Up, Down and Around the Stack:
ISP Characterization from Network Intensive Applications

Zachary S. Bischof
Northwestern University
zbischof@eecs.northwestern.edu

John S. Otto
Northwestern University
jotto@eecs.northwestern.edu

Fabián E. Bustamante
Northwestern University
fabianb@eecs.northwestern.edu

ABSTRACT

Broadband characterization has recently attracted much attention from the research community and the general public. Given this interest and the important business and policy implications of residential Internet service characterization, recent years have brought a variety of approaches to profiling Internet services, ranging from Web-based platforms to dedicated infrastructure inside home networks. We have previously argued that network-intensive applications provide an almost ideal vantage point for broadband service characterization at sufficient scale, nearly continuously and from end users. While we have shown that the approach is indeed effective at characterization and can enable performance comparisons between service providers and geographic regions, a key unanswered question is how well the performance characteristics captured by these network-intensive applications can predict the overall user experience with other applications.

In this paper, using BitTorrent as an example network-intensive application, we present initial results that demonstrate how to obtain estimates of bandwidth and latency of a network connection by leveraging passive monitoring and limited active measurements from network intensive applications. We then analyze user experienced web performance under a variety of network conditions and show how estimated metrics from this network intensive application can serve as good web performance predictors.

Categories and Subject Descriptors

C.2.2 [**Communication Systems Organization**]: Computer Communication Networks—*Network Protocols*; C.2.5 [**Communication Networks**]: Local and Wide-Area Networks—
Internet; C.4 [**Performance of Systems**]: Performance Attributes

General Terms

Experimentation, Performance, Measurement

Keywords

Broadband access networks, ISP characterization, Web performance

W-MUST'12, August 17, 2012, Helsinki, Finland.
Copyright 2012 ACM 978-1-4503-1476-3/12/08 ...$15.00.

1. INTRODUCTION

Characterization of broadband Internet services has recently attracted much attention from consumers, researchers and policymakers [1, 2, 5–7, 10–14]. Broadband subscription has seen unprecedented growth over the last few years, with one quarter of the Internet population being residential broadband subscribers. Our ability to measure these networks, however, has not kept pace [3].

In previous work [2] we have argued that network-intensive applications can provide a nearly ideal vantage point for broadband characterization. The low equipment and logistical costs of a software-based approach facilitate large scale deployments that can capture the diversity of available Internet services. Running inside an application also enables continuous monitoring, necessary to witness dynamic changes due to network management policies (e.g. traffic shaping or oversubscribed networks) and unscheduled events (e.g. service interruptions). Last, since these vantage points are provided by end users, it is easier to ensure an objective service characterization.

While we have shown that this approach is indeed effective at broadband service characterization and may enable service comparison *across ISPs*, a key unanswered question is how well the performance captured by these network-intensive applications can predict user perceived performance in *other* applications (e.g. Web or VoIP). In this work, we present preliminary work showing that these network-intensive applications can be leveraged to accurately identify several important metrics of the user's broadband connection and estimate a user's expected web performance.

Using BitTorrent as an example host application, we present preliminary results that demonstrate how to obtain estimates of the bandwidth and latency of a network connection by leveraging passive monitoring and selected active measurements from network intensive applications (§ 3). Our results show that it is possible to infer these metrics with high accuracy (§ 4). In the case of latency, our measurement approach using traceroute probes yields results comparable to values previously reported [12, 14] for several major ISPs in the United States. We show that 3 hours of BitTorrent usage (aggregated across sessions) is sufficient to accurately determine the user's bandwidth. In this case, we find that the maximum observed BitTorrent download rate is strongly correlated ($r > 0.75$) with the maximum download speed reported by the Network Diagnostic Tool (NDT) [9].

