

UC San Diego

Technical Reports

Title

Pinpoint: Identifying Packet Loss Culprits Using Adaptive Sampling

Permalink

<https://escholarship.org/uc/item/4p74g215>

Authors

Ettinger, Evan

McFee, Brian

Freund, Yoav

Publication Date

2007-05-04

Peer reviewed

Pinpoint: Identifying Packet Loss Culprits Using Adaptive Sampling

Evan Ettinger, Brian McFee, Yoav Freund
University of California at San Diego
{ettinger, bmcfee, yfreund}@cs.ucsd.edu

ABSTRACT

Accurately estimating all link-level properties of a large network has proven to be very difficult. The measurements used for these estimates require significant collaboration from all endpoints on the network, significantly reducing their applicability for large scale Internet measurements.

We present a scalable approach using a small number of hosts without collaboration from existing routers and minimal collaboration between the hosts. Our approach is based on adaptive sampling. Initially, each host probes a set of receivers at a low frequency. When packet losses are detected, the sampling rate increases. By detecting correlations between time series and combining them with information about network connectivity, the host identifies a set of suspected lossy routers. Hosts then communicate with each other, combining evidence to identify routers with high packet loss. Our experiments show that using a relatively small set of hosts and receivers, we can gather sufficient evidence to identify a small number of routers that cause most of the packet loss in a geographically diverse sample of the Internet. We deployed our method for one month on 68 PlanetLab nodes. As a result of that deployment, we identified 128 routers of the $\approx 4,500$ accounting for 87% of the observed packet loss.

General Terms

Network measurement, Network tomography, Fault detection

1. INTRODUCTION

Localization of faults in the Internet is not only important for improving its structure, but could also provide useful information to overlay network providers and end users as well [26, 3]. Identifying and locating such faults can aid in providing an acceptable quality of service (QoS) to real time Internet applications such as VoIP, teleconferencing and online gaming.

In particular, VoIP has become increasingly popular in recent years. Studies have shown that high levels of latency, packet loss and jitter are the major contributors to poor QoS for VoIP [8, 10]. Our methods are extensible for other quantities of interest as well. Packet loss is

particularly problematic for VoIP because of its bursty nature [23, 10]. Typical packet loss bursts last around a second, during which which more than half of the packets are lost. This type of behavior is very difficult for error correcting codes to overcome, and the user will likely experience poor audio quality. In order to carry VOIP in the Internet at an acceptable QoS, the level of packet loss, especially bursty packet loss, has to be significantly reduced. Identification of the routers that drop many packets is an important step towards achieving this reduction.

Most current work on pinpointing network anomalies is based on simulations and on experiments using small networks where all nodes are under the control of the experimenter [25, 27, 16]. This ensures that the network substrate is uniform and known, that the number of end-points and measurements is sufficient to cover the network, and as a result that the conclusions derived from the measurements are reliable. The perceived problem with performing such experiments on the Internet is that the Internet is so large and diverse that it is impossible to draw trustworthy conclusions from measurements made at a small number of endpoints and without any special collaboration from the network [24, 14]. Consequently, it stands to reason that one would have to flood the Internet with probes in order to reliably identify lossy routers.

Our thesis is that although there is no a-priori reason to believe that lossy routers can be identified by statistical probing, as a matter of fact, *most* of the lossy routers on the Internet *can* be identified by light-weight active monitoring. The reason is that although packet loss is extremely bursty, with burst typically lasting a few seconds, the cause of bursts is a very small fraction of the routers. As a result, integrating information from probes over long periods of time (hours to weeks) reveals the identity of the culprit routers. To appreciate the significance of this finding, imagine an alternative situation in which packet loss on the Internet is highly distributed across many routers, each of which causes a packet loss burst very rarely. If this was the case, then detecting the culprits would be next to impossible.

By the time a burst is detected, the culprit router has stopped dropping packets, and will not drop a packet for several days, making it impossible to identify. Our experiments show that most packet loss culprits on the internet are repeat offenders, and thus can be reliably identified. In our experiments we used 68 PlanetLab [1] end-nodes to monitor packet loss on about 4500 routers. Our measurements implicate about 128 (3%) of these routers in 87% of the overall observed packet loss. Because the PlanetLab nodes are distributed around the world, it is likely that these ratios are typical. Because most links are well-behaved the majority of the time, we can use methods of adaptive sampling to identify the relatively few bottleneck routers causing loss at any given time. Moreover, our findings suggest that investment in reducing packet loss has high economic leverage. Reducing packet loss on a relatively *small* number of routers would make a significant difference in QoS for *many* end-users.

Most statistical network probing is based on *static* sampling. In other words, the set of probes is chosen before the experiment starts and is kept constant throughout the experiment. The problem with that approach is that in order to detect the relatively rare faults, probes have to be sent at a high rate. On the other hand, the vast majority of these probes are not informative, because they are sent along routes with less than 1% loss rate [31]. To overcome this problem we use *adaptive sampling*. The algorithm starts by probing at a low rate, and when packet loss is detected, the sampling rate is increased. In addition, if an end-node detects lossy routes whose packet loss events are correlated, it formulates a set of shared routers between the routes and for verification purposes starts to monitor these shared routers. In this way we can probe lossy network locations at a sufficient rate to draw reliable conclusions while not overloading the endpoints or the network with too many probes.

Our method escalates its probing and analysis in a series of modes which starts by probing all end-points uniformly and gradually becomes more focused on specific routers. The end result is a short list of routers with verified high packet loss rate.

We arrived at our particular sequence of measurement and analysis by trial and error. We have not strived to find an optimal probing method. Rather, our emphasis is on simplicity, low load on the probing host and on the network, and minimal interdependence between the hosts. We created a method which can be easily deployed and scales to large sets of nodes. We show that useful conclusions can be drawn using a deployment over just 68 PlanetLab hosts. Further increase in the deployment does not require increasing the load on any one host and will likely give a comprehensive analysis of the high-loss routers across the whole

Internet. The strength of our method is demonstrated by the repeated localizations to particular routers over time and across multiple vantage points. While it is possible, even likely, that some of the routers that we identify as lossy are in fact not dropping many packets, it is very unlikely that routers detected as lossy many times over and from several different end-points, are in fact not lossy. We therefore remain quite confident in the correctness of the list of 128 routers, and that by fixing these routers and/or increasing their capacity the rate of packet loss on the Internet can be significantly decreased.

The rest of the paper is organized as follows. In Section 2 we survey related work. In Section 3 we describe the Pinpoint methodology. In Section 4 we discuss the results of some experiments run on PlanetLab [1]. We discuss future directions for Pinpoint in Section 5 and conclude in Section 6.

2. RELATED WORK

There are several approaches to estimating link level loss in networks using only end-to-end measurements. One approach is that of “network tomography” overviewed in [9, 11]. Network tomography methods make a large set of end-to-end measurements, express the problem of estimating the router loss as an under-constrained system of equations, and then search for a solution that maximizes a regularization measure. Various techniques have been employed to solve these problems, the majority of which have been dominated by maximum likelihood, random sampling and expectation-maximization (EM) approaches. A significant recent contribution to this line of work has been made by Zhao et al. [32]. They define the concept of a “minimum identifiable link sequence” (MILS) which makes solving the system of equations tractable. Zhao et al. describe experiments on the Internet using PlanetLab nodes. However their probes span a very short period of time. In contrast, our experiments were conducted for over a month, while our adaptive approach minimizes the impact of the long experiment on the PlanetLab computers and on the networks.

