Characteristics of Network Delays in Wide Area File Transfers

Paul Barford, David Donoho, Ana Georgina Flesia and Vinod Yegneswaran

Abstract— **In this paper we present an analysis of over 236,000 file transfers between 10 widely distributed Internet hosts. The goal of this work is to broaden the understanding of how network path and congestion properties contribute to delays in TCP file transfers. The first part of our analysis investigates how end-to-end path properties (eg. physical distance, router hops, autonomous system hops, or bottleneck bandwidth) relate to file transfer latency and variability. We evaluate end-to-end path properties as a predictor of file transfer latency, and employ dimensionality-reducing techniques to identify clustering in path space. We find that expected transfer latency can be effectively predicted by a number of path properties and that the relationship between paths and latency is strongly linear with some intense outliers. The second part of our analysis employs critical path techniques to break down the network component of file transfer latency into three categories: propagation, queuing and loss. We compare the contribution of each of these components to delays along particular paths, and their effect on variability of total delay. We find that propagation delay is the dominant aspect of expected total delay for most paths and that queuing and loss are substantial effects typically for a minority of paths. On these paths, queuing contributes most significantly to periodicity in total delays while loss contributes most significantly to variability in total delay.**

I. Introduction

Many aspects of current Internet infrastructure and protocols have been developed or enhanced as a direct result of careful analysis of network measurement data. Examples include application-level protocols such as HTTP [1], [2], transport protocols such as TCP [3], [4] and TFRC [5], and distributed caching mechanisms such as [6], [7]. While the body of measurement-based work upon which these and other developments have been founded is significant, many of the complexities of Internet interactions (which if understood could lead to future improvements) remain unstudied.

The general objective of our study is to broaden understanding of the factors that contribute to delays in wide area TCP file transfers, using a large data set collected in a distributed Internet measurement infrastructure. The data set consists of measurements of over 236,000 file transfers taken over the period of 45 days along 90 distinct Internet paths in the WAWM infrastructure [8]. Our analysis of this data is focused on understanding the details of end–to–end delays caused by *the network itself*. In particular we address the question, "how does the network component of delay[1] in wide area TCP transfers break down into subcomponents of propagation, queuing and loss?" Consideration of this question led us to also consider the influence of the properties of the network paths between end hosts and how network delays break down into the different subcomponents for different paths. To that end, we also collected over 79,000 `traceroute` [9] measurements taken along the same set of paths during the same time as the data transfer measurements.

The challenges in this work arose in two principal areas in addition to the typical difficulties associated with wide area measurement-based study [10]. The first was the task of breaking down data transfer delays into subcomponents. We addressed this by developing a robust, kernel-level implementation of critical path analysis of TCP transfers [11]. Our kernel-level implementation has two important benefits: it enables the calculation of subcomponents of data transfer delays *in real time* and expands the number of sub-components of delays from the six described in [11] to nine. The details of the new subcomponents are described in Section III.

The combination of path property measurements and detailed data transfer delay measurements over 90 paths for 45 days led to a very large and highly dimensional data set. Our second challenge was organizing and reducing this data set to extract key results. We addressed this initially through visualizations of the time series of data transfer delays for each path. This enabled us to not only focus on the important behaviors that make up the key results of the paper but also on out-lier data points which represented pathological behaviors and required closer attention. Next, we systematically employed a number of multivariate analysis techniques to understand details of behavior along different paths and to identify path proper-

P. Barford and V. Yegneswaran are members of the Computer Science Department at the University of Wisconsin, Madison. E-mail: pb,vinod@cs.wisc.edu. D. Donoho and G. Flesia are members of the Statistics Department at Stanford University. E-mail: donoho,flesia@stat.stanford.edu.

[1] End hosts would be the other primary component of delay.

ties which are predictive of transfer delay sub-components.

Our analysis indicates that most paths typically operate along what we define as an *efficient frontier*. This is the data transfer rate which is principally limited by round trip times (speed of light delay plus aggregate switching time at routers) and not by network congestion effects. Operation along the efficient frontier is more common as the size of transfers decreases. There are however a number of paths whose normal operating behavior is quite far from the efficient frontier. The common feature of these paths is that they all contain commodity Internet links (*ie.* links maintained by commercial Internet Service Providers). Variability for these paths is dominated by packet loss and periodicity is dominated by queuing. We find that for large files, *both* queuing and loss are significant effects on overall transfer times. We also evaluate the predictive capability of path properties versus delay and each sub-component of delay. We find that appropriate linear combinations of path properties can be very good predictors of each of these features.

Our findings have implications for analytic models of TCP throughput such as [12], [13], [14]. These models provide a simple mechanism for estimating expected throughput based on RTT and average packet loss rate as inputs related to network conditions. We acknowledge the merits of simplicity in these models and the insights on TCP behavior that they provide. However, we show that for our data, these models can be quite poor at predicting throughput. We attribute this to two factors: variability in RTT caused by queuing delays and more complex loss processes than are assumed in the model. These results suggest additional factors in throughput models to account for a broader range of network conditions.

Our results also have implications in the network operations domain. The first and most obvious is for network managers to focus traffic engineering efforts on paths in their networks that do not operate on the efficient frontier. A second example are efforts to estimate distances between nodes as a means for directing clients to appropriate mirror servers such as [15]. Our findings indicate that static path properties such as physical distance are often as accurate as more dynamic features like RTT in terms of predicting expected throughput for many paths. However, we find that both static *and* dynamic features can be very inaccurate throughput predictors for a minority of paths. Another related example is in the area of routing overlay networks [16]. Our results indicate that it may be more appropriate to select an overlay path between client and server that combines paths on the efficient frontier instead of a more direct, shorter path that does not operate on the efficient frontier.

II. Related Work

There are a growing number of measurement-based studies of wide area network behavior that have shed considerable light on factors including performance, stability and growth. Examples of these that relate to our work include studies that investigate basic characteristics of packet dynamics [17], [18], [19], [20], [21], and studies that assess Internet routing and path characteristics [22], [23], [24]. Of these, the work that is perhaps most similar to ours is that of Zhang *et al.* [21]. In that study, the authors use a large measurement data set to define characteristics of network "path constancy" related to packet loss, packet delay and TCP throughput. We do not specifically treat issues of path constancy in our work. However, results of both their study and ours can be applied to the problems of TCP performance modeling and distance estimation used in systems such as [15].

Estimations of distance have been evaluated extensively in caching literature. Various techniques for placing content near clients to improve performance have been considered including geographical distance [25], topological distance [26] and latency [27]. End-to-end distance metrics are a topic of on-going study. An example is recent work by Huffaker in [28] who finds that geographical distance is a reasonable indicator of round trip time within the US.

Another important aspect of measurement-based study is to attempt to establish *invariant* behavioral characteristics [29] . Perhaps the most successful work in this regard has been the establishment of self-similarity as a fundamental characteristic of network packet traffic [30], [31], [32]. We do not attempt to establish or advocate invariant properties for the characteristics that we treat in this paper.

