

Correlations in End-to-End Network Metrics: Impact on Large Scale Network Monitoring

Praveen Yalagandula, Sung-Ju Lee, Puneet Sharma and Sujata Banerjee
Hewlett-Packard Labs, Palo Alto, CA

Abstract—With the ever growing size of the Internet and increasing popularity of the overlay and peer-to-peer networks, scalable end-to-end (e2e) network monitoring is essential for better network management and application performance. For large scale networks, an e2e monitoring infrastructure should minimize the measurement cost while ensuring that the network is still monitored at fine enough time-scales required for each application flow. We explore the relationships between different e2e network metrics with the aim of leveraging such relationships for reducing monitoring costs while maintaining measurement accuracy. We analyze long range network measurements from PlanetLab, where we collected e2e network data (route, number of hops, capacity bandwidth and available bandwidth) for about two years on several thousand paths. We also present a few schemes to leverage the metric correlations and reduce the monitoring cost. Our preliminary results indicate that in some cases, we can reduce the monitoring costs by 75% while maintaining the accuracy at about 88%.

I. INTRODUCTION

The goal of most network monitoring systems is to capture the dynamic state of the end-to-end (e2e) network paths in near-real time so as to enable the control systems to react to these changes. The network monitoring systems thus directly impact the applications and services running on top of the network infrastructure. Since changes are unpredictable, a naïve network monitoring system would measure all end-to-end metrics of interest as frequently as possible on all paths. However, this might consume a large fraction of network resources. For example, metrics such as bandwidth require tools that have significant probing packet overhead and time (seconds to multiple minutes) to obtain a statistically significant estimate. Thus measuring all possible metrics at very high frequencies is not practical. On the other hand, measuring these metrics at low frequencies can cause a monitoring system to miss important network dynamics and handicap the control capabilities impacting the application/service performance.

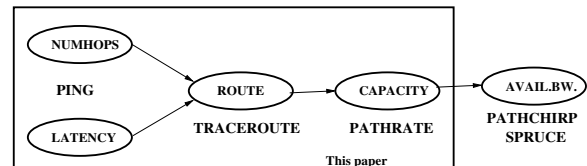


Fig. 1. Correlations in different e2e network metrics along with some measurement tools for each of those metrics.

The goal of our “Network Genome” project is to study the correlations in different end-to-end network metrics and their impact on large scale network monitoring. We study both auto-correlation and cross-correlation for different network metrics on a given e2e path and across the entire network with the goal of leveraging the observed correlations for reducing the network monitoring costs. Note that several previous projects study some subsets of these correlations. For example, systems such as GNP [4], Meridian [10], NetVigator [7], and Vivaldi [1] leverage correlation of e2e latency across different paths and measure only a few paths to infer latency for all other paths. Our research goal is to explore the correlations between other metrics, and if any correlation exists, design monitoring mechanisms that exploit such correlations.

In this paper, we consider the following fundamental question:

Are the changes in various end-to-end network metrics correlated such that changes in a metric with a lower measurement cost can indicate changes in other metrics with a higher measurement cost?

The answer to the above question will provide clues for optimizing large scale network measurement systems.

As a first step, our focus is on exploring the cross-correlations between different end-to-end network metrics as shown in Figure 1. The figure also shows several measurement tools that are available for measuring those metrics. Using the S³ (Scalable Sensing Service) monitoring system [11] on PlanetLab, we have collected route information, capacity, and available bandwidth on

several thousand paths between PlanetLab nodes for about 2 years since January 2006. In this paper, we target the following correlations: (a) route changes and changes in the number of hops and latency and (b) route changes and changes in capacity. If there is a strong correlation between changes in the number of hops, latency, and route, a monitoring infrastructure can use low cost pings to detect changes in the number of hops and latency and invoke relatively expensive traceroutes for the route information only when a change is detected. Note that the remaining TTL value in IP headers of ping responses can be used to determine the number of hops [8], [12]. Similarly, if there is a strong correlation between changes in a route and changes in the capacity on a path, relatively inexpensive traceroutes (in comparison to capacity measurements) can be used to detect a capacity change and only then the expensive capacity measurement can be triggered.

There has been a large body of research in studying various properties of the Internet paths. Paxson [5] presented the end-to-end routing behavior in the Internet including the instabilities in the paths and path asymmetry. Zhang *et al.* [13] studied the stationarity of Internet path properties. Though proposals such as [8], [12] assume a correlation in hop count changes and route changes, neither presents any quantitative proof of such correlation for any deployment. To the best of our knowledge, this paper is the first to quantify the correlations between different metrics to explore the optimization of monitoring infrastructure cost.

