Point-to-point and congestion bandwidth estimation: experimental evaluation on PlanetLab

Lionel Eyraud-Dubois, Przemyslaw Uznanski

Point-to-point and congestion bandwidth estimation: experimental evaluation on PlanetLab

Lionel Eyraud-Dubois and Przemysław Uznański

INRIA Bordeaux – Sud-Ouest; LaBRI, Université de Bordeaux
351, av. de la Libération, F-33405 Talence Cedex

Abstract. In large scale Internet platforms, measuring the available bandwidth between nodes of the platform is difficult and costly. However, having access to this information allows to design clever algorithms to optimize resource usage for some collective communications, like broadcasting a message or organizing master/slave computations.
In this paper, we analyze the feasibility to provide estimations, based on a limited number of measurements, for the point-to-point available bandwidth values, and for the congestion which happens when several communications take place at the same time. We present a dataset obtained with both types of measurements performed on a set of nodes from the PlanetLab platform. We show that matrix factorization techniques are quite efficient at predicting point-to-point available bandwidth, but are not adapted for congestion analysis. However, a LastMile modeling of the platform allows to perform congestion predictions with a reasonable level of accuracy, even with a small amount of information, despite the variability of the measured platform.

1 Introduction

In many Internet applications, network-awareness is an important part of achieving good performance or lowering resource usage. In the case of delivering video on demand [16], or performing peer-assisted streaming [12] for example, estimations of available bandwidth allow the construction of an efficient overlay topology. However, it is often not desirable to perform explicit end-to-end path measurements because of the high cost of such measurements. Also, the dynamicity of the platform implies that the representation of the platform is never up-to-date.

In order to perform resource optimization in large scale platforms, it is thus necessary to summarize the network performance in the platform, in a way that providing such a summary can be done with a reasonable amount of measurements. This idea has led to the design of Network Coordinate Systems (NCS), which embed the nodes of the platform in a metric space. Appropriate metric spaces and efficient algorithms have been proposed for estimating latency over the Internet (Vivaldi [5] is a good example). In this paper, we are interested in estimating available bandwidth between the nodes of the platform. Indeed, in many applications (such as content delivery or video streaming), large amounts of data need to be exchanged between nodes, making available bandwidth the important metric for application performance. Furthermore, when several large communications take place at the same time, they are expected to interfere with each other. Being able to predict and model this interference is also important for optimizing the resource usage of an application.

In this paper, our main focus is on the LastMile model [2] and on Decentralized Matrix Factorization (DMF) [10]. Both are good candidates for available bandwidth NCS, mainly because they are able to give asymmetric estimation, which is impossible for all NCS based on a metric space embedding. In Section 3, we analyze the estimation precision of both systems based on PlanetLab measurements from S-cube [17]. Then, in Section 4, we go beyond estimation of point-to-point performance and address the problem of congestion: *is it possible to predict the performance obtained when several communications take place at the same time?* The Last Mile model, used as a communication model in several algorithmic studies [3, 12, 1], actually specifies that communications can happen in parallel, as long as the outgoing or incoming bandwidth limits of each node are satisfied. In Section 4.1, we present dedicated measurements performed on PlanetLab specifically to study this question. In Section 4.3, we assess the accuracy of several estimation for total throughput in congestion scenarios, and we show that LastMile allows to perform this prediction with a reasonable accuracy. Finally, we give concluding remarks in Section 5.

2 Related works

Network Coordinate Systems have received a lot of attention recently, especially in the context of latency estimation. Original systems, like GNP [14], relied on landmarks to make the predictions – landmarks are special nodes whose positions are computed first, and all nodes of the system compute their position with respect to these landmark nodes. Afterwards, more distributed systems, like Vivaldi [5], have been designed, in which all nodes have the same role, leading to more precise and robust estimations. The term Network Coordinate System comes from the fact that all those systems embed the nodes in a metric space, hence assigning coordinates to all nodes, and use the distance in this space as an approximation of latency. A notable exception to this is the Matrix Factorization [10] technique, in which the rationale is to approximate the distance matrix by a low rank matrix by assigning a column and a row vector to each node of the system.

Matrix Factorization has been originally designed for latency estimation [13], and later extended to estimate network performance classes [11]. In this paper we are interested in estimating available bandwidth, and it seems natural to observe how well Matrix Factorization performs in this context.