The use of path properties to understand network state was investigated by Allman and Paxson in [33]. That work considers path state which can be inferred from TCP traces as a means for refining RTO and bandwidth estimates. A variety of tools have also been developed to assess path properties such as bandwidth and loss characteristics: examples include [34] and [35]. Our work differs from these in our investigation of more static notions of path properties (ie. route characteristics) as predictors of TCP data transfer latency.

III. Data

A. Measurement Environment

Our data was collected in the WAWM infrastructure [8]. Currently this environment comprises 10 dedicated PCs, with 90 end to end paths, spanning 31 distinct Autonomous Systems (AS) across 3 continents. The systems

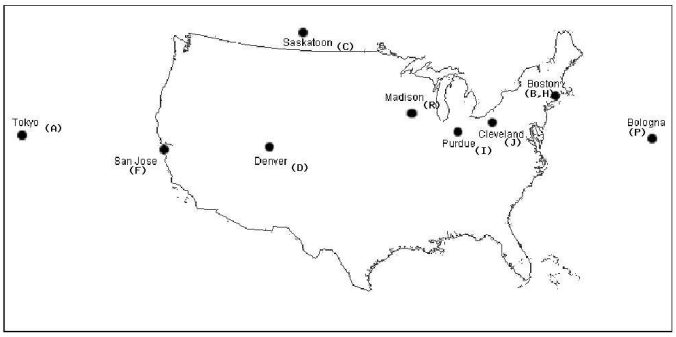

Fig. 1. Deployment of WAWM nodes (2 systems in Boston). Included are abbreviations used to identify hosts.

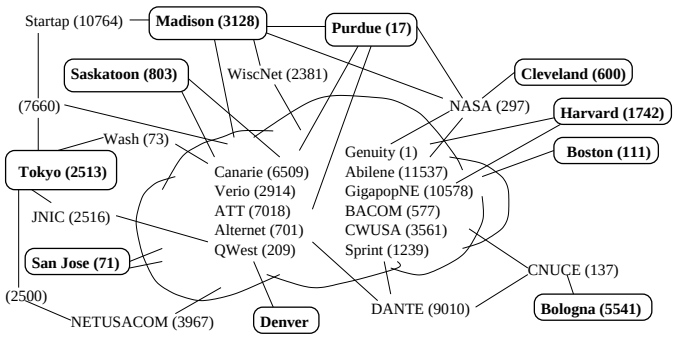

Fig. 2. Autonomous systems connecting WAWM nodes.

reside in universities, research institutions and commercial companies, providing a 34/56 mix of paths in commodity (commercial service providers) and non-commodity (such as Internet2/Abilene) administrative domains. Figures 1 and 2 depict maps of deployed WAWM nodes and the ASs they span. All systems are well connected to their networks via 10/100Mbps Ethernet cards.

B. Path Data

To assess the relationship between the data transfer latency along a path and that path's underlying physical properties we focused on the following six measurable path characteristics:

1. **Administrative Domain (Admin)**: We differentiate between two general categories of administrative domains in paths. If a path contains *any* commodity ISP then we designate that a *commodity path*. All others are *non–commodity* and are made up of the following carriers: Internet2/Abilene, NASA, RIPE, APAN and Canarie.

2. **Bottleneck Bandwidth (B–BW)**: the bottleneck bandwidth or the static path capacities were measured using two tools – ***pathrate*** [34] and ***nettimer*** [36]. Measured bottleneck bandwidths ranged from 3 to 80 Mbps.

3. **Router Hops(RHops)**: `traceroute` was used to measure the number of router hops for each path. Router hop counts ranged from 8 to 29.

4. **Round Trip Time (RTT)**: We considered two alternatives to measure the RTT – `traceroute` data and estimates from packet traces. The packet traces from file downloads are more likely to experience queuing delays. We were interested in the *minimum* delays due to propagation, thus we decided to use the RTTs from `traceroutes`. The average of the median RTTs from each hourly `traceroute` was used to obtain typical round trip time estimates between the paths. These varied from 2ms to 160ms, and 37ms to 330ms for transcontinental paths and transoceanic paths respectively.

5. **Autonomous System Hops (ASHops)**: ***ASRoute*** [37] is a tool built on top of `traceroute` that summarizes traceroute data at the autonomous system level. Besides estimating AS-hops, *ASRoute* also helps distinguish between InterAS and IntraAS route fluctuations. AS-hop counts in our data set ranged between 3 to 7.

6. **Physical Distance (Dist)**: corresponds to the road/air distances between cities as appropriate. These values can be obtained from popular on-line mileage calculators. These ranged from less than 50 miles between Boston Univ./Harvard Univ. to 6700 miles between Boston Univ./Tokyo.

Summary statistics for these values organized by administrative domain is given in Table I.

C. Data Collection Method

Measurements were taken hourly across the full mesh of systems for a period of 45 days from December, 2001 through January, 2002. Each measurement consisted of a `traceroute` (from server to client) followed by file transfers using `wget` [38] of 3 distinct files. For unbiased measurements, an exponentially distributed delay with a mean value of 6 seconds was introduced between the individual file downloads and their order was randomized [39]. File sizes used in downloads were 5KB, 100KB and 1MB. Our intention in choosing these file sizes was in an attempt to exercise TCP in a variety of operating regimes that might be considered "typical" [40]. The 1MB and to a lesser extent the 100KB file transfers should be dominated by TCP's congestion avoidance regime while connection set up and tear down should have a significant impact on the 5KB file transfers. A summary of the data we collected for this study is given in Table II. There are slight variations in the total number of files transferred due to outages and/or censoring of significant out-lier samples (caused by a variety of problems including measurement misconfigurations and delay spikes [21]) during the measurement period.

We used two tools different tools to assess the static bottleneck bandwidths for each path in our measurement en-

TABLE I
MEAN/VARIANCE VALUES OF PHYSICAL PROPERTIES WITH PATHS IN *Commodity(0)* AND *Non-Commodity(1)* ADM DOMAINS

Admin	**Num Paths**	μ / υ **AS hops**	μ / υ **Router Hops**	μ / υ **Distance(mi)**	μ / υ **RTT(ms)**	μ / υ **B–BW(Mbps)**
0	34	4.10 / 0.72	17.04 / 16.02	2,672.65 / 2,963,849	111.02 / 2,581	6.23 / 11.63
1	56	4.40 / 0.84	15.40 / 18.30	3,210.18 / 5,215,036	112.92 / 6,674	52.02 / 868.02

vironment: `pathrate` [34] and `nettimer` [36]. Our decision to use both was based on the fact that there is limited, wide-spread deployment and experience with these tools . An important issue with both tools is that they exert significant load on the network and as such their use in our study was limited. Our approach was to take daily measurements along all paths for a week and to then average the results for each tool.

