Computer Sciences Department

Characteristics of Network Delays In Wide Area File Transfers

Paul Barford
David Donoho
Ana Georgina Flesia
Vinod Yegneswaran

Technical Report #1443

July 2002

UNIVERSITY OF WISCONSIN MADISON

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. Finally, we show that path properties can also be effective predictors of both queuing and loss.**

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.

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.

The general objective of our study is to broaden understanding of the factors that contribute to delays in wide area TCP data transfers, using a large data set collected in a distributed Internet measurement infrastructure. The data set consists of measurements of over 236,000 data 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 sub-components of propagation, queuing and loss?" Consideration of this question led us to consider the influence of the properties of the network paths between end hosts and how network delays break down into the different sub-components 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 sub-components. 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 sub-components 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 sub-components 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 at-

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

tention. Next, we systematically employed a number of multivariate analysis techniques to understand details of behavior along different paths and to identify path properties 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. Not surprisingly, these are all paths that contain commodity Internet links (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]. 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. 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 obvious is for network engineers 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 [13]. Our findings indicate that static path properties such as physical distance are often as accurate as more dynamic features like RTT in term 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 [14]. 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.

The organization of the remainder of this paper is as follows. In Section II, we discuss prior work related to our study. In Section III, we describe our measurement methods and details of the data collected for this study. In Section IV, we describe our analysis methods and the results of their application to our data. We descuss the implications of our results in the areas of analytic modeling and network operations in Section V. We summarize, conclude and discuss future work in Section VI

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 [15], [16], [17], [18], [19], and studies that assess Internet routing and path characteristics [20], [21], [22]. Of these, the work that is perhaps most similar to ours is that of Zhang *et al.* [19]. 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 [13].

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 [23], topological distance [24] and latency [25]. End-to-end distance metrics are a topic of on-going study. An example is recent work by Huffaker in [26] 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 [27] . Perhaps the most successful work in this regard has been the establishment of self-similarity as a fundamental characteristic of network packet traffic [28], [29], [30]. 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 [31]. 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 [32] and [33]. Our work differs from these

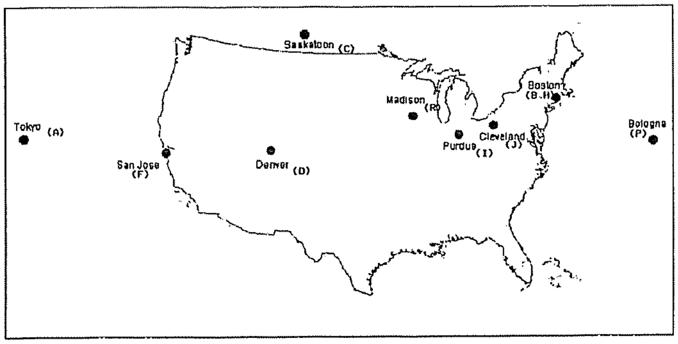

Fig. 1. Deployment of WAWM nodes (2 systems in Boston). Included are abbreviations used to identify hosts.

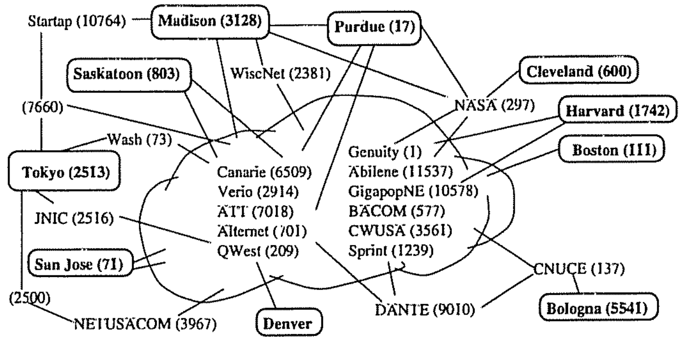

Fig. 2. Autonomous systems connecting WAWM nodes.

in our investigation of more static notions of path properties (ie. route characteristics) as predictors of TCP data transfer latency.

A series of models for steady-state TCP throughput and latency have been developed in [12], [34], [35]. These models require parameters that define TCP's different operating regimes - two of the most important of these are RTT and loss. Our work can serve as a basis for validating these models and for identifying conditions under which they are more and less effective.

