Evaluation of Active Measurement Tools for Bandwidth Estimation in Real Environment

Y. Labit, P.Owezarski, N. Larrieu
LAAS-CNRS
7, avenue du Colonel Roche
31077 TOULOUSE Cedex 7
FRANCE
{ylabit, owe, nlarrieu}@laas.fr

Abstract

Available bandwidth – as well as capacity or achievable bandwidth – on a path or a link is one of the very important parameters to measure or estimate in a network: it is of high interest for many networking functions (routing, admission and congestion control, load balancing, etc). Active probing techniques provide the easiest and the more flexible approach, for estimating available bandwidth. In addition, they can be used for different network technologies or structures. Many techniques and tools for available bandwidth estimation appeared recently, but little attention has been given to the accuracy of the estimated values in the real Internet, most of previous studies focusing on validating the accuracy of these tools on local platform. Therefore, this paper deals with evaluating the accuracy of active estimation tools in the real wide area Internet. We use passive monitoring tools for this purpose. We then built a platform combining active and passive equipments, and define a methodology for evaluating active probing techniques using passive tools. The passive evaluation relies on DAG system cards that represent references for such kind of measurements. This paper then discusses the results we got in the different experiments with different tools. In particular, we use traffic generators for changing the characteristics of the traffic on the Internet paths, which we are making our measurements on. It is useful for analyzing the accuracy of active estimation tools according to network and traffic conditions.

Keywords

Passive evaluation of active measurements, available bandwidth estimation, measurement analysis, Internet networks.

1. Introduction

Having an accurate estimation of available bandwidth on network links or on end-to-end paths is of high interest for many functions in networking as admission control, load balancing, quality of service (QoS)-routing, congestion control, etc. Passive monitoring tools are certainly the most appropriate tools for this purpose. But they are most of the time not accessible to users that need such information. Even for carriers or ISP that manage their own domain or autonomous system, and that then can have access to any information they need about their own network state, they miss the same type of information for the networks of other carriers or ISP they are connected to. As a consequence, tools for estimating available bandwidth on an end-to-end path are based on active measurement techniques, which are said to be user oriented, at the opposite of passive measurements which are carrier or ISP oriented. With the active approach, these tools then provide a solution for having an easy access to such network feature estimations, and this can be used for any network structures and technologies. Many tools for estimating available bandwidth have appeared in the recent years as Abing [9], Spruce [7], Pathload [3], [4], IGI-PTR [2], Pathchirp [6], etc. But tools based on active measurements for available bandwidth only make possible to get estimations on this parameter, while passive monitoring tools can measure it in a very accurate way. The question then deals with the accuracy of available bandwidth estimation tools based on active techniques. In addition, there are very few comparisons between all these tools in real environments as the actual Internet. Existing literature essentially focuses on evaluating these tools on local and fully controlled platforms. It is then very difficult for potential users to select the best tool depending on their requirements. And we are facing this kind of problem: we need to estimate available bandwidth, but we are unable based on the current literature to find out the best suited tool for our need. We then started a study on the accuracy and efficiency of the main available bandwidth estimation tools.

However, it is important to recall that active measurements consist in generating probe traffic in the network, and then observing the impact of network components and protocols on traffic: loss rate, delays, RTT, etc. Therefore, as active measurement tools generate traffic in the network, one of their major drawbacks is related to the disturbance introduced by the probe traffic which can make the network QoS change, and thus provide erroneous measures. Sometimes, active probing traffic can be seen as denial of service attacks, scanning, etc; but in any case as hacker acts. Probe traffic is then discarded, and its source can be blacklisted. Intrusiveness of probe traffic is then one of the key features which active measurement tools have to care about. Besides, much work addresses this issue of probe traffic intrusiveness, trying to minimize the number of sent packets as well as their impacts on the network QoS. In addition, if an active measurement tool generates only few packets, it would certainly provide estimation results in a very short time, what is an important performance parameter in the Internet whose traffic is very versatile.

This paper evaluates the accuracy of active measurement tools aiming at measuring the available bandwidth on a path from a source to a destination workstation, as well

as its intrusiveness level and response time. This evaluation relies on the use of very accurate passive monitoring tools, based on the DAG card [1] which is an absolute reference. This paper then first presents the main metrics for active measurement tools, and a list of tools which have been evaluated: these tools are classified according to their estimation / measurement technique, but also according to the kind of parameters they measure / evaluate (section 2). For instance, some, already quoted, measure available bandwidth, while other, as Clink [11], Pchar [12], Pathchar [10], etc, measure links or paths capacities. These two families of tools are important for this evaluation work as some of the available bandwidth estimation tools need to know the link or path physical capacity. The study and analysis proposed in this paper is being performed in the framework of the French Metropolis project which is presented in section 3. In particular, it is also explained in this section how the evaluation and analysis is going to be performed. It describes how active and passive measurement equipments, composing the Metropolis monitoring and measurement platform, can be jointly used for this purpose. Finally, section 4 presents results for the two families of tools we are considering, i.e. the one of tools measuring link or path capacity, whose results will be used for evaluating the results of the second tool family dealing with available bandwidth estimation.

2. Metrics, Techniques and Tools

Before sending data on a path of the network, users may want to know some information concerning QoS. It can be the same for network operators who want to optimize their routing strategy. Evaluating QoS and performances on a path most of the time deals with measuring or estimating capacity, available bandwidth, utilization level, loss ratio, etc. These parameters will give an idea of the QoS and performances users can expect. The following definitions present the main metrics to be used in this paper.

