Large scale probabilistic available bandwidth estimation

Frederic Thouin[a,*], Mark Coates[a], Michael Rabbat[a]

[a]*McGill University, Department of Electrical and Computer Engineering, 3480 University, Montreal, Quebec, Canada, H2A 3A7*

Abstract

The common utilization-based definition of available bandwidth and many of the existing tools to estimate it suffer from several important weaknesses: i) most tools report a point estimate of average available bandwidth over a measurement interval and do not provide a confidence interval; ii) the commonly adopted models used to relate the available bandwidth metric to the measured data are invalid in almost all practical scenarios; iii) existing tools do not scale well and are not suited to the task of multi-path estimation in large-scale networks; iv) almost all tools use ad-hoc techniques to address measurement noise; and v) tools do not provide enough flexibility in terms of accuracy, overhead, latency and reliability to adapt to the requirements of various applications. In this paper we propose a new definition for available bandwidth and a novel framework that addresses these issues. We define *probabilistic available bandwidth* (PAB) as the largest input rate at which we can send a traffic flow along a path while achieving, with specified probability, an output rate that is almost as large as the input rate. PAB is expressed directly in terms of the measurable output rate and includes adjustable parameters that allow the user to adapt to different application requirements. Our probabilistic framework to estimate network-wide probabilistic available bandwidth is based on packet trains, Bayesian inference, factor graphs and active sampling. We deploy our tool on the PlanetLab network and our results show that we can obtain accurate estimates with a much smaller measurement overhead than Pathload.

Keywords: Bayesian inference, active sampling, belief propagation, network monitoring.

1. Introduction

Recent work has shown that the performance of applications such as overlay network routing [1, 2] and anomaly detection [3] can be improved significantly when the network-wide available bandwidth is known. There are many more applications (SLA compliance, network management, transport protocols, traffic engineering, admission control) that could also benefit from this information, but existing tools that measure available bandwidth generally do not meet the requirements of these applications in terms of accuracy, overhead, timeliness and reliability [4]. In addition to the lack of flexibility, existing models and tools suffer from four other major weaknesses:

1. The vast majority report a single value representing average available bandwidth and the usefulness of this single value is questionable. Available bandwidth is typically defined as the capacity of a path unused by cross-traffic over a specified time period. Most tools produce a single point estimate of the available bandwidth by making multiple measurements using probes sent throughout the time period of interest. The cross-traffic often fluctuates significantly over the time period, so probes experience very different network conditions; an estimate formed from such data can be a high-variance quantity making a confidence interval very valuable. Service (or response) curves are more informative than single average estimates; they present the statistical mean (asymptotic average) of the output rate for an entire range of input rates [5]. However, each point on the curve is still an average that does not really provide a meaningful reflection of the burstiness of the traffic and the variability of the available bandwidth metric. A more robust and practically-relevant manner to express the available bandwidth is the variation range (confidence interval) proposed by Jain and Dovrolis [6].
2. The observation model relating measured data to the utilization-based definition of available bandwidth is inaccurate and biased in most practical situations. As a result, the value provided by most tools does not genuinely reflect the quantity the tools claim to estimate. The fluid cross-traffic assumption underpins the vast majority of models used for inference. Liu et al. [5] show that the assumed relationships between the measured quantities (packet dispersion, one-way delay, output rate) and the estimated value (utilization, cross-traffic) are not sound; even for simple, slightly more realistic scenarios, the adoption of a fluid model leads to significant underestimates of the available bandwidth.
3. The mechanisms used by most tools to handle measurement noise are ad-hoc and, in many cases, inadequate. Measurement errors and noise generated by the end-hosts and routers along the end-to-end path are unavoidable in practice. Common issues include route changes, out-of-order packet delivery, packet replications, and errors in

*Corresponding author. Tel.: +1-514-398-5516. Fax: +1-514-398-3127

Email addresses: frederic.thouin@mail.mcgill.ca (Frederic Thouin), mark.coates@mcgill.ca (Mark Coates), michael.rabbat@mcgill.ca (Michael Rabbat)

the probing packets due to link quality issues, incorrect packet time stamps, and poor Network Interface Card utilizations. Although measures can be adopted to prevent some of these errors, it is impossible to eradicate them all. It is important that the model and inference technique are robust, and that they can tolerate and handle noisy measurements. One example of a technique that does handle noise more robustly is Traceband [7], which employs a hidden Markov model that allows the technique to statistically adjust to noise in the measurements.

4. Current tools cannot be applied to larger networks to simultaneously estimate the available bandwidths of multiple paths. Using existing tools, probing all paths concurrently not only introduces an unacceptable overhead and overloads hosts, but also leads to significant underestimation due to interference between the probes on links shared by multiple paths [8]. The alternative to simultaneous measurements is to sequentially probe each path independently. This is unacceptably time-consuming and very inefficient, however, because it ignores the significant correlations that arise in available bandwidth metrics when the network paths share links.

In this paper, we tackle the problem of network-wide (multi-path) available bandwidth estimation. In developing our approach, we strive to address the issues we have identified above. Our solution includes i) a probabilistic-rate-base definition for the available bandwidth and ii) a network-wide estimation tool. Our implementation uses the Bayesian inference framework, factor graphs and the belief propagation algorithm to fuse the information obtained from all measurements. We adopt a model that relates the PAB of each path to the PAB of its constituent links; the factor graph provides a mechanism for capturing this model and enables computationally efficient inference. These techniques have been successfully used in large-scale network problems, such as link loss inference applications [9, 10] and the computation of conditional entropies for both fault diagnosis and most informative test selection [11–13], but not yet in the context of available bandwidth estimation.

Another novel contribution is our algorithm to determine which path and rate to probe at each iteration; a process that can be related to sequential Bayesian sampling [14] and active/adaptive sampling [15]. This sampling strategy consists of selecting the next measurement(s) based on the information acquired previously, such that the expected information gain is maximized. In networking, it has been used in the context of network tomography to determine the measurements that provide the best information gain about the network path property given their probing overhead [16], but has yet to be applied to available bandwidth estimation.

The rest of this paper is organized as follows. In Sect. 2, we review existing techniques to estimate available bandwidth. In Sect. 3, we introduce a new metric, *probabilistic available bandwidth*. In Sect. 4, we formally state the estimation problem. In Sect. 5, we detail our novel probabilistic framework, which is the first to combine factor graphs and active sampling to estimate available bandwidth. In Sect. 6, we present results from our simulations and online experiments on the PlanetLab network. In Sect. 7, we summarize our contributions and discuss future work.

2. Background and Related Work

Each link in a network has a physical capacity c_ℓ, which does not change over time, that represents the maximal rate at which data can be transmitted on that link. The end-to-end capacity of a path c_p is equal to the smallest link capacity among all its constituent links. As long as the set of links on the path do not change, this value is a constant. The link on a path that has the smallest capacity is called the narrow link. The *available bandwidth* $A(t)$ is the unused portion of the capacity[1]. Let $\lambda(t)$ be the instantaneous rate of cross-traffic in bps on a link or path and $u(t) = \lambda(t)/c$ the fraction of the capacity of a link or path used by cross-traffic. The available bandwidth of a link $A_\ell(t)$ can thus be expressed both in terms of capacity and cross-traffic, $c_\ell - \lambda_\ell(t)$, or capacity and utilization, $c_\ell(1 - u_\ell(t))$. For a multi-hop path, the available bandwidth $A_p(t)$ is equal to the smallest available bandwidth among all links that constitute the path. For each path, the tight link is defined as the link along the path with the smallest available bandwidth. A link might be tight on one path, but not necessarily on an other. Also, the status of tight link can change at any instant t; cross-traffic increasing/decreasing on the link or other links on shared path, or changes in routing matrices that change the set of links of a path.

The most popular estimation tools are founded on either the probe-gap (PGM) or probe-rate model (PRM). The PGM assumes a single-hop path with FIFO queuing and fluid cross-traffic[2]. One measurement consists of sampling cross-traffic by observing the gap between a packet pair at both the input and the output. With every measurement, a single point estimate of the available bandwidth can be produced as long as i) the capacity of the tight link is known, ii) there is only one tight link and it is the same as the narrow link and iii) the end-nodes can transmit faster than the available bandwidth. PGM-based tools (e.g., Delphi [17], IGI [18], Spruce [19], ABwE [20], Traceband [7]) are lightweight and fast, but are unable to estimate with acceptable accuracy the available bandwidth of multi-hop paths [21]. The probe-rate model (PRM) also assumes fluid cross-traffic, but is more robust. The PRM relies on the principle of self-induced congestion probing [22]: if probes are sent at a rate smaller than the available bandwidth then the output rate matches the probing rate. However, if the probing rate is greater than the available bandwidth, packets get queued, which results in unusual delays and a smaller output rate. Algorithms constructed using the PRM (e.g., TOPP [23, 24], Pathload [6], pathChirp [22], PTR [18], Yaz [25], ASSOLO [26]) consist

[1] Dynamic (or time-varying) metrics, such as available bandwidth, can be expressed either as averages or instantaneous values. Unless otherwise specified, time-varying metrics will be expressed as instantaneous values.

[2] Traffic is modelled as a continuum of infinitely small packets with an average rate that changes slowly.

of varying the probing rate to identify the boundary that separates the two different behaviours described above: an input rate where probes start experiencing unusual delays. According to various empirical studies, these methods generate more accurate estimates than PGM-based tools, but they are also more intrusive because they require multiple iterations at different probing rates [4, 27–29].