Another approach to this problem is to explore how multicast measurements can be used to estimate link level properties [2, 6, 7, 12]. However, due to the lack of widespread multicast support, implementing multicast probing in the Internet is impractical. Duffield finds that stripes of unicast probes can estimate accurately link level loss characteristics in a variety of network conditions [13]. In particular, he investigates how large the stripes need to be in order to identify correlations between links, and finds that stripes of size 2–4 are more than adequate. Since the Internet supports unicast everywhere, we choose to use stripes of unicast probes as well.

Although active probing dominates the literature, there has been some work on localizing packet loss localization using passive collection of data. These approaches have the benefit that the measurements cannot affect the entities being measured. Padmanabhan et al. [21] collect TCP traces from msn.com servers and employ a variety of methods to infer link level loss percentages. They utilize random sampling, linear programming and Gibbs sampling. However, their results are derived from a small data set and are hard to validate since they happened in the past and are processed off-line. In contrast, our algorithm was developed for an online setting and we validate our results in real time.

Another area of work that has recently been studied is how probes should be sent in order to obtain accurate link level measurements. The PASTA principle (Poisson Arrivals See Time Averages) says that Poisson modulated probes used in a system that obeys a Poisson process will converge to the true value when averaged [30]. Recent work has shown that in the active probing paradigm, the PASTA principle can be well approximated by much simpler probes including randomized uniformly separated probes [4]. We utilize randomized uniformly separated probes.

A different correlation technique is employed by Rubenstein et al. [25]. Here they develop a comparison-based correlation test, rather than threshold-based, to determine whether two flows share a point of congestion. Their experiments do not attempt to localize this shared point, merely determine its existence. Kim et al. [16] also develop a correlation-based algorithm, but apply it to queuing delay rather than packet loss. We chose to instantiate our fault localization algorithm by thresholding our correlations, and in future work we plan to experiment with other statistical tests for detecting correlations.

An attempt to localize congestion along routes is presented by the Tulip project [19]. In this work, the authors present a method to identify a bottleneck along a *single* route. They present sequential and binary searching techniques along a single lossy route to localize a lossy router. Their methods involve sending a special sequence of round-trip probes, designed to determine whether a loss event happened on the forward or reverse path. Their method is highly sensitive to other simultaneous measurement, so it is not applicable for detecting shared loss across many routes.

Finally, the iPlane project is a large-scale measurement project to aggregate statistics and map the topology of the Internet. Similarly, Huffaker et al. [15] present another large-scale mapping system that combines IP and BGP routing information, and give some results connecting geographic distribution of nodes to observed latency. However, the primary goal of these projects is to construct an accurate map of the Internet, whereas

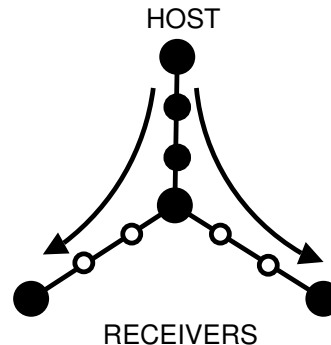


Figure 1: A single host sending simultaneous probes to two receiver nodes. Each dot represents a router on the traceroute from host to receiver.

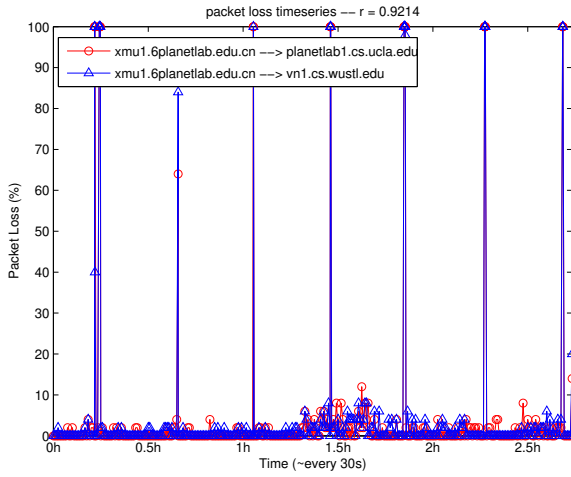
our goal uses small views of Internet connectivity as a means toward failure localization. At present, iPlane does not provide analysis of their data in terms of shared packet loss bottlenecks [18]. In this sense, the Pinpoint project is novel in its aim to develop a practical and lightweight method to diagnose packet loss failures in the Internet in real time.

3. THE PINPOINT ALGORITHM

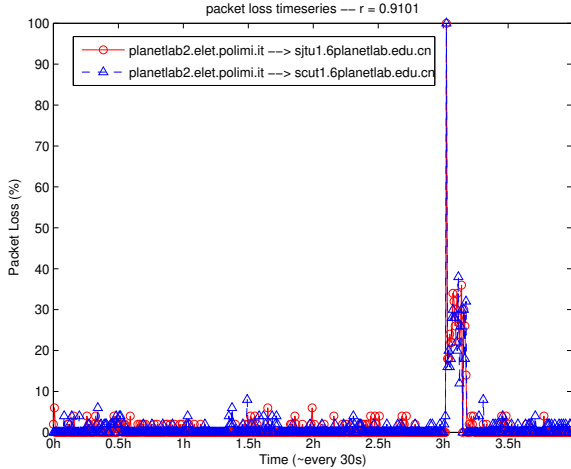
Imagine the simple case of Figure 1 where a single host is sending probes to two receiver nodes. In an ideal world we could simultaneously measure the one way loss to each of the receivers. If both receivers shared a single lossy router on both of their routes, then we would expect to see very similar loss patterns between the two. This is the core of the Pinpoint algorithm. The most common measure of similarity used in statistics is correlation. Let $X = (x_1, \dots, x_N), Y = (y_1, \dots, y_N)$ be the packet loss time series for each receiver for times $t \in [1, N]$, then we define the sample correlation between these two time series as

$$\text{Corr}(X, Y) = \frac{\frac{1}{N-1} \sum_t (x_t - \bar{x})(y_t - \bar{y})}{\sigma_x \sigma_y}.$$

Here \bar{x}, \bar{y} denote the means of X and Y , σ_x, σ_y are the standard deviations, and N is the number of measurements taken. We expect high positive correlation when both series are simultaneously above their respective means. In other words, when we see a spike in packet loss above the norm on one receiver, we see a corresponding spike at the same time on the other. If we can find several coinciding events, then we have strong evidence that these packet loss events share a common cause. Figure 2 shows two examples of highly correlated packet loss time series, one set with frequently repeating “blips” of packet loss, and the other with a single sustained loss event significantly above the aver-



(a)



(b)

Figure 2: Two sets of correlated packet loss time series from a single host to two distinct receivers with (a) repeated spikes ($r = 0.9214$), (b) a single long-duration packet loss event ($r = 0.9101$). Measurements were taken every 3 seconds.

age background loss.

