCP connections from the home router can improve cache hit rates; we

use a trace-based emulation to show that prefetching can help achieve these performance improvements in real home networks. We now describe each of these contributions in more detail.

First, we measure Web performance from 5,556 broadband access networks to nine popular Web sites and identify bottlenecks that contribute to Web page load time in these networks (Section 7.2). Our results suggest that latency is the main bottleneck for Web page load times for access links whose downstream throughput exceeds about 16 Mbits/s. Last-mile latency is an important contributor to the end-to-end latency, and an increase in last-mile latency of just 10 milliseconds can sometimes induce delays of hundreds of milliseconds for page load times of popular sites. In the case of small objects, we find that TCP latency overhead exceeds the actual download time of the object. Our results corroborate and quantify anecdotal evidence from users, Internet service providers, and content providers who are increasingly finding that latency is becoming a critical performance bottleneck [64, 108].

Second, we use Mirage to deconstruct page load time into its constituent components (*i.e.*, DNS lookup, TCP connection setup, content download) and show that even small improvements in latency can yield significant improvements in overall page load times (Section 7.3). To our knowledge, this paper presents the first study to compare the relative benefits of content caching, DNS caching, and TCP connection caching from within home networks. As part of this analysis, we explore how the page load time that Mirage measures relates to Web page load time measured by other tools (*e.g.*, Phantomjs, Web browsers). We find that latency is a significant contributor to all factors that affect page load time. These results—in conjunction with our results in Chapter 5 that observed that characteristics of the access network can introduce significant latency—present the case for *home caching*, the process of caching DNS lookups, TCP connections, and content from the home router.

Third, we deploy an OpenWrt module that performs various caching optimizations in home routers on the BISmark testbed and show that such a cache can yield improvements in page load time, even if it does not cache content, and *even if the browser is already performing similar optimizations* (Section 7.4). As expected, content caching offers the

Table 9: Performance metrics. For each per-object metric, Mirage measures the maximum, minimum, and average times for each object in the transaction.

Metric	Type	Description
Page load time	Total	The time to look up DNS, set up TCP connections and retrieve all objects.
DNS lookup time	Per Domain	The DNS lookout time for the main domain of the site.
Time to first byte	Per Object	The time from the initiation of the TCP connection to the arrival of the first byte of the requested object (including server processing time).
Object download time	Per Object	The time to download an object, excluding the DNS lookup time and time to first byte.

most significant reductions in page load time and can reduce page load times by up to 53% in some cases. Yet, simply caching TCP connections and DNS records at the home router can reduce mean page load times by 20% and 7%, respectively, even if the ISP and browser are also independently performing their own caching optimizations. We build on these insights to develop *popularity-based prefetching*, which prefetches DNS records and TCP connections for Web sites that are commonly accessed from a home network.

The Web performance measurements from the SamKnows deployment are available on the FCC Web site [151]. We have published both the Web performance measurements from the BISmark experiments [27] and the OpenWrt module that performs popularity-based prefetching [133].

7.1 Measuring Page Load Time

The notion of page load time depends on many factors, including the underlying network, the design of the Web site, and the endhost, including the browser. Every browser downloads pages using different methods (and different optimizations), and the pages themselves may be optimized for different browsers and devices. In this paper, we explore how the characteristics of broadband networks affect Web page load times. Page load time will

necessarily vary depending on the above characteristics. Thus, to conduct controlled experiments of the effects of network characteristics on page load time, we use a single tool on a common platform to measure page load times as various network conditions change. We now explain the design of the active measurement tool that we use, and how the numbers it yields compares to that of a browser.

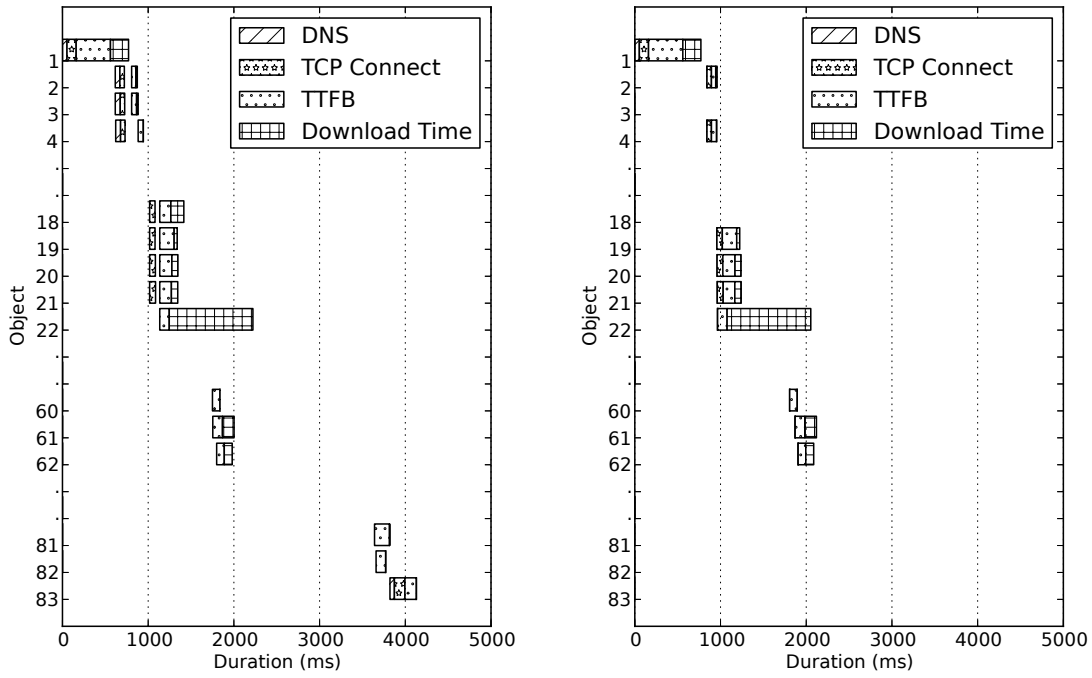
7.1.1 Mirage: Home Router-Based Web Testing

We use measurements from the FCC/SamKnows deployment of routers across the US [155]. The deployment uses a custom tool, which we call *Mirage*, to measure page load time. Mirage is a headless Web client designed by SamKnows for deployment on home router devices; the initial deployment of Mirage was sponsored by the Federal Communications Commission (FCC) on more than 5,000 homes across the US. We also use it in the BISmark deployment.

7.1.1.1 How Mirage Works

Mirage downloads the home page of a Web site and parses it to determine the static objects that are needed to render the page. It then performs all the DNS lookups at once before downloading the rest of the objects. The tool is based on `libcurl`, which can decompose the overall page load time into the download times for individual objects. Mirage separates the page load time into the time to perform DNS lookups, the time to first byte (which combines the TCP connection time and the server processing time), and the actual load time for each object. It uses persistent TCP connections if the server supports them and up to eight parallel TCP connections to download objects. Because Mirage uses many of the basic network optimizations that a browser uses including persistent TCP connections and parallel TCP connections, it approximates the that a real browser might see, even though it does not emulate any particular browser. Table 9 shows the performance metrics for each download and how the performance measurements from Mirage compare to those from `webpagetest.org`.

Mirage is ideal for our goal of studying the effect of the characteristics of broadband access networks on Web performance for a number of reasons. First, because Mirage can



(a) Real Browser

(b) Mirage

Figure 43: Comparison of a real browser (from `webpagetest.org`) with Mirage for `www.ebay.com`; some objects are omitted. The browser optimizes object retrieval differently, and also retrieves more objects. Still, the performance bottlenecks are similar for both.

be deployed directly on home routers, it provides measurements from a single platform that is directly connected to the access link, thereby normalizing some of the effects that might otherwise affect page load times (*e.g.*, the use of different browsers, home network effects). Second, Mirage is already in use in the large FCC/SamKnows deployment, which provides longitudinal measurements of the same set of diverse access links. Because Mirage breaks down each object into its constituent parts, it exposes important performance bottlenecks in access networks. Finally, Mirage is freely available and portable; we have used Mirage in our own BISmark deployment.

7.1.1.2 Validation

Comparing Mirage to real browser behavior Mirage's behavior differs from a browser in several ways. We explain these differences using the example download in Figure 43. This

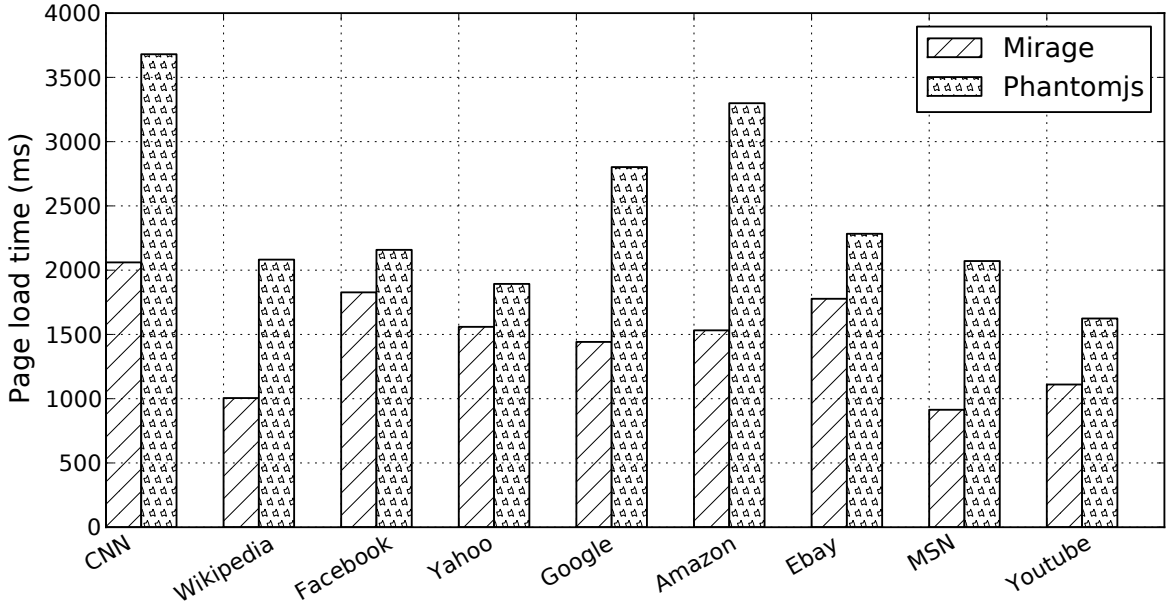


Figure 44: Comparison of Mirage to Phantomjs. We see that Mirage underestimates the page load times: real load times are higher than what we measure.

Table 10: Properties of the Web sites in our data set. The values represent the average of the parameters over all transactions. *Objects* denotes the number of objects that must be downloaded; *lookups* are the number of DNS lookups required; and *connections* are the number of unique TCP connections the client set up to download all objects in the page. The number of connections depends on whether the server supports persistent connections, whether the objects are located on one or more domains, and the order in which objects are retrieved. *Size* denotes the number of bytes for all of the objects for a site.

Target	**Objects**			**Lookups**			**Connections**			**Size (KB)**		
	SK	B		SK	B		SK	B		SK	B	
		U	n-U		U	n-U		U	n-U		U	n-U
`edition.cnn.com`	26	25	26	4	4	4	12	12	12	1199	1022	1023
`www.amazon.com`	24	31	32	4	4	4	21	24	23	589	840	851
`www.ebay.com`	29	33	32	12	14	14	16	17	19	595	613	615
`www.facebook.com`	8	8	7	2	2	2	7	8	7	437	389	289
`www.google.com/mobile`	32	20	20	1	1	1	8	8	8	1398	289	291
`www.msn.com`	24	24	54	8	8	8	14	14	16	377	348	641
`www.wikipedia.org`	16	15	16	1	1	1	16	15	15	56	56	56
`www.yahoo.com`	74	69	66	7	7	8	32	32	29	927	887	818
`www.youtube.com`	8	7	8	2	2	2	9	8	8	488	423	414

figure shows the partial breakdown of the page load time for Ebay both using a real browser through `webpagetest.org` and Mirage. First, Mirage waits to download the home page before processing it and downloading subsequent objects; in contrast, many modern browsers start downloading objects as they process the home page. This difference is visible in objects 2–4 in Figure 43, where the real browser initiates the download before object 1 is

complete. Mirage also performs DNS queries for all the unique domains after parsing the home page before downloading any of the remaining objects; it adds the maximum DNS time to the total time to download the objects. Although Mirage performs the DNS queries before downloading any objects, the effect on total page load time is not significant. The time to retrieve each individual object and the time spent on each component of the object retrieval is nearly identical for the case of the real browser and Mirage.

Mirage also downloads a slightly different, smaller set of objects than a real browser. Modern Web sites, especially content-heavy ones, employ active scripts. These scripts result in additional processing latency and also frequently result in the browser downloading more objects. Mirage only processes static objects, so it downloads a subset of the objects that are downloaded by a real browser, usually resulting in a smaller page load time than a normal browser would see. For example, in Figure 43, the browser downloads 83 objects, while Mirage downloads 62 objects.

Validation with Phantomjs To understand how the differences between Mirage and a real browser affect overall page load time, we compare Mirage's measurements to those of a real browser environment. For this comparison, we use Phantomjs [131], a headless client that implements the Webkit browser engine and has a JavaScript API. Phantomjs is used extensively for Web benchmarking and testing [132].

