Speed Measurements of Residential Internet Access

Oana Goga, Renata Teixeira

CNRS and UPMC Sorbonne Universités, Paris, France

Abstract. The spread of residential broadband Internet access is raising the question of how to measure Internet speed. We argue that available bandwidth is a key metric of access link speed. Unfortunately, the performance of available bandwidth estimation tools has rarely been tested from hosts connected to residential networks. This paper compares the accuracy and overhead of state-of-the-art available bandwidth estimation tools from hosts connected to commercial ADSL and cable networks. Our results show that, when using default settings, some tools underestimate the available bandwidth by more than 60%. We demonstrate using controlled testbeds that this happens because current home gateways have a limited packet forwarding rate.

1 Introduction

Broadband Internet Service Providers (ISPs) often advertise a maximum download and upload Internet speed, but home users have no reliable means to verify if the performance they get is even close to these maximum values. Government agencies and regulatory bodies are stepping in to help broadband users. For instance, the UK communications regulator (Ofcom) and the American Federal Communications Commission (FCC) are working with SamKnows to distribute routers for home users to test their broadband speed. Regulation bodies are actively seeking better standards to measure Internet speed (take as example the recent FCC challenge [1]). In this context, the Internet measurement community needs to answer the question: How to accurately measure Internet access speed?

The Internet measurement literature defines three metrics of network speed [20]. *Capacity* is the maximum rate at which a link can transmit packets. It is not a good metric for regulating access speed, because it does not capture network speed variations that would occur if, for instance, an ISP assigned too many subscribers to the same link. *TCP achievable throughput* is the maximum throughput obtained by a single TCP connection [12]. It captures the speed users get, but it cannot be used alone to regulate ISPs because the throughput of a single TCP depends on many factors that are exogenous to the access ISP (such as TCP advertised window or RTT). Finally, *available bandwidth* is the residual capacity of a link. Available bandwidth better reflects what ISPs can guarantee to home users and is therefore an essential metric for regulating ISPs. Available bandwidth, however, is not well defined because of the elasticity of TCP cross traffic.

Popular speed tests today such as Speedtest.net or SamKnows mainly estimate the available bandwidth by performing large parallel TCP transfers and post-processing the measurements to minimize the bias of TCP [4]. These tools flood the access link, so we refer to them as *flooding-based tools*. The large parallel TCP transfers will cause other TCP cross traffic to back off in the same way that the home users' traffic would. Thus, flooding-based tools measure the effective available bandwidth. The drawback of this technique is that flooding the link with probes introduces a large overhead. This overhead may not be a problem when a user occasionally runs a speed test, but it will disrupt some of the applications users may run during the test and it will consume network resources. Hence, this overhead may prevent users from signing up for initiatives that run periodic tests over a long period of time such as SamKnows (in particular in cases where home subscribers are limited by monthly bandwidth caps).

Over the last decade, the research community has developed numerous tools to estimate available bandwidth with reduced overhead and in the presence of diverse types of cross traffic [5, 11, 13, 22, 25]. We call these tools *optimized-probing tools*. These tools, however, have not been adopted in current speed tests. Only few of these tools were even tested in residential access networks. In particular, abwprobe [5] was designed and tested for ADSL networks [5], and pathload [13] and spruce [25] were tested in cable networks [16]. Comparative evaluations of available bandwidth estimation have been in core or academic networks and do not include flooding-based tools [3, 8, 19, 23, 25]. Thus, it is unclear how these tools compare in general when running in residential access networks, where bandwidth is asymmetric, traffic is often subject to a token bucket rate shaping [26], residential gateways have limited capacity [10] and very large buffers [15].

This paper studies the performance of available bandwidth estimation in residential access networks. We select tools that are known to work well in backbone and academic networks (§2). Then, we compare the accuracy and overhead of these tools in a setting where hosts are connected to commercial ADSL and cable providers for which we know the access link capacity and we can control cross traffic from the home (§3). This setting ensures that tools experience real access network behavior and yet we can estimate expected bounds of available bandwidth.

