Bedibe: Datasets and Software Tools for Distributed Bandwidth Prediction

Lionel Eyraud-Dubois, Przemyslaw Uznanski

Bedibe: Datasets and Software Tools for Distributed Bandwidth Prediction

Lionel Eyraud-Dubois[1] [†] and Przemysław Uznański[1]

[1] *INRIA Bordeaux Sud-Ouest F-78000 and LaBRI - Univ of Bordeaux F-33400 and LaBRI - CNRS F-33400*

Pouvoir prédire la bande passante disponible est une problématique cruciale pour un grand nombre d'applications distribuées sur Internet. Plusieurs solutions ont été proposées, mais l'absence d'implémentations communes et de jeux de données reconnus rend difficile la comparaison et la reproductibilité des résultats. Dans cet article, nous présentons `bedibe`, la combinaison de mesures de bande passante effectuées sur Planet-Lab et d'un logiciel pour faciliter l'écriture et l'étude d'algorithmes pour la prédiction de bande passante. `bedibe` inclut les implémentations des meilleures solutions de la littérature, et a pour but de faciliter la comparaison des résultats obtenus par les différentes équipes qui travaillent sur ce thème.

Keywords: available bandwidth prediction, network coordinates systems, distributed systems, planet lab, open science, network measurement

1 Introduction

Predicting network performance (latency or available bandwidth) is important for many Internet applications. For video on demand [SDK+07] and peer-assisted streaming [LZSJ+08] for example, estimations of available bandwidth allow the construction of an efficient overlay topology.

A number of measurement tools have been developed [GS10] which measure the available bandwidth on the path between two given Internet nodes. However, in a large scale system, performing measurements between all pairs of nodes would incur too much overhead. Thus, there is a need for the possibility to predict the unmeasured bandwidth values from a limited number of actual available measurements.

For latency estimation, several solutions have been successfully proposed (like GNP [NZ02], Vivaldi [DCKM04], and others), under the global terminology of Network Coordinate Systems. Available bandwidth estimation has been recently considered as well. A number of solutions have been proposed, and there is at the moment no clear consensus about which solutions are most efficient. Because bandwidth measurements are more expensive, few datasets are available to compare the solutions. This paper presents the `bedibe` project, which regroups a software for "BEnchmarking DIstributed Bandwidth Estimation", and a set of available bandwidth measurements performed on the Planet-Lab platform using SPLAY [LRF09]. The common goals of both contributions is to help promote the diffusion of results in the community, providing a framework to help design and compare tools for available bandwidth estimation, thus allowing to easily reproduce results obtained by others.

Related works Several bandwidth prediction algorithms have been proposed in the literature. Many have focused on embedding the nodes into a metric space, more precisely a hyperbolic structure [Din08]. An example of such a system is the Sequoia algorithm [RMK+09], which embeds the nodes as the leaves of a weighted tree, and uses distances in the tree as approximations of actual distances. The Sequoia algorithm comes with a theoretically proved performance guarantee, and can be applied to both latency and bandwidth estimation. However, the algorithm is quite sensitive to violations of triangular inequalities, which are common in the Internet. Sequoia has later been improved with a decentralized version [SKBS11], in order to lower the load of the root of the tree, and also to make use of a smaller number of measurements.

[†] This work is partially supported by the ANR SONGS project.

To cope with triangular inequality violations, several studies have thus considered non-metric embeddings. IDES [MSS06] uses matrix factorization to approximate the large measurement matrix by the product of two smaller matrices. The IDES system is based on a set of landmark nodes, and recently a decentralized version has been proposed, called DMF [LGL10] for decentralized matrix factorization. DMF is an iterative procedure in which each node locally minimizes the prediction error by solving a least square problem. Phoenix [CWS$^+$11] improves over DMF by adding features which are commonly found in NCS to cope with node churn and to add more credibility to nodes that have stayed in the system for a long time.

Last-mile [BEDW11] is a prediction mechanism based on the assumption that available bandwidth on a path is limited only by the the first and last links of the path. In this model, each node is characterized by its incoming and outgoing bandwidth. Despite the simplicity of this model, a procedure to compute these values has been proposed which achieves reasonable accuracy.

Other works include PathGuru [XCY09] which embeds nodes in an ultra-metric space but provides rather poor estimates, and BRoute [HS05] which requires network management tools (such as traceroute and BGP routing information) to identify the bottleneck links near each source and destination.

2 Bandwidth measurements on Planet-Lab

Context The S-cube project from HP [YSB$^+$06] was aimed at monitoring the large scale distributed platform Planet-Lab [‡], and in particular provided available bandwidth measurements between pairs of nodes of this platform, obtained with the Spruce tool. These datasets have been used by the community to validate bandwidth estimation tools. However, this project is now stopped, and only the last snapshot is available.

Planet-Lab is a large-scale, worldwide distributed platform which provides an access to nodes on more than 500 sites across the world. It has become a standard for conducting large scale Internet experiments, and is thus well suited for our purpose. Although it is mainly based on academic networks and thus not representative of the global Internet, measurements on this platform are very valuable for designing sound experiments on Planet-Lab. Furthermore, its accessibility makes it relatively easy to conduct the required measurements.

