Fast Available Bandwidth sampling for ADSL links: rethinking the estimation for larger-scale measurements

Daniele Croce, Taoufik En-Najjary, Guillaume Urvoy-Keller and Ernst W. Biersack

EURECOM, Sophia Antipolis, France
Email: {croce,ennajjar,urvoy,erbi}@eurecom.fr

Abstract. Most existing tools for measuring the end-to-end available bandwidth require access to both end-hosts of the measured path, which severely restricts their usability. Few tools have been developed to overcome this limitation, but all of them focus on achieving high precision and are not suitable for large campaigns. In this paper we develop *FAB-Probe*, a tool aimed at characterizing the available bandwidth of a large number of paths, adapting it particularly for ADSL settings. FAB-Probe is an evolution of *ABwProbe*, a tool that estimates the available bandwidth in non-cooperative ADSL environments. Analyzing carefully the needs of such a characterization tool, we optimize and rethink ABwProbe for larger-scale measurements. The validation of FAB-Probe is obtained both "in-lab", with ADSL hosts under our control, as well as under real traffic conditions, with the help of an ISP. Finally, as a proof of concept, we analyze the available bandwidth of over 1300 hosts participating to the KAD DHT used by eMule, periodically monitoring some static peers for over ten days.[⋆]

1 Introduction and motivation

Nowadays, over 221 million users access the Internet through broadband links and in many developed countries broadband technologies have reached more than half of the households, with peaks of over 90% penetration [10]. Moreover, subscribers are expected to double in the next four years, confirming the exponential increase observed until now [13]. Over 137 millions broadband users (62% of all subscribers) access the Internet through an ADSL link. Broadband technologies have allowed the development and the success of new bandwidth intensive applications (peer-to-peer applications for example). This have pushed the consumption of bandwidth and, therefore, the development of new, high capacity technologies. Despite these improvements, broadband technology is still the fundamental bottleneck for achieving higher performances [3].

Researchers had, up to now, limited access to information related to the characteristics of residential broadband networks. This was mainly caused by the difficulties in measuring these networks without explicit cooperation of the end hosts or the ISPs. Indeed, only few existing tools have been designed to work in non-cooperative[1] environments and *all focus on achieving high precision* while are not suitable for large

[⋆] This work is supported in part by the NANODATA-CENTERS program (FP7-ICT-223850) of the EU.

[1] Following the terminology used in [2, 11], with "non-cooperative" we mean that access to the remote host is not available.

measurement campaigns. Moreover, to the best of our knowledge, only ABwProbe [2] is tuned precisely for the available bandwidth (avail-bw) estimation of ADSL links.

In this paper, we propose FAB-Probe which is the natural evolution of ABwProbe. We rethink and optimize the tool for larger-scale measurements, giving particular attention to the *time needed for obtaining a good estimate*. Observing common traffic patterns, we elaborate a new strategy for sampling the avail-bw efficiently, adapting it to the absolute capacity of the ADSL. We reduce at minimum the number of probes and we refine existing cross-traffic filtering techniques. We also develop a method for detecting uplink congestion, which can alter the measurements. After validating FAB-Probe "in-lab" against ADSL hosts under our control, we test it in real traffic conditions with the collaboration of an ISP, obtaining very good results. Finally, as a proof of concept, we use FAB-Probe for measuring over 1300 peers participating in KAD, the DHT used by eMule: we measure the hosts closely in time so to obtain a snapshot of the avail-bw distribution, and we select 82 with static addresses that we measure periodically for over 10 days. We also provide FAB-Probe for public evaluation, available at [5].

2 Background and related work

2.1 Active measurements in non-cooperative environments

In a cooperative environment, active measurements are usually done in the following way: a specific number of probing packets with appropriate characteristics (size, rate, etc.) are transmitted through the network, link or device, that has to be measured, towards a receiver. On the receiver side, the probes are captured and some metrics (latency, rate, inter-packet gap, etc.) are computed. By analyzing and comparing these metrics at the receiver, it is possible to infer some characteristics of the network traversed and to estimate the desired quantity.