- Concerning data transmission, the term bandwidth or "capacity" is related to the width of the communication pipe and how quickly bits can be sent. The capacity can be defined as the maximum quantity of data per time unit when there is no cross traffic. We will speak about capacity of a link or a path. By considering a path of N links l_1, l_2, l_3 ..., l_N, we define the capacity of each link by C_1, C_2, C_3 ..., C_N. The capacity C of a path is determined by the minimum capacity of a link. This link is called the narrow link. Let's note: $C = \min(C_1, C_2, C_3..., C_N)$. In the following example (See Figure 1), the capacity of the path corresponds to the one of the narrowest link which is C_1.
- The utilization of a link is the consumed part of the link capacity. Let's note U_i the utilization of the ith link.
- The available bandwidth is defined as the unused capacity in the link independently of the transport protocol. The available bandwidth is a function resulting from the utilization and the capacity. Let's consider the first path, made of N links: the available bandwidth for the ith link is defined by:

$$AvB_i = C_i (1 - U_i) \quad (1)$$

The available bandwidth of a path is designed by the link which has the lowest available bandwidth:

$$AvB = \min (AvB_1, AvB_2, AvB_3 \ldots, AvB_N) \quad (2)$$

The link having the minimum available bandwidth is called the tight link. In the example below, the tight link defining the available bandwidth is l_3 and *AvB* equals to AvB_3.

- Intrusiveness can be defined as the percentage of capacity that is consumed for the measurements. It means that intrusiveness I_X is equal to:

$$I_X (\%) = 100 (C_X/C) \quad (3)$$

where C_X and C are respectively the amount of traffic sent in one second and the link capacity. C_X is the amount of bits generated in one run by the tool X during the probing time (time between the first and the last probe bit generated by the tool X). The probing time is different from the response time: The first probing bit is not necessarily sent to the destination at the moment the tool starts and the tool does not necessarily return its estimation right after sending the last probe bit.

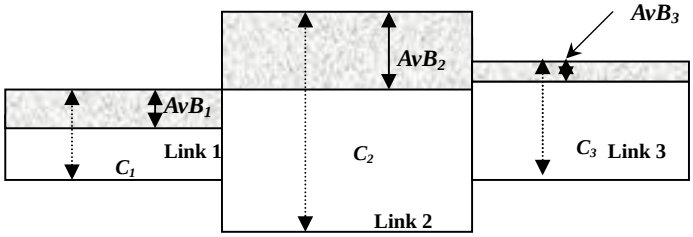

Figure 1: Narrow link (link 1) and tight link (link 3) on a path.

This paper then focuses on active probing tools estimating *AvB*, as well as the ones estimating the capacity. This last value is important for estimating *AvB*, as explained in equation (1). All these tools use theoretical network properties that are described in the already quoted literature. We do not describe here existing techniques of active probing tools as many papers already present this state of the art [4], [8] and provide taxonomies of these tools according to their measurement techniques. According to these taxonomies, the tools belong to the four following families:

- Variable Packet Size (VPS) probing which estimates the capacity of individual hops (Examples: Clink, Pchar, Pathchar, Bing [13]).
- Packet Pair/Train Dispersion (PPTD) which estimates end-to-end capacity (Examples: Abing, Spruce, Pipechar [15], bprobe [17], cprobe [17], Pathrate [16], sprobe [14]).

- Self-Loading Periodic Streams (SLoPS) which estimates end-to-end available bandwidth (Examples: Pathchirp, IGI, Pathload).
- Trains of Packet Pairs (TOPP) which estimate end-to-end available bandwidth [5].

For more details about active probing techniques, the reader can look at [8]. A short taxonomy of tools evaluated in this paper is proposed in table 1. It gives for each tool, the name, author, version (Release), technique (Methodology), protocols, some interesting characteristic, the target (path or link), the need of root privileges, operating systems (L: Linux, B: BSD, Su: Sun, So: Solaris, I: Irix, F: FreeBSD, N: NetBSD, O: OpenBSD, A: AIX) and the number of hosts required (sender only (S) or, sender and receiver (S & R)).

Table 1: Taxonomy of evaluated tools.

Name	*Clink*	*Pchar*	*Pathchar*
Authors	*Downey*	*Mah*	*V. Jacobson*
Release	*1-0*	*1-4*	*2-0-30*
Methodoloy	*VPS*	*VPS*	*VPS*
Protocol	*UDP*	*UDP, ICMP*	*UDP, ICMP*
Characteristic	*Bandwidth*	*Bandwidth*	*Bandwidth*
Path/Link	*Link*	*Link*	*Link*
Root	*yes*	*yes*	*yes*
OS	*L,B,Su*	*L,So,I,F,N,O*	*F,N,O,L,So*
Host	*S*	*S*	*S*

Name	*Abing*	*Spruce*	*Pipechar*	*Pathchirp*	*IGI*
Authors	*Navratil*	*Strauss*	*Guojon*	*Ribeiro*	*Hu*
Release	*2-1-4*	*0-2*	*2K1*	*2-3-7*	*1-0*
Methodoloy	*Packet Pair*	*Packet Pair*	*Packet Train*	*SLoPS*	*SLoPS*
Protocol	*UDP*	*UDP*	*UDP*	*UDP*	*UDP*
Characteristic	*AvB*	*AvB*	*AvB*	*AvB*	*AvB*
Path/Link	*Path*	*Path*	*Link*	*Path*	*Path*
Root	*no*	*no*	*yes*	*no*	*no*
OS	*L*	*L,B*	*L,So,I,F,A*	*L,F,Su,A,I*	*F,L,Su*
Host	*S & R*	*S & R*	*S*	*S & R*	*S & R*