Although the PRM does not require information about the path capacity (unlike the PGM), it involves sending probes at a rate as high as the available bandwidth, which can result in a significant load on the network. To address this issue, Neginhal et al. [30] propose Forecaster; a technique that sends probing streams at rates lower than the available bandwidth and measures the fraction of packets that experienced queuing to estimate the available bandwidth. Another weakness of the PGM and PRM is that they use a single hop model for paths or model multi-hop as a sequence of independent hops. Hága et al. [31] develop a new framework based on traffic flows, which enter and leave at arbitrary hop, to model multi-hop paths; dispersion curve can be calculated through hop by hop iteration of the output spacing and the effective probe packet size. Another multi-hop model is presented by Liebeherr et al. [32, 33]; they apply the convolution operator of the min-plus algebra to compute the service curves of end-to-end paths using the available bandwidth of multiple links. They use service curves to explain how bandwidth estimation methods infer information about a network and observe that maximal information is obtained at a point where the network crosses from a linear to non-linear regime.

Although these techniques can provide accurate results for single-path available bandwidth estimation, they cannot be applied directly to estimate multiple paths simultaneously. The multi-path estimation problem can be related to large-scale network inference. One of the most promising techniques for performing large-scale network inference is *network tomography*[3], which consists of estimating either i) link-level parameters based on end-to-end measurements; or ii) path-level traffic intensity based on link-level traffic measurements [35]. However, there are two key differences. First, tomography involves a mapping from path-level measurements to link-level metrics or vice versa; in the network-wide available bandwidth problem we are interested in estimating path-level metrics from path-level measurements. Second, in most network tomography problems, there is a linear relation of the form $y = Ax$ between measurements y and network parameters x, where A is a routing matrix. In our problem, this relationship is non-linear; one of our modelling assumptions is that the available bandwidth of a path is the minimum of the available bandwidths of all its constituent links.

The task is more closely related to the problem of *network kriging* [36], which involves estimating (functions of) path-level metrics throughout a network using end-to-end path measurements. This problem was also addressed in [37, 38], where an algebraic approach was proposed for exactly recovering, under the assumption of no noise, the path level metrics of all the end-to-end paths in a network by monitoring only a small subset of the paths. The method in [36] reduced this monitoring cost even further, at the expense of introducing a small error in the estimated metrics. For real-time applications, estimates must not only be produced with minimal overhead, but also in a timely manner. To meet these requirements, measurements, even for a reduced subset of paths, must be scheduled at the same time. To avoid simultaneous probes interfering with each other and overloading nodes, Song and Yalagandula [39] propose a resource-aware technique that achieves better accuracy than resource-oblivious methods at the cost of using more measurement data. All of these approaches, as well as the wavelet-based methodology described in [40], are only appropriate for (approximately) additive metrics, such as loss or delay, where a linear relationship can be constructed between the link-level and path-level metrics. However, Song and Yalagandula [39] suggest that their approach could be extended to available bandwidth estimation by selecting paths such that the load of their probes only represents a small fraction of the capacity of each link.

Large-scale (multi-path) estimation of available bandwidth has not received as much attention as other metrics. To limit measurement overhead, BRoute [41] capitalizes on the spatial correlation between links shared by many paths and the observation that 86% of Internet bottleneck links are within four hops (end-segments) from end nodes [42]. The tool first uses traceroute landmarks to identify AS-level end segments for each node, and then measures available bandwidth on these segments by using landmarks with high downstream bandwidth. Maniymaran and Maheswaran [43] propose a more efficient landmark-based approach that is similar to BRoute but has reduced storage and inference complexity. Another approach to large scale available bandwidth estimation is to exploit the correlation between various metrics (route, number of hops, capacity and available bandwidth); since the measurement cost for each metric is different, monitoring those that have a cheaper cost can reduce the load on the network [44]. To further reduce the amount of probing overhead, Man et al. [45] propose to reshape existing TCP traffic to look like packet pairs, trains or chirps so that no extra traffic is injected in the network. Despite these efforts to minimize the overhead of the estimation procedure, most of these network-wide tools do not address any of the concerns mentioned earlier; they are neither flexible nor robust to noisy measurements, they produce a single average value for each path and they are based on an invalid mapping between measurements and the inferred metrics.

3. Probabilistic Available Bandwidth

We specify the *probabilistic available bandwidth* (PAB) metric directly in terms of input rates and output rates of traffic on a path. We are interested in determining the largest input rate r_p at which we can send a traffic flow along a path while achieving an output rate r'_p that is almost (within ϵ) as large as the input

[3] See [34] and references therein for a review of network tomography.

rate, with specified probability[4] at least γ. More formally, for given $\epsilon > 0$ and $\gamma > 0$, we seek the largest input rate such that $\Pr(r'_p > r_p - \epsilon) \geq \gamma$. We denote the largest such ingress rate by y_p and refer to it as the probabilistic available bandwidth for path p:

$$y_p = \max\{r_p : \Pr(r'_p > r_p - \epsilon) \geq \gamma\}.$$

The probabilistic available bandwidth is located at the boundary of two regions with different behaviours (i.e., where we can expect different outputs). For smaller rates, $r_p \leq y_p$, there is a probability greater or equal to γ that the output rate will be within a margin of ϵ of the input rate. For input rates greater than the PAB, $r_p > y_p$, this probability is not guaranteed.

The values of the two parameters ϵ and γ are defined by the user based on application requirements and the network environment. The rate difference ϵ represents the tolerance in reduction between the input and output rate. In a network where there is high variability in the amount of cross-traffic or frequent packet drops, it can be preferable to choose a larger value of ϵ such that the value of the PAB remains more stable. This choice also depends on the rate reduction an application can tolerate. For applications such as admission control and SLA compliance that require high accuracy, a smaller ϵ would be a better choice. The parameter γ controls the tightness of the probabilistic guarantee provided by the PAB. If the user transmits a flow with rate y_p or less, then he (effectively) has probability $1 - \gamma$ of causing congestion. The application he has in mind, and the extent to which he is willing to risk causing congestion in the network, will dictate the value of γ. It is important to mention that there are no optimal values for these parameters; the methodology we present in this paper is valid for any choice of ϵ and γ. The impact of this choice will become clearer when we describe the likelihood function in Sect. 5.3.3.

We believe that this new definition for available bandwidth is more robust and practical for several important reasons. First, it provides a more valid mapping between the measured and inferred quantities. By expressing available bandwidth directly in terms of the input and output rates, there is no longer a need to bridge the gap between packet dispersion and utilization (or cross-traffic) through generally invalid modelling assumptions. Second, the probabilistic framework gives flexibility to the user and is more resistant to variability (cross-traffic burstiness) and noise (errors) in the measurements. Last, it represents a more practical and concrete quantity: the probability that transmitting data at a given rate will yield the desired output rate.

4. Problem Statement

We focus on the problem of network-wide available bandwidth estimation, but in terms of our newly introduced metric, probabilistic available bandwidth. More formally, for a specified (ϵ, γ) and network that consists of a set of N links $\mathcal{L}$ and M paths $\mathcal{P}$, we wish to form estimates of the probabilistic available bandwidths of all paths in the network. Let the PAB of each path p be modelled as a discrete[5] random variable y_p; e.g., $\Pr(y_p = r)$ being the probability that the PAB on path p is r. Then at any given instant k, our goals are to 1) identify and execute the most informative measurement z_k and 2) compute marginal posteriors $\Pr(y_p|\mathbf{z})$ for every path p, where $\mathbf{z} = [z_1, \ldots, z_k]$ is the vector of k collected measurements (one at each time step). Rather than forming a point estimate of the PAB, our goal is to develop a method that produces a confidence interval containing the PAB.

5. Methodology

Our main challenge is to develop a technique to estimate probabilistic available bandwidth that is efficient and scales well with the number of paths. We can divide this problem into the following three tasks: i) probe a path and produce a measurement outcome, ii) compute the marginal posteriors of the path's probabilistic available bandwidth from measurement outcomes and establish confidence intervals for the PAB, and iii) identify measurements (choose the path and probing rate) at each iteration that will minimize the overhead on the network.

A general overview of our approach is presented in Fig. 1. The remainder of this section gives a detailed description of each step shown in Fig. 1.

```
1  create factor graph using known topology;
2  while ∃p s.t. β_p > β do
3  |   choose path to probe next;
4  |   choose rate to probe;
5  |   take new measurement;
6  |   run belief propagation (update marginal posteriors
   |   Pr{y_p|z});
7  |   if maximum number of probes is reached then
8  |   |   break;
9  |   end
10 end
```

Figure 1: Multipath probabilistic available bandwidth estimation algorithm.

5.1. Assumptions

Our method is based on four main assumptions.

1. At the start of each link is a store-and-forward first-come first-served router/switch that dictates the behaviour of the link (in terms of delay, loss, utilization). If the network uses priority queueing or some other form of router-level Quality-of-Service provisioning, then our method will infer the probabilistic available bandwidth as seen by the class of packets transmitted as probes.

[4] The probability is defined over all possible multi-packet flows of average rate equal to the input rate that can complete transmission during the specified measurement period.

[5] We chose to define y_p as a discrete, rather than continuous, random variable because it is not practically meaningful to have an infinite precision on the transmission rates.