In a non-cooperative environment, instead, there is no control on the receiving host and, thus, there is no way to analyze the probes received at the destination. Therefore, all statistical information on the probes must be obtained in a different and indirect manner. An idea that has been adopted in various tools [1–3, 6, 7, 11], is to send particular types of probes which should induce the receiving host to reply with some other packets, "echoing" the received probes. Examples of probes that have this quality are: ICMP Echo and Timestamp requests, most TCP packets (ACKs, SYNs, FINs, etc. but not RSTs), UDP packets sent to closed ports.

When undertaking measurements in this non-cooperative way, it becomes challenging to distinguish between the characteristics of the forward and of the reverse path: when measuring the forward path, we must verify that the observed metrics capture the characteristics of the forward path and not the characteristics of the reverse path (and viceversa). An important feature that makes TCP-based probes attractive, is that the packet sent back in reply by the non-cooperative host is a 40 byte packet (in most cases a RST) regardless of the size of the probe that has generated it. This property is fundamental for measuring the avail-bw: since packets traverse the network in both directions, the size of the probes must be adjusted to differentiate between forward and reverse path and interpret the results correctly. *In this work we focus on the estimation of the downlink avail-bw*, as the uplink estimation is conceptually similar.

2.2 Tools for non-cooperative available bandwidth estimation

While a multitude of utilities exist for cooperative environments, very few are designed to measure the avail-bw in settings where cooperation is not granted. Two tools, *pathneck* [7] and *Sprobe* [11], do not require access to the receiving host, however, the first one measures the Average Dispersion Rate (ADR, an upper bound of the avail-bw) and, since it is based on ICMP probes, can suffer from ICMP rate limiting; the second one is based on the Probe Gap Model which has been criticized and proven inaccurate in [8]. Some techniques dedicated to broadband networks (ADSL and Cable) are described in [3] for the estimation of capacity, buffer depth, queue management and others. However, the avail-bw is not taken into account.

To the best of our knowledge, the only tool designed especially to measure the avail-bw in non-cooperative environments is ABwProbe, which is particularly tuned to measure non-cooperative ADSL hosts. FAB-Probe is thus a natural evolution of ABwProbe which in turn is inspired by Pathload [4]. Both ABwProbe and FAB-Probe use the RTT in place of the One-Way Delay (OWD) of Pathload: if no cross-traffic disturbs the RSTs *on the reverse path*, indeed, an increasing trend in the OWDs will be preserved and will appear in the RTT samples as well. ACK probes are sent at rate R to the non-cooperative host which replies with fixed-size RSTs. The probes are only echoed by the receiver and at the sender the RTT is computed. This way, the RTT measured is used in place of the OWD (which is impossible to measure) for detecting if $R > A$ or not, A being the avail-bw of the path. However, since packets traverse the network a second time, the received RSTs will carry both information of the probes from the forward path, but will also be influenced by the characteristics of the reverse path. This becomes an important problem in asymmetric environments such ADSL because RSTs have to traverse the uplink of the ADSL which has much lower capacity (and usually lower avail-bw) than the downlink.

To overcome this asymmetry, in ABwProbe (and in FAB-Probe as well) we exploit the fact that TCP RSTs are always 40 bytes long regardless of the size of the ACKs: if the probes are very large, say 1500 bytes, then the rate generated on the downlink will be $1500/40 = 37.5$ times higher than the load on the uplink (Layer-2 overheads excluded, see [2]). Now, most ADSLs have uplink capacities that are less than 8 times lower than the downlink [3] so this measure should be sufficient to overcome the *capacity* asymmetry. In FAB-Probe an additional test is implemented to detect when the uplink *avail-bw* is insufficient for the RSTs to be transmitted on the uplink.

Finally, even when the avail-bw on the uplink is sufficient, it has been shown in [2] that large cross-traffic packets interfering on the uplink can significantly affect the measurement because of the compression of the RSTs queued behind. Since an MTU packet can take several tens of milliseconds to be transmitted, this effect is not negligible. In ABwProbe some techniques were proposed to detect and filter RTT samples affected by RST compression. As we will explain later, in FAB-Probe we improve the filtering by combining the proposed methods to better exploit their strengths.

Like other tools, ABwProbe too is aimed at providing very high precision in estimating the avail-bw. In FAB-Probe we intentionally renounce to some measurement resolution, in favor of the measurement latency which is remarkably reduced.