Our assessment of the data revealed high variability in the values returned by `nettimer`, as well as a number of noticeable discrepancies. For example, on a path from Madison to Italy with a bottlneck of 10Mbps, `nettimer` consistently produced B–BW between 60-200Mbps. Hence we decided to restrict our analysis to be based on the values obtained from `pathrate`. The values shown for B–BW in Table I reflect the summary statistics for commodity and non-commodity paths. The almost order of magnitude difference in B–BW between the two path groups is quite likely an important factor in the prevelance of commodity links off the efficient frontier. Of course there were many route changes measured (ranging between 3 and 47 for individual paths) over the course of our entire study which could potentially lead to possibly significant differences in B–BW for individual paths. The B–BW measurements returned by `pathrate` for our paths were in fact quite stable over the course of the measurement period. We attribute this to most of the fluctuations being minor intra-AS route fluctuations (often happening far away from the endpoints), that usually have little or no effect on the bottlenecks.

D. Extracting Components of Transfer Delay

The original algorithm for applying critical path analysis (CPA) to a TCP transaction is described in [11]. That algorithm enabled total delay for TCP data transfers to be decomposed into six distinct sub-components (server delay, client delay and four different aspects of network delay). The algorithm was based on the observation that not all packets in TCP data transfers actually contribute to total delay since multiple data packets can be sent per round trip time. Therefore, the sequence of packets that *are* responsible for total delay (the critical path) must be identified before components of delay can be established.

TABLE II
SUMMARY OF MEASUREMENT DATA USED IN THIS STUDY

Data Type	**Total transfers**	**Transfers with loss**
5KB File	80,870	1,205
100KB File	77,299	8,702
1MB File	77,986	16,194
Traceroute	79,358	

The first step in identifying the critical path in TCP transfers is to associate sets of data packets with the ACK packet that triggered their emission in a given transfer using packet traces collected at each end host (refer to [11] for a description of handling connection set-up and tear-down). This is done by essentially simulating the TCP algorithm based on the observed ACK sequence and inferred packet loss. Next, the critical path is established by tracing back up from the *last data packet* to the ACK that triggered it, to the data packet that triggered that ACK, etcetera, back to the first data packet sent. The resulting sequence of data and ACK packets form the critical path for the transfer.

After establishing the critical path, components of delay are identified through a profiling process. This process results in the division of the total delay for a given data transfer into different latency categories. The first category, server delay, is the sum of differences between the receipt of ACKs on the critical path and the emission of data triggered by those ACKs. The second category, client delay, is the sum of differences between the receipt of data on the critical path and emission of ACKs. While critical path analysis enables delays caused by end hosts to be isolated, we do not consider them in this study[2]. Our focus was on delays caused by the network which is divided into the following sub-components:

1. *Propagation Delay*: The minimum packet transit time in each direction multiplied by number of packets transferred in each direction that are on the critical path.
2. *Queuing Delay*: Sum of differences between propagation delay and actual delay for all packets on the critical

[2]WAWM hosts used for this work were not used for any other purposes thus end host delays were minimal.

path.
3. *Network Variation Delay*: A simple way to detect route fluctuations is to look for changes in the TTL field in the IP header. This suffers from the weaknesses that is not possible to detect route fluctuations where the number of hops don't change. We contend that the effects of this possibility would be minimal and that our methodology would tend to detect almost all route fluctuations that would have a significant impact on delay.
4. *Fast Retransmit Delay*: Sum of delays of packet losses recognized by the TCP fast retransmit mechanism.
5. *Coarse Timeout Delay*: Sum of delays of packet losses recognized by the TCP coarse-grained timeout mechanism.
6. *Exponential Back-off Delay*: Sum of delays of packet losses recognized by the TCP coarse-grained timeout mechanism which were then followed by a loss of the same packet resulting in the use of TCP's exponential back-off mechanism.
7. *DNS Delay*: Total delay caused by name resolution process.

For this study, we made the following enhancements to the original algorithm for applying CPA to TCP transactions:

- ability to monitor critical path from a single end point
- establish critical path and delay breakdowns in real time
- relaxed requirement for tightly synchronized clocks
- separation of queuing and network variation delays (as explained above)
- separation of coarse timeout delay and exponential back-off delay

To facilitate these enhancements, we instrumented Linux-2.2 kernels running on the measurement hosts to add an extra TCP option for CPA and also disabled SACK. The modifications to the sender, receiver and timers were relatively minor. An additional benefit of this kernel-level implementation of CPA is that details of transfer latency can now be evaluated for *any application* which uses TCP as its transport. The implementation in [11] was specific to HTTP transactions.

In general, the task of critical path profiling is complicated by the fact that the end hosts do not have synchronized clocks. This is handled through the use of GPS clocks in [11], however that requirement significantly restricts wide spread deployment of CPA capabilities. We treated this problem in our new CPA implementation. By calculating latencies for every round (as opposed to every packet), it is possible to account for clock synchronization problems[3]. Our assumption is that although the clock offset is non-zero, the clock skew is negligible during the course of a single download which is the only clock synchronization pathology that would affect our implementation.

[3] See [41] for a treatment of clock synchronization problems in wide area measurements

To reduce the dimensionality of our data, we consider three sub-components of total delay: propagation, queuing and loss. We observe that network variation delay almost never occurs (we observed changes in TTL values during only 53 transfers). Loss delay in our study is the sum of fast retransmit, coarse timeout and exponential back-off delays measured in each transfer.

IV. Results

Our consideration of both the transfer delay properties and path properties resulted in a highly dimensional data set. We systematically employed different statistical tests to evaluate characteristics of network delays and how they relate to path properties. In this section, we describe four specific focus areas of our study, the analysis methods we use and the results of the evaluations.

Throughout this section we refer to individual paths using abbreviations that consist of two letters. The first indicates the client and the second indicates the server. The abbreviations used for each host are shown in Figure 1. It would clearly be better to investigate *distributional* characteristics of the features that we measured. However, the dimensionality of our data significantly complicates this approach and would limit the breadth of the results we could report. As a result, all of our analysis focuses on mean and variance values for each property under consideration for each path (unless otherwise specified).

A. Path Properties

Simple qualitative assessment of the total network delay data showed distinctly different characteristics for different paths. Examples of the time series for two different commodity paths can be seen in Figures 3. The figure on top illustrates the typical behavior of a path on the efficient frontier, one that is highly predictable and where delays are dominated by propogation. This is in contrast to the path on the bottom that experiences significant variability that is most often associated with queuing and loss.