When we find strong positive correlation we know that the lossy router is very likely to be in the shared portion of the tree. When we observe uncorrelated loss events, we know that the cause of the failures are more likely to be in the individual branches.

We now extend this idea to a multicast tree where a single host is sending probes to many receiver nodes as shown in Figure 3. (Multicast probing is not supported everywhere in the Internet, but tightly coupled unicast probes can be used as an approximation [13].) A pinpoint event occurs whenever a set of correlated receivers is identified. The pinpointed set is found by first identifying the root of the smallest subtree containing the

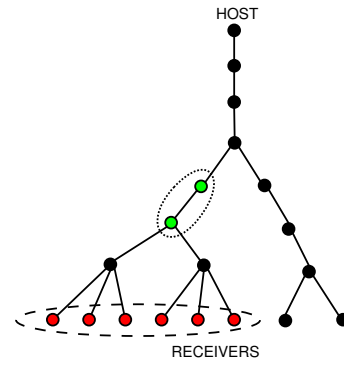


Figure 3: A single host sending simultaneous probes to many receiver nodes. A single correlated set is illustrated in red and the pinpointed set of routers in green.

entire correlated set. Then from this node, we traverse up the tree until we find the next highest branching point, allowing us to rule out any routers also common to the remainder of the (uncorrelated) receivers. All of the routers from the root of the subtree to the child of the next highest branching point define the pinpointed set. We expect to find the culprit lossy router within this pinpointed set.

Combining these ideas with adaptive sampling rates, we arrive at our methodology in five distinct steps:

1. **Background probing:** Each endpoint initially probes each of the other endpoints at a relatively low frequency. We denote the node sending the probes as the *host* and the nodes being measured as the *receiver set*.
2. **Accelerated probing:** If the background probing detects packet loss, it triggers a faster sampling rate for the offending receivers. The background probing rate is maintained for all other non-lossy receivers.
3. **Correlation detection:** Each pair of nodes in the receiver set exhibiting packet loss are tested to see if their packet loss events are correlated over time. A high correlation coefficient is evidence that the router causing the packet loss lies along the common path segment to the two end-nodes. By analyzing the correlated set and relating it to a tree of traceroutes to the receiver set, the method identifies a set of suspected routers. This set is called the *pinpoint set*.
4. **Verification:** By probing the routers in the pinpoint set, the method narrows down the cause of the loss to a single router. Statistical tests are used to compare the loss rate of routers along the pinpoint set and identify the hop on which there

is a significant increase in packet loss. This set of routers is called the *verified set*.

5. **Aggregation:** Each host creates its own list of lossy routers (pinpoint sets and verified sets). By combining these lists from all of the hosts, we can create a master list which identifies those routers that are identified as lossy many times and across many hosts.

In the Internet we face many problems that complicate the idealized model. First, obtaining one-way measurements of packet loss requires control over and synchronization between both of the endpoints. We do not assume control over the receiver set, because this is impractical for large scale measurement in the Internet. As an approximation, round trip measurements can be used in place of one-way measurements.

Second, round trip measurements may be dropped along unmeasurable reverse traceroute paths. Packets going on the forward path to the destination need not return along the same path. This problem could be overcome by synchronization between the host and receiver set, but again, this would not scale for very large receiver sets. However, it has been observed that the majority of forward and reverse traceroutes differ by at most one hop [22]. We simplify the problem by assuming symmetry along forward and reverse paths.

Third, the path between any two nodes in the Internet is not guaranteed to be constant for long periods of time. Routing updates happen often throughout the Internet, but it has been shown that most routes remain stable on the order of hours to days [22].

Finally, not all nodes in the Internet respond to measurement techniques. For example, some routers may choose to filter ICMP packets. Packet filtering affects all active probing techniques. Our statistical tests attempt to overcome some easily detected forms of filtering (i.e. by only counting positive increases in packet loss rate for verification), but otherwise do not differentiate filtering from failure.

Nevertheless, the aim of the pinpoint methodology is to be robust enough so that when a correlated set of receivers is found, the corresponding pinpoint set contains a lossy culprit router a significant portion of the time. We now present experiments that explore how robust our pinpoint methodology is against these problems.

4. INTERNET EXPERIMENTS

The following experiments were conducted on 68 geographically well distributed PlanetLab nodes between December 6, 2006 and January 11, 2007. Table 1 shows the geographic distribution of our set of nodes. Each node acts as an autonomous host, using the remaining nodes as its receiver set.

Continent	# of nodes
North America	22
Europe	24
Asia	21
Australia	1

Table 1: Geographic distribution of selected PlanetLab nodes.

4.1 Methodology

Figure 5 shows the core of our pinpoint module. An $|\mathcal{H}| \times N$ matrix is maintained to record the packet loss measurements of our tightly coupled unicast probes to each receiver. Each row in this matrix corresponds to a receiver being monitored, and contains the last N average packet loss measurements (we set $N = 1200$).

To reduce the network load incurred by uninformative observations, we partition the receiver set into two sets: a fast set \mathcal{F} with a high sampling rate, and a slow set \mathcal{S} with a lower sampling rate. All nodes start out in the slow set, where the host sends a ping of 50 packets every 21 seconds and records how many return. When the algorithm observes packet loss above 2% (i.e. at least one probe is lost), the offending receiver is moved into the fast set, where the sampling rate is increased to once every 3 seconds, again in bursts of 50 probes¹. Each probe consists of a 56-byte ICMP ping.

The measurement matrix contains entries for every 3 seconds, and infers a zero loss rate in between samples for the slow set. If a node in the fast set has 7 consecutive measurements of 0 (i.e. over 21 seconds has only 0 packet loss measurements), it is moved back to the slow set. The assumption behind these two speeds is that when an initial loss event is observed, it is more likely that more loss events can be observed soon thereafter. Because of the transient nature of packet loss events, we aim to improve correlation accuracy by obtaining more samples during the current loss event, and avoid wasteful sampling on routes exhibiting no packet loss.

We also compile traceroute trees by joining traceroutes from the host to each member of the receiver set in a bottom-up fashion. The traceroute tree is updated every 4 hours.

We continually update an $|\mathcal{H}| \times |\mathcal{H}|$ correlation matrix with each incoming set of measurements. A complete undirected graph is then constructed with the receiver set as the vertices and the pairwise correlation coefficients as the edge weights. All edges with a correlation coefficient below 0.90 are pruned, and we find connected components on the resulting graph. Our choice

¹In our implementation, the 3- and 21-second delays between samples for the fast and slow sets hold only in expectation. Each second, we generate a random number $r \sim \mathcal{U}[0, 1]$, and initiate probes for the fast and slow sets if $r \leq 1/3$ or $r \leq 1/21$ respectively.

```

PINPOINTMONITOR( $\mathcal{H}$ )
1  $\mathcal{F} \leftarrow \emptyset$ 
2  $\mathcal{S} \leftarrow \mathcal{H}$ 
3 repeat
4     With probability 1/3, probe  $\mathcal{F}$  and move nodes with 7 consecutive 0 measurements into  $\mathcal{S}$ 
5     With probability 1/21, probe  $\mathcal{S}$  and move nodes with non-zero measurements into  $\mathcal{F}$ 
6     Update correlation matrix
7     Find connected components in correlation graph
8      $\mathcal{R} \leftarrow$  pinpointed sets  $\cup$  children  $\cup$  parent
9      $\mathcal{S} \leftarrow \mathcal{S} \cup \mathcal{R}$ 
10    Delete from  $\mathcal{F}$  and  $\mathcal{S}$  any routers in  $\mathcal{H}$  monitored for at least 4 hours
        that are no longer in a pinpointed set
11    Sleep 1 second

```

Figure 4: Pseudo-code for the main components to the pinpoint experiment. \mathcal{H} here is our subset of 68 PlanetLab nodes, minus the host itself.

