On the performance of bandwidth estimation tools*

Alessio Botta, Antonio Pescapé, and Giorgio Ventre†

Abstract

In this paper we first present a complete survey on available bandwidth estimation tools. More precisely, according to the classification proposed on IEEE Network by Prasad et al. in 2003 we categorize them in (i) End-to-End Capacity Estimation Tools; (ii) Available Bandwidth Estimation Tools; (iii) TCP Throughput and Bulk Transfer Capacity Measurement Tools. After a complete survey, we present our proposal, which is based on an integration of previous tools, aiming to improve a fitness function composed of accuracy and total time of estimation. Experimental results are given with respect to comparative analysis between our proposal and existing tools, interference analysis, and finally, wireless links.

1 Introduction

Among the different parameters characterizing traffic and networks, one of the most significant is the available bandwidth of network paths. This is due to the important role of available bandwidth in traffic engineering algorithms as well as in other scenarios like file sharing, server selection, and in general network aware applications.

Several methods and tools aiming to measure the available bandwidth (AB) have been presented in literature. Each of them presents pros and cons. This paper presents a complete and up to date survey on available bandwidth tools. Based on the analysis of the current proposals, we introduce a new tool called Bandwidth Estimation Tool (BET). BET represents an efficient combination of existing techniques implemented by taking into account several performance issues. Based on our experience we present a comparative analysis among BET and other well know tools ($Pathload$, $Patchirp$) with respect to total time of estimation and measure accuracy. Also, as for the experimental analysis, in addition to a comparative analysis we present other two kinds of measurements. First, we expose a performance analysis in which we measure the interferences between available bandwidth tools and other applications running over the same machine. Second, we present a brief analysis of the behavior of available bandwidth tools over wireless links. As for the interference analysis, to the best of our knowledge there are no available studies that present the effects of concurrent running applications on the available bandwidth measurement equipment.

The rest of the paper is organized as follows. In Section 2 we present a survey of available bandwidth estimation tools. Section 3 introduces our proposal, called Bandwidth Estimation Tool (BET), and its architectural details. In Section 4 the results of our performance analysis are described whereas in Section 5 a summary of results is presented.

2 A survey on available bandwidth tools

This Section contains an exhaustive and up to date survey on available bandwidth estimation tools. With such survey we want to give a contribution aiming to illustrate the state of the art. Therefore, our aim is not to make a comparison among all the tools we cite, we would rather make a clear description of the available tools. After the survey, we present a short depiction of our proposal (for deeper details refer to [34]) and make a comparison between it and mostly used applications.

Many up to date on-line taxonomies of network measurement tools are maintained. Among them we suggest [3] and [4]. To provide a clear and complete classification of available bandwidth estimation tools we adopt the guidelines indicated in [1] and we divide related works in the following categories: (i) End-to-End Capacity Estimation Tools; (ii) Available Bandwidth Estimation Tools; (iii) TCP Throughput and Bulk Transfer Capacity Measurement Tools.

2.1 End-to-end capacity estimation tools

$Pathchar$ [11] was the first tool to implement *Variable Packet Size* probing, opening the area of bandwidth estima-

*This work has been partially supported by the Italian Ministry for Education, University and Research (MIUR) in the framework of the WEB-MINDS FIRB Project, by Regione Campania in the framework of "Centro di Competenza Regionale ICT", and finally by the E-NEXT IST European project.

†A. Pescapé and G. Ventre are with the Dipartimento di Informatica e Sistemistica, Università di Napoli "Federico II", Naples, Italy. {pescape,giorgio}@unina.it. A. Botta is with the Consorzio Interuniversitario Nazionale per l'Informatica, Naples (Italy), {abotta}@napoli.consorzio-cini.it.