2. The routing topology of this network is known, as embodied in the set of paths $\mathcal{P}$, and that it remains fixed for the duration of our experiments. More precisely, we construct a MxN binary path matrix $\mathbf{P}$, where $\mathbf{P}(i, j)$ is equal to one if link j is on path i. To populate the matrix, we infer links and the mapping from IP addresses to routers using `traceroute`[6]. If `traceroute` cannot complete the topology extraction procedure properly, the PAB estimation is done with an incomplete $\mathbf{P}$ matrix.
3. There is a unique path between each of the hosts involved in probing. If there is per-packet load balancing in the network, our `traceroute`-based procedure will identify only one of these paths traversed by packets. This result in missing entries (0's that should be 1's) and/or missing links (rows) in the $\mathbf{P}$ matrix. Our method is unaffected by destination-based load balancing.
4. Like the majority of utilization-based available bandwidth estimation tools, we assume that there is only one tight link on each path that essentially determines the probabilistic available bandwidth of that path[7]. More formally, each path consists of the set of links $L_p = \{\ell_1, \ell_2, \ldots, \ell_n\}$ and one tight link $\ell^* \in L_p$. This allows us to i) perform efficient inference using path-level data and ii) use logical topologies (combine all links that are in a series) rather than routing topologies to reduce the number of links and the complexity of the factor graph. Jain and Dovrolis [6] show that multiple tight links can lead to an underestimation of the available bandwidth. In our case, we interpret the presence of more than one tight link as a modelling inaccuracy that creates noise during the estimation procedure.

In Sect. 6.2, we show how the accuracy and speed of convergence (number of measurements) are affected when the $\mathbf{P}$ matrix is not accurate (topology extraction errors, load-balacing, etc.).

5.2. Probing Strategy

Our probing strategy (line 5 in Fig. 1) is based on the principle of self-induced congestion [22]. A single measurement consists of sending N_t trains of L_s UDP packets of P_{size} bytes at a constant rate r_p and observing the rate r'_p at the receiver side. We then take the median of r'_p obtained from each of the N_t trains and determine the binary outcome z of the measurement using the following relation: $z = \mathbf{1}\{r'_p \geq r_p - \epsilon\}$ where $\mathbf{1}\{x\}$ is the indicator function (equal to one if x is true and zero otherwise)[8].

To achieve a given input rate r_p, we fix the packet size and calculate the time interval, τ, between the departure of consecutive packets according the the following relation: $r_p = P_{size}/\tau$. The receiving rate is calculated similarly by dividing the total number of bytes received by the amount of time that elapsed between the reception of the first and last packet. However, due to task interruption on the sender side, there can be unusual delays between the departure of two consecutive packets ($t_i > t_{i-1} + \tau$ where t_i is the departure time of packet i). We consider these packets invalid and exclude them before calculating the output rate. Upon reception of the last packet of a train, we construct a set V of all the indices $i > 1$ of valid packets and calculate r'_p as follows: $r'_p = (|V| \cdot P_{size})/\left(\sum_{i \in V} t_i - t_{i-1}\right)$.

The probing rate is selected at every iteration, but the other parameters are pre-determined before the beginning of the estimation procedure. The choice of these values is made to minimize the overhead while making sure that results are accurate. Although using multiple trains ($N_t > 1$) and taking the median of the output rates increases the overhead on the network, it is also a way to mitigate the impact of a noisy measurement sequence (e.g. packet train with many invalid packets). A similar logic applies when choosing the size of each probe, P_{size}, and the number of probes in a train, L_s. Larger probes and longer trains provide more samples over which to average r'_p, but also leads to a more significant load on the network and a longer sampling period. However, the choice of packet size must be made carefully to make sure that the packet spacing τ is achievable. According to our observations and the settings in Pathload [6], it is preferable to keep $\tau \geq 80\mu s$. The packet size is then chosen such that this condition is satisfied for every possible probing rate. If a node is unable to send a packet stream with the desired spacing τ, the packets will all be considered invalid on the receiver side. In Sect. 6, we specify and justify our choices for each of these parameters.

5.3. Computing Marginal Posteriors

Bayesian inference is a classical way to update the knowledge about unknown parameters based on new observations. In this framework, the posterior distribution $\Pr(y_p|z_k)$ is proportional to the product of the conditional probability $\Pr(z_k|y_p)$, also called likelihood function, and the prior probability $\Pr(y_p)$: $\Pr(y_p|z_k) \propto \Pr(z_k|y_p)\Pr(y_p)$[9]. To capture the correlations between paths, due to links that are shared by multiple paths, we need to specify a model for $\Pr(y_1, ..., y_M|\mathbf{z})$, the joint posterior distribution of available bandwidth on all paths. The joint probability distribution is complex but it is factorizable and can therefore be captured with a factor graph (line 1 in Fig. 1). A factor graph is a graphical model "that indicates how a joint function of many variables factors into a product of functions

[6] `traceroute`-like methods have been known to inflate the number of observed routers, record incorrect links and bias router degree distributions [46]. These errors result in invalid entries in the matrix $\mathbf{P}$, but do not prevent our algorithm from producing PAB estimates.

[7] We derive this relationship more formally in Sect. 5.3.2.

[8] Despite the loss of information, we choose to produce a binary outcome rather than use the output rate directly for two reasons. First, a binary outcome is more robust and less sensitive to noisy measurements. Second, there is no available likelihood model for the output rate and it is easier to construct empirically an accurate one for the binary outcome.

[9] Given a discrete prior and a discrete likelihood function, the normalization to construct a posterior lying in [0,1] is trivial (just divide by the sum of all values in the discrete vector). The explicit computation of $\Pr(z_k)$ is unnecessary. Moreover, an unnormalized posterior is sufficient to compute confidence intervals and the bisection point, which are the true outputs and operations in our estimation procedure. For example, the bisection point is just the median value, which will not change if all entries are rescaled by a constant.

of smaller sets of variables" [47]. They are composed of two types of nodes (variable and factor nodes) and edges that show dependencies between the variables and the factors. In our case, the variables are discrete random variables of the probabilistic available bandwidth of each link, x_ℓ, and path, y_p. There are three functions that are represented by factor nodes in the graph: i) the prior knowledge about the links, f_x, ii) the relation between the PAB of links and paths, $f_{x,y}$; and iii) the likelihood of an observation on a given path, $f_{y,z}$.

The marginal posteriors are computed (line 6 in Fig. 1) by running belief propagation on the factor graph [48]. The algorithm starts with each one of the leaf nodes (prior and likelihood) sending a message to its adjacent node. Messages are then computed using the sum-product algorithm and continue to propagate until the algorithm stabilizes, i.e. there is minimal or no variation between a newly computed message and the one previously sent of the same edge[10]. Upon completion it is possible to compute the marginal at the variable node (links and paths) by taking the product of all messages incoming on its edges.

Example: In Figure 2, we show an example of a simple logical topology of a network. In this example, there are four nodes interconnected using $N = 3$ different links labeled ℓ_1, ℓ_2, ℓ_3 and we consider $M = 2$ paths (dashed line: p_1, solid line: p_2) where nodes 1 and 2 are the sources and node 4 is the destination. From the logical topology, we can populate the path matrix $\mathbf{P}$ and use it to construct the factor graph.

$$\mathbf{P} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

In Figure 3, we show the factor graph representation of the joint distribution used to compute marginal posteriors of the PAB of each of the three links and two paths. The edges show the variables that the factors depend on. In this case, the prior function is identical for all links. So each variable node x_ℓ is connected to a factor node f_x in the graph. However, we could easily use different functions for each link. Each path and its underlying set of links L_p are connected together to a factor node $f_{x,y}$ (there is an edge for every $\mathbf{P}(i, j) = 1$ in the path matrix). Finally, we see that this specific factor graph includes information from a single observation that was performed on path p_1. For each additional measurement, a new factor node f_{y,z_k} is added to the factor graph.

5.3.1. Prior function

The first function to define is the prior f_x. We use a non-informative prior model for the PAB of a path; a uniform distribution in the range $[B_{min}, B_{max}]$:

$$f_x \sim \mathcal{U}[B_{min}, B_{max}],$$

[10]Belief propagation will converge in cyclic factor graphs under certain conditions, but is not guaranteed to do so [49]. Through our extensive simulations, we did not encounter any convergence issues. To ensure completion, we set the maximum number of messages between two nodes to five during one run of the belief propagation algorithm.

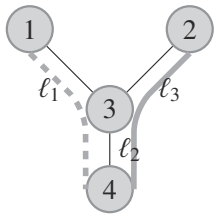

Figure 2: Logical topology of a 4 nodes network with $N = 3$ links (ℓ_1, ℓ_2, ℓ_3) and $M = 2$ paths; p_1 (dashed) and p_2 (solid).

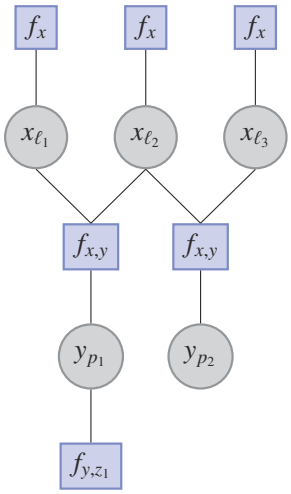

Figure 3: Factor graph representation used to estimate the PAB of the two paths in the topology depicted in Fig. 2.

where $B_{\min}$ and $B_{\max}$ are conservative estimates of the minimum and maximum probabilistic available bandwidths of links. Our choice is due to the lack of any prior information about the PAB of links or paths.

5.3.2. Relation between links and paths

Our inference procedure relies on a relationship between the PAB of a path and the PABs of its constituent links. For the classical utilization-based definition of available bandwidth, it is often assumed that there is a tight link on each path that determines that path's available bandwidth. We develop a similar relationship for the probabilistic available bandwidth.