We show that the proposed approach is able to measure settings in a variety of simulated network environments, varying both last-mile latency and bandwidth capacity, and that the derived values can be used to infer page rendering times (§5). We show that improving both latency and bandwidth can improve overall web performance – but only to a certain point, past which bandwidth

Application	Bandwidth	Latency
File transfers	Elastic	Not sensitive
HD video streaming	Need >5 Mbps	Need ~<100 ms
Web browsing	Sensitive	Sensitive

Table 1: Summary of the importance of connection metrics for several popular applications.

and latency improvements do not result in significantly faster web page rendering times. We close with a brief discussion of additional issues (§ 6).

2. BACKGROUND

Broadband characterization is attractive to users, policymakers and researchers alike; it allows users to evaluate the quality of service they received, it can inform policymakers of coverage and available services levels to guide recommendations and legislation, and it provides researchers with an interesting and challenging problem domain. Not surprisingly, a number of platforms and approaches to broadband characterization have been proposed [1,5, 7,10,14] and several reports have been made public (e.g. [11,12]).

Many of these approaches rely heavily on active measurement for characterization. Active measurement allows for high degree of control over when, how, and in what context a measurement experiment is launched. This approach, however, can result in significant measurement overhead. The increasing adoption of tiered charging model by ISPs, e.g. with quotas on monthly volume transferred, make this overhead increasingly important.

Consider Comcast[1] or AT&T U-verse,[2] for instance, both imposing a 250 GB monthly cap. Based on the traffic volume estimates for the SamKnows deployment,[3] AT&T U-verse users on the fastest plan (24 Mbps downstream) would consume 146 GB of data each month – 58% of that user's quota – to measure the user's connection. For subscribers of Comcast's fastest (105 Mbps) plan, SamKnows measurements would use 625 GB – *2.5x* more than the user's monthly quota.[4]

We have previously argued [2] that network-intensive applications provide an alternative, low-overhead approach for broadband characterization at scale, continuously and from end-users. In this work, we present preliminary work showing that these applications can be leveraged to accurately estimate several important metrics of the user's broadband connection and estimate a user's expected web performance. There is a common, well-understood set of low-level metrics that can characterize the service received by end users. In this work, we focus on two of them, latency and bandwidth, given the impact they have on the performance of all network applications (Tab. 1).

3. METHODOLOGY

Our analysis is based on data contributed by a subset of users of a BitTorrent plugin as well as additional BitTorrent and web-performance traces collected in a controlled setting. The following paragraphs describe both in detail.

3.1 Wide-area BitTorrent traces

A large part of our data set is comprised of traces of BitTorrent activity collected by our previously released extensions for the Vuze BitTorrent client [15]. We use this data to evaluate the feasibility of using a network-intensive application to estimate metrics of a user's Internet connection across a wide range of scenarios.

This data set comprises a combination of passive and limited active measurements. With the user's permission, each participating client reports anonymized traces that include snapshots of total BitTorrent bandwidth use at 30-second intervals and the results of active measurements such as pings, traceroutes, and Network Diagnostic Tool (NDT) probes [9].

The traces used were collected from a subset of users during February and March 2012. Specifically, the traces we analyze for this work come from 6188 unique installations reporting from 134 countries and 1737 ASes. Users of these extensions are predominantly located in residential broadband networks. All major access technologies are represented, including DSL, cable, fiber, satellite and 3G/4G wireless, with a wide range of access latency (up to 560 ms round-trip-time) and bandwidth (512 Kbps to >100 Mbps) [8].

3.2 Controlled experiments

We also conducted experiments in a controlled setting to evaluate the accuracy of our connection metric estimates and to capture page-rendering time as a measure of user-perceived performance.

Starting with a wired Ethernet connection to a well-provisioned university network (100 Mbps capacity, <2 ms latency to `google.com`), we used the `ipfw` and `dummynet` traffic shaper tools to emulate access links with varying characteristics: link latency and bandwidth. For each condition tested, we first collected a BitTorrent session trace while downloading a well-seeded torrent (so that the aggregate upload capacity of peers in the swarm would be able to saturate our downstream connection). This part of the experiment uses the same methodology as our wide-area BitTorrent traces, enabling us to determine the accuracy of the metrics we infer from users in the wild.