tion research. The main idea is to measure the data transmission time on each link. This is done by computing the difference between the RTTs from the source to two adjacent routers. To filter out measurement noise $Pathchar$ needs to send a large number of probing packets, identifying the smallest RTT values for the final calculation. As a result, it has a large probing overhead. $Clink$ [15] is an open source tool to perform *Variable Packet Size* probing. The original tool runs only on Linux. $Clink$ differs from $Pathchar$ by using an even-odd technique to generate interval capacity estimates. Also, when encountering routing instability, $Clink$ collects data for all the paths it encounters until one of the paths generates enough data to yield a statistically significant estimate. $Pchar$ [12] is another open source implementation of *Variable Packet Size* probing. Libpcap is used to obtain kernel-level timestamps. $Pchar$ provides three different linear regression algorithms to obtain the slope of the minimum RTT measures against the probing packet size. $Cartouche$ [16] uses a packet train that combines packets of different sizes and exploits differences in how different-sized packets are handled to measure the bandwidth for any segment of the network path. $Sprobe$ [33] is a capacity estimation tool that provides a quick capacity estimate. To measure the source to destination path capacity, $Sprobe$ sends a few packet pairs (normally TCP SYN packets) to the remote host. The remote host replies with TCP RST packets, allowing the sender to estimate the packet pair dispersion in the forward path. $Nettimer$ [19] is a capacity estimation packet pair tool. $Nettimer$ uses a statistical technique called *kernel density estimation* to process packet pair measurements. A kernel density estimator identifies the dominant mode in the distribution of packet pair measurements without assuming a certain origin for the bandwidth distribution, overcoming the corresponding limitation of histogram-based techniques. $CapProbe$ [27] relies on packet pair dispersion technique. It is based on the consideration that both compression and expansion of packet pair dispersion are the consequence of queuing due to cross-traffic. Therefore, it estimates the end-to-end capacity by filtering out queuing effects from packet pair samples.

2.2 Available bandwidth estimation tools

$Pathload$ [22] uses the network self congestion paradigm for available bandwidth estimation. Network congestion is obtained by sending periodic streams (flows of packets) with increasing bit rates. When the trend the packets' one way delays is found to be increasing the congestion is detected and the available bandwidth is evaluated. IGI [7] in order to measure the available bandwidth sends a train of probing packet pairs in quick succession. Competing traffic packets may be inserted between them, thus modifying the gaps. As a result, the received packets' gaps depend on competing traffic rate, which can be, therefore, estimated. IGI computes the available bandwidth by subtracting the estimated amount of competing traffic from the bottleneck link capacity. $Spruce$ [17] is based on the Probe Gap Model (PGM). Like other PGM tools [7], [8], Spruce assumes a single bottleneck link In [23] $netest$ is introduced. It is meant to measure available bandwidth and achievable throughput in a minimally intrusive fashion. Furthermore, it provides information useful to improve achievable throughput of TCP applications. Work [25] presents a tool called $Bprobe$. In the work the authors present the tool as follows. "*The essential idea behind the probe is this: if two packets can be caused to travel together such that they are queued as a pair at the bottleneck link, with no packets intervening between them, then the inter-packet spacing will be proportional to the processing time required for the bottleneck router to process the second packet of the pair*". $Probegap$ [28] has been developed for overcoming difficulties found by other tools on broadband access network. It has been tested over a number of network scenarios that include cable modem and 802.11-based wireless network. $Pathmon$ [29] uses a two step measurement. The first step measures jitter and allows for statistical analysis of network delay. The second step calculates the delay measure using cumulative packet delay intervals. $Pathchirp$ [24] uses the self inducted congestion paradigm. It sends exponential flight pattern of probes (called a chirp) for causing the self induced congestion on the network. By rapidly increasing the probing rate within each chirp, $Pathchirp$ obtains a rich set of information from which it dynamically estimates the available bandwidth. In [31] the authors propose a new sampling method of available bandwidth called *ab-probe*. The *ab-probe* method uses an intuitive model that helps to understand and correct the error introduced by other methods. Furthermore, they theoretically compare the new model with previous ones, exploring their differences, observability and robustness. A tool that is also able to locate the tight link – i.e. the link with less available bandwidth than all the links preceding them on end-to-end network paths – is Spatio-Temporal Available Bandwidth estimator ($STAB$) [32]. The tool uses special chirp-probing trains, featuring an exponential flight pattern of packets, which have the advantage of employing few packets while giving an accurate estimate of available bandwidth.