For a path p consisting of the set of links $L_p = \{1, 2, \ldots, n\}$, it is possible to identify small constants $0 < \epsilon_\ell < \sum_{\ell \in L_p} \epsilon_\ell < \epsilon$ and $0 < \delta_\ell < \sum_{\ell \in L_p} \delta_\ell < 1 - \gamma$ such that:

$$\Pr(r'_\ell \leq r_\ell - \epsilon_\ell) \leq \delta_\ell \quad \text{for all} \quad r_\ell \leq y_p(\epsilon, \gamma). \tag{1}$$

but

$$\Pr(r'_\ell \leq r_\ell - \epsilon_\ell) > \delta_\ell \quad \text{for all} \quad r_\ell > y_p(\epsilon, \gamma). \tag{2}$$

We can apply the union bound on the links to establish:

$$\Pr\left(\bigcup_{\ell \in L_p} \{r'_\ell \leq r_\ell - \epsilon_\ell\}\right) \leq \sum_{\ell \in L_p} \delta_\ell. \tag{3}$$

The complement of this union bound is that the condition $r'_\ell > r_\ell - \epsilon_\ell$ holds for each link. Then we have the following relationship between the path and link input and output rates:

$$\begin{aligned} r_1 &= r_p \\ r_2 &= r'_1 > r_p - \epsilon_1 \\ r_3 &= r'_2 > r_p - \epsilon_1 - \epsilon_2 \\ &\vdots \\ r'_p &= r'_n > r_p - \sum_{i=1}^{n} \epsilon_i. \end{aligned}$$

This relationship and the union bound in (3) imply the following:

$$\Pr\left(r'_p > r_p - \sum_{\ell \in L_p} \epsilon_\ell \right) \geq 1 - \sum_{\ell \in L_p} \delta_\ell. \quad (4)$$

Moreover, we assume that there is a *tight link* $\ell^* \in L_p$ which essentially determines the probabilistic available bandwidth on the path p. This means that it is possible, for all $\ell \in L_p$, $\ell \neq \ell^*$, to identify $\epsilon_\ell \ll \epsilon$ and $\delta_\ell \ll 1 - \gamma$ that satisfy (1). In the case of ℓ^*, however, the smallest $\epsilon_{\ell^*} < \epsilon$ and $\delta_{\ell^*} < 1 - \gamma$ pair that satisfy (1) have the property $\epsilon_{\ell^*} \approx \epsilon$ and $\delta_{\ell^*} \approx 1 - \gamma$. The tight link assumption implies that $\sum_{\ell \in L_p} \epsilon_\ell \approx \epsilon_{\ell^*} \approx \epsilon$ and $\sum_{\ell \in L_p} \delta_\ell \approx \delta_{\ell^*} \approx 1 - \gamma$. This property, together with (1), (2), and (4), imply that $y_p \approx x_{\ell^*}$ where x_ℓ is the PAB of link ℓ. Another way of interpreting this assumption, is that the PAB of any link $\ell \in L_p$, $\ell \neq \ell^*$ is significantly greater than y_p. This relationship is expressed mathematically as

$$f_{x,y}(y_p, \{x_\ell | \ell \in L_p\}) = \mathbf{1}\{y_p = \min_{\ell \in L_p}(x_\ell)\},$$

where $\mathbf{1}\{x\}$ is the indicator function.

5.3.3. Likelihood Model

We specify a likelihood function, $f_{y,z}$, learned from empirical training data, that relates a measurement outcome to the probing rate and the underlying PAB of the probed path. More precisely, it relates the probability that $z = 1$ to the difference between the probing rate and the PAB of the path. In general, we have $f_{y,z} = L(\alpha, y_p, r_p)$, where α is a set of parameters for the likelihood function L. The strategy is then to identify a parametric function L and train the parameters α based on multiple sets of data collected across the network. Although we expect the parameters α to be the same for all measurements, the PAB y_p (which is also unknown) varies from path to path. We decide to co-jointly estimate the values of y_p along with the parameters α through a single regression procedure where we determine the best fit by minimizing the MSE.

Example: We construct a likelihood model for the network we used (PlanetLab) for our experiments using $\epsilon = 5$ Mbps and a range of values where $B_{min} = 1$ Mbps and $B_{max} = 100$ Mbps[11].

[11]The choice of ϵ directly affects the likelihood function. We choose 5 Mbps because it is the smallest value for which the level of noise in the empirical data is acceptable to train a parametric function accurately.

Intuitively, when the probing rate r_p is well below the PAB, we expect the probability of observing $z = 1$ to be very high and, similarly, when r_p is well over the PAB, this probability should be very close to zero. Based on these intuitive expectations and experimental data (Fig. 4), we adopt a sigmoid likelihood model

$$L(z = 1 | y_p, r_p) = \text{logsig}(-\alpha(r_p - y_p))$$

for the measurements, where α is a small positive constant learned empirically[12].

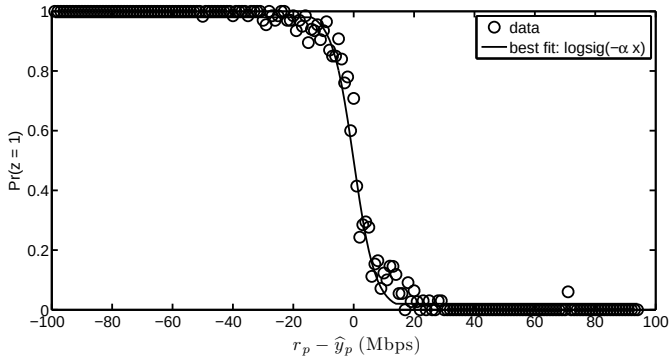

Figure 4: Empirical data and regression fit for the likelihood model. $Pr(z = 1)$ is a function of the difference between the probing rate and estimated available bandwidth. Each data point is obtained by averaging the result of 10 packet trains with $\epsilon = 5$ over five different paths. The best fit is obtained by performing a regression for parameters α and y_p.

We first gather data from five different paths: 500 measurements from non-consecutive packet trains at each rate between B_{min} and B_{max}. We then repeat this experiment five times at different periods of the day resulting in 25 sets of 500 measurements. We normalize each of the 25 experiments and combine all the data in a single plot as a function of $r_p - \widehat{y_p}$. The result is shown in Fig. 4 where each data point is the result of averaging all values which had the same value of $r_p - \widehat{y_p}$; all experiments for which the distance between r_p and $\widehat{y_p}$ is identical. In our case, the regression identifies $\alpha = 0.28$ with a MSE of 0.08 over the range from $[1, 100]$ Mbps. The function depicted is for $\gamma = 0.5$, but it can be easily modified (without any further measurements) for any other value of γ: it consists of aligning the desired value of γ on the curve with the point on the x-axis where $r_p - \widehat{y_p} = 0$.

It is important to note that our estimation procedure is not sensitive to the exact choice of likelihood model (which depends on the probing strategy, the network and the value of ϵ). If the empirical data collected in a particular scenario is a poor fit to the sigmoid model presented here, a different likelihood model can easily be used within our estimation. That said, in the experiments conducted (over multiple weeks on different topologies, days and times-of-day), we have observed that the

[12]The sigmoid function rapidly decays to zero when the probing rate is greater than the available bandwidth, even for the best possible parameter fit. We wish to be careful and prefer a slightly less aggressive approach where we assign some likelihood to unexpected measurement outcomes at all ingress rates. For that reason, we introduce a small constant κ and bound our likelihood function to lie in the range $[\kappa, 1 - \kappa]$; in our experiments $\kappa = 0.02$.

values of α, which specifies the rate of decay of the sigmoid function, occupy a small range. For this reason, the likelihood model only needs to be trained rarely and it can be employed for a long period of time. Furthermore, once the initial likelihood model is trained, it is possible to refine it without any further measurements by using probes from the bandwidth estimation procedure as training data.

5.4. Producing Confidence Intervals

For a given distribution, such as the one depicted in Fig. 5, the confidence interval of size β_p with confidence limits $[\beta_{min}, \beta_{max}]$ is the smallest interval that has a confidence level (fraction of probability mass) greater than or equal to η. The confidence level η is the probability that y_p lies in the confidence interval.

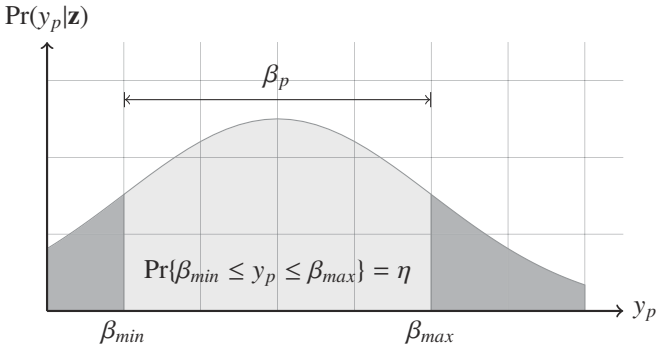

Figure 5: Graphic representation of the probabilistic available bandwidth. The probability that y_p lies in the confidence range $[\beta_{min}, \beta_{max}]$ of size β_p is equal to η (confidence level).

We decide to use β_p as one of our stopping criteria for the estimation procedure (line 2 in Fig. 1); it terminates when the size of the confidence interval of each path is smaller than β ($\forall p : \beta_p \le \beta$). For the cases when the variability of the measurements is too high to meet the desired tightness for confidence intervals, the procedure also stops when the maximum number of iterations is reached (lines 7-9 in Fig. 1).

It is important to note that these parameters (β, η and the maximum number of probes) are not part of the definition of the PAB. They are defined by the user to control how many measurements should be taken to produce the estimates; i.e., how fast, accurate and intrusive the tool is. In Sect. 6.1, we show how the choice of β and γ impact the accuracy and intrusiveness (number of measurements) of the tool.