Our first step in attempting to evaluate network delay characteristics was to assess similarities in path properties. Establishing similarities based on path properties may help explain variability in data transfer measurements and may also serve as a basis for operational application of our results. The difficulty in assessing similarities in path properties was that there were six features which could be considered (Dist, RTT, RHops, ASHops, Admin, B–BW) and

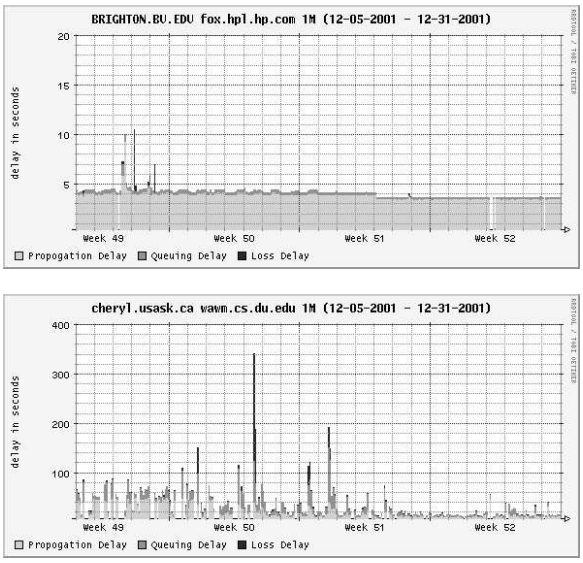

Fig. 3. Network delay time series of 1MB transfers for two paths over a 4 week period–(top: Boston-San Jose)(bottom: Saskatoon-Denver). The delay in the y-axis ranges from 0-20 secs for the top fi gure and from 0-400 secs for the bottom

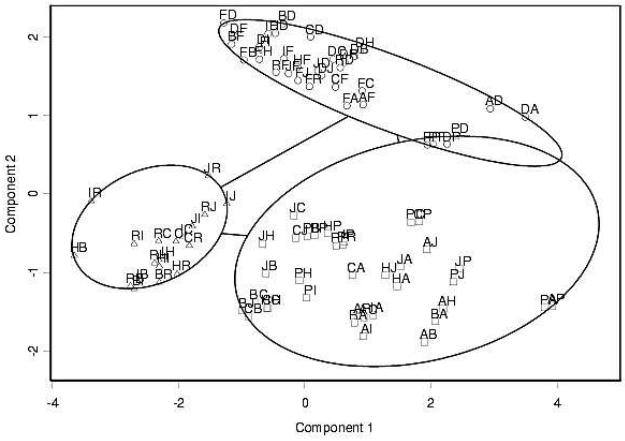

Fig. 4. Clustering results using fi rst two components from PCA

we needed to reduce dimensionality to determine whether or not there were natural groupings among paths.

Our solution was to apply Principal Components Analysis (PCA) to the mean values of our path property data. PCA is a standard method for combining features to evaluate groups. PCA constructs linear combinations of features in order to project high–dimensional data into lower dimensional (component) space. Our application of PCA showed that the first two principal components were sufficient to explain 73.3% of the variability in the path property data. We then used the *agglomerative hierarchical method* suggested in [42] to cluster paths using the first two principle components. Qualitative assessment reveals three or four distinct clusters as shown in Figure 4. Closer evaluation of these clusters shows that they generally separate based on administrative domain type and by distance (between client and server) where any measure of distance works well as a discriminator.

B. Data Transfer Latency and Path Properties

We considered mean transfer times for 1MB to begin the process of characterizing the relationships between path properties and data transfer delays. An additional objective of this analysis was to understand the predictive capability of path properties vis–a–vis data transfer delay. Once again, the number of the different path properties was a complicating factor.

Our approach to this analysis was to use *robust regression analysis* for mean values of four path properties (RHops, ASHops, Dist. and RTT) versus mean total delay (we omit B–BW from consideration due to the limited size of that data set). Robust regression (MM–estimator) performs high breakdown point and high efficiency regression with a test for bias according to the method proposed in [43]. This algorithm results in estimates that are strongly consistent and asymptotically normal.

In assessing the results of the robust regression analysis, we find a strong linear relationship between each of the path properties and mean total delay. We also find that there are a number of significant outliers – these are all from paths over commodity networks. The best fit comes from the comparison of mean total delay and RTT shown in Figure 5 (refer to [44] for details on residuals for this regression). We note two interesting findings. Almost *all* of the path properties (except for B–BW) lead to virtually identical relationship. Even the most static property – physical distance – is a good indicator of typical transfer time for most paths as can be seen in Figure 6. Secondly, the shortest (HB - Harvard, Boston) and longest paths (AP, PA - Japan, Italy) seem to fall at the two extremes of our linear fits.

The absence of a similar linear fit with respect to B–BW is quite apparant from Figure 7. We believe that this is an artifact of TCP window size limitations (default 64 KB in our installations) that prevent connections from exercising the full bandwidth of the end to end path. Although we considered increasing the buffer size, it seemed to be in conflict with our desire to measure latencies experienced by typical file transfers. This lack of correlation seems to indicate the need for a critical reexamination of existing window sizes and also illustrates opportunities for dynamic window scaling algorithms [45].

These plots highlight the fact that most of the paths typ-

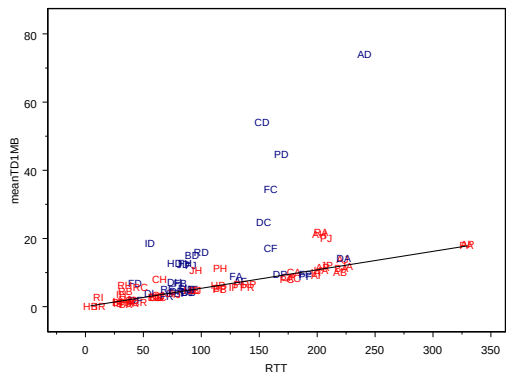

Fig. 5. Robust regression results comparing mean total delay for 1MB data transfers(meanTD1Mb) in secs vs RTT in ms

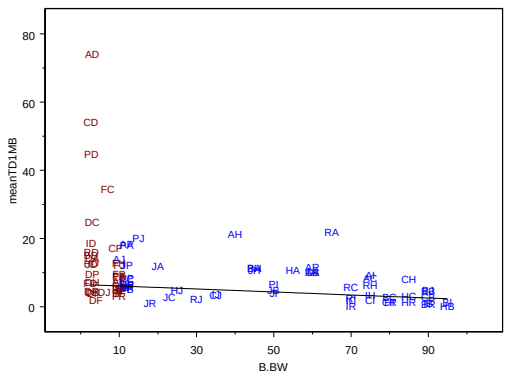

Fig. 7. Scatter plot comparing mean total delay for 1MB data transfers(meanTD1Mb) in secs vs B-BW in Mbps

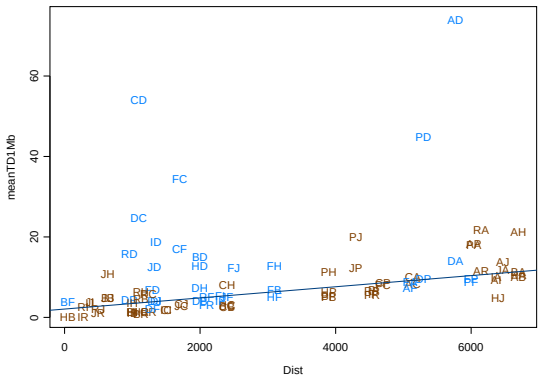

Fig. 6. Robust regression results comparing total delay for 1MB transfers(meanTD1Mb) in secs vs physical dist in miles

