Comparative Analysis of Active Bandwidth Estimation Tools

Federico Montesino-Pouzols

Instituto de Microelectrnica de Sevilla (IMSE-CNM) – CSIC
Avda. Reina Mercedes s/n. Edif. CICA. E-41012 Seville, Spain
Federico.Montesino@imse.cnm.es

Abstract. A comparative analysis of state-of-the-art active probing tools for bandwidth estimation is outlined. Techniques and tools for capacity, available bandwidth and bulk transfer capacity estimation are simultaneously assessed. First, a generic framework for the design of active bandwidth estimation tools is proposed as a result of our analysis of the implementation and performance of a number of available techniques and tools. Then, we describe a first and comprehensive series of tests performed over a broad range of bandwidth estimation tools, scenarios and experimental conditions. These tests have been done with the aim to develop a set of practices, procedures and tools for the comparative analysis of active bandwidth estimation techniques and tools, which are presented. Finally, Overall conclusions are discussed and pointers to publicly available records and tools are given.

1 Introduction

Active bandwidth estimation tools can not only provide network operators with useful information on network characteristics and performance, but also can enable end users (and user applications) to perform independent network auditing, load balancing, and server selection tasks, among many others, without requiring access to network elements or administrative resources.

The research community is developing a set of metrics and techniques for active bandwidth measurement. Many of them [1] are well understood and can provide accurate estimates under certain conditions.

Some institutions have announced initiatives to deploy test platforms for active and passive bandwidth estimation as well as other related techniques, though no substantial results have been reported as for active bandwidth estimation. Also, some partial measurement and evaluation studies of bandwidth estimation tools have been published [2,3]. Nonetheless, we draw attention to the lack of publicly available comprehensive and comparative experimental results. We also note the lack of common and consistent practices and procedures to test available active bandwidth measurement tools.

Our aim is to develop a set of practices, procedures and tools for the comparative analysis of active bandwidth estimation techniques and tools. This way, we expect to fulfill two goals:

– Ease the deployment of platforms that take advantage of the huge amount of available experimental resources to provide a solid experimental basis for research on active bandwidth estimation.
– Assess to what extend current bandwidth estimation tools can be used as basis for providing a bandwidth estimation service for user applications as well as networks operation.

2 Analysis of Bandwidth Estimation Techniques

Our study covers measurement techniques as well as implementations. In this section, an study of measurement techniques for different metrics and techniques is outlined, ignoring implementation details of tools. The nomenclature and taxonomy of metrics, measurement techniques and tools given in [1] is considered as reference.

Our approach is to simultaneously assess techniques and tools for capacity, available bandwidth and bulk transfer capacity estimation, which allows us to abstract a common design model for these tools.

In this section, we present a generic framework for the design of active bandwidth estimation tools that will be referenced in following sections. Though this analysis can be generalized to network performance and dynamics analysis techniques, we restrict the following discussion to bandwidth estimation techniques.

A review of the source code of a number of active bandwidth estimation tools has been performed. As in early stages of development of bandwidth estimation tools there has been no clear taxonomy of techniques and no consensus on nomenclature and definition of bandwidth metrics, a great deal of code duplication among bandwidth estimation tools has been found.

The facts that `pathChirp` [4] has been implemented using the `NetDyn` [5] code as starting point, and that `cprobe` [6] and `pipechar` [7] provide estimates of the asymptotic dispersion rate [8,3] rather than the available bandwidth, are revealing. Thus, we have paid special attention to isolating reusable components in bandwidth estimation tools.

A simplified scheme that summarizes components and relations of a bandwidth estimation tool according to our analysis is depicted in Figure 1. Two main stages in these tools are identified: measurement and estimation. *Measurement* involves the generation of a probe packet pattern, its transmission through the network, and its reception and measurement. *Estimation* comprises statistical and heuristic processing of measurements according to some network model.

A major component is what we call probe packets generation system; both the measurement and the estimation stages depend on it. Isolation of this function as a component allows for the efficient implementation of tools for a set of measurement techniques, since facilities for common traffic patterns generation can be reused.