3 FAB-Probe: Fast Available Bandwidth sampling for ADSL links

3.1 What should we measure exactly?

Most tools measuring the avail-bw aim at providing an *absolute* estimation regardless of the capacity of the link. This makes sense for applications (such as video streaming) that require a certain amount of bandwidth to work properly. In general, however, to characterize a network and the load generated by the users, it is more interesting to know the avail-bw *relative* to the absolute capacity. For example, it is often more useful to know that the bottleneck link has 30% left of its spare capacity rather than knowing there are 200 kbps available, as the first metric intrinsically provides an idea of link quality while the latter does not. Additionally, for most large-scale characterization purposes, it is critical to have a fast estimation process while it is less important to have extremely high precision. For example, knowing that the avail-bw is in between 90% and 100% of the total capacity is usually enough for classifying the link as *inactive*. This aspect is very important because most estimation tools (including ABwProbe) reiterate the probing phase until the desired precision is obtained, i.e. higher precision translates in longer running time. The intuition behind FAB-Probe is thus to analyze the avail-bw relative to the capacity, probing the avail-bw at few key values (only 5 by default) and providing a quick, but still accurate, estimate of the avail-bw of the link. In this paper the focus is on the avail-bw estimation and discussions on how to estimate the capacity are out of its scope. For a description of the capacity estimation technique used in FAB-Probe the interested reader can refer to [1].

Focusing on larger-scale measurements, it is important to note that the large majority of users are either *inactive*, i.e. not using their bandwidth, or *active*, meaning they are actively using their connection. This active-inactive behavior has already been observed in [14] and is confirmed by our experimental results. Recalling that the network bottleneck has been shown to be at the edge of the network [3], there are high chances that active periods tend to bring the ADSL close to congestion. We can thus design the avail-bw estimation algorithm exploiting this *a priori* information. In particular, with an active-inactive pattern, the measurements should be more fine-grained on extreme avail-bw values (closer to 0 and 100%) while can be less precise on “middle” values. In FAB-Probe we thus limit the exploration of the avail-bw range by testing only few key values, distributed in the following way: 10%, 25%, 50%, 75% and 90% and the avail-bw will be captured in between two of these values (and 0 and 100% but these values are obviously not probed). This guarantees that the number of probing iterations is always less than 5 (see details below), achieving a considerable speedup of the measurement process while limiting the probing to significant values of the avail-bw range. Clearly, the precision can be increased (or reduced) by changing the number of predefined values and also the distribution of the values itself can be changed to better fit the avail-bw distribution – if this is available. Finally, note that the *absolute* avail-bw can still be obtained from the relative values of FAB-Probe because the capacity is known.

3.2 Measurement algorithm and speedup

In FAB-Probe we first measure the downlink capacity C with the techniques in [1]. Then, we explore the avail-bw range with a binary search-like algorithm, modifying the one in ABwProbe: the path is probed with a fleet of packets at an initial rate $R = 75\% \times C$ and the RTT of the packets is analyzed to detect if there is an *increasing trend*. An increasing trend would indicate self-induced congestion on the path, thus meaning that $R > A$. The rate is then reduced (increased) if an increasing trend (no trend) in the RTTs is detected. Consequently, R is updated to a lower (higher) value, i.e. 50 or 90%, and another fleet is sent. The process is iterated until the avail-bw is captured between two probed values, where one has shown an increasing trend, the other one no trend. For example, if $A = 60\% \times C$, the first fleet at 75% would have an increasing trend, while the second fleet at 50% would show no trend. Since there are no other measuring values in between 50 and 75%, the algorithm stops and FAB-Probe would output an avail-bw range of 50-75%. The main difference with previous algorithms here, is that instead of changing R in a pure binary search fashion, in FAB-Probe the probing rate R assumes few deterministic values, significantly reducing the running time.