We employ critical path analysis techniques described in [11] to break down total transfer delay into subcomponents. Establishing the critical path in a system where events happen in parallel enables causes for delays to be pinpointed. We extend this notion of critical path analysis for TCP transactions in a number of ways which are described in the next section.

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 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*** [32] 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 propogation, 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 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 (either commodity or non-commodity) is given in Table I.

TABLE I

MEAN/VARIANCE VALUES OF PHYSICAL PROPERTIES WITH PATHS IN BOTH ADMINISTRATIVE DOMAINS. *Adm: 0 = Commodity, 1 = Non-Commodity*

Adm	Num Paths	μ / v AS hops	μ / v Router Hops	μ / v Distance(mi)	μ / v RTT(ms)	μ / v B–BW(Mbps)
0	34	4.10 / 0.72	17.04 / 16.02	2672.65 / 2,963,849	111.02 / 2,581	6.23 / 11.63
1	56	4.40 / 0.84	15.40 / 18.30	3210.18 / 5,215,036	112.92 / 6,674	52.02 / 868.02

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 out-lier samples during the measurement period.

We used two tools different tools to assess the static bottleneck bandwidths for each path in our measurement environment: `pathrate` [32] and `nettimer` [36]. Our decision to use both was based on the fact that there is limited, wide-spread deployment of these tools and our experience with both was minimal. An important issue with both tools is that they exert significant load on the network and as such we decided to limit their use in our study. 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 few 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. 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.

TABLE II

SUMMARY OF MEASUREMENT DATA USED IN THIS STUDY

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

D. Extracting Components of Transfer Delay

The original method of applying Critical path analysis (CPA) to TCP transaction is described in [11]. That work develops an algorithm for applying CPA to packet traces collected at end hosts. The algorithm extracts the exact sequence of packets that determines the total file transfer delay. It then uses this sequence of packets to decompose the total transfer delay into six distinct sub-components: server delay, client delay, packet loss delay (time-out and fast retransmit), network variation delay and propagation delay. A conceptual diagram of this process is shown in Figure 3.

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
- ability to establish critical path and delay breakdowns in real time
- relaxed requirement for tightly synchronized clocks
- separation of coarse timeouts and exponential back-off delays
- separation of queuing and network variation delays

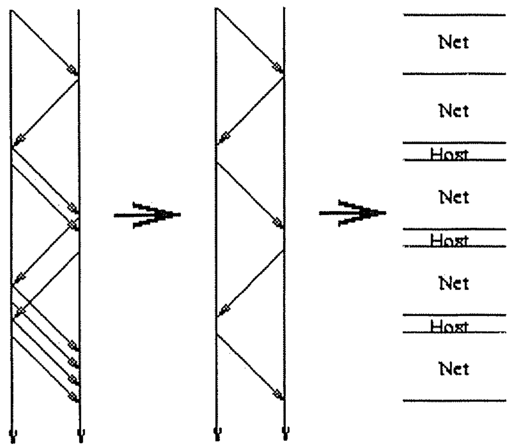

Fig. 3. Conceptual diagram of the breakdown of an original packet flow (left) into the critical path (center) then decomposed into subcomponents of network and host delay (left).

To facilitate these enhancements we implemented CPA for TCP via kernel instrumentation. Linux-2.2 kernels running on the measurement hosts were instrumented to add an extra TCP option for CPA and SACK was disabled. The modifications to the sender, receiver and timers were relatively minor. Unlike the original critical path algorithm, our kernel implementation allows construction of the critical path in a single pass. This is because the critical path options recorded by the packet traces allow to us unambiguously match an acknowledgment with the packet that triggered it. An additional benefit of our 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.

Once the critical path has been constructed, critical path profiling involves assigning the total delay in each round to 9 different latency categories. 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 and the sub-components consist of:
1. *Propagation Delay*: Sum of minimum transit times for packets in each direction on the critical path.
2. *Queuing Delay*: Sum of differences between propagation delay and actual delay for packets on the critical 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.

The task of critical path profiling is complicated by the fact that the hosts are not synchronized. 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. For every packet on the critical path, except for the last packet there is an acknowledgment that is also on the critical path. 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.