Note that we do not a priori claim that there exists any or no correlation between any two e2e network metrics. Our goal is to first explore and quantify such correlations and then propose schemes that can help in reducing the monitoring cost in large-scale distributed systems.

Our analysis is still preliminary, but the results are promising for optimizing large scale monitoring infrastructures. Note that the specific optimizations need to be closely coupled with the monitoring observations on a particular network. This paper provides a framework and methodology that can be applied broadly but the results of this paper can only be reliably used for end-to-end PlanetLab paths.

II. S³ DATA

For our analysis, we use data collected from the Scalable Sensing Service (S³) [11] running on PlanetLab since January 2006 (<http://networking.hpl.hp.com/s-cube>). PlanetLab currently consists of 840 nodes at 416 sites. The S³ system is run as a loosely coupled Service

Oriented Architecture (SOA) with a web-services interface for tools and collects different all-pair metrics: latency, available bandwidth, capacity bandwidth, and lossrate. For latency, we perform traceroutes from all nodes to approximately 20 “landmark” nodes distributed across the globe (US, Canada, Sweden, Brazil, Italy, Korea, Singapore), once about every 30 minutes, and use NetVigator [7] to infer the all-pair latency. We use Pathchirp [6] and Spruce [9] for available bandwidth, Pathrate [2] for capacity, and Tulip [3] for lossrate measurements. While many of these tools have been developed a while ago, deploying them in the large scale is still a challenge [11]. Significant engineering effort has been spent in making sure that the tools run reliably and with reasonable accuracy.

We utilize the traceroute measurements for studying path changes, end-to-end hops, and latency. The traceroute data set contains approximately 15 million data points for up to 14,000 source-destination pairs. We observe that the minimum, maximum and average number of hops in the dataset are 4, 30 and 16.27, respectively. Similarly, the minimum, maximum and average latency are 0.324 ms, 28026.931 ms and 149.383 ms, respectively. Note that because of a few outliers, we observe a very high maximum latency value in our measurements. The 99-percentile latency value is 671.21 ms. We use pathrate measurements along with the traceroute measurements for the analysis of correlation between capacity changes and path changes.

To obtain quick estimates of capacity, we run pathrate in the Quick Termination mode. We use results only when the coefficient of variation is between 0 and 1. We run these measurements in a loop at each source node measuring each destination in a round-robin fashion. It takes approximately a day on average to complete an entire cycle of measurements for all PlanetLab nodes.

III. NUMBER OF HOPS, LATENCY, AND ROUTE

In this section, we study the correlation between the changes in the number of hops, latency and routes on a path. Specifically, we are interested in determining if changes in the number of hops, referred as ‘numhops,’ and/or changes in the latency can be used as a reliable test to detect route changes. Route changes are important to detect, since they can affect other metrics such as capacity and available bandwidth and thus the application performance. Without the knowledge of the correlations between these network metrics, a monitoring infrastructure needs to continuously perform traceroutes to keep track of the current route of a path. However, if

correlations exist, a monitoring infrastructure can first perform inexpensive ping measurements to determine both latency and number of hops. Only upon detecting any change in numhops and/or latency, the monitoring system can perform relatively expensive traceroutes to determine the changed route. Note that the changes in the number of hops certainly indicate a route change, but latency changes may occur due to factors other than route changes, such as network load. On the other hand, a route change may not always cause a change in the number of hops. It is important to understand the timescales at which measurements must be done, so as to minimize the measurement cost while still maintaining an accurate description of the network conditions.

A. Methodology

We denote changes in numhops, latency, and route with H, L, and R, respectively. From the S³ dataset, we have a series of traceroute measurement samples for several e2e paths, thus providing all three metrics above. Below we describe how we determine changes in these different metrics along with the methodology for our Cost-Accuracy tradeoff analysis.

1) *Defining Route Changes:* For every sample of an e2e path, we analyze if the route is different compared with the previous sample (denoted as R=1) or the same (R=0). Note however that not all measurements can be 100% successful. We observed that many traceroute measurements suffer from different levels of imprecision. There were many instances where a traceroute measurement returns a “*” on some hops as the intermediate routers corresponding to those hops did not respond with ICMP TTL-Exceeded messages; either they silently dropped the packets that the source sent or the packets got lost in the route from the source to those routers or back. For the analysis of this paper, we deem a route to be changed in a sample only if there is at least a hop where we observe a different router IP address in comparison with the previous sample.