To further speedup the measurement time, we have worked also on the number of fleets to send. In Pathload, to detect if there is an increasing trend or not, 12 independent fleets are sent at the same rate R. Then, the trend is computed on all these fleets and the decision if $R > A$ or not is taken based on the fraction of fleets agreeing. This process is repeated for all probing rates regardless of the trend intensity. In [2] we already proposed a strategy to accelerate the decision process, reducing on-the-fly the number of fleets when the trends are particularly pronounced, thus adapting the number of fleets to the difference between R and A. In FAB-Probe we push the idea to the limit as we simply do not investigate bandwidth rates that are too close to the avail-bw. The decision is taken based on the result of one fleet only: if the trend is clearly noticeable, the measurement is valid and the rate R is reduced (increased) right away. Otherwise, if the trend is not pronounced, this means that the avail-bw is close to the rate R, the fleet is marked as "grey", and two other fleets are sent at higher and lower rate just as a counter check. The rationale behind this is the following: suppose the avail-bw is close to 70% of the capacity. If the probing rate R is at 75% of the capacity, there will be a very weak increasing trend in the RTT samples. In a situation like this, ABwProbe would send more and more fleets (up to 12) to have good confidence before declaring that $R > A$, getting "stuck" on this rate value. In FAB-Probe instead, we immediately probe the link at 50% and 90% (the values just above and below 75%) and, if the avail-bw is confirmed to be in between, the range given in output would be 50-90% – instead of 50-75%. This way we obtain an acceptable result with very few fleets (only 3 in the example), trading off precision for running time, and obtaining a great speedup.

3.3 Uplink congestion and cross-traffic

Suppose an increasing trend in the RTTs is detected while measuring the downlink avail-bw. Since the estimation is done in a non-cooperative environment, we must verify that the observed increase in RTT is due to the self-induced congestion on the downlink and not to compression of the RSTs on the reverse path. The uplink of the ADSL, in particular, is critical because of the very low bandwidth that can be easily saturated. For this reason, after the first increasing fleet is detected, we make this simple test to check the conditions of the uplink: we send a second fleet in which the probes have the same spacing between each other but the size is at minimum (40 Bytes). This fleet reproduces on the reverse path a sequence of RSTs with the same *rate* of the fleet before (same size, same gap), however, on the downlink the rate is at least 16 times lower (more if ATM is not adopted, see [2]). Now, if this fleet still shows an increasing trend either (i) the downlink is completely congested, or (ii) the uplink is loaded and the RSTs on the reverse path are compressed. In both cases, however, estimating the avail-bw precisely in these critical conditions becomes problematic (also in terms of measuring time because the rates are extremely low) therefore we classify the ADSL as "congested" without investigating further.

In FAB-Probe we also refine the filtering methods presented in [2]. Indeed, uplink cross-traffic can compress the RSTs on the uplink, even without serious congestion, generating spikes in the RTT that temporarily "hide" the temporal properties of the downlink. In [2] two techniques are proposed, one statistical based on a robust method called Iteratively Re-wighted Least Squares, and one deterministic, aimed at detecting the consecutive decreasing RTT samples resulting from the compression of the RSTs. The two methods have interesting properties but also downsides: the statistical method suffers in high cross-traffic conditions while the deterministic one is ineffective when the cross-traffic packets are small. In FAB-Probe we combine the two methods to benefit from both approaches: first we filter the large majority of affected RTT samples detecting consecutive decreasing trends, then we filter the remaining samples applying the robust statistical method. As we will show, this allows us to obtain correct measurements even when the uplink is highly loaded.

4 Validation

To validate the accuracy of FAB-Probe, we have tested our tool both on some ADSL hosts under our control and against hosts belonging to an ADSL service provider from which we obtained the traffic traces as "ground truth". The measuring host was always a well-connected host, with a 100Mbps connection and with at least 90 Mbps avail-bw, so that the bottleneck link was the ADSL on the other edge of the network.

4.1 In-lab validation

We tested the accuracy of FAB-Probe in different traffic conditions and with various cross-traffic rates. From a third well-connected host, we used a traffic generator to inject CBR traffic towards the ADSL host. We vary the load on the downlink spacing several rates close to the ones used by FAB-Probe to probe, so to verify that no error occurs when R and A are close – the worst case. For example, we test FAB-Probe with 45% and 55% avail-bw because FAB-Probe runs fleets at 50%, thus verifying if the trend is correctly dectected. Tab. 1 summarizes the results given in output by FAB-Probe as well as the total number of fleets sent and the running time. The improvement compared to ABwProbe is remarkable: the number of fleets is an order of magnitude lower, thus reducing significantly the intrusiveness of the measurement, and the avail-bw estimate is obtained over 5 times faster while still providing precise avail-bw ranges.