3. Evaluation Methodology

The evaluation of these tools is being performed on the Metropolis monitoring and measurement platform. Metropolis is a project, funded by the French Network for Research in Telecommunications (RNRT) which federates most of the public research on network monitoring and metrology in France. In terms of measurement techniques, Metropolis aims at combining active and passive measurements to get the advantages

of both techniques. The monitoring and measurement platform has been designed and developed, and it is now deployed on the Renater network (the French network for education and research). This platform consists of:

- Some passive microscopic monitoring probes based on the famous DAG card (designed and provided by the University of Waikato and Endace society in New Zeeland) [1];
- Some active measurement equipments: These active probes rely on the RIPE boxes that have been extended to support the NIMI software, as well as a self designed and developed measurement software environment called MetroMI (derived from the NIMI's one). The Metropolis monitoring and measurement platform is depicted on the Figure 2.

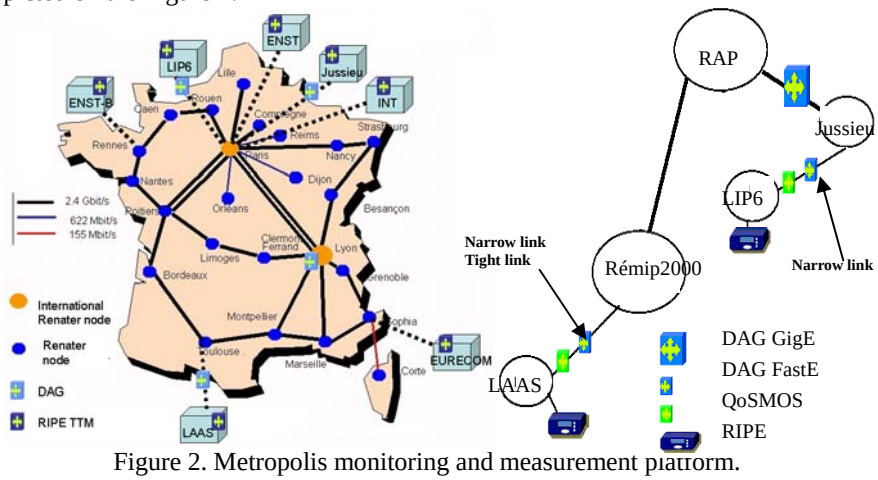

Figure 2. Metropolis monitoring and measurement platform.

The evaluation procedure under consideration in this paper relies on the use of passive measurements for evaluating the accuracy of active probing tools. We run the tools under concern for estimating the available bandwidth, and compare their results with the measurements made with the DAG based monitoring systems. Indeed, we are able to collect traffic traces at the edge of the Renater network, and then to capture the traffic going to and from the core network, including the probe traffic. In addition, the core network is largely over provisioned: links have several Gbps capacity and are never charged more than 50 %. At the opposite, access links are Fast Ethernet (100 Mbps) or GigE (1Gbps) links. FastEthernet as well as GigE links are then the narrowest links on the paths between any of the RIPE equipments. These narrow links are then monitored using the DAG systems. It is important to note, and this is the proof for our evaluation methodology, that DAG systems are perfectly provisioned for capturing the traffic and very accurate: packet headers are extracted on the fly and timestamped with a GPS clock on the dedicated DAG hardware, and the resulting sample is stored on the hard drive of the machine after crossing an overprovisioned bus. This system is then insensitive to network load compared to some software dump

solutions (as tcpdump for example). It is also important to mention the accuracy of the timestamp, the clock of the DAG system being synchronized on a GPS signal. Our RIPE boxes are also synchronized on a GPS signal, meaning that all the monitoring and measurement boxes are perfectly synchronized on the universal temporal reference. It also avoids any temporal drift as boxes resynchronized on the GPS pulse every second. The vendors of the GPS cards ensure less than 2 µs accuracy, what is largely sufficient in our case. The DAG based monitoring system is then an absolute reference and will allow us to evaluate the accuracy of the active measurement tools estimating the available bandwidth on the path from a given source to a given destination. The evaluation results presented in section 4 of this paper have been obtained between LAAS in Toulouse and LIP6 in Paris. LAAS is connected to RENATER with a FastEthernet link, while LIP6 has a GigE link (See Figure 2). Note that these Figures are largely simplified compared to the real complexity of the network between LAAS and LIP6. The detail of links between LAAS and LIP6 as seen by *traceroute* is shown in table 2. This traceroute, as the experiment described in the following, has been run between polka.laas.fr (140.93.192.71) and adonis.lip6.fr (132.227.74.18).

