

End-to-End Available Bandwidth Estimation Tools, an Experimental Comparison

Emanuele Goldoni¹ and Marco Schivi²

¹ University of Pavia, Dept. of Electronics,
27100-Pavia, Italy
emanuele.goldoni@unipv.it

² University of Pavia, Dept. of Computer Engineering and Systems Science,
27100-Pavia, Italy
marco.schivi01@ateneopv.it

Abstract. The available bandwidth of a network path impacts the performance of many applications, such as VoIP calls, video streaming and P2P content distribution systems. Several tools for bandwidth estimation have been proposed in the last years but there is still uncertainty in their accuracy and efficiency under different network conditions. Although a number of experimental evaluations have been carried out in order to compare some of these methods, a comprehensive evaluation of all the existing active tools for available bandwidth estimation is still missing. This article introduces an empirical comparison of most of the active estimation tools actually implemented and freely available nowadays. Abing, ASSOLO, DietTopp, IGI, pathChirp, Pathload, PTR, Spruce and Yaz have been compared in a controlled environment and in presence of different sources of cross-traffic. The performance of each tool has been investigated in terms of accuracy, time and traffic injected into the network to perform an estimation.

1 Introduction

Available bandwidth is a fundamental metric for describing the performance of a network path. This parameter is used in many applications, from routing algorithms to congestion control mechanisms and multimedia services. For example, in [1, 2] the authors investigated the importance of the available bandwidth for adaptive content delivery in peer-to-peer (P2P) or video streaming systems.

The easiest and most effective method for estimating the available bandwidth is active probing – a few test packets are transmitted through the path and are used to infer the network status. The problem of end-to-end estimation has received considerable attention and a number of active probing tools have emerged in recent years [3]. Nevertheless, producing reliable estimations still remains challenging: the measurement process should be accurate, non-intrusive and robust at the same time.

Considerable efforts have also been put in comparison projects aiming to analyze the performances of existing methods in different network scenarios.

Nevertheless, many issues remain still unresolved and the quest for the best available bandwidth estimation tool is still open [4].

Compared to previous works, this paper proposes the largest comparison of available bandwidth estimation methods. The performances of 9 different tools are investigated in terms of accuracy, time and intrusiveness. All the experiments have been conducted in a low-cost and flexible testbed which could be easily extended to simulate more complex and realistic network topologies.

The remainder of the article is organized as follows. Section 2 briefly presents related works on measurement tools and past performance comparisons. In Section 3 we describe the testbed and the methodology adopted for the experimental comparison of the tools. Next, Section 4 includes the preliminary results obtained during the performance tests we performed in our laboratory. Finally, the conclusions drawn from this study are presented in Section 5.

2 Background and Related Work

Many software tools for network bandwidth monitoring have been developed in the last years by both independent scientists and collaborative research projects. Although designed for the same purpose, these tools are based on different principles and implement various techniques. This section briefly introduces the main methodologies proposed in literature and also describes the previous works carried out to compare them.

2.1 Measurement Techniques

Several active end-to-end measurement tools have been proposed in the last years. Looking at the big picture, these systems infer the available bandwidth of a network path by sending a few packets and analyzing the effects on the probe frames of intermediate nodes and cross-traffic.

Examples of probing tools which have emerged in recent years are Pathload [5], IGI/PTR [6], Abing [7], Spruce [8], pathChirp [9], DietTopp [10], Yaz [11], and ASSOLO [12]. These methods differ in the size and temporal structure of probe streams, and in the way the available bandwidth is derived from the received packets.

Spruce [8] uses tens of packet pairs having an input rate chosen to be roughly around to the capacity of the path, which is assumed to be known. Moreover, packets are spaced with exponential intervals in order to emulate a poissonian sampling process.

IGI [6] uses a sequence of about 60 unevenly spaced packets to probe the network and the gap between two consecutive packets is increased until the average output and initial gaps match. Similarly, PTR relies on unevenly spaced packets but the background traffic is detected through a comparison of the time intervals at the source with those found on the destination side.