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). We do not consider DNS delay as a focus of this work. 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 five 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

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

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

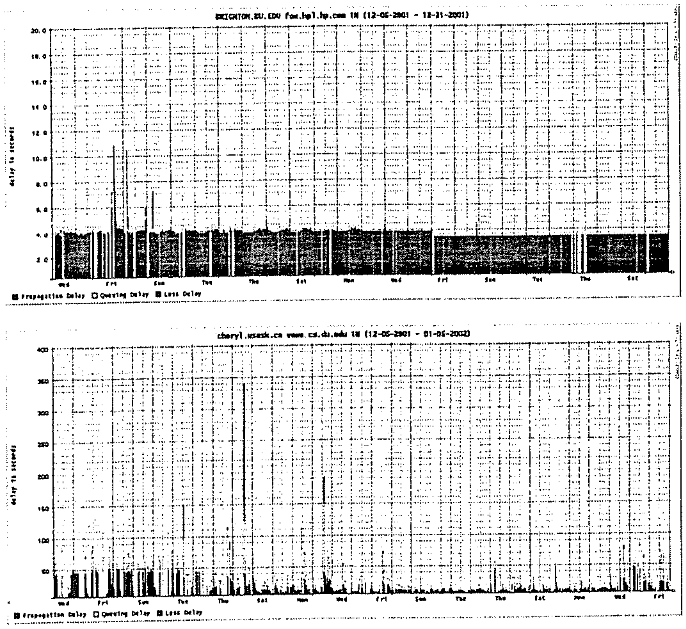

Fig. 4. Network delay time series for two paths over a 4 week period – (top: Boston-San Jose) (bottom: Saskatoon-Denver). The delay in the y-axis ranges from 0-20 seconds for the top figure and from 0-400 seconds for the bottom

abbreviations used for each host are shown in Figure 1. It is clearly better to investigate *distributional* characteristics. However the dimensionality of this data significantly complicates this approach. Hence 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 4. 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, 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 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 standard a 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. It seeks the best representation of the components from the least squares perspective. Following PCA we apply cluster analysis to the first two components using the Partition Around Medoids (PAM) method [42]. While PAM is applied in the full 6–dimensional space, it is displayed in graphically in 2–dimensional space. This is why some subclustering that seems visually evident in the 2–dimensional display does not end up being reflected in the full 2–dimensional solution.

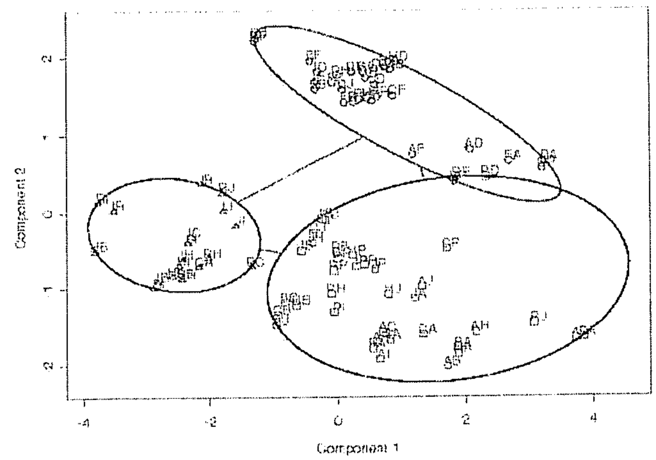

Fig. 5. Clustering results using first two components from PCA

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. This suggests that linear combinations of properties are quite effective in identifying paths with similar characteristics. We then used the *agglomerative hierarchical method* suggested in [42] to cluster paths using the first two principle components. Qualitative assessment of this result indicates that there are three or four distinct clusters in the path properties. Figure 5 shows the clusters of paths that arise using these methods. Closer evaluation of these clusters shows that they generally separate based on administrative backbone 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 – not surprisingly these are all from paths over commodity networks. The best fit comes from the comparison of mean total delay and RTT shown in Figures 6 and 7. 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 8. 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 9. 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 [44].