Table 2: Traceroute results

traceroute to 132.227.74.18 (132.227.74.18), 30 hops max, 40 byte packets					
1	*braveheart*	*(140.93.0.75)*	*0.744 ms*	*0.721 ms*	*0.813 ms*
2	*remip-v2*	*(195.83.132.129)*	*0.213 ms*	*0.211 ms*	*0.207 ms*
3	*195.220.57.25*	*(195.220.57.25)*	*0.734 ms*	*0.700 ms*	*0.669 ms*
4	*193.52.8.1*	*(193.52.8.1)*	*0.951 ms*	*0.984 ms*	*0.884 ms*
5	*193.55.105.238*	*(193.55.105.238)*	*1.353 ms*	*0.880 ms*	*0.993 ms*
6	*toulouse-g3-1.cssi.renater.fr*	*(193.51.181.178)*	*1.137 ms*	*1.086 ms*	*1.008 ms*
7	*bordeaux-pos2-0.cssi.renater.fr*	*(193.51.180.13)*	*12.025 ms*	*11.524 ms*	*11.608 ms*
8	*poitiers-pos1-0.cssi.renater.fr*	*(193.51.179.253)*	*11.611 ms*	*12.234 ms*	*12.103 ms*
9	*nri-b-pos4-0.cssi.renater.fr*	*(193.51.179.133)*	*11.969 ms*	*12.201 ms*	*11.798 ms*
10	*jussieu-pos4-0.cssi.renater.fr*	*(193.51.180.157)*	*11.469 ms*	*11.753 ms*	*15.453 ms*
11	*193.50.20.73*	*(193.50.20.73)*	*15.667 ms*	*11.702 ms*	*11.301 ms*
12	*jussieu-rap.rap.prd.fr*	*(195.221.127.182)*	*11.962 ms*	*11.680 ms*	*11.463 ms*
13	*r-scott.reseau.jussieu.fr*	*(134.157.254.10)*	*13.327 ms*	*13.334 ms*	*15.345 ms*
14	*olympe-gw.lip6.fr*	*(132.227.109.1)*	*12.668 ms*	*13.334 ms*	*12.409 ms*
15	*adonis.lip6.fr*	*(132.227.74.18)*	*12.526 ms*	*13.147 ms*	*12.328 ms*

However, for fully evaluating the accuracy and performances of active probing tools in different network and traffic conditions, we use a traffic generator. Its first ability is to generate a constant traffic at a given rate. By changing this rate, we will emulate networks with different load / capacities, and it will then be possible to evaluate the accuracy of the active tools in different networks proposing a full range of capacities. As we generate only constant traffic, this additional traffic has a limited impact on the

dynamics of the global Internet traffic throughput which keeps the same variations as without the traffic generator. In addition, by generating other traffic models than the constant one, we will also be able to evaluate the accuracy of available bandwidth estimation tools when confronted to cross traffic having very different properties and in particular the ones of current Internet traffic. This would help us to analyze in what conditions these tools are providing accurate and efficient results (or not), and why?
The results presented in section 4 have been obtained following this methodology. The first part of section 4 presents the evaluation results of Clink, Pchar, and Pathchar. It gives the per hop bandwidth estimation. After evaluating the capacity of the links, especially the one of the narrow link, the second part of the section 4 presents evaluation results for the tools estimating available bandwidth: Abing, Spruce, Pipechar, IGI and Pathchirp. As for capacities, this second part presents the end-to-end available bandwidth estimation on a path which has its beginning at LAAS and its end at LIP6. IPERF has been used to generate constant UDP traffic on the LAAS' access link. Destination of IPERF traffic was ENSICA, an engineering school in Toulouse area, thus "reducing" the available bandwidth on the LAAS' access link without impacting the rest of the path from LAAS to LIP6.

4. Results

This section describes the experiments with available bandwidth estimation tools previously discussed (See Table 1). We present results in two steps: We first present the results of bandwidth (capacity) estimation using clink, Pchar and Pathchar, and discussed their estimation error ratio, response time and intrusiveness. We conclude this first experiment with the reliability (and utility) of these tools in networking. We secondly present the results of the available bandwidth estimation at the output of LAAS using Abing, Spruce, IGI, Pathchirp and Pipechar. We discuss their estimation error ratio, response time, intrusiveness and reliability. We compare the available bandwidth estimation with DAG measurements. DAG measurements will also give informations (probing time, amount of traffic generated by each tool) to calculate intrusiveness. All tools have been run more than 600 times to get consistent results.

4.1 Evaluation of Clink, Pchar and Pathchar

Clink, Pchar and Pathchar have been used for estimating the bandwidth for every link of the path between LAAS and LIP6 (15 hops in the path). We are especially interested by the estimation bandwidth for the LAAS output, normally a fast Ethernet link (100 Mbps). Figures 4 (a)-(b) describe the bandwidth estimation without any Iperf cross traffic and with a 50 Mbps Iperf cross traffic respectively. One can observe that all these tools produce a bandwidth estimate far from the actual value. Clink proposes three values of the bandwidth: a low one, a high one and a best supposed one. Clink and Pathchar are approximately constant but these tools overestimate the bandwidth (case without Iperf traffic). At the same time, Pchar is very unstable. And, it presents unrealistic disruptions (when there is no Iperf cross traffic). With a 50

Mbps Iperf cross traffic, these three tools propose different estimations but the conclusion is similar: none of the tools produce good values. The capacity measured by Clink is negative for its three values. Pchar and Pathchar most of the time crashed.