To prove the robustness of FAB-Probe and the effectiveness of the uplink cross-traffic filtering, we have measured the downlink both in idle and loaded conditions

		Downlink avail-bw range (relative to capacity)							
	Real avail-bw	95%	85%	70%	55%	45%	30%	20%	5%
FAB-P.	Measured range	90-100%	75-90%	50-75%	50-75%	25-50%	25-50%	10-25%	cong.
	# of fleets	2	2*	3*	3*	3*	3*	3*	1*
	Time (sec)	5.6	7.6	14.4	14.5	13.5	14.7	23.0	6.0
ABP	# of fleets	17	19	21	24	23	18	21	16
	Time (sec)	31.8	40.2	57.4	68.4	70.3	56.9	88.3	200.9

Table 1. Accuracy and speed of FAB-Probe: the output range effectively captures the avail-bw. Compared to ABwProbe (ABP), the number of probing fleets and the running time are considerably lower. The asterisk indicates that an additional fleet with 40 bytes probe packets was sent to verify uplink congestion.

	Uplink cross-traffic (load relative to uplink capacity)							
	0%	15%	25%	35%	45%	55%	65%	75%
Measured range (100% avail-bw)	**90-100%**	90-100%	90-100%	90-100%	75-100%	75-100%	75-90%	cong.
Measured range (30% avail-bw)	**25-50%**	25-50%	25-50%	25-50%	25-50%	25-50%	10-25%	cong.

Table 2. Impact of uplink cross-traffic: FAB-Probe correctly measures the avail-bw in presence of cross-traffic both when the downlink is idle and loaded (30% avail-bw).

while injecting increasing cross-traffic on the uplink. We generated CBR traffic from the ADSL host towards a third machine using MTU size packets (thus causing the highest RST compression, see [2]). As shown in Tab. 2, the cross-traffic filtering allows FAB-Probe to correctly estimate the downlink avail-bw both when the link is idle and loaded at 70%, and the congestion test operates when the cross-traffic is too high (over 450 kbps in the case shown, with an uplink of 578 kbps). Only a slight underestimation occurs in extreme cross-traffic conditions.

4.2 Validation "in the wild"

To further test the accuracy of FAB-Probe, we validated the tool in real-world traffic conditions. With the help of an ADSL service provider, we first selected few hundred hosts which were not idle nor completely saturated, in order to have some avail-bw variability. Then, we periodically measured 13 ADSL hosts during a one hour period. The ISP provided us the time series of the incoming and outgoing traffic of the measured hosts and we could thus compute the amount of traffic actually traversing the ADSL. The results obtained matched very well the ones provided by FAB-Probe and two interesting examples are provided in Fig. 1. In Fig. 1(a), FAB-Probe was precisely measuring the avail-bw, being conservative and widening the range given in output when the probing rate was too close to the real value ("grey" trend). In Fig. 1(b) the host measured is a bit more loaded and the variability of the avail-bw process is higher. Nevertheless, FAB-Probe provides very good estimates, with only one inaccurate sample.

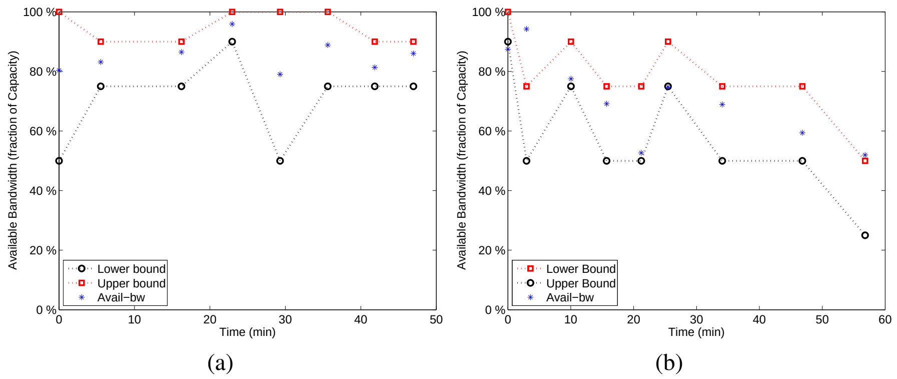


Fig. 1. Real-world traffic conditions: FAB-Probe effectively captures the avail-bw of the hosts.