Figure 44 shows the median page load time of Mirage and Phantomjs for an emulated 10 Mbits/s access link with a last-mile latency of 40 ms; the results also hold for lower latency (10 ms) links. Mirage always underestimates the actual page load time because it downloads fewer objects than a real browser would. Depending on the amount of dynamic content on the site, the difference may vary: in some cases, Mirage underestimates load times by up to 50%; in others, its measured page load time is close to the load time that Phantomjs sees.

The implementation differences between Mirage and real browsers imply that the page load times that Mirage sees may not reflect the times that any real browser would see. Yet, page load times will always differ across different browsers, and we do not aim to estimate

Table 11: The SamKnows deployment in the US.

ISP	Number of homes	Avg. last-mile latency (ms)	Avg. downstream tput (Mbits/s)
AT&T	718	18	8.7
Cablevision	209	6	34.1
CenturyLink	248	23	5.8
Charter	579	7	27.4
Comcast	932	8	22.8
Cox	618	7	21.2
Mediacom	242	14	18.7
TimeWarner	952	8	16.0
Qwest	299	30	10.0
Verizon	608	5	38.0
Windstream	251	20	5.9

Table 12: The BISmark deployment across the world.

Location	# of homes	Avg. last-mile latency (ms)	Avg. downstream tput (Mbits/s)
US	43	12	24.2
Europe	3	24	8.3
N. Amer. (non-U.S.)	3	15	5.5
E. Asia/Australia	4	3	46.5
Southeast Asia	8	12	5.7

page load time from any particular browser. Our goal is to illustrate how components of network latency (*e.g.*, DNS lookup time, TCP connect time) contribute to Web page load times. Mirage decomposes Web page load time into these components, which will be the same regardless of browser or any optimizations that a browser might perform. Mirage also allows us to evaluate how optimizations that a browser might perform can mitigate various network bottlenecks under different network conditions. When we evaluate the effects of different optimizations in Section 7.4, we again "close the loop" by showing that the benefits as predicted by Mirage are realized, even from a browser that is already performing its own optimizations.

7.1.2 Deployment

We use data from the FCC/SamKnows deployment, spanning 5,556 homes in the US; and the BISmark deployment, spanning 61 homes across the world. For both deployments, we use Mirage to characterize nine sites as chosen by SamKnows/FCC. Because it is difficult

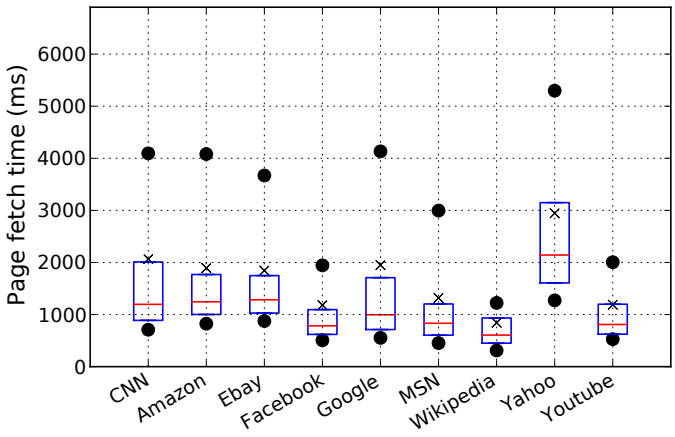
to measure a large number of Web sites from home routers and constrained access links, we focus on nine of the most popular sites around the world. Table 10 shows the sites that we measure and their properties. All of the measurements from both deployments are public. In some cases, the properties of the sites are different across deployments. Because both deployments use the same tool, these differences may result from either the vantage point or the time that the measurements were conducted. Because we do not compare measurements across the two deployments, however, these differences are not consequential.

- We analyze measurements from 5,556 participants in the FCC/SamKnows study across 11 ISPs in the US from October 2011, April 2012, and September 2012. We include only users who have reported more than 100 measurements during the duration of the study from ISPs with more than 100 users. Table 11 summarizes the deployment; our previous study describes it in more detail [155]. We report on results from September 2012; the results that we present are consistent with measurements from October 2011 and April 2012.
- We also analyze measurements from the BISmark deployment described in Chapter 4. Table 12 characterizes the homes in the BISmark deployment by region. *BISmark-US* collects measurements from 44 routers across the US, and *BISmark-nonUS* collects measurements from 18 routers in other parts of the world, including Europe, North America (excluding the US), and Asia. The BISmark-US data is from May 17–June 7, 2012, and the BISmark-nonUS data is from May 24–June 7, 2012. The URLs are the same for both datasets, and we rely on DNS to locate the local version of a site, if one exists.

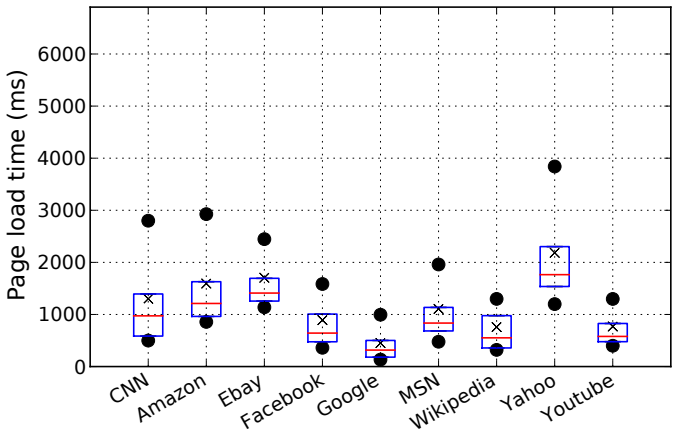
7.2 Characterizing Bottlenecks

We study page load times for popular sites and evaluate how downstream throughput and latency of an access link affects these times.

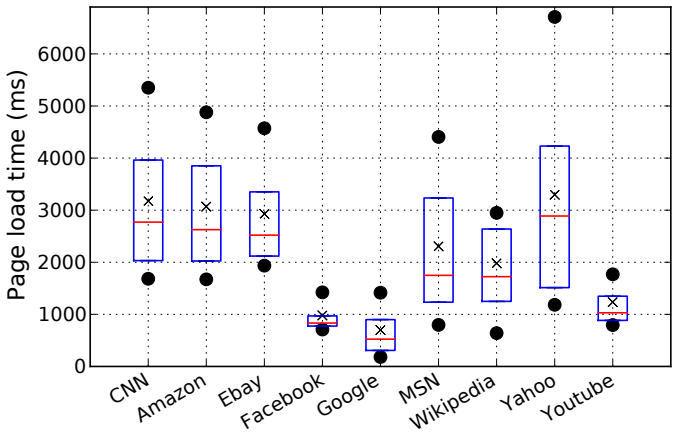
We find that *latency is a performance bottleneck for Web page load time in access networks whose downstream throughput exceeds about 16 Mbits/s.* Last-mile latency (*i.e.*, the latency between the home network and the first hop in the access ISP) is a significant overall



(a) SamKnows



(b) BISmark-US



(c) BISmark-nonUS

Figure 45: Page load times for popular sites. The lower edge of each box indicates the 25^{th} percentile of the distribution of page load times for each site, the upper edge is the 75^{th} percentile, the middle line is the median, the cross represents the average, and the dots the 10^{th} and 90^{th} percentile page load times.

contributor to both DNS lookup times and the time to first byte. Therefore, *even when Web caches are deployed at the edge of access ISPs, optimizations in the home network that reduce the effects of last-mile latency can still offer significant performance improvements in page load time.*

7.2.1 Page Load Times of Popular Web Sites

We study the page load times to nine popular Web sites. Figure 45a shows the load times for access links in the SamKnows deployment while Figure 45b shows load times for links in the BISmark deployment (both in the US). Figure 45c shows page load times for links in the BISmark deployment outside of the US. As expected, page load time varies both by site and the location of the access network. Some variability results from differences in page size and design (see Table 10); the largest four sites (CNN, Yahoo, Amazon, and Ebay) also have the largest load times (*e.g.*, the median for CNN in the US is more than one second).

Figure 45c shows that access links outside of the US typically experience higher page load times for a given site than links in the US. The median and variance is higher for all sites we measure from outside the US, as well. A few sites have different sizes depending on the from which the page is requested, but most performance differences result from the fact that content is farther away from clients that are outside of the US. Figure 46 illustrates this phenomenon; the figure shows that the average time to first byte is in general higher in most regions outside the US. Our measurements also indicate that the global deployment of content distribution networks is somewhat spotty for certain sites. Sites with more expansive CDNs (*e.g.*, Google, YouTube) have low median and maximum page load times, whereas other sites have more variable performance, both in the US and abroad. Even Google has relatively poor performance from Southeast Asia; we discussed this phenomenon with network operators at Google, who confirmed that Google's CDN deployment is not extensive in that region.

7.2.2 Effects of Downstream Throughput

We study how page load time and its components vary with downstream throughput using measurements from the SamKnows deployment. We use the 95^{th} percentile of the distribution of downstream throughput over the duration of the measurements for a given user to capture the capacity of each access link. We group access links according to downstream throughput into seven bins that reflect common ranges of Internet access plans in the dataset: 0–1 Mbits/s, 1–2 Mbits/s, 2–4 Mbits/s, 4–8 Mbits/s, 8–16 Mbits/s, 16–32 Mbits/s, and 32–64 Mbits/s. Figure 47a shows the median page load time for each category for five representative sites.

Median page load time decreases as downstream throughput increases, up to about 16 Mbits/s. As downstream throughput increases further, page load times decrease only modestly. For example, the median time for CNN is 8.4 seconds for links with throughput 0–1 Mbits/s and 1.3 seconds when throughput is 8–16 Mbits/s. Yet, when downstream throughput exceeds 32 Mbits/s, the page load time is 790 ms, only slightly better than for links with 8–16 Mbits/s.

We study how each component of page load time varies with access link throughput. Page load time is heavily influenced by the maximum DNS lookup time and the maximum time to download a single object (regardless of any caching, parallel lookups, or other optimizations), so we can interpret the maximum times for these values as a lower bound on page load time. Large objects also have a correspondingly lower TCP overhead. Figure 47b shows how these values decrease as throughput increases; each point shows the median value for the group of hosts with the corresponding range of downstream throughput. As downstream throughput increases to 32–64 Mbits/sec, object load time decreases from 3.2 seconds to 530 ms. In contrast, the time to first byte decreases as the throughput of the link increases from 0–1 to 1–2 Mbits/s, but does not improve further for higher values of throughput. DNS lookup time decreases from about 50 ms to about 15 ms. In summary, as downstream throughput increases beyond 8–16 Mbits/sec, time to first byte and DNS times become a larger component of page load time and depend more on latency than on throughput—even for large objects.

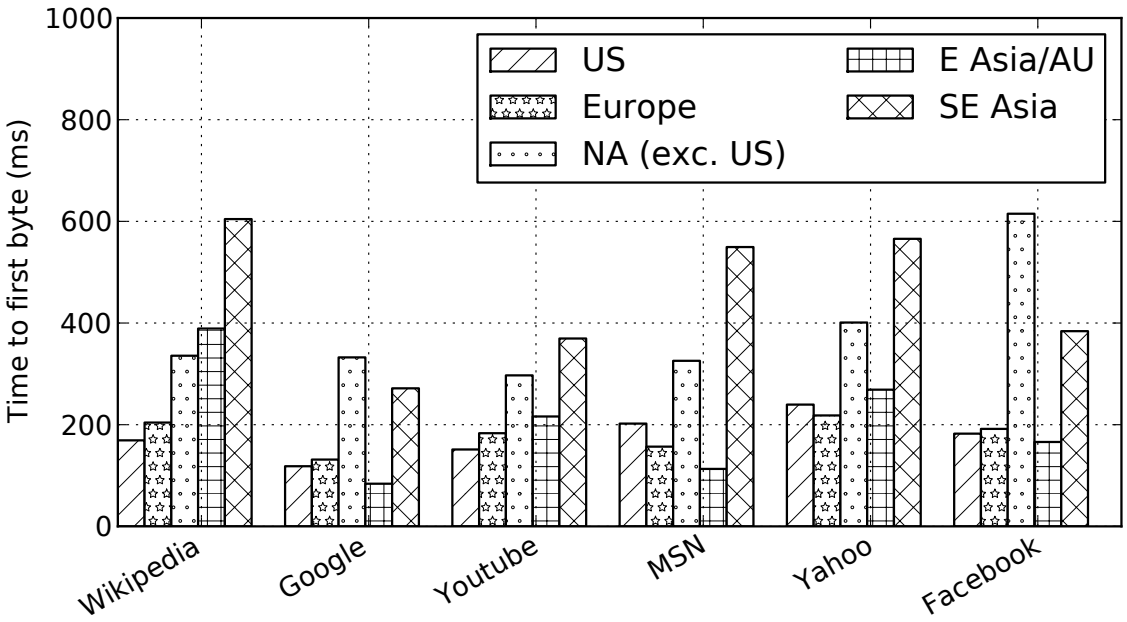
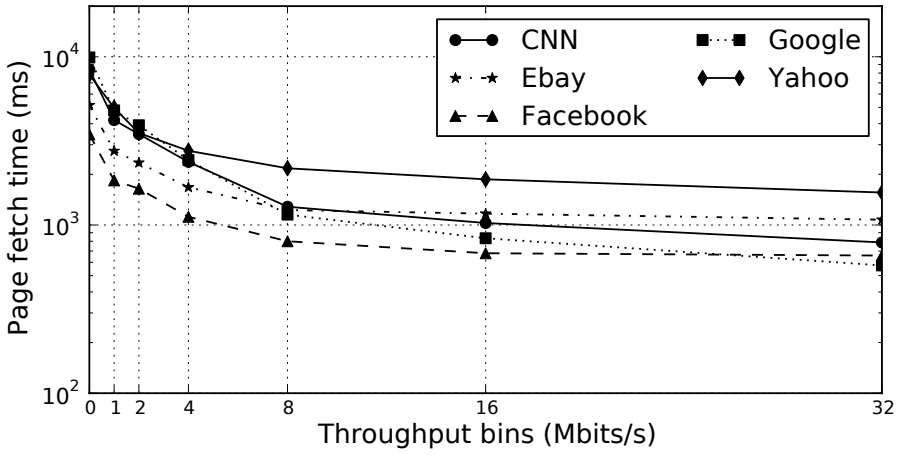


Figure 46: Average time to first byte to six representative sites from BISmark clients broken down by location.