2.3 TCP throughput and bulk transfer capacity measurement tools

$Iperf$ [5] is a tool that computes the end-to-end available bandwidth on the basis of the following figures: number of bytes sent and time elapsed for the packets to travel from the source to the destination. It is a benchmarking tool.

In the same class there are $TTCP$ [14] and $NetPerf$ [13]. These benchmarking tools use large TCP transfers to measure the achievable throughput in an end-to-end path. The user can control the socket buffer sizes and thus the maximum windows size for the transfer. $NetPerf$ and $Iperf$ with respect to $TTCP$ have improved the measurement process and can handle multiple parallel transfers. These three tools require access at both ends of the path, but do not require superuser privileges. On the web site [26], the tool called $Bing$ is presented. It is a point-to-point bandwidth measurement tool (hence the 'b'), based on ping. $Bing$ determines the real (raw, as opposed to available or average) throughput on a link by measuring ICMP echo requests round trip times for different packet sizes for each end of the link. $Cprobe$ [25] was the first tool to attempt to measure end-to-end available bandwidth. $Cprobe$ measures the dispersion of a train of eight maximum-sized packets. In [20] and [21] it has been shown that the dispersion of long packet trains measures the dispersion rate, which is not the same as the end-to-end available bandwidth. In [30] the authors present $Pathneck$, a tool that allows end users to efficiently and accurately locate the bottleneck link on an Internet path. $Pathneck$ is based on a novel probing technique called Recursive Packet Train (RPT) and does not require access to the destination. $BFind$ [10] detects the bottleneck position by injecting a steady UDP flow into the network path, and by gradually increasing its bit rate to amplify the congestion at the bottleneck router. $ABwE$ [18] is based on measuring packet pair dispersion and is designed for measuring available bandwidth on high-speed links (up to 1 Gb/s). $ABwE$ uses packet pairs with a fixed size and initial delay between packets. Conversion from the receiver measured delay to available bandwidth is based on empirically taken lengths of the cross traffic packets which could be expected on the links. This value is normalized to obtain the throughput which is achievable for TCP traffic.

3 Bandwidth Estimation Tool (BET)

After the analysis of previously cited tools we found a necessity of sound and complete bandwidth estimators. We proposed *Bandwidth Estimation Tool* (BET), that represents an effective combination of different techniques aiming to exploit the positive aspects of each of them. More precisely, BET integrates (in a cascade architecture) the following different measurement techniques: (i) '*packet train dispersion*' [9] used for capacity estimation; (ii) an efficient combination of the '*Self Loading of Exponential Chirp*' (SLoEC) [24] and '*Self Loading of Periodic Stream*' (SLoPS) [6] used for available bandwidth estimation. With such an architecture the measurement process is realized thanks to a successive refinements fashion. This allows a reduction of total measurement time and, consequently, of the total amount of probe traffic. Furthermore, in order to improve measure accuracy two techniques are introduced: i) the use of *signal handlers* for precise timing of packets sending process; ii) a *kernel level timestamping* for received packets. In Figure 1 the main modules constituting the BET architecture are summarized. They are four, with one that controls and coordinates the operations of the remaining ones, which are responsible for the measurement process. In [34] BET architecture, with a clear description

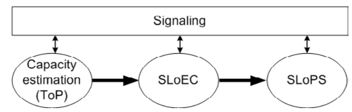


Figure 1. Modules of BET

of each module as well as a deep experimental validation is presented.

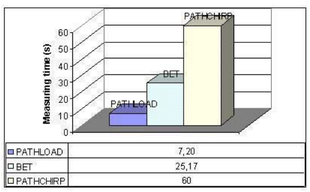
4 Experimental Analysis

This Section presents three different types of experimental analysis. First we present a comparative evaluation of some analyzed bandwidth tools, including a comparison with BET with respect to measure accuracy and total measurement time. Second, an interference analysis between the measurement tool and other applications running on the same machine is presented. Finally, we show the behavior of these tools over a wireless link. In Figure 3 the testbed used for both comparative and interference analysis is depicted. In particular in such Figure it is possible to see that the probe traffic (continuous line) was in all cases sent by *Aglaope* to *Partenope*.