Available bandwidth datasets are quite rare in the literature. The S-cube project [17], which measured available bandwidth between nodes of PlanetLab, is now discontinued, and we are not aware of other active similar projects. Several tools exist for measuring available bandwidth (*i.e.* the minimum remaining capacity on all links on the path) [7]. 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. However, using them on PlanetLab is not easy and is not reliable [8]. Furthermore, available bandwidth is not always related to the available TCP throughput, which is the metric that actually influence the performance of applications.

3 Distance Labeling for Bandwidth Estimation

In this section, we present two models (namely LastMile and DMF) which can be used to provide estimates for available bandwidth from a limited number of measurements. Then, we provide experimental evaluation for DMF and LastMile bandwidth estimation.

3.1 LastMile

The LastMile model is based on the assumption that contention only happens on the periphery of the network (on the “last-mile” link that connects each participating node to the network). This assumption is realistic in many scenarios (like for example when the system consists of home computers connected to the network by DSL connections), hence this model has been used in several studies to design or analyze communication algorithms for video-on-demand [3], peer-assisted streaming [12] or master-slave tasking [1]. The LastMile model assigns to each node i an outgoing bandwidth b_i^{out} and an incoming bandwidth b_i^{in}, and then the available bandwidth between two nodes $BW(i,j)$ can be computed easily:

$$BW(i,j) = \min(b_i^{out}, b_i^{in})$$

This LastMile model has been presented as a method for bandwidth estimation in [2], together with a distributed algorithm for dynamic computation of both b^{out} and b^{in} values. It was shown that the precision of such estimates was rather good compared to other estimation techniques like Vivaldi [5] and Sequoia [15]. The inability of such techniques to provide asymmetric estimates makes them unusable for bandwidth estimation.

3.2 DMF

Matrix Factorization has recently been proposed as a novel approach for distance estimation in the Internet [13]. This approach has a strong difference with the LastMile model, as it does not attempt to make any assumption on the underlying network. Rather, the objective is to approximate the bandwidth matrix (i.e. the n by n matrix $\mathcal{M}$ such that $\mathcal{M}_{i,j}$ is the measured available bandwidth between i and j) by a low rank matrix. In Matrix Factorization, we thus search for matrices P and Q (of dimension n by p and p by n respectively, where p is a fixed parameter) such that the product $P.Q$ is "close to" the measured matrix $\mathcal{M}$. If the closeness property is defined by the quadratic error $err(P,Q) = \sum_{i,j}(\mathcal{M}_{i,j} - (P.Q)_{i,j})^2$, then optimal P and Q can be computed from $\mathcal{M}$ by Singular Value Decomposition, or by iterative optimization if some values are missing.

A distributed algorithm has also been proposed [10], in which each node is in charge of its own values in matrices P and Q, and iteratively, nodes optimize their values based on the current values of their neighbors. It has been shown experimentally that this algorithm converges, and gives good estimates when considering latency values; this algorithm has also been used to perform classification of paths in either "good" or "bad" performance [11]. As far as we know, there is no experimental evaluation validation of DMF for bandwidth estimation.

3.3 Evaluation of LastMile and DMF

In this section, we present an evaluation of the precision of LastMile (both in its plain and iterated versions, as described in [2]) and DMF, obtained using a software named `bedibe`[1].

To evaluate the precision of estimation algorithms, we use the standard *relative error* metric, which is defined as $e = \max(\frac{p}{v}, \frac{v}{p})$, where p is the predicted value and v is the actual measured value. This relative error is computed for every source/destination pair, and we are interested in the *distribution* of all relative error values on all pairs.

We compare DMF and LastMile using two datasets: the first one is a snapshot of the PlanetLab platform of April 2th, 2010 taken from the (now discontinued) S-cube project [17], which contains available bandwidth measurements between 426 hosts, with many missing measurements, and we extracted a set of 308 hosts for which the complete measurement matrix is available. The second one is obtained with `bedibe` measurement methodology (described in Section 4.1), and contains achievable TCP throughput between 50 hosts of PlanetLab. In this dataset, about 17% of measurements are unavailable. In order to account for the fact that performing all end-to-end measurements is not reasonable in a practical setting, a subset of 20% of values are randomly chosen and given to the estimation algorithms. Note however that algorithms provide estimations for all values, and that the relative errors are computed on the whole matrix (for comparison, we also provide plots how algorithms perform given full matrix as an input).