Findings. We find that optimized-probing tools that use small probes such as pathload systematically underestimate the bandwidth of residential networks (§4.1). We show using fully controlled testbeds that this underestimation happens because home gateways cannot sustain the high probing rate of these optimized-probing tools (§4.2). Further, we identify poor hardware and overhead of network-address translation as the main limiting factors. Our results show that pathload modified to send large probes, spruce and parallel TCP are the most accurate in estimating the available bandwidth of residential networks (§4.1). Finally, we show that the probing overhead of optimized-probing tools can be as much as 72 times smaller than that of flooding-based tools (§5).

2 Internet speed measurements

This section briefly describes the techniques to measure speed: first the flooding-based tools used in most popular speed tests, then the tools and techniques designed by the research community to measure available bandwidth.

2.1 Flooding-based tools

Flooding-based tools use multiple parallel TCP connections and compute the combined rate of all connections. Multiple TCP connections are preferred over a single connection because they are more robust to loss [2]. This method also gives more accurate estimates when hosts advertise a small TCP receiver window [4].

SamKnows uses three parallel connections and performs a warm-up transfer to avoid measuring during slow-start [26]. Speedtest.net executes a small download to estimate the connection speed. Based on the result, it selects the size of the transfer file and the number of parallel connections for the test (up to eight). It reports the average throughput achieved during the test after removing the lowest 30% and the highest 10% samples [4]. Netalyzr [15] is a more general tool, but as part of its tests it also reports the line speed. Netalyzr sends a single stream of UDP probes with rates increasing in a slow-start fashion to flood the link. A study of residential broadband capacity used a flood-based technique to infer the capacity of a large number of cable and DSL lines [7].

We use *iperf* to emulate flooding-based tools, because it allows us to set the number of parallel connections, the duration or the size of the transfer among other parameters.

2.2 Optimized-probing tools

Optimized-probing tools use two main techniques: the Probe Gap Model (PGM) and the Probe Rate Model (PRM) [25]. Here, we summarize the tools that are already known to perform well in backbone and academic networks [3, 8, 19, 25].

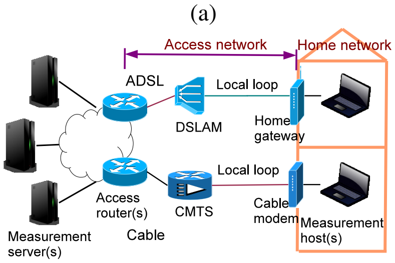
PGM tools send back-to-back probes and estimates the available bandwidth based on the dispersion observed at the receiver. They require an a priori knowledge of the capacity of the bottleneck link. We test *Spruce* [25] and *igi/ptr* [11] from these tools.

PRM tools send probes at different rates. If the probe rate is higher than the available bandwidth, then the probes are received at a lower rate (the packets are buffered at the bottleneck link). The available bandwidth is the maximum rate at which the sending rate matches the receiving rate. PRM tools are usually more accurate but have higher overhead than PGM tools, due to the iterative nature [9]. *Pathload* [13] uses one-way delays of consecutive packets to estimate the available bandwidth. The algorithm outputs the maximum rate at which consecutive one-way delays do not increase. Yaz [24] is a calibrated version of pathload, which minimizes the overhead. We choose to test pathload over Yaz, because pathload has been tested in many different settings and is hence an important comparison point. Moreover, pathload's code is more stable. *Pathchirp* [22] reduces probing overhead with respect to pathload by sending a train of packets with exponentially-reduced inter-packet gaps that probes a large range of rates.

Abwprobe [5] was designed especially for ADSL links. It is based on pathload, but it only requires access to the sender because it uses RTTs instead of one-way delays to estimate the available bandwidth. In our tests, however, abwprobe performs poorly because the high asymmetry between downlink and uplink capacities causes congestion in the uplink. When abwprobe detects that the uplink is congested it aborts the test. The authors discuss the fact that high asymmetry may be a problem and propose some heuristics to address this issue [5], but these heuristics are not implemented yet. Therefore, we do not present the results of abwprobe.