For each condition, we also evaluated the user-perceived web performance in Firefox v.11 over HTTP[5] for the 20 most popular websites in the US.[6] Specifically, we instrumented the browser using Selenium[7] and Firebug[8] to measure page rendering time as a proxy for user-perceived performance. These traces include HTTP archives (HAR) which can also measure time to first byte and page loading time.

Every time before fetching a page, we cleared the DNS and web object caches in the host and browser. To prevent the browser from using data in the local content cache, we created a new user profile in the browser for each page fetch. To ensure that the local DNS caches were cleared, we flushed the host's DNS cache (`dscacheutil -flushcache`). We loaded each page twice in close succession; the first request primes the *remote* caches of the DNS resolver, web caches, and content delivery replica servers. We use the timing statistics from the second request in our analysis. In general, we find no significant difference between the first and second page render times. We attribute this to the fact that we are querying popular websites and all DNS names and content are likely present in the DNS and web caches at the time of the

[1] http://xfinity.comcast.net/terms/network/amendment/

[2] http://www.att.com/esupport/internet/usage.jsp

[3] http://transition.fcc.gov/cgb/measuringbroadbandreport/technical_appendix/Technical_Appendix_Full.pdf

[4] These values were estimated using the bandwidth usage values listed in the SamKnows technical report (footnote 3).

[5] The SPDY protocol is disabled by default in this browser version.

[6] Ranked by `alexa.com`

[7] http://seleniumhq.org/

[8] http://getfirebug.com/

first request. All measurements were performed on a quad core 2.66 GHz Intel Core i7 machine with 8 GB of memory running Mac OS X v.10.7.

4. ESTIMATING METRICS

In this section, we build on the previously described methodology to evaluate the extent to which network-intensive applications and limited active measurements can be leveraged to estimate characteristics about a user's Internet connection. Using BitTorrent as an example application, we analyze the distributions of latency and download bandwidth capacity, and evaluate our accuracy in estimating these values.

4.1 Latency

To evaluate the contribution of several components of total latency across a user's connection, we use latency measurements from traceroutes to remote destinations. While these traceroutes are destined for other peers in the BitTorrent network, we focus on the first few hops – these correspond to the last-meter and last-mile latencies of the user's home network and the ISP's access network, respectively. Prior work has shown that last-mile latency is often a large fraction of total latency [4], and varies significantly across access technologies.

We use the latency to the first hop responding from a public IP address to estimate a user's last-mile latency. In some cases, we find that the first hop with a public IP address is actually the user's own public IP due to a middlebox or modem that is responding from this interface. In these cases, we treat it as a response from the private network and use the next public IP address to estimate last-mile latency.

Additionally, we identify the latency contribution of the private network by measuring the latency to the last responding private IP address. Measuring the latency added by the private network allows us to distinguish between the latency added by the last-mile (first public IP) versus the home network. However, in cases where the ISP itself is providing NATed IP addresses, we are currently unable to disambiguate this from latency added by a user's private home network.

In cases where a user does have a private network, we attempt to identify the type of connection between their host machine and the middlebox. For example, on systems running Mac OS X, we use the system profiler (`system_profiler SPNetworkDataType`) to match the current active interface with the device's profile. If the device's hardware and type fields are listed as AirPort, we classify the connection as wireless, and if they are listed as Ethernet, we classify the device as wired. In all cases, we record the name of the interface, as well, to verify names and to check cases that we were unable to classify automatically.

Figure 1 shows the distribution of median latency across the first public hop for each client that ran traceroute more than 100 times. To calculate last-mile latency, we subtract the latency to the last private IP from the latency to the first public IP. For this section, we exclude users that report a high latency in their private network (greater than 10% of the latency to the first public hop or greater than 5 ms), since the latency of the private network may limit the accuracy of our measurements across the last-mile. Overall, we find that the majority of users fall between 10 and 100 ms. For users recording very high latency (>1000 ms), many were subscribed to wireless services such as Clearwire or tethering with 3G/4G services – wireless technologies that typically have higher latency than wired connections.