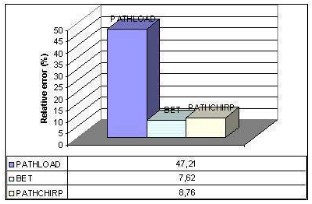
4.1 Comparative analysis

In this analysis we compare BET to two widely spread available bandwidth estimation tools: $Pathload$ and $Pathchirp$. The tools were confronted with respect to achieved accuracy and total time used to complete the measurement process. They were tested with different cross traffic (CT) profiles at different bit rates. Indeed, thanks to the use of a powerful traffic generator D-ITG (Distributed Internet Traffic Generator) [35] we were able to use a large number of random variables to profile packets' Inter Departure Time (IDT) and Packet Size (PS) of the cross traffic. As for $pathchirp$, different results have been otained by varying the combination of input parameters. For a better comparison, in this work only its best results are presented. In Figure 3(a) the controlled testbed used in these tests is shown, the "link under test" is a $100Mbps$ Ethernet.

In this paper we present a comparative analysis with two types of cross traffic. In [34] a more detailed analysis with several cross traffic patterns is exposed. As for this paper, the first cross traffic type (**type1**) has been generated with constant PS and exponentially distributed IDT. The second one (**type2**) has been produced by using constant PS and Poisson profiled IDT. Furthermore, for every cross traffic profile we used three different bit rates: 10, 50 and $90Mbps$. Also, we performed 10 test repetitions for every experiment and we averaged the obtained results in order to minimize the influence of random error on measured values. Figure 2



(a) Measurement time



(b) Relative error of AB measure

Figure 2. Comparative analysis with constant PS, Exponential IDT, and bit rate equal to $50Mbps$.

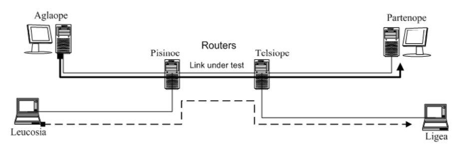
shows results obtained with **type1** cross traffic and bit rate equal to $50Mbps$. In particular, in such Figure the total measurement time and the relative error of measured available bandwidth are depicted ($RE = \frac{EV-MV}{EV}$ where RE is the relative error, EV is the expected value, and MV is the measured value). In such case BET achieved the best performance in terms of relative error, therefore of measure accuracy. Due to space constrain we can not provide graphical results of all the experiments of the comparative analysis. Therefore, all the other results are summarized in Table 1. In particular, with **type1** cross traffic and bit rate equal to $10Mbps$ both BET and $Pathchirp$ achieved the best performance in terms of measure accuracy. Instead, when cross traffic rate was equal to $90Mbps$ the best performance in terms of accuracy was achieved by $Pathchirp$ only. When we generated cross traffic according to **type2**, and bit rate equal to $50Mbps$, the three applications obtained the same performance in terms of accuracy. With cross traffic rate equal to $10Mbps$ BET and $Pathload$ achieved the best performance in terms of measure accuracy. Finally, when the cross traffic rate was $90Mbps$ the best performance in terms of accuracy was achieved by $Pathchirp$. Based on our practical experience it is difficult to find an available bandwidth estimation tool better than all others in all network conditions. At opposite side, we are able to understand the behavior of the widely used tools with different traffic patterns. We prove that BET represents a good compromise among all variables in a fitness function that considers accuracy and total estimation time.

Table 1. Experimental results of the comparative analysis with different traffic profiles

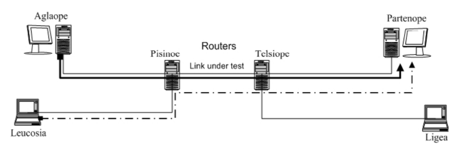
CT (Mbps)	**PS**	**IDT**	Parameter	**BET**	**Path Load**	**Path Chirp**
90	CONST	EXP	**AB** (Mbps)	3.8	0.08	9.40
			RE (%)	62.00	99.20	6.02
			TMT (s)	58	87.89	10
50	CONST	EXP	**AB** (Mbps)	53.81	26.40	45.62
			RE (%)	7.62	47.21	8.76
			TMT (s)	25.17	7.2	60
10	CONST	EXP	**AB** (Mbps)	91.18	89.33	81.14
			RE (%)	1.31	0.75	9.84
			TMT (s)	25	9.08	30
90	CONST	POIS	**AB** (Mbps)	5.27	3.71	9.37
			RE (%)	47.35	62.93	6.27
			TMT (s)	51	34.83	10
50	CONST	POIS	**AB** (Mbps)	52.1	51.66	51.52
			RE (%)	4.20	3.32	3.04
			TMT (s)	22.12	4.93	30
10	CONST	POIS	**AB** (Mbps)	101.2	102.37	88.66
			RE (%)	12.44	13.74	29.4
			TMT (s)	21.3	5.47	30