3 Measurement method

Our goal is to evaluate the tools discussed in the previous section for residential access networks, yet we need enough control to be able to interpret the results. Thus, we opt



(b)

	Free	Orange	AT&T	Comcast
City	Paris	Paris	Atlanta	Menlo Park
Advertised (Dw) **access speed (Up)**	28M ATM 1M ATM	18M IP 800k IP	3M IP 384KIP	20M IP 2M IP
Local loop **Dw. capacity**	18M IP	18M IP	3M IP	–
Baseline RTT	20ms	40ms	12ms	18ms
Gateway/ **Modem**	FreeBox v5	LiveBox v1.2 Thomson	NOX Box v0.5.0	ComcastBox Ubee U10C035

Fig. 1. Testbed setup and line characteristics.

for a semi-controlled testbed. Tests run over commercial ADSL and cable lines from different ISPs, where we know the line characteristics and we control both end-hosts as well as the traffic coming from the home. Hence, we can estimate expected bounds of available bandwidth. Our testbed includes two lines in France (ADSL from Free and Orange) and three lines in the United States (two cable lines from Comcast and one ADSL line from AT&T). Fig. 1(b) summarizes the properties of each line. The results for both Comcast lines (one in Atlanta and the other in Menlo Park) were equivalent, for space constraints we only present the properties and results for the Menlo Park line.

Setup. The testbed, illustrated in Fig. 1(a), can be divided in four parts. First, the *measurement servers*, a 2×Six-Core AMD Opteron (in Paris, UPMC), an Intel Core2 Quad (in Atalanta, Georgia Tech) and an Intel Pentium 4 (in Mountain View, UCSC) connected to the Internet through Gigabit Ethernet (GbE) links. Second, the network between the measurement server and the access router is out of our control, but we issue traceroutes every 5 minutes to verify that the RTT and number of hops of the end-to-end path remain stable. Third, the *access network* connects the home network to the access router. In ADSL networks, the link between the DSL modem or gateway and the DSL Access Multiplexer (DSLAM) is dedicated to a subscriber, whereas the link between the DSLAM and the access router is shared with other subscribers. In cable networks, subscribers connect via a shared medium (typically coaxial cable) to a Cable Modem Termination System (CMTS). In cable networks, the service plan rate limit is configured at the cable modem and is typically implemented using a token bucket rate shaper. The physical connection between the home and the DSLAM or the CMTS is called *local loop* or *last mile*. Fig. 1(b) also shows the local loop capacity we estimate for each tested ADSL line.[1] We know that the access links are the bottlenecks in our tests, because our tests show that the available bandwidth from the measurement server to the access router is always higher than the end-to-end available bandwidth. Finally, the *home network* is connected to the Internet through a home gateway or ADSL/cable modem. The *home gateway* combines the functionalities of an ADSL/cable modem and a router/access point. Fig. 1(b) presents the home gateway model for each line. All lines have a home gateway, except for Comcast, which has a cable modem. The *measurement host*, a ThinkPad T60, is directly connected to the home gateway via a GbE link. We connect the measurement host over Ethernet to emulate a setup similar to SamKnows, where bandwidth is measured directly from home gateways. Since our goal is to study

[1] We query the line characteristics from the gateway or ISP page (depending on the line) and obtain the estimated capacity using the Kitz site (`http://www.kitz.co.uk/adsl/max_speed_calc.php`).

access performance, placing measurements directly at the gateway avoids interference from wireless or cross traffic in the home [26].

Configuration of tools. To emulate the flooding-based tools we used iperf with 10 parallel connections and a transfer of 10 s which is enough in our case to even out the slow start. We run all the optimized-probing tools with default settings, unless stated otherwise. Interrupt coalescence (the network card waits to have multiple packets before interrupting the CPU) reduces the accuracy of optimized-probing tools because it alters the dispersion of packet pairs and the one-way delays of packet trains [14, 21]. §4.1 presents results with interrupt coalescence disabled.

Testing method. A *test* measures the available bandwidth with each different configuration of every tool back-to-back. During all tests we collect packet traces using tcpdump at the measurement server and the host. For each line we perform tests with all three servers. The results presented in §4.1 are for the closest server (in terms of RTT, see Fig. 1(b) for precise values) for each line. Results for the other servers are similar.