2) *Defining Hop Changes:* Since traceroutes return the number of hops, we compute for each path whether the numhops changed (denoted as H=1) or not (H=0) by comparing the numhops in a sample with the numhops in the previous sample. Since we can compute the number of hops only when the destination is reachable using traceroutes, we discard all measurements that do not reach the destination node.

3) *Defining Latency Changes:* We use traceroute’s observed round-trip times as latency. Since latency is a continuous value, we deem that the latency has changed

(L=1) only when the latency observed in a sample differs from the one observed in the previous sample by $p_l\%$ or more. Otherwise we assume that the latency has not changed (L=0). We use $p_l = 5$ in the analysis for this paper and we later describe the reason.

4) *Cost-Accuracy Tradeoff:* As mentioned earlier, our goal is to optimize the network monitoring system by understanding the dependence of metrics with higher measurement cost on those that have lower measurement cost. If such a strong dependence is observed, the idea is to frequently measure the lighter-weight (LW) metric and based on changes in the metric value, trigger the heavier-weight (HW) measurement. For this simple algorithm, the savings in the probing traffic, in comparison to a naïve strategy that only performs heavier-weight measurements, can be represented as: $(M_{HW} * F - M_{HW} * \text{changeFrequency}_{LW} - M_{LW} * F)$, where M_x is the measurement cost in bytes for metric x and F is the measurement frequency.

It must be noted that such savings in probing traffic often come at the expense of accuracy. This straw-man scheme will be inaccurate in the case when LW metric based detection has false negatives, i.e., the lighter weight metric did not change while the HW metric changes. We hence enhance the above strategy to also perform heavier-weight measurements but at low frequencies in conjunction with the measurements described above. We study the performance of this enhanced strategy (both accuracy and costs) and compare it with a naïve strategy that only performs HW measurements.

B. Analysis

We determined an appropriate p_l for defining latency changes as follows. For each sample, we computed the percentage change in latency in comparison with the previous sample. We then computed the median for the samples in R=0 case and R=1 case separately for each path. We observed that the averages are 1.6% and 10.6% across all paths for R=0 and R=1, respectively. This implies that the latency value changed significantly whenever there was a change in the route. Hence, we chose $p_l = 5$, a setting that falls between the above values.

1) *Correlation Between Numhops, Latency, and Route Changes:* Every successive pairwise samples of a path can be categorized to one of the eight combinations based on the H, L and R values. For each path, we compute the percentage of samples that fall into each of those combinations. In Table I, we present the average

TABLE I

NUMHOPS, LATENCY, AND ROUTE CHANGE CORRELATION: % SAMPLES FOR DIFFERENT CASES AVERAGED ACROSS ALL PATHS.

Case	Avg. %	Case	Avg. %
H=0, L=0, R=0	77.75	H=0, R=0	93.21
H=0, L=0, R=1	2.43	H=0, R=1	3.73
H=0, L=1, R=0	15.46	H=1, R=0	0
H=0, L=1, R=1	1.30	H=1, R=1	3.06
H=1, L=0, R=0	0.00	L=0, R=0	77.75
H=1, L=0, R=1	1.74	L=0, R=1	4.17
H=1, L=1, R=0	0.00	L=1, R=0	15.46
H=1, L=1, R=1	1.32	L=1, R=1	2.62

for these cases across all paths. As expected, numhops metric has very high correlation with the route changes. Only in 3.73% samples on average across paths, H and R values are different (H=0 and R=1 case). Remember that a change in numhops always implies a change in the route (count for cases with H=1 and R=0 is always zero), but the converse is not true. On the other hand, changes in the latency have a modest positive correlation with changes in the route for a path. If we consider both changes in latency and numhops in conjunction, only 2.43% samples fall into the case of H=0 and L=0, but R=1. This implies that a hybrid predictor that takes changes in both numhop and latency metrics to determine changes in route will perform better than a predictor based on any one of those metrics.

2) *Cost and Accuracy Tradeoff*: We now investigate the tradeoff between the measurement cost and the accuracy. For this analysis, we consider the series of our traceroute measurements as the baseline accuracy and measurement cost. We assume that each sample is obtained at an interval of approximately static period of T . Given any measurement strategy, we estimate the traceroute entries for all time intervals in the baseline. For the intervals, where we do not conduct a new traceroute, the route is assumed to be the same as the last measurement. We compute the inaccuracy of the strategy as the number of intervals where that strategy’s estimate is different from the baseline, divided by the total number of intervals considered.