5.5. Active Sampling

The estimation of available bandwidth based on self-induced congestion is an iterative process. At every iteration, the probing rate is chosen according to some rules. In the case of network-wide estimation, we must also determine which path to probe. The possible sampling rules used to make these selections can be divided in two groups: adaptive (active) or non-adaptive (passive). Non-adaptive sampling means that the sequence of measurements is pre-determined; the probing rate at step k is not affected by previous measurements. These strategies are simple and easy to implement, but can be inefficient. Adaptive (active) selection algorithms, which use information extracted from previous measurements to make decisions about the future, can provide important reductions in the number of probes.

5.5.1. Path Selection

We designed two greedy active learning procedures to select the path to probe at each iteration (line 3 in Fig. 1). Both algorithms are probabilistic in nature: they determine the probability that each path is chosen, and then the choice is accomplished by making a random selection according to the specified probabilities. The first algorithm is called weighted entropy (WE). For each path, we can calculate the entropy of the marginal posterior distribution of its PAB. The entropy is an indication of the uncertainty associated with the current estimate; so WE assigns a probability that a path is selected is proportional to the entropy of the distribution. The second algorithm, called weighted confidence interval (WCI), assigns a selection probability to each path that is proportional to the size of the current confidence interval β_p of the path's PAB; it then chooses a path at random according to the assigned probabilities. In both algorithms, paths are more likely to be probed if there is more uncertainty about their PABs and the probability of probing a path that already satisfies our stopping criteria ($\beta_p \le \beta$) is zero.

5.5.2. Rate Selection

To decide on the probing rate (line 4 in Fig. 1), previous estimation tools either use deterministic binary search or simply increase the probing rate (linearly or exponentially) until it is greater than the available bandwidth. Our Bayesian framework allows us to adopt a more efficient and informative approach. We choose the rate that bisects the marginal posterior distribution of the path. By probing at the median, there is equal probability (according to our current knowledge) that the binary outcome will be $z_k = 1$ or $z_k = 0$. We therefore maximize the expected information gain from our measurement; it is equivalent to conducting a probabilistic binary search for the available bandwidth on path p [15]. By using a probabilistic rather than deterministic approach in rate selection, hard decisions (which could be incorrect) are not enforced.

6. Results and Discussion

6.1. Path Selection Simulations

The purpose of the simulations described in this subsection is to assess the accuracy and speed of convergence of our proposed active sampling strategies. These are not network simulations, so they do not test modelling assumptions at all (that is the purpose of the simulations in Sect. 6.2 and the online experiments in Sect. 6.3).

We use the HOT topology generated using Orbis[13], which includes 939 nodes (896 end nodes) and 988 links. From this

[13] `http://www.sysnet.ucsd.edu/~pmahadevan/topo_research/topo.html`

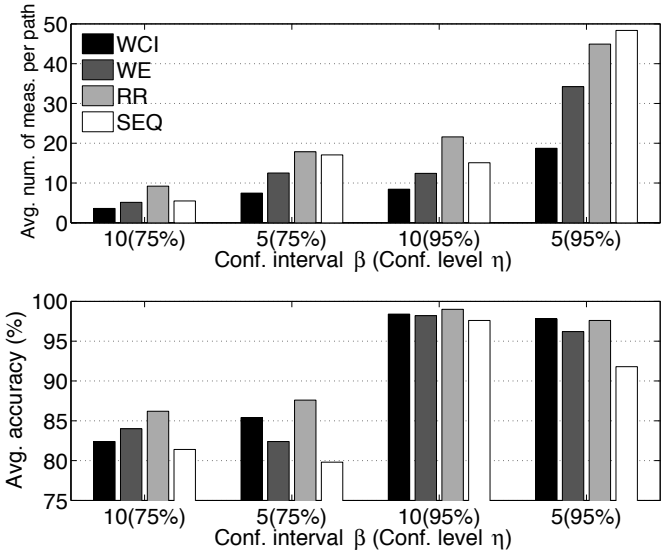

Figure 6: Simulation results: measurements required and accuracy achieved. Results are averaged over 500 topologies of various sizes for different confidence levels η and intervals β.

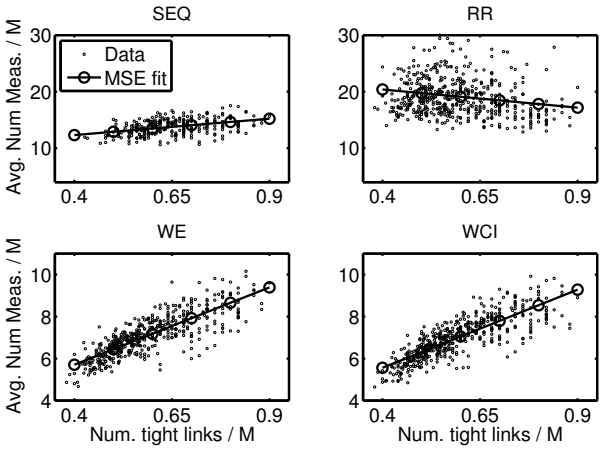

Figure 7: Simulated average number of measurements as a function of the number of tight links in the topology. Both values are normalized by the number of paths M. We show all the simulated values and a first degree polynomial fit for each technique.

set of links and nodes, we construct a distance matrix between all the nodes using shortest path routing and identify 2232 paths (source-destination pairs) that consist of at least seven links. For our simulations, we wish to test our algorithm on topologies of different sizes and vary the number of paths over the range $M = 50, 100, 150, 200, 250$. For each value of M, we randomly select ten different subsets of M paths from the entire set of 2232 paths. For each of these 50 topologies, we assign link PABs using a uniform distribution between $[1, 100]$ and repeat this process ten times to generate a total of 500 topologies.

At each iteration, probe outcomes are generated according to the likelihood model we constructed empirically in Sect 5.3.3 ($\alpha = 0.28$, $\epsilon = 5$). For all simulations, $\gamma = 0.5$, which means that the value of the likelihood function at $y_p = r_p$ is 0.5. We compare three path selection algorithms (Round Robin (RR), WE and WCI) and also show the average number of measurements and accuracy required when our active learning algorithm is run independently and sequentially on each path (SEQ). We use different values of β and η as stopping criteria; the algorithm stops when the size of the confidence interval β_p is smaller than β for all paths p. If these conditions are not met, the algorithm stops after 10000 iterations.

Fig. 6 shows the number of measurements per path required for the algorithm to terminate, as well as the accuracy (an estimate is considered accurate if the real PAB lies within the confidence limits: $\beta_{min} \leq y_p \leq \beta_{max}$). In most cases, SEQ requires fewer measurements than the round-robin strategy with the graphical model. This is due to the fact that not all paths require the same number of measurements. In the RR case, the algorithm iterates through all paths, including those that have already met the required confidence criteria, which is not the case in SEQ. Both data-driven approaches, WCI and WE, significantly reduce the number of measurements required while achieving satisfactory accuracy (i.e., the accuracy exceeds the requested confidence level η).

We investigate the number of iterations for the case where $\eta = 0.95, \beta = 10$ in Fig. 7; we show the average number of measurements per path as a function of the number of tight links per path in the network. Due to the nature of our model, we can identify the PAB of each path if we know the PAB of all the tight links in the network. Therefore, we expect to make greater savings in terms of number of probes when the total number of tight links is small relative to the total number of paths (or, in other words, when the number of paths that share a single tight link is high). The average number of measurements per path required by WCI is between 46 – 73% lower than the number required by RR and 39 – 55% lower than SEQ. WE and WCI provide important savings in terms of time and measurements without affecting the accuracy, but since WCI is slightly better in terms of average number of measurements, we use WCI for our online experiments. As expected, when tight links are located on non-shared links, more measurements are required to achieve the same level of accuracy.

6.2. Topology Accuracy Simulations

One of the main conditions for our methodology to work is that the logical topology is known and that we can construct a path matrix **P**. However, there are many factors that can affect the accuracy of this matrix (e.g., incorrect or incomplete extraction using `traceroute`, load-balancing, changes in the topology during the estimation procedure, etc.) that all have a similar impact: missing/superfluous links (rows in **P**) and missing/superfluous entries (flipped bits in **P**). In this section, we explore how noise (errors) in the path matrix affects the accuracy and speed of convergence of our methodology.

Let TE be the probability that path p is incorrectly extracted using `traceroute`. For each erroneously extracted path, there is a probability q_{flip} that each link in the set $\mathcal{L}$ is mistakenly

identified as either present or missing from path p[14]. More concretely, for each row of **P**, there is a probability TE that every column entry is flipped with probability q_{flip}. The result is a noisy factor graph (path matrix) that propagates inaccurate information because of invalid edges between path and link variable nodes.

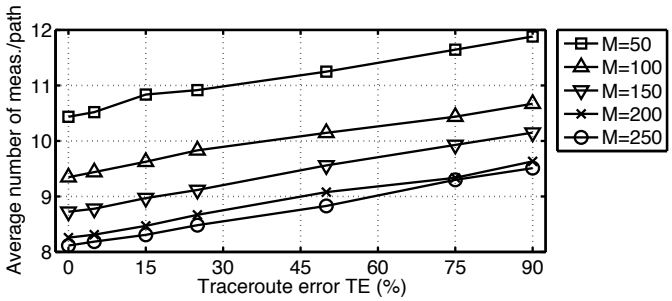

Figure 8: Average number of measurements per path as a function of the traceroute error for topologies with different number of paths M.