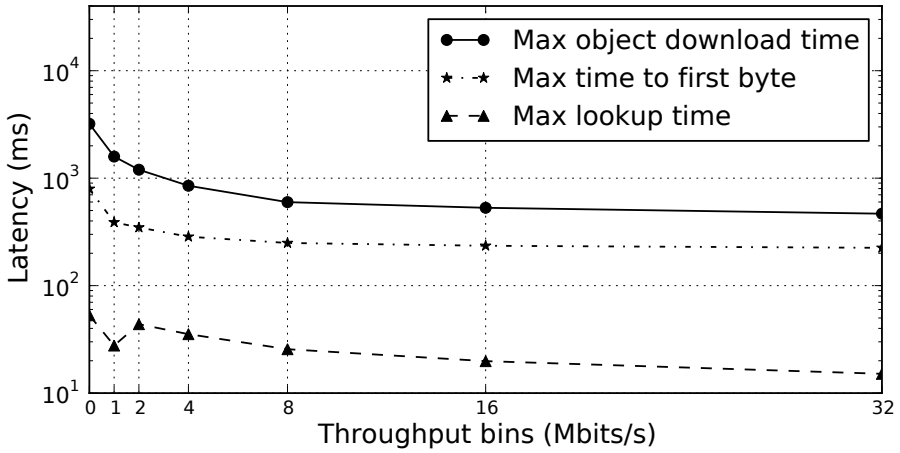
Page load times for clients outside of the US illustrate the effect of latency on page load time. For example, the average page load time for Facebook is 1.7 seconds in southeast Asia, 990 ms in east Asia and Australia, and 924 ms in Europe. Table 12 shows that clients in east Asia have higher average throughput than Europe, but do not necessarily see a corresponding improvement in page load times because latency bottlenecks negate the effects of higher throughput.

7.2.3 Effects of Last-Mile Latency

Our study in Chapter 5 observed that last-mile latency in access links in the US contributes significantly to end-to-end latencies [155]. To study the effect of last-mile latency on page load times, we group access links into 10 ms bins, according to the 10th percentile last-mile latency. Figure 48a shows the median page load time for each group for five representative sites. In general last-mile latency has a multiplicative effect on page load time, which is intuitive because it affects all packets in the transaction. The increase we see is not monotonic because other factors such as downstream throughput also affect page load time and some groups have more links than others: 75% of links have less than 10 ms last-mile latency.



(a) Page load times

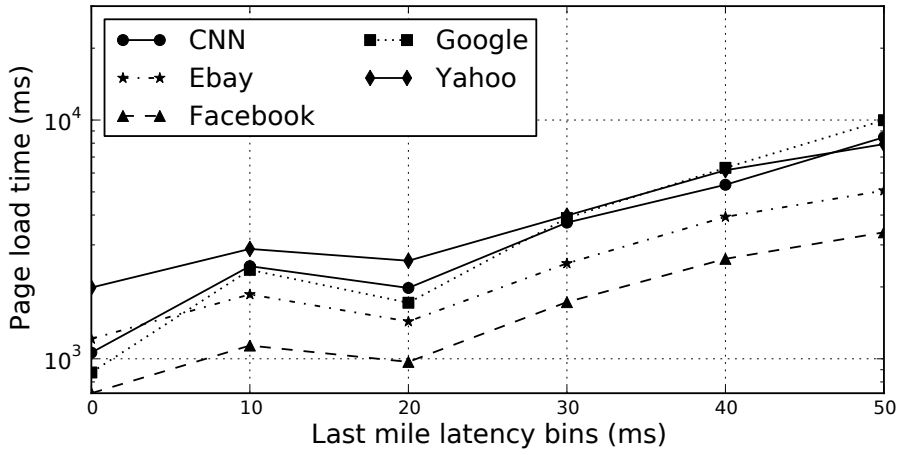


(b) Components of page load time

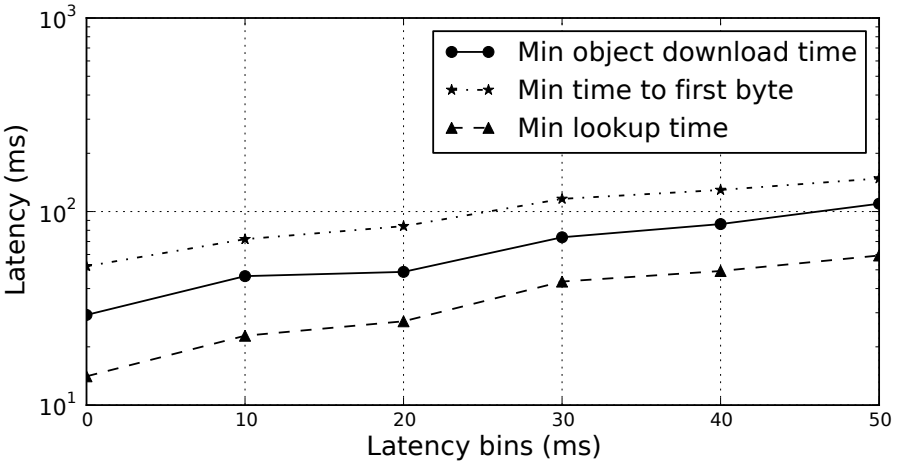
Figure 47: Page load times decrease with downstream throughput, but only up to 8–16 Mbits/s. X-axis labels denote the start of each throughput bin (*e.g.*, "0" is the set of users with downstream throughput up to 1 Mbits/s.) (SamKnows)

Even accounting for differences in throughput, an increase in last-mile latency of only 10 ms can result in a few hundred milliseconds increase in page load time. Increasing downstream throughput beyond 16 Mbits/s yields only marginal improvements, but decreasing last-mile latency can have a larger impact on page load time.

To understand the overhead of TCP on small objects, we look at the minimum object download time and compare it to the minimum time to first byte and DNS lookup time. Figure 48b shows the median of each of these values for each group of links. For smaller objects, the time to establish the TCP connection to the server is often greater than the time to actually transfer the object. Although the effect is magnified in smaller objects, we



(a) Page load times



(b) Components of page load time

Figure 48: Page load times increase with last-mile latency. X-axis labels denote the start of each latency bin. (SamKnows)

saw that it exists across all objects; in fact, we found that the average time to first byte ranges from 6.1% (for Yahoo) to 23% (for Wikipedia) of the total page load time. We also observed that last-mile latency can be as much as 23% of the time to first byte.

7.2.4 Conclusion: Optimize Latency

The results in this section suggest that for the increasing number of homes that have downstream throughput above 16 Mbits/s, the most effective way to reduce Web page load time is to focus on reducing latency as much as possible.

Improving Object Fetch Time Because most pages have many objects, the time to download all of them can dominate other factors when downstream throughput is small. Links with less than 16 Mbits/s will thus benefit the most from content caching. For links with higher downstream throughput, however, reducing object download time is less important than reducing latency. Thus, intuition suggests that faster access links can realize significant improvements in Web page load time simply by caching DNS records and TCP connections, without even having to cache any content. Our evaluation in Section 7.3 confirms this intuition.

Improving DNS Lookup Time Figure 48b shows that minimum lookup time increases as last-mile latency increases. Minimum lookup times are about 15 ms, and they increase as the last-mile latency increases. The only approach to eliminate the last-mile latency is to cache DNS records inside the home itself. Even when the resolver is well-placed in the access ISP network, DNS lookup latencies are bound by last-mile latency (previous work confirms this observation [4]).

Improving Time to First Byte Optimizations in the home network cannot improve server processing time, but they can improve TCP connection setup time. The connection setup time depends on the round-trip latency to the server. Web service providers use content distribution networks to place servers as close to users as possible. Figure 46 shows that servers are in general closer to homes in the US, but even users in the US can experience slowdowns in TCP connection setup time due to the last-mile latency. Client-side optimizations such as connection caching [66] can reduce this latency by maintaining TCP connections to popular sites, thereby reducing overhead for new connections.

7.3 The Case for Home Caching

We use Mirage to evaluate the benefits of DNS caching, TCP connection caching, and content caching in a home network. Although the optimizations that we evaluate are well known, the placement of these optimizations in the home router is new. Our analysis in this section offers two important contributions: (1) It is the first study to quantify the benefits of

deploying them in the home network, where many users now access Web content; (2) To our knowledge, it is also the first study to quantify both the relative benefits to DNS caching, TCP connection caching, and content caching across a large number of popular sites and the holistic benefits of performing all three optimizations together.

Perhaps the most important takeaway from this section is that *optimizations that reduce latency can significantly reduce page load time, even if content caching is not used.* This result offers good news, since so much Web page content is dynamic, and since caching a large amount of content on end-hosts or inside home networks may prove infeasible. This finding also emphasizes the importance of placing optimizations inside the home network, since even the last-mile access link can introduce additional latency on the order of tens of milliseconds [155]. Similar optimizations already exist in the browser; we explain how a home cache provides additional benefits beyond browser-based optimizations in Section 7.4.

7.3.1 Experiment Setup

We develop a controlled experiment to investigate how deploying three different caching optimizations in the home—DNS caching, TCP connection caching, and content caching—contribute to reducing page load time. We use the BISmark deployment described in Section 7.1 for our measurements. Mirage runs on the BISmark router; it uses a locally running DNS resolver and an HTTP proxy. `dnsmasq` is a lightweight caching DNS resolver [58] that caches up to 150 domains and honors the TTL values of the lookups. To evaluate TCP connection and content caching, we use `polipo`, a HTTP proxy that splits the TCP connection by opening a connection to the requested domain on behalf of the client and communicates with the client over a separate connection and reuses TCP connections where possible. We run `polipo` with a 4 MByte cache in RAM.

Measuring Baseline Performance Table 13 illustrates how we measure baseline performance and compare it with performance for each optimization. To measure the three *baseline* performance measurements, the client first performs the following three sets of requests:

Table 13: The measurements we perform to evaluate the benefits of DNS, connection, and content caching in the home.

Measurement	Proxy Location	DNS	Conn.	Content
Baseline Measurements				
No Proxy, ISP DNS	—			
Cold Home Proxy	—	•		
ISP Proxy	Network	•	•	•
Optimizations				
Home DNS	Home	•		
Home Conn. Caching	Home	•	•	
Home Proxy	Home	•	•	•

1. **ISP DNS Cache ("No Proxy, ISP DNS").** The client clears the local DNS cache and fetches the page directly from the server. This measures the baseline performance of fetching the Web page. The DNS lookups required for this measurement may reflect caching in the ISP's DNS resolver.

2. **Empty Caching Proxy in the Home ("Cold Home Proxy").** The client fetches the same page by directing the request through a fresh `polipo` instance running in the router. Because `polipo`'s cache is empty at this point, this measurement reflects the performance of a "cold" proxy. This step takes advantage of any DNS caching that `dnsmasq` performs in the previous step.

3. **Shared ISP Caching Proxy ("ISP Proxy").** We cannot control a Web cache in a real ISP's network, so we approximate the behavior of an ISP cache by deploying a caching proxy in our university network. To measure the benefits of performing DNS, connection, and content caching at a shared proxy (the most common setup for content caches), the client first fetches the page through a `polipo` proxy running on a university server to warm the cache. It then immediately repeats the step. We perform the measurements from eleven BISmark routers that are less than 35 ms away from the proxy, so that it emulates nearby ISP caches.

Quantifying the Benefits of Caching in Home Networks After collecting the baseline measurements, the client then performs three additional requests to measure the relative benefit of performing different optimizations in the home.

4. **Home DNS caching only ("Home DNS").** The client fetches the same page directly from the servers. This measures the benefits of DNS caching in the home (since `dnsmasq` caches the DNS responses from earlier measurements).

5. **Home proxy with DNS caching, persistent connections, and content caching ("Home Proxy")** The client fetches the page through the local `polipo` again; this measurement takes advantage of DNS, content and connection caching, since the proxy would have cached any cacheable objects and reused TCP connections where possible from the requests in the "Cold Home Proxy" experiment.

