Sampling Biases in Network Path Measurements and What To Do About It

Srikanth Kandula Ratul Mahajan

Microsoft Research

ABSTRACT

We show that currently prevalent practices for network path measurements can produce inaccurate inferences because of sampling biases. The inferred mean path latency can be more than a factor of two off the true mean. We present the Broom toolkit that has three methods to correct for this bias. Broom places no burden on the measurement process itself and can be applied post hoc to any measured data set. Our evaluation finds that two of the methods are particularly effective. One of them estimates missing path samples by embedding the nodes in a low-dimensional coordinate space. For realistic sampling rates, the quality of its estimates for path latency approximates ideal, unbiased sampling. The other method is based on a view of network paths as being composed of source-specific, destination-specific, and shared components. It reduces bias for a wide range of path properties, such as latency, hop count and capacity. Applying Broom to data from a real measurement study leads to substantial changes in the resulting inferences. For some networks, the post-correction estimate is 30% higher than the original.

Categories and Subject Descriptors

C.4 [**Performance of Systems**]: Measurement techniques

General Terms

Measurement, Algorithms

Keywords

Network measurement, Sampling bias, Coordinate embedding, Path decomposition

1. INTRODUCTION

Measurement is a key component of research, development and practice in many domains. It is used to validate hypotheses, understand complex dynamics, and inform policy. Examples include studies of new medicines, habits of drug addicts, electoral voting tendencies, and tolerable stress level for industrial components.

Computer networks are no different in this regard. Measurements are used extensively to understand network behavior and optimize performance. The community devotes much effort towards developing and refining measurement methods.

IMC'09, November 4–6, 2009, Chicago, Illinois, USA.

However, unlike other domains, relatively little attention has been paid to how sampling biases in network measurements can taint results. A notable exception is the study of bias in network topology measurement [5, 14], where researchers show how sampling biases can lead to inferences that differ markedly from the true topology.

In this paper, we focus on sampling biases in network path measurements, which are frequently used to measure performance and optimize protocols.

Consider a typical measurement experiment, for instance, to characterize network path latency between a set of sources and destinations. Experimenters get as many vantage points as they can (e.g., PlanetLab nodes) and measure paths to destinations of interest. The vantage point is not necessarily a source of interest but the source-to-destination component can be isolated by "subtracting" the vantage-point to source component.

There are many challenges in drawing accurate inferences from such an experiment. One challenge is to accurately infer properties of measured paths, by accounting for the possible sources of error in the raw measurements. Example sources of error include routing changes during measurements and delayed response by destinations. Researchers have developed and continue to evolve techniques to minimize the impact of such errors. These techniques are not the focus of our work.

Assuming accurate path measurements have been obtained, a second challenge is to infer network-wide properties, e.g., mean latency across all paths. The difficulty is that measuring all paths of interest is rarely feasible because of limitations in where vantage points are available, and a significant fraction of paths (50-95%) can go unmeasured. The current practice tends to assume that the set of measured paths are representative and thus accurate inferences can be drawn based on those paths.

We show that this is a flawed assumption because the set of measurements is biased by which sources contribute. Using simulations over realistic network topologies, we find that the resulting inferences can be significantly inaccurate. For instance, when 5% of the paths are measured, the inferred mean can be more than a factor of two off the true mean. The error is not purely because of measuring too few paths. The median error is lower by a factor of five with an ideal (but impractical) sampling method that measures the same number of paths. Even worse, because the measured data is biased, standard statistical methods for estimating the uncertainty in the inferences are rendered ineffective. The 99% confidence intervals around the mean contain the true mean less than half the time. Confidence intervals for the ideal sampling method almost always contain the true mean.

How can we remedy this bias? At one level, it might appear that not much can be done. Based on the measured paths alone, one cannot tell if the unmeasured paths are statistically different. But we are encouraged by the fact that despite similar limitations researchers in other domains have developed credible methods over the years to correct for measurement bias [4, 11, 15, 21]. These methods are by necessity specific to their target domain, but their general approaches apply.

Based on these approaches, we develop and evaluate Broom, a toolkit of three methods that correct for bias in network path measurements. The methods in Broom do not burden the measurement process; rather they can post-process any collected data. We find two of the methods to be particularly effective.

One of the methods leverages the observation that network distances can be well-approximated by embedding the nodes in a low-dimensional coordinate space [8, 16, 20]. This observation has been used in the past to avoid the need for all-to-all measurements. We co-opt it for correcting bias by using measured data to embed nodes in a coordinate space. This embedding lets us estimate the properties of paths that were not directly measured and thus obtain an unbiased view of the network.

Another method in Broom starts with the observation that network paths can be considered as being composed of a source-specific component, a destination-specific component, and a component shared by all paths (i.e., the network core). It infers the properties of these components by formulating an inference problem that uses measured data as constraints that must be satisfied. Like coordinate embedding, it can then estimate the properties of unmeasured paths.

We also capture the uncertainty inherent in measured data due to possible under-sampling. When too few paths are measured multiple coordinate embeddings and path decompositions are consistent with the measured data. We use this observation to recover multiple possible network configurations. The variance observed across these configurations lets us estimate confidence intervals in a way that they contain the true value with a high probability.

We show that coordinate embedding is highly effective at inferring network latency. It reduces the median error by a factor of five and approximates well the ideal sampling method for realistic sampling rates. Path decomposition is slightly inferior than coordinate embedding for inferring latency but is more broadly applicable. It can be used to infer mean path capacity and hop count as well, for which it comes close to ideal sampling.

We also show how correcting for bias changes inferences for real measurement studies. We apply Broom to two data sets collected by Netdiff, a system to compare backbone ISPs [18]. For one data set, correcting for bias increases the inferred mean latency for some ISPs by as much as 30%. For the other data set, the post-correction confidence intervals are much wider, indicating that some of the conclusions that the original study drew are likely problematic.

We do not claim to be the first to worry about sampling biases in network path measurements. Indeed, most researchers are aware of this possibility. They either confirm the absence of bias (if possible) or carefully list the assumptions regarding its absence that must hold for the conclusions to be valid. The contribution of this paper is a systematic study of the impact of sampling biases in path measurements and the development of the first set of methods that can be used to correct for bias. Future research will hopefully build upon these methods.

2. SAMPLING BIASES IN NETWORK PATH MEASUREMENTS

In this section, we evaluate the quality of inferences obtained using current measurement practices. Consider the following common scenario. Given a network with a set S of sources and a set D of destinations, we are interested in accurately estimating some aggregate property of the set of paths from sources to destinations. These two sets need not be disjoint, and may even be identical. While we will evaluate other possibilities later in the section, we begin by focusing on latency as the path property of interest and on mean as the aggregate measure.

In most network measurement scenarios, not all paths can be directly measured and thus the estimates must be obtained by sampling a subset. Which paths can be sampled is limited by many practical constraints such as where measurement vantage points are available. We assume that for paths that are sampled, accurate path-level properties have been obtained by applying appropriate techniques to account for any error in the raw measurements.

We study three different methods for sampling paths. One is an ideal but impractical method that can measure any path. The other two methods are limited to a certain set of sources that contribute measurements. These contributing sources can then measure the path to any destination. The three sampling methods are:

- **Uniform path sampling** is an ideal method that samples each network path with the same probability and is thus capable of estimating aggregate path properties without any bias.
- **Uniform source sampling** picks contributing sources with equal probability. This sampling method can yield biased results when the aggregate properties of paths from selected sources differ from other sources.

  In practice, uniform source sampling is rarely feasible and experimenters must work with whatever sources are available. For instance, a measurement system might be limited by where PlanetLab nodes are available. The range of errors in estimates with uniform source sampling for a given fraction of sources represents how bad the error can be in any given situation that uses only that fraction of the sources.
- **Degree-biased source sampling** picks contributing sources with probability proportional to their degree. It simulates a sampling scenario in which well-connected sources are more likely to contribute measurements, for instance, because they are present in bigger cities.

  We also studied other sampling methods that sample sources non-uniformly, including one where each source has a randomly assigned weight and one where some sources are completely unavailable. We find that the bias because of them is similar or less than degree-biased source sampling. We omit results for these other methods from this paper.

What percentage of paths are typically sampled in measurement experiments? To understand the impact of the sampling percentage across the board, we show results for percentages ranging from 1–90%. In §6, however, we show that a real study that uses a setup similar to the one that we outlined above measures 5–50% of the paths. We thus focus on this range when discussing the results. For convenience, the graphs highlight this range.

To sample K paths in a source sampling method, we first pick $\lceil \frac{K}{|D|} \rceil$ sources. All picked sources except the last one measure paths to all destinations. The last source may measure to fewer destinations, which are randomly selected with equal probability.

For faster simulations, we sample paths and sources with replacement. We find that it yields results similar to sampling without replacement except when the sampling percentage is well over 50%. This regime is outside our sampling region of interest.

Evaluation criterion: We evaluate the sampling methods using two measures. First, relative error measures how far the estimated value is from the true value. That is:

$$\text{relative error} = \frac{\text{estimated value} - \text{true value}}{\text{true value}}$$

AS	#Nodes, #Links	
Genuity (1)	42,	110
UUNET (701)	83,	438
Verio (2914)	70,	222
Level 3 (3356)	63,	570
Global Crossing (3549)	61,	972
Cable & Wireless (3561)	92,	658
Globix (4513)	13,	26
AT&T (7018)	115,	296

Table 1: Rocketfuel topologies used in our study.

The estimated mean for a sampling method is simply the mean of the sampled paths.