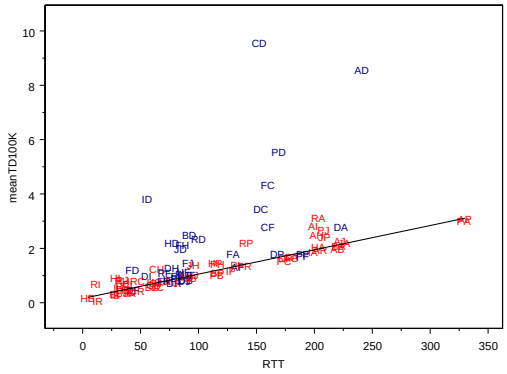

Fig. 8. Robust regression results comparing mean total delay for 100KB transfers(meanTD100K) in secs vs RTT in ms

ically operate very close to their maximum performance capability (*ie.* data transfer latency is limited by speed of light delays, router switching delays and receive window size). We call this regime the *efficient frontier*. Smaller files demonstrate this property most strongly as we will see shortly. However, there are some paths along which large files typically are transferred with great difficulty *ie.* their typical behavior is quite far from the efficient frontier. These observations suggest that for most paths, repeated probing to determine "distance" may be unnecessary. Likewise for paths which do not operate along the efficient frontier, these results suggest that *no* path property is a good indicator of data transfer delay.

Robust regressions for transfer times of 100KB and 5KB files versus path properties reveal an even stronger linear relationship with fewer intense outliers than the 1MB files. Once again, all of the path properties are good indicators of transfer delay with fewer distinct outliers. We show the results when comparing delay for these smaller file to RTT in Figures 8 and 9. These results indicate that predictive capability for throughput increases as file size decreases.

Next, we consider broader aspects of the distributions of transfer times of 1MB files. In Figure 10, we show cumulative distribution function (CDF) graphs of the five paths furthest from the efficient frontier along with five paths with approximately the same RTT measurements. Contrasts in distributional characteristics are obvious with the paths furthest from the efficient frontier showing much greater variability. Another way to consider variability is seen in Figure 11 which compares the standard deviation of transfer times to the mean (and includes a robust regression line). This figure shows a reasonably linear relationship for all but a few paths indicating that the distributional shape of most paths is similar but scaled by physical distance. Paths which do not have this property are the same paths which typically operate far from the efficient frontier.

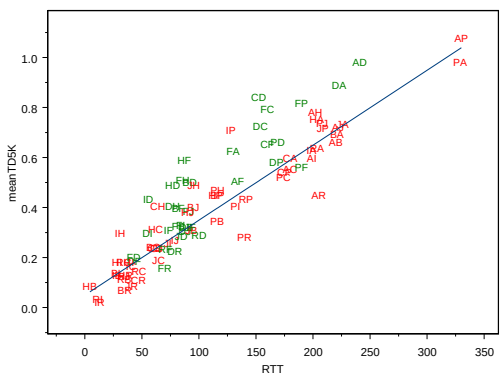

Fig. 9. Robust regression results comparing mean total delay for 5KB transfers (meanTD5K) in secs vs RTT in ms

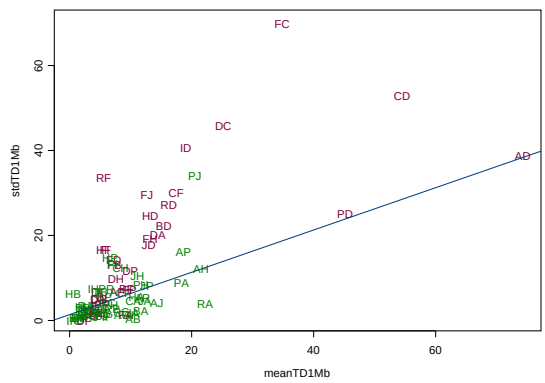

Fig. 11. Robust regression of standard deviation (stdTD1Mb) in secs vs mean (meanTD1Mb) in secs for 1MB transfers

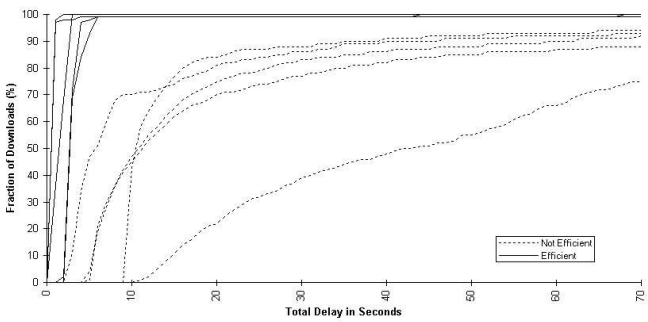

Fig. 10. Cumulative distributions of transfer times of 1MB files for 5 paths which operate on the efficient frontier and the 5 which are furthest from the efficient frontier.

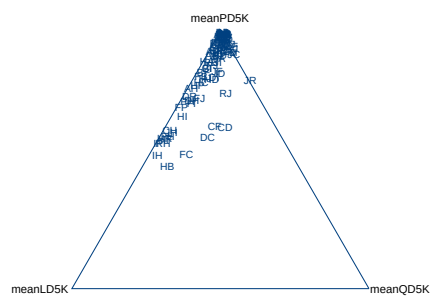

Fig. 12. Triangle plot comparing relative contribution to mean total transfer delay of propagation (meanPD5K), queuing (meanQD5K)and loss (meanLD5K) for 5KB file

C. *Effects of Propagation, Queuing and Loss on Transfer Latency*

To characterize the contributions of propagation delay, queuing delay and loss delay to total data transfer latency, we create triangle plots for each of the three file sizes shown in Figures 12, 13 and 14. The triangle plots represent the contribution of the mean of each sub–component normalized to the mean of the total transfer delay. If a path point is near a vertex in these diagrams that means that transfer delays along that path are strongly influenced by one of the three sub–components. Likewise a path point in the middle of the triangle means that there is equal contribution of each sub–component to total transfer delay.

The triangle plots show that propagation delay is the most significant sub–component of mean total delay for most paths and for all three file sizes. This reinforces the observations about operation along the efficient frontier; namely that data transfers are typically limited by speed of light considerations for most paths. Closer examination of the 1MB data transfers indicates that the effects of queuing and loss are roughly equivalent and may, in fact, be biased toward queuing for paths that are less dominated by propagation. Once again, the paths that have the largest queuing and loss components are almost all commodity.

Paths with significant loss and queuing components for 100KB and 5KB files are also almost all commodity, however their characteristics are quite different that the 1MB data transfers. For the smaller files, loss appears to be a more important component of total mean delay – especially for the 5K files which seem to typically unaffected by queuing delay.

Another consideration is the effect that queuing and loss have on the variability of mean total delay. We evaluate this question by assessing plots of standard deviation of total delay for 1MB data transfers versus mean loss and queuing delays in Figures 15 and 16. These figures include robust regressions of standard deviation of total delay versus mean loss and queuing delays and indicate, not surprisingly, that paths with higher loss and queuing have higher variability in delays.