6. **Home proxy with DNS caching and persistent connections only ("Home Connection Caching")** The client clears the `polipo` cache on the home router — this gets rid of the content in the cache, *but keeps the TCP connections from the previous step alive.* It then fetches the page through the proxy again. All content is retrieved again from the origin service, but the TCP connections from the previous step are reused.

These experiments allow us to isolate the effects of (1) the relative benefits of performing DNS, connection, and content caching inside the home network and (2) placing a cache inside the home versus elsewhere (*e.g.*, in the ISP).

7.3.2 Effects of Home Caching on Latency

Benefits of DNS caching in the home vs. in the ISP To quantify the benefits of DNS caching in the home, we compare the maximum lookup time for the *Home DNS Cache* and *No Proxy, ISP DNS* cases. Figure 49 shows the CDF of the improvement in the maximum lookup time for the page load. In the median case, DNS caching in the home reduces the maximum lookup time by 15–50 ms, depending on the site. Certain clients outside the US can reduce their lookup time by several hundred milliseconds for certain sites like Ebay and CNN by caching DNS responses in the home. For a small fraction of cases, DNS caching in the home actually slightly impairs performance; in these cases, the detriment in small and may result from changes in network conditions between experiments (this effect occurs for some of our TCP connection caching and content caching experiments, too).

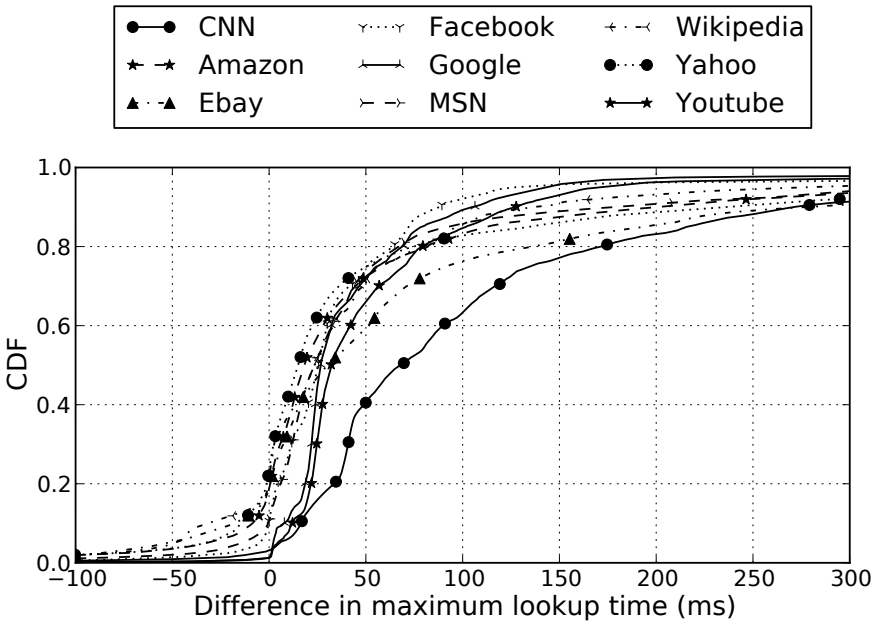


Figure 49: Caching DNS in the home can reduce the maximum DNS lookup time by 15–50 ms. (*Home DNS* Measurement vs. *No Proxy, ISP DNS* Measurement)

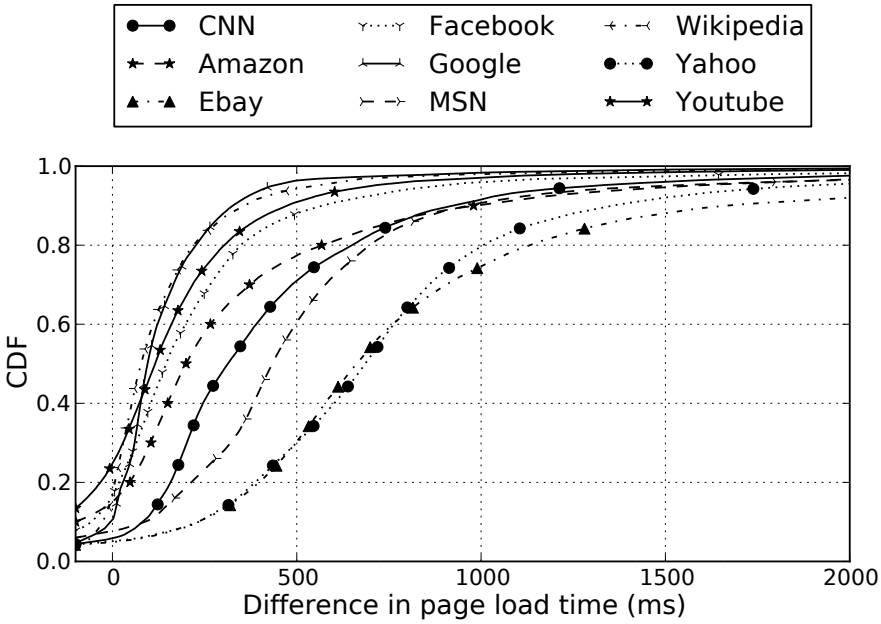


Figure 50: Connection caching in the home can reduce median page load times by 100–750 ms.(*Home Connection Proxy* vs. *Home DNS* Measurements)

Benefits of TCP connection caching in the home Figure 50 shows the additional improvement in page load time due to connection caching by measuring the difference between the load times for the *Home Connection Caching* and the *Home DNS Cache* measurements. The median improvement varies from 100–750 ms depending on the site. Ebay and Yahoo experience the most improvement in load times because both sites require many objects from many domains to render; connection caching can significantly reduce TCP overhead in such cases.

7.3.3 Effects of Home Caching on Throughput

Benefits of content caching vs. connection caching Figure 51 shows the improvement in page load time due to content caching over connection caching. We compute the improvement by subtracting the page load time for the *Home Proxy* experiment from that for the *Home Connection Caching* experiment. Caching content inside the home can decrease median page load times in the US by 75–400 ms over connection caching, depending on the site. Obviously, sites with more cacheable content will benefit more. Our analysis shows that this benefit is even more significant for clients outside the US; at least 20% of clients experienced an improvement of 500 ms or more for all sites.

Benefits of content caching in the home vs. in the ISP We compare the Web page load time when using a remote HTTP proxy (the *ISP Proxy* measurement from Table 13) versus using a local HTTP proxy running on the router (the *Home Proxy* measurement). Figure 52 shows that a proxy in the home can offer a median improvement in page load time of 150–600 ms, depending on the site. Yahoo and CNN experience the most benefits, likely because these pages are larger and have many objects (Table 10). A cache in the upstream ISP is still constrained by the access link's throughput and last-mile latency, while a local cache is not. For some sites, the remote proxy performs better than the home proxy about 20% of the time, perhaps because of varying access link characteristics across tests (due to cross traffic) or because the proxy is in a university network that potentially has better connectivity to these sites.

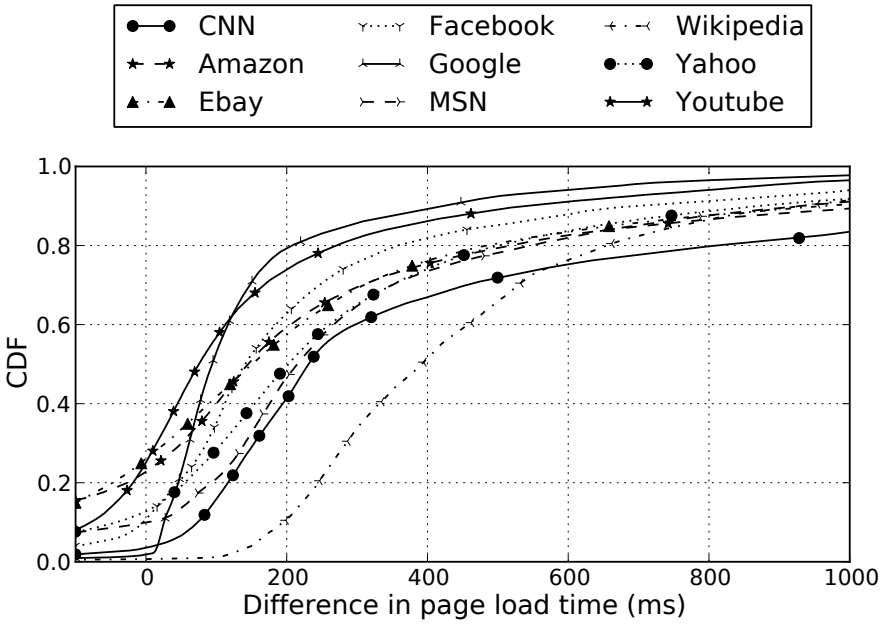


Figure 51: Content caching reduces the median page load time by 75–400 ms over connection caching alone. For sites with more cacheable content, the benefit is greater (*Home Proxy* vs. *Home Connection Caching* Measurements)

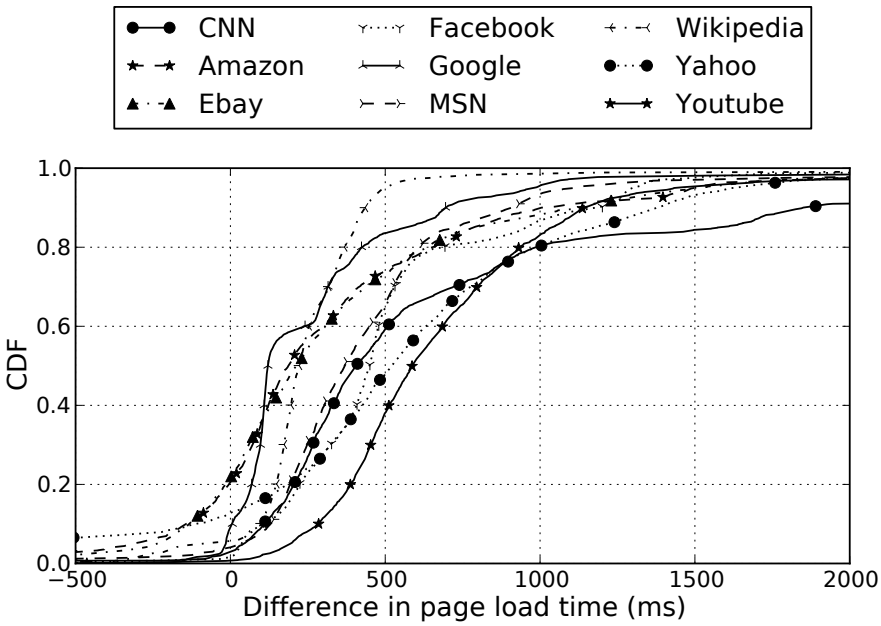
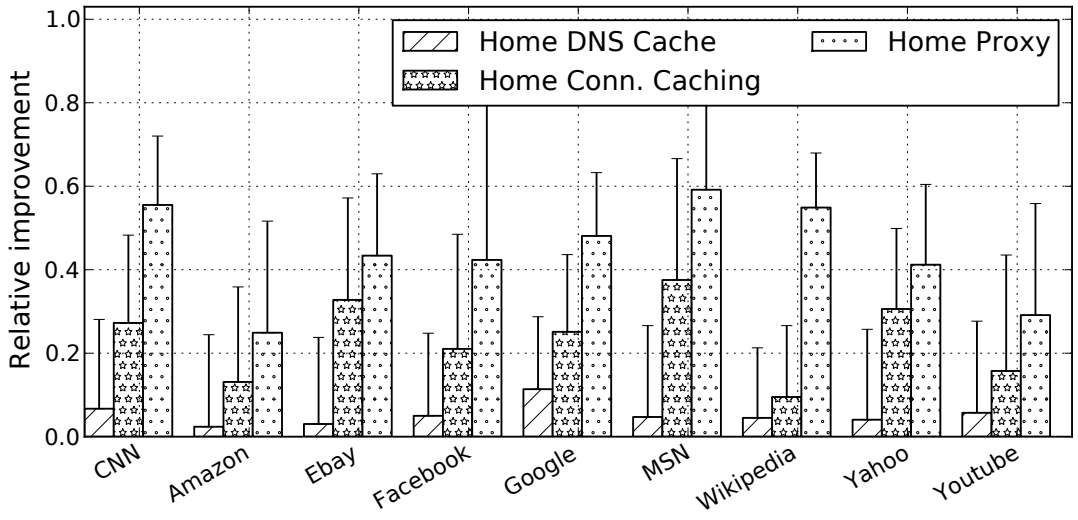
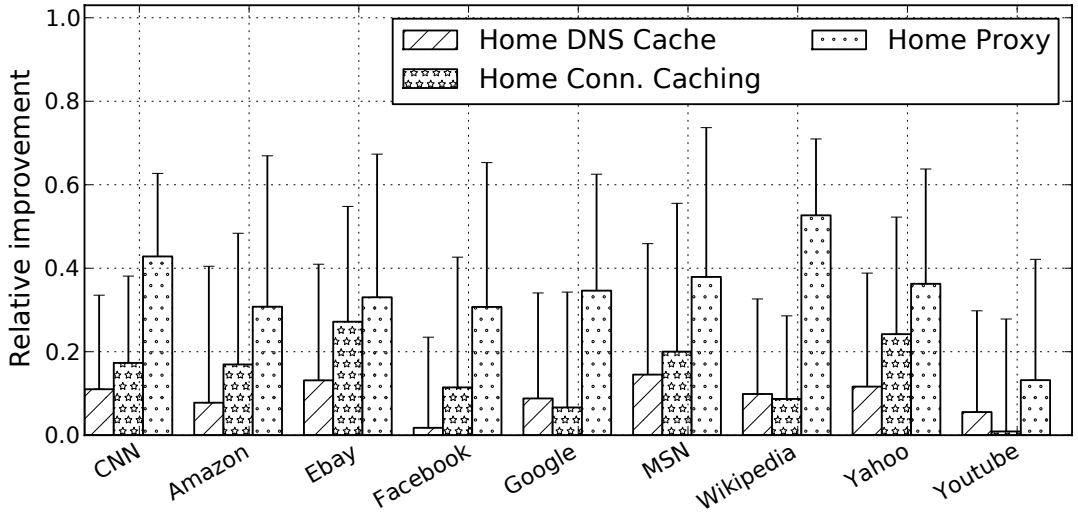


Figure 52: Running a proxy in the home improves median page load times by 150–600 ms versus running a proxy in the ISP. (*Home Proxy* vs. *ISP Proxy* Measurement)



(a) BISmark-US



(b) BISmark-nonUS

Figure 53: Average relative improvement in page load times for various optimizations, as observed from the router. Error bars denote standard deviation.