Sampling methods should provide not only the best estimate of the true value but also a confidence interval within which the true value lies with a high probability. This interval lets measurement consumers judge the uncertainty in the estimate. When too few paths are sampled, the possibility of a high relative error is unavoidable even if sampling is unbiased. However, regardless of how many paths are sampled, the true value should lie within the estimated confidence interval with a high probability.

Our second measure computes how often the true value lies within the estimated confidence interval. We compute confidence intervals around the mean using Student's t-distribution [3], treating each path measurement as an independent observation. The alternative of treating all measurements from a source as one independent observation leads to much wider confidence intervals as fewer independent observations are available. Excessively wide confidence ranges are not useful. In the extreme, a range of $-\infty$ to ∞ always contain the true value, but its practical utility is minimal. Hence, we examine the sizes of the confidence intervals as well when comparing their frequency of containing the true value.

We compare sampling methods by providing each with the same number of paths to isolate the estimation error induced by sampling too few paths from the bias introduced by the sampling method.

Network topologies: We use Rocketfuel's inferred city-level topologies of ISP networks, shown in Table 1 [25]. These topologies are annotated with link latencies and approximate link weights used for routing [17]. The annotations let us compute routing paths and latencies between any pair of nodes. We consider each node in a topology to be a source as well as a destination of interest.

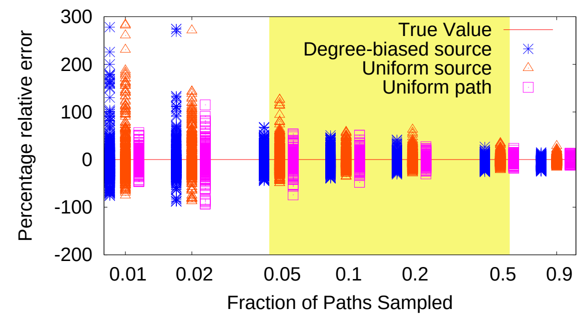
Rocketfuel topologies are a good vehicle for our work because they are reasonably complete. They let us estimate the true value of the network property of interest and compare that to what is obtained in a sampling experiment. (Sampling experiments do not typically uncover the whole topology.)

To verify that our results are general, we also use 100-node synthetic topologies. These topologies have heavy-tailed node degree distribution [27] and are generated using Brite [19].

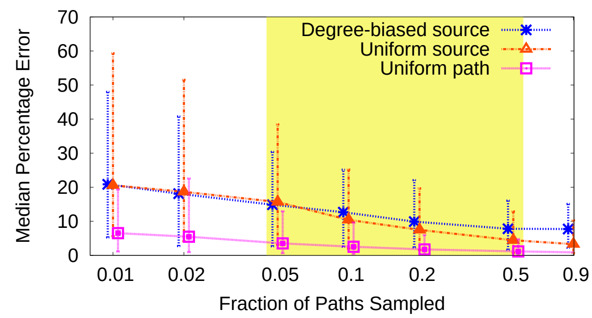
2.1 Estimating Path Latency

Figure 1(a) depicts a scatter plot of the error in estimating mean latency for Rocketfuel topologies as the percentage of paths sampled is varied from 1–90%. The shading in the graph corresponds to the sampling range of interest (5–50%). For each combination of topology, sampling method and sampling percentage, we perform 500 experiments with different random seeds.

We see that the error with source sampling is much higher than that with uniform path sampling. When 5% of the paths are sampled, the estimated mean can be off by more than 100% (i.e., a factor of two) with source sampling. In contrast, the error is under 50%



(a) Scatterplot of relative error in estimating mean path latency.



(b) Median, 10th and 90th percentile of the magnitude of the relative error

Figure 1: Error in estimating mean path latency for Rocketfuel topologies. Source sampling methods have high relative error.

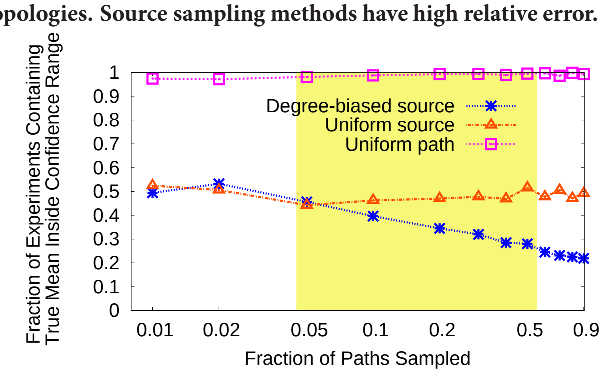


Figure 2: The frequency with which the true mean is within the estimated 99% confidence interval. The true mean lies outside the confidence intervals more than half the time for source sampling methods.

with uniform path sampling. As more paths are sampled, the relative error reduces for all sampling methods. But even when 20% of the paths are sampled, the relative error with source sampling can be as high as 50%, while it is below 20% with uniform path sampling.

For the same data, Figure 1(b) plots the median, 10th and 90th percentiles for the magnitude of the relative error. When 5% of the paths are sampled, the median error with source sampling (15%) is worse by a factor of five compared to the median error with uniform path sampling (3%).

The higher relative error of a sampling method may be tolerable if the uncertainty in the estimate is correctly captured. That is, the true value lies within the estimated confidence interval. Figure 2 shows that this property often does not hold for source sampling. Almost half the time, the true mean is outside the estimated confidence interval. For uniform path sampling, the true value is within the con-

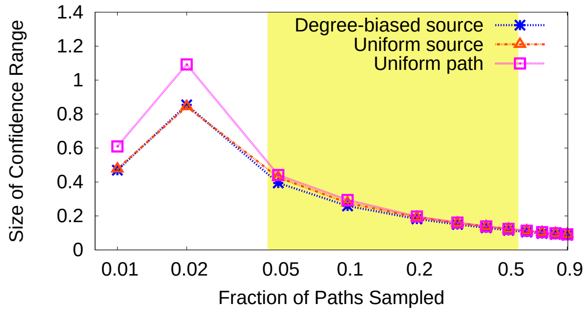
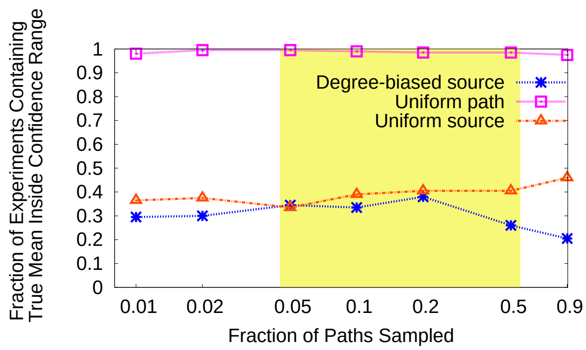
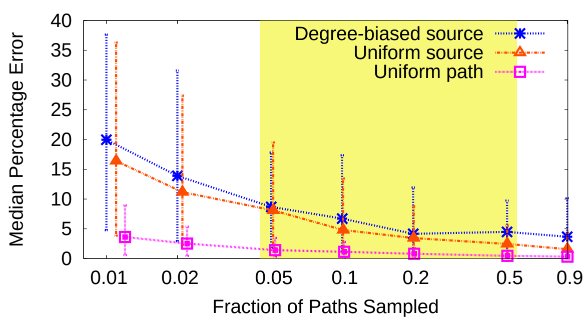


Figure 3: The size of confidence intervals normalized by the true mean. All sampling methods have similar interval sizes.



(a) Fraction of experiments where the true mean is within the 99% confidence interval.



(b) Median, 10th, and 90th percentile of the magnitude of the relative error.

Figure 4: Bias in latency measurements for Brite topologies. Source sampling methods yield inaccurate inferences.

fidence intervals with a very high frequency. Thus, not only do the source sampling methods have higher error, they also are unable to tell when their estimates are inaccurate.

Of course, by estimating large confidence intervals, any sampling method can produce confidence intervals that contain the true mean with a high probability. Figure 3 shows that the ability of uniform path sampling to contain the true mean does not arise from bigger confidence intervals. All three methods have similar confidence interval sizes.

Figure 4 confirms that the bias in latency inferences is not limited to Rocketfuel topologies. A similar sampling bias exists for Brite topologies as well.

While the results above are for mean latency, we show in §5.2 that erroneous estimates due to bias plague other aggregate measures such as median and 90th percentile as well.

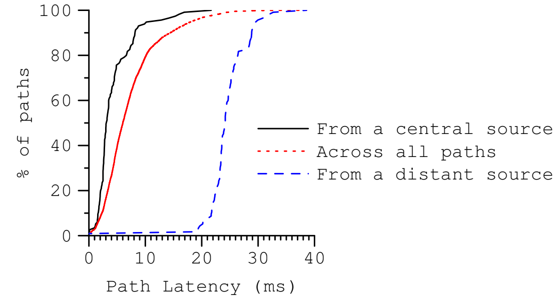


Figure 5: The distribution of path latencies in the entire network and from two sources in the AT&T network. Latency distributions of individual sources are different from those of the entire network.

2.2 Understanding Bias in Source Sampling

Why does source sampling produce biased results even when sources are sampled uniformly at random? The central reason is that the distribution of path latencies from any given source can be very different from the distribution across all paths.

Consider an ISP network spread across the continental USA. If we measure paths from a node that is in one corner of the country (e.g., Miami), the mean latency will likely be higher than the true mean. Conversely, if we measure paths from a more central location (e.g., Chicago), the mean will be likely be lower. Figure 5 shows this effect. For the AT&T network, it plots the path latency across all paths, for paths from a central source, and for paths from a distant source. The three latency distributions are different.