Abing [7] relies on packet pair dispersion technique. Typically, 10 or 20 closely spaced probes are sent to one destination as a train. The evaluation of the

observed packet pairs delays and the estimation of the available bandwidth are based on a technical analysis of the problems that the frames could meet in the routers or other network devices.

Pathload [5] and DietTopp [10] use constant bit-rate streams and change the sending rate every round. Although both tool try to identify the turning point, DietTopp increases linearly the sending rate in successive streams while Pathload varies the probing rate using a binary search scheme. Yaz [11] is a similar estimation tool derived from Pathload which should reports results more quickly and with increased accuracy with respect to its predecessor.

PathChirp [9] sends a variable bit-rate stream consisting of exponentially spaced packets. The actual unused capacity is inferred from the rate responsible for increasing delays at the receiver side. ASSOLO [12] is a tool based on the same principle, but it features a different probing traffic profile and uses a filter to improve the accuracy and stability of results.

Other works like AB-Shoot [13], S-chirp [14], FEAT [15], BART [16] or MR-BART [17] have also been proposed in the past. However, the source codes of these tools have never been released publicly or the methods have been implemented only in simulations.

A detailed analysis of the existing estimation techniques is outside the scope of this paper – a proposed taxonomy has been developed by [8], while more information on specific tools can be found in the original papers.

2.2 Past comparisons

Most of tools' proponents have compared the performance of their solution against that of others researchers. For example, in [11] Sommers et al. compared Yaz with Pathload and Spruce in a controlled environment, while Strauss and his colleagues [8] investigated the performances of Spruce against IGI and Pathload over hundreds of real Internet paths. Ribeiro et al. [9] tested pathChirp against Pathload and TOPP through emulation. In [12] the performance of pathChirp has been compared to that of ASSOLO in a laboratory network setup.

Unfortunately, the works mentioned above covered only a small number of tools and the scenarios investigated are limited too. A more comprehensive evaluation has been performed by Shriram et al. [18], who compared Abing, pathChirp, Pathload and Spruce on a high-speed testbed and on real world GigE paths. The specific features of the network paths also allowed the researchers to investigate timing issues related to high-speed links and network interfaces. A similar work has been carried out by Labit et al. [19], that tested Abing, Spruce, Netest/Pipechar, pathChirp and IGI over a real Internet path of the French national monitoring and measurement platform Metropolis.

Angrisani et al. [20] compared IGI, pathChirp and Pathload in a testbed equipped with a proper measurement station. The adoption of a performance evaluation methodology relying on the use of electronics instrumentation for time measurements allowed the authors to focusing on concurrence, repeatability and bias of the results obtained from the testbed. Furthermore, an optimal setting of each tool has been identified thanks to the experimental activity.

In [21] the authors presented a comparative study of DietTopp, Pathload and pathChirp in a mobile transport network. However, all the results presented have been generated only from simulations using *ns2*. The ns2 network simulator has been used also by Shriram and Kaur [22] to evaluate the performance of Pathload, pathChirp, Spruce, IGI and Cprobe under different network conditions.

Two additional works in this research fields are [23] and [24]. In the first paper the authors proposed a comparative analysis of Spruce, Pathload, pathChirp and IGI in a simple testbed and they analyzed in depth the measurement errors and the uncertainty of the tools. On the other hand, in the latter article Urvoy-Keller et al. investigated the long-term behavior and the biases of Pathload and Spruce collecting data from real Internet paths.

Finally, Guerrero and Labrador [25] presented a low cost and flexible testbed and they evaluated Pathload, IGI, and Spruce in a common environment in presence of different cross-traffic loads. The same authors included in [4] more tools in the performance evaluation, comparing Pathload, pathChirp, Spruce, IGI and Abing. The considered scenarios were extended too, examining varying packet loss rate, cross-traffic packet size, link capacity and delay. In addition, the newest article pointed out which tools might be the best choices for particular applications and environments.