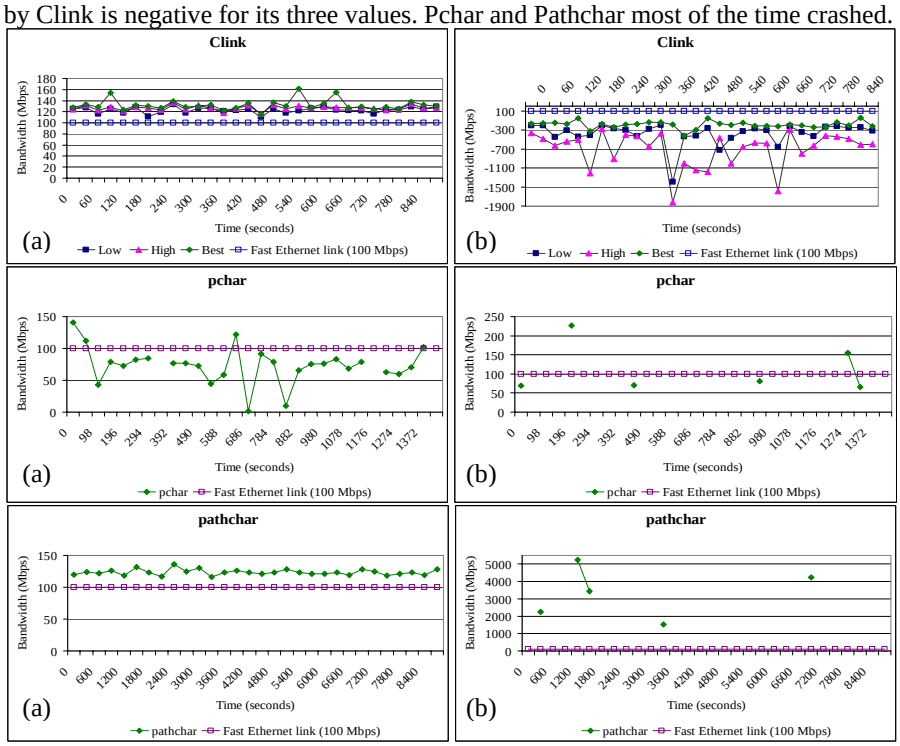

Figure 4: Bandwidth estimation: no Iperf cross traffic (a) – with 50 Mbps Iperf cross traffic (b).

The preceding experiments show very bad estimation results at least when confronted to two cases of cross traffic. Figure 5 (a) then extends these results by showing the bandwidth estimation for many cross traffic values ranging from 0 to 100 Mbps (every 5 Mbps). For each value of the cross traffic, 30 experiments have been run. Figure 5 (a) presents the estimation average for each of these cases, when possible. Indeed, the average for Pchar is not exploitable because results from one experiment to the other differ so much that it is not possible to get a useful estimate. For some values of cross traffic, Pchar also crashes most of the time, thus making the computing of an average impossible. Identically, Clink bandwidth estimation values (Low, High, Best) are very far from the real values: The best-supposed value appears as the worst estimate. Figure 5 (b) shows the estimation error rate which confirms the inaccuracy of Clink and Pathchar.

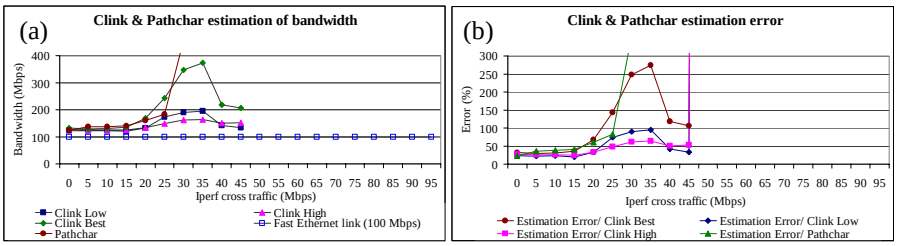

Figure 5: Bandwidth estimation average (a) and error (b).

The preceding experiments clearly demonstrate that the tools are not accurate, reliable and robust. We nevertheless require a method for having an acceptable estimate of the narrow link bandwidth on the path from LAAS to LIP6. We then designed a very basic tool based on the use of 4 *ping* – *StupPing* – to compute a rough estimate of the bandwidth of the narrow link. Given the limited effort spent for designing this almost "stupid" tool, we do not consider it as a possible contribution in this research area. In addition, it should work only in our specific case for which the narrow link on the path is the closest one from the probing source. The principle of StupPing is illustrated on our specific case, i.e. for estimating bandwidth between LAAS (Braveheart) and REMIP (Remip-v2). The first requirement for using StupPing is to get the list of routers and links which will be crossed between the source and destination: this can be obtained using *traceroute*, for example. Then, the StupPing process uses *ping* four times. First, ping the near end of a link with two different packet sizes. Next, use the same two packet sizes to ping the far end of the link. Let us call P_l and P_s the largest and smallest packet sizes (in bytes), $T1_l$ and $T1_s$ the *ping* times for the largest and smallest packets to the nearer interface (in seconds), and $T2_l$ and $T2_s$ the *ping* times for the largest and smallest packets to the distant interface (also in seconds). Finally, the difference $(T2_l - T2_s) - (T1_l - T1_s)$ represents the amount of time to send the additional data over the last link in the path. Therefore, the formula for bandwidth estimation is:

$$C = 16(P_l - P_s)/((T2_l - T2_s) - (T1_l - T1_s)) \quad (4)$$

Figures 6 (a)-(b) and 7 (a)-(b) show the results got with StupPing. This 5-lines binary program computes a bandwidth near the actuel capacity when there is no IPerf cross traffic. The average of the bandwidth estimation is around 92 Mbps, what is quite close from the actual value. As for other tools, when there is IPerf cross traffic (higher than 35 Mbps), this binary program is out of range. Finally, StupPing performs better than other tools when cross traffic is less than 35 Mbps. Another advantage of this tool is that it does not require the root privileges.