Thus, each source provides a biased window into the path latency distribution. The combination of multiple, randomly selected sources is not guaranteed to fix this bias. Uniform path sampling on the other hand does well by sampling directly from the true underlying distribution.

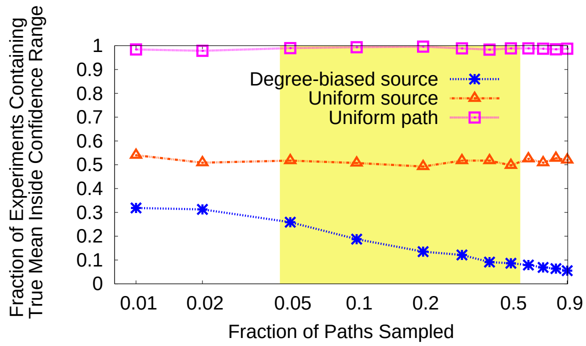
In theory, source sampling can sample unbiasedly if we sample only one path to a randomly selected destination from each source. But this constraint drastically reduces the number of available measurements. In a network with a hundred destination nodes, sampling only one path per source reduces the number of available samples by two orders of magnitude. With such a reduction, even if sampling is unbiased, the relative error can be significantly higher and the uncertainty in the estimate will be high. It is preferable instead to collect more samples and correct the data for any bias. In Section 4, we present three potential correction techniques.

2.3 Estimating Properties Besides Latency

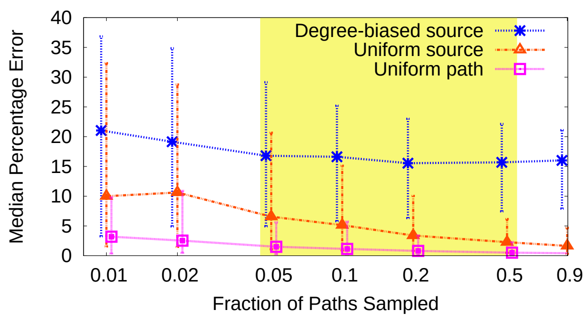
The presence of bias is not unique to path latency measurements but plagues other path properties as well. In this section, we show this effect for hop count and path capacity measurements.

Hop count: Figure 6 shows the results for hop count measurements. We see significant bias in terms of both the relative error and the ability to estimate the confidence interval around the mean.

Another factor that stands out in these graphs is that degree-biased source sampling is much worse than uniform source sampling. To explain this difference, Figure 7 plots the CDF of hop counts observed with the two source sampling methods across all experiments. We see that uniform source sampling results in a distribution that closely approximates the distribution of the set of all paths (though the degree of error may differ across the measurement experiments). On the other hand, the distribution of degree-biased source sampling is consistently to the left of the true distribution. This discrepancy occurs because nodes with high degree, which are

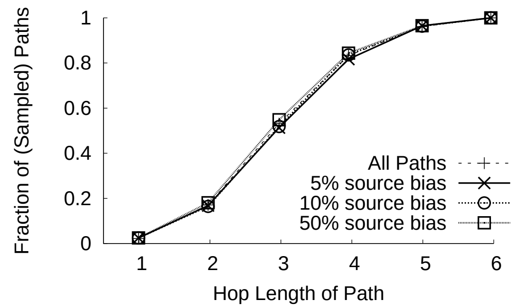


(a) Fraction of experiments where the true mean is within the 99% confidence interval.

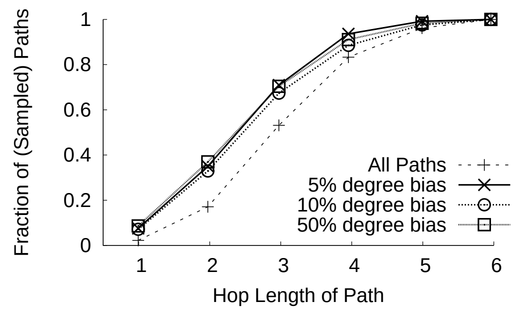


(b) Median, 10th, and 90th percentile of the magnitude of the relative error.

Figure 6: Bias in path hop count measurements with different sampling methods. Source sampling methods yield inaccurate inferences. Degree-biased source sampling is particularly poor.



(a) Uniform source sampling



(b) Degree-biased source sampling

Figure 7: Hop count distributions observed with the two source sampling methods. Degree-biased source sampling produces a view in which paths have fewer hops.

more likely to be chosen as measurement sources, happen to be centrally located in the graph; paths from them to various destinations tend to have fewer hops than those from other sources. This difference implies that degree-biased source sampling creates a more biased view of path hop count.

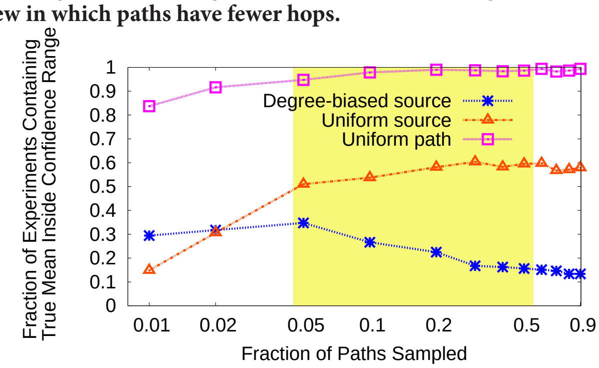
Path capacity: Figure 8 shows the bias in path capacity measurement with different sampling measurements. For link capacities, we use the values assigned by Kandula *et al.* [13] for the same Rocketfuel topologies. They use a simple model in which link capacities have a bimodal distribution between 2.5 and 10 Gbps; the capacity of a link is based on its centrality in the topology. Thus, path capacities in our experiments also have a bimodal distribution between the two values. Despite this simplicity of the underlying property, our results show that inferences with source sampling methods can be significantly flawed due to bias.

3. GENERAL APPROACHES TO CORRECT FOR BIAS

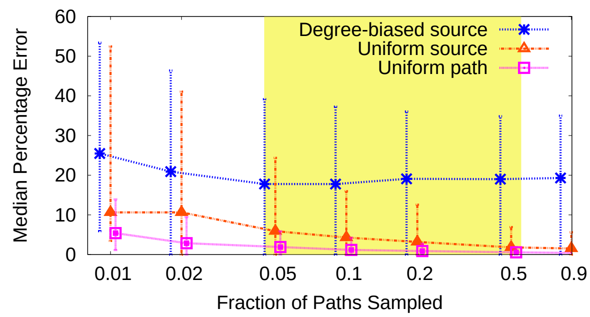
Before we present the bias correction methods that are part of our Broom toolkit, we survey in this section general approaches that we find being employed in other domains.

3.1 Minimizing source dependency

Since each starting point (that is, a measurement source in our terminology) introduces a bias of its own, one approach to reducing bias is to minimize the impact of the starting point. Sociological experiments, for instance, to estimate the drug use habits of addicts, often start with biased sources (e.g., users personally known to the researcher) and recursively recruit acquaintances of existing users.



(a) Fraction of experiments where the true mean is within the 99% confidence interval.



(b) Median, 10th, and 90th percentile of the magnitude of the relative error.

Figure 8: Bias in path capacity measurements with different sampling methods. Source sampling methods yields inaccurate inferences. In particular, degree-biased source sampling has high error even when more than 50% of paths are sampled.

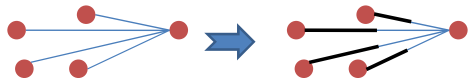


Figure 9: Extrapolation from the tail half avoids bias from repeatedly sampling portions of the paths close to the source.

The drug use habits across all users thus surveyed can be highly biased by the initial set. To minimize this bias, experimenters keep track of connectivity among the sampled users and after-the-fact ignore users within a few hops of the initial set [11, 12]. Under certain conditions, this filtering yields unbiased steady state distribution.

Another technique to minimize source dependency has been used for sampling graphs such as peer-to-peer networks [26]. Experimenters employ a random walk in the graph structure to arrive at another node before starting the measurement.

There are two potential ways to apply this approach in our setting. The first is to use source routing such that measurement probes are bounced off of a random node and only the part of the path from the redirecting node to the destination is considered. Because source routing is not widely deployed in the Internet, this technique is unlikely to be practical.

An alternative is to ignore some part of the path that is close to the source and extrapolate from the rest. We present a simple method based on such tail extrapolation in §4.1.

3.2 Explicitly computing source contribution

In some settings, the bias introduced by a source can be inferred in a way that makes it possible to remove it from the measurements. An example of this setting is the bias of an MRI (magnetic resonance imaging) machine. Correlating multiple images taken by the same machine enables estimation of the imperfections of an MRI machine [15]. These imperfections can then be subtracted from the individual images to obtain versions that are not tainted by machine-specific bias.

We present in §4.3 a method based on path decomposition that is based on this approach.

3.3 Using known properties of the system

Another approach to correct for bias is to leverage known properties of the system being sampled. A method that belongs to this approach is used by election surveys [21, 23]. These surveys leverage prior knowledge about the underlying distribution of likely voters along several key dimensions (e.g., race, age, sex, etc.). This allows survey collectors to randomly sample the entire population instead of first identifying likely voters. The bias due to the difference in the two distributions can then be corrected by re-weighting the results of each class of voters using their known ratio among likely voters.

This method can be adapted to our setting if we knew, for instance that there were well-defined link or path types (e.g., local, national) and their relative frequency of occurrence. Then, based on the measured latencies of each type and this frequency distribution, we could re-weight the measured data as above to estimate the mean latency in the network. However, such a breakdown into types is not known for links (or paths) in the Internet.