These results are shown on Figure 1, where we plot the Cumulative Distribution Function of relative error for each estimation algorithm. For example, a point at coordinates $(1.5, 0.8)$ for DMF on the S-cube dataset means that for this

[1] `bedibe` (described in more details in [6]) 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, and can be downloaded at http://bedibe.gforge.inria.fr/.

dataset, 80% of all source/destination pairs are predicted with an error below 1.5. Hence, plots closer to the upper left corner of the graph represent better estimations.

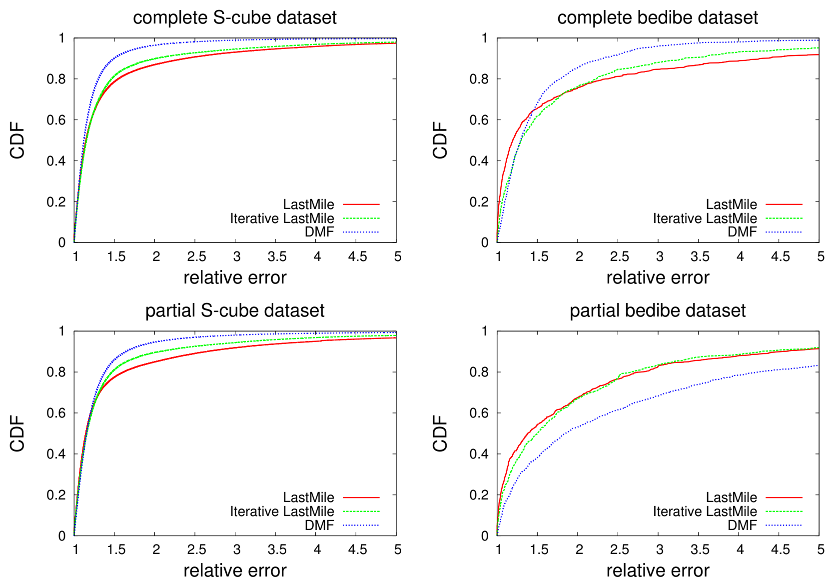

Fig. 1. Comparison of performance of DMF and LastMile, done on S-cube dataset (on the left), `bedibe` dataset (on the right), full information (top) and partial information (bottom).

The analysis of Figure 1 shows that:

- On the S-cube dataset, DMF clearly outperforms other algorithms, and provides reasonably good predictions, even when only partial information is available. We can also see, as in [2], that the iterated improvement of LastMile helps to obtain better predictions.
- On the `bedibe` dataset, DMF is only able to provide reasonable predictions if given the full matrix, and its performance drops by a large factor when given only partial data. The precision of LastMile suffers a much smaller drop and outperforms DMF on partial data.
- The `bedibe` dataset is harder to estimate than the S-cube dataset: the 80th percentile relative error is less than 1.5 for all algorithms on the S-cube dataset, but is around 2 for the `bedibe` dataset.

4 Evaluating bandwidth sharing on PlanetLab

In this section, we evaluate the ability of bandwidth estimation tools to predict throughput in the presence of congestion. We first describe the setting which we have used to perform the measurements for this study, and provide some analysis of this data to show its usability. Then, we use this data to assess the precision that can be achieved for total throughput with different estimation methods.

4.1 Measurement methodology

The measures have been performed on the PlanetLab platform[2] [4]. PlanetLab 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.

The measurements were performed using the SPLAY middleware[3] [9], which aims at simplifying the prototyping and development of large scale distributed applications and overlay networks. It is based on Lua language and provides tools for deploying and controlling a distributed application on a large platform.

As mentioned above, in this paper we focus on application-level measurements, in order to observe the platform as it would be accessible to the application. Hence, we measure available TCP throughput, which is the steady-state reachable throughput that can be achieved with a TCP connection.

Experiment design We performed two types of experiments: individual end-to-end throughput measurements, as well as contention experiments for measuring the performance achieved when multiple communications take place at the same time. To keep them as simple as possible, we concentrated on the particular situation with one sender and two receivers, which on the considered platform is enough to generate contention, and thus allows to capture congestion and sharing mechanisms. The situation with two senders and one receiver would be interesting to observe as well, and is left for future work.