These plots highlight the fact that most of the paths typically 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

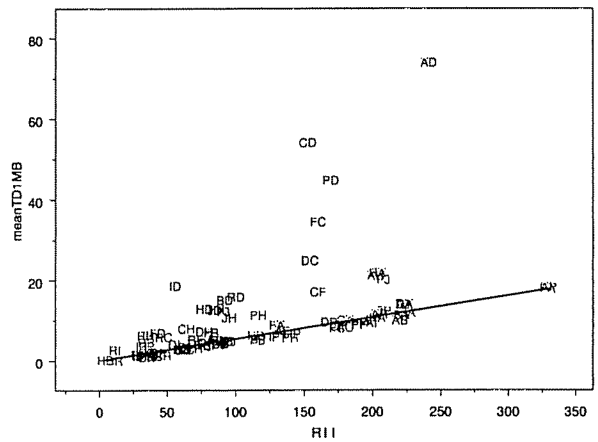

Fig. 6. Robust regression results comparing mean total delay for 1MB data transfers (meanTD1Mb in seconds) versus RTT (seconds)

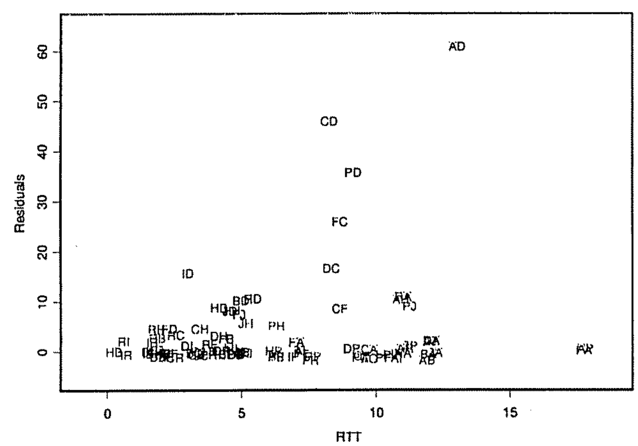

Fig. 7. Residuals (reg$residuals) for robust regression comparing mean total delay (seconds) for 1MB versus RTT (seconds)

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 10 and 11. 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 12, 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.

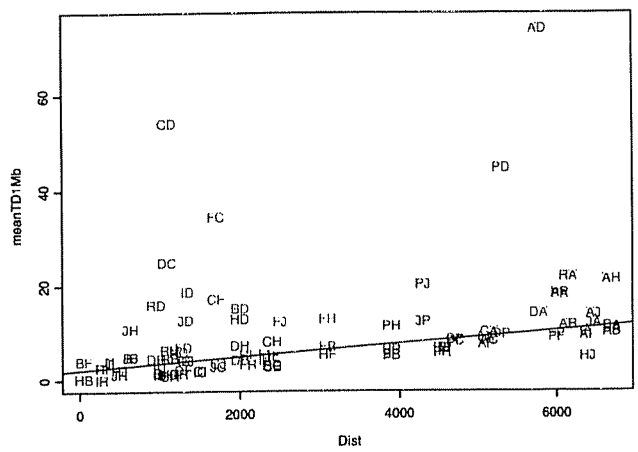

Fig. 8. Robust regression results comparing mean total delay for 1MB data transfers (meanTD1Mb in seconds) versus physical distance (miles)

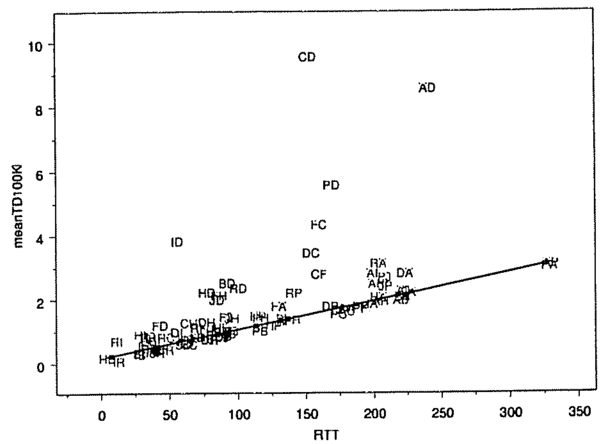

Fig. 10. Robust regression results comparing mean total delay for 100KB data transfers (meanTD100K in seconds) versus RTT (seconds)