Although great efforts have been made to compare the existing estimation methods, all past works considered only part of the existing measurement tools. The above-mentioned experiments have also been performed considering different scenarios and testbed configurations, thus making the various results not easily comparable.

We advocate the need for a unified, flexible and low-cost platform for independent evaluations of measurements tools, and we propose in this paper a testbed solution based on free GPL-licensed software alternative to the one described in [4]. Our study also takes one step further with respect to previous works, since it proposes the largest comparison of available bandwidth estimation tools – the performances of 9 software programs are examined in terms of accuracy, time and intrusiveness. All the tools have been ported to a recent operating system, and the changes required to make older software work on a newer system have been publicly released [26].

3 Testbed setup

All the experimental results reported in this paper have been obtained using the simple testbed setup depicted in Figure 3. Our controlled network is based on general purpose PCs running only open source software. Two low-cost computers running Ubuntu GNU/Linux 8.04 are connected together through a 100 Mbps Fast Ethernet link and serve as routers. Two other machines of the testbed have been used to load the network with a source of controlled traffic originated by D-ITG [27] traffic generator. Finally, we installed the client and the server of each measurement tool on two additional computer running Ubuntu GNU/Linux 8.04

and we connected these two end-host to the testbed. We also added two Fast Ethernet switches to the network in order to make the tests more realistic.

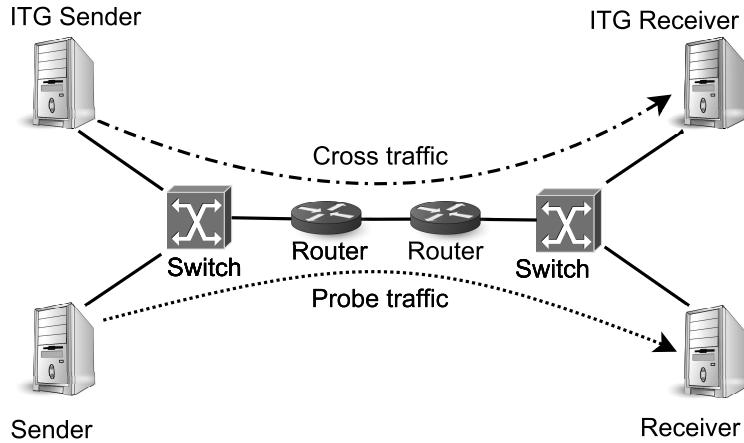


Fig. 1. Testbed setup used to compare available bandwidth estimation tools.

The two intermediate routers which emulate the multi-hop network path are based on Linux 2.6. The routers also contains `iproute2 tc` [28], an utility used to configure traffic control mechanisms in the Linux kernel. With `tc` it is possible to change the capacity of each interface, limiting the output in terms of packets or byte per second. `tc` also supports different packet queuing disciplines, and it can emulate the properties of wide area networks, such as variable delay, loss, duplication and re-ordering, through the `netem` kernel component.

The traffic generator D-ITG allowed us to produce traffic at packet level, replicating stochastic processes having both temporal and packet size distributed as random variables with known properties. D-ITG is also capable to generate traffic at network, transport, and application layer, and it can also use real traffic traces. In our experiments we loaded the network with poissonian or constant bit rate (CBR) cross-traffic with varying rate from 0 to 64 Mbps and we did not introduce any traffic shaping policy.

The final topology of the testbed and the scenarios considered are simple and admittedly unrealistic, but sufficient to perform a preliminary evaluation of the various measurement tools. A similar configuration has been used for example in [20] and [23], and the resulting system has the same features and flexibility of the testbed proposed in [25].

All the tools considered in this work must be executed on the two terminal hosts of the measured path, using a regular user account (administrator privileges are not required). For each measurement tool we left the configuration parameters untouched, using the default values suggested by the authors in the original paper or directly within the software – a list of list of the main configuration settings for each tested program is given in [26]. Although better results

might be obtained using different setups, an optimal tuning of the tools is outside the scope of this paper. We also run one measurement tool at a time – as shown in [29], current techniques can offer good estimates when used alone, but they might not work if several estimations severely interfere with each other.