In order to make sure that we observe a steady-state, and thus avoid the slow-start mechanisms of TCP, the experimental setting for individual end-to-end measures was the following: data is sent from the sending node to the receiving node on a TCP socket for 20 seconds. The first 15 seconds are note measured, and only used to "warm-up" the connection. The receiver measures how much data is received for the last 5 seconds, and uses this value to compute an average throughput over these 5 seconds. Contention experiments used a similar setting, with both receivers measuring how much data they receive on the last 5 seconds.

[2] http://www.planet-lab.org
[3] http://www.splay-project.org

With the dynamic nature of the PlanetLab platform, both kinds of measurements suffer from a high variability. In order to overcome this variability, we perform at least 10 repetitions of the measurement for each configuration (same sender-receiver pair for end-to-end measures, and same sender-receivers triplet for contention measures). The variability of measures is analyzed in Section 4.2.

PlanetLab limitations Our choice to perform the experiments on PlanetLab influenced heavily how they were performed. The PlanetLab platform itself, because of the fact that it is shared among a large number of users, comes with a number of restrictions.

In order to avoid flooding, PlanetLab has a policy that each node has a daily data transfer limit. Together with our measurement methodology which requires to send data for 20 seconds to perform one measurement, this means that it takes several days to gather an exhaustive and broad enough dataset, even for a small number of nodes. Hence, the number of nodes must be kept at reasonable size if we want our datasets to span relatively short periods of time.

PlanetLab nodes are under heavy usage, both in terms of CPU and bandwidth. Because of heavy CPU usage, and special process scheduler, time measures are unreliable (under 10ms error), which makes latency measures over network unpractical, and makes it very difficult to use state-of-the art available bandwidth estimation tools, which rely on precise timings of packet arrival times. Our bandwidth measures however do not suffer from this problem, since we measure time at a much coarser scale. Heavy bandwidth usage from other PlanetLab users yields a very high variability of the measures, even when performing one measure just after another.

The PlanetLab platform is not a "typical" Internet platform, in the sense that it consists of servers hosted by universities or research institutions, often connected through high-speed and high-bandwidth academic networks. They are also usually close to the main Internet routes. It is important to keep in mind that this platform is thus not representative of a typical peer-to-peer situation. However, its size and geographically distributed nature make it interesting to observe and analyze it. Furthermore, we can see it as a "worst-case" for the LastMile model in particular, since typical peer-to-peer platforms with DSL connected nodes are likely to be closer from a LastMile model than PlanetLab.

Datasets The measures performed are grouped in two datasets. The first dataset contains our end-to-end bandwidth measures. It was obtained by randomly selecting 50 nodes of PlanetLab, among which we performed measures over the course of one month, with the objective of having 10 measures for each sender/receiver pair. Because of node unavailability and connection problems, we could not obtain a complete matrix. A subset of 15 nodes was selected, among which it was possible to obtain a complete set of measurements. This data was collected between December 20th, 2012 and January 16th, 2013.

The second dataset contains our congestion measurements. It consists of 87 measures of bandwidth shared between triplets of nodes (one sender and two receivers), where each measure over each triplet was performed 10 times in about 10 minutes of time. Triplets were selected at random among the set of 15 nodes which had been selected for the complete end-to-end measurements as described above. This data was collected between January 4th, 2013 and February 4th, 2013.

Both of these datasets are available as part of `bedibe`, and can be downloaded (together with the code used to obtain the plots of this paper) at the following address: https://gforge.inria.fr/frs/download.php/32092/data.zip.

4.2 Data analysis

Variability of measurements In order to evaluate the variability of our measurements, we focus on a subset of 15 nodes, for which we have performed a larger number of measurements. Between all pairs of those nodes, a total of 10 individual bandwidth measurements were performed. This allows to compute mean and standard deviation for each given source/destination pair, and we express how variable the results of a measurement between two given nodes can be by computing the relative standard deviation (the standard deviation divided by the mean). On Figure 2, we can see that some pairs feature very stable measurements. However, in many cases the variability is high, with relative standard error around or above 0.5. This variability is not a surprise, given how the PlanetLab platform is shared among many users, and this shows that providing estimates for available bandwidth is certainly challenging.