7.3.4 Putting It Together

We now quantify the collective benefit from performing all three optimizations. We use Mirage's *No Proxy, ISP DNS* measurement as the baseline. We compute the *relative improvement* as $(b - v)/b$, where b is the baseline load time and v is the optimized load time.

Figure 53a shows the relative improvement of each optimization relative to the baseline of performing no optimizations in BISmark-US. Applying all three optimizations improves

page load time by as much as 60%. *Even without content caching, connection caching can yield up to a 35% improvement in load time, and DNS caching alone can improve page load time by as much as 10%.* Figure 53b shows the relative improvement for clients outside the US. The improvement is slightly less than for the users within the US because the absolute page load times for users outside the US are already higher (Figure 45). The variance is also higher because of the wide range of operating conditions outside the US (Table 12 and Figure 46). The improvements we measure represent the best-case scenario because the quick succession of Web page retrievals will always induce a cache hit at the router. In the next section, we explore how to design a cache in the home that achieves high cache hit rates in practice.

7.4 Home Caching in Practice

The previous section demonstrated how home caching can improve page load times in the best-case scenario, but the results measured the benefits in isolation from other optimizations (*e.g.*, browser caching) and also assumed that these cache hits could be realized in practice. Thus, our experiments from Section 7.3 raise two questions:

1. *Are the performance benefits that home caching offers significant, in light of the fact that browsers already perform some amount of caching and prefetching?* To answer this question, we retrieve Web pages from a laptop in a home network using Phantomjs under a variety of emulated access link characteristics with various caching optimizations enabled on the home router. (Section 7.4.2)

2. *How can users inside a home network practically realize cache hits on a home cache?* Prefetching DNS queries and maintaining open TCP connections to all sites visited by a user is not practical. To intelligently determine which DNS records and TCP connections to cache, we implemented a lightweight router-based *popularity-based prefetching* system that prefetches and caches DNS records and maintains active TCP connections to popular domains to help improve cache hit rates. We analyzed cache hit ratios resulting from this system with a trace-driven simulation using passive traces collected from twelve homes and show that it improves hit rates for DNS and TCP

connection caches significantly [159].

7.4.1 Popularity-based Prefetching

We design, implement, and evaluate a *popularity-based caching and prefetching* systemthat prefetches DNS records and keeps TCP connections to Web servers active based on the sites that users in the household visit most frequently. We develop a proof-of-concept OpenWrt module, which is publicly available [133]. The system consists of (`dnsmasq`) and (`polipo`), instrumented to track popular DNS lookups and HTTP domains respectively. The system tracks popularity and refreshes DNS lookups and maintains an active TCP connection to popular domains by using a simple caching mechanism. The system aims to maximize the hit rate of the DNS and TCP connection caches. The two parameters that affect the hit rate are (1) *the number of domains to be tracked*: the system actively prefetches DNS records and maintains active connections to these domains; the system maintains the two lists separately; and (2) *timeout thresholds*: the system tracks the time since a lookup or a TCP connection was requested to a domain and removes the domain from the popular list if this time exceeds a threshold. The system does not prefetch content but exploits any content caching that `polipo` performs by default.

7.4.2 Benefits of Home Caching from Browsers

We evaluate the benefit of home caching as seen from a browser by analyzing the improvement in page load times from Phantomjs with various caching optimizations enabled. We measure page load times from a Web browser using Phantomjs running on a Linux laptop that is connected through a wired link to a Netgear home router; the router shapes the uplink to represent different home network throughputs and latencies. We use two settings for the downstream throughput of the access link (10 Mbits/s and 20 Mbits/s) and three settings for the latency of the access link (0 ms, 20 ms, and 40 ms). We download each of the sites 25 times for each setting. These parameters approximate common access-link throughputs and last-mile latencies. The router also runs popularity-based prefetching, discussed in Section 7.4.1.

Home caching complements browser optimizations Figure 54 shows the relative improvement in Web page load times as a result of deploying various caching optimizations from a home router, as observed from the laptop running Phantomjs. Because we had problems with Phantomjs using `polipo` for Google, we omit results for it. DNS caching improves page load times by as much as 7%; connection and DNS caching improve load times by about 20%; all three optimizations together reduce load times by as much as 60%, depending on the site. The benefits as measured using Phantomjs are lower, but they are comparable to what we observe at the router in Figure 53. Although the improvements in page load time that a browser realizes obviously depends on many factors, our results demonstrate that *home caching complements existing optimizations that browsers already perform.*

To further illustrate these benefits, we show the page load time for Ebay, for a single experiment. Figure 55 shows how DNS and TCP connection caching improves per-object downloads. It shows the time to download the first ten objects of the site using Phantomjs. We measure the total time for the first ten objects. The left plot shows the baseline case without optimization, and the second plot shows the benefits that the browser observes when the home router performs DNS and connection caching. We observe an 6.4% reduction in page load time for the home page alone when the home router performs both DNS and TCP caching. The improvements are relatively more significant for smaller objects. The numbers next to the objects show the percentage improvement in the baseline. All objects show some improvement; objects that are cacheable see more improvement.

7.5 Takeaways

We presented the first large-scale study of Web performance bottlenecks in broadband access networks. We first characterize performance to nine popular Web sites from 5,556 access networks using Mirage, a router-based Web measurement tool, and identify factors that create Web performance bottlenecks. Regardless of the optimizations that clients, servers, and browsers may perform, the access link is a fundamental limit to the efficacy of optimizations. As throughput increases, latency becomes more important, to the extent that

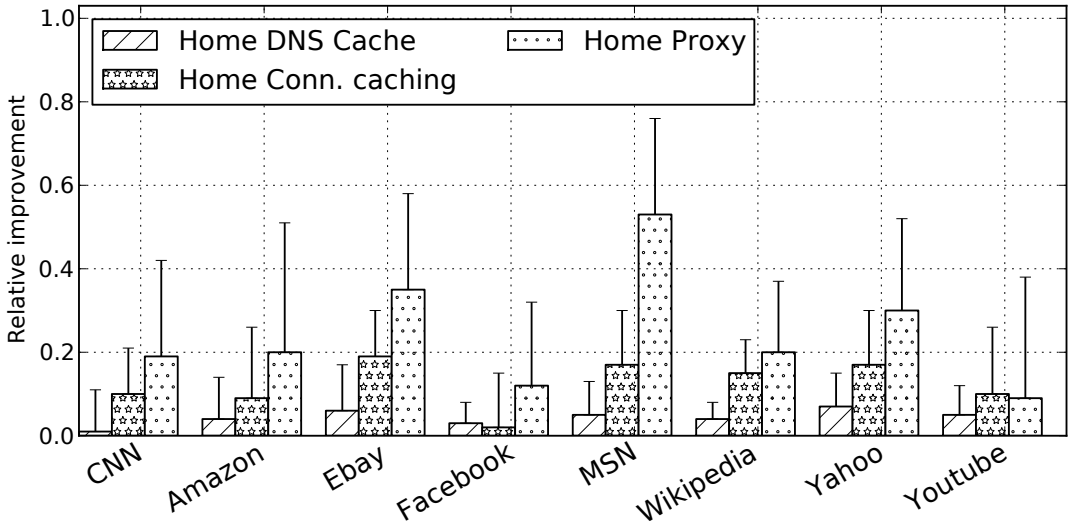


Figure 54: Average relative improvement in page load times for various optimizations, as observed from the browser. Error bars denote standard deviation.

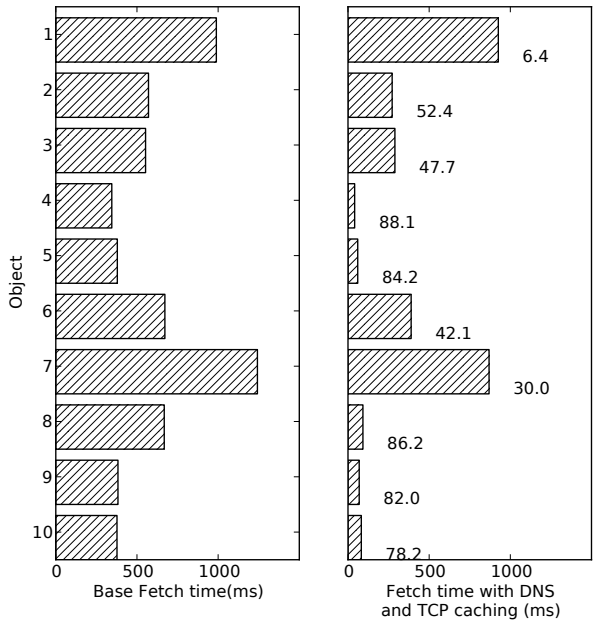


Figure 55: The time to load the first ten objects for Ebay using Phantomjs. We plot the total download time for the first ten objects. The first plot shows the breakdown for the base case; the second when there is DNS and TCP connection caching. The numbers in the second plot denote the percentage improvement over the base case.

even having caches on the edge of ISPs could prove to be insufficient. We show that Web page load times stop improving as throughput rates increase beyond 16 Mbits/s. We also show that last-mile latency contributes to the time required for performing DNS lookups, completing TCP connection setup, and downloading Web objects and can be up to 23% of the time to first byte.

Moreover, we evaluate techniques to mitigate latency bottlenecks by implementing *home caching*, which performs DNS and TCP connection caching and prefetching inside the home network. We demonstrate that home caching can improve Web performance, even if the ISP and browser are already independently performing similar optimizations. TCP connection caching alone can improve Web page load times by up to 20%, depending on the site. Performing all three optimizations together from the home router can reduce page load times by up to 53%.

Although the optimizations we describe can be implemented anywhere in the home network, we believe that the home router is a reasonable location for caching and prefetching DNS records and TCP connections. In addition to further reducing page load times, even in the presence of similar optimizations that browsers already implement, it is worth noting that there are many Web-based applications that do not run from a standard browser. Any device in the home network that retrieves Web content can benefit from home caching. Indeed, as Singhal and Paoli state, "apps—not just browsers—should get faster too" [150]. Our publicly available OpenWrt module for popularity-based prefetching may serve as yet another important component for reducing page load times in home networks.

CHAPTER 8

CONCLUDING REMARKS

The Internet has grown exponentially in recent years and has now become a central part of our lives. It is therefore critical that we understand its properties and its problems so that we can continue to make it better, enable more services, and make it more accessible to a larger slice of the population. With the growth of broadband networks as the primary mode of Internet access in many parts of the world, focus has shifted towards understanding how these networks perform. While there have been many attempts to characterize broadband performance, the unique nature of these networks make it hard to do so. Home and access networks (the "last mile" in the end-to-end path) are host to a wide variety of technologies and applications. Each of these technologies are different in their operating conditions and how they potentially affect application performance.

The biggest challenge with characterizing performance in the last mile is that there is no single notion of performance; it could mean different things depending on context or who you are. It could mean the raw capacity of the access link or the wireless link, or it could mean whether the application is getting bottlenecked or not, or it might just mean the latency that the application sees. While many studies have attempted to shed light on performance in the last mile, they all suffer from a common drawback which makes such studies incomplete: they lack a holistic view of the last mile. This dissertation proposed the use of the home gateway as a vantage point that solves this drawback; we showed how it allows us to do a thorough characterization of the home network, the access link, and even certain applications.

8.1 Summary of contributions

In this dissertation, we showed how viewing the last mile through the prism of the home gateway allows us to better understand and tackle performance issues. We demonstrated the importance of good data and the need to develop new methods and techniques to aid

our pursuit. In the context of broadband networks, the ability to obtain repeatable and unclouded measurements – both active and passive, of the access link, the home network, and of application performance – from the gateway is critical to our understanding of these networks. It allows us to draw meaningful conclusions and to develop new methods and techniques for solving problems in these networks. The dissertation offered the following contributions:

1. **The design and development of a testbed to experiment on home and access networks.** We described the design, development, and growth of BISmark, a home gateway testbed to enable experimentation on home and access networks. BISmark has a presence in nearly 200 homes in more than 20 countries. We describe our design choices, and the technical and social challenges we faced during the growth of the deployment. We demonstrated its value to the networking research community as a platform for conducting continuous and repeatable experiments that offer a unique view into such networks.