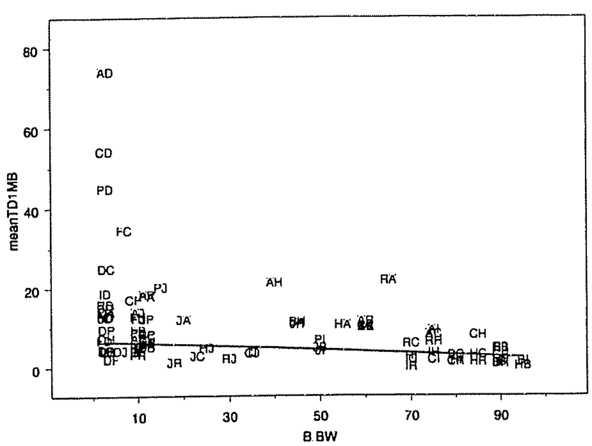

Fig. 9. Scatter plot comparing mean total delay for 1MB data transfers (meanTD1Mb in seconds) versus Bottleneck Bandwidth(Mbps)

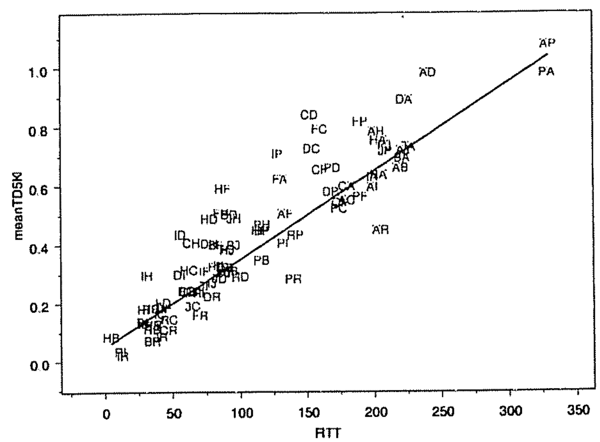

Fig. 11. Robust regression results comparing mean total delay for 5KB data transfers (meanTD5K in seconds) versus RTT (seconds)

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 13 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, not surprisingly, the same paths which typically operate far from the efficient frontier. Another approach to assessing variability is shown in Figure 14. This figure which shows the CDF of the difference between mean latency and measured latency for all 1MB data transfers along all paths. This figure indicates that only a very small percentage (about 5%) of measurements of mean transfer times vary greatly from their mean. Variability for smaller files is considerably less than the 1MB 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 15, 16 and 17. The triangle plots represent the contribution of the mean of each sub–component

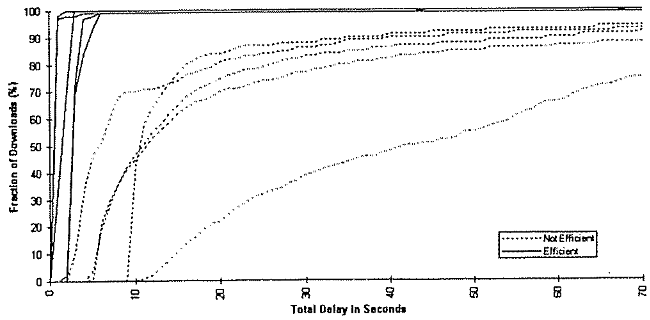

Fig. 12. Cumulative distributions of transfer times of 1MB files for 5 paths which operate on the efficient frontier and the 5 which are furthest from the efficient frontier.

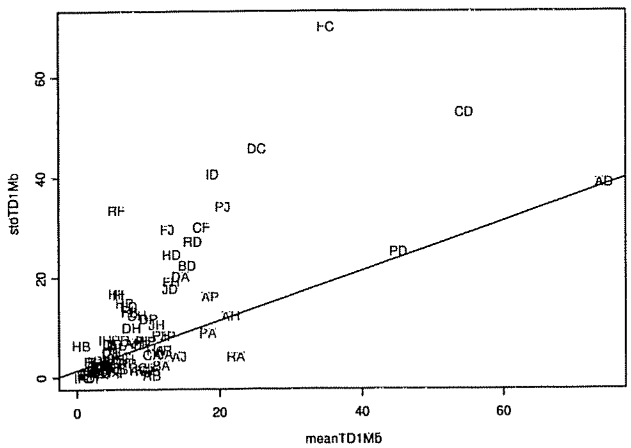

Fig. 13. Robust regression of standard deviation (stdTD1Mb) versus mean (meanTD1Mb) for 1MB data transfers