The probe packet generation component should provide generic services such as generation of packet trains, which currently can be found implemented as functions or methods in most tools based on packet trains. In case of packet

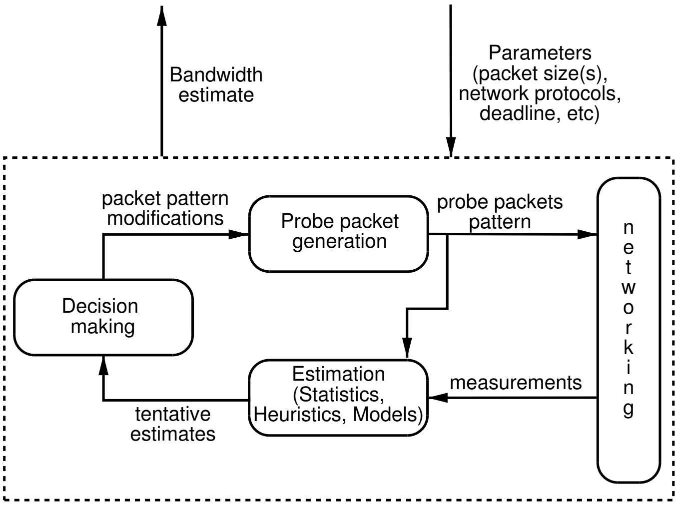

Fig. 1. Design scheme of an active bandwidth estimation tool

trains for techniques that follow the probe gap model [3], the packet gap can be fixed (as in `IGI/PTR` [2]), or a Poisson process (as in `Spruce` [3]. Similarly, in case of techniques that use the probe rate model [3], facilities can be defined for different patterns of packet trains as found in tools such as `Pathload` [9], `PathChrip` [4] and `TOPP` [10].

Also, within this design framework, traffic generation tools (such as Iperf [11]), often employed jointly with bandwidth estimation tools in order to perform tests, can be integrated as additional facilities provided by the probe packets generation component.

A feedback line from the estimation component to the probe packets generation component can improve performance by reducing network load and required estimation time. Within this feedback line, we consider a third component, *decision making*, in charge of conducting the probe packet generation process depending on the estimated accuracy and convergence of tentative estimates from the estimation component.

Keeping the probe packets generation system as a generic component separate from the decision making component makes it possible to implement tools that combine different probe packets patterns to estimate bandwidth, in a way similar to that of `IGI/PTR`. Some tools, such as early versions of `pchar` [12] lack the decision making module. In such cases, the probe packets pattern is fixed for each set of input parameters.

Note that the simplified scheme shown omits some reusable subsystems such as route change detection. In case of tools that run both at the receiver and the sender side, the scheme in Figure 1 corresponds to the end system that sends probe packets (usually the client). A more in depth description of this design can be found at [13]

3 Comparative Analysis of Active Bandwidth Estimation Tools

In this section, we outline the comparative analysis of active bandwidth estimation tools carried out. We describe the scenarios considered as well as overall practices and procedures followed to conduct experiments. General conclusions from our experiments are summarized.

We evaluate tools rather than techniques, focusing on design and implementation details that affect estimates accuracy, efficiency, speed and reliability. We put the focus on analyzing experimental results for developing and refining practices, procedures and tools for iterative tests performance.

Previous initiatives conducted by a number of institutions have been taken into account. In particular, we have partially based our work on studies performed on the old TEN-155 European research and education network [14] (which reports a detailed study, based on `netperf` [15], performed over a number of network paths and throughout long periods of time), a more recent analysis performed for GÉANT [16] (the current European research and education network), and the Internet Monitoring Program at Stanford Linear Accelerator Center [17]

These initiatives are aimed at studying the deployment of bandwidth estimation tools in production networks as the basis for an additional service available for end-user applications as well as network operators. However, no systematic evaluation of a significant subset of currently available active bandwidth estimation tools with respect to a complete set of criteria, across a wide variety of network paths and experimental conditions, using a well defined set of procedures, has been conducted.

Since we focus on experimental results, we do not perform tests through simulation but only using emulated and real scenarios.

Tools such as dummynet and NIST Net running on FreeBSD and GNU/Linux boxes are being employed for scenario emulation. Emulated scenarios make it possible for us to test the effects both of extreme network configurations and traffic shapers on estimation tools.

Real scenarios range from LAN environments and cable subscriber networks operated by private ISPs to the high performance Spanish NREN (national research and education network) backbone operated by RedIRIS, with connections to GÉANT as well as private ISPs. Tests are being conducted on network paths with variable length, ranging from single links to paths that traverse one or more administrative domains.

Most tests have been performed through paths between the following three end points:

- Cable modem end point (private ISP subscriber line).
- Several hosts at a LAN environment within a research institution connected to the Spanish NREN.
- Host near the RedIRIS backbone.