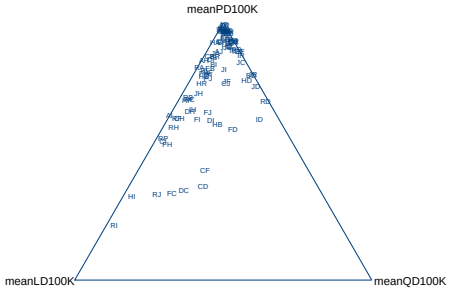

Fig. 13. Triangle plot comparing relative contribution to mean total transfer delay of propagation (meanPD100K), queuing (meanQD100K)and loss (meanLD100K) for 100KB fi le

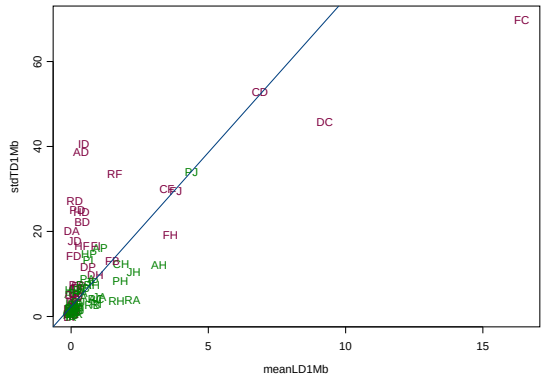

Fig. 15. Robust regression of std dev of total delay for 1MB fi le (stdTD1Mb) in secs vs mean loss delay (meanLD1MB) in secs

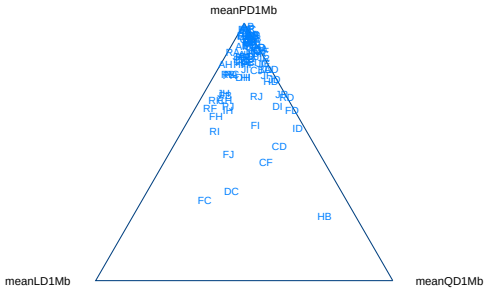

Fig. 14. Triangle plot comparing relative contribution to mean total transfer delay of propagation (meanPD1Mb), queuing (meanQD1Mb)and loss (meanLD1Mb) for 1MB fi le

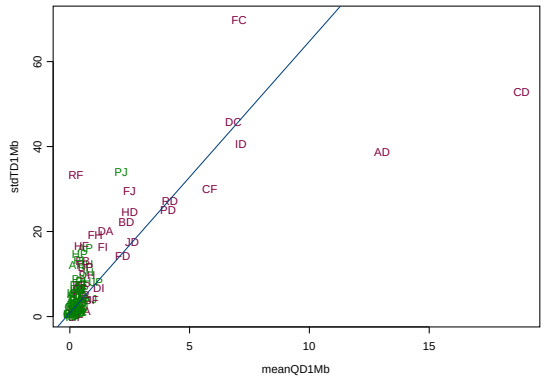

Fig. 16. Robust regression of std dev of total delay for 1MB fi le (stdTD1Mb) in secs vs mean queuing delay (meanQD1MB) in secs

D. Periodicity in Network Delays

A well known phenomenon of network traffic is its characteristic diurnal behavior. We expected to see similar effects in data transfer delays and the sub–components; specifically that delays would tend to increase during the day and subside to the efficient frontier at night. However, qualitative assessment of the time series of delays for paths showed that in many instances this was not clearly the case.

To investigate this we developed a sum of squares method for evaluating periodicity of data transfer delays for each path as follows:

- $y'(i)$ = Average Queuing in $Hour(i)$ $[i = 1...24]$
- $\sigma(i)$ = Std. Dev of Queuing in $Hour(i)$ $[i = 1...24]$
- $y'' = \frac{\sum_{i=1}^{24} y'(i)}{24}$
- $SS_{Diurnal} = \sum_{i=1}^{24} (y'(i) - y'')^2$
- $SS_{Residual} = \sum_{i=1}^{24} \sigma(i)^2$
- $R^2 = \frac{SS_{Diurnal}}{(SS_{Residual} + SS_{Diurnal})}$

In this case, $0 < R^2 < 1$. Here R^2 is a measure of *periodicity* and not accuracy and hence *any* value of R^2 greater than 0 is an indication of periodicity in the data. It makes sense that R^2 values are low, because most paths operate in the efficient frontier where paths do not exhibit diurnal behavior. Our purpose was to identify the levels to which individual delay components affected periodicity.

The results of running this algorithm on the 1MB transfer data are given in Table III. The summary statistics indicate a lower periodicity in total mean delay in the non–commodity paths. Interestingly, they also show that the strongest periodic sub–component was in queuing delay in commodity paths and loss delay in non–commodity paths.

TABLE III
MEAN/VARIANCE OF R^2 VALUES OF PERIODICITY OF DELAYS IN BOTH ADMINISTRATIVE DOMAINS.

Admin	μ/v Total Delay	μ/v Prop Delay	μ/v Q-ing Delay	μ/v Loss Delay
Both	0.0521/0.0018	0.0488/0.0017	0.0644/0.0043	0.0593/0.0056
Commodity	0.0681/0.0022	0.0578/0.002	0.0814/0.0059	0.0577/0.0054
Non-Commodity	0.0435/0.0018	0.0432/0.0014	0.0538/0.0031	0.0603/0.0057
5 Non-Efficient	0.115/0.0028	0.1/0.0047	0.181/0.0048	0.081/0.0014
5 Efficient	0.047/0.001	0.04/0.001	0.033/0.0003	0.016/0.0001

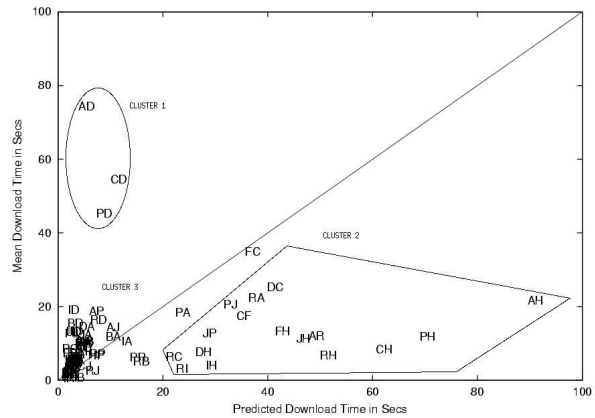

Fig. 17. Scatter plots of predicted versus measured throughput for 1MB files for all paths

V. DISCUSSION

A critical aspect of any measurement study is it's relevance to important network design questions. In this section we discuss the implications of our results in the areas of analytic modeling of TCP throughput and operational aspects of wide area networking.

A. Implications for TCP Throughput Modeling

A series of models have been developed which attempt to predict throughput of TCP Reno beginning with [13] and culminating with [12]. A recent application of the throughput model proposed in [14] is as means for modulating send rate in a transport level congestion control scheme [5]. Clearly, accuracy of the model in this context is important. These models use only average RTT and average packet loss rate as input related to network conditions.