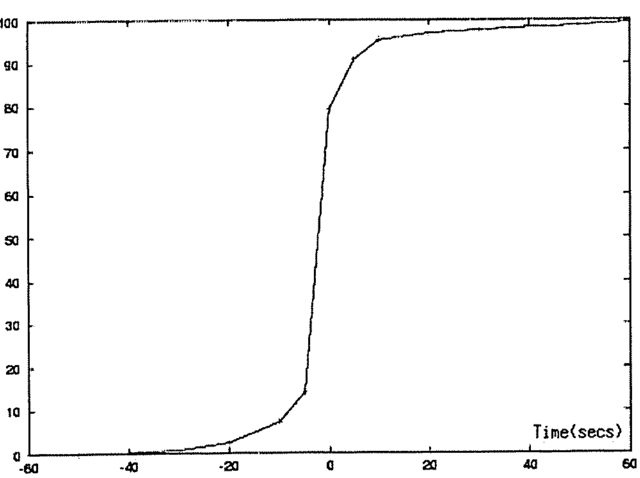

Fig. 14. Cumulative distribution of difference between mean latency and measured latency for all 1MB data transfers

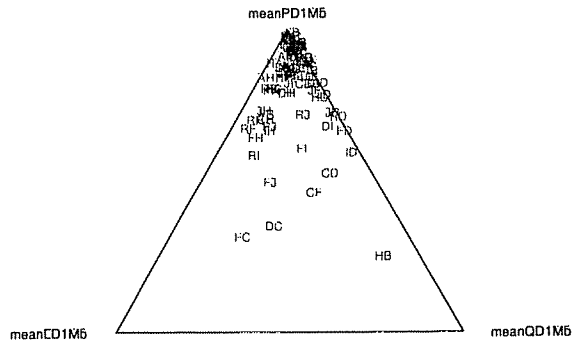

Fig. 15. Triangle plot comparing relative contribution to mean total transfer delay of propagation (meanPD1Mb), queuing (meanQD1Mb)and loss (meanLD1Mb) for 1MB file

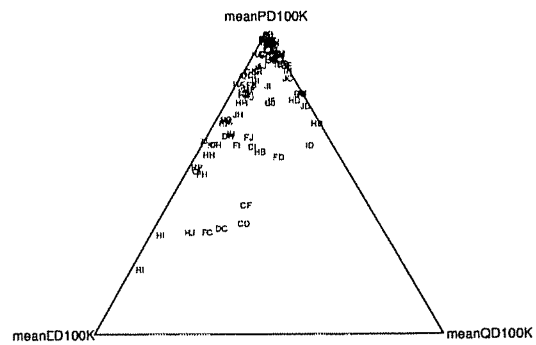

Fig. 16. Triangle plot comparing relative contribution to mean total transfer delay of propagation (meanPD100K), queuing (meanQD100K)and loss (meanLD100K) for 100KB file

normalized to the mean of the total transfer delay. If a path point is near a corner 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

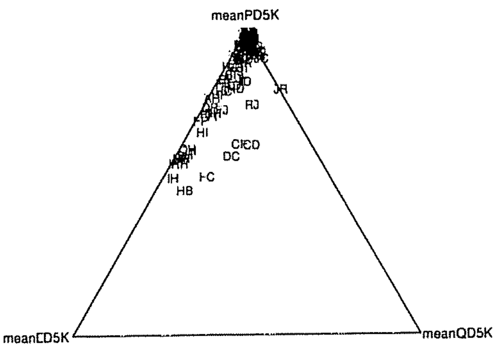

Fig. 17. Triangle plot comparing relative contribution to mean total transfer delay of propagation (meanPD5K), queuing (meanQD5K)and loss (meanLD5K) for 5KB file

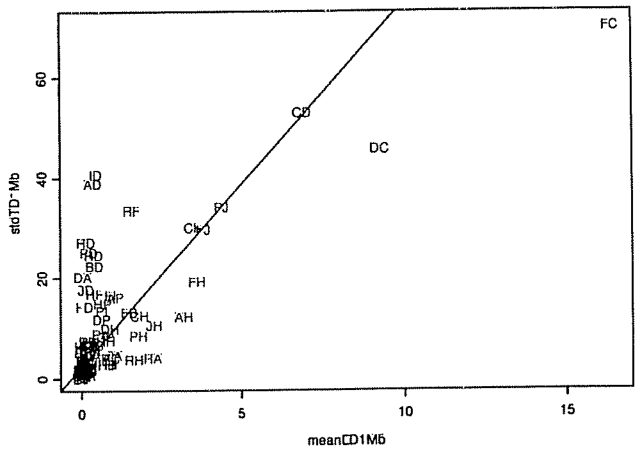

Fig. 18. Robust regression of standard deviation of total delay for 1MB file (stdTD1Mb) versus mean loss delay (meanLD1MB in seconds)

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 18 and 19. 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.