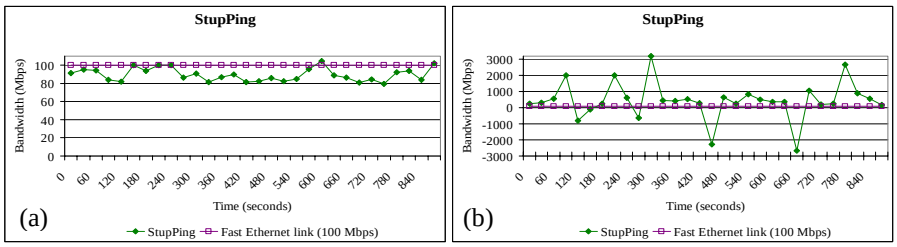

Figure 6: Bandwidth estimation: no Iperf cross traffic (a) – with 50 Mbps Iperf cross traffic (b).

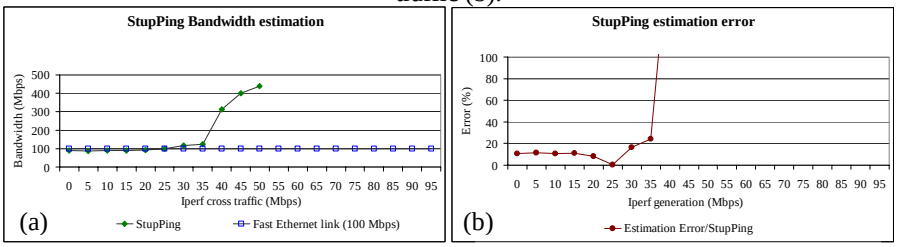

Figure 7: StupPing bandwidth estimation average (a) and error rate (b).

Table 3 summarizes the results got with the probing tools for intrusiveness, and response time, analyzed thanks to the DAG card. We define the “Laas response time” as the time for estimating the bandwidth on the link between Braveheart and Remip-v2 (as this link is the first on the path between LAAS and LIP6, bandwidth estimation tools first provide the bandwidth estimation for this link, and then successively the bandwidth estimation for the following links). In all the tools analyzed, there are some differences in response and probing time, as well as in the probing traffic amount. Clink and StupPing have short (laas)response time (Figure 8). Pchar is also quite fast but Pathchar takes 5 minutes (300 s) to provide the result for the Braveheart and Remip-v2 link. The probing time also varies a lot for Clink, Pchar and Pathchar. It is quite constant for StupPing which only evaluates the bandwidth for one link (LAAS- Remip-v2). But all other tools generate a lot of packets: Pathchar which takes more than 37 minutes (2232 s) for the probing time generates 110,3 Mbits per estimation, Clink around 54,3 Mbits and Pchar 6,9 Mbits. StupPing is the less intrusive tool with 0,13 Mbits sent. We finished this evaluation description with the intrusiveness. For all these four tools, results are good as they appear as lowly intrusive (less than 0,06% of additional probing traffic compared to actual operational traffic). StupPing shows an intrusiveness less than 0,01% (thanks to few ping and ICMP echos). Table 3 nevertheless points out that Clink and Pathchar generates lot of traffic even if their intrusiveness is low. But this means that the probing and response times are consequently too long.

Few concluding remarks for these first experiments:

- Most of these tools either rely on ICMP or UDP probe packets. Such ICMP pac kets are useful for determining information about a network like its capacity, R

TT, loss, etc. But one of their biggest drawbacks is that normal users are not permitted to generate ICMP packets: root privileges are most commonly required.

- ICMP packet can overflow some hosts / routers if not used carefully (ICMP flooding).
- These tools are somewhat basic at this point, slow, not robust and not accurate. Dealing with accuracy of the results, StuPing is the only one to provide a correct estimation of the bandwidth (without overestimation), but of course, just for the closed link of the path. And when cross traffic increases (because of Iperf in our experiments), these active tools crash, this exhibiting the limitations of the VPS probing technique.

Table 3: Evaluation results

	Clink	*Pchar*	*Pathchar*	*StupPing*
Response time (s)	*1112*	*354*	*2400*	*16*
Probing Time (s)	*980*	*318*	*2232*	*14,3*
Laas response time (s)	*21*	*49*	*300*	*16*
Probing traffic Amount (Mbits)	*54,3*	*6,92*	*110,31*	*0,132*
Intrusiveness (%)	*0,0583*	*0,0217*	*0,05202*	*0,00974*