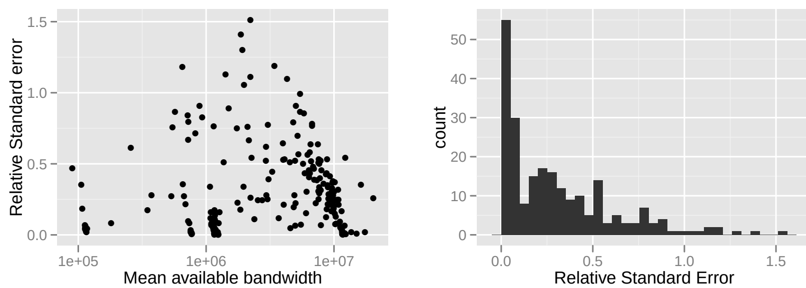

Fig. 2. Variability of individual measurements: relative standard deviation as a function of mean available bandwidth (on the left), and distribution of relative standard deviation among all source/destination pairs (on the right).

Bandwidth sharing measurements In the bandwidth sharing experiments, we measure the throughput received by the two receivers. Here also, we want to observe the variability of these measurements. For each configuration (i.e. for each choice of one sending and two receiving nodes), the measure was performed 10 times, and reports the throughput received by each receiving node (b_1 and b_2, as well as the sum of these throughputs $b_{tot} = b_1 + b_2$. Similarly to the previous paragraph, for each configuration we compute the relative standard error of b_1, b_2 and b_{tot}, as an indication of how variable these measures are. The results are shown on Figure 3, and we can see that the total throughput is much more stable than the individual throughput received by a given node. This high variability for individual values is the reason why this paper does not attempt to predict how the sharing is done between the two receivers, but instead focuses on predicting the total throughput.

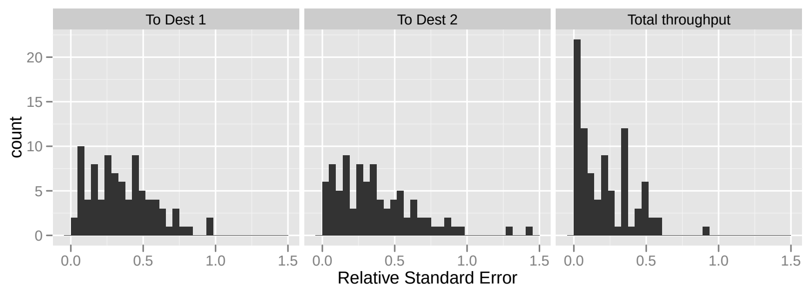

Fig. 3. Variability of sharing measurements: distribution of the relative standard error of the throughput received by the first node, by the second node, and of the total throughput received by both of them.

4.3 Predicting total throughput in congestion scenarios

In this section, we study the possibility to predict the total achievable throughput when a sending node sends data to several receiving nodes. In this paper however, we restrict to the case of two receiving nodes, because this is already enough to achieve congestion and thus obtain meaningful results.

For this study, we use both `bedibe` datasets described above. We try to predict the value $TT_{i,j,k}$, being the amount of data node i is able to transmit to nodes j and k simultaneously. For this prediction, we tried several possibilities:

- **LastMile**: we compute LastMile values (incoming and outgoing parameters) for all nodes, and we use the LastMile assumption to predict the available bandwidth: it is either limited by the sending capacity of the sender, or by the sum of the receiving capacities of the receivers: $\mathcal{P}_{i,j,k}^{LM} = \min(b_i^{out}, b_j^{in} + b_k^{in})$.
- **Avg, Sum, Max**: we use end-to-end individual measurements between the sender and each of the receivers, and use the average, total, or maximum

value as a prediction:

$$\mathcal{P}^{avg}_{i,j,k} = \frac{BW(i,j) + BW(i,k)}{2} \qquad \mathcal{P}^{sum}_{i,j,k} = BW(i,j) + BW(i,k)$$
$$\mathcal{P}^{max}_{i,j,k} = \max(BW(i,j), BW(i,k))$$

– **AvgDMF, SumDMF, MaxDMF**: we compute the DMF predictions for the available bandwidth between the sender and each of the receivers, and similarly use the average, total, or maximum value as a prediction.

We then compute the relative error of each prediction, with the same definition as in Section 3: $e^X_{i,j,k} = \max(\frac{TT_{i,j,k}}{\mathcal{P}^X_{i,j,k}}, \frac{\mathcal{P}^X_{i,j,k}}{TT_{i,j,k}})$. To evaluate all prediction techniques, we thus analyze the distribution of the error ratios for all the measured triplets in our experiments. The results are shown on Figure 4, and mean and median values are reported in Table 1.