4 Experimental Results

Using the testbed described before, we evaluated Abing, ASSOLO, DietTopp, IGI, pathChirp, Pathload, PTR, Spruce and Yaz in terms of estimation time, overhead and accuracy. For each tool, we considered respectively 5 CBR and 5 additional poissonian cross-traffic scenarios with varying intensity. We loaded the network using sources of 64, 32, 16 and 8 Mbps and, finally, we turned off the traffic generator. We repeated the measurement process from scratch 10 times before calculating the averaged results for each scenario.

The convergence time and the amount of probe traffic transmitted have been calculated considering only actual probing frames. For example, we did not consider the delay of the initial control connection phase which most of the tools use to synchronize the client and the server.

4.1 Accuracy

Figures 2 and 3 show the average results obtained from our experiments. Abing, Spruce and DietToop provide good estimations in presence low-rate cross-traffics, but the accuracy decreases significantly when the network load increases. On the contrary, the stability and the accuracy of measurements obtained with IGI and PTR increase when the intensity of the cross-traffic is higher. PathChirp constantly overestimates available bandwidth and its measurements are quite unstable – this is a well-known problem of this tool and similar results have been obtained in [16], [18], [23]. Pathload and Yaz are quite accurate and their results are similar; this is justified by the fact that Yaz is a modified version of Pathload. Comparable results in terms of accuracy are also provided by ASSOLO.

It is worth noting that the measured values do not exhibit significant differences with respect to the kind of cross-traffic source – the tools performed in the same way regardless of the use of CBR or poissonian distributed packets.

4.2 Intrusiveness

Table 1 shows the preliminary data obtained from the testbed network in presence of a 16 Mbps CBR cross-traffic load, that is an available bandwidth of around 80 Mbps. We ran the measurement process for each tool in this scenario and we used a network protocol sniffer [30] to evaluate the exact time required to provide an estimation and the amount of probe traffic injected into the path.

During tests we calculated only the actual estimation time and the probe traffic, not considering for example the delay introduced by an eventual initial control connection phase. Similarly, we ignored the traffic and the time required