Cross traffic. Previous work has already studied how cross traffic impacts available bandwidth estimation [3,8,9,16,19,23,24]. Hence, we perform our tests during the night (from 11PM to 6AM) and with no cross traffic from the home network to minimize any congestion in the access link. In this scenario, results are easier to interpret, because the available bandwidth matches the capacity.

4 Accuracy

This section evaluates the accuracy of available bandwidth estimation in the residential access links described in the previous section. We first compare the tools using default settings. We find that tools that use small packets underestimate the available bandwidth. Then, we show that the home gateways are to blame because they cannot sustain the high packet rate that results from sending small probe packets.

4.1 Comparison

Tab. 1 presents the available bandwidth inferred by iperf with parallel TCP transfers (‘parallel TCP’), spruce, pathload, pathchirp, and igi/ptr for each tested access link. We run 15 tests with each line. The inferred bandwidth is stable across all tests of a given tool, so we summarize the results with the averages and the standard deviations (STDs). Pathload gives intervals as estimations rather than single values, so we present the averages of the minimum and of the maximum value. Pathchirp gives several instantaneous samples of bandwidth for each test. We present the average of the samples per test.

Benchmark. The first line of Tab. 1 helps us interpret the results in this section. It shows the UDP capacity obtained when flooding the link with iperf in UDP. The UDP capacity is computed using a `tcpdump` trace, as the total amount of bytes (including the IP and UDP headers) going through the link per time unit. The *UDP capacity* represents the maximum achievable IP rate for each line. These values are consistent with the capacity of ADSL local loops and advertised speeds for Comcast in Fig. 1(b). Comcast offers Powerboost, which allows users to download at higher rates at the beginning of the connection. The UDP capacity during the Powerboost period is 22.93 Mbps, whereas

Table 1. Accuracy of available bandwidth estimation. (*) 10s/6s(Powerboost)

Tool	Pckt.	Free		Orange		Comcast		AT&T	
	Size	Avg (Mbps)	STD	Avg (Mbps)	STD	Avg (Mbps)	STD	Avg (Mbps)	STD
UDP capacity	1440B	16.30	0.16	15.80	0.10	20.60/22.93*	0.04/0.06*	2.85	0.004
parallel TCP	1440B	15.41	0.15	15.04	0.15	19.20/22.00*	0.33/0.18*	2.70	0.10
spruce	1440B	16.67	0.69	15.77	0.55	23.35	0.7	2.43	0.01
pathchirp	1000B	17.51	0.43	16.48	1.32	33.40	1.59	2.68	0.12
ptr	500B	11.09	1.70	11.76	2.52	19.45	2.90	1.88	0.15
igi	500B	10.73	1.03	12.52	2.69	26.10	10.62	2.04	0.06
pathload	200B	6.09 – 6.27	0.12	12.29 – 12.81	0.36	21.88 – 22.02	0.5	2.39 – 2.41	0.08
pathload	1440B	16.29 – 16.32	0.10	15.52 – 15.66	0.06	22.87 – 23.10	0.15	2.87-2.91	0.12

the sustainable rate is 13.04 Mbps. Hence, for Comcast, Tab. 1 presents UDP capacity and parallel TCP tests averaging the rate during 10 s (which includes 7 s of Powerboost and 3 s in the sustainable rate) and during only the first 6 s (which only measures the Powerboost rate). Optimized-probing tools can only measure the Powerboost rate. In our tests, where the local loop is the bottleneck and there is no cross traffic from the home, available bandwidth estimates should match the UDP capacity.

Results. The top part of Tab. 1 compares optimized-probing tools using the default configuration. *Spruce* gives the most accurate estimates when using default settings. Spruce had lower accuracy when interrupt coalescing was turned on. Although previous work shows that the probe-gap model underlying spruce works poorly under multiple bottlenecks [17, 18, 27], this scenario does not arise in our tests. The results for other optimized-probing tools are not accurate. Consistent with previous observations [8, 19], *pathchirp* overestimates the available bandwidth for most lines. Pathchirp's algorithm is extremely sensitive to interrupt latency variations and timer accuracy [14]. *Igi/ptr* underestimate the available bandwidth by 3 to 5 Mbps for Free and Orange and has large standard deviations. More surprisingly, *pathload*, which is known to report accurate estimates [3, 19, 24], significantly underestimates the available bandwidth in some cases (for instance, for Free by more than 60%).