For each of the 500 topologies we used in Sect. 6.1, we generate seven topologies by varying TE over the range $0\%, 5\%, 15\%, 25\%, 50\%, 75\%, 90\%$. For the simulations, we use WCI for path selection, the same likelihood model with $\alpha = 0.28$ and set $\gamma = 0.5$, $\epsilon = 5$ Mbps, $\eta = 0.95$, $\beta = 10$ Mbps, $B_{min} = 1$ Mbps, $B_{max} = 100$ Mbps. In Fig. 8, we show the average number of measurements per path as a function of TE. As expected, the number of iterations required to achieve the requested confidence level and tightness increases for topologies with a greater probability of `traceroute` error. However, this augmentation is not significant; even with $TE = 90\%$, the estimation requires only 1.5 more measurements per path on average.

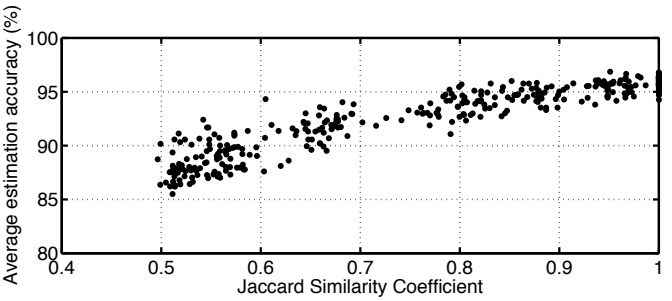

Figure 9: Average estimation accuracy (Jaccard Similarity Coefficient = $|A \cap B|/|A \cup B|$) as a function of the topology accuracy for all topologies.

To quantify the similarity between topologies and provide a more meaningful metric than TE, we use the Jaccard similarity coefficient. It is equal to the size of the intersection (number of correctly identified links) divided by the size of the union (all links from both topologies) [50]. We display the average accuracy of our estimates over all topologies in Fig. 9. Our simulation results show that, for topologies of any size, as long as the `traceroute` methodology produces a matrix **P** with a similarity coefficient greater than 0.5, 85% of the paths are estimated accurately on average. Therefore, even when it uses an inaccurate path matrix, our methodology can generate reasonably precise estimates without any significant inflation in the number of probes required.

Another part of our assumption is that **P** does not change during the estimation procedure, which does not necessarily imply that the physical topology remains unchanged. This is the case because we work with a logical topology and the mapping to a specific node or link is irrelevant. We have not performed any simulations to validate this assumption, but we have studied it empirically. Before each of our online experiments, we generated the matrix **P** and observed that it was almost always identical to previous matrices extracted from the same set of nodes (Song and Yalagandula [39] made similar observations about the PlanetLab network). This evidence is sufficient to conclude that the path matrix does not suffer significant modifications during the time interval over which the estimation procedure is performed.

6.3. Online experiments

6.3.1. Experimental Setup

For our online experiments, we have deployed our measurement software coded in C on various nodes on the PlanetLab network[15]. We use a topology with six nodes[16], $M = 30$ paths and $N = 65$ logical links. For all our experiments, the likelihood model is the one presented in Sect. 5.3.3 ($\alpha = 0.28$, $\epsilon = 5$) and WCI is used to select the path to probe at each iteration. We choose $B_{min} = 1$ Mbps and $B_{max} = 100$ Mbps as conservative estimates of the PAB of each link (we assume that the links with the highest capacity are 100 Mbps links). To make sure that $\tau \geq 80\mu s$ even when probing at B_{max}, we choose $P_{size} = 1000$ bytes. Each measurement consists of $N_t = 3$ trains of packets.

6.3.2. Testing Procedure

Since we are not testing in a controlled environment, we do not have access to the true value of the PAB (or even of the available bandwidth). Although some tools are known to provide accurate estimates of the available bandwidth, none of them have established themselves as a true reference or benchmark. Therefore, to validate our results, we propose a method that consists of sending trains of 2400 packets of 1000 bytes (the equivalent of 60 seconds of video encoded at 320 kbps), observing the output rate and calculating $z = \mathbf{1}\{r'_p \geq r_p - \epsilon\}$. This test assesses whether or not it is possible to achieve a given probing rate, which is the information we are interested in.

Because it would be impractical to probe every single rate in $[B_{min}, B_{max}]$, we choose four different rates that correspond

[14] This probability is chosen such that the average path length remains constant. Based on the topologies we used for our simulations, this probability depends on the number of links in the network and varies between 1-3%.

[15] Although the PlanetLab (`http://www.planet-lab.org/`) network was once believed to be too heavily loaded, Spring et al. [51] explained that PlanetLab has evolved and this is no longer true.

[16] planetlab3.csail.mit.edu, planetlab-1.cs.unibas.ch, planlab1.cs.caltech.edu, planetlab2.acis.ufl.edu, planetlab1.cs.stevens-tech.edu, planetlab2.csg.uzh.ch.

to potential estimates of the PAB. These rates are chosen from the confidence intervals produced with our estimation procedure: the lower bound of the confidence interval β_{min}, the middle of the confidence interval $\beta_{mean} = (\beta_{min} + \beta_{max})/2$, the upper bound β_{max}, and 5 Mbps above the upper bound. For each of these rates, we choose four disjointed paths and compute the empirical probability $\widehat{\Pr}(z = 1|r_p)$. This testing procedure is performed when the estimation terminates (stopping criteria are met; $\beta_p \leq 10$Mbps and $\eta_p \geq 0.95$).

6.3.3. Train size

In this first experiment, we set $\gamma = 0.5$ and compute the empirical probability of success $\widehat{\Pr}(z = 1|r_p)$, for the different values of r_p probed in our testing procedure, averaged over 20 runs. We vary the number of packets in each train in the range $L_s = [25, 50, 100, 150, 200, 250]$. The results are shown in Fig. 10. The first observation is that the number of packets used in trains induces very little variation in empirical probability for all the probing rates. This suggests that, for this network at least, 25 packets per train would suffice.

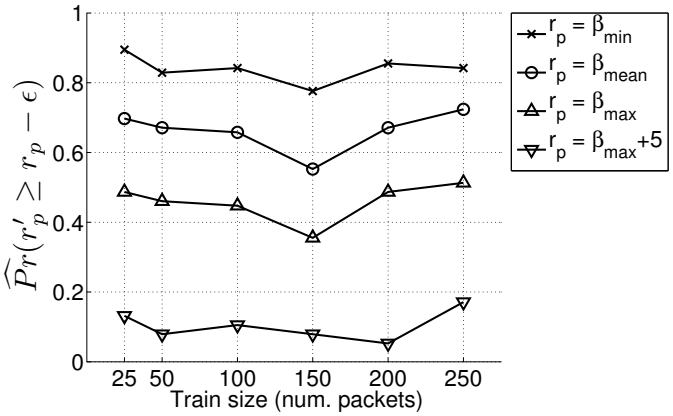

Figure 10: Empirical probability that the output rate is within ϵ Mbps of the input rate. Each point represents the average of 80 test results (20 runs).

For all the train sizes we tested, the desired probability $\gamma = 0.5$ is included in the probability interval of β_{min} and β_{max}. This result confirms that our method is able to produce intervals that include the value of the PAB accurately. The fact that $\gamma = 0.5$ is very close to the upper bound suggests that we might underestimate the PAB (we discuss possible reasons for this below). However, when the probing rate is 5Mbps over the upper bound of the interval, the empirical probability of success drops well below 0.2, which indicates that we are not drastically underestimating the PAB.

We investigate the impact of the train size by using the raw data collected at each node during the 20 runs (18000 measurements for each value of L_s). In Fig. 11, we show the average empirical probability of observing $z = 1$ as a function of the difference between the probing rate and our estimate of the PAB (we use the marginal maximum a posteriori (MAP) estimate). We anticipate this plot to be comparable to the likelihood model depicted in Fig. 4. Since we set $\gamma = 0.5$, we expect the probability of observing $z = 1$ to be near 0.5 when the probing rate

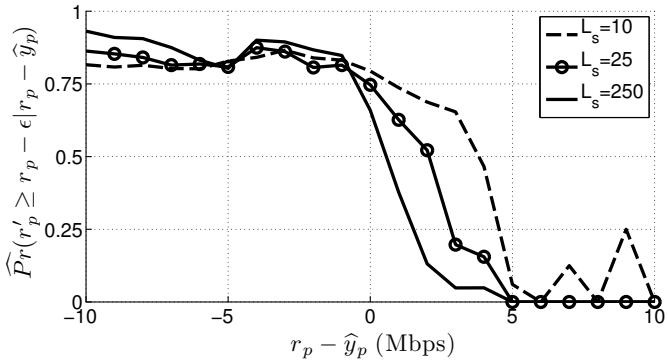

Figure 11: Empirical probability of observing $z = 1$ averaged over 17966 measurements as a function of the difference between the probing rate and the estimated PAB (MAP of the marginal posterior).

is equal to the PAB ($r_p - \hat{y}_p = 0$). However, what we observe is that the probability is closer to 0.75 at that point, which is approximately the average empirical probability at $\beta_{min} + \epsilon$ in Fig. 10. This confirms a slight underestimation of the PAB, which is probably due to an inaccurate likelihood model. The figure also shows that as the train size is reduced, the measurements become noisier and the bias (underestimation) becomes more significant. In future work, we will explore other likelihood models (possibly a combination of two sigmoids) to match the asymmetry observed here.