We extracted average RTT and average packet loss rates from our 1MB file transfers and plug them into the model described by Cardwell *et al.* [12]. We then compare these predicted values for expected average throughput to the measured values and plotted the results for all paths. Our motivation behind this undertaking was to better understand properties for paths operating far from the efficient frontier. We expected the model to be able to predict delays accurately for paths on the efficient frontier, and to be less accurate for the other paths.

However, as can be seen in Figure 17, there is little correlation between *any* predicted and measured values of throughput. Further examination reveals two distinct groups of paths which deviate significantly from the predicted throughput; those that have throughput that is lower than the predicted (cluster 1) and those that have throughput that is higher than predicted (cluster 2).

The set of paths with average throughput lower than predicted by the Cardwell model (cluster 1) have transfer times that include a significantly larger queuing component as is seen in Figure 18. Queuing delay is not considered by current throughput models. Delays for the paths in cluster 3 (most of which are paths on the efficient frontier), also tend to be underpredicted by the model for the same reason. Our results suggest that one means for addressing this would be to add a queuing delay factor to the models that is proportional to file size and depends on whether or not the transfer is on a path that operates on the efficient frontier.

The set of paths with average throughput that is higher than predicted by the Cardwell model (cluster 2) are more difficult to understand. Figure 18 shows that these paths typically have greater loss delays. Studies of packet loss behavior (*eg.* [20], [21]), show that losses are typically bursty, but loss episodes can be well modeled as an IID process. Current throughput models however assume a uniform loss process. Our conjecture is that the more uniform loss process leads to smaller average congestion window sizes and thus lower overall throughput than a more bursty process. We intend to investigate this in future work. If our conjecture is correct, this would suggest that a more accurate loss model could greatly benefit existing throughput models.

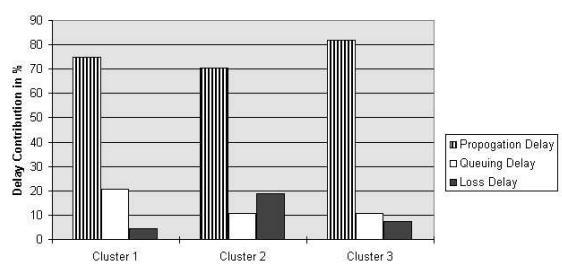

Fig. 18. A comparison of network delays for the three path clusters that arise when comparing measured throughput to modeled throughput for 1Mb files.

B. Implications for Network Operations

Our results related to operation along an efficient frontier have implications for for engineers who are responsible for daily operations of wide area networks. In particular, tomographic techniques [46] could be employed to isolate specific parts of paths that are responsible for poor performance. These could then be either upgraded or avoided altogether.

A second example an application for our findings in the operational domain is in the area of distance measurement such as the IDMaps system [15]. An example of a potential use for this kind of measurement is in mapping customer requests to caches in content delivery networks. The fact that static path properties such as physical distance are often as accurate as more dynamic features for predicting expected throughput could greatly simplify this task.

A final example of a domain that could utilize our results is in the area of routing overlay networks [16]. These networks are conceived to provide application level routing capability between hosts so as to improve robustness and availability in the network. Our results indicate that it may be more appropriate to select an overlay path between client and server that combines paths on the efficient frontier instead of a more direct, shorter path that does not operate on the efficient frontier.

VI. Conclusions and Future Work

We present an analysis of a large set of TCP file transfers and `traceroute` measurements between 10 Internet hosts measured over a period of 45 days. Our objective was to expand the understanding of delays in end–to–end data transfers and how those delays relate to the underlying path properties. We developed a new, kernel–level implementation of critical path analysis for TCP transactions which enabled establishment and profiling of the critical path in real time. Our implementation also expanded upon prior work by enabling additional properties of network delays such as packet loss resulting in exponential back-off to be recognized. After gathering the data, we employed multivariate analysis techniques to evaluate both path and delay properties.

Our analysis provides insight on the details of network data transfer delays in a number of ways. First, we show that linear combinations of path properties are sufficient to group paths that may not otherwise appear to be similar. Clustering based on principal components resulted in groups differentiable by backbone provider and physical distance. Second, we evaluate end to end path properties as a predictor of total mean file transfer latency, and find that expected transfer latency can be effectively predicted by a number of path properties. We present the notion of an *efficient frontier* of performance for most network paths. However, some paths have typical behavior that is far from this threshold. Third, we break down the network component of file transfer latency into the three categories of propagation, queuing and loss and compare the contribution of each of these components for three distinct file sizes. We find that propagation delay is the dominant aspect of expected total delay for most paths. We also find that the effects of queuing and loss depend on file size and are evident typically for a minority of paths. On these paths, queuing contributes most significantly to periodicity in total delays while loss contributes most significantly to variability in total delay.

Our results can have implications in the area of protocol and throughput performance modeling as well as in operational system deployments. The distinct regimes of path behavior either close to or far from the efficient frontier may mean that additional considerations could be made in throughput models. Operational system deployments such as distributed caching and routing overlay infrastructures which use distance metrics to associate clients with mirror servers may be justified in employing simple distance metrics such as physical distance for a majority of their systems.

Next steps include additional analysis along other dimensions of our data. For example, one aspect of critical path analysis is that we are able to capture the evolution of delays as they progress through a transaction. A potential application for this capability would be in dynamic routing or mirror selection during the course of transactions. We also plan to investigate our entire data set using signal analysis methods since our focus on mean delay values overlooks what very well may be important signals in our data. Finally, we plan to investigate other variants of TCP and to use the techniques established in this paper for measuring and evaluating non–TCP traffic.

References

[1] V. Padmanabhan and J. Mogul, "Improving HTTP latency," *Computer Networks and ISDN Systems*, vol. 28, pp. 25–35, December 1995.

[2] H. Frystyk-Nielsen, J. Gettys, A. Baird-Smith, E. Prud'hommeaux, H. Wium-Lie, and C. Lilley, "Network performance effects of HTTP/1.1, CSS1 and PNG," in *Proceedings of ACM SIGCOMM '97*, Cannes, France, Setpember 1997.

[3] J. Hoe, "Improving the start-up behavior of a congestion control scheme for TCP," in *Proceedings of ACM SIGCOMM '96*, Palo Alto, CA, August 1996.

[4] V. Padmanabhan and R. Katz, "TCP fast start: A technique for speeding up Web transfers," in *Proceedings of the IEEE GLOBECOM '98*, November 1998.

[5] S. Floyd, M. Handley, J. Padhye, and J. Widmer, "Equation-based

congestion control for unicast applications," in *Proceedings of ACM SIGCOMM '00*, Stockholm, Sweden, September 2000.

[6] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and Zipf-like distributions: Evidence and implications," in *Proceedings of IEEE INFOCOM '99*, New York, NY, March 1999.

[7] S. Michel, K. Nguyen, A. Rosenstein, S. Floyd, and V.Jacobson, "Adaptive Web caching: Towards a new global caching architecture," in *Proceedings of the 3rd Web Caching Workshop*, Manchester, England, June 1998.