The inspection of packet traces collected during the tests helps explain these results. We observe that both pathload and igi/ptr use small packets by default, whereas spruce and pathchirp use large packets. For igi/ptr, our tests showed that increasing the packet size does not improve a lot the estimation accuracy. Indeed, igi/ptr estimation algorithm is tuned to work best with 500 bytes to 800 bytes packets. However, for pathload, the bottom part of Tab. 1 shows that a larger probe size (1440 bytes) greatly improves the accuracy for all lines. Pathload's estimation with large probes matches perfectly the UDP capacity. We conclude that the use of small packets is the main cause of the poor performance of pathload with default parameters and we investigate this further in §4.2.

Parallel TCP is able to fully utilize all links and reports values that are close to the UDP capacity. The value is slightly below the UDP capacity because the rate computed by flooding-based tools is calculated as the total amount of data sent divided by the transfer time, which does not include TCP/IP headers and retransmissions. We perform few tests with a concurrent TCP connection from another computer in the home as cross traffic. In this experiment, parallel TCP reports values around 80% of the capacity, whereas pathload with large probes and spruce report available bandwidth close to zero. Both techniques are accurate in this case, this disparity comes from the difference in the definition of available bandwidth between flooding-based and optimized-probing tools. Parallel TCP captures the effective bandwidth available for home subscribes, but

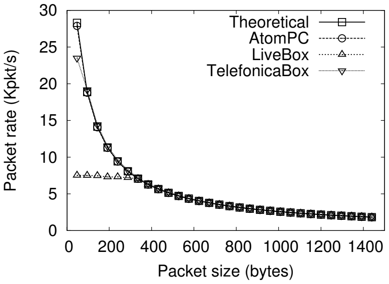


Fig. 2. Maximum packet rate for different packet sizes and different gateways.

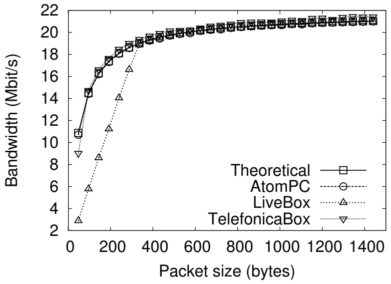


Fig. 3. Maximum bandwidth for different packet sizes and different gateways

pathload and spruce capture the residual capacity. A system that uses a combination of these two types of tools can give a better understanding of broadband speed.

Takeaway. Spruce, pathload with large probes, and parallel TCP are the most accurate tools to estimate available bandwidth.

4.2 Explanation of bandwidth limit with small probes

We investigate the reasons for tools that use small packets to underestimate the available bandwidth in some cases. The main difference between the tests with small packets versus tests with large packets is the probing rate. For a probing bandwidth of 16 Mbps, pathload sends 10,000 probes/s when using 200-bytes probes; this rate is only 1,389 probes/s for 1440-bytes probes. Given that pathload works well in other scenarios [3, 9, 24], our hypothesis is that one of the equipments between the access router and the measurement host in Fig. 1(a) limits or cannot sustain high probing rates.

Testbed description. To identify the limiting equipment we use a fully controlled testbed made of an Avicenia-L DSLAM, a measurement server, a measurement host, and three gateways: a *LiveBox*, the same used for Orange line in the previous section; a *TelefonicaBox* (Thomson TG797), with the same hardware as the LiveBox but with a proprietary real-time OS designed to minimize interrupt latency, instead of a plain linux; and the *AtomPC*, a more powerful Intel ATOM Z530 computer equipped with an ADSL card. The other gateways cannot be used in this testbed. The measurement server and host are Pentium PCs. The gateways are directly connected to the DSLAM and their ADSL links are configured at 24 Mbps ATM downlink and 1.2 Mbps ATM uplink. The measurement server is connected directly to the DSLAM via a GbE link. The measurement host is directly connected to the gateway via a GbE link.