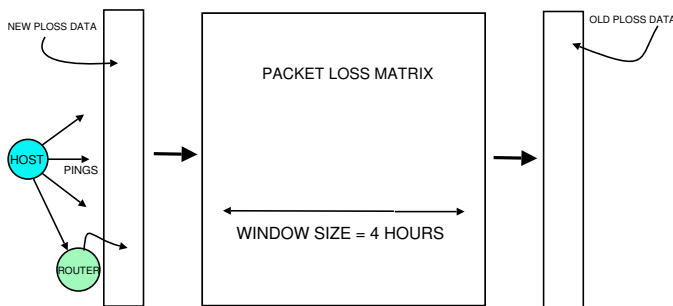


Figure 5: A single PlanetLab node periodically probes all other PlanetLab nodes, creating a matrix of packet loss measurements. The correlation matrix is updated after each shift of the matrix, and pinpoint events occur when highly correlated sets of rows are found.

of threshold here is consistent with the distribution of correlation coefficients for shared end-points presented by Kim et al. [16]. Any connected component of at least two nodes in the correlation graph defines a correlated set, and the corresponding pinpoint set is computed from the traceroute tree as described in Section 3. We then monitor these routers in the same fashion as described above for a minimum of one hour, but these routers are omitted from any subsequent correlated set calculations (only endpoints are considered). For validation purposes, we also include the routers one hop above and below the pinpoint set in the tree.

4.2 Results

The following sections discuss the results of our PlanetLab experiments.

4.2.1 Diagnosis Granularity

Diagnosis granularity refers to the size of the pinpointed sets. Smaller diagnosis granularity implies a more specific predicted location of the lossy router. Figure 6 shows a CDF of the diagnosis granularity of the identified pinpointed sets. Almost 80% are of size two or less, and almost 90% are of size four or less. This is comparable to the results of [32]. The granularity here is quite low for most pinpoint events, indicating that obtaining precise information in the Internet may not be as difficult as anticipated. We expect the granularity to decrease even further with the addition of more endpoints to the receiver set.

These results are promising for scaling Pinpoint as a monitoring service for the Internet. One potential challenge for scalability lies in choosing a receiver set that shares enough traceroute information for pinpoint events to diagnose to a fine granularity. The problem of selecting IPs that share favorable branching points in their traceroutes is particularly difficult. Our results indicate that this problem may not need much attention at all. Since most pinpoint sets are small even with our very geographically diverse set of nodes, it follows that we would achieve comparable diagnosis granularity for much larger receiver sets.

4.2.2 Success Rate of Pinpoint Events

Once a pinpointed set is identified, we begin monitoring each of the routers in the set, and record their average packet losses over the next 15 minutes. We deem a pinpoint event successful if the pinpointed set contains some router with significantly higher average packet loss than its parent. Note that the parent need not be within the pinpointed set.

Validating the success of a pinpoint event is inherently difficult. Because we have no control over any of the encountered routers, we cannot directly verify

whether a loss can be attributed to it at the time of a given pinpoint event. Therefore, actually observing the router dropping packets after it has been pinpointed becomes problematic because of the short, bursty nature of loss events. The best one can hope to do is to probe the pinpointed router while the problem persists. However, the probes need not follow the same route as in our current traceroute tree. Moreover, observed loss to the router need not have happened at that particular router, but may have happened anywhere along the probe’s route. These problems do not present a significant hurdle, given that most of our probes result in zero packet loss. With this in mind, we attempt to actively probe our pinpointed routers for verification purposes.

We use a paired hypothesis test where the null hypothesis is that the parent’s average is equal to the child’s, and the alternative is that the parent’s average is less than the child’s. We utilize the Wald statistic, which is defined as follows:

$$W = \frac{(\hat{p}_{\text{parent}} - \hat{p}_{\text{child}})\sqrt{n}}{\sqrt{\hat{p}_{\text{parent}}(1 - \hat{p}_{\text{parent}}) + \hat{p}_{\text{child}}(1 - \hat{p}_{\text{child}})}}$$

The null hypothesis is rejected when $|W| > z_{\alpha/2}$, that is, when $|W|$ lies in the upper $\alpha/2$ -quantile of the normal distribution $\mathcal{N}(0, 1)$. Here, n is the total number of measurement packets sent over a given time window, and $\hat{p}_{(\cdot)}$ denote the empirical average packet loss. We then use Bonferonni correction within each pinpoint event with a significance value $\alpha = 0.05^2$. We deem a pinpoint event *successful* whenever a parent and child pair exists within the pinpointed set where their loss rates pass the test above, and the child’s average packet loss is greater than that of its parent. If this is the case, we say that the lossy child router has been *verified*.

Figure 7 shows how often our pinpointed sets contain verified routers as a function of the size of the original correlated set. This graph shows the probability of success given that the correlated receiver set is of size at least x . As expected, our success rate increases with the number of nodes in the correlated set because we have more evidence towards a shared lossy router.

Figure 8 shows the average number of hops from a pinpointed router to its closest end point (either the host or any receiver node). Many routers are 7 or 8 hops away from the closest endpoint, implying that we are finding mostly internal routers, rather than “last mile” routers. This may be due to the geographic sparsity of our receiver set. However, our results show a surprising degree of packet loss at these internal routers, contrary to the common view that most performance bottlenecks occur in the last mile.

The fact that success generally increases as a function of the size of the correlated set, particularly for the

²See Wasserman for a more complete treatment of hypothesis testing [29].

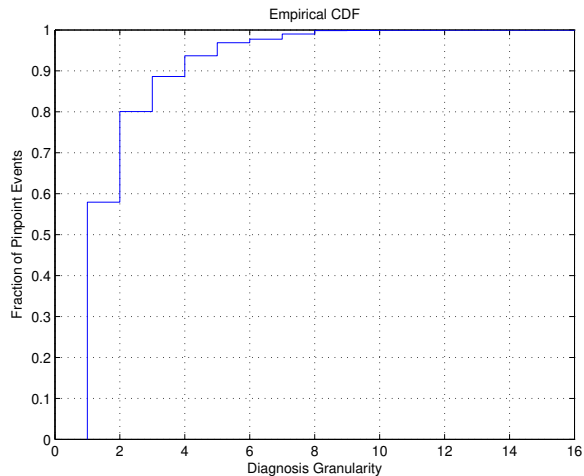


Figure 6: The diagnosis granularity of the pinpointed sets found. 80% of the sets are of size 2 or less.