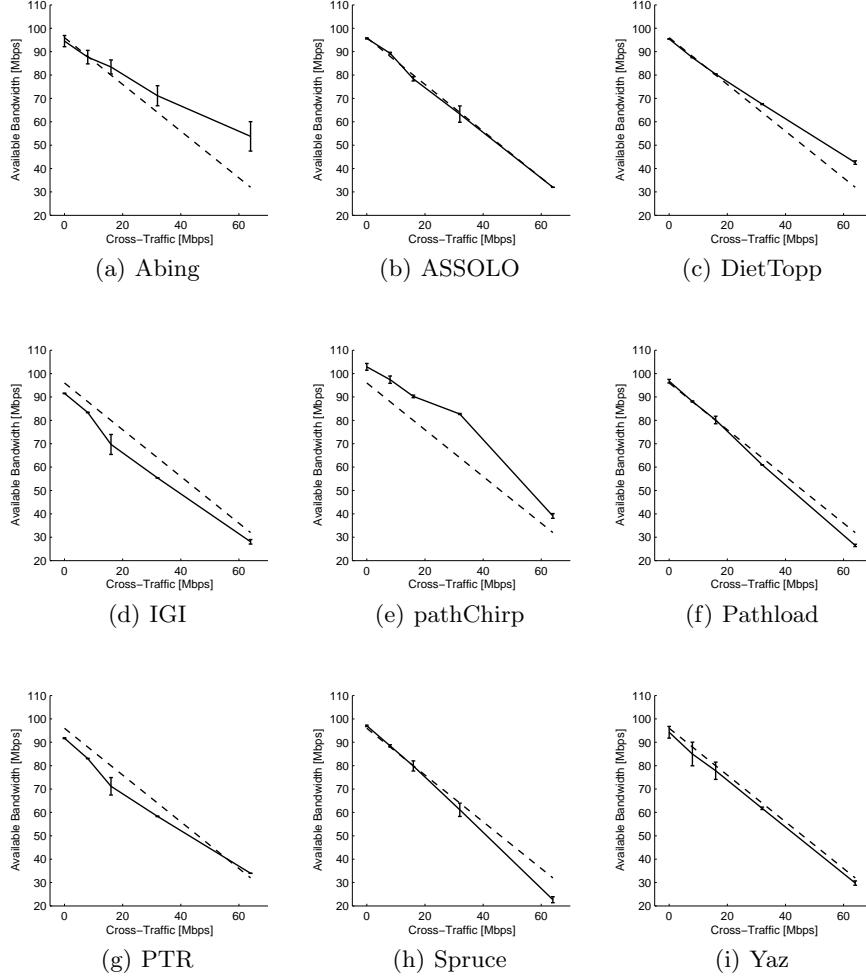


Fig. 2. Experimental results (solid line) obtained in presence of Constant Bit Rate cross-traffic with varying rate from 0 to 64 Mbps (dashed line).

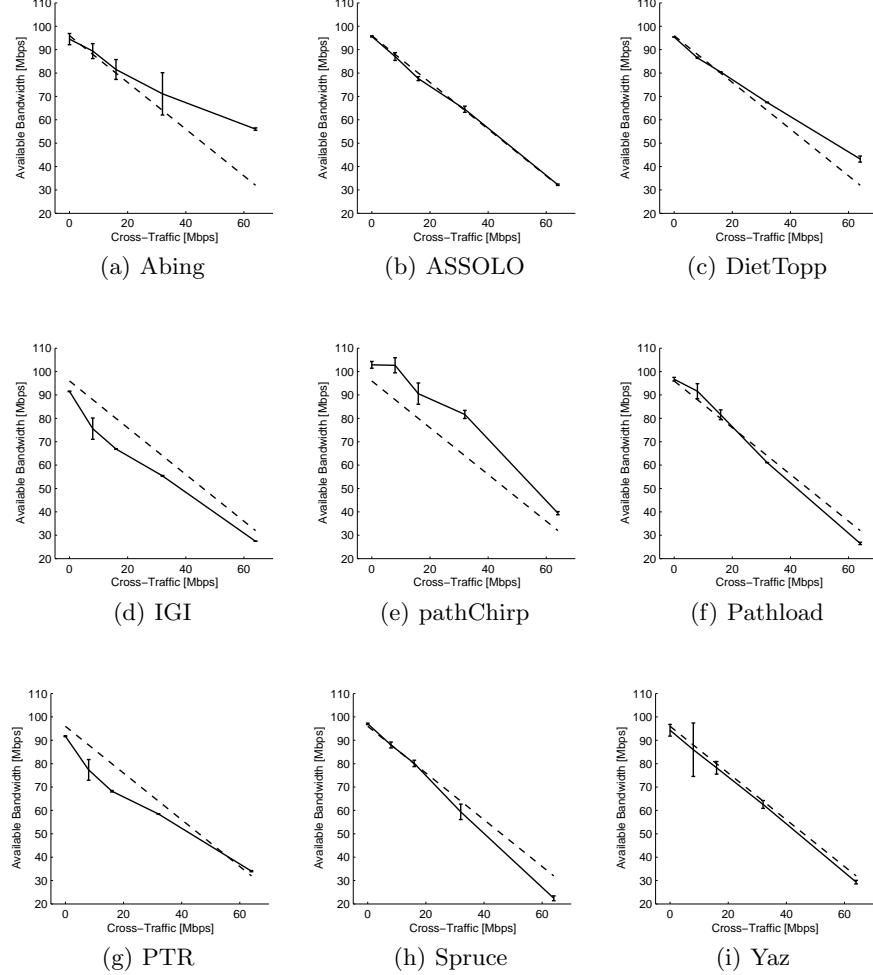


Fig. 3. Experimental results (solid line) obtained in presence of poissonian cross-traffic with varying rate from 0 to 64 Mbps (dashed line).

by IGI and PTR to measure the initial capacity. For each tool, we considered a single estimation, although tools like ASSOLO and pathChirp usually repeat the process a number of times and produce a final value using filtering techniques.