To validate our latency measurements, we focus on a single ISP and compare against the values in the SamKnows dataset reported by Sundaresan et al. [14]. In AT&T's network, they report an average latency of 25.23 ms with a standard deviation of 33.47 ms. Fig. 2 plots the distribution of the minimum and median latency for each user in AT&T's network (AS7132) as measured by our extension. Considering the distribution of each user's *minimum* last-mile latency (the "Min" curve), the median user's latency is 19 ms – within 25% of the SamKnows reported value of 25.23 ms. For 95% of our users in AT&T's network, their minimum last-mile latency is within one standard deviation of the SamKnows reported value. We also plot the distribution of users' *median* last-mile latency to capture the *typical* case. As one would expect these latencies are higher – by 53% in the median case. Still, 77% of users fall within one standard deviation of the SamKnows reported mean. Since our latency distribution matches closely that reported by SamKnows for users in AT&T's network, we conclude that this approach for measuring last-mile latency is accurate.

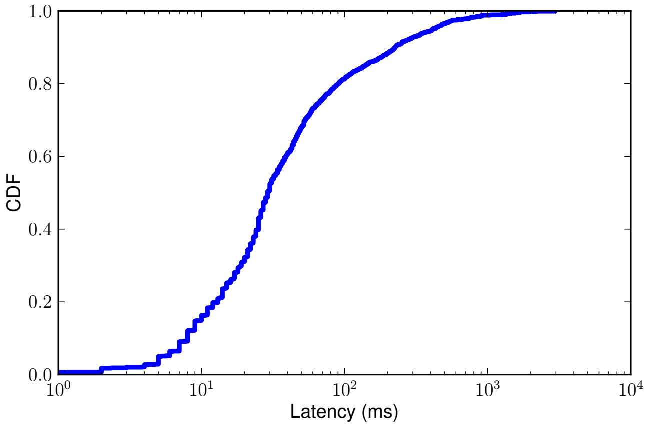

Figure 1: CDF of the last-mile latency per user as measured from our extension. For the majority of users (about 65%), this falls between 10 and 100 ms.

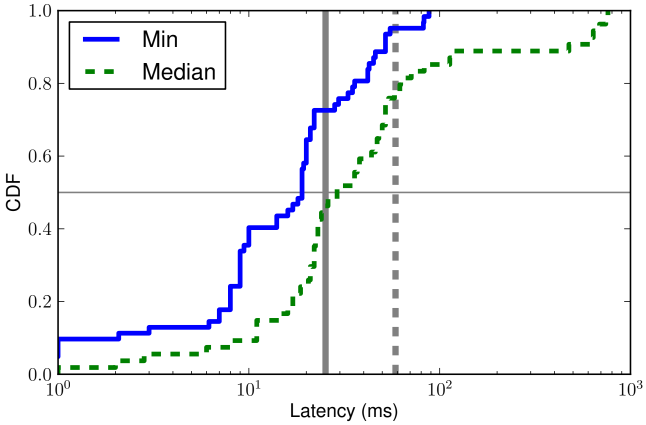

Figure 2: CDF of the last-mile latency measured from clients in AT&T's network. The curve labeled "Min" represents the minimum latency recorded by each user, while the curve labeled "Median" corresponds to each user's median recorded latency. The vertical solid line represents the average latency AT&T last-mile latency reported by SamKnows [14] (25.23 ms); the dashed vertical line shows the mean plus one standard deviation (58.7 ms).