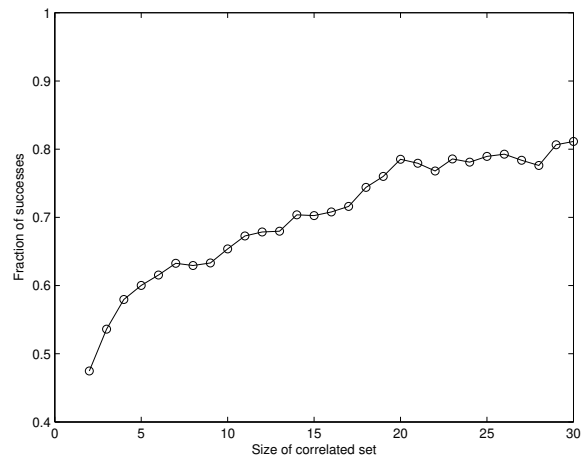


Figure 7: The fraction of successful pinpoint events given that the correlated set contained at least x nodes.

common case of small correlated sets, leads to an intuitive notion of confidence in our predictions. The more nodes in the correlated set, the more confident we are in the corresponding pinpoint set. In terms of scalability, it may prove useful to refine the algorithm and factor correlated set size into prediction confidence.

4.2.3 The Repetitive Nature of Pinpoint Events

Figure 9 shows a CDF of all the routers we encountered and how often they were in a pinpointed set over the entire experiment. Almost 85% of the routers were pinpointed to either 0 or 1 times and less than 10% of the routers were pinpointed to over 10 times. Very

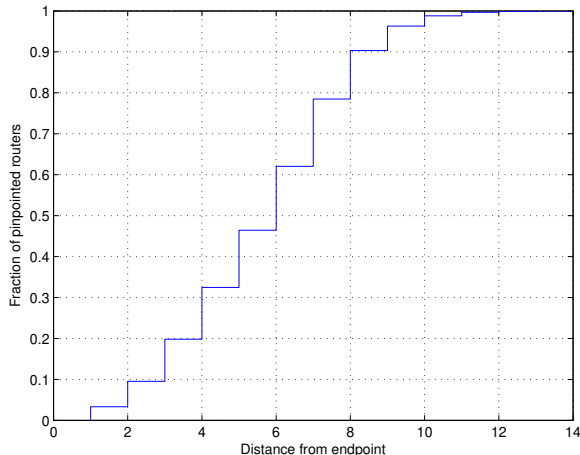


Figure 8: CDF of distance from a pinpointed router to either the host or its closest receiver.

few routers are pinpointed to multiple times, and the fraction of routers pinpointed to x times decays exponentially as x increases. This indicates a small fraction of routers that repeatedly drop packets over our month-long experiment period.

An alternative way of viewing this phenomenon is presented in Figure 10. For each router that we encountered during the experiment, we counted the number of pinpoint events that included it. We then calculated the fraction of all observed pinpoint events that include one or more of the top x most frequently pinpointed routers. Approximately 150 routers account for 80% of all pinpoint events. Only ≈ 800 of the observed routers were ever included in a pinpoint event, allowing our algorithm to focus resources on less than 20% of the set of potential culprits.

Similarly, Figure 11 shows the CDF of the fraction of verified routers, ordered by decreasing verification frequency. Here we maintain counts for each router indicating how many times it was in a verified set. Again, approximately 150 routers account for 80% of the total number of all verified routers observed. Of all routers encountered, only ≈ 500 were ever verified as lossy. Since a router must be pinpointed before it can be verified, this implies that about 60% of routers that belong to some pinpointed set were verified.

Figure 12 illustrates the distribution of the number of hosts that pinpoint a given router. Almost 80% of routers were pinpointed to from two or fewer hosts, but a surprisingly small number of routers were pinpointed from many different vantage points. Because multiple hosts with different views of the routing topology agree on this small set of routers, we expect these predictions to be especially strong.

If this result scales, then it follows that very few

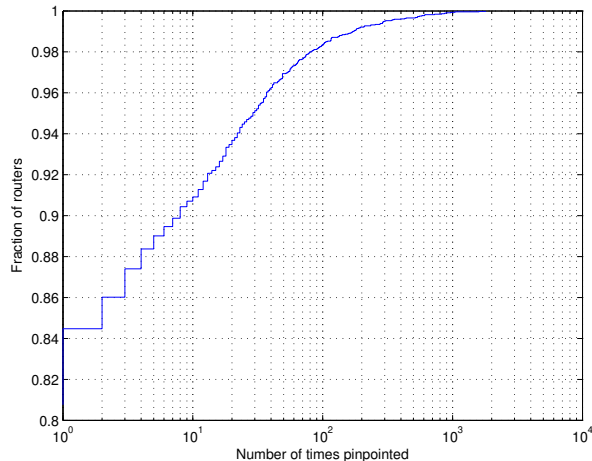


Figure 9: The fraction of routers pinpointed x or fewer times. A very small percentage of routers are pinpointed to more than ten times.

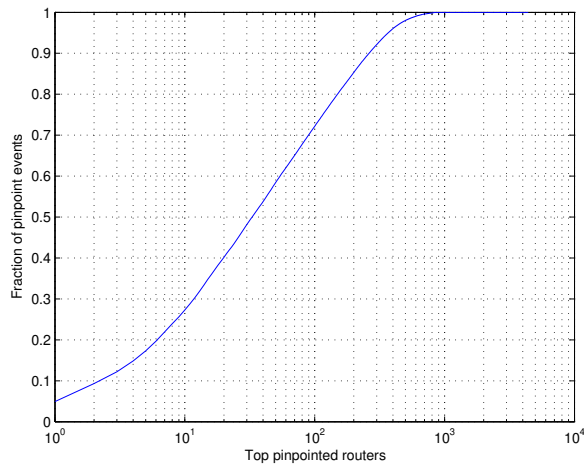


Figure 10: Fraction of all pinpoint events that include one or more of the top x most frequently pinpointed routers. Only ≈ 800 of the observed routers are ever included in a pinpointed set.

routers will be pinpointed many times. This leads to a natural way of finding such suspicious routers and verifying whether or not they are lossy. The Pinpoint methodology could be deployed from many vantage points across the Internet, and those routers that are pinpointed to the most frequently, and from the most distinct vantage points, are the ones that should be monitored further to investigate their loss properties.

4.2.4 Packet Loss Explanation

To evaluate our methodology, we quantify the amount of observed packet loss accounted for by the most fre-

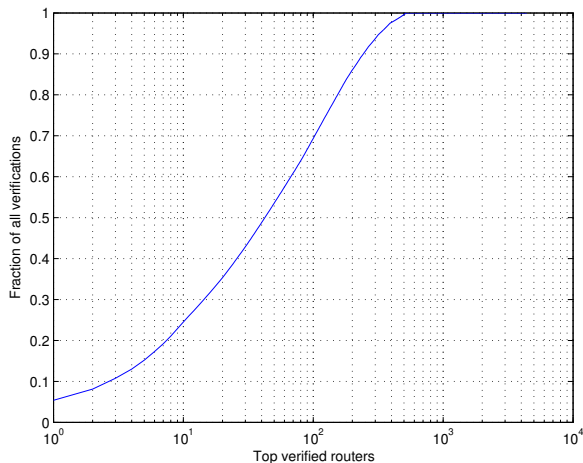


Figure 11: Fraction of passed hypothesis tests up to the top x most frequently verified routers. Only ≈ 500 routers are verified.