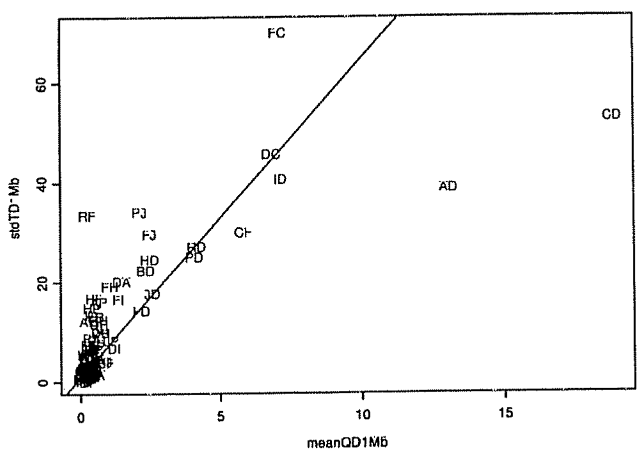

Fig. 19. Robust regression of standard deviation of mean total data transfer delay for 1MB file (stdTD1Mb) versus mean queuing delay (meanQD1MB in seconds)

D. Queuing, Loss and Path Properties

As with total delay, we investigate the use of path properties as means for distinguishing between paths that have significant queuing and loss delay. From the triangle diagrams in the prior section we can see clear discrimination between paths dominated by propagation and those that have larger queuing and loss components.

Using scatter plots, we qualitatively assess the sum of queuing and loss (to enhance differences between paths that are dominated by propagation) versus four path properties (RTT, RHops, ASHops and Dist). We find that there is clear discrimination between paths with and without queuing and loss using any of the four path properties. Figures 20 and 21 illustrate this effect using RTT and Dist. Once again, there are paths that clearly operate near an efficient frontier and a small set of outliers that do not.

E. 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

TABLE III

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

Adm	μ / v Total Delay	μ / v Propagation Delay	μ / v Queuing 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

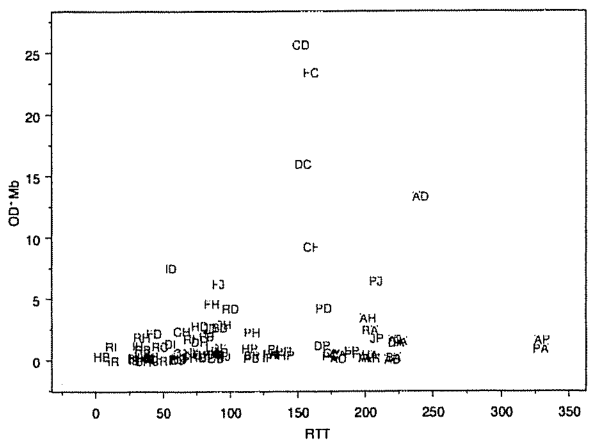

Fig. 20. Scatter plot for sum of queuing and loss delays (OD in seconds) versus RTT (seconds) for 1MB data transfers

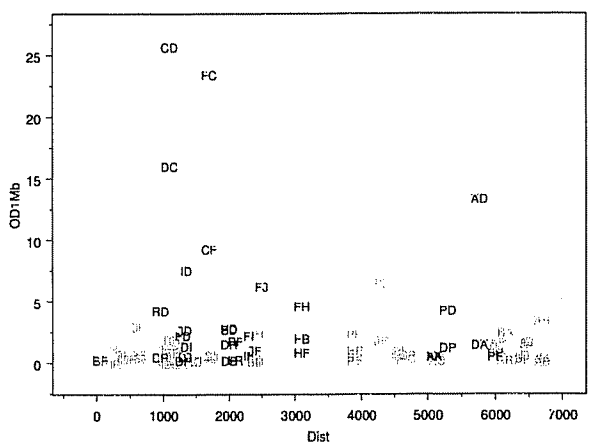

Fig. 21. Scatter plot for sum of queuing and loss delays (OD in seconds) versus physical distance (miles) for 1MB data transfers

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.

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 [34] and culminating with [12]. A recent application of the