[8] P. Barford and M. Crovella, "Measuring Web performance in the wide area," *Performance Evaluation Review*, August 1999.

[9] V. Jacobson, "traceroute," ftp://ftp.ee.lbl.gov/traceroute.tar.Z, 1989.

[10] V. Paxson, ," Invited talk, Workshop on NetworkRelated Data Management, May 2001.

[11] P. Barford and M. Crovella, "Critical path analysis of TCP transactions," in *Proceedings of ACM SIGCOMM '00*, Stockholm, Sweden, September 2000.

[12] N. Cardwell, S. Savage, and T. Anderson, "Modeling TCP latency," in *Proceedings of IEEE INFOCOM '00*, Tel-Aviv, Israel, March 2000.

[13] M. Mathis, J. Semke, J. Mahdavi, and T. Ott, "The macroscopic behavior of the TCP congestion avoidance algorithm," *Computer Communications Review*, vol. 27(3), July 1997.

[14] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP throughput: A simple model and its empirical validation," in *Proceedings of ACM SIGCOMM '98*, Vancouver, Canada, Setpember 1998.

[15] P. Francis, S. Jamin, C. Jin, Y. Jin, D. Raz, Y. Shavitt, and L. Zhang, "IDMaps: a global internet host distance estimation service," *IEEE/ACM Transactions on Networking*, vol. 9(5), October 2001.

[16] D. Andersen, H. Balakrishnan, M.F. Kaashoek, and R. Morris, "Resilient overlay networks," in *Proceedings of ACM SOSP*, Baniff, Canada, October 2001.

[17] R. Cáceres, P. Danzig, S. Jamin, and D. Mitzel, "Characteristics of wide-area TCP/IP conversations," in *Proceedings of ACM SIGCOMM '91*, September 1991.

[18] J. Bolot, "End-to-end packet delay and loss behavior in the Internet," in *Proceedings of ACM SIGCOMM '93*, San Francisco, Setpember 1993.

[19] V. Paxson, "End-to-end Internet packet dynamics," in *Proceedings of ACM SIGCOMM '97*, Cannes, France, September 1997.

[20] M. Yajnik, S. Moon, J. Kurose, and D. Towsley, "Measurement and modeling of temporal dependence in packet loss," in *Proceedings of IEEE INFOCOM '99*, New York, NY, March 1999.

[21] Y. Zhang, N. Duffield, V. Paxson, and S. Shenker, "On the constancy of Internet path properties," in *Proceedings of ACM SIGCOMM Internet Measurement Workshop '01*, San Francisco, November 2001.

[22] V. Paxson, "End-to-end routing behavior in the Internet," in *Proceedings of ACM SIGCOMM '96*, Palo Alto, CA, August 1996.

[23] R. Govindan and A. Reddy, "An analysis of internet inter-domain topology and route stability," in *Proceedings of IEEE INFOCOM '97*, Kobe, Japan, April 1997.

[24] S. Savage, A. Collins, E. Hoffman, J. Snell, and T. Anderson, "The endtoend effects of Internet path selection," in *Proceedings of ACM SIGCOMM '99*, Boston, MA, September 1999.

[25] J. Gwertzman and M. Seltzer, "The case for geographical push caching," in *Proceedings of the 5th Workshop of Hot Topics in Operating Systems*, 1995.

[26] C. Cuñha, *Trace Analysis and its Applications to Performance Enhancements of Distributed Information Systems*, Ph.D. thesis, Boston University, 1997.

[27] R. Carter and M. Crovella, "Dynamic server selection in the internet," in *Proceedings of IEEE Workshop on Architecture and Implementation of High Performance Communications Subsystems*, June 1995.

[28] B. Huffaker, "Comparison of end to end distance metrics," ISMA Routing and Topology Analysis Workshop, December 2001.

[29] S. Floyd and V. Paxson, "Difficulties in simulating the Internet," *IEEE/ACM Transactions on Networking*, vol. 9(4), pp. 392–403, August 2001.

[30] W. Leland, M. Taqqu, W. Willinger, and D. Wilson, "On the self-similar nature of Ethernet traffic (extended version)," *IEEE/ACM Transactions on Networking*, pp. 2:1–15, 1994.

[31] V. Paxson and S. Floyd, "Wide-area traffic: The failure of poisson modeling," *IEEE/ACM Transactions on Networking*, vol. 3(3), pp. 226–244, June 1995.

[32] M. Crovella and A. Bestavros, "Self-similarity in World Wide Web traffic: Evidence and possible causes," *IEEE/ACM Transactions on Networking*, vol. 5, no. 6, pp. 835–846, December 1997.

[33] M. Allman and V. Paxson, "On estimating endtoend network path properties," in *Proceedings of ACM SIGCOMM '99*, Boston, MA, September 1999.

[34] C. Dovrolis, P. Ramanathan, and D. Moore, "What do packet dispersion techniques measure?," in *Proceedings of IEEE INFOCOM '01*, Anchorage, Alaska, April 2001.

[35] S. Savage, "Sting: A tool for measuring one way packet loss," in *Proceedings of IEEE INFOCOM '00*, Tel Aviv, Israel, April 2000.

[36] K. Lai and M. Baker, "Nettimer: A tool for measuring bottleneck link bandwidth," in *Proceedings of the USENIX Symposium on Internet Technologies and Systems*, March 2001.

[37] J. Gast and P. Barford, "Resource deployment based on autonomous system clustering," Submitted for publication, 2002.

[38] "wget," $http://www.wget.org$, 2002.

[39] V. Paxson, *Measurements and Analysis of End-to-End Internet Dynamics*, Ph.D. thesis, University of California Berkeley, 1997.

[40] P. Barford, A. Bestavros, A. Bradley, and M. Crovella, "Changes in Web client access patterns," *Special Issue on World Wide Web Characterization and Performance Evaluation; World Wide Web Journal*, December 1998.

[41] V. Paxson, "On calibrating measurements of packet transit times," in *Proceedings of ACM SIGMETRICS '98*, Madison, WI, June 1998, pp. 11–21.

[42] W. Venables and B. Ripley, *Modern Applied Statistics with Splus*, Springer–Verlag, New York, 3rd edition, 2000.

[43] V. Yohai, W. Stahel, and R. Zamar, *A Procedure for Robust Estimation and Inference in Linear Regression*, chapter Directions in Robust Statistics and Diagnostics, Part II, Springer-Verlag, 1991.

[44] P. Barford, D. Donoho, A. Flesia, and V. Yegneswaran, "Characteristics of network delays in wide area file transfers," Tech. Rep. 1441, University of Wisconsin - Madison, March 2002.

[45] E. Weigle and Wu chun Feng, "Dynamic right sizing: A simulatino study," in *Proceedings of the 10th IEEE conference on Computer Communications and Networks*, October 2001.

[46] T. Bu, N. Duffield, F. Lo Presti, and D. Towsley, "Network tomography on general topologies," in *Proceedings of ACM SIGMETRICS '02*, Marina Del Rey, CA, June 2002.