5 Experimental Results on KAD

As a proof of concept, in this section we show the results obtained with FAB-Probe on over 1300 ADSL hosts participating to the Kademlia DHT [9] used by applications like eMule. We used the KAD crawler of [12] to retrieve the IP addresses of the KAD hosts and then selected with the Maxmind database only those connecting through an ADSL service provider. We have run two different experiments: one aiming at analyzing a large variety of hosts closely in time, generating a snapshot of the avail-bw distribution, and another measuring a particular subset of hosts for over 10 days.

5.1 A snapshot of the avail-bw distribution

We have measured 1244 hosts from various ISPs in US and Europe analyzing both capacity and avail-bw (although we omit the discussion on the capacity, see [1] on this matter). Avail-bw measurements lasted on average less than 6 seconds per host. Fig. 2(a)

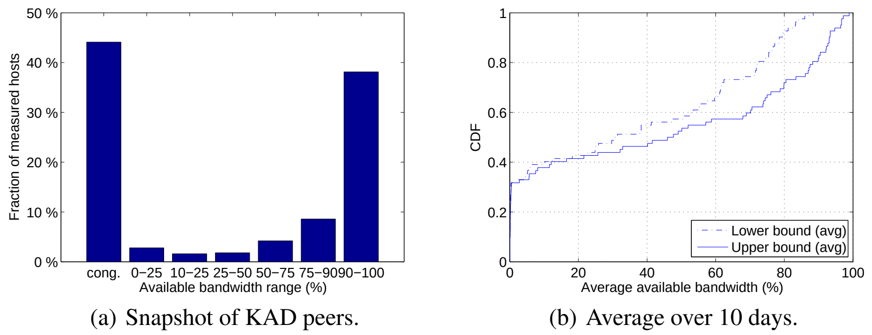
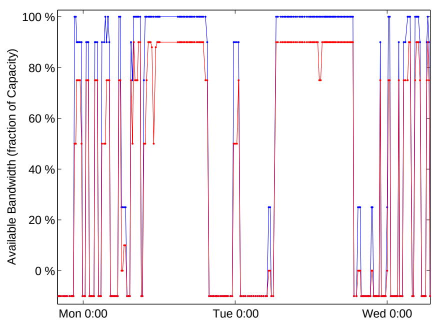
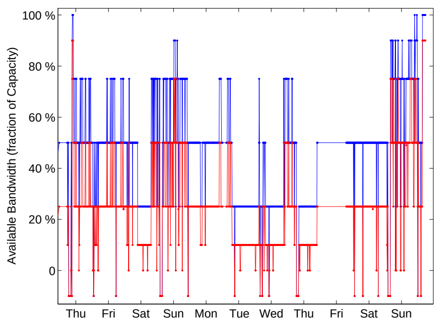


Fig. 2. Avail-bw in KAD: snapshot (a) and 10 days average (b) of the downlink avail-bw.



(a) Daily pattern (zoom).



(b) No daily pattern.

Fig. 3. Avail-bw evolution over time: comparison of two hosts that had different avail-bw patterns. Negative values indicate that the uplink was congested.

shows the downlink avail-bw range for these hosts. Confirming the results in [14], hosts are sharply partitioned in two groups, *idle* or *active*, since more than 80% have either over 90% avail-bw or a congested uplink[2]. Less than 2% of the hosts measured had a fleet with unclear trend ("grey" in Pathload) resulting in a wider avail-bw range, such as 50%-90%, so we omitted them from the figure.

5.2 A ten-day case study

From the hosts participating in KAD, we selected the ones that were continuously online for over one month using the same (static) IP address. These hosts are a good opportunity for a long-term analysis of the avail-bw evolution. However, they constitute a particular subset of eMule hosts that might continuously be seeding the file distributions acting as traditional servers. We have selected 82 of these hosts and measured them every 5 minutes for over 10 days. The capacity was also monitored to make sure it did not change during that period. In Fig. 2(b) are shown the *average* avail-bw lower and upper bounds considering the entire measurement period. Over 30% of the hosts had the uplink continuously congested meaning they were always uploading large amounts of data. Over 25% of the hosts were instead frequently idle, as on average, they had *at least* 75% avail-bw. We emphasize that the distribution is less sharp than the snapshot in Fig. 2(a) because Fig. 2(b) shows the CDF of averages computed over almost 3000 estimates per host, while the latter is a one shot measurement.