4.2 Interference analysis

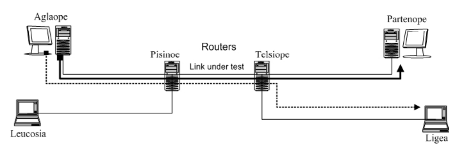
In this section we present results obtained in our experimental *interference analysis*. With such analysis we investigated what happens when other network applications run over the machine used for available bandwidth estimation (at sender or receiver side), that is, when other applications share resources with the measurement tool. This situation is quite common for currently used network applications (i.e. network aware applications, peer-to-peer file distribution and applications, dynamic server selection, etc.). Indeed, it is frequent that modern network applications need an available bandwidth estimate while performing other tasks. This



(a) Comparative analysis



(b) Receiver side interference



(c) Sender side interference

Figure 3. Network Testbeds: probe traffic (continuous line) and cross traffic (discontinuous lines)

analysis has been conducted on the same testbed of the comparative analysis opportunely modified. Indeed, the nominal capacity of the "link under test" has been reduced to $10Mbps$. In such a way the cross traffic generation process was less resource-consuming. To investigate about the interference we divided the experiments in two parts. In the first one we configured our testbed in order to have both probing and cross traffic receivers on the same machine. In the second part of experiments the testbed was modified in order to have both senders of probing and cross traffic on the same machine.

4.2.1 Receiver side interference analysis

This configuration is depicted in Figure 3(b) where the continuous line represents probe traffic, and the dashed-dotted one concurring traffic.

Table 2. Results of interference analysis

Receiver side interference					
CT [Mbps]	0	2	4	6	8
AB [Mbps]	10	8	6	4	2
BET	9.1	8.4	7.5	6.05	1.21
PathLoad	11.13	9.01	7.51	3.11	1.55
PathChirp	9.36	4.66	7.89	5.09	1.19
Sender side interference					
CT [Mbps]	0	2	4	6	8
AB [Mbps]	10	8	6	4	2
BET	8.54	8.54	8.54	8.54	8.75
PathLoad	9.74	9.74	9.05	7.92	12.66
PathChirp	9.74	9.06	4.78	8.15	8.99

In Table 2 the results obtained by this analysis are presented. This Table shows that by using the same machine for receiving both cross and probe traffic i) the measure accuracy is less than the case in which two different receiving machine are used ii) BET presented the least accuracy degradation thus it can be used in such environment.

4.2.2 Sender side interference analysis

In Figure 3(c) the used testbed is depicted. The continuous line represents the probe traffic and the dotted line represents cross traffic. Table 2 presents the results obtained and it shows that in this configuration the performance of the measurement process, in terms of accuracy, are seriously compromised. Indeed, none of the applications was able to estimate available bandwidth with an acceptable accuracy.

4.3 Experimental analysis over wireless links

Available bandwidth estimation over wireless links represents an important challenge to be dealt with. Indeed, over wireless networks, available bandwidth estimation algorithms that step from the assumption of a stationary channel, fail their mission. Moreover, some papers presenting tools that aim to solve this problem do not provide an experimental analysis over wireless network [27] or, when provided [28], the experimental results are not so much satisfying thus indicating a still open issue.