We present instead in §4.2 a bias correction technique that uses the property that network distances can be well-approximated by placing the nodes in a low-dimension coordinate space [8, 16, 20].

3.4 Using control groups

The use of control groups is prevalent in medical experiments and drug trials [9]. To account for the uncontrollable sources of bias (e.g., income level, race, age, etc.) that might affect test results, it is a de-facto standard to recruit a control group that has the same statistical properties as the group undergoing evaluation and validate the results achieved in the primary group by comparing against the control group. We do not know how this approach can be adapted to our setting.

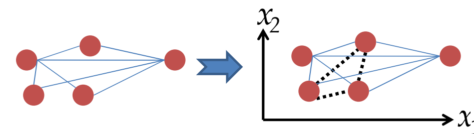


Figure 10: Embedding nodes in a coordinate space, such that distances in the space approximate the path property being estimated. This embedding helps correct for bias by providing an estimate for paths that are not measured directly.

4. THE Broom TOOLKIT

Inspired by the high-level approaches outlined above, we now present three methods to correct for bias in network path measurements. These methods place no burden on the measurement process and they can be applied to the data after collection. Each method makes different assumptions about network paths, which we discuss below. We evaluate these methods in the next section.

4.1 Extrapolating the Tail Half

Our first method attempts to correct for bias by minimizing the source dependency in the measured data. It follows a simple intuition. With source sampling, a major factor underlying the bias stems from repeatedly sampling links that are close to the selected sources. If these links happen to be longer or shorter than the average, the data will be non-representative and the estimate erroneous. Because links closer to destinations are more extensively sampled in a measurement experiment, it is likely that the path estimates based on the tail half of the path will be more representative.

Hence, as shown in Figure 9, we may reduce bias by considering only the tail half. We use hop count to divide the path into two halves. That is, for a four-hop path, the tail consists of the last two hops. We then extrapolate the values of full paths using the tail halves. The extrapolation is property dependent. For latency, we use double the tail value, and for capacity the value is unchanged. Tail extrapolation assumes that the measurements yield a per-hop breakdown in the property being estimated. For some properties, such as latency, providing this breakdown is straightforward (e.g., using traceroute) but for others, such as capacity, it may be difficult.

To effectively remove bias, this method makes two assumptions. First, the distribution of number of hops per measured path should be similar to that for the overall distribution. Second, the distribution of the per-link property values (e.g., link latency) for path tails should be similar to the overall distribution. If either of these conditions are not true, the extrapolation will be biased.

In §5, we find that these assumptions can be too strong in some settings. In particular, tail extrapolation provides little benefit with degree-biased source sampling because the first assumption does not hold with such sampling. Nevertheless, we include tail exploration in Broom because of its simplicity, and as we show in §5, it reduces some types of biases.

4.2 Embedding Into a Coordinate Space

Our second method attempts to correct for bias by using a known property of the network. The property in question is that path measures such as latency can be well-approximated by embedding the nodes into a low-dimensional coordinate space; that is, the path

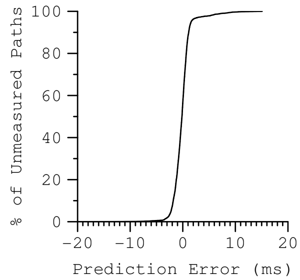


Figure 11: Error in predicting latency for the unmeasured paths in the AT&T network after embedding the nodes in a coordinate space using the measured 10% of the paths. The prediction error is low for the vast majority of the unmeasured paths.

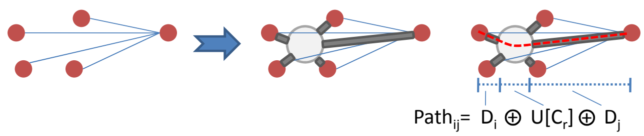


Figure 12: Decomposing a path into two node-specific components and a random hop on a shared ring. Estimating these components helps correct for bias by providing an estimate for paths that were not directly measured.

measure between two nodes can be approximated by the distance between their coordinates. Several researchers have shown this property to hold in the Internet [8, 16, 20].

Such an embedding can be used to correct for bias, as illustrated in Figure 10. The measured data can be used to embed nodes into a coordinate space. Once an embedding is obtained, it lets us estimate the path measure for paths that could not be directly measured (i.e., the dotted paths in the figure). These estimates, which were previously missing, let us create a more complete and thus less biased view of the network.

We embed nodes into a coordinate space by considering each path measurement as a constraint — the coordinate distance between two nodes should roughly equal the measured value. The embedding process satisfies as many constraints as possible. If the measure being estimated is embeddable [8, 20], e.g., triangle inequality violations are not common, the resulting coordinate distances closely approximate the path measure. For computing appropriate coordinates, we use an existing implementation of Vivaldi [2] as a black box. It uses a four-dimensional coordinate space along with a node-specific "height" factor [8]. After coordinates have been assigned to all nodes in the network, we can estimate measures for paths in the system that have not been directly measured.

Figure 11 shows the result of this process for an example experiment using the AT&T network in which 10% of the paths are sampled using degree-biased source sampling. It plots the error in the predicted latencies of the unmeasured 90% of the paths based on embedding the nodes in the coordinate space. We see that the predictions are quite accurate for the vast majority of the paths. Combining these predictions with measured data yields a more accurate view of the network than that obtained using measurements alone.

4.3 Path Decomposition

Our third method attempts to correct for bias by explicitly computing the source (and destination) contributions. We model network paths as concatenation of a component that is unique to the source, a component that is shared across paths, and a component that is unique to the destination. This model is motivated by the nature of topology and routing in the Internet. Packets along a path typically traverse from the source to the network core, then traverse within the core, and finally traverse from the core to the destination.

Term	Meaning
i, j	Indices for nodes
a, b	Indices for rings
V_{ij}	Measured path value between nodes i and j
C_a	Circumference of Ring a
$U[C_a]$	Random number sampled uniformly in [0, C_a]
D_i	Height of node i away from its ring
D_{ab}	Inter-ring distance between rings a and b
S_{ij}	Estimated path value between nodes i and j
N_a	Number of nodes in Ring a
R_i	Ring to which node i belongs
R, N	Number of Rings, Nodes respectively

Table 2: Definition of terms in the path decomposition problem.

This model can be used to correct for bias if we can approximate the source, destination, and core components. That approximation lets us estimate the measure for paths in the network that were not directly measured. Like coordinate embedding, this estimation creates a more complete view of network. But unlike coordinate embedding, which has been shown to work only for path properties such as latency, path decomposition is general and applies to path properties such as capacity and hop count as well.

Figure 12 shows a simplified view of how we approximate path components. We model the network core as a ring and posit that the path between a source-destination pair traverses a random distance inside that ring. By formulating a constraint satisfaction problem over the measured data, we obtain best-fit values for the ring size and the distance of each node from the edge of the ring. From these best fit values, we can estimate the path measure for unmeasured paths (e.g., the paths between the four nodes on the left in Figure 12). To drive the decomposition towards more realistic network topologies, we also impose parsimonious constraints such as minimizing the sum of the ring size and the node distances. For the example in Figure 12, because of such constraints, the benefit of reducing the distances for all other nodes outweighs the cost of assigning a larger distance to the one source on the right.

Modeling the network core as a single ring is problematic, however. For topologies that tend to be combinations of clusters, e.g., because they span continents, using only one ring over-constrains the underlying space and the estimates of ring sizes and distances are poor approximations of measured values. See the grayed out possibilities in the middle of Figure 13. Ideally, we would like to posit precisely as many rings as necessary to obtain a good fit. See the decomposition on the right in Figure 13.

So we generalize our model to allow multiple rings. Each node attaches to exactly one ring. As before, we infer best-fit values for the distance from a node to its ring and the ring sizes. Additionally, we also infer the inter-ring distances. For nodes that attach to the same ring, the path between them is modeled as before. The path for nodes that belong to different rings traverses the inter-ring component, the two rings and the two node-specific components.

Formally, path decomposition solves the following optimization problem, where Table 2 defines the notation:

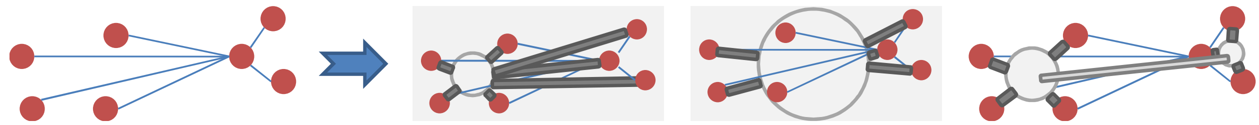


Figure 13: Approximating the network on the left with just one ring in the core overestimates the latency for many paths. The two grayed out decompositions in the middle show this effect. We model network core using multiple rings, which leads to decompositions such as the one on the right. This decomposition better matches the measured data.

$$\min \sum_{i,j} (V_{ij} - S_{ij})^2 + \sum_a N_a C_a + 0.1 \sum_i D_i + \frac{0.1 \sum_{ab} D_{ab}}{R-1}$$

s.t.

$$S_{ij} = D_i \oplus D_j \oplus (1 + U[\epsilon]) \begin{cases} U[C_{R_i}] & if\ R_i = R_j \\ D_{R_i R_j} \oplus U[C_{R_i}] \oplus U[C_{R_j}] & o/w \end{cases}$$

$C_a, D_{ab} \geq 0 \ \forall$ rings

The definition of S_{ij} describes how the path property can be composed from the node and ring components. Mathematically, this composition (represented by $\oplus$) depends on the property of interest. For additive metrics, such as latency, hop count or (log of) loss rate, it involves addition. For others, such as capacity, it involves taking the minimum of the three portions. The $U[*]$ values are assigned at run time using a random number generator. The factor $1 + U[\epsilon]$, where ϵ is a small constant, ensures that a different path decomposition is obtained each time the problem is solved. It helps compute the uncertainty in the measured data (see §4.4).