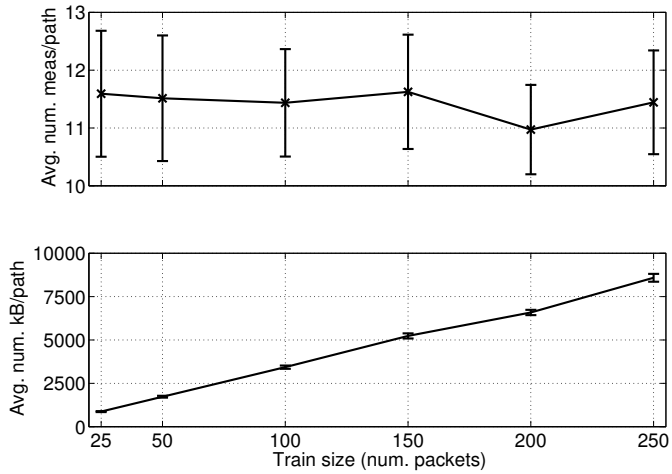

Figure 12: Number of measurements (TOP) and bytes (BOTTOM) used per path (averaged over 20 runs for each train sizes L_s) during the estimation procedure.

Figs. 10 and 11 indicate that the accuracy obtained when using $L_s = 25$ and $L_s = 250$ packets is similar. In Fig. 12, we show the average number of measurements and bytes per path required to complete the estimation procedure as a function of the train size. Since the number of measurements is constant for all values of L_s, we observe a linear growth in the number of bytes required to achieve the desired accuracy. From these results, it is now clear that using 25 packets per train is optimal as it provides similar accuracy to larger train sizes with significant savings in terms of number of probes.

6.3.4. Probability of success γ

We investigate the accuracy and the intrusiveness of our tool when γ is increased from 0.5 to 0.9. We perform the same testing procedure as before and show the results, averaged over 20 runs, in Tab. 1 (with standard deviations) and Tab. 2. We observe that increasing γ leads to a greater empirical probability of success at the expense of overhead and time. However, the overhead is still significantly smaller than using Pathload sequentially. It is encouraging to see that by modifying a single parameter it is possible to obtain estimates of the PAB with different probability of success, which was our initial objective. We are currently working on a different probing strategy based on chirps that will provide the same flexibility and accuracy with much more acceptable and practical overhead costs.

Table 1: Average (over 20 runs) overhead and latency shown with standard deviation as a function of γ compared with Pathload (PL).

Averages per path	$\gamma = 0.5$	$\gamma = 0.9$	PL[6]
Latency (sec.)	9.3 ± 0.9	25 ± 2	34 ± 2
Overhead (kB)	869 ± 82	2483 ± 38	4664 ± 410

Table 2: Empirical probability of avoiding congestion (averaged over 20 runs) as a function of γ compared with Pathload (PL).

$\widehat{\Pr}(z = 1\|r_p)$	$\gamma = 0.5$	$\gamma = 0.9$	PL[6]
$r_p = \beta_{min}$	0.89	0.97	0.65
$r_p = \beta_{mean}$	0.70	0.86	0.63
$r_p = \beta_{max} + 5$	0.13	0.44	0.45

6.3.5. Comparison with other tools

It is interesting to compare our estimation methodology to another tool based on the classical definition of available bandwidth to examine the extent of correlation between the two metrics. We choose to compare our results with those obtained using Pathload (version 1.3.2) [6] because it is one of the most accurate techniques [4, 27, 29]. We run Pathload sequentially on every single path (of the topology described in Sect. 6.3.1). We then run the testing procedure outlined in Sect. 6.3.2 and assess accuracy, overhead and latency of both techniques.

The first observation is that Pathload fails to provide any estimate far more often than our approach. In the 20 runs, Pathload only converged to valid bounds for 62% of the paths whereas our approach always converges to a confidence interval of size β for every path[17]. To compare overhead and accuracy, we only consider the paths for which Pathload's variation range has valid bounds.

The two techniques estimate different metrics (PAB versus utilization-based available bandwidth) but in practice the Pathload metric is often used to answer the same question, i.e. what is the maximum rate at which a flow can be sent along a path without inducing congestion? Based on this, we can compare the accuracies of the tools by sending traffic flows at and just above the identified estimate and assessing how often congestion is avoided at the two rates. It is important to mention that the size of the variation range varies whereas the confidence intervals produced by our approach is at most $\beta = 10$Mbps. Although the variance is greater, the median size for Pathload's variation ranges (0.46Mbps) is smaller. For that reason, the difference between the empirical PAB (shown in Tab. 2) at the lower bound and the mean of the variation range is very small.

The results show that we meet our probabilistic guarantees and that Pathload's estimates avoid congestion 60% of the time. To show that our estimates are not too conservative and close to the maximum rate at which we can send while avoiding congestion (with probability γ), we also test at rates that exceed the identified rates. By probing at rate 5Mbps over the upper bound of the confidence interval (variation range in the case of Pathload), we see that the probability of success decreases significantly and drops below γ. This suggests that the maximum rate lies without our confidence interval (for any value of γ) whereas Pathload's would overestimate the available bandwidth in cases where $\gamma > 0.65$. As opposed to our approach, there is no parameter that explicitly affects the probability avoiding congestion to make Pathload suitable for these scenarios.

Furthermore, as we can see from Tab. 1, Pathload is significantly more intrusive and time consuming than our methodology. In the case of $\gamma = 0.5$, the probability of success of our approach is greater than Pathload's and still provides significant gains in terms of measurement latency (73% savings) and overhead (81% savings).

Comparing the overhead of our technique with Pathload's confirms that previous tools are not well suited to multi-path estimation. The only other approaches that can produce efficient network-wide AB estimates are BRoute [41] and bandwidth landmarking [43]. In both cases, little detail is provided regarding the actual overhead incurred by their techniques and their software is not publicly available. Hu and Steenkiste [41] claim that 80% of the available bandwidth estimates obtained from BRoute are accurate within 50% when using a subset that includes only 7% of all paths. However, there is no mention of how many measurements are required for each path.

6.3.6. PlanetLab Observations

In Fig. 13, we display the confidence intervals as well as the test results (probe rate and output rate) for one of the runs performed with $L_s = 25$ and $\gamma = 0.5$. The outcome of this particular run demonstrate the clear heterogeneity of the PlanetLab network; over 25% of the paths have small (less than 20 Mbps) PAB whereas the other 75% have PAB greater than 80 Mbps. The tight links on the paths with lower PAB could either be heavily utilized 100 Mbps links or, more likely, 10 Mbps links with small amounts of cross-traffic. These findings about the PlanetLab network correspond to those of Lee et al. [52].

7. Conclusion

In this paper, we presented a novel technique based on a probabilistic framework to estimate network-wide probabilistic

[17] The Pathload algorithm fails to converge when there are frequent context switches at the sender or receiver or when the packet loss rate is too high.

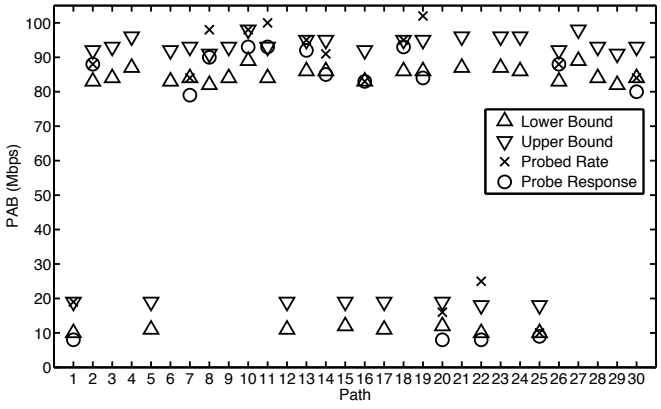

Figure 13: Bounds of the confidence intervals for a 30 paths topology in a sample run performed for $L_s = 25$ and $\gamma = 0.5$.

available bandwidth. We introduced PAB, a new metric with adjustable parameters that addresses issues related to the dynamics and variability of available bandwidth. Our methodology based on factor graphs and active sampling is the first to combine both techniques in the context of available bandwidth estimation. To further reduce the overhead of our technique, we are currently working on a new measurement strategy and likelihood model based on chirps rather than trains of packets, which, from our preliminary results, can achieve significant savings in terms of probing overhead. Furthermore, we plan to extend our block-based estimation framework to track the PAB in time.

References

[1] Y. Hiraoka, G. Hasegawa, M. Murata, Effectiveness of overlay routing based on delay and bandwidth information, in: Proc. Australasian Telecommunication Networks and Applications Conf., Christchurch, New Zealand, 2007.

[2] S.-J. Lee, S. Banerjee, P. Sharma, P. Yalagandula, S. Basu, Bandwidth-aware routing in overlay networks, in: Proc. IEEE Int. Conf. Computer Communications, Phoenix, AZ, 2008.

[3] L. He, S. Yu, M. Li, Anomaly Detection based on Available Bandwidth Estimation, in: Proc. IFIP Int. Conf. Network and Parallel Computing, Shanghai, China, 2008.

[4] C. D. Guerrero, M. A. Labrador, On the applicability of available bandwidth estimation techniques and tools, Computer Communications 33 (1) (2009) 11–22.

[5] X. Liu, K. Ravindran, D. Loguinov, A Stochastic Foundation of Available Bandwidth Estimation: Multi-Hop Analysis, IEEE/ACM Trans. Networking 16 (1) (2008) 130–143.

[6] M. Jain, C. Dovrolis, End-to-end available bandwidth: measurement methodology, dynamics, and relation with TCP throughput, IEEE/ACM Trans. Networking 11 (4) (2003) 537–549.