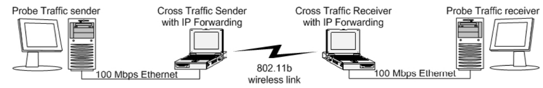


Figure 4. Wireless Scenario Testbed

In Figure 4 the testbed used in this analysis is depicted. To avoid the "store-and-forward" effect of layer 2 devices, the "link under test" is an 802.11b wireless link operating in ad-hoc mode. We worked with a nominal capacity equal to $11Mbps$. Table 3 presents the results of such analysis. This Table shows that, also in this case, none of the three applications was able to estimate the available bandwidth with an acceptable accuracy. Indeed, the measurement results seem to be indifferent to cross traffic rate. This is probably due to the fact that the techniques which the three applications are based on step from the assumption of a stationary channel. This hypothesis is not longer valid for 802.11b wireless links, for which the nominal capacity depends on link quality. For this reason there is need for new tools able to operate in such an environment that includes non stationary channels.

Table 3. Results of analysis on wireless link

CT [Mbps]	0	1	2	4
AB [Mbps]	11	10	9	7
BET	2.42	1.98	1.74	0.34
Pathload	4.84	5.86	5.43	3.35
Pathchirp	2.45	2.01	1.68	1.04

5 Conclusion

In this work we presented a complete survey on available bandwidth estimation tools with a clear distinction between them according to [1]. Furthermore we presented a new tool called BET which performs a measurement of available bandwidth with a successive refinements technique. Also, we presented a three steps experimental analysis which includes a study we called *interference analysis*. The experimental results of the first analysis (comparative) have shown that, in many cases, our application achieves better performance than other ones. But it does not exists '*the best tool for the available bandwidth estimation*'. As for the interference analysis, we have shown that when two applications concur for network resources at receiver side an accurate estimation is still possible. Instead, when the interference is created at sender side none of the considered applications was able to find an estimate with acceptable accuracy. As for the analysis on wireless links, we have learned that the existing applications fail in their mission. In our ongoing work we are studying the behavior of these tools in more controlled testbeds. Indeed, in order to understand the wireless channel fluctuations and the effects of interferences on the measurement process we are working on a testbed located in an anechoic chamber. Furthermore, we are working on a modification to the SLoPS algorithm aiming to improve the performance of the measurement process. We called this modification SAWT ([34]). Finally, we think that even if a great effort has been done other contributions are still necessary.

References

[1] R. Prasad, C. Dovrolis, M. Murray and kc Claffy, "Bandwidth Estimation: Metrics, Measurement Techniques, and Tools", IEEE Network, Nov/Dec 2003 pp. 27-35

[2] kc Claffy, S. McCreary, "Internet measurement and analysis: passive and active measurement", http://www.caida.org/outreach/papers/1999/Nae4hansen/

[3] http://www.caida.org/tools/taxonomy

[4] http://www.slac.stanford.edu/xorg/nmtf/nmtf-tools.html

[5] Iperf, http://dast.nlanr.net/Projects/Iperf

[6] M. Jain, C. Dovrolis, "End-to-End Available Bandwidth: Measurement Methodology, Dynamics and Relation with TCP Throughput" *Proc. of* ACM SIGCOMM, Aug.2002.

[7] N. Hu and P. Steenkiste, "Evaluation and Characterization of Available Bandwidth Probing Techniques", IEEE JSAC, Vol. 21, No. 6, August 2003.

[8] V. J. Ribeiro, M. Coates, R. H. Riedi, S. Sarvotham, and R. G. Baraniuk. "Multifractal cross traffic estimation". *Proc. of* ITC specialist seminar on IP traffic Measurement, September 2000.

[9] R. L. Carter and M. E. Crovella "Measuring bottleneck link speed in packet-switched networks", Performance Evaluation Journal, October 1996.

[10] A. Akella, S. Seshan, and A. Shaikh. "An empirical evaluation of wide-area internet bottlenecks". *Proc. of* ACM IMC, October 2003.

[11] V. Jacobson. "pathchar - a tool to infer characteristics of Internet paths", 1997. Presented as April '97 MSRI talk,

[12] http://www.kitchenlab.org/www/bmah/Software/pchar/

[13] http://www.netperf.org/netperf/NetperfPage.html

[14] http://www.pcausa.com/Utilities/pcattcp.htm

[15] A.B. Downey, "Using Pathchar to Estimate Internet Link Characteristics", *Proc. of* ACM SIGCOMM, Sept. 1999, pp. 222−223.