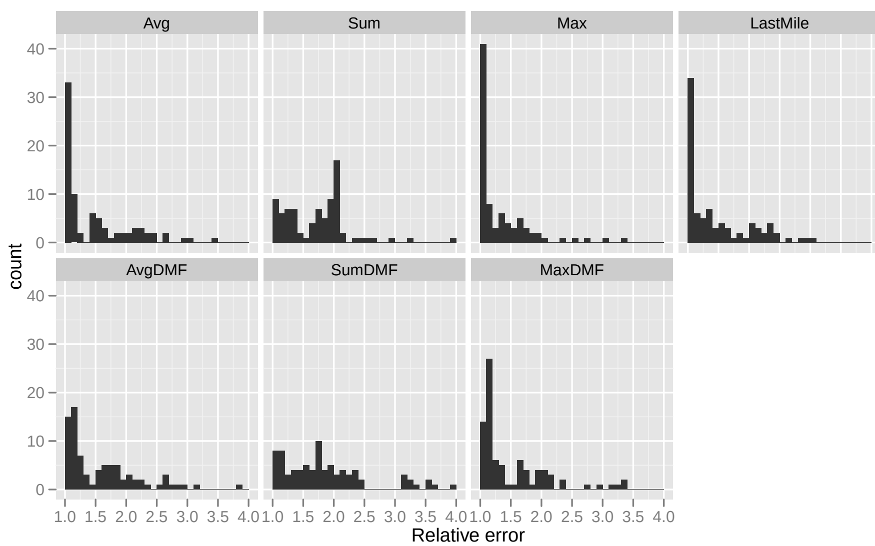

Fig. 4. Distribution of error ratios for 7 estimates of total throughput when one sender sends to two receivers.

The results show that **Sum** and **SumDMF** provide very bad estimations. This comes from the fact that in PlanetLab nodes, the congestion often takes place at the sending node, even with only two receivers. Hence summing the individual performance of both receivers yields a large over-estimate of the actual total throughput. This explains why using average or maximum values give better predictions. Actually, using the **Max** estimate gives the best results in

Est.	LM	Avg	Sum	Max	AvgDMF	SumDMF	MaxDMF
Mean	**0.851**	1.32	1.62	0.983	1.89	2.04	0.941
Median	0.236	0.217	0.823	**0.15**	0.564	0.755	0.218

Table 1. Mean and median relative error for 7 estimation techniques

most cases, because the maximum measured individual throughput is very often close to the outgoing bandwidth of the sender. This explains that the median error ratio of **Max** is much lower than all other estimates. For the same reasons, **MaxDMF** also has a rather good median error ratio, not as good as **Max** because of the imprecisions incurred by DMF. However, in some cases **Max** and **MaxDMF** provide estimates which are off by a larger factor, whereas **LastMile** predictions are more stable, as can be seen by the lower mean error ratio.

It is important to note that in a practical setting, **Max** estimates can only be obtained by actually performing both individual end-to-end measures, whereas by design **LastMile** and **MaxDMF** can be computed for all possible triplets by using only a smaller number of measurements. Furthermore, from an algorithmic point of view, these results encourage the use of the LastMile model for the design of bandwidth allocation algorithms, even in settings where the "last-mile" assumption is not completely valid.

5 Conclusions

In this paper, we analyze the possibility to provide estimations for available bandwidth in large scale platforms. Estimation techniques and models exist for the latency metric, but their extension to available bandwidth is not always possible. We focus on the LastMile model, widely used as a communication model in algorithmic works, and on Decentralized Matrix Factorization (DMF), originally designed for latency estimation. We show that DMF is able to provide very good estimations for available bandwidth as well, but that its precision drops when fewer measurements are available. Furthermore, we analyze contention in the presence of multiple simultaneous communications. We performed experiments on the PlanetLab platform, and we observed that the total throughput can be predicted by the LastMile model with a precision almost comparable to estimations based on the complete set of measurements.