Considering the paths between these end points, we have applied a set of general practices. The most important ones are the following:

– For each path, whenever possible, tests are performed in both directions, distinguishing symmetric and asymmetric paths.
– Estimation tools are run through `GNU time`, recording its output together with the tool logs.
– Attach to test logs output from helper tools that provide basic information about network paths, such as DNS lookups, `traceroute` and `ping`.
– When testing each of the analyzed tools, perform tests with only one instance of the tool, and also with multiple instances of the same tool running at the same time.
– Perform tests both with overall idle systems and heavily loaded systems. For the latter case, we distinguish three types of system load:
  - cpu intensive tasks
  - input/output intensive tasks
  - both cpu and input/output intensive tasks
– For tools based on ICMP, perform tests with several levels of ICMP rate limits.
– In order to analyze the effects of simple traffic patterns, we use automated traffic generation tools such as Iperf
  (see http://www.icir.org/models/trafficgenerators.html).
– Consider path properties that imply known limitations for which there is no known solution, among which the effect of layer-2 store-and-forward devices on capacity estimation tools based on the variable packet size technique has been taken into account for our LAN experiments.

For simplicity, we are using just IPv4, as it is the network protocol common to the whole set of tools under test.attention has being paid to failure conditions and to the set of parameters accepted as user input by the tools being tested. We have performed small modifications to the source code as a way to apply some of our procedures.

Some tests have been performed using tools for estimating capacity , available bandwidth and bulk transfer capacity under the same network configuration and conditions. In such cases, for each path and each hop in the path, the theoretical relationships between these metrics provide a method to detect wrong estimates. These tests also reveal similar and disparate effects of practical conditions and scenarios on different estimation techniques.

We have completed a source code review for a comprehensive number of active bandwidth estimation tools available from the research community (see the lists of tools provided by Cooperative Association for Internet Data Analysis (CAIDA) and Stanford Linear Accelerator Center [17],among other organizations.

Tools and versions analyzed to date are enumerated in Table 1. A taxonomy as well as a brief discussion of many of them can be found in [1]. Previous

Tool	Version
Capacity	
bprobe	1.0
clink [18]	1.0
Nettimer [19]	2.3.8
pathrate	2.3.3
pchar	1.4 (+ Debian GNU/Linux package 1.4-4 patches)
sprobe [20]	0.3
Available Bandwidth	
cprobe	1.0
IGI/PTR	1.0
NetDyn	-
pathChrip	2.3.3
pathload	1.1.1
pipechar	Mar25-2K1
Spruce	0.2
Bulk Transfer Capacity	
Iperf [11]	1.7.0
netperf	2.2pl4
Treno	961001
nttcp	1.43

Table 1. Tools and versions analyzed

comparisons of some of the available bandwidth estimation tools are also published [2,3]. For a complete and updated enumeration of the tools analyzed refer to [13].

As we perform tests over these tools, comparative analysis procedures are being developed. Among these, we highlight a number of generic comparison criteria, though we note that not every criterion can be applied to each and every tool.

We have classified these criteria into two groups: a first set of criteria specified as simple numerical and scalar metrics that provide a partial comparison of tools performance, and a second set of criteria that provide a comparative evaluation of tools performance.

The first set includes metrics that can be useful to quickly choose or discard some techniques and tools according to simple constraints such as required time or allowed probe bandwidth, some of these metrics are the following:

- Total probe traffic generated.
- Maximum attainable accuracy.
- Total estimation time.

The second set of criteria, suited for comparative analysis between different techniques includes but is not limited to the following:

– Accuracy: maximum accuracy for a fixed time, maximum accuracy for a limited amount of probe traffic, dependency of accuracy on estimation time, and dependency of accuracy on probe traffic volume.
– Consistency of estimates (as discussed in [18]).
– Efficiency: required network load for a given accuracy, and required network load for a given estimation time.
– Estimation time: required time for a given accuracy.
– Dependency of accuracy, efficiency and estimation time on overall network load as well as overall machine (both sender and receiver) load.
– Dependency of accuracy, efficiency and estimation time on path properties such as number of hops and rount trip time.