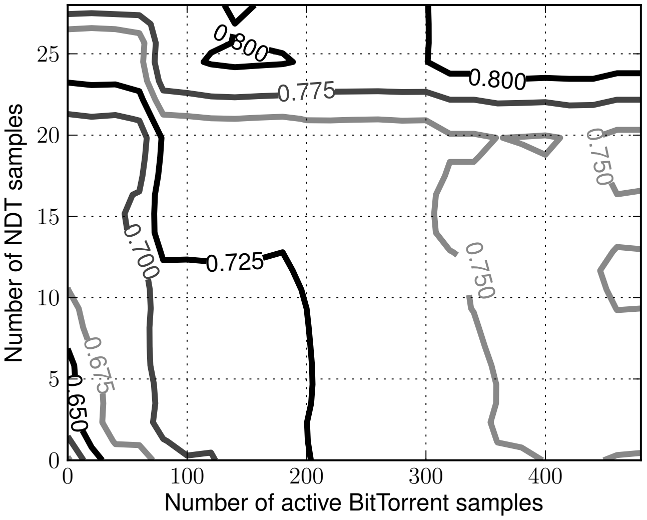

Figure 3: Correlation between all users' maximum BitTorrent download throughput and download throughput measured by NDT given at least a number of samples. The x-axis represents the minimum number of BitTorrent samples and the y-axis represents the minimum number of NDT samples needed to be included in the calculation of the correlation coefficient.

4.2 Bandwidth

To extract a connection's download throughput rate by passively observing BitTorrent activity, we use snapshots provided by our extension to monitor the application's maximum achieved rate while downloading content. Due to the fact that some users may have caps that are lower than their connection's full capacity, we only include snapshots when users have not imposed an application-level bandwidth limit, since such caps could prevent BitTorrent from utilizing the full capacity of the user's connection.

One issue with passive monitoring to infer maximum download rate, however, is that BitTorrent does not always saturate the user's connection (e.g. when the user is not downloading any files). Logically, the longer we are able to passively monitor BitTorrent activity, the more likely we are to capture a saturated connection. In order to determine how long a user needs to run BitTorrent to observe this, we correlate each user's maximum BitTorrent download rate with the maximum download rate reported by Network Diagnostic Tool (NDT).

Figure 3 shows a contour plot of the correlation coefficient between NDT and BitTorrent maximum achieved speeds across users with at least X BitTorrent samples and Y NDT measurements. For example, with users that have at least 325 samples from BitTorrent and at least 25 samples from NDT, we find a strong correlation between speeds reported by BitTorrent and NDT ($r > 0.80$). In general, the longer a user provides data by running BitTorrent, the more likely the application will saturate the link, and enable us to capture the connection's capacity.

To understand how BitTorrent and NDT throughput measurements compare, we analyze the distribution of the ratio between the metrics. We consider the subset of users for which we have sufficient data to infer their download capacity – at least 325 BitTorrent samples and 25 NDT measurements. Figure 4 shows a CDF of the ratios of the maximum throughput values measured by BitTorrent and NDT for each user. In cases where the ratio is significantly lower than 1, BitTorrent is unable to achieve speeds as high as NDT. This could be caused by a number of factors, such as a user downloading content from a poorly seeded torrent, resulting in download speeds being limited by the combined upload capacity of their peers. If a user subscribes to a service with PowerBoost – which temporarily gives users faster download speeds when starting to download a file – NDT may be able to achieve speeds higher than BitTorrent. This is because BitTorrent may not be able to leverage the faster downloading period because of the delay in establishing sufficient connections to other peers.

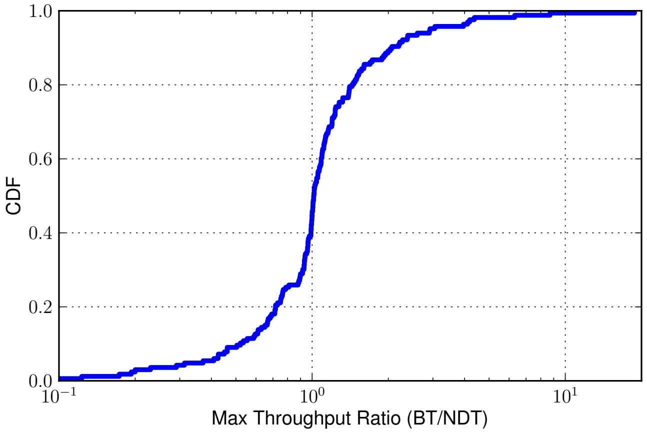

Figure 4: CDF of the ratio between users' maximum BitTorrent transfer rate and the maximum speed measured by NDT.