In the minimization goal, $\sum (V_{ij} - S_{ij})^2$ ensures that estimated path latencies S_{ij} lie close to the measured value V_{ij}. The rest of the goal imposes a parsimony constraint. We attempt to minimize the distances of nodes to the ring D_i, the ring sizes C_a and the inter-ring distance D_{ab}. This minimization and the relative weights of 0.1 for the last two terms reflects the following concerns:

- We should change the size of a ring, which impacts latencies for all nodes belonging to the ring, only if it leads to a better fit in the distances of many nodes that attach to that ring.
- We should change the inter-ring separation, which impacts the latencies for all paths spanning the rings, if it results in a better fit for the sizes of the rings and hence latencies for paths within the rings.

The optimization problem above takes the number of rings and node to ring mapping as input. Formulating a single optimization problem that determines these two factors along with component sizes leads to a problem that is too slow to solve. We instead use a simple clustering procedure to obtain these two factors as a pre-processing step. We divide nodes into two groups. The first group, called *close-by*, is the largest subset of nodes such that no pair has a path value greater than a threshold k. The remaining nodes are in the *faraway* group. We then recursively split the *faraway* group in a similar manner. If the *close-by* group has only one node at some recursion level, that node is merged with the *close-by* group from the previous level instead of being in its own group. We merge such nodes because a single distant node (or, more generally, multiple distant nodes that are not close to each other) can be modeled with one ring. Only when multiple distant nodes are close to one another, we need an additional ring to model the data. Each group thus obtained defines a ring of its own.

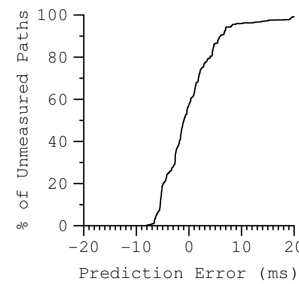


Figure 14: Error in latencies predicted using path decomposition for the same sampling experiment as Figure 11. Prediction errors are low but higher than those for coordinate embedding.

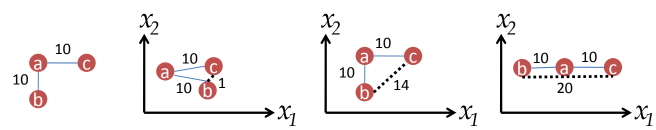


Figure 15: If the network is under-sampled, there are multiple coordinate assignments that match measured data. We capture this uncertainty by computing confidence intervals based on multiple random assignments.

We choose k to be slightly smaller than the median observed path value. This choice is based on the trade-off that for obtaining a decomposition that well approximates the measured data, an extra ring that may not be needed is better than omitting a needed ring.

Once various node-to-ring and inter-ring component values are computed, we can estimate the values of unmeasured paths by appropriately composing the components. Figure 14 shows the result of this process for the same sampling experiment as Figure 11. We see that the prediction error is low but higher than coordinate embedding. This higher error stems in part from the fact that our formulation of path decomposition is geared towards reducing the mean error rather than the error for individual paths. For example, even if a pair of nodes happens to have a short latency path between them, their shortest distance in the path decomposition embedding would still consist of a random hop on a ring. The formulation makes up for such over estimation of small values by corresponding under-estimation of the larger values since the stable point for the minimization goal preserves the mean. As we show later, this formulation is a good vehicle for estimating means, but is inaccurate for estimating percentiles on either the low or the high ends.

4.4 Capturing uncertainty in the estimates

Coordinate embedding and path decomposition are essentially constraint solving problems. When not enough measured data is available, these problems are under-constrained. That is, there are multiple (non-isomorphic) coordinate assignments or path decompositions that satisfy the measured values. This scenario is illustrated in Figure 15. In such cases, the network configuration obtained is a random one amongst all feasible ones. If we were to base our estimate on only one possible configuration, the estimated value could be significantly different from the true value.

Thus, along with an estimated value, we want to also compute a confidence interval such that the true value is contained in the interval with a high probability. Ideally, we would directly obtain the two extreme network configurations that match the measured data and compute the minimum and the maximum value for the measure of interest. However, we do not know of a way to directly infer such configurations. (Vivaldi and other scalable coordinate embedding systems assign coordinates based on local actions rather than a central global optimization.)

Instead, we estimate the confidence interval empirically, by executing the underlying constraint solver multiple times. The inputs to the solver are randomly perturbed such that the output is essentially a different random network configuration of all feasible ones. For Vivaldi, we randomize initial node positions. For path decomposition, different values for the various random numbers help in obtaining different decompositions.

Once multiple network configurations are derived as above, we compute K-th percentile confidence interval for given values of K. The lower end of this interval is the K-th percentile of the paths in the hypothetical network configuration in which the value of each path is the minimum observed across these iterations. Similarly, the higher end is based on a hypothetical network configuration with the maximum value observed for each path. We output the best estimate of the mean as the half-way point between these two extreme network configurations.

These hypothetical networks may not be consistent with the measured data but the bounds thus computed contain the true value with a high probability. In our experiments, we find that as few as ten iterations suffice to provide good confidence intervals, and the incremental value of more iterations is negligible.

While we use it to estimate confidence intervals after the data has been collected, an online version of the procedure above can also guide the measurement process itself if the experimenter wants a tight confidence interval with as few measurements as possible.

5. EVALUATION

We now evaluate empirically the three methods in the Broom toolkit. We use a methodology similar to that of §2. After sampling paths, we post-process the data using each method. We then measure their effectiveness in terms of the relative error in the estimated values and the ability to compute confidence intervals that contain the true value. Our key findings are:

- Tail extrapolation is of limited value. With uniform source sampling, it improves inferences compared to not using any correction method. But it provides no benefit with degree-biased source sampling.
- Coordinate embedding is the best method for correcting bias in path latency measurements. It reduces median error by a factor of five compared to not using bias correction. In the sampling range of interest, its effectiveness approaches that of the ideal sampling method, that is, uniform path sampling. Coordinate embedding is less effective at correcting for bias in hop count and capacity measurements, though it does provide benefit compared to not correcting for bias.
- Path decomposition is broadly useful. It is slightly inferior to coordinate embedding for latency measurements but vastly superior for hop count and capacity measurements. For all three measures, it comes close to ideal sampling in the sampling range of interest when estimating mean.

We now present our results in detail, starting with the task of estimating mean path latency. We then consider estimating aggregate measures other than the mean, followed by estimating path properties other than latency. Unless otherwise stated, the results are for the Rocketfuel topologies (Table 1).

As before, we show results for the entire 1–90% sampling range to study the behavior of Broom's methods across the board and their limits. But we focus on the 5–50% range while discussing the results and highlight this range in our graphs.

5.1 Estimating mean latency

We first consider uniform source sampling in Rocketfuel topologies, followed by degree-biased source sampling. We then show that our key conclusions also hold for the Brite topologies.

Uniform source sampling: Figure 16 shows the results for the case where paths are sampled using uniform source sampling. For comparison, it also shows the results when bias is not corrected and when the same number of paths are sampled using ideal (uniform path) sampling.

Figure 16(a) plots the frequency with which the true mean lies inside the 99% confidence interval. We see that each of Broom's three methods improves over the case where bias is not corrected. Even the simple tail extrapolation method helps, by increasing the frequency from 50% to 75%. When 5% of the paths are sampled, the other two methods improve this frequency to over 90%. Beyond that, they are indistinguishable from ideal sampling.

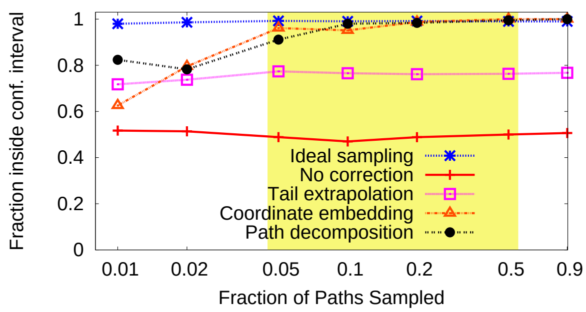
By simply increasing the confidence interval size, it is easy for any method to ensure that the true mean lies inside the interval. Figure 16(b) shows that the size of the confidence interval is not a major factor behind the improvement brought about by Broom. Their confidence intervals are similar to those for the case where bias is not corrected.

Figure 16(c) shows the median, 10th and 90th percentile of relative error in estimating the mean latency of the network. When 5% of the paths are sampled, the median error without bias correction is about 15%. Tail extrapolation reduces this error to 10%. Path decomposition reduces it by half to 7%. Coordinate embedding matches ideal sampling with only 3% median error. When over 20% of the paths are sampled, path decomposition too begins to match ideal sampling.

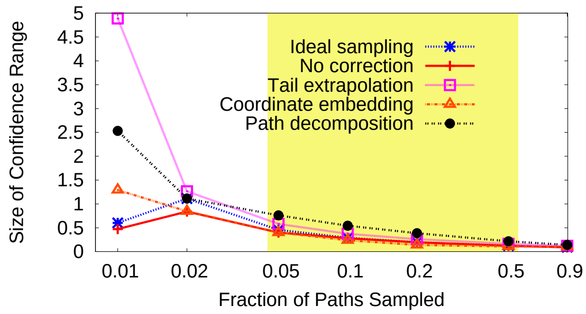
Degree-biased source sampling: We now study the impact of degree-biased source sampling, which we showed earlier as producing more biased results than uniform source sampling. Figure 17 shows that the effectiveness of path decomposition and coordinate embedding is similar to the case of uniform source sampling.