We compare our algorithms that utilize the correlation between the metrics against a simple periodic probing approach. This simple approach, which we label as “plain,” conducts traceroutes at a predefined frequency. For example, conducting traceroutes at half the frequency of the baseline (i.e., once every two T periods, and hence a sampling factor of 2) will reduce the probing cost to half of the baseline. However, its inaccuracy could increase up to 0.5 as there might be route changes in all the instances when traceroutes were not performed.

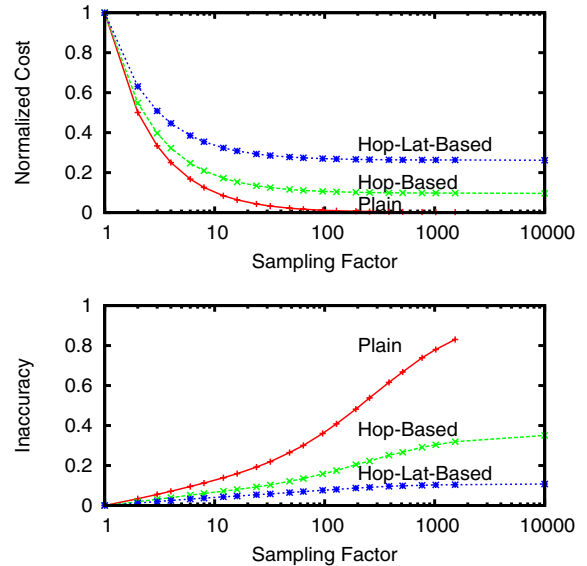


Fig. 2. Cost and accuracy tradeoff curves for different strategies.

Let us continue to use the example with the sampling factor of 2 to describe the other approaches. In the “hop based” strategy, we perform traceroutes every two T periods. In addition, we perform ping in the period when a traceroute is not performed. If we detect a change in the number of hops via ping, we then perform a traceroute. This approach therefore, has more cost than the plain approach at a given sampling factor, but is possibly more accurate.

Finally, the “hop-and-latency based” approach is similar to the “hop based” except that we use both the numhops and the latency. If we detect a change in either of those metrics, we perform a traceroute.

Figure 2 shows the mean of the inaccuracy and normalized cost for the above three mechanisms against the baseline, as we vary the sampling factor. The right-most point ($x=10,000$) corresponds to a sampling factor of infinity, i.e., where we do not perform any periodic scheduled traceroutes and rely on changes in numhops and hop-and-latency to invoke a traceroute.

We observe that the hop based and the hop-and-latency based approaches are good indicators for detecting route changes. With the infinity sampling factor, the hop-based approach reduces the cost to 0.08 fraction with about 33% inaccuracy. For the hop-and-latency approach, the cost is reduced to 0.25 fraction with only 12% inaccuracy. Note that choosing a good sampling factor in the plain approach can lead to better performance than the hop-based or hop-and-latency based approaches. For example at sampling factor 12, the plain approach has only 14% inaccuracy with a cost reduction to about 0.08.

However, finding such a good frequency to perform

TABLE II
LINK TYPES USED FOR LINK MAPPING.

LinkType	ISDN	DSL	T1	E1	T2	E2	LAN10	E3	T3	OC-1	LAN100	OC-9	OC-12
Capacity(Mbps)	.064	.128	1.54	2.05	6.31	8.44	10	34.36	44.73	51.84	100	466	622

measurements in the plain approach is a challenge. From the graphs, we can see that performing measurements at high frequency can be inefficient as the routes do not change very often. On the other hand, performing them at very low frequencies can lead to a rapid increase in the inaccuracies. Thus, using the hop based or hop-and-latency based strategy ensures that, in contrast to the plain-strategy, (i) the accuracy of the measurements does not deteriorate rapidly with reducing the frequency of measurements and (ii) the inaccuracy is bounded by a value much smaller than 1 irrespective of the sampling factor (0.33 in the case of hop and 0.12 in the case of hop-lat).

IV. ROUTE AND CAPACITY

We now explore the correlations in route changes and capacity changes on a path. Route changes can be monitored by traceroute which consumes much less bandwidth than the capacity monitoring tools such as Pathrate. If there exists a high correlation between the route changes and the capacity changes, relatively inexpensive traceroutes can be used to detect route changes. Only upon detection of a route change, expensive capacity measurements need to be performed.

A. Methodology

We denote the changes in route and changes in capacity with boolean variables R and C respectively. We have a series of pathrate and traceroute measurement samples for several e2e paths from the S³ dataset. Hence we have measurements for both metrics, although at different sample rates.