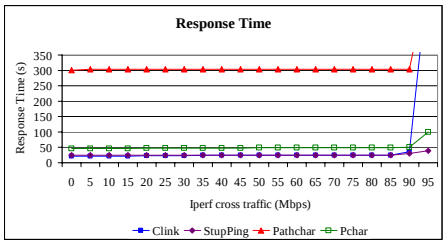

Figure 10: Laas Response times

4.2 Evaluation of Abing, Spruce, Patchirp, IGI and Pipechar

The first tools evaluated were supposed to give the bandwidth of the link between LAAS and Remip: we concluded that none of the tested tools produce accurate enough results. We just have a good estimation with StupPing. This part now focuses on tools able to estimate the available bandwidth on the path between LAAS and LIP6, i.e. on the link between LAAS and Remip. Figures 9 (a)-(b) describe respectively the available bandwidth estimation without cross traffic and with a 50 Mbps IPerf cross traffic, for Abing, Spruce and Pipechar (which use the same probing technique). We describe at the end of this section some experiments with IGI and Pathchirp, two tools using the SLoPS probing methodology. Results on Figure 9 (a) show an inaccurate available bandwidth estimation, with Abing and Spruce (when no Iperf traffic is generated). In this case, these tools underestimate the available bandwidth. Moreover, Abing provides unstable estimations. On the other side, when IPerf generates 50 Mbps of cross traffic, estimation results are better (See Figure 9 (b)). The third tool –Pipechar– gives good estimations in both cases (See Figures 9 (a)-(b)). The major drawback of this tool is that it crashes very often (Figures 9 (a)-(b) and 10 (b)). Given these first good results when evaluating these tools with 50Mbps IPerf cross traffic, Figure 10 (a) presents the estimation results obtained with the three tools when IPerf cross traffic is ranging from 0 to 100 Mbps (for each value, each tool

has been run 30 times). Figure 10 (a) shows the average for the 30 estimations for each tool and each IPerf traffic level. Figure 10 (b) presents the related estimation error. Note that we got the same results with Spruce and Abing: this was expected as both tools use the same technique (packet pair). However, the hypothesis of Spruce which assumes that there is only a narrow link on the path, and that the narrow link is also the tight link is very strong. This can explain why we got bad estimation results. And it is difficult to conclude for Pipechar accuracy as it crashes very often (and also needs root privileges). Table 4 and Figure 11 summarize the results we got with the 3 considered tools for response times and intrusiveness:

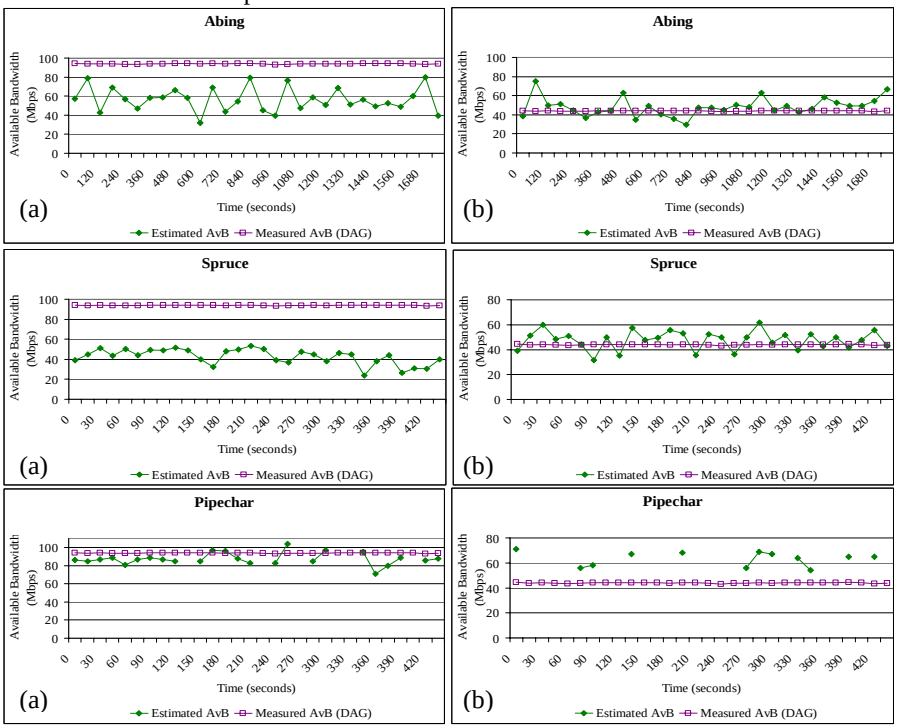

Figure 9: Available bandwidth estimation: no Iperf cross traffic (a) - with 50 Mbps Iperf cross traffic (b).

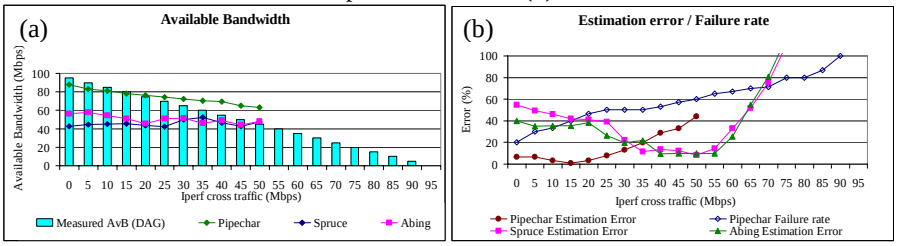

Figure 13: Available Bandwidth estimation average (a), error and failure rate (b).

Table 5: Evaluation results

	Abing	*Spruce*	*Pipechar*
Response time (s)	*1,1*	*11*	*65*
Probing Time (s)	*0,96*	*9,8*	*41,2*
Laas response time (s)	*N/A*	*N/A*	*65*
Probe traffic Amount (Mbits)	*0,464*	*2,34*	*38,5*
Intrusiveness (%)	*0,509*	*0,251*	*0,286*

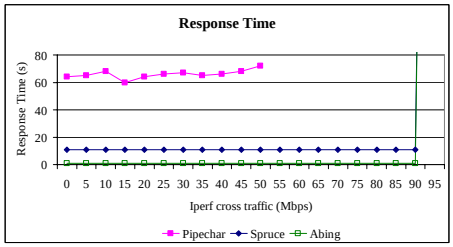

Figure 11: Response times

It then appears that these tools are not very intrusive (less than 0,6%) and have short response time. But the available bandwidth estimations are not good, except for Pipechar with a low level of cross traffic. We conclude these experiments with some evaluation of IGI and Pathchirp (with 50Mbps of IPerf cross traffic), shown on Figure 12 (a)-(b). It appears that Pathchirp overestimates the available bandwidth and IGI is unstable (but its average is not really far from the actual value).