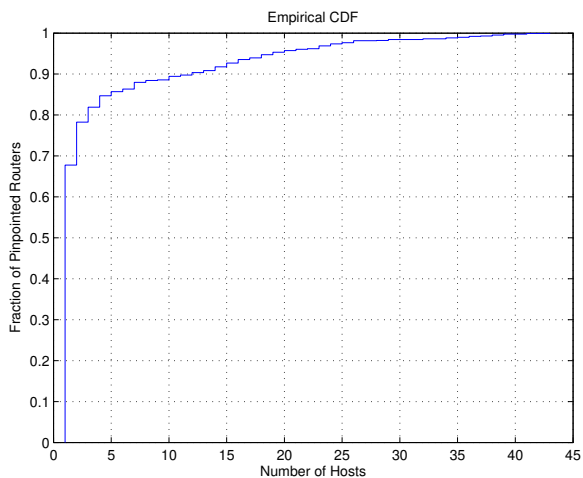


Figure 12: Fraction of routers pinpointed by at most x distinct hosts. A small percentage of routers are pinpointed from several different vantage points.

quently pinpointed and verified routers.

Figure 13 illustrates the distribution of verified lossy routers aggregated over all observed hosts and lossy receivers. For each end-to-end link exhibiting at least 3% packet loss for a given window³, we checked if the path contained one of the top x most frequently verified routers, and then counted the amount of observed packet loss. Of the 4413 routers encountered throughout the duration of the experiment, the top 128 (2.9%) appear on routes accounting for 87% of all observed

³The windows here are periodic snapshots of the host’s packet loss matrix as defined in Section 3.

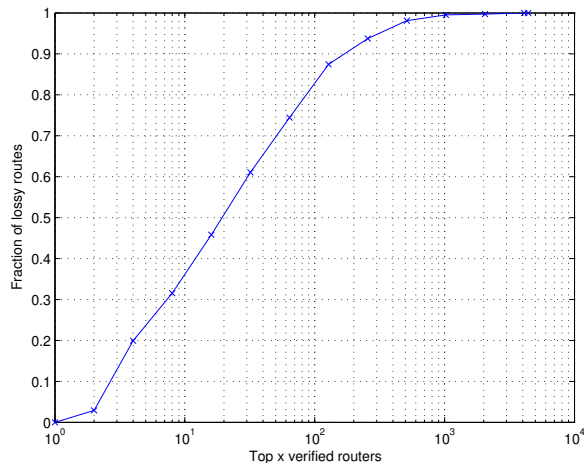


Figure 13: The fraction of packet loss observed at receivers exhibiting packet loss of at least 3% that also contain one of the top x most frequently verified routers along the path from the host. 87% of all packet loss occurs along routes containing a router in the top 128.

packet loss. We repeated this process for several thresholds, ranging from 0.1% to 10%, and the results did not change significantly. This confirms our thesis that end-point correlation over a relatively small receiver set can detect a significant portion of the observed packet loss. But do these routers really explain the observed loss?

Figure 14 illustrates the maximum amount of packet loss from any monitoring host to two of the most frequently verified routers over the entire duration of the experiment. In both cases, loss remains low for long periods of time, interrupted by bursts of extremely high packet loss. These graphs are typical of the top 15 verified routers listed in Table 2. The high packet loss observed at these routers indicates that the pinpoint methodology does succeed in locating causes of end-point loss.

These results confirm our intuition that real-time adaptation allows us to focus resources on the most problematic areas, while not sacrificing much information from the less-frequently sampled routers.

4.2.5 AS Investigation

During our experiment, we encountered routers from 154 different autonomous systems. Table 2 shows the AS information for the top 15 most frequently verified routers. Surprisingly, they belong to a very small set of autonomous systems.

Although we do not have direct access to the AS topology, we can draw inferences from the traceroute tree. In the following, we define a *gateway router* as a node in the traceroute tree adjacent to another node

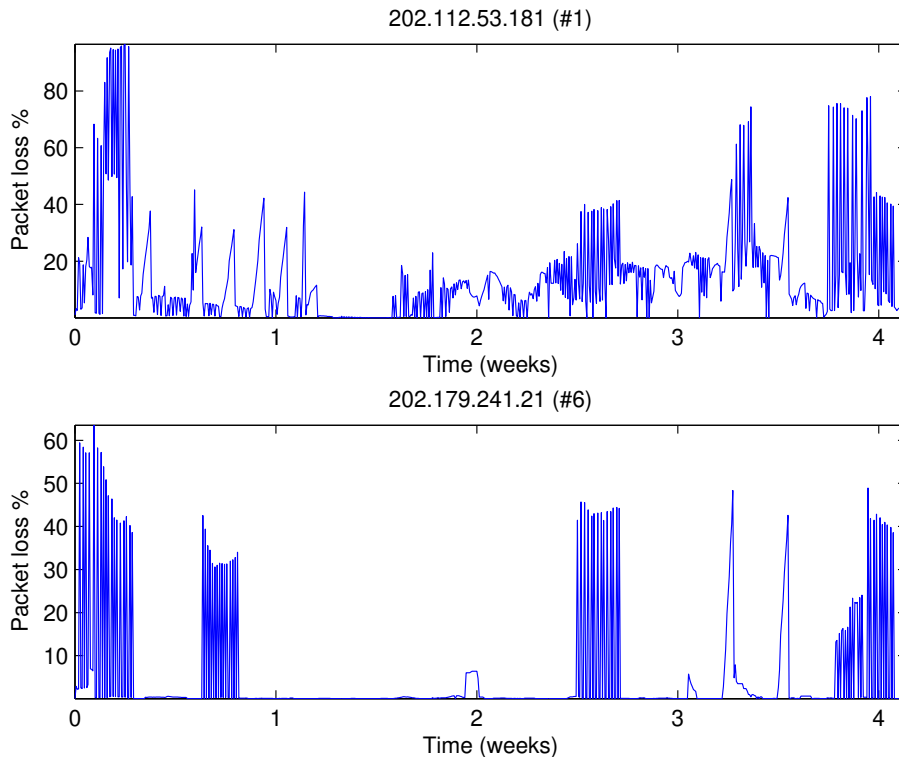


Figure 14: Packet loss observed throughout the experiment for the most frequently verified router (top) and a typical router in the top 15 (sixth most frequently verified, bottom). Each point corresponds to the maximum loss from any host monitoring that router in a given hour.

from a different AS, that is, their link crosses AS borders. The actual border routers lie somewhere in between two gateway routers. Ten of the top 15 verified routers lie on some AS boundary, including all of those that belong to Abilene-Internet2.

Figure 15 illustrates the relationship between gateway routers and verification frequency. For the top x frequently verified routers, we counted the fraction of gateway routers included. Strikingly, of the top ten most frequently verified routers, nine lie on links spanning autonomous system boundaries. The general trend is that the more often a router is pinpointed and verified, the more likely it is to be on the boundary between two autonomous systems. This may be indicative of traffic shaping or poor BGP policies at the provider, and lends support to previous claims that BGP problems can cause substantial service outages at the endpoints[3].