In addition to these numerical criteria for comparative analysis, we have analyzed the response of estimation tools to the following factors:

– Effects of path asymmetry.
– Effects of layer-2 store-and-forward devices.
– Route changes.
– Installation required on both ends of the path.
– Possibility to estimate per hop properties.
– What path property is actually measured for paths with traffic shaping nodes.
– Reliability in case of loss of connectivity between sender and receiver.
– Possibility to run multiple instances simultaneously. Particularly, as already noted in [3], train generation tools hold the CPU for the full packet train generation interval, preventing the simultaneous generation of several trains. In low bandwidth links, the delay is high enough so as to use system calls for scheduling. However, high bandwidth links require the design of packet train generators able to cope with multiple packet trains.
– Congestion avoidance mechanisms implemented in tools.
– Effects of cross traffic.
– Upper and lower bandwidth limits for which the tool works.
– Operating system privileges required to run the tool (privileged socket numbers, raw sockets interface, etc.).
– Underlying suppositions in the network model and estimation technique followed by the tool. path is a static characteristic. As an example, tools based on the probe gap model [3] (as `Spruce`, and `IGI`) assume there is a single bottleneck in the path (which is both the narrow and tight link), and also assume that the queue of that link does not become empty between the probe packets [2,3]
– Support for important generic parameters, such as socket numbers and constraints for probe packets size.

Repeated application of the aforementioned practices and procedures has lead to the developemnt of a system of tests automation as well as results processing and recording scripts implemented in Ruby. These have been complemented with additional scripts for visualization based on the comparison criteria described, making up a framework for the comparative analysis of active bandwidth estimation tools.

As for the comparison of output from different tools, we point out the need for a common output format for active bandwidth estimation tools. Output from scripts of the test framework is kept in NetLogger ULM format [21]. Also, we are currently developing scripts that convert, when possible, specific log formats to the NetLogger ULM format (already supported by `pathload` and `pathrate`) for reporting estimates.

Due to space constraints we do not include a complete description of experiments. A more comprehensive description can be found at [13].

We note, however, that current tools are implementations of one bandwidth estimation technique. For this reason, differences in networking code (particularly those portions related to timestamping) may have an undesired impact on comparing similar techniques implemented through different tools. These and other implementation issues are hard to quantify and even isolate.

It has been found that sometimes the analyzed tools provide very wrong estimates or even no estimates at all likely as a consequence of implementation issues, specially noticeable when estimating high bandwidths (around 100 Mbps and above), but also relatively often when estimating bandwidth of 10Mbps or even lower.

Limited clock resolution and system I/O throughput, as well as the fact that these tools are usually run on non real-time systems seem to be the main underlying reasons. Recent developments [22] and analysis [23] confirm our observations.

4 Conclusions and Future Work

We have presented the first systematic comparative analysis of active bandwidth estimation tools performed over a broad range of tools, scenarios (both emulated and real) as well as experimental conditions, which results are publicly available from [13]. Techniques and tools to estimate capacity, available bandwidth and bulk transfer capacity have been studied jointly.

With this work we aim to lay out a common framework for the design and implementation of component based active bandwidth estimation tools. In addition, a set of practices, procedures and tools which make up a test framework for the comparative analysis of active bandwidth estimation tools has been developed. The test framework as well as records from our experiments are also available from [13].

As a result from the tests carried out, a number of failure conditions for the analyzed tools have been identified, as well as the dependency of the estimates accuracy on factors such as system load and network path properties. How these factors impact estimates of the three metrics studied (capacity, available bandwidth and bulk transfer capacity) has been jointly analyzed.

As further development we plan to extend the set of considered scenarios to include network testbeds such as PlanetLab [24]. Also, we are currently working on a web interface for the presented test framework which is intended to be a first experimental platform for cooperative comparative analysis of bandwidth estimation tools through the Internet, based on the practices, procedures and tools developed.

Finally, we expect to raise a well defined set of questions (and rationales from our experiments) in order to establish a technical comparison criteria that could eventually lead to the development and standarization of benchmarks for active bandwidth estimation tools.

5 Acknowledgement

We would like to thank D.R. López and A. Barriga for their useful comments and assistance. We would also like to thank IMSE and RedIRIS for allowing us to use their resources in our experiments