We test the maximum achievable packet rate between the measurement server and the measurement host using all gateways. If the packet rate limitation comes from the DSLAM, then all gateways should achieve similar packet rates; otherwise, if the limitation is at the gateway, then different gateways will reach different packet rates. We test the packet-rate achieved for different packet sizes using iperf/UDP and varying the IP packet size from 40 bytes to 1488 bytes with 48 bytes increments (to avoid ATM padding). We collect packet traces at the receiver to compute the packet rate.

Livebox and Freebox cannot achieve high packet rates. Fig. 2 presents the average packet rate of iperf/UDP across the ten runs for the LiveBox, the TelefonicaBox, and

Table 2. Maximum packet rate of home gateways

Vendor/Model	Maximum pkts/s	Vendor/Model	Maximum pkts/s
FreeBox/ v5	3,344	D-Link/DIR-300	10,844
LiveBox/Thomson v1.2	6,859	D-Link/DIR-615	40,983
TelefonicaBox/Thomson TG797	22,021	Linksys/BEFSR41v2	2,228
ComcastBox/Ubee U10C035	45,312	Linksys/WR54G	10,225
Belkin/Enhanced N150	3,126	Netgear/WNR2000-100PES	659
Belkin/Wireless G Plus	6,777	Netgear/WGT624 v3	20,208

the AtomPC (the coefficient of variation is less than 0.01 for each packet size). This figure also shows the theoretical packet rate, which takes into account the ATM download speed, the encapsulation into cells, and layer-2 overhead, 40 bytes in our case. The lines of AtomPC and the TelefonicaBox closely follow the theoretical line. The AtomPC behaves according to the theoretical limit even when packets are very small. On the other hand, the LiveBox cannot sustain packet rates higher than 6,859 pkts/s. Such low achievable packet rates limit the maximum achievable throughput for small packets. Fig. 3 shows the achieved throughput for the corresponding tests. The LiveBox cannot fully utilize the link when packets are smaller than ≈400 bytes, because it cannot process packets fast enough. Hence, the speed decreases linearly as packets become smaller. For pathload's default of 200-bytes packets, Fig. 3 shows that the LiveBox's average throughput reaches only 12.6 Mbps. Pathload with small packets underestimates the available bandwidth, because it measures the LiveBox's maximum achievable packet rate. Now that we identified that gateways limit the packet rate, we can test the maximum packet rate of the FreeBox with iperf/UDP and 40-bytes packets using the setup of §3. The FreeBox limit is even lower: 3,344 pkts/s. Hence, the FreeBox can only reach an average of 6.1 Mbps for 200-bytes packets.

Other home gateways have similar limits on maximum achieved packet rates. We study the maximum achieved packet rate for a number of popular home gateways using the testbed deployed by Hatonen et al. [10]. In this testbed, gateways are connected to the server and the measurement host via a 100 Mbps Ethernet, so there is no limitation that comes from ADSL or cable technologies, just from the gateway hardware or software. We perform the iperf/UDP test with 40-bytes packets for 38 gateways tested in their study [10]. Tab. 2 presents the maximum packet rates for a selection of the gateways (due to space constraints, we only present the gateway model with lowest and highest maximum packet rate for the four main vendors). Tab. 2 also shows the maximum packet rates for each of the gateways tested earlier in this section. Although the maximum packet rate varies considerably even for gateways of the same vendor, most gateways do have a packet rate limit which is lower than pathload's probing rate.

Network-address translation slows down packet forwarding. In general, home gateways are known to have limited resources [10], but these achievable packet rates are extremely low. We see that either more powerful hardware (as in the AtomPC) or optimized operating system (as in the TelefonicaBox) can improve the maximum packet rate. We also observe that the ComcastBox achieves much higher packet rates. For Comcast, the measurement host is directly connected to the modem and has a public IP address. Hence, in this context there is no network-address translation (NAT). We conjecture that NAT explains at least part of the per-packet processing overhead. All the gateways in Tab. 2 but the ComcastBox perform address translation. To test this hypothesis, we run tests in the same Comcast line with the measurement host connected to

Table 3. Overhead of available bandwidth estimation. (*) 10s/6s(Powerboost)

Tool	**Pckt. Size**	**Free (Kbytes)**	**Orange (Kbytes)**	**Comcast (Kbytes)**	**AT&T (Kbytes)**
parallel TCP	1440B	20,769	20,908	26,844/19,177*	3,587
pathload	1440B	15,992	8,528	12,582	14,160
pathload	200B	2,053	622	2,041	2,067
spruce	1440B	288	288	288	288
pathchirp	1000B	1,140	1,140	1,140	1,105
igi/ptr	500B	414	475	457	475

the modem via a Belkin N1 Wireless Router. In this scenario, when NAT is disabled we achieve 31,458 pkts/s, whereas when NAT is enabled we achieve only 17,430 pkts/s.