References

1. Beaumont, O., Eyraud-Dubois, L., Rejeb, H., Thraves, C.: Allocation of clients to multiple servers on large scale heterogeneous platforms. In: Parallel, Distributed and Network-Based Processing (PDP), 2010 18th Euromicro International Conference on. pp. 3 –10 (feb 2010)
2. Beaumont, O., Eyraud-Dubois, L., Won, Y.: Using the last-mile model as a distributed scheme for available bandwidth prediction. In: Proceedings of the EuroPar 2011 conference (2011)

3. Bonald, T., Massoulié, L., Mathieu, F., Perino, D., Twigg, A.: Epidemic live streaming: optimal performance trade-offs. ACM SIGMETRICS Perform. Eval. Rev. 36, 325–336 (June 2008), http://doi.acm.org/10.1145/1384529.1375494
4. Chun, B., Culler, D., Roscoe, T., Bavier, A., Peterson, L., Wawrzoniak, M., Bowman, M.: Planetlab: an overlay testbed for broad-coverage services. ACM SIGCOMM Comput. Commun. Rev. 33, 3–12 (July 2003), http://doi.acm.org/10.1145/956993.956995
5. Dabek, F., Cox, R., Kaashoek, F., Morris, R.: Vivaldi: a decentralized network coordinate system. In: SIGCOMM '04: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications. pp. 15–26. ACM, New York, NY, USA (2004), http://doi.acm.org/10.1145/1015467.1015471
6. Eyraud-Dubois, L., Uznanski, P.: Bedibe: Datasets and software tools for distributed bandwidth prediction. In: AlgoTel'2012-Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications (2012)
7. Goldoni, E., Schivi, M.: End-to-end available bandwidth estimation tools, an experimental comparison. In: Ricciato, F., Mellia, M., Biersack, E. (eds.) Traffic Monitoring and Analysis, Lecture Notes in Computer Science, vol. 6003, pp. 171–182. Springer Berlin Heidelberg (2010), http://dx.doi.org/10.1007/978-3-642-12365-8_13
8. Lee, S.J., Sharma, P., Banerjee, S., Basu, S., Fonseca, R.: Measuring bandwidth between planetlab nodes. In: Dovrolis, C. (ed.) Passive and Active Network Measurement, Lecture Notes in Computer Science, vol. 3431, pp. 292–305. Springer Berlin Heidelberg (2005), http://dx.doi.org/10.1007/978-3-540-31966-5_23
9. Leonini, L., Rivière, E., Felber, P.: Splay: distributed systems evaluation made simple. In: Proceedings of the 6th USENIX symposium on Networked systems design and implementation. pp. 185–198. NSDI'09, USENIX Association, Berkeley, CA, USA (2009), http://dl.acm.org/citation.cfm?id=1558977.1558990
10. Liao, Y., Geurts, P., Leduc, G.: Network distance prediction based on decentralized matrix factorization. In: IFIP NETWORKING '10. pp. 15–26 (May 2010)
11. Liao, Y., Du, W., Geurts, P., Leduc, G.: Decentralized prediction of end-to-end network performance classes. In: Proceedings of the Seventh COnference on emerging Networking EXperiments and Technologies. p. 14. ACM (2011)
12. Liu, S., Zhang-Shen, R., Jiang, W., Rexford, J., Chiang, M.: Performance bounds for peer-assisted live streaming. ACM SIGMETRICS Perform. Eval. Rev. 36, 313–324 (June 2008), http://doi.acm.org/10.1145/1384529.1375493
13. Mao, Y., Saul, L.K.: Modeling distances in large scale by matrix factorization. In: IMC '04. pp. 278–287. ACM (Oct 2004)
14. Ng, T., Zhang, H.: Predicting internet network distance with coordinates-based approaches. In: IEEE INFOCOM '02. pp. 170–179 (June 2002)
15. Ramasubramanian, V., Malkhi, D., Kuhn, F., Balakrishnan, M., Gupta, A., Akella, A.: On the treeness of internet latency and bandwidth. In: SIGMETRICS '09: Proceedings of the eleventh international joint conference on Measurement and modeling of computer systems. pp. 61–72. ACM, New York, NY, USA (2009)
16. Suh, K., Diot, C., Kurose, J., Massoulie, L., Neumann, C., Towsley, D., Varvello, M.: Push-to-peer video-on-demand system: Design and evaluation. IEEE JSAC 25(9), 1706 –1716 (Dec 2007)
17. Yalagandula, P., Sharma, P., Banerjee, S., Basu, S., Lee, S.J.: S3: a scalable sensing service for monitoring large networked systems. In: Proceedings of the 2006 SIGCOMM workshop on Internet network management. pp. 71–76. INM '06, ACM, New York, NY, USA (2006), http://doi.acm.org/10.1145/1162638.1162650