However, tail extrapolation completely falls apart. Its frequency for containing the true mean is similar to not correcting for bias at all, and its relative error is substantial. This ineffectiveness is a result of the fact that the number of hops in paths with degree-biased source sampling tends to be fewer than the true underlying distribution (as shown in Figure 7). Extrapolating from the tail halves of such paths does not reduce bias because even the tail halves are systematically shorter than a typical half-path in the network.

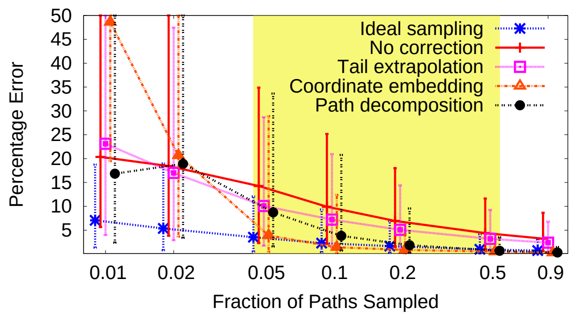
Brite topologies: Figure 18 shows the results for the Brite topologies; we present results only for degree-biased source sampling. We see that the results are qualitatively similar to those for the Rocketfuel topologies. One exception is that tail extrapolation appears to be a little more effective than it is for Rocketfuel topologies with degree-biased source sampling. The continued effectiveness of the



(a) Fraction of experiments where the true mean is within the 99% confidence interval.



(b) 99% confidence interval size normalized by the true mean.



(c) Median, 10th, and 90th percentile of the magnitude of the relative error.

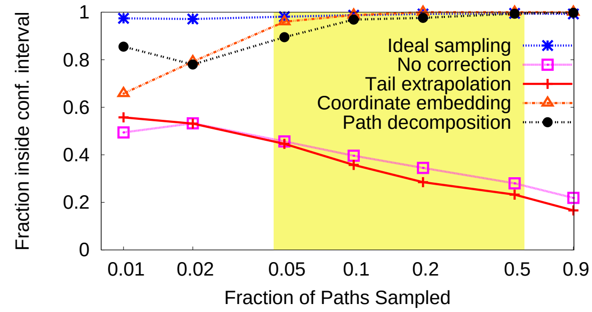
Figure 16: Correcting for bias in latency measurements when uniform source sampling is used. Each of the three methods improves the accuracy of the inferences. Coordinate embedding is the most effective method.

other two methods on Brite topologies suggests that they are robust to the nature of the underlying topology.

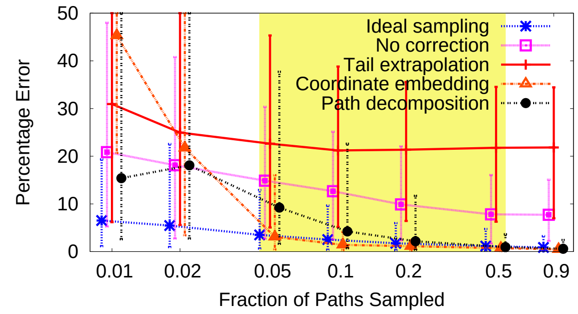
5.2 Estimating other aggregate measures

We now study if Broom can correct for bias in aggregate measures other than the mean. This investigation is preliminary, and a more detailed study is a subject of ongoing work. We consider the task of estimating the 50th (median) and the 90th percentiles in path latency distributions. We quantify estimation error using the magnitude of the difference between the estimated and true values divided by the true mean. The division gives a sense of the significance of the error compared to the mean path latency.

Figure 19 plots the median, 10th and 90th percentile of the error in estimating the two percentiles. We see that coordinate embedding is remarkably effective. In the sampling sampling region of interest, its errors are comparable to those of ideal sampling.



(a) Fraction of experiments where the true mean is within the 99% confidence interval.



(b) Median, 10th, and 90th percentile of the magnitude of the relative error.

Figure 17: Correcting for bias in latency measurement when degree-biased source sampling is used. Tail extrapolation brings no benefit but the other two methods are effective.

Path decomposition, however, is not effective at estimating percentile values. As mentioned before, this shortcoming stems from formulating the decomposition problem in a way that focuses on reducing the error in computing the mean rather than obtaining good estimates for each path. In the future, we will consider formulations using which both the mean and percentile values can be well-approximated.

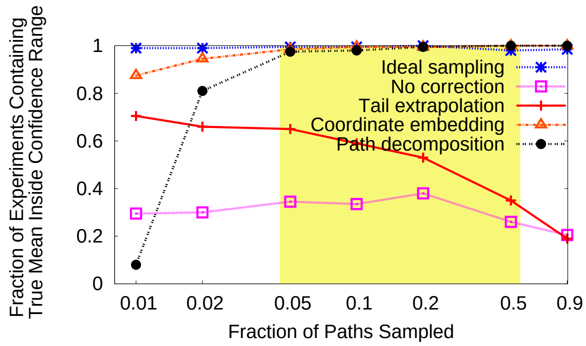
5.3 Estimating other path properties

We now study the effectiveness of Broom for other kinds of path measurements. In particular, we consider hop count and path capacity. We present results only for the more challenging case of degree-biased source sampling.

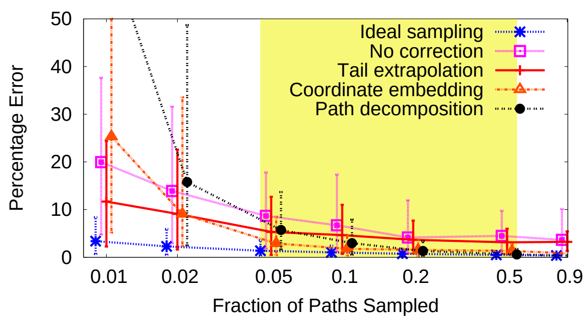
Path hop count: Figure 20 shows the results for hop count measurements. We see that for coordinate embedding this measure presents a more challenging case. It improves results compared to not correcting for bias; when 10% of the paths are sampled, it encompasses the true mean roughly 70% of the time compared to 20% for not correcting for bias, and its median relative error is roughly half as high. But it is far from ideal sampling and is unable to match the effectiveness it showed for latency measurements. This behavior suggests that Internet path hop counts are not amenable to being characterized using a low-dimensional coordinate space.

Path decomposition on the other hand continues to be effective at correcting for bias in hop count measurements. In the sampling region of interest, its effectiveness is similar to that for estimating mean latency.

Path capacity: Figure 21 shows the results for path capacity measurements. As before, we see that coordinate embedding is not as ef-

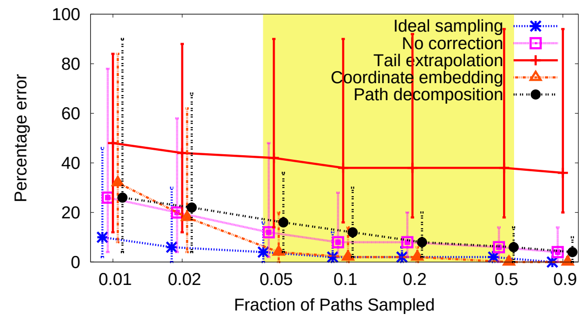


(a) Fraction of experiments where the true mean is within the confidence interval.

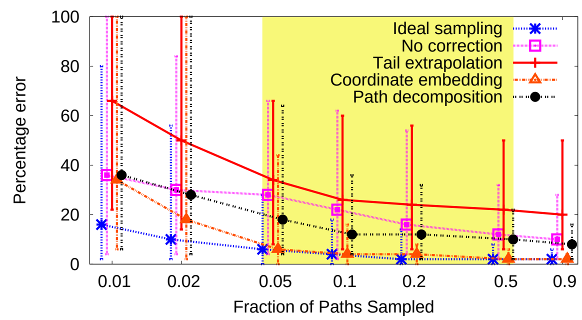


(b) Median, 10th, and 90th percentile of the magnitude of the relative error.

Figure 18: Correcting for bias in latency measurement when degree-biased source sampling is used over Brite topologies. Coordinate embedding and path decomposition are effective.



(a) 50th percentile (median)



(b) 90th percentile

Figure 19: The median, 10th and 90th percentile error in estimating two percentile values for path latency distribution when degree-based source sampling is used. Coordinate embedding provides a good estimate of percentile values as well.

fective as it is for latency, though it improves over the no correction case. In contrast, path decomposition remains remarkably effective. It comes close to ideal sampling whenever more than 5% of the paths are sampled.

6. A CASE STUDY OF APPLYING Broom

We now show that the potential existence of bias and its removal are not merely theoretical concerns. The results of real measurement studies can differ significantly depending of whether the data is corrected for bias.

The data that we use for this case study is collected by Netdiff, a system to quantify and compare the performance of backbone ISPs [18]. Netdiff characterizes ISP performance in terms of path latency. It collects two types of data sets. The first is for paths internal to an ISP. Such paths start from an ISP PoP (point of presence) and end at another PoP. This set of paths is thus similar to those in our experimental setup in the previous section. The second data set is for paths from an ISP PoP to an eventual destination. The set of destinations correspond to BGP atoms that represents a group of IP address prefixes with a common routing policy [1].