Takeaway. Home gateways can have low packet forwarding rates. Hence, measurements of available bandwidth from homes need to ensure that the probing rate is lower than the gateway limit.[2]

5 Overhead

Each of the tools we study uses a different probing method with varying overhead. Tab. 3 presents the overhead for each tool. We define the overhead as the total number of bytes on the forward and reverse path during a test. We measure overhead by capturing all packets at the measurement host.

Parallel TCP's flooding-based approach is extremely invasive. For Free and Orange where the access link capacity is approximately 16 Mbps, parallel TCP will send over 20 Mbytes of probes during a 10-seconds test. Pathload with large packets also has significant overhead, which varies with the number of steps needed to converge. In fact, pathload by default uses small packets precisely to control the overhead (see pathload with 200-bytes packets). It is possible to reduce pathload's overhead by selecting the smallest probe size that leads to a probing rate lower than the packet rate limitation. Alternatively, one may use a tool such as Yaz [24], designed to reduce pathload's overhead. Spruce's overhead is independent of the access link capacity, because it always sends 100 pairs of probe packets. As access link capacity increases, the overhead of parallel TCP transfers becomes even higher. For instance, the overhead of parallel TCP is only 12 times higher than spruce's for AT&T, but it is 72 times higher for Orange.

Takeaway. Spruce has the lowest overhead of all tools. The overhead for flooding-based techniques is significantly higher than for optimized-probing tools.

6 Conclusion

This paper showed that current home gateways have low packet forwarding rates, which hurts the accuracy of some existing available bandwidth estimation tools. We found that spruce can accurately estimate the available bandwidth with minimum overhead. Spruce takes the capacity as input and requires interrupt coalescing to be turned off. In a setting like SamKnows, where measurements run from a fully controlled router, it is possible to turn off interrupt coalescing. Moreover, SamKnows tests download and upload speeds every two hours, so it is important to minimize the probing overhead and

[2] To find the gateway limit, measurement tools only need to perform the same iperf/UDP test with 40-bytes packets that we use here.

the initial test to measure the link capacity can be amortized over time. When turning off interrupt coalescing is not possible, pathload with a probe size optimized to cope with the gateway limitation is the best choice to measure available bandwidth. These optimized-probing tools, however, cannot fully replace flooding-based tools, because of the elasticity of TCP cross traffic. An approach that combines frequent measurements with spruce or pathload and less frequent flooding-based measurements will give the best compromise between accuracy and overhead.

Acknowledgements. We thank Technicolor for the access to the DSLAM testbed and P. Le Guyadec and L. Di Cioccio for their technical help; S. Hatonen, M. Kojo, A. Nyrhinen, and L. Eggert for the access and help with the gateway testbed; and S. Sundaresan for the access to the AT&T and the Comcast lines. We also thank I. Cunha, N. Feamster, D. Joumblatt, P. Loiseau, F. Schneider, and S. Sundaresan for the helpful comments. This work was conducted at the LINCS and supported by the ANR grant C'MON.