During the first round of tests the average end-to-end delay in our testbed was around 2 milliseconds – this is a reasonable value for short FastEthernet links. We repeated all the experiments enabling the *netem* module on the two routers in order to emulate a real Internet path with a symmetric One-Way Delay of 125 ms.

Table 1. Estimation time (in seconds) and amount of probe traffic (in Megabyte) associated to the tools analyzed.

Tool	Traffic	$Time_{OWD=2ms}$	$Time_{OWD=125ms}$
Abing	0.6	1.0	1.2
ASSOLO	>0.1	0.4	0.5
DietTopp	7.6	1.7	1.9
IGI	9.1	0.9	1.1
pathChirp	>0.1	0.4	0.5
Pathload	40.6	7.6	18.2
PTR	9.1	0.9	1.1
Spruce	0.3	10.0	10.1
Yaz	6.4	4.3	8.1

Results show that DietTopp, IGI and PTR are quite fast, but they also inject a significant amount of probe traffic in the network; on the other hand, Spruce could take seconds but it is more lightweight. Pathload and Yaz are quite slow and intrusive, while ASSOLO, Abing and pathChirp appear to have a good trade-off between speed and intrusiveness. It is worth pointing out that ASSOLO, DietTopp, pathChirp, Pathload, Yaz and PTR are based on the concept of self-induced congestion – the search of the available bandwidth is performed by transmitting probe traffic at a rate higher than the unused capacity of the path. The major drawback of this approach is that one or more intermediate queues will fill up during the measurement process – the existing network traffic will be delayed and some packet could even be discarded by the congested bottleneck. The remaining tools are based instead on the probe gap model, which infers the available bandwidth by observing the gap measured on a packet pair injected into the path. Although this method limits the interference between probe and exiting traffic, it has been proved to be less accurate in some network scenarios [31].

The total estimation time of some tools also depends on the Round Trip Time of the observed network path – the results change significantly when iterative programs like Pathload or Yaz are used over links with sizable delays. On the other hand, the impact of the one-way delay on the direct tools is negligible since they relay on a single stream to produce an estimation.

5 Conclusion

In this work we presented the largest experimental comparison of existing available bandwidth measurement tools on a laboratory testbed. We compared tools' performance in terms of intrusiveness, response time and accuracy in presence of different cross-traffics. All the tests have been carried out on a flexible and highly customizable laboratory testbed based only on open-source software, low-cost personal computers and simple network devices. Preliminary results shows that ASSOLO, Pathload and Yaz are accurate and scale well with increasing traffic loads, while Abing seems to be the best choice from the speed-intrusiveness point of view.

Although the programs considered in this work represent the majority of existing active estimation tools, we intend to extend the experimental comparison to more candidates as they will become freely available and usable. Ongoing works are devoted to include Netest [32] in our testbed – this was a promising tool but the source code is based on a home-made and unmaintained build system which does not compile successfully under any modern GNU/Linux environment. BART is another recent tool which is being used for experiments over the European research measurement infrastructure Etomic. However, Ericsson owns BART intellectual property rights and the code has not yet been freely released for scientific purposes.

The laboratory testbed we used is actually quite simple: the single-bottleneck topology and the limited number of links oversimplify reality, and the CBR or poissonian cross-traffic sources do not fully catch the complexity of actual communication flows. We also did not consider longterm oscillations or biases in the estimations and the analysis we performed does not include highly congested scenarios. Although quite promising, the preliminary results obtained from our tests are not sufficient to draw any definitive conclusions on how the tools will behave on real networks. As further development, we plan to complete the analysis extending the set of considered scenarios to actual Internet paths or using a testbed having a more complex topology and loaded with real-world traffic traces.