2. **Using home gateways to characterize performance, and understand and mitigate bottlenecks in the last mile.** We use home gateways deployments to characterize last mile performance. We look at three important components in the last mile: the access link, the home network, and a popular application, the Web. We first evaluate access link performance in the US using active measurements from the gateway; we evaluate existing methods and propose new ways to characterize broadband networks and how it should be presented to users. We then characterize last mile bottlenecks using passive measurements of user traffic; we develop and deploy new techniques to localize performance bottlenecks to within the home network or outside. We then use active measurements from a custom Web browser emulator to show how the last-mile, particularly the latency introduced by it, can be a critical bottleneck in Web performance. We show the limits of throughput in improving page load time, characterize the overhead of the last-mile on page load time, and propose techniques to mitigate them.

8.2 Lessons learnt

Over the course of working on the components that make up this dissertation, we have learnt valuable lessons that apply to performance characterization on the edge of the Internet, particularly using large-scale infrastructure. We list the important ones below.

Lesson 1 *Last-mile characterization is hard. The sheer variety of technologies that exist in the last mile, and the applications that people use, makes it non-trivial to adequately characterize performance.*

It is worth reiterating a point that has been made multiple times in this dissertation: performance characterization in the last mile is more than any one parameter. Working on this problem forces us to step outside the comfort zone of any one network layer that we might be proficient in. For example, wireless network subtleties force us to understand how the MAC and physical layers work, while understanding application performance require us to delve deep into the application in question. Ultimately, the question of how these things interact and affect user experience requires us to understand end-to-end principles. And then, of course, there is the challenge of actually presenting this information to the users in a form that they will understand. One approach that we have proposed consist of providing a "nutrition label" to the user that would help break down the properties of the network so that the user can understand it better and make informed decision about their choice of ISP and service plan. The label consists of a suite of parameter values that cover different use cases [156]. Similar approaches might be required to ensure that users understand what is happening inside their network.

Lesson 2 *A good vantage point is vital for good characterization. In particular, an available vantage point that provides an unobstructed view of the last-mile enables us to approach characterization in a more structured way.*

One reason why characterizing the last mile is difficult is because data from good vantage points has been traditionally hard to obtain. The complexity and heterogeneity of the last mile makes it essential to have a vantage point that can minimize the number of

confounding factors that affect characterization. The gateway serves as a good vantage point in this situation, but it is also important to note that it is not a magic pill; it is a light-weight, resource-constrained device with limited memory and computational power. These constraints limit the kind of measurements we can collect; *e.g.*, involved application performance diagnosis may not be possible due to limits to the amount of data we can collect and process on the device. The lack of visibility into the Internet beyond the ISP is also a drawback. Ultimately, understanding the limitations of current approaches and developing new methods for tackling issues is critical in the rapidly changing environment that is the modern Internet.

Lesson 3 *New settings call for new measurement metrics and techniques. It is sometimes beneficial to extend old techniques and metrics, and maybe even define new ones to suit the new setting.*

The uniqueness of the gateway as a vantage point, as well as its shortcomings, made it necessary to rethink how we characterize the last mile. In Chapter 5, we go beyond regular throughput and latency measurements and show how newer techniques such as last mile latency, and latency under load provide more insight. In Chapter 6, we describe how to detect bottlenecks using passively collected data under severe hardware constraints, while in Chapter 7 we use a Web browser emulator that is targeted towards identifying network bottlenecks. We also implement optimizations to improve performance on the gateway. The above techniques would be either not possible or not required in a different setting, but in the gateway they allow us to gain new insight into last-mile performance.

Lesson 4 *Reappraising conventional wisdom gives us new insight. With the rate at which the Internet keeps changing, a fresh look at accepted truths can throw up surprises.*

When we started out with trying to understand Web performance, we were faced with a body of work that went back more than a decade. However, we had a unique viewpoint that previous work did not have. Our work in Chapter 7 reaffirms a lot of the previous work — it showed how latency is the primary bottleneck especially as throughput goes up — but

our data offered new insight into the nature of the latency and throughput bottlenecks in the last mile. We were able to tease out the contribution of various latency components in a way that had not been done before. We were also able to test optimizations that had been proposed before but in a different setting, *i.e.*, in the home and characterize how they can provide even more benefit.

Lesson 5 *Deployment in real homes is hard. Technical challenges exist, but real-world constraints and the social aspect make us rethink the concept of "scale" when it comes to such deployments.*

The goal of having a deployment in the form of devices that people use in their day to day lives proved more challenging than the mere development of a stable platform. Finding trustworthy volunteers, who in turn trusted us to not affect their home network or violate their privacy, and engaging with them so that they have an incentive to help us keep the deployment up, proved challenging. Designing experiments that could collect good data without affecting performance for the user or violating their privacy, while also being mindful of the resource constraints was another major challenge. We had to go through multiple revisions of hardware, firmware, software, data collection and privacy policies during the course of developing and deploying the platform.

8.3 Future Work

Developing a more user-friendly platform for home networking research Chapter 4 describes how we attempt to enable experimentation on the BISmark platform. However, much work remains to be done to ensure that experiments can be enabled in a secure manner and with automated verification, with tight bounds on what sort of traffic they can generate and how much traffic they can generate. There are also concerns about the ethics of running certain kinds of experiments from certain regions — testing for censorship in a highly restrictive region might land the user in trouble with the authorities – that make the verification problem harder. However, a platform with the geographic and network diversity that BISmark has can potentially be of great service to the research community.

Mobile Networks As mobile networks continue to grow and get faster, they introduce new performance issues. Although modern mobile networks are engineered to an impressive level, performance still lags considerably compared to broadband networks. Application (and protocol) performance is not as good or as reliable as on comparable wired networks. One of the primary reasons for this is latency, in particular access latency. This is a gap that needs to be closed as mobile becomes the primary mode of Internet access for billions of people. Mobile networks are in many ways fundamentally different from broadband networks; network design and the high cost of data make us rethink system and measurement design, particularly in developing regions. However, we believe we can apply insights, and measurement and design methods from this dissertation to solve these problems.

End-to-end fault detection and localization Chapter 6 deals with localizing faults in the last mile to either the wireless network or the access network. While this is a first step, it merely scratches the surface of a much bigger problem. It is difficult to diagnose the true source of problems in the end-to-end path — it could be the application itself, end host, the server, or anything in between. Better application performance diagnosis is badly needed; *e.g.*, if a video starts buffering, how can we diagnose and localize the issue? Some principles and insights from Chapter 6 could potentially be applied. However, this is potentially a significant new area of research.

Characterizing applications We characterize the performance of the Web in Chapter 7 because it is one of the most popular applications today. However there are many other critical applications that we do not fully understand, *e.g.* video streaming. It may not be possible or even desirable to characterize all of them from the gateway. It is clear, however, that we need to do so to help users get the best out of the network; such work may even help enable newer applications as broadband and mobile networks get faster.

REFERENCES

[1] "tcptrace: A TCP Connection Analysis Tool." `http://irg.cs.ohiou.edu/software/tcptrace/`.

[2] Adya, A., Bahl, P., Chandra, R., and Qiu, L., "Architecture and techniques for diagnosing faults in ieee 802.11 infrastructure networks," in *Proceedings of the 10th annual international conference on Mobile computing and networking*, MobiCom '04, (New York, NY, USA), pp. 30–44, ACM, 2004.

[3] Ager, B., Schneider, F., Kim, J., and Feldmann, A., "Revisiting cacheability in times of user generated content," in *INFOCOM IEEE Conference on Computer Communications Workshops, 2010*, pp. 1–6, IEEE, 2010.

[4] Ager, B., Mühlbauer, W., Smaragdakis, G., and Uhlig, S., "Comparing dns resolvers in the wild," in *Proceedings of the 10th annual conference on Internet measurement*, IMC '10, (New York, NY, USA), pp. 15–21, ACM, 2010.

[5] Aggarwal, B., Bhagwan, R., Das, T., Eswaran, S., Padmanabhan, V. N., and Voelker, G. M., "Netprints: diagnosing home network misconfigurations using shared knowledge," in *Proceedings of the 6th USENIX symposium on Networked systems design and implementation*, NSDI'09, (Berkeley, CA, USA), pp. 349–364, USENIX Association, 2009.

[6] Ahmed, N., Ismail, U., Keshav, S., and Papagiannaki, K., "Online estimation of rf interference," in *Proceedings of the 2008 ACM CoNEXT Conference*, CoNEXT '08, (New York, NY, USA), pp. 4:1–4:12, ACM, 2008.

[7] Akella, A., Maggs, B., Seshan, S., and Shaikh, A., "On the performance benefits of multihoming route control," *IEEE/ACM Transactions on Networking*, vol. 16, Feb. 2008.

[8] Akella, A., Maggs, B., Seshan, S., Shaikh, A., and Sitaraman, R., "A measurement-based analysis of multihoming," in *Proc. ACM SIGCOMM*, (Karlsruhe, Germany), Aug. 2003.

[9] Akella, A., Pang, J., Maggs, B., Seshan, S., and Shaikh, A., "A comparison of overlay routing and multihoming route control," in *Proc. ACM SIGCOMM*, (Portland, OR), Aug. 2004.

[10] Akella, A., Seshan, S., and Shaikh, A., "Multihoming performance benefits: An experimental evaluation of practical enterprise strategies," in *Proc. USENIX Annual Technical Conference*, (Boston, MA), June 2004.

[11] Al-Fares, M., Elmeleegy, K., Reed, B., and Gashinsky, I., "Overclocking the Yahoo! CDN for faster Web page loads," in *Proceedings of Internet Measurement Conference*, 2011.

[12] Altman, E., Avrachenkov, K., and Barakat, C., "A stochastic model of tcp/ip with stationary random losses," in *ACM SIGCOMM*, 2000.

[13] Andersen, D. G., Balakrishnan, H., Kaashoek, M. F., and Morris, R., "Resilient Overlay Networks," in *Proc. 18th ACM Symposium on Operating Systems Principles (SOSP)*, (Banff, Canada), pp. 131–145, Oct. 2001.

[14] Antoniades, D., Athanatos, M., Papadogiannakis, A., Markatos, E., and Dovrolis, C., "Available bandwidth measurement as simple as running wget," in *Proc. of Passive and Active Measurement Conference (PAM 2006)*, pp. 61–70, Citeseer, 2006.

[15] Arlitt, M., Krishnamurthy, B., and Mogul, J., "Predicting Short-transfer Latency from TCP arcana: a Trace-based Validation.," in *Proc. ACM SIGCOMM Internet Measurement Conference*, (New Orleans, LA), Oct. 2005.

[16] "Does broadband need its own government nutrition label?." `http://arstechnica.com/tech-policy/news/2009/10/does-broadband-needs-its-own-government-nutrition-label.ars`, Oct. 2010. Ars Technica.

[17] Barford, P. and Crovella, M., "Critical path analysis of tcp transactions," in *IEEE/ACM Transactions on Networking*, 2000.

[18] Bastian, C., Klieber, T., Livingood, J., J.Mills, and Woundy, R., *Comcast's Protocol-Agnostic Congestion Management System*. Internet Engineering Task Force, Dec. 2010. RFC 6057.

[19] Belshe, M., "A Client-Side Argument for Changing TCP Slow Start." `http://goo.gl/UDKXz`.

[20] Belshe, M., "More Bandwidth Doesn't Matter (much)." `http://goo.gl/OIv47`.

[21] Bernardi, G. and Marina, M. K., "Bsense: a system for enabling automated broadband census: short paper," in *Proc. of the 4th ACM Workshop on Networked Systems for Developing Regions (NSDR '10), June 2010.*, 2010.

[22] Biaz, S. and Vaidya, N. H., "Discriminating congestion losses from wireless losses using inter-arrival times at the receiver," in *Proceedings of the 1999 IEEE Symposium on Application - Specific Systems and Software Engineering and Technology*, ASSET '99, (Washington, DC, USA), IEEE Computer Society, 1999.

[23] Bicket, J., "Bit-rate selection in wireless networks," Master's thesis, Massachusetts Institute of Technology, Feb. 2005.

[24] Bicket, J., Aguayo, D., Biswas, S., and Morris, R., "Architecture and evaluation of an unplanned 802.11b mesh network," in *Proc. ACM Mobicom*, (Cologne, Germany), Sept. 2005.

[25] "BISmark Project Partners with Comcast." `http://noise-lab.net/2013/05/19/bismark-project-partners-with-comcast`. Retrieved: September 2013.

[26] "BISmark privacy statement." http://projectbismark.net/participant/privacy.

[27] "BISmark Web Performance data." http://data.gtnoise.net/bismark/imc2013/webperf/bismark_webperf_data.tgz.

[28] "BISmark uploads." http://uploads.projectbismark.net.

[29] Bode, K., "FCC: One Million Speedtests and Counting." http://www.dslreports.com/shownews/FCC-One-Million-Speedtests-And-Counting-109440, July 2010.