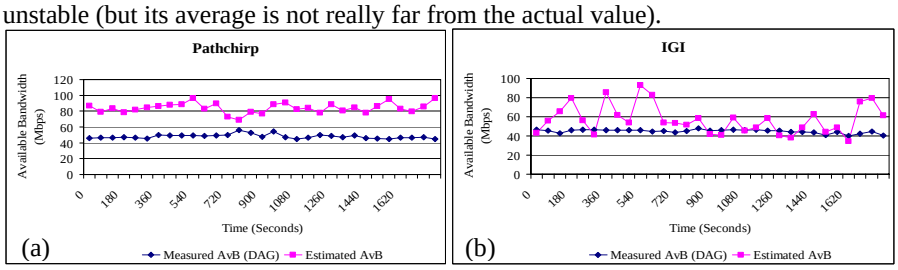

Figure 12: PathChirp (a) and IGI (b) (With 50Mbps IPerf traffic (UDP)).

5. Conclusion

In this paper, we presented an evaluation of active probing tools for estimating capacity and available bandwidth on a link/path in a real Internet environment. The tools which have been evaluated are Clink, Pchar and Pathchar for estimating link capacity, and Abing, Spruce, Pipechar, Pathchirp and IGI for available bandwidth on a path. These experiments show bad results for all these tools in real environment. In addition, most of the tested tools hugely overestimate bandwidth. Such overestimation is really dangerous. For example, for a rate based congestion control, using PathChirp estimations will cause huge congestion phenomena. It would be safer to underestimate the available bandwidth. But, functions using such results would not be optimal. In addition, these tools are not robust enough especially when there is cross traffic. Given the bad results got while cross traffic was constant, we even did not evaluate them with a highly variable traffic. But we can guess that the results would not be very good given the large response time of these tools: there is a level of magnitude between the variation rate of current Internet traffic and response time of these active probing tools. As a conclusion, we are not convinced by any of these tools.

ACKNOWLEDGMENT

We would like to thank C. Nicolescu for his helpful comments and assistance. We would also like to thank ENSICA and LIP6 for authorizing us to take advantage of their resources in our experiments.

References

[1] J. Cleary, S. Donnely, I. Graham, A. McGregor, M. Pearson, "Design principles for accurate passive measurement", *Passive and Active Measurements (PAM) Workshop*, Hamilton, New Zealand, April 2000.

[2] N. Hu, P. Steenkiste, "Evaluation and Characterization of Available Bandwidth Probing Techniques", *IEEE Journal on Selected Areas in Communication*, No 21, 2003.

[3] M. Jain, C. Dovrolis, "Pathload: A measurement tool for end-to-end probing and available bandwidth", *proceedings of Passive and Active Measurement workshop (PAM)*, 2002.

[4] M. Jain, C. Dovrolis, "End-to-End Available Bandwidth: Measurement Methodology, Dynamics, and Relation with TCP Throughput", In: *ACM SIGCOMM*, Pittsburgh, PA, 2002.

[5] B. Melander, M. Bjorkman, P. Gunningberg, "A new end-to-end probing and analysis method for estimating bandwidth bottlenecks", *IEEE Globecom-Global Internet Symposium*, 2000.

[6] V.J. Ribeiro, R.H. Riedi, R.G. Baraniuk, J. Navratil, L. Cottrell, "PathChirp: Efficient Available Bandwidth Estimation for Network Paths", *Passive and Active Measurement Workshop*, 2003.

[7] J. Strauss, D. Katabi, F. Kaashoek, "A Measurement Study of Available Bandwidth Estimation Tools", *ACM SIGCOMM Internet Measurement Workshop*, 2003.

[8] R.S. Prasad, M. Murray, C. Dovrolis, K. Claffy: "Bandwidth estimation:metrics, measurement techniques, and tools". IEEE Network Magazine (2003), http://www.cc.gatech.edu/fac/Constantinos.Dovrolis/publications.html.

[9] J. Navratil. "ABwE: A Practical Approach to Available Bandwidth Estimation", PAM 2003, La Jolla, April 2003.

[10] V. Jacobson. "Pathchar -- a tool to infer characteristics of internet paths". April 1997.

[11] A. B. Downey, "Using Pathchar to Estimate Internet Link Characteristics", ACM SIGCOMM, 1999, pp. 222-23.

[12] B. A. Mah. Pchar. http://www.ca.sandia.gov/ bmah/Software/ Pchar/, 2000.

[13] Pierre Beyssac. BING: Bandwidth pING, March 1998. http://www.cnam.fr/reseau/bing.html.

[14] S. Saroiu, P. K. Gummadi, and S. D. Gribble, "*Sprobe: A fast technique for measuring bottleneck bandwidth in uncooperative environments*," Infocom 2002 Technical Conference, 2002.

[15] J. Guojon. Pipechar, http://www-didc.lbl.gov/Pipechar

[16] C. Dovrolis. Pathrate, http://www.pathrate.org.

[17] R. Carter. bprobe - cprobe. http://cs-people.bu.edu/carter/tools/Tools.html.