References

1. Chuan, W., Baochun, L., Shuqiao, Z.: Characterizing Peer-to-Peer Streaming Flows. In: IEEE JSAC, vol 25(9), 1612–1626 (2007)
2. Favalli, L., Folli, M., Lombardo, A., Reforgiato, D., Schembra, G.: A Bandwidth-Aware P2P Platform for the Transmission of Multipoint Multiple Description Video Streams In: Proceedings of the Italian Networking Workshop 2009, (2009)
3. Shamsi, J., Brockmeyer, M.: Principles of Network Measurement. In: Misra, S., Misra, S. C., Woungang, I. (eds) Selected Topics Communication Networks and Distributed Systems, pp. 1–40, World Scientific, Singapore (2010)
4. Guerrero, C.D., Labrador, M.A.: On the applicability of available bandwidth estimation techniques and tools. In: Computer Communications, vol. 33(1), 11–22 (2010)

5. Jain, M., Dovrolis, C.: Pathload: A measurement tool for end-to-end available bandwidth. In: Proceedings of the 3th International workshop on Passive and Active network Measurement PAM 2002 (2002)
6. Hu, N., Steenkiste, P.: Evaluation and Characterization of Available Bandwidth Probing Techniques. In: IEEE JSAC, vol. 21(6), 879–894 (2003)
7. Navratil, J., Cottrell, R.L.: ABwE: A Practical Approach to Available Bandwidth. In: Proceedings of the 4th International workshop on Passive and Active network Measurement PAM 2003 (2003)
8. Strauss, J., Katabi, D., Kaashoek, F.: A measurement study of available bandwidth estimation tools. In: Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement IMC'03 (2003)
9. Ribeiro, V., Riedi, R., Baraniuk, R., Navratil, J., Cottrell, L.: pathChirp: Efficient Available Bandwidth Estimation for Network Paths. In: Proceedings of the 4th International workshop on Passive and Active network Measurement PAM 2003 (2003)
10. Johnsson, A., Melander, B., Bjorkman, M.: DietTopp: A First Implementation and Evaluation of a Simplified Bandwidth Measurement Method. In: Proceedings of the 2nd Swedish National Computer Networking Workshop (2004)
11. Sommers, J., Barford, P., Willinger, W.: A Proposed Framework for Calibration of Available Bandwidth Estimation Tools. In: Proceedings of the 11th IEEE Symposium on Computers and Communications ISCC'06, pp. 709–718 (2006)
12. Goldoni, E., Rossi, G., Torelli, A.: Assolo, a New Method for Available Bandwidth Estimation. In: Proceedings of the Fourth International Conference on Internet Monitoring ICIMP'09, pp. 130–136 (May 2009)
13. Tan, W., Zhanikeev, M., Tanaka, Y.: ABshoot: A Reliable and Efficient Scheme for End-to-End Available Bandwidth Measurement. In: Proceedings of the IEEE Region 10 Conference TENCON 2006, pp. 1–4 (2006)
14. Pasztor, A.: Accurate Active Measurement in the Internet and its Applications. University of Melbourne, Department of Electrical and Electronic Engineering, Ph.D. Thesis (2003)
15. Qiang, W., Liang, C.: FEAT: Improving Accuracy in End-to-end Available Bandwidth Measurement. In: Proceedings of IEEE Global Telecommunications Conference GLOBECOM '06, pp. 1–4 (2006)
16. Ekelin, S., Nilsson, M., Hartikainen, E., Johnsson, A., Mangs, J.-E., Melander, B., Bjorkman, M.: Real-Time Measurement of End-to-End Available Bandwidth using Kalman Filtering. In: Proceedings of 10th IEEE/IFIP Network Operations and Management Symposium NOMS 2006, pp. 73–84 (2006)
17. Sedighzad, M., Seyfe, B., Navaie, K.: MR-BART: multi-rate available bandwidth estimation in real-time In: Proceedings of the 3rd ACM workshop on Performance monitoring and measurement of heterogeneous wireless and wired networks PM2HW2N'08, pp. 1–8 (2008)
18. Shriram, A., Murray, M., Hyun, Y., Brownlee, N., Broido, A., Fomenkov, M., claffy, K., Comparison of public end-to-end bandwidth estimation tools on high-speed links. In: Proceedings of the 6th International workshop on Passive and Active network Measurement PAM 2005, LCNS, vol. 3431, 306–320, Springer, Heidelberg (2005)
19. Labit, Y., Owezarski, P., Larrieu, N.: Evaluation of active measurement tools for bandwidth estimation in real environment. In: Proceedings of the IEEE/IFIP Workshop on End-to-End Monitoring Techniques and Services E2EMON 2005, pp. 71–85 (2005)