[30] Botta, A. Dainotti, A. and Pescapé, A., "Multi-protocol and multi-platform traffic generation and measurement." IEEE INFOCOM, Demo session, May 2007.

[31] Brutlag, J., "Speed matters for Google Web search." http://services.google.com/fh/files/blogs/google_delayexp.pdf, June 2009.

[32] Butkiewicz, M., Madhyastha, H., and Sekar, V., "Understanding website complexity: Measurements, metrics, and implications," in *Proc. Internet Measurement Conference*, (Berlin, Germany), Nov. 2010.

[33] Caceres, R., Douglis, F., Feldmann, A., Glass, G., and Rabinovich, M., "Web proxy caching: The devil is in the details," June 1998.

[34] Calder, M., Fan, X., Hu, Z., Katz-Basset, E., Heidemann, J., and Govindan, R., "Mapping the expansion of google's serving infrastructure," in *Proceedings of the 13nd ACM SIGCOMM conference on Internet measurement*, IMC '13, 2013.

[35] Canadi, I., Barford, P., and Sommers, J., "Revisiting broadband performance," in *Proceedings of the 2012 ACM Conference on Internet Measurement Conference*, IMC '12, (New York, NY, USA), pp. 273–286, ACM, 2012.

[36] Cao, J., Cleveland, W. S., Gao, Y., Jeffay, K., Smith, F. D., and Weigle, M., "Stochastic models for generating synthetic http source traffic," in *IN PROCEEDINGS OF IEEE INFOCOM*, 2004.

[37] Cappos, J., Beschastnikh, I., Krishnamurthy, A., and Anderson, T., "Seattle: a platform for educational cloud computing," in *ACM SIGCSE Bulletin*, vol. 41, pp. 111–115, ACM, 2009.

[38] Cardwell, N., Savage, S., and Anderson, T., "Modeling tcp latency," in *Proc. IEEE INFOCOM*, (Tel-Aviv, Israel), Mar. 2000.

[39] Carlson, R., "Network Diagnostic Tool." http://e2epi.internet2.edu/ndt/.

[40] Chen, Y., Mahajan, R., Sridharan, B., and Zhang, Z.-L., "A provider-side view of web search response time," in *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM*, SIGCOMM '13, pp. 243–254, ACM, 2013.

[41] Cheng, Y., Bellardo, J., Benko, P., Snoeren, A. C., Voelker, G. M., and Savage, S., "Jigsaw: Solving the puzzle of enterprise 802.11 analysis," in *Proc. ACM SIGCOMM*, (Pisa, Italy), Aug. 2006.

[42] Cheng, Y. and Others, *TCP Fast Open*, Sept. 2011. http://www.ietf.org/id/draft-cheng-tcpm-fastopen-00.txt.

[43] Cheng, Y.-C., Afanasyev, M., Verkaik, P., Benkö, P., Chiang, J., Snoeren, A. C., Savage, S., and Voelker, G. M., "Automating cross-layer diagnosis of enterprise wireless networks," *SIGCOMM Comput. Commun. Rev.*, vol. 37, pp. 25–36, Aug. 2007.

[44] Chetty, M., Sundaresan, S., Muckaden, S., Feamster, N., and Calandro, E., "Measuring broadband performance in south africa," in *Proceedings of the 4th Annual Symposium on Computing for Development*, ACM DEV-4 '13, (New York, NY, USA), pp. 1:1–1:10, ACM, 2013.

[45] Chetty, M., Sundaresan, S., Muckaden, S., Feamster, N., and Calandro, E., "Measuring broadband performance in south africa," in *Proceedings of the 4th ACM Annual Symposium on Computing for Development*, DEV 4, 2013.

[46] Cho, K., Fukuda, K., Esaki, H., and Kato, A., "The impact and implications of the growth in residential user-to-user traffic," in *ACM SIGCOMM 2006*, 2006.

[47] Chu, J. and Others, *Increasing TCP's Initial Window*, Oct. 2011. http://tools.ietf.org/html/draft-ietf-tcpm-initcwnd-01.

[48] Cohen, E. and Kaplan, H., "Prefetching the means for document transfer: A new approach for reducing Web latency," in *Proc. IEEE INFOCOM*, vol. 2, (Tel-Aviv, Israel), pp. 854–863, Mar. 2000.

[49] Cohen, E. and Kaplan, H., "Proactive caching of DNS records: Addressing a performance bottleneck," in *Symposium on Applications and the Internet (SAINT)*, pp. 85–94, 2001.

[50] "Comcast FAQ." http://customer.comcast.com/Pages/FAQViewer.aspx?Guid=024f23d4-c316-4a58-89f6-f5f3f5dbdcf6, Oct. 2007.

[51] Compton, C.L. Woundy, R. and Leddy, J., "Method and packet-level device for traffic regulation in a data network." U.S. Patent 7,289,447 B2, Oct. 2007.

[52] "Compuware." http://www.compuware.com/en_us/application-performance-management/products/application-aware-network-monitoring/web-services/overview.html.

[53] Croce, D., En-Najjary, T., Urvoy-Keller, G., and Biersack, E., "Capacity Estimation of ADSL links," in *Proc. CoNEXT*, Dec. 2008.

[54] de Donato, W., Sundaresan, S., Feamster, N., Teixeira, R., and Pescapé, A. in *USENIX NSDI Poster Session*, NSDI'11, 2011.

[55] Dischinger, M., Marcon, M., Guha, S., Gummadi, K., Mahajan, R., and Saroiu, S., "Glasnost: Enabling end users to detect traffic differentiation," in *Proceedings of the 7th USENIX conference on Networked systems design and implementation*, pp. 27–27, USENIX Association, 2010.

[56] Dischinger, M., Haeberlen, A., Gummadi, K. P., and Saroiu, S., "Characterizing residential broadband networks," in *Proc. ACM SIGCOMM Internet Measurement Conference*, (San Diego, CA, USA), Oct. 2007.

[57] "DNS Prefetching (or Pre-Resolving)." `http://blog.chromium.org/2008/09/dns-prefetching-or-pre-resolving.html`.

[58] "Dnsmasq." `http://thekelleys.org.uk/dnsmasq/doc.html`.

[59] Droms, R., *Dynamic Host Configuration Protocol*. Internet Engineering Task Force, Oct. 1993. RFC 1531.

[60] Droms, R., *Dynamic Host Configuration Protocol*. Internet Engineering Task Force, Mar. 1997. RFC 2131.

[61] Dukkipati, N., Refice, T., Cheng, Y., Chu, J., Herbert, T., Agarwal, A., Jain, A., and Sutin, N., "An argument for increasing tcp's initial congestion window," *SIGCOMM Comput. Commun. Rev.*, vol. 40, pp. 26–33, June 2010.

[62] "Emulab." `http://www.emulab.net/`.

[63] Erman, J., Gerber, A., Hajiaghayi, M., Pei, D., and Spatscheck, O., "Network-aware forward caching," in *Proceedings of the 18th international conference on World wide web*, 2009.

[64] "FCC Measuring Broadband America Report." `http://www.fcc.gov/measuring-broadband-america/2012/july`, July 2012.

[65] "National Broadband Plan." `http://www.broadband.gov/`.

[66] Feldmann, A., Caceres, R., Douglis, F., Glass, G., and Rabinovich, M., "Performance of web proxy caching in heterogeneous bandwidth environments," in *Proc. IEEE INFOCOM*, (New York, NY), Mar. 1999.

[67] Filastò, A. and Appelbaum, J., "Ooni: Open observatory of network interference," in *USENIX FOCI*, Aug. 2012.

[68] Flach, T., Dukkipati, N., Terzis, A., Raghavan, B., Cardwell, N., Cheng, Y., Jain, A., Hao, S., Katz-Bassett, E., and Govindan, R., "Reducing web latency: The virtue of gentle aggression," *SIGCOMM Comput. Commun. Rev.*, vol. 43, pp. 159–170, Aug. 2013.

[69] Gettys, J., "Bufferbloat." `http://www.bufferbloat.net/`.

[70] "Glasnost: Bringing Transparency to the Internet." `http://broadband.mpi-sws.mpg.de/transparency`.

[71] "Grenouille." `http://www.grenouille.com/`.

[72] Gribble, S. and Brewer, E., "System Design Issues for Internet Middleware Services: Deductions from a Large Client Trace," in *Proc. 1st USENIX Symposium on Internet Technologies and Systems (USITS)*, (Monterey, CA), Dec. 1997.

[73] Grover, S., Sundaresan, S., Park, M. S., Burnett, S., Kim, H., Ravi, B., and Feamster, N., "Peeking behind the nat: An empirical study of home networks," in *Proceedings of the 13nd ACM SIGCOMM conference on Internet measurement*, IMC '13, 2013.

[74] Han, D., Agarwala, A., Andersen, D. G., Kaminsky, M., Papagiannaki, K., and Seshan, S., "Mark-and-Sweep: Getting the "inside" scoop on neighborhood networks," in *Proc. Internet Measurement Conference*, (Vouliagmeni, Greece), Oct. 2008.

[75] Hu, N., Li, L., Mao, Z., Steenkiste, P., and Wang, J., "A measurement study of internet bottlenecks," in *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, vol. 3, pp. 1689 – 1700 vol. 3, march 2005.

[76] Hu, N., Li, L. E., and Mao, Z. M., "Locating Internet bottlenecks: Algorithms, measurements, and implications," in *Proc. ACM SIGCOMM*, (Portland, OR), pp. 41–54, Aug. 2004.

[77] Ihm, S. and Pai, V., "Towards understanding modern web traffic," in *Proc. Internet Measurement Conference*, (Berlin, Germany), Nov. 2010.

[78] Internet Engineering Task Force, *IP Network Address Translator (NAT) Terminology and Considerations*, Aug. 1999. RFC 2663.

[79] "Internet World Stats." `http://www.internetworldstats.com/dsl.htm`.

[80] "Internet Usage for all the Americas." `http://www.internetworldstats.com/stat2.htm`.

[81] "Internet Usage in Asia." `http://www.internetworldstats.com/stat3.htm`.

[82] ITU, "Ict facts and figures," jan 2012. `http://www.itu.int/ITU-D/ict/facts/2011/material/ICTFactsFigures2011.pdf`.

[83] "Asymmetric Digital Subscriber Line Transceivers." ITU-T G.992.1, 1999.

[84] "Asymmetric Digital Subscriber Line (ADSL) Transceivers - Extended Bandwidth ADSL2 (ADSL2Plus)." ITU-T G.992.5, 2003.

[85] "Data-over-cable service interface specifications: Radio-frequency interface specification." ITU-T J.112, 2004.

[86] Jain, M. and Dovrolis, C., "Pathload: A measurement tool for end-to-end available bandwidth," in *In Proceedings of Passive and Active Measurements (PAM) Workshop*, pp. 14–25, 2002.

[87] Jamieson, K. and Balakrishnan, H., "PPR: partial packet recovery for wireless networks," in *Proc. ACM SIGCOMM*, (Kyoto, Japan), Aug. 2007.

[88] jan Su, A., Choffnes, D. R., Kuzmanovic, A., and Bustamante, F. E., "Drafting behind akamai (travelocity-based detouring," in *Proc. ACM SIGCOMM*, (Pisa, Italy), Aug. 2006.

[89] Jr., C. R. S. and Riley, G. F., "Neti@home: A distributed approach to collecting end-to-end network performance measurements," in *Passive & Active Measurement (PAM)*, (Antibes Juan-les-Pins, France), Apr. 2004.

[90] Judd, G. and Steenkiste, P., "Understanding Link-level 802.11 Behavior: Replacing Convention with Measurement," in *Wireless Internet Conference 2007 (Wicon07)*, (Austin, TX), Oct. 2007.

[91] Jung, J., Berger, A. W., and Balakrishnan, H., "Modeling TTL-based Internet Caches," in *IEEE Infocom 2003*, (San Francisco, CA), April 2003.

[92] Jung, J., Sit, E., Balakrishnan, H., and Morris, R., "DNS Performance and the Effectiveness of Caching," in *Proc. ACM SIGCOMM Internet Measurement Workshop*, (San Fransisco, CA), Nov. 2001.

[93] Kanuparthy, P. and Dovrolis, C., "Diffprobe: detecting isp service discrimination," in *Proceedings of the 29th conference on Information communications*, INFOCOM'10, (Piscataway, NJ, USA), pp. 1649–1657, IEEE Press, 2010.

[94] Kanuparthy, P. and Dovrolis, C., "Shaperprobe: End-to-end detection of isp traffic shaping using active methods," in *Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference*, IMC '11, pp. 473–482, ACM, 2011.

[95] Kanuparthy, P., Dovrolis, C., and Ammar, M., "Spectral probing, crosstalk and frequency multiplexing in internet paths," in *Proceedings of the 8th ACM SIGCOMM conference on Internet measurement*, IMC '08, (New York, NY, USA), pp. 291–304, ACM, 2008.

[96] Kanuparthy, P., Dovrolis, C., Papagiannaki, K., Seshan, S., and Steenkiste, P., "Can user-level probing detect and diagnose common home-wlan pathologies," *SIGCOMM Comput. Commun. Rev.*, vol. 42, pp. 7–15, Jan. 2012.