SPLAY [LRF09] [§] is a middleware and a development framework which aims at simplifying the prototyping and development of large scale distributed applications and overlay networks. It is based on Lua and provides tools for deploying and controlling a distributed application on a large platform. SPLAY also provides an RPC mechanism for communications between nodes.

Applicative measurements A number of tools have been developed for available bandwidth measurements [GS10]. Generally speaking, they rely on sending a few packets along the path, and analyze the effects of intermediate nodes and cross traffic on these probe packets. Although they do not require privilege access, these tools require a fine grain access to the network.

To simplify the deployment and remain within the SPLAY framework, we decided to perform applicative-level measurements: we measure the performance of the network as it is effectively available to the SPLAY application. To this end, we measured the time required to perform an RPC call with a reasonably large payload (256KB of data) between all pairs of nodes. This measurement allows to take into account all the aspects that impact communication performance for the application, and we see it as an interesting complement to low-level measurement.

To perform our measurements, we select about 100 of the most responsive nodes from Planet-Lab, and each of them simultaneously contacts a random participant to request a measurement. A distributed synchronization mechanism ensures that no two measurements are performed on the same host at the same time. However, two independent measures (concerning different sources and destinations) take place in parallel. This allows to perform one round of measures (between 140 nodes) in about 1 hour, including the timeouts when nodes respond slowly or fail during the procedure.

Description of the datasets We have performed the measurements every day for a week in November and December 2011, and in February 2012. Each of these datasets involves a hundred nodes. The large overlap

[‡] `http://www.planet-lab.org`

[§] `http://www.splay-project.org`

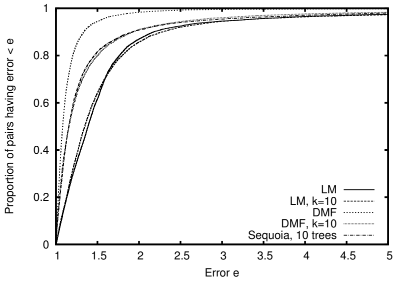

Figure 1: Comparison of heuristics on a single dataset

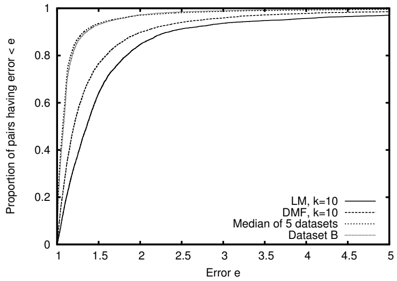

Figure 2: Prediction capability of previous datasets.

between participating nodes for all these measurements allow to see the evolution of network performance over time.We are currently working on improving our measurement procedure, to be able to use low-level measurement tools in addition to our application-level measures. This will ensure that the datasets produced are as comprehensive as possible.

3 The `bedibe` software

`Bedibe` is a tool for benchmarking bandwidth estimations. Its purpose is the development, testing, benchmarking and visualization of bandwidth estimation algorithms. It is written in Python, a very popular scripting language. It can be downloaded at `http://bedibe.gforge.inria.fr/`.

Data is read from CSV files (Comma Separated Values), with source and target hostnames, and measured values (this is also the format of SPLAY measures). Because network measures have high variation, datasets often contain multiple values per pair of nodes. Because of this, we provide a set of tools for easy picking representative value or set of values (in case of fuzzy computation) from several of them. We provide implementations of the state-of-the art bandwidth estimation algorithms: DMF, LastMile, Sequoia, DSequoia. We also provide a library of decorators (function transformation) that for example, allow to transform functions that operate on matrices into ones working in our enviroment of CSV style data. Other decorators allow easy control over what type of preprocessing an algorithm supports.

Estimations done by algorithms can be stored into CSV files, or used in several comparison tools. Our module for comparing allows easy comparison of the error ratio between different measures or estimations, or calculation of various standard metrics (stress, 80th percentile error...). Visualization tools allow to create images out of the computed matrices (assigning different colors to high/low bandwidth, or accurate/inaccurate estimation). Additionally, a plotting module can be used to produce (using `gnuplot`) different types of plots, for example a CDF of the dataset, or a CDF of relative error of the estimates. Various parameters of the plots can also be customized from inside the code.

In summary, a programmer simply needs to write an implementation of his algorithm, and a few extra lines of code allows the algorithm to be tested on real life data, tweaked, analyzed or compared to other algorithms from the literature. Finally, the tool can be used to output plots useful for performance visualization in a scientific paper.

Some results Figures 1 and 2 exhibit an example of the comparisons that our tool allows with minimal effort. On Figure 1, we plot the CDF of the error ratio (ie, $\max(\frac{m}{p}, \frac{p}{m}) = 1 + \frac{|m-p|}{\min m,p}$ where p is the prediction and m is the measured value) for 5 estimation heuristics: DMF, Last-mile, Sequoia, all three with access to the complete set of measurements, and also DMF and Last-mile restricted to only $k = 10$ measurements per node. On Figure 2, we analyze the prediction capability of a previous dataset B to predict the performance of the next dataset A. We also plot how well DMF and Last-Mile can predict the performance of dataset A when given dataset B as input. These results were obtained with very little coding effort thanks to the large panel of helper functions in the `bedibe` tool.