5. FUTURE WORK

In the future, the Pinpoint project plans on doing more extensive Internet measurements and developing a monitoring service that can produce a real-time list of the worst offending routers in the current Internet.

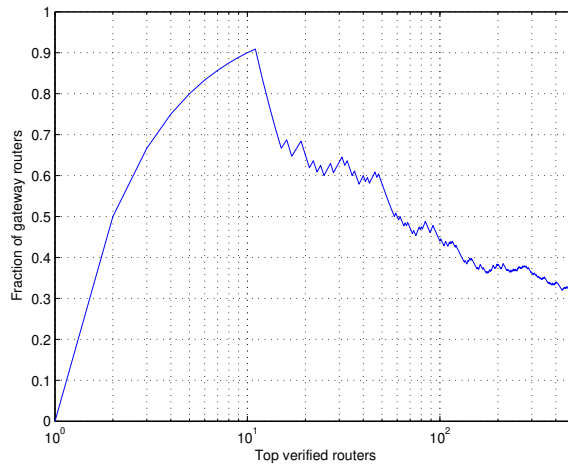


Figure 15: The fraction of the top x verified routers that lie on links spanning AS boundaries.

We now propose some extensions to the present work to improve the accuracy of the algorithm and further reduce the incurred network load due to active probing.

IP addr.	V Rank	P Rank	ASN	AS Name	Country	Gateway?
202.112.53.181	1	5	4538	ERX-CERNET-BKB China Education	CN	No
202.112.61.13	2	30	4538	ERX-CERNET-BKB China Education	CN	Yes
62.40.112.133	3	8	20965	GEANT The GEANT IP Service	NL	Yes
202.112.53.17	4	12	4538	ERX-CERNET-BKB China Education	CN	Yes
202.112.61.9	5	22	4538	ERX-CERNET-BKB China Education	CN	Yes
202.179.241.21	6	24	24489	TEIN2-NORTH-AP Trans-Eurasia I	CN	Yes
198.32.8.76	7	11	11537	ABILENE - Internet2	US	Yes
62.40.124.121	8	55	20965	GEANT The GEANT IP Service	NL	Yes
64.57.28.7	9	16	11537	ABILENE - Internet2	US	Yes
202.179.241.26	10	23	24489	TEIN2-NORTH-AP Trans-Eurasia I	CN	Yes
198.32.11.102	11	40	11537	ABILENE - Internet2	US	Yes
202.112.53.18	12	1	4538	ERX-CERNET-BKB China Education	CN	No
62.40.112.57	13	25	20965	GEANT The GEANT IP Service	NL	No
62.40.112.134	14	19	20965	GEANT The GEANT IP Service	NL	No
202.127.216.22	15	67	4538	ERX-CERNET-BKB China Education	CN	No

Table 2: Top 15 routers found. We show what rank in terms of number of verified sets the router was involved in (V Rank) and also in terms of the total number of times pinpointed to (P Rank). We also show what AS number the IP belongs to (ASN), the name of that AS, the country that AS is in, and whether the router borders two ASes on observed routes. Visit our website for a more complete list.

5.1 Robust Similarity

One of the major underlying problems with using correlation is its relative sensitivity to noise. To correlate at a level of 0.9 or higher is a stringent condition that is easily breakable. Imagine the case where two endpoints observe loss as a result of two independent routers, where one router lies on the shared path segment to the endpoints and the other lies below the branching point. The two endpoints will in some sense have similar time series, but correlation alone is not robust enough to capture this form of similarity. We would then like to be able to separate the contributions of the observed time series due to each router in order to derive a new notion of similarity.

There have been several techniques proposed in the statistics and machine learning literature designed to address this type of overlapping signals problem, including Independent Components Analysis (ICA) and Non-negative Matrix Factorization (NMF) [5, 20]. Because of the positive additive nature of the composition of packet loss events, we briefly experimented with NMF-based similarity metrics for our time series data. We were unable to find a sufficiently robust tuning of the parameters to employ in the online setting, but our experiments were by no means exhaustive. More research in this direction would help to generalize our model to handle simultaneous router failures along non-overlapping path segments. On the other hand, because of the fact that most routers tend to not drop any packets most of the time, this scenario where a route has multiple simultaneous lossy routers may be rare enough

to be neglected.

A larger looming concern is the effect of false positive correlations in our work. Because of the short bursty nature of packet loss and the discrete nature of our measurement abilities we are typically only able to catch a select few high loss bursts in a given window to correlate on. Utilizing statistical tests for correlation may help separate the false positives from the real correlations.

5.2 Increased Adaptivity

The pinpoint methodology proposed is entirely geared for *online* discovery of lossy shared routers. Something that we initially examined in the experiments discussed above is the adaptive sampling component to the online algorithm. The present work is adaptive in two distinct ways. First, the sampling rate of a node being monitored varies as function of the observed packet loss. Secondly, the set of nodes being observed is highly variable, and depends on long-term interactions between members of the receiver set.

In our experiments, we restricted our initial receiver set to 68 nodes, and used a simple threshold rule to decide when to increase the sampling rate. However, when the receiver set becomes much larger, we run the risk of saturating the host’s connection with measurement packets. Therefore, we would like to be able to maximize the information gained by our measurements while simultaneously minimizing the contributed load on the network. Intuitively, a node having low-variance packet loss should be easier to predict than one with high variance. The less predictable the time series is, the more

information we gain from each sample. Therefore, the sampling rate of a node should also depend on the variance of its time series, rather than a simple threshold of its latest observation. This intuition leads to the addition of a weighting for each receiver that depends directly on the variance of its time series, and could be updated online with the *exponential weights* rule, which has seen significant success in the field of online machine learning [17, 28]. It should be noted that our current threshold rule necessarily entails a dependence on variance. In addition to the low-variance, high-loss links, every high-variance link will be monitored because it must have crossed the loss threshold to achieve high variance. The present algorithm is therefore not optimized for network load, but provides a superset of the observations, is slightly more intuitive, and has fewer parameters to tune.

Similarly, we could maintain a separate weight that tells us how stable a particular node’s traceroute is over time. If it is very stable, then we shouldn’t need to calculate it often, but if it is continually making large changes then we should update its traceroute more often. A simple set difference or edit distance metric seems logical to employ here, and could reduce the false positive rate of the pinpoint algorithm by improving the accuracy of the tree.

6. CONCLUSIONS

We have presented an adaptive measurement paradigm for pinpointing lossy routers in the Internet. Our experiments focused mainly on packet loss, but the core ideas put forward could easily be extended to other quantities of interest, e.g. latency. The experiments on PlanetLab show that the methodology does in fact work quite well in the real Internet despite the many real world problems that work against the model. We found that relatively few routers are responsible for the majority of packet loss observed by end-users.