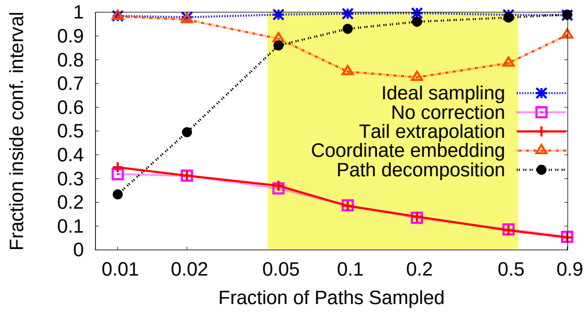
Not all paths of either type can be directly measured in a practical manner. Following standard practices, Netdiff measures the latency of as many paths as it can, in this case from the PlanetLab nodes. It then computes and reports the mean and confidence intervals around the mean for each type of path. The authors assume that the set of measured paths are not biased; see §6.2 of the Netdiff paper [18].

Table 3 characterizes the Netdiff data sets that we use. The data contain measurements of eighteen ISPs collected on one day (Feb 14, 2007).[1] Results for other days are qualitatively similar to those presented below. The second column of the table lists the number of PoPs inside each ISP. The third column shows the number of internal paths measured for each ISP. The fourth column shows the number of destinations present in the data set; the set of destinations measured differs across ISPs. The fifth column shows the number of destination paths that are directly measured. The third and fifth columns also show in parenthesis what percentage of paths of each type are measured. For the first data set, of internal paths that are similar to those we study before, we see that more than 5% of the paths are measured in each case.

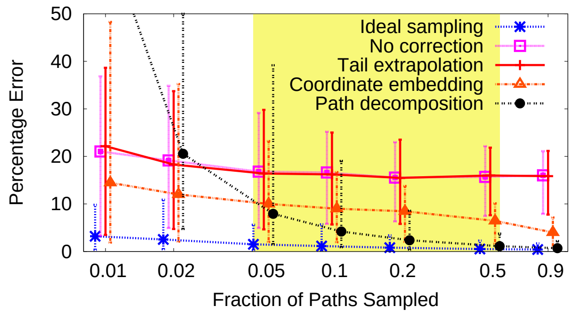
For these two data sets, we computed the mean latency and confidence intervals without bias correction. We also processed the data using coordinate embedding, our best method for latency measurements, which yields bias-corrected estimates of the mean and confidence intervals. We compare the two sets of estimates below.

Data set 1 – Internal paths: Figure 22(a) shows the directly measured mean latency and 95% confidence interval and the latency estimates after correcting for bias. We see that correction can produce significantly different estimates. For Qwest, GlobalX, and VSNL, the difference is close to 30%. For seven of the eighteen ISPs, the two confidence intervals are non-overlapping, which means that at most one of the two methods captures the true mean. If the confidence interval after correcting for bias contains the true mean, as suggested by the results from the previous section, the pre-correction confidence interval does not.

[1] While some of the Netdiff and Rocketfuel ISPs (Table 1) are common, the topologies are not directly comparable because they were collected roughly six years apart.

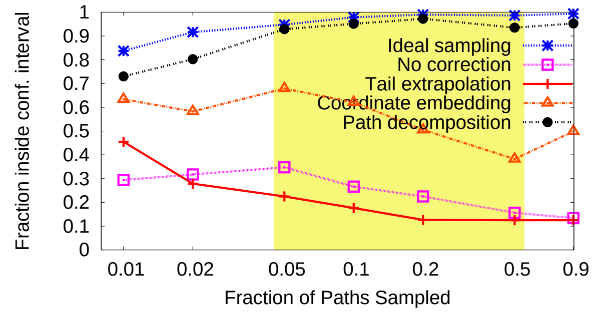


(a) Fraction of experiments where the true mean is within the confidence interval.

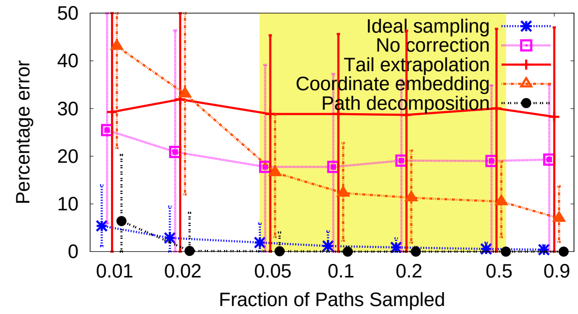


(b) Median, 10th, and 90th percentile of the magnitude of the relative error.

Figure 20: Correcting for bias in hop count measurement when degree-biased source sampling is used. Path decomposition is the most effective method.



(a) Fraction of experiments where the true mean is within the 99% confidence interval.



(b) Median, 10th, and 90th percentile of the magnitude of the relative error.

Figure 21: Correcting for bias in capacity measurement when degree-biased source sampling is used. Path decomposition is the most effective method.

Figure 22(a) also shows that the latency estimates after correcting for bias tend to be higher. We speculate that measured data is more likely to contain paths from PoPs in big cities because it is easier to find measurement vantage points there. The nature of ISP topologies is such that paths from bigger cities tend to be shorter than those from smaller cities [25]. Latency estimates computed after correcting for this bias will thus be higher.

Data set 2 – Destination paths: Figure 22(b) shows the latency estimates for destination paths. For this type of paths, the key difference between directly computed latency and that estimated after correcting for bias is the size of the confidence interval. Correcting for bias produces much bigger confidence intervals. In most cases a small fraction of the paths are measured, and thus it is likely that the measurements do not describe a particularly restrictive way of mapping nodes to a coordinate space. As a result, the space of possibilities with respect to latencies of unobserved paths is higher, which translates to bigger confidence intervals.

Results after correcting for bias indicate that there is not enough data to characterize most ISPs' performance to measured destinations. Directly computed latency and confidence intervals (likely erroneously) suggest otherwise.

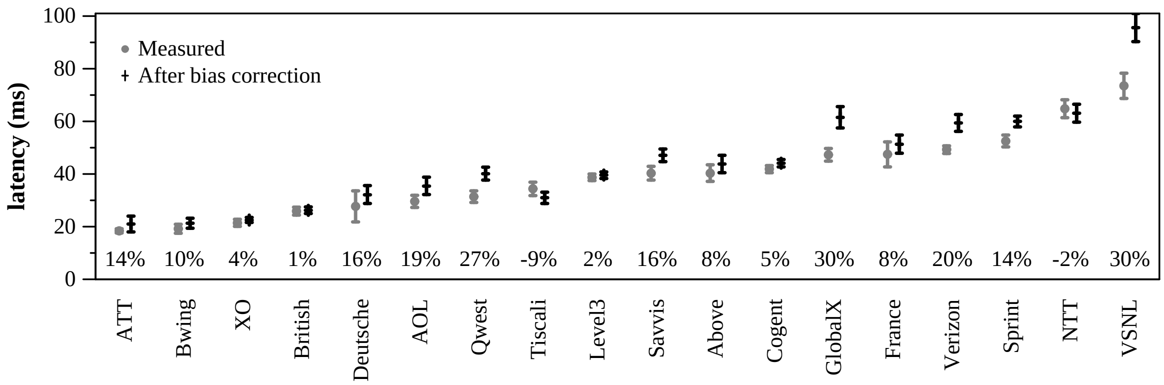
7. FUTURE WORK

The work in this paper points to a few promising directions for further investigation.

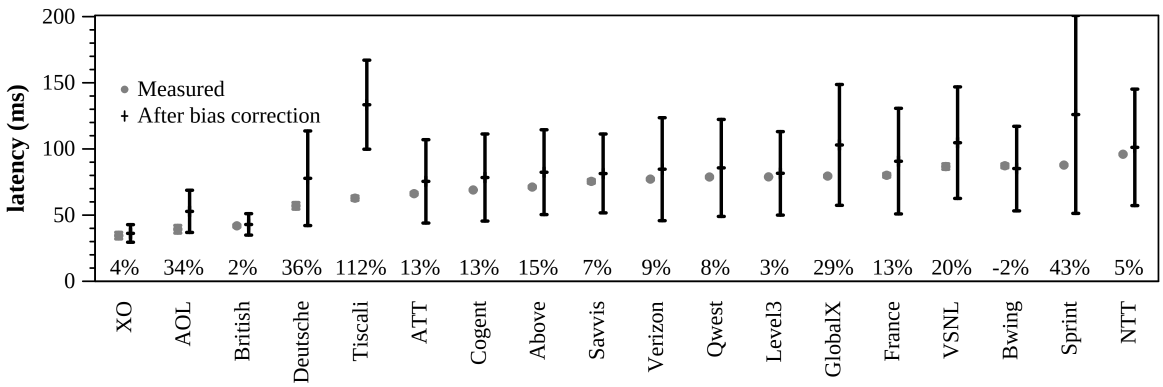
Analytical modeling of bias correction: Our bias correction methods are essentially empirical but we believe that their function can be analytically modeled and explained. Such modeling can provide more insight into why they work. A starting point towards developing these analytic models is recognizing that at their core coordinate embedding and path decomposition are *stochastic imputation* techniques. Stochastic imputation refers to compensating for missing data in measurements by instantiating an underlying model that can estimate the values of missing data. Both of our methods do such compensation, though the models they use are different — low-dimensional coordinate space versus composition of network paths from individual components. Formalizing these models and

	# PoPs	# internal paths (% of total)	# dst. in trace	# dst. paths (% of total)
AOL	26	263 (40.5)	383	1065 (10.7)
ATT	112	881 (7.1)	7138	15350 (1.9)
Above	20	277 (72.9)	5483	19818 (18.1)
British	31	463 (49.8)	650	6675 (33.1)
Bwing	23	165 (32.6)	4074	10337 (11.0)
Cogent	70	1885 (39.0)	12808	69977 (7.8)
Deutsche	50	132 (5.4)	1961	3066 (3.1)
France	25	249 (41.5)	3853	10165 (10.6)
GlobalX	58	698 (21.1)	9506	28439 (5.2)
Level3	62	1595 (42.2)	13847	102138 (11.9)
NTT	40	567 (36.3)	10499	43711 (10.4)
Qwest	51	759 (29.8)	11426	36792 (6.3)
Savvis	40	541 (34.7)	3011	11294 (9.4)
Sprint	53	1238 (44.9)	10924	60891 (10.5)
Tiscali	36	333 (26.4)	2610	7478 (8.0)
VSNL	41	557 (34.0)	2019	6066 (7.3)
Verizon	154	2209 (9.4)	8961	32328 (2.3)
XO	45	558 (28.2)	310	2178 (15.6)

Table 3: Characteristics of the Netdiff data sets.