References

1. FCC Challenge, 2011. `http://challenge.gov/FCC/114-fcc-open-internet-apps-challenge`.
2. E. Altman, D. Barman, B. Tuffin, and M. Vojnovic. Parallel TCP sockets: Simple model, throughput and validation. In *Proc. IEEE INFOCOM*, 2006.
3. L. Angrisani, S. D'Antonio, M. Esposito, and M. Vadursi. Techniques for available bandwidth measurement in IP networks: a performance comparison. *Computer Networks*, 50(3), 2006.
4. S. Bauer, D. Clark, and W. Lehr. Understanding broadband speed measurements. *MITAS project white paper*, 2010.
5. D. Croce, T. En Najjary, G. Urvoy Keller, and E. W. Biersack. Non-cooperative available bandwidth estimation towards ADSL links. In *IEEE INFOCOM Workshops 2008, the 11th Global Internet Symposium*, 2008.
6. D. Croce, T. En-Najjary, G. Urvoy-Keller, and E. W. Biersack. Fast available bandwidth sampling for adsl links: Rethinking the estimation for larger-scale measurements. In *PAM*, 2009.
7. M. Dischinger, A. Haeberlen, K. P. Gummadi, and S. Saroiu. Characterizing Residential Broadband Networks. In *IMC*, 2007.
8. E. Goldoni and M. Schivi. End-to-end available bandwidth estimation tools, an experimental comparison. In *Proc. Traffic Monitoring and Analysis Workshop*, 2010.
9. C. D. Guerrero and M. A. Labrador. On the applicability of available bandwidth estimation techniques and tools. *Computer Communications*, 33(1), 2010.
10. S. Hatonen, A. Nyrhinen, L. Eggert, S. Strowes, P. Sarolahti, and M. Kojo. An experimental study of home gateway characteristics. In *IMC*, 2010.
11. N. Hu and P. Steenkiste. Evaluation and characterization of available bandwidth probing techniques. *IEEE J. Selected Areas in Communications*, 21(6), 2003.
12. V. Jacobson, R. Braden, and D. Borman. A Framework for Defining Empirical Bulk Transfer Capacity Metrics Status of this Memo. RFC 3148, 2001.
13. M. Jain and C. Dovrolis. End-to-end available bandwidth: Measurement methodology, dynamics, and relation with TCP throughput. In *Proc. ACM SIGCOMM*, 2002.
14. G. Jin and B. L. Tierney. System capability effects on algorithms for network bandwidth measurement. In *IMC*, 2003.
15. C. Kreibich, N. Weaver, B. Nechaev, and V. Paxson. Netalyzr: Illuminating the edge network. In *IMC*, 2010.
16. K. Lakshminarayanan, V. N. Padmanabhan, and J. Padhye. Bandwidth estimation in broadband access networks. In *IMC*, 2004.
17. L. Lao, C. Dovrolis, and M. Y. Sanadidi. The probe gap model can underestimate the available bandwidth of multihop paths. *ACM CCR*, 36(5), 2006.
18. X. Liu, K. Ravindran, and D. Loguinov. Multi-hop probing asymptotics in available bandwidth estimation: stochastic analysis. In *IMC*, 2005.
19. E. T. On, A. Shriram, M. Murray, Y. Hyun, N. Brownlee, A. Broido, M. Fomenkov, and K. Claffy. Comparison of public end-to-end bandwidth. In *PAM*, 2005.
20. R. Prasad, C. Dovrolis, M. Murray, and K. Claffy. Bandwidth estimation: metrics, measurement techniques, and tools. *IEEE Network Magazine*, 17(6), 2003.
21. R. Prasad, M. Jain, and C. Dovrolis. Effects of interrupt coalescence on network measurements. In *PAM*, 2004.
22. V. J. Ribeiro, R. H. Riedi, R. G. Baraniuk Jiri Navratil, and L. Cottrell. pathChirp: Efficient available bandwidth estimation for network paths. In *PAM*, 2003.
23. A. Shriram and J. Kaur. Empirical evaluation of techniques for measuring available bandwidth. In *Proc. IEEE INFOCOM*, 2007.
24. J. Sommers, P. Barford, and W. Willinger. Laboratory-based calibration of available bandwidth estimation tools. *Microprocessors and Microsystems Journal*, 31, 2007.
25. J. Strauss, D. Katabi, and F. Kaashoek. A measurement study of available bandwidth estimation tools. In *IMC*, 2003.
26. S. Sundaresan, W. de Donato, N. Feamster, R. Teixeira, S. Crawford, and A. Pescapè. Broadband internet performance: A view from the gateway. In *Proc. ACM SIGCOMM*, 2011.
27. G. Urvoy-Keller, T. En-Najjary, and A. Sorniotti. Operational comparison of available bandwidth estimation tools. *ACM CCR*, 38(1), 2008.