20. Angrisani, L., D'Antonio, S., Esposito, E., Vardusi, M.: Techniques for available bandwidth measurement in IP networks: a performance comparison. In: Elsevier Computer Networks, vol. 50(3), 332–349 (2006)
21. Castellanos, C.U., Villa, D.L., Teyeb, O.M., Elling, J., Wigard, J.: Comparison of Available Bandwidth Estimation Techniques in Packet-Switched Mobile Networks. In: Proceedings of the 17th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, pp. 1–5 (2006)
22. Shriram, A., Kaur, J.: Empirical Evaluation of Techniques for Measuring Available Bandwidth. In: Proceedings of 26th IEEE International Conference on Computer Communications INFOCOM 2007, pp. 2162–2170 (2007)
23. Ait Ali, A., Michaut, F., Lepage, F.: End-to-End Available Bandwidth Measurement Tools: A Comparative Evaluation of Performances. In: Proceedings of the 4th International Workshop on Internet Performance, Simulation, Monitoring and Measurements IPS-MoMe 2006, pp. 1–14 (2006)
24. Urvoy-Keller, G., En-Najjary, T., Sorniotti, A.: Operational comparison of available bandwidth estimation tools. In: ACM SIGCOMM Comput. Commun. Rev., vol. 38(1), 39–42, (2008)
25. Guerrero, C.D., Labrador, M.A.: Experimental and Analytical Evaluation of Available Bandwidth Estimation Tools. In: Proceedings of the 31st IEEE Conference on Local Computer Networks 2006, pp. 710–717 (2006)
26. University of Pavia, Networking Lab: Collection of Available Bandwidth Estimation Tools, <http://netlab-mn.unipv.it/avail-bw/>
27. Avallone, S., Guadagno, S., Emma, D., Pescap, A., Ventre, G.: D-ITG Distributed Internet Traffic Generator. In: Proceedings of the First International Conference on Quantitative Evaluation of Systems QEST'04, pp. 316–317 (2004)
28. Hemminger, S., Kuznetsov, A. et al.: iproute2 utility suite, <http://www.linuxfoundation.org/collaborate/workgroups/networking/iproute2>
29. Croce, D., Mellia, M., Leonardi, E.: The Quest for Bandwidth Estimation Techniques for large-scale Distributed Systems. In: Proceedings of ACM HotMetrics 2009 (2009)
30. Combs, G. et al.: The Wireshark Network Protocol Analyzer, <http://www.wireshark.org>
31. Lao, L., Dovrolis, C., Sanadidi, M.Y.: The probe gap model can underestimate the available bandwidth of multihop paths. In: ACM SIGCOMM Comput. Commun. Rev., vol. 36(5), 29–34, (2006)
32. Jin, G., Tierney, B.: Netest: a tool to measure the maximum burst size, available bandwidth and achievable throughput In: Proceedings of the International Conference on Information Technology: Research and Education ITRE'2003, pp. 578–582 (2003)