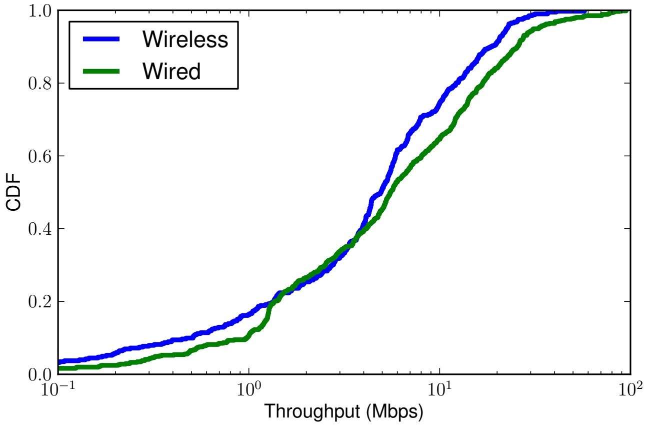

Figure 5: CDF of the download throughput rates achievable by clients using wired or wireless connections in their private network. Both distributions are generally similar, though wired clients are able to achieve rates above 55 Mbps.

In some cases, BitTorrent is able to achieve significantly faster speeds than NDT. Some possible causes include instances where an NDT measurement server is located far away from a user's machine. This would cause the download throughput to be dominated by the longer RTT, due to the fact that NDT uses a single TCP connection to measure throughput and the upper-bound on TCP's receive window size.

Finally, we evaluate the impact of using wireless in the home network on a user's download throughput relative to a wired home network connection. There were no cases in our dataset for which we could conduct a direct comparison (e.g. a user who used at different times a wired and wireless connection in a given network). Therefore, we compared the distributions of achieved download

Download Throughput		**Instrumented Last Mile Latency**	
Setting Mbps	**Measured** Mbps	**Setting** ms	**Measured** ms (std)
.512	0.495	**0**	1.3 (2.3)*
1	0.960	2	2.4 (2.2)
2	1.93	4	4.6 (3.5)
4	3.73	8	8.3 (3.1)
8	7.64	16	16 (1.4)
16	15.2	32	33 (4.3)
32	31.1	64	64 (2.9)
64	64.3	128	128 (3.0)
100	97.7	256	256 (3.8)
-	-	512	512 (2.3)
-	-	1024	1025 (3.9)

Table 2: For each metric, the setting used to simulate the last-mile metric ("Setting") as well as the value of each setting as measured by our extension ("Measured"). The measured latency value for an instrumented last mile value of 0 corresponds to the baseline latency to the first hop (*), which is subtracted from the subsequent RTT values. Bold values correspond to the default settings.

throughput rates between users on wired or wireless home network connections, shown in Fig. 5. In general, the wireless users have slightly slower speeds, differing by 33% at the 95th percentile; however there is only a 10% difference in the median case. For both wired and wireless, about 40% of users are able to achieve about 5 Mbps download speeds. In general, the similarities between the distributions indicate that using a wireless home network connection does not impose a significant bottleneck on maximum achievable download rate.

In summary, we are able to infer low-level metrics such as throughput and latency by leveraging passive traces of BitTorrent activity and running limited active measurements. In the next section, we show how we can relate these measurements to the performance of other applications.

5. CONTROLLED EXPERIMENTS

In this section, we first evaluate our extension's ability to measure the download throughput and last-mile latency in a controlled environment with emulated access link characteristics. Then, we study how modifying latency and throughput of the last-mile connection affects user-perceived web performance, in terms of page load & rendering time. Of the websites we tested, we present results for four (Twitter, YouTube, Craigslist, and Bing) that capture the spectrum of what we observed across all sites.