4 Concluding remarks

Estimating the available bandwidth between nodes in a large scale distributed platform is a crucial issue in many distributed applications. Several solutions have been proposed to perform this prediction while using as few measurements as possible. Validating and comparing these solutions requires that their implementation are available, preferably in a common framework. It is also necessary to use common datasets.

In this paper, we present `bedibe`, a Python framework to simplify the design, implementation and validation of bandwidth estimation algorithms. We also present datasets obtained from application-level measurements on the Planet-Lab platform using the SPLAY middleware. We hope that these tools will help the community to propose efficient solutions for bandwidth estimation, and will promote the diffusion of knowledge in this field.

In the future work, we plan to continue the development of the tool to increase further its usability. We also plan to make the list of featured algorithms as complete as possible. Last but not least, we plan to include several of the existing low-level measurement tools [GS10] in our measurement procedure, so that `bedibe` will feature the main solutions for available bandwidth measurement and prediction from the literature.

References

[BEDW11] O. Beaumont, L. Eyraud-Dubois, and Y. Won. Using the last-mile model as a distributed scheme for available bandwidth prediction. In *Proceedings of the EuroPar 2011 conference*, 2011.

[CWS$^+$11] Y. Chen, X. Wang, C. Shi, E. K. Lua, X. Fu, B. Deng, and X. Li. Phoenix: A weight-based network coordinate system using matrix factorization. *IEEE Transactions on Network and Service Management*, 8(4):334–347, December 2011.

[DCKM04] F. Dabek, R. Cox, F. Kaashoek, and R. Morris. Vivaldi: a decentralized network coordinate system. In *ACM SIGCOMM '04*, pages 15–26, Portland, OR, USA, Sept. 2004.

[Din08] M. Dinitz. Online, dynamic, and distributed embeddings of approximate ultrametrics. In *Distributed Computing '08*, volume 5218 of *LNCS*, pages 152–166. 2008.

[GS10] E. Goldoni and M. Schivi. End-to-end available bandwidth estimation tools, an experimental comparison. In *Traffic Monitoring and Analysis*, volume 6003 of *LNCS*, pages 171–182. 2010.

[HS05] N. Hu and P. Steenkiste. Exploiting internet route sharing for large scale available bandwidth estimation. In *IMC '05*, pages 16–16, Oct. 2005.

[LGL10] Y. Liao, P. Geurts, and G. Leduc. Network distance prediction based on decentralized matrix factorization. In *IFIP NETWORKING '10*, pages 15–26, May 2010.

[LRF09] L. Leonini, É. Rivière, and P. Felber. Splay: distributed systems evaluation made simple. In *Proceedings of the 6th USENIX symposium on Networked systems design and implementation*, NSDI'09, pages 185–198, Berkeley, CA, USA, 2009. USENIX Association.

[LZSJ$^+$08] S. Liu, R. Zhang-Shen, W. Jiang, J. Rexford, and M. Chiang. Performance bounds for peer-assisted live streaming. *ACM SIGMETRICS Perform. Eval. Rev.*, 36:313–324, June 2008.

[MSS06] Y. Mao, L.K. Saul, and J.M. Smith. Ides: An internet distance estimation service for large networks. *IEEE JSAC*, 24(12):2273 –2284, Dec. 2006.

[NZ02] T.S.E. Ng and H. Zhang. Predicting internet network distance with coordinates-based approaches. In *IEEE INFOCOM '02*, pages 170–179, June 2002.

[RMK$^+$09] V. Ramasubramanian, D. Malkhi, F. Kuhn, M. Balakrishnan, A. Gupta, and A. Akella. On the treeness of internet latency and bandwidth. In *ACM SIGMETRICS '09*, pages 61–72, Seattle, WA, USA, June 2009.

[SDK$^+$07] K. Suh, C. Diot, J. Kurose, L. Massoulie, C. Neumann, D. Towsley, and M. Varvello. Push-to-peer video-on-demand system: Design and evaluation. *IEEE JSAC*, 25(9):1706 –1716, Dec. 2007.

[SKBS11] S. Song, P. Keleher, B. Bhattacharjee, and A. Sussman. Decentralized, accurate, and low-cost network bandwidth prediction. In *INFOCOM, 2011 Proceedings IEEE*, pages 6 –10, april 2011.

[XCY09] C. Xing, M. Chen, and L. Yan. Predicting available bandwidth of internet path with ultra metric space-based approaches. In *IEEE GLOBECOM '09*, Dec. 2009.

[YSB$^+$06] P. Yalagandula, P. Sharma, S. Banerjee, S. Basu, and S.-J. Lee. S3: a scalable sensing service for monitoring large networked systems. In *ACM SIGCOMM workshop on Internet network management*, pages 71–76, Pisa, Italy, Sept. 2006.