(a) Internal paths (between PoPs)



(b) Destination paths (from PoPs to destination atoms)

Figure 22: Latency estimated using two different methods for the two kinds of paths. In gray, the graphs show the directly measured mean and the 95% confidence intervals around the mean. In black, they show the estimated mean and 95% confidence intervals after bias correction. The numbers close to the x-axis show the relative difference between the two means. In each graph, the ISPs are ordered based on the mean of direct measurements. Observe that bias correction significantly changes the estimated mean and confidence internals for many ISPs.

the process for conducting measurements on them can provide an analytic basis for both estimating and correcting bias.

Using additional information for bias correction: The current set of methods in Broom only requires basic information that is available in all path measurement experiments — measured path values between pairs of nodes. (Tail extrapolation assumes the availability of per-hop values as well.) A question that remains open is whether leveraging additional information that may be available in some experiments can improve bias correction. We plan to investigate the types of additional information that are commonly available and how they can help correct for bias.

For example, assume that topology information is available, perhaps through out-of-band channels such as public ISP maps. Assume also that routing information, i.e., the series of links traversed between pairs of nodes, is available. Then, using measured path values along with this additional information, we can infer per-link values (e.g., latency). One way to do this inference is by solving a set of equations that constrain the composition of the per-link values along a path to equal the measured path value (akin to Mahajan *et al.* [17]). These inferred per-link values can then be used to estimate the path values for unmeasured paths. Of course, for such a method to be useful when too few paths are sampled, we would also need a mechanism to capture the uncertainty in the estimates.

Alternate network models: Other network models that can work with the same basic information as Broom exist. One such possibility is presented by Ramasubramanian *et al.*, who show that Internet paths can be modeled as a tree for properties such as latency and capacity [22]. Investigating the effectiveness of these other network models at correcting bias is open for future work.

8. RELATED WORK

While we investigate the presence and reduction of bias in network path measurements, other researchers have studied bias in network topology measurements. Lakhina *et al.* focus on biases in measuring router degree distributions [14]. They show that for realistic network topologies this bias can be high unless a large number of sources are used for measurements. They present a statistical test for detecting bias but do not propose methods to correct it. Chang *et al.* show that AS-level connectivity graphs inferred from RouteViews-like sources are biased in that they tend to miss a higher proportion of a particular kind of inter-AS edges [5]. He *et al.* propose methods to reduce this bias by combining information from multiple sources [10].

Researchers have also investigated biases in sampling peers in several large peer-to-peer networks [26]. The methods in this domain rely on random walks, which are typically not possible for Internet path measurements.

A related line of work develops methods to obtain a representative view of the network by carefully selecting a subset of paths to measure [6, 7, 24]. These works assume that any relevant network path can be directly measured. The ability to measure any path is present in some settings; for instance, the network path between any pair of overlay nodes can be measured. But it is typically absent in Internet path measurements, which is why we focus on removing bias from whatever path measurements are available.

9. CONCLUSIONS

We showed that the inferences obtained using prevalent methodology for network path measurements can be highly inaccurate because of sampling biases. For instance, the estimated mean path latency can be a factor of two off the true mean. We presented the Broom toolkit with three methods to reduce sampling biases in measured data without burdening the measurement process itself. We showed that two of those methods are highly effective. One of them uses coordinate embedding. It is extremely effective for latency measurements, where it came close to ideal, bias-free sampling. The other is based on decomposing the path into source, destination, and network core components. It is broadly useful because it can correct for bias in a wider class of path properties such as hop count and capacity. Applying Broom to two real data sets significantly altered their inferences, providing further evidence that prevalent practices run the risk of invalid conclusions unless they correct for sampling biases in their measurements.

Network path measurements, while important and common, represent only one aspect of the overall network measurement landscape. Researchers conduct many other types of measurements as well, and the presence of bias in those measurements has not been explored. We hope that our work triggers a broader debate on sampling bias in network measurements and more active research on understanding and correcting for such bias.

Acknowledgments
This work has benefited from early discussions with Hanhee Song and Yin Zhang. We are also grateful for feedback from John Byers, Mark Crovella, Rob Sherwood, and Walter Willinger.

10. REFERENCES

[1] Y. Afek, O. Ben-Shalom, and A. Bremler-Barr. On the structure and application of BGP policy atoms. In *IMW*, 2002.

[2] S. Agarwal and J. R. Lorch. Matchmaking for online games and other latency-sensitive P2P systems. In *SIGCOMM*, 2009.

[3] H. M. Blalock. *Social statistics*. McGraw-Hill Book Company, 1972.

[4] A. Bryk and H. Weisberg. A developmental perspective on statistical methods for removing bias in quasi-experiments. National Institute of Education Project No. C-74-0125, 1975.

[5] H. Chang, R. Govindan, S. Jamin, S. J. Shenker, and W. Willinger. Towards capturing representative AS-level Internet topologies. *Computer Networks Journal*, 44(6), 2004.

[6] Y. Chen, D. Bindel, H. Song, and R. H. Katz. An algebraic approach to practical and scalable overlay network monitoring. In *SIGCOMM*, 2004.

[7] D. B. Chua, E. D. Kolaczyk, and M. Crovella. Network kriging. *IEEE JSAC*, 24(12), 2006.

[8] F. Dabek, R. Cox, F. Kaashoek, and R. Morris. Vivaldi: A decentralized network coordinate system. In *SIGCOMM*, 2004.

[9] Choice of control group and related issues in clinical trials. http://www.fda.gov/CDER/GUIDANCE/4155fnl.htm, 2001.

[10] Y. He, G. Siganos, M. Faloutsos, and S. Krishnamurthy. A systematic framework for unearthing the missing links: Measurements and impact. In *NSDI*, 2007.

[11] D. D. Heckathorn. Respondent-driven sampling: A new approach to the study of hidden populations. *Social Problems*, 44(2), 1997.

[12] D. D. Heckathorn. Respondent-drive sampling II: Deriving valid population estimates from chain-referral samples of hidden populations. *Social Problems*, 2002.

[13] S. Kandula, D. Katabi, B. Davie, and A. Charny. Walking the tightrope: Responsive yet stable traffic engineering. In *SIGCOMM*, Philadelphia, PA, August 2005.

[14] A. Lakhina, J. W. Byers, M. Crovella, and P. Xie. Sampling biases in IP topology measurements. In *INFOCOM*, 2003.

[15] E. Learned-Miller and V. Jain. Many heads are better than one: Jointly removing bias from multiple MRIs using nonparametric maximum likelihood. In *Information Processing in Medical Imaging*, 2005.

[16] J. Ledlie, P. Gardner, and M. Seltzer. Network coordinates in the wild. In *NSDI*, 2007.

[17] R. Mahajan, N. Spring, D. Wetherall, and T. Anderson. Inferring link weights using end-to-end measurements. In *IMW*, 2002.

[18] R. Mahajan, M. Zhang, L. Poole, and V. Pai. Uncovering performance differences among backbone ISPs with Netdiff. In *NSDI*, 2008.

[19] A. Medina, A. Lakhina, I. Matta, and J. Byers. BRITE: An approach to universal topology generation. In *MASCOTS*, 2001.

[20] T. S. E. Ng and H. Zhang. Predicting Internet network distance with coordinates-based approaches. In *INFOCOM*, 2002.

[21] P. Perry and Gallup. Certain problems in election survey methodology. *Oxford Journals*, 1979.

[22] V. Ramasubramanian, D. Malkhi, F. Kuhn, M. Balakrishnan, A. Gupta, and A. Akella. On the Treeness of Internet Latency and Bandwidth. In *SIGMETRICS*, 2009.

[23] D. Romer, K. Kenski, K. Winneg, C. Adasiewicz, and K. H. Jameison. *Capturing Campaign Dynamics 2000 & 2004*. University of Pennsylvania Press, 2006.

[24] H. H. Song, L. Qiu, and Y. Zhang. NetQuest: A flexible framework for large-scale network measurement. In *SIGMETRICS*, 2006.

[25] N. Spring, R. Mahajan, D. Wetherall, and T. Anderson. Measuring ISP topologies with Rocketfuel. *IEEE/ACM ToN*, 12(1), 2004.

[26] D. Stutzbach, R. Rejaie, N. G. Duffield, S. Sen, and W. Willinger. Sampling techniques for large, dynamic graphs. In *INFOCOM*, 2006.

[27] H. Tangmunarunkit, R. Govindan, S. Jamin, S. Shenker, and W. Willinger. Network topology generators: Degree-based vs. structural. In *SIGCOMM*, 2002.