Using the setup described in Sec. 3, we measure our ability to accurately measure last-mile link characteristics configured with a traffic shaper. Table 2 shows the settings we used to simulate different aspects of each quality of a last-mile link. We tested the impact of each metric individually. For example, to test the impact of a download throughput limit of 1 Mbps, we downloaded via BitTorrent a Linux CD image for 10 minutes. With this setting, we measured a maximum achievable throughput of 0.960Mbps at the application level. Across all the throughput and latency settings we tested, the measured values via our BitTorrent-based approach closely followed the parameter settings. In the following subsections, we discuss the impact of each individual metric on page load times.

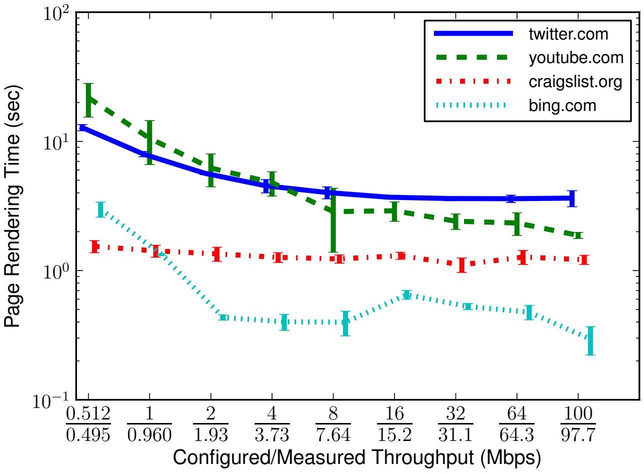

Figure 6: Page rendering time (log scale) for each website across various download throughput limits. Error bars show the standard deviation across 10 runs.

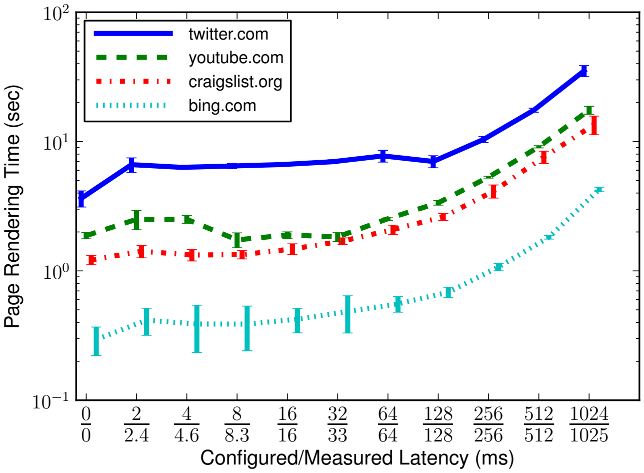

Figure 7: Page rendering time (log scale) for each website seen with various last-mile latencies. Error bars show std. dev. across 10 runs.

5.1 Bandwidth

We study the impact of last-mile download throughput limits on page load times, shown in Fig. 6. For pages such as Craigslist, we find that page load times are generally consistent across all tested bandwidth settings, which we attribute to its small page size.

For pages similar to YouTube, we find that increasing bandwidth throughput continues to improve page rendering times until reaching 8 Mbps. Websites such as Twitter and Bing see gains until approximately 4 Mbps and 2 Mbps, respectively. The majority of the web pages we tested were similar to these two sites. In general, we note that there is a trend of diminishing returns as download bandwidth increases.

5.2 Latency

Next, we looked at the impact of last-mile latency on web performance. Figure 7 shows the page rendering times as we increase the latency across our emulated last-mile link. For all websites, we find that excessively high last-mile latency dramatically increases page rendering time. However, as seen in Fig. 1, 50% of users have a last-mile latency of less than 30 ms. As with our download bandwidth experiment, decreasing last-mile latency past this point shows diminishing performance gains. This is because with lower latencies across the last-mile, RTT from the host machine to the web server is increasingly dominated by the in-network latency. Therefore, it appears that the performance for most users is not significantly limited by their last-mile latency.