[7] C. D. Guerrero, M. A. Labrador, Traceband: A Fast, Low Overhead and Accurate Tool for Available Bandwidth Estimation and Monitoring, Computer Networks 54 (6) (2010) 977–990.

[8] D. Croce, M. Mellia, E. Leonardi, The Quest for Bandwidth Estimation Techniques The Quest for Bandwidth Estimation Techniques for Large-Scale Distributed Systems, in: Proc. ACM Work. Hot Topics Measurement and Modelling of Computer Systems, Seattle, WA, 2009.

[9] M. J. Coates, R. Nowak, Networks for networks: Internet analysis using graphical statistical models, in: Proc. IEEE Work. Neural Networks for Signal Processing, Sydney, Australia, 2000.

[10] Y. Mao, F. Kschischang, B. Li, S. Pasupathy, A factor graph approach to link loss monitoring in wireless sensor networks, IEEE J. Selected Areas Communications 23 (4) (2005) 820–829.

[11] I. Rish, Distributed systems diagnosis using belief propagation, in: Proc. Allerton Conf. Communication, Control and Computing, Monticello, IL, 2005.

[12] A. Zheng, I. Rish, A. Beygelzimer, Efficient Test Selection in Active Diagnosis via Entropy Approximation, in: Proc. Conf. Uncertainty in Artificial Intelligence, Edinburgh, Scotland, 2005.

[13] I. Rish, Information-theoretic approaches to cost-efficient diagnosis, in: Proc. Information Theory and Applications Inaugural Work., San Diego, CA, 2006.

[14] M. A. El-Gamal, R. D. McKelvey, T. R. Palfrey, A Bayesian Sequential Experimental Study of Learning in Games, J. of the American Statistical Association 88 (422) (1993) 428–435.

[15] R. Castro, R. Nowak, Active Learning and Sampling, in: A. Hero, D. Castanon, D. Cochran, K. Kastella (Eds.), Foundations and Applications of Sensor Management, Springer-Verlag, 177–200, 2007.

[16] H. H. Song, L. Qiu, Y. Zhang, NetQuest: A Flexible Framework for Large-Scale Network Management, IEEE/ACM Trans. Networking 17 (1) (2009) 106–119.

[17] V. Ribeiro, M. Coates, R. Riedi, S. Sarvotham, B. Hendricks, R. Baraniuk, Multifractal cross-traffic estimation, in: Proc. ITC Specialist Seminar on IP Trac Measurement, Modeling, and Management, Monterey, CA, 2000.

[18] N. Hu, P. Steenkiste, Evaluation and Characterization of Available Bandwidth Probing Techniques, IEEE J. Selected Areas Communications 21 (6) (2003) 879–894.

[19] J. Strauss, D. Katabi, F. Kaashoek, A Measurement Study of Available Bandwidth Estimation Tools, in: Proc. ACM SIGCOMM Internet Measurement Conf., Miami Beach, FL, 2003.

[20] J. Navratil, R. Les Cottrell, ABwE: A Practical Approach to Available Bandwidth Estimation, in: Proc. Passive and Active Measurement Conf., La Jolla, CA, 2003.

[21] L. Lao, C. Dovrolis, M. Sanadidi, The probe gap model can underestimate the available bandwidth of multihop paths, ACM SIGCOMM Computer Communication Review 36 (5) (2006) 29–34.

[22] V. Ribeiro, R. Riedi, R. Baraniuk, J. Navratil, L. Cottrell, pathChirp: Efficient Available Bandwidth Estimation for Network Paths, in: Proc. Passive and Active Measurement Conf., La Jolla, CA, 2003.

[23] B. Melander, M. Bjorkman, P. Gunningberg, A new end-to-end probing and analysis method for estimating bandwidth bottlenecks, in: Proc. IEEE Global Telecommunications Conf., San Francisco, CA, 2000.

[24] B. Melander, M. Bjorkman, P. Gunningberg, Regression-based available bandwidth measurements, in: Proc. Int. Symp. Performance Evaluation of Computer and Telecommunications Systems, San Diego, CA, 2002.

[25] J. Sommers, P. Barford, W. Willinger, Laboratory-based calibration of available bandwidth estimation tools, Microprocessors and Microsystems 31 (4) (2007) 222–235.

[26] E. Goldoni, G. Rossi, A. Torelli, Assolo, a New Method for Available Bandwidth Estimation, in: Proc. IEEE Int. Conf. Internet Monitoring and Protection, Venice, Italy, 2009.

[27] A. Shriram, M. Murray, Y. Hyun, N. Brownlee, A. Broido, M. Fomenkov, K. C. Claffy, Comparison of Public End-to-End Bandwidth Estimation Tools on High-Speed Links, in: Proc. Passive and Active Measurement Conf., Boston, MA, 2005.

[28] A. Shriram, J. Kaur, Empirical Evaluation of Techniques for Measuring Available Bandwidth, in: Proc. IEEE Int. Conf. Computer Communications, Anchorage, AK, 2007.

[29] E. Goldoni, M. Schivi, End-to-End Available Bandwidth Estimation Tools, An Experimental Comparison, in: Proc. Traffic Monitoring and Analysis Work., Zurich, Switzerland, 2010.

[30] M. Neginhal, K. Harfoush, H. Perros, Measuring Bandwidth Signatures of Network Paths, in: Proc. IFIP Networking, Atlanta, GA, 2007.

[31] P. Hága, P. Mátray, I. Csabai, G. Vattay, Modelling packet pair dispersion in multi hop networks with correlated traffic, in: Proc. European Conf. Complex Systems, Warwick, UK, 2009.

[32] J. Liebeherr, M. Fidler, S. Valaee, A Min-Plus System Interpretation of Bandwidth Estimation, in: Proc. IEEE Int. Conf. Computer Communications, Anchorage, AK, 1127–1135, 2007.

[33] J. Liebeherr, M. Fidler, S. Valaee, A System Theoretic Approach to Bandwidth Estimation, Tech. Rep., University of Toronto, 2008.

[34] R. Castro, M. Coates, G. Liang, R. Nowak, B. Yu, Network tomography:

Recent developments, Statistical Science 19 (3) (2004) 499–517.
[35] Y. Vardi, Network Tomography: Estimating Source-Destination Traffic Intensities from Link Data., J. of the American Statistical Association 91 (433).
[36] D. Chua, E. Kolaczyk, M. Crovella, Network Kriging, IEEE J. Selected Areas Communications 24 (12) (2006) 2263–2272.
[37] Y. Chen, D. Bindel, R. Katz, Tomography-based overlay network monitoring, in: Proc. ACM SIGCOMM Internet Measurement Conf., Miami, FL, 2003.
[38] Y. Chen, D. Bindel, H. H. Song, R. H. Katz, Algebra-based scalable overlay network monitoring: algorithms, evaluation, and applications, IEEE/ACM Trans. Networking 15 (5) (2007) 1084–1097.
[39] H. H. Song, P. Yalagandula, Real-time End-to-end Network Monitoring in Large Distributed Systems, in: Proc. Int. Conf. Communication Systems Software and Middleware, Bangalore, India, 2007.
[40] M. J. Coates, Y. Pointurier, M. Rabbat, Compressed network monitoring, in: Proc. IEEE Work. Statistical Signal Processing, Madison, WI., 2007.
[41] N. Hu, P. Steenkiste, Exploiting internet route sharing for large scale available bandwidth estimation, in: Proc. ACM SIGCOMM Internet Measurement Conf., Berkeley, CA, 2005.
[42] N. Hu, L. E. Li, Z. M. Mao, P. Steenkiste, J. Wang, Locating Internet Bottlenecks: Algorithms, Measurements and Implications, in: Proc. ACM SIGCOMM, Portland, OR, 2004.
[43] B. Maniymaran, M. Maheswaran, Bandwidth landmarking: A scalable bandwidth prediction mechanism for distributed systems, in: Proc. IEEE Global Telecommunications Conf., Washington, DC, 2007.
[44] P. Yalagandula, S.-J. Lee, P. Sharma, S. Banerjee, Correlations in End-to-End Network Metrics: Impact on Large Scale Network Monitoring, in: Proc. IEEE Int. Conf. Computer Communications Work., Phoenix, AZ, 2008.
[45] C. Man, G. Hasegawa, M. Murata, Inferring available bandwidth of overlay network paths based on inline network measurement, in: Proc. Int. Conf. Internet Monitoring and Protection, Silicon Valley, CA, 2007.
[46] R. Sherwood, A. Bender, N. Spring, DisCarte: A Disjunctive Internet Cartographer, in: Proc. SIGCOMM, Seattle, WA, 2008.
[47] B. Frey, Graphical models for machine learning and digital communication, MIT Press, 1998.
[48] J. Pearl, Probabilistic reasoning in intelligent systems: networks of plausible inference, Morgan Kaufmann, San Francisco, CA, 1988.
[49] J. Mooij, H. Kappen, Sufficient Conditions for Convergence of the Sum–Product Algorithm, IEEE Trans. Information Theory 53 (12) (2007) 4422–4437.
[50] P. Jaccard, Étude comparative de la distribution florale dans une portion des Alpes et des Jura, Bulletin de la Société Vaudoise des Sciences Naturelles 37 (1901) 547–579.
[51] N. Spring, L. Peterson, A. Bavier, V. Pai, Using PlanetLab for network research: myths, realities, and best practices, ACM SIGOPS Operating System Review 40 (1) (2006) 17–24.
[52] S.-J. Lee, P. Sharma, S. Banerjee, S. Basu, R. Fonseca, Measuring Bandwidth Between PlanetLab Nodes, in: Proc. Passive and Active Measurement Conf., Boston, MA, 2005.