1) *Defining Capacity Changes*: Since pathrate outputs a continuous float value as a capacity measurement, we discretize it to detect the changes. We analyze data with two different definitions for the capacity changes: (i) Link-Mapping technique: We select a set of link types with known capacity values as presented in Table III-B2 and map the measured capacity value to the link that has the capacity closest to the measured value. (ii) Percentage Change: We assume that the capacity of a path changed when the measured capacity in the current sample is larger or smaller than $p_c\%$ of the previous value. We set $p_c = 10$ for the analysis in this section. The capacity measurement tools are prone to PlanetLab

TABLE III
ROUTE AND CAPACITY CHANGE CORRELATION: % OF SAMPLES FOR FOUR DIFFERENT CASES AVERAGED ACROSS ALL PATHS.

Case	Average Percentage	
	Link-Mapping	Percent-Change
R=0, C=0	53.76	42.15
R=1, C=1	8.96	15.63
R=1, C=0	25.07	18.42
R=0, C=1	12.21	23.80

imposed bandwidth restrictions and we are exploring better mechanisms for marking capacity changes.

2) *Sample Set*: As described in Section II, the sampling rate for capacity measurements in S³ deployment is about once a day for a path. Whereas the sampling rate for routes (using traceroutes) is approximately once every 15 minutes, but for a subset of paths. For this section, we consider only paths for which we have both route and capacity measurement data. For each capacity measurement sample on a path, we pick the traceroute measurement for that path that is performed closest in time to the time of the capacity measurement. The data we consider for this section thus has the same sampling rate as the capacity measurement data.

B. Analysis

1) *Path Changes and Capacity Changes Correlation*: For each sample for a given path, we have four cases depending on whether capacity and/or route changed in comparison with the previous sample. If there is a change in the capacity, we denote it as the C=1 case, and otherwise as the C=0 case. Similarly, R=1 denotes a change in the route and R=0 denotes otherwise. For each path, we compute the percentage of samples that fall into each of these cases. In Table III, we present the averages for these cases across all paths. In about 63% of the samples with Link-Mapping and 58% of the samples in Percent-Change cases, we observe that R and C take on the same value. This data implies a modest positive correlation between these two metrics. Similar to the cost-accuracy tradeoff analysis in the previous section, we further analyze the data to understand if this correlation can be helpful in reducing the monitoring cost while maintaining the accuracy.

2) *Cost and Accuracy Tradeoff*: We compare the monitoring cost and accuracy of the different capacity monitoring schemes. The baseline case consists of conducting pathrate measurements at an interval period

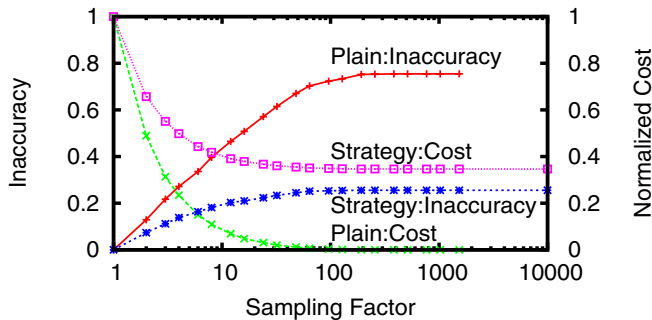


Fig. 3. Cost and accuracy tradeoff curves for different strategies when capacity changes as defined using Link-Mapping.

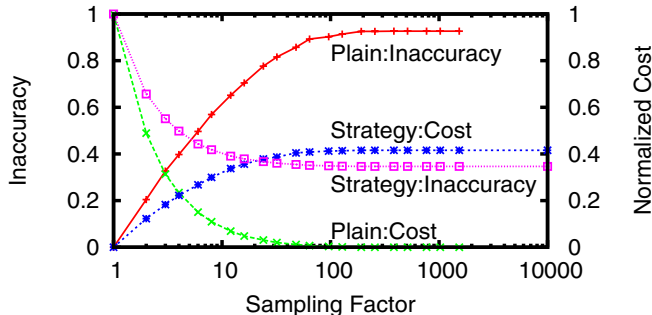


Fig. 4. Cost and accuracy tradeoff curves for different strategies when capacity changes as defined using Percent-Change.

equal to T . The “plain” method uses Pathrate to estimate capacity at a defined frequency (say for example, every two T periods). Using the example sampling factor of two again, the “strategy” method conducts Pathrate measurements every two T periods. It performs a traceroute measurement when a pathrate is not conducted. Only when a path/route change is detected, a pathrate measurement is then conducted.