For both latency and download throughput, we find that improving last-mile link characteristics does help web performance – to a point. This trend reveals that, depending on the scenario, the last-mile link may not be the web performance bottleneck. In future work, we plan to study how combinations of these metrics affect web performance.

6. DISCUSSION

We have shown that it is possible to infer application (e.g. web) performance using the view from a network-intensive application and limited active measurements. One advantage of leveraging network-intensive applications is the ability to avoid most onerous active measurements when characterizing a user's service. For instance, we are able to *passively* infer a user's download throughput without incurring the bandwidth "cost" of active download speed tests. While our approach does use limited active measurements, such as traceroutes and pings, their bandwidth demand is comparatively minor. We have also shown that passively-collected BitTorrent download snapshots are able to detect users' connection speeds, and examined the difference in download performance between wired and wireless configurations.

Leveraging network-intensive applications for broadband characterization is not without challenges. Perhaps among the most important challenges is the lack of control over the user's behavior (i.e. when we are able to run). We believe this to be a potentially solvable problem given a sufficiently large and rich dataset as that one could collect from most network-intensive applications.

7. ACKNOWLEDGEMENTS

We would like to thank our anonymous reviewers for their detailed and helpful feedback. We are always grateful to Paul Gardner for his assistance with Vuze and the users of our software for their invaluable data. This work is supported by the National Science Foundation through Awards CNS 0644062, CNS 0917233 and CNS 0855253, and by a generous Google Research Award.

8. REFERENCES

[1] G. Bernardi and M. K. Marina. Bsense: a system for enabling automated broadband census: short paper. In *Proc. of NSDR*, 2010.

[2] Z. S. Bischof, J. S. Otto, M. A. Sánchez, J. P. Rula, D. R. Choffnes, and F. E. Bustamante. Crowdsourcing ISP characterization to the network edge. In *Proc. of W-MUST*, 2011.

[3] M. Casado, T. Garfinkel, W. Cui, V. Paxson, and S. Savage. Opportunistic measurement: Extracting insight from spurious traffic. In *Proc. of HotNets*, 2005.

[4] Dabek, Cox, Kaashoek, and R. Morris. Vivaldi: A decentralized network coordinate system. In *Proc. of ACM SIGCOMM*, 2004.

[5] Grenouille. `http://grenouille.com`.

[6] D. Joumblatt, R. Teixeira, J. Chandrashekar, and N. Taft. Perspectives on tracing end-hosts: a survey summary. *SIGCOMM Comput. Commun. Rev.*, 40(2):51–55, Apr. 2010.

[7] C. Kreibich, N. Weaver, B. Nechaev, and V. Paxson. Netalyzr: Illuminating the edge network. In *Proc. of IMC*, 2010.

[8] J. F. Kurose and K. W. Ross. *Computer networking*. Addison-Wesley, 5th edition, 2010.

[9] MLabs. Network diagnostic tool. `http://www.measurementlab.net/run-ndt/`.

[10] SamKnows. Accurate broadband information for consumers, governments and ISPs. `http://www.samknows.com/`.

[11] SamKnows. Samknows & Ofcom UK broadband performance testing. `http://www.samknows.com/broadband/ofcom_and_samknows`, June 2009.

[12] SamKnows. Samknows & the FCC American broadband performance measurement. `http://www.samknows.com/broadband/fcc_and_samknows`, June 2009.

[13] C. R. Simpson, Jr., and G. F. Riley. Neti@home: A distributed approach to collecting end-to-end network performance measurements. In *Proc. of PAM*, 2004.

[14] S. Sundaresan, W. de Donato, N. Feamster, R. Teixeira, S. Crawford, and A. Pescapè. Broadband internet performance: a view from the gateway. In *Proc. of ACM SIGCOMM*, 2011.

[15] Vuze, Inc. Vuze. http://www.vuze.com.