It is worth emphasizing the possible implications of our findings on the economics of QoS on the Internet. In the current state of affairs, there is no incentive for ISPs and backbone providers to improve QoS since doing so would not directly impact their prices and their revenues. On the other hand, there is a growing demand for QoS from customers that want to use their Internet connections for VoIP, online games and other real-time interactions. If our conclusions are correct and if they scale up to the Internet as a whole, then a small number of bottleneck routers have a large impact on packet loss that would be experienced by end-users and thus represents a significant economic opportunity. If this small number of routers (or the routing policies which control them) are upgraded to match their load so that they would not drop packets, a large number of people would get the benefits of higher QoS connections,

which is likely to make them prefer an upgraded network over a network that drops packets. If the number of bottleneck routers is small and the number of Internet users that benefit from an upgrade is large, then there is a large economic incentive for upgrading the routers. We therefore believe that our work represents a significant step towards making Internet QoS more transparent and setting the path towards higher QoS on the Internet. This higher QoS would not rely on “hard” guarantees, but rather on actual performance. This is similar to quality of service given to the consumer by airlines, which relies on the availability of statistics to the public on cancelled and delayed flights, on flight service quality, etc. rather than on any written guarantees.

7. REFERENCES

- [1] Planetlab. <http://www.planet-lab.org>.
- [2] A. Adams, B. Tian, T. Friedman, J. Horowitz, D. Towsley, R. Caceres, N. Duffield, F. Presti, S. Moon, and V. Paxson. The use of end-to-end multicast measurements for characterizing internal network behavior. *Communications Magazine, IEEE*, 38(5):152–159, May 2000.
- [3] D. Andersen, H. Balakrishnan, M. Kaashoek, and R. Morris. The case for resilient overlay networks. In *Hot Topics in Operating Systems, 2001. Proceedings of the Eight Workshop on*, pages 152–157, 2001.
- [4] F. Baccelli, S. Machiraju, D. Veitch, and J. Bolot. The role of pasta in network measurement. In *SIGCOMM ’06: Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 231–242, New York, NY, USA, 2006. ACM Press.
- [5] A. J. Bell and T. J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7:1129–1159, 1995.
- [6] T. Bu, N. Duffield, F. L. Presti, and D. Towsley. Network tomography on general topologies. *SIGMETRICS Perform. Eval. Rev.*, 30(1):21–30, 2002.
- [7] R. Caceres, N. Duffield, J. Horowitz, and D. Towsley. Multicast-based inference of network-internal loss characteristics. *IEEE Transactions on Information Theory*, 45(7):2462–2480, Nov 1999.
- [8] P. Calyam, M. Sridharan, W. Mandrawa, and P. Schopis. Performance measurement and analysis of h.323 traffic. *Passive and Active Measurement Workshop*, 2004.
- [9] R. Castro, M. Coates, G. Liang, R. Nowak, and B. Yu. Network tomography: Recent

- developments. *Statistical Science*, 19:499–517, 2004.
- [10] K.-T. Chen, C.-Y. Huang, P. Huang, and C.-L. Lei. Quantifying skype user satisfaction. In *SIGCOMM '06: Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 399–410, New York, NY, USA, 2006. ACM Press.
- [11] M. Coates, A. Hero, R. Nowak, and B. Yu. Internet tomography. *Signal Processing Magazine, IEEE*, 19(3):47–65, May 2002.
- [12] N. Duffield, J. Horowitz, W. Wei, and T. Freidman. Multicast-based loss inference with missing data. *IEEE Journal on Selected Areas in Communications*, 20(4):700–713, May 2002.
- [13] N. Duffield, F. L. Presti, V. Paxson, and D. Towsley. Inferring link-loss using striped unicast probes. *IEEE INFOCOM*, 2:915–923, 2001.
- [14] J. D. Horton and A. López-Ortiz. On the number of distributed measurement points for network tomography. In *IMC '03: Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement*, pages 204–209, New York, NY, USA, 2003. ACM Press.
- [15] B. Huffak, D. Plummer, D. Moore, , and k. Claffy. Topology discovery by active probing. In *SAINT-W '02: Proceedings of the 2002 Symposium on Applications and the Internet (SAINT) Workshops*, page 90, Washington, DC, USA, 2002. IEEE Computer Society.
- [16] M. S. Kim, T. Kim, Y. Shin, S. S. Lam, and E. J. Powers. A wavelet-based approach to detect shared congestion. In *SIGCOMM '04: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 293–306, New York, NY, USA, 2004. ACM Press.
- [17] N. Littlestone and M. K. Warmuth. The weighted majority algorithm. In *IEEE Symposium on Foundations of Computer Science*, pages 256–261, 1989.
- [18] H. Madhyastha, T. Isdal, M. Piatek, C. Dixon, T. Anderson, A. Krishnamurthy, and A. Venkataramani. iplane: An information plane for distributed services. *OSDI*, 2006.
- [19] R. Mahajan, N. Spring, D. Wetherall, and T. Anderson. User-level internet path diagnosis. In *SOSP '03: Proceedings of the nineteenth ACM symposium on Operating systems principles*, pages 106–119, New York, NY, USA, 2003. ACM Press.
- [20] P. Paatero and U. Tapper. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5:111–126, 1994.
- [21] V. Padmanabhan, L. Qiu, and H. Wang. Server-based inference of internet link lossiness. *IEEE INFOCOM*, 1:145–155, 2003.
- [22] V. Paxson. End-to-end routing behavior in the internet. *IEEE/ACM Trans. Netw.*, 5(5):601–615, 1997.
- [23] V. Paxson. End-to-end internet packet dynamics. *IEEE/ACM Trans. Netw.*, 7(3):277–292, 1999.
- [24] V. Paxson and S. Floyd. Why we don't know how to simulate the internet. In *WSC '97: Proceedings of the 29th conference on Winter simulation*, pages 1037–1044, 1997.
- [25] D. Rubenstein, J. Kurose, and D. Towsley. Detecting shared congestion of flows via end-to-end measurement. In *SIGMETRICS '00: Proceedings of the 2000 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, pages 145–155, New York, NY, USA, 2000. ACM Press.
- [26] S. Savage, T. Anderson, A. Aggarwal, D. Becker, N. Cardwell, A. Collins, E. Hoffman, J. Snell, A. Vahdat, G. Voelker, and J. Zahorjan. Detour: Informed internet routing and transport. *IEEE Micro*, 19(1):50–59, 1999.
- [27] J. Sommers, P. Barford, N. Duffield, and A. Ron. Improving accuracy in end-to-end packet loss measurement. In *SIGCOMM '05: Proceedings of the 2005 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 157–168, New York, NY, USA, 2005. ACM Press.
- [28] V. G. Vovk. Aggregating strategies. In *COLT '90: Proceedings of the third annual workshop on Computational learning theory*, pages 371–386, San Francisco, CA, USA, 1990. Morgan Kaufmann Publishers Inc.
- [29] L. Wasserman. *All of Statistics: A concise course in statistical inference*. Springer Texts in Statistics, New York, NY, 2004.
- [30] R. Wolff. Poisson arrivals see time averages. *Operations Research*, 1982.
- [31] Y. Zhang and N. Duffield. On the constancy of internet path properties. In *IMW '01: Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement*, pages 197–211, New York, NY, USA, 2001. ACM Press.
- [32] Y. Zhao, Y. Chen, and D. Bindel. Towards unbiased end-to-end network diagnosis. In *SIGCOMM '06: Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 219–230, New York, NY, USA, 2006. ACM Press.