Figures 3 and 4 show the mean of the inaccuracy and normalized cost for the “plain” and “strategy” mechanisms against the baseline, as we vary the sampling factor. Similar to Figure 2, the right-most point ($x=10,000$) correspond to a sampling factor of infinity.

We observe that the route change based approaches are good indicators for detecting capacity changes. With the infinity sampling factor, the route-based approach reduces the measurement cost to 0.35 fraction with about 25% and 42% inaccuracies as defined according to Link-Mapping and Percent-Change techniques, respectively. Note that in both graphs, though increasing sampling factor for the “plain” strategy reduces the measurement cost drastically, it also causes a rapid increase in the inaccuracy. On the other hand, by leveraging the correlation between the route changes and the capacity changes, “strategy” mechanism can choose any sampling factor for pathrate measurements while ensuring a modest inaccuracy at all factors.

V. CONCLUDING REMARKS

We consider the problem of optimizing network monitoring infrastructures based on the observed dependence in various metrics of interest. Since different metrics have varying probe overheads, not all metrics can be scalably measured at high frequencies. Thus, one optimization is to trigger the higher cost measurements on a need basis when the lower cost measurements detect a change in the end-to-end path. The S^3 system on PlanetLab has been monitoring network metrics on PlanetLab since January 2006. We present our analysis of this long range dataset studying the correlations between routes, number of hops, capacity, and available bandwidth. Based on this analysis, we present a framework and schemes to optimize the monitoring infrastructure. Our preliminary results with the PlanetLab dataset are promising and demonstrate that with the existing correlations, in some cases it is possible to reduce the monitoring cost to about 25% while maintaining the accuracy levels to about 88%.

REFERENCES

- [1] F. Dabek, R. Cox, F. Kaashoek, and R. Morris. Vivaldi: A Decentralized Network Coordinate System. In *SIGCOMM'04*.
- [2] C. Dovrolis, P. Ramanathan, and D. Moore. Packet-Dispersion Techniques and a Capacity-Estimation Methodology. *IEEE/ACM Transactions on Networking*, 12(6), Dec 2004.
- [3] R. Mahajan, N. Spring, D. Wetherall, and T. Anderson. User-level Internet Path Diagnosis. In *ACM SOSP 2003*.
- [4] T. S. E. Ng and H. Zhang. Predicting Internet Network Distance with Coordinates-Based Approaches. In *Proceedings of the IEEE INFOCOM 2002*, New York, NY, June 2002.
- [5] V. Paxson. End-to-end Routing Behavior in the Internet. In *Proc. of the ACM SIGCOMM'96*, Aug. 1996.
- [6] V. Ribeiro, R. Riedi, R. Baraniuk, J. Navratil, and L. Cottrell. pathChirp: Efficient Available Bandwidth Estimation for Network Paths. In *Proc. of the PAM 2003*, April 2003.
- [7] P. Sharma, Z. Xu, S. Banerjee, and S.-J. Lee. Estimating Network Proximity and Latency. *ACM Computer Communications Review*, 36(3):41–50, July 2006.
- [8] H. Song, L. Qiu, and Y. Zhang. NetQuest: A Flexible Framework for Large-Scale Network Measurement. In *Proc. of the ACM SIGMETRICS 2006*.
- [9] J. Strauss, D. Katabi, and F. Kaashoek. A Measurement Study of Available Bandwidth Estimation Tools. In *Proceedings of the ACM IMC 2003*, Miami, FL, October 2003.
- [10] B. Wong, A. Slivkins, and E. G. Sirer. Meridian: A Lightweight Network Location Service without Virtual Coordinates. In *Proceedings of the ACM SIGCOMM 2005*.
- [11] P. Yalagandula, P. Sharma, S. Banerjee, S.-J. Lee, and S. Basu. S^3 : A Scalable Sensing Service for Monitoring Large Networked Systems. In *SIGCOMM INM Workshop*, 2006.
- [12] M. Zhang, C. Zhang, V. Pai, L. Peterson, and R. Wang. PlanetSeer: Internet Path Failure Monitoring and Characterization in Wide-Area Services. In *Proc. of the Usenix OSDI 2004*.
- [13] Y. Zhang, V. Paxson, and S. Shenkar. The Stationarity of Internet Path Properties: Routing, Loss, and Throughput. Technical report, AT&T Center for Internet Research at ICSI, May 2000.