Analyzing more closely the results, some hosts showed sharp daily patterns in the avail-bw, while others did not. For example, Fig. 3(a) shows a host that was basically idle during night hours while was active during the day, while Fig. 3(b) shows another host which showed more slow-varying patterns during the measurement campaign. In

[2] Modern applications for P2P file distribution employ "tit-for-tat" algorithms that encourage the the peers to actively participate to the distribution process. This means that peers upload data while downloading, thus using their bandwidth in a more symmetrical fashion compared to client-server applications. Now, the uplink of an ADSL being much smaller than the downlink, this explains why eMule peers suffer from uplink congestion before the downlink is saturated.

general, however, this set of hosts showed little change in their avail-bw between day and night, yet maintaining an active-inactive pattern with sometimes steep variations.

6 Conclusions

In this paper we presented FAB-Probe, a new measurement tool for non-cooperative ADSL environments, designed for larger-scale measurements. Rethinking previous techniques and studying the requirements in terms of accuracy and time, FAB-Probe is the evolution of ABwProbe, from which the basic algorithm was acquired. We developed a new sampling strategy, balancing accuracy and measuring time to sample efficiently the avail-bw; we minimized the number of probing fleets; we studied a method to detect uplink congestion; and we improved the cross-traffic filtering technique. We proved the accuracy of FAB-Probe both "in-lab", against ADSL hosts under our control, and in real traffic conditions, with the cooperation of an ISP. Finally, as a proof of concept, we measured with FAB-Probe over 1300 ADSL hosts participating in KAD, for a snapshot of the avail-bw distribution as well as a characterization of 82 hosts for over ten days.

Acknowledgments

We are extremely thankful to Marcin Pietrzyk for providing us the traces used in the validation and to Moritz Steiner for his help in obtaining the IPs of the KAD users.

References

1. D. Croce, T. En-Najjary, G. Urvoy-Keller, and E. Biersack. Capacity Estimation of ADSL links. In *CoNEXT*, Dec. 2008.
2. D. Croce, T. En-Najjary, G. Urvoy-Keller, and E. Biersack. Non-cooperative Available Bandwidth Estimation towards ADSL links. In *Proc. Global Internet Symposium 2008*, Apr. 2008.
3. M. Dischinger, A. Haeberlen, K. P. Gummadi, and S. Saroiu. Characterizing residential broadband networks. In *IMC*, Oct. 2007.
4. C. Dovrolis and M. Jain. End-to-end available bandwidth: Measurement methodology, dynamics, and relation with TCP throughput. In *SIGCOMM*, Pittsburgh, USA, Aug. 2002.
5. The FAB-Probe Project website: *http://www.eurecom.fr/~btroup/fabprobe.html*
6. A. Haeberlen, M. Dischinger, K. P. Gummadi, and S. Saroiu. Monarch: A tool to emulate transport protocol flows over the internet at large. In *IMC*, Oct 2006.
7. N. Hu, L. E. Li, Z. M. Mao, P. Steenkiste, and J. Wang. Locating internet bottlenecks: algorithms, measurements, and implications. In *SIGCOMM*, Portland, USA, 2004.
8. L. Lao, C. Dovrolis, and M. Y. Sanadidi. The probe gap model can underestimate the available bandwidth of multihop paths. *Computer Communication Review*, 36(5):29–34, 2006.
9. P. Maymounkov and D. Mazieres. Kademlia: A Peer-to-peer informatiion system based on the XOR metric. In *IPTPS*, Mar. 2002.
10. OECD. OECD broadband statistics, *http://www.oecd.org/sti/ict/broadband*, June 2007.
11. S. Saroiu, P. K. Gummadi, and S. D. Gribble. Sprobe: A fast technique for measuring bottleneck bandwidth in uncooperative environments. *http://sprobe.cs.washinton.edu*, 2002.
12. M. Steiner, T. En-Najjary, and E. W. Biersack. A Global View of KAD. In *IMC*, 2007.
13. L. Windsor Oaks Group. Annual market outlook report, 2006.
14. D. Zaragoza and C. Belo. Experimental validation of the on-off packet-level model for ip traffic. *Comput. Commun.*, 30(5):975–989, 2007.