[16] K. Harfoush, A. Bestavros, and J. Byers. "Measuring bottleneck bandwidth of targeted path segments". *Proc. of* IEEE INFOCOM, April 2003.

[17] J. Strauss, D. Katabi, and F. Kaashoek. "A measurement study of available bandwidth estimation tools" Proc. of ACM IMC, October 2003.

[18] J. Navratil, R.L Cottrell. "ABwE: A Practical Approach to Available Bandwidth Estimation" *Proc. of* PAM 2003, La Jolla, April 2003.

[19] K. Lai and M. Baker. "Nettimer: A tool for measuring bottleneck link bandwidth". *Proc. of* the USENIX Symp. on Internet Technologies and Systems, March 2001.

[20] C. Dovrolis, P. Ramanathan, and D. Moore, "What do Packet Dispersion Techniques Measure?" *Proc. of* IEEE INFOCOM, Apr. 2001, pp. 90514.

[21] B. Melander, M. Bjorkman, and P. Gunningberg, "A New End-to-End Probing and Analysis Method for Estimating Bandwidth Bottlenecks", *Proc. of* IEEE Global Internet Symp., 2000.

[22] M. Jain, C. Dovrolis, "Pathload: a Measurement Tool for Available Bandwidth Estimation", *Proc. of* PAM 2002,

[23] G. Jin, B. Tierney, "Netest: A Tool to Measure the Maximum Burst Size, Available Bandwidth and Achievable Throughput", *Proc. of* The 8th IFIP/IEEE International Symposium on Integrated Network Management March 24-28, 2003

[24] V. Ribeiro, R. Riedi, R. Baraniuk, J. Navratil, L. Cottrell, "PathChirp: Efficient Available Bandwidth Estimation for Network Paths", *Proc. of* PAM 2003.

[25] R. L. Carter, M. E. Crovella, "Dynamic Server Selection using Bandwidth Probing in WideArea Networks", TR BU-CS-96-007, Computer Science Department, Boston University, March 1996.

[26] http://www.freenix.fr/freenix/logiciels/bing.html

[27] R. Kapoor, L.-J. Chen, A. Nandan, M. Gerla, M. Y. Sanadidi, "CapProbe: A Simple and Accurate Capacity Estimation Technique for Wired and Wireless Environments", ACM SIGMETRICS Performance Evaluation Review, Volume 32, Issue 1, June 2004.

[28] K. Lakshminarayanan, V. N. Padmanabhan, J. Padhye, "Bandwidth Estimation in Broadband Access Networks", *Proc. of* Internet Measurement Conference 2004.

[29] D. Kiwior, J. Kingston, A. Spratt, "PATHMON, A Methodology for Determining Available Bandwidth over an Unknown Network", *Proc. of* IEEE Sarnoff Symposium, April 2004.

[30] Ningning Hu, Li (Erran) Li, Zhuoqing Morley Mao, "Locating Internet Bottlenecks: Algorithms, Measurements, and Implications", *Proc. of* SIGCOMM 2004, pp. 41-54.

[31] M. Kazantzidis, D. Maggiorini, and M. Gerla, "Network Independent Available Bandwidth Sampling and Measurement", *Proc. of* Second International Workshop on Quality of Service in Multiservice IP Networks, 2003, pp. 117-130 .

[32] V. J. Ribeiro, R. H. Riedi, and R. G. Baraniuk, "Locating Available Bandwidth Bottlenecks", IEEE Internet Computing, September 2004, pp. 34-41.

[33] S. Saroiu, P. K. Gummadi, S. D. Gribble, "A Fast Technique for Measuring Bottleneck Bandwidth in Uncooperative Environments", *Proc. of* INFOCOM 2002.

[34] A. Botta, S. D'Antonio, A. Pescapé, G. Ventre, "BET: A Hybrid Bandwidth Estimation Tool", IEEE International Workshop on Performance Modeling and Analysis of Communication in Parallel, Distributed, and Grid Networks (PMAC-PDG'05), Fukuoka , Japan, July 2005.

[35] http://www.grid.unina.it/software/ITG