References

1. Prasad, R.S., Murray, M., Dovrolis, C., Claffy, K.: Bandwidth estimation: metrics, measurement techniques, and tools. IEEE Network Magazine (2003) http://www.cc.gatech.edu/fac/Constantinos.Dovrolis/publications.html.
2. Hu, N., Steenkiste, P.: Evaluation and Characterization of Available Bandwidth Probing Techniques. IEEE Journal on Selected Areas in Communication **21** (2003) 879–894 http://gs274.sp.cs.cmu.edu/www/igi/.
3. Strauss, J., Katabi, D., Kaashoek, F.: A Measurement Study of Available Bandwidth Estimation Tools. In: ACM SIGCOMM Internet Measurement Workshop. (2003) http://www.icir.org/vern/imc-2003/program.html.
4. Ribeiro, V.J., Riedi, R.H., Baraniuk, R.G., Navratil, J., Cottrell, L.: pathChirp: Efficient Available Bandwidth Estimation for Network Paths. In: Passive and Active Measurement Workshop. (2003) http://www.pam2003.org/proceedings.html.
5. Banerjee, S., Agrawala, A.K.: Estimating Available Capacity of a Network Connection. In: IEEE International Conference on Networks, Singapore (2000) http://www.cs.umd.edu/ suman/.
6. Carter, R.L., Crovella, M.E.: Dynamic Server Selection Using Bandwidth Probing in Wide-Area Networks. Technical Report TR-96-007, Boston University. Computer Science Department (1996) http://cs-www.bu.edu/techreports/pdf/1996-007-dss-using-bandwidth.pdf.
7. Jin, G., Yang, G., Crowley, B., Agarwal, D.: Network Characterization Service (NCS). In: 10th IEEE Symposium on High Performance Distributed Computing. (2001) http://dsd.lbl.gov/DIDC/papers/NCS.HPDC01.pdf.
8. Dovrolis, C., Ramanathanm, P., Moore, D.: What Do Packet Dispersion Techniques Measure? In: INFOCOM. (2001) 905–914 http://www.cc.gatech.edu/fac/Constantinos.Dovrolis/publications.html.
9. Jain, M., Dovrolis, C.: End-to-End Available Bandwidth: Measurement Methodology, Dynamics, and Relation with TCP Throughput. In: ACM SIGCOMM, Pittsburgh, PA (2002) http://www.cc.gatech.edu/fac/Constantinos.Dovrolis/publications.html.

10. Melander, B., Bjorkman, M., unningberg, P.: A New End-to-End Probing and Analysis Method for Estimating Bandwidth Bottlenecks. In: IEEE Globecom – Global Internet Symposium, San Francisco (2000)
11. Tirumala, A., Qin, F., Dugan, J., Ferguson, J., Gibbs, K.: Iperf Version 1.7.0. NLANR Distributed Applications Support Team (2003) http://http://dast.nlanr.net/Projects/Iperf/.
12. Mah, B.A.: pchar: A Tool for Measuring Internet Path Characteristics (2001) http://www.employees.org/~bmah/Software/pchar/.
13. Federico Montesino-Pouzols et al.: Active Bandwidth Estimation Tools: ... (2004) http://www.imse.cnm.es/~fedemp/abet/.
14. Blom, H., de Laat, C.: User Level Network Performance Monitoring Program. TERENA Technical Report (2000) http://www.terena.nl.
15. Jones, R.: Public Netperf Homepage (2004) http://www.netperf.org/netperf/NetperfPage.html.
16. Leinen, S., Reijs, V.: GEANT Deliverable D9.7: Testing of Traffic Measurement Tools. Technical report, DANTE (2002) http://archive.dante.net/tf-ngn/activities.html.
17. Cottrell, L., Matthews, W.: Internet Monitoring at Stanford Linear Accelerator Center (2002) http://www.slac.stanford.edu/comp/net/wan-mon.html.
18. Downey, A.B.: Using pathchar to estimate Internet link characteristics. In: ACM SIGCOMM, Cambridge, MA (1999) 241–250 http://allendowney.com/research/clink/.
19. Lai, K., Baker, M.: Nettimer: A Tool for Measuring Bottleneck Link Bandwidth. In: USENIX Symposium on Internet Technologies and Systems. (2001) http://mosquitonet.stanford.edu/~laik/projects/nettimer/.
20. Saroiu, S., Gummadi, P.K., Gribble, S.D.: SProbe: A Fast Technique for Measuring Bottleneck Bandwidth in Uncooperative Environments. http://sprobe.cs.washington.edu/ (2002)
21. Abela, J., and, T.D.: Universal Format for Logger Messages. Internet draft (1999) draft-abela-ulm-05.txt.
22. Alberi, J.L., McIntosh, A., Pucci, M., Raleigh, T.: Overcoming Precision Limitations in Adaptive Bandwidth Measurements. In: 3rd New York Metro Area Networking Workshop (NYMAN 2003), New York (2003) http://www.nyman-workshop.org/2003/program.html.
23. Jin, G., Tierney, B.L.: System Capability Effects on Algorithms for Network Bandwidth Measurement. In: Internet Measurement Conference, Miami Beach, FL, USA (2003) 27–38 http://www.icir.org/vern/imc-2003/program.html.
24. PlanetLab: An Open Platform for Developing, Deploying, and Accessing Planetary-Scale Services. (2004) http://www.planet-lab.org.