[97] Katabi, D. and Blake, C., "Inferring congestion sharing and path characteristics from packet interarrival times," Tech. Rep. MIT-LCS-TR-828, Massachusetts Institute of Technology, 2002.

[98] "Keynote." `http://www.keynote.com/products/web_performance/web-performance-testing.html`.

[99] Kreibich, C., Weaver, N., Nechaev, B., and Paxson, V., "Netalyzr: Illuminating the edge network," in *Proc. Internet Measurement Conference*, (Melbourne, Australia), Nov. 2010.

[100] Kreibich, C., Weaver, N., Nechaev, B., and Paxson, V., "Netalyzr: illuminating the edge network," in *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*, pp. 246–259, ACM, 2010.

[101] Krishnamurthy, B. and Wills, C., "Analyzing factors that influence end-to-end Web performance," in *Proc. Twelfth International World Wide Web Conference*, (Amsterdam, The Netherlands), May 2000.

[102] Krishnan, R., Madhyastha, H. V., Jain, S., Srinivasan, S., Krishnamurthy, A., Anderson, T., and Gao, J., "Moving beyond end-to-end path information to optimize CDN performance," in *Proc. Internet Measurement Conference*, 2009.

[103] Lai, K. and Baker, M., "Nettimer: A tool for measuring bottleneck link bandwidth," in *Proceedings of the USENIX Symposium on Internet Technologies and Systems*, vol. 134, 2001.

[104] Lakshminarayanan, K. and Padmanabhan, V. N., "Some findings on the network performance of broadband hosts," in *Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement*, IMC '03, (New York, NY, USA), pp. 45–50, ACM, 2003.

[105] Lakshminarayanan, K., Sapra, S., Seshan, S., and Steenkiste, P., "Rfdump: an architecture for monitoring the wireless ether," in *Proceedings of the 5th international conference on Emerging networking experiments and technologies*, CoNEXT '09, pp. 253–264, 2009.

[106] Li, Z., Zhang, M., Zhu, Z., Chen, Y., Greenberg, A., and Wang, Y.-M., "Webprophet: Automating performance prediction for web services," in *Proc. 7th USENIX NSDI*, (San Jose, CA), Apr. 2010.

[107] "Link Prefetching FAQ." `https://developer.mozilla.org/En/Link_prefetching_FAQ`.

[108] Lohr, S., "For Impatient Web Users, an Eye Blink Is Just Too Long to Wait." `http://www.nytimes.com/2012/03/01/technology/impatient-web-users-flee-slow-loading-sites.html`, Mar. 2012.

[109] Mahajan, R., Rodrig, M., Wetherall, D., and Zahorjan, J., "Analyzing the mac-level behavior of wireless networks in the wild," in *Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications*, SIGCOMM '06, pp. 75–86, 2006.

[110] Maier, G., Feldmann, A., Paxson, V., and Allman, M., "On dominant characteristics of residential broadband internet traffic," in *Proc. Internet Measurement Conference*, (Chicago, Illinois), Oct. 2009.

[111] Mathis, M., Heffner, J., and Reddy, R., "Network Path and Application Diagnosis." `http://www.psc.edu/networking/projects/pathdiag/`.

[112] "Minstrel rate adaptation algorithm." `http://goo.gl/5xPSC`.

[113] "Measurement Lab." `http://measurementlab.net`, Jan. 2009.

[114] Mogul, J. C., Chan, Y. M., and Kelly, T., "Design, implementation, and evaluation of duplicate transfer detection in HTTP," in *Proc. First Symposium on Networked Systems Design and Implementation (NSDI)*, (San Francisco, CA), Mar. 2004.

[115] Morton, A. and Claise, B., *Packet Delay Variation Applicability Statement*. Internet Engineering Task Force, Mar. 2009. RFC 5481.

[116] "Netalyzr." `http://netalyzr.icsi.berkeley.edu/`.

[117] "Netflix performance on top isp networks." http://techblog.netflix.com/2011/01/netflix-performance-on-top-isp-networks.html, Jan. 2011.

[118] "Network Dashboard." http://networkdashboard.org/.

[119] Niculescu, D., "Interference map for 802.11 networks," in *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, IMC '07, (New York, NY, USA), pp. 339–350, ACM, 2007.

[120] Nielsen, H., Gettys, J., Baird-Smith, A., Prud'hommeaux, E., Lie, H. W., and Lilley, C., "Network performance effects of http/1.1, css1, and png," in *Proc. ACM SIGCOMM*, (Cannes, France), Sept. 1997.

[121] Norton, B., "Peering in africa," aug 2012. http://drpeering.net/AskDrPeering/blog/articles/Ask_DrPeering/Entries/2012/8/29_Peering_in_Africa.html.

[122] OECD, *OECD Communications Outlook*. OECD Publishing, July 2011.

[123] "Speedtest.net by Ookla - The Global Broadband Speed Test." http://www.speedtest.net/.

[124] "OpenWrt." https://openwrt.org, Sept. 2013.

[125] "OpenWrt supported hardware." http://wiki.openwrt.org/TableOfHardware, Sept. 2013.

[126] Padhye, J., Firoiu, V., Towsley, D., and Kurose, J., "Modeling TCP Throughput: A Simple Model and its Empirical Validation," in *Proc. ACM SIGCOMM*, (Vancouver, British Columbia, Canada), pp. 303–323, Sept. 1998.

[127] Padmanabhan, V. and Mogul, J., "Using predictive prefetching to improve world wide web latency," *ACM SIGCOMM Computer Communication Review*, vol. 26, no. 3, pp. 22–36, 1996.

[128] "Public access wifi service." http://publicaccesswifi.org/. Retrieved: September 2013.

[129] Peterson, L., Anderson, T., Culler, D., and Roscoe, T., "A blueprint for introducing disruptive technology into the Internet," in *Proc. 1st ACM Workshop on Hot Topics in Networks (Hotnets-I)*, (Princeton, NJ), Oct. 2002.

[130] Peterson, L., Bavier, A., Fiuczynski, M. E., and Muir, S., "Experiences building PlanetLab," in *Proceedings of the 7th symposium on Operating systems design and implementation*, pp. 351–366, USENIX Association, 2006.

[131] "Phantomjs." http://phantomjs.org/.

[132] "Phantomjs Users." https://github.com/ariya/phantomjs/wiki/Users.

[133] "OpenWRT Module for Popularity-based Prefetching." http://data.gtnoise.net/bismark/imc2013/webperf/popularity_prefetch.tgz.

[134] "Project BISmark: Open development portal." http://projectbismark.github.io.

[135] "QUIC: Quick udp internet connections." http://goo.gl/O2r6rM.

[136] "Radiotap." http://radiotap.org.

[137] Rayanchu, S., Mishra, A., Agrawal, D., Saha, S., and Banerjee, S., "Diagnosing wireless packet losses in 802.11: Separating collision from weak signal," in *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, pp. 735 –743, april 2008.

[138] Rayanchu, S., Patro, A., and Banerjee, S., "Airshark: detecting non-wifi rf devices using commodity wifi hardware," in *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, IMC '11, (New York, NY, USA), pp. 137–154, ACM, 2011.

[139] Rayanchu, S., Patro, A., and Banerjee, S., "Catching whales and minnows using wifinet: deconstructing non-wifi interference using wifi hardware," in *Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation*, NSDI'12, (Berkeley, CA, USA), pp. 5–5, USENIX Association, 2012.

[140] Rekhter, Y., Moskowitz, B., Karrenberg, D., Groot, G. J. d., and Lear, E., *Address Allocation for Private Internets*. United States, 1996.

[141] Ribeiro, V., Riedi, R., Baraniuk, R., Navratil, J., and Cottrell, L., "pathchirp: Efficient available bandwidth estimation for network paths," in *Passive and active measurement workshop*, vol. 4, 2003.

[142] "RIPE Atlas." https://atlas.ripe.net.

[143] Roy, S. and Feamster, N., "Characterizing correlated latency anomalies in broadband access networks," in *Proceedings of ACM SIGCOMM*, pp. 525–526, ACM, 2013.

[144] "Samknows." http://samknows.com/. Retrieved: September 2013.

[145] Sánchez, M. A., Otto, J. S., Bischof, Z. S., Choffnes, D. R., Bustamante, F. E., Krishnamurthy, B., and Willinger, W., "Dasu: Pushing experiments to the internets edge," in *Proc. of USENIX NSDI*, 2013.

[146] Saroiu, S., Gummadi, P., and Gribble, S., "Sprobe: A fast technique for measuring bottleneck bandwidth in uncooperative environments," in *IEEE INFOCOM*, p. 1, 2002.

[147] Savage, S., "Sting: a tcp-based network measurement tool," in *Proceedings of the 1999 USENIX Symposium on Internet Technologies and Systems*, pp. 71–79, 1999.

[148] "Shaperprobe." http://www.cc.gatech.edu/~partha/diffprobe/shaperprobe.html.

[149] Siekkinen, M., Collange, D., Urvoy-Keller, G., and Biersack, E., "Performance limitations of ADSL users: A case study," in *Passive and Active Measurement Conference (PAM)*, 2007.

[150] Singhal, S. and Paoli, J., "Speed and Mobility: An Approach for HTTP 2.0 to Make Mobile Apps and the Web Faster," Mar. 2012. http://goo.gl/1uWCl.

[151] "April 2012 FCC/SamKnows data." http://www.fcc.gov/measuring-broadband-america/2012/validated-data-april-2012.

[152] Souders, S., "Velocity and the bottom line." http://radar.oreilly.com/2009/07/velocity-making-your-site-fast.html, July 2009.

[153] "SPDY: An experimental protocol for a faster web." http://www.chromium.org/spdy/spdy-whitepaper.

[154] Spring, N. T., Wetherall, D., and Anderson, T., "Scriptroute: A public internet measurement facility," in *Proc. 4th USENIX Symposium on Internet Technologies and Systems (USITS)*, (Seattle, Washington), Mar. 2003.

[155] Sundaresan, S., de Donato, W., Feamster, N., Teixeira, R., Crawford, S., and Pescapè, A., "Broadband internet performance: A view from the gateway," in *Proc. ACM SIGCOMM*, (Toronto, Ontario, Canada), Aug. 2011.

[156] Sundaresan, S., de Donato, W., Feamster, N., Teixeira, R., Crawford, S., and Pescapè, A., "Helping users shop for isps with internet nutrition labels," in *ACM SIGCOMM Workshop on Home Networking (Homenets)*, (Toronto, Ontario, Canada), May 2011.

[157] Sundaresan, S., Burnett, S., Feamster, N., and de Donato, W., "Bismark: A testbed for deploying measurements and applications in broadband access networks," in *2014 USENIX Annual Technical Conference (USENIX ATC 14)*, (Philadelphia, PA), pp. 383–394, USENIX Association, June 2014.

[158] Sundaresan, S., de Donato, W., Feamster, N., Teixeira, R., Crawford, S., and Pescapè, A., "Measuring home broadband performance," *Commun. ACM*, vol. 55, pp. 100–109, Nov. 2012.

[159] Sundaresan, S., Feamster, N., Teixeira, R., and Magharei, N., "Measuring and mitigating web performance bottlenecks in broadband access networks," in *Proceedings of the 13nd ACM SIGCOMM conference on Internet measurement*, IMC '13, 2013.

[160] Sundaresan, S., Magharei, N., Feamster, N., and Teixeira, R., "Accelerating last-mile web performance with popularity-based prefetching," vol. 42, pp. 303–304, ACM, Aug. 2012.

[161] Sundaresan, S., Magharei, N., Feamster, N., Teixeira, R., and Crawford, S., "Web performance bottlenecks in broadband access networks," vol. 41, pp. 383–384, ACM, June 2013.

[162] Tariq, M., Motiwala, M., Feamster, N., and Ammar, M., "Detecting network neutrality violations with causal inference," in *Proceedings of the 5th international conference on Emerging networking experiments and technologies*, pp. 289–300, ACM, 2009.

[163] Vorhaus, D., "A New Way to Measure Broadband in America." http://blog.broadband.gov/?entryId=359987, Apr. 2010.

[164] Wang, X. S., Balasubramanian, A., Krishnamurthy, A., and Wetherall, D., "Demystifying page load performance with wprof," Apr. 2013.

[165] Wearden, G., "Ofcom: Broadband isps are pulling a fast one." http://www.theguardian.com/business/2010/jul/27/telecoms-btgroup. Retrieved: July 2010.

[166] "Web attack knows where you live." http://www.bbc.co.uk/news/technology-10850875, Aug. 2010.

[167] "Web page test." http://webpagetest.org/.

[168] Wolman, A., Voelker, G. M., Sharma, N., Cardwell, N., Karlin, A., and Levy, H. M., "On the scale and performance of cooperative web proxy caching," in *Proc. 17th ACM Symposium on Operating Systems Principles (SOSP)*, (Kiawah Island, SC), Dec. 1999.

[169] Zhang, Y., Breslau, L., Paxson, V., and Shenker, S., "On the characteristics and origins of internet flow rates," in *Proc. ACM SIGCOMM*, (Pittsburgh, PA), Aug. 2002.

[170] Zhou, W., Li, Q., Caesar, M., and Godfrey, P., "Asap: A low-latency transport layer," in *Proceedings of the Seventh COnference on emerging Networking EXperiments and Technologies*, p. 20, ACM, 2011.