In Proc. ACM SIGMETRICS Conference on Measurement and Modeling of ComputerSystems, June 1997.

Analyzing Stability in Wide-Area Network Performance

Hari Balakrishnan+
hari@cs.berkeley.edu

Srinivasan Seshan*
srini@watson.ibm.com

Mark Stemm+
stemm@cs.berkeley.edu

Randy H. Katz+
randy@cs.berkeley.edu

+Computer Science Division
University of California at Berkeley
Berkeley, CA 94720

**IBM T.J. Watson Research Center*
Yorktown Heights, NY 10598

Abstract

The Internet is a very large scale, complex, dynamical system that is hard to model and analyze. In this paper, we develop and analyze statistical models for the observed end-to-end network performance based on extensive packet-level traces (consisting of approximately 1.5 billion packets) collected from the primary Web site for the Atlanta Summer Olympic Games in 1996. We find that observed mean throughputs for these transfers measured over 60 million complete connections vary widely as a function of end-host location and time of day, confirming that the Internet is characterized by a large degree of heterogeneity. Despite this heterogeneity, we find (using best-fit linear regression techniques) that we can express the throughput for Web transfers to most hosts as a random variable with a log-normal distribution. Then, using observed throughput as the control parameter, we attempt to quantify the *spatial* (statistical similarity across neighboring hosts) and *temporal* (persistence over time) stability of network performance. We find that Internet hosts that are close to each other often have almost identically distributed probability distributions of throughput. We also find that throughputs to individual hosts often do not change appreciably for several minutes. Overall, these results indicate that there is promise in protocol mechanisms that cache and share network characteristics both within a single host and amongst nearby hosts.

1. Introduction

One of the fundamental philosophies upon which the Internet operates is that it is a best-effort network without any support for reservations or performance guarantees. This results in a relatively simple, scalable design of the internals of the network, and the complexities associated with robust and reliable data transfer are moved to the end hosts. To obtain good overall performance, hosts must not load the network with traffic incommensurate with what the network can support at any time. Network protocols such as TCP and applications must therefore periodically probe the network to determine the availability of resources (e.g. bandwidth) available in the network and adapt to changing network conditions. The control algorithms that govern the adaptation of protocols result in a very complex dynamical system. Understanding and modeling various aspects of Internet performance is therefore a challenging problem, and the insights gained from a detailed analysis will have important implications on protocol, application, and network systems design.

In this paper, we develop and analyze statistical models for the observed end-to-end network performance for individual hosts in the Internet. This is based on extensive and detailed packet-level traces collected from the primary Web server, run by IBM, for the 1996 Atlanta Summer Olympic Games. The traces consist of tens of millions of Web transfers and approximately 1.5 billion packets. Of particular interest to us in this work is the analysis of the *stability* of network parameters, especially the observed throughputs for data transfers. This is motivated by the fact that *probing* and *adaptation* are key components of many network protocols and applications in best-effort networks, and the observation that the effectiveness of adaptation is determined by the degree of stability in the network.

If certain network performance metrics and parameters are indeed stable, it will enable the sharing of network information amongst protocols and applications. Useful information can also be cached for future use by protocols and applications. In addition, the stability of network performance across groups of hosts that are geographically close to each other promotes the attractive possibility of sharing this information between hosts.

We analyze two forms of stability in this paper: *spatial stability*, which determines what the variation in the observed parameter is to a single host as well as to a cluster of neighboring hosts, and *temporal stability,* which determines the time scales over which various observed parameters are valid. We seek to answer the following specific questions:

1. At any time, can the throughput that an individual host observes be described by an analytic probability distribution with a high degree of confidence? If so, what is the form of such a distribution?
2. Do hosts that are close to each other in the network see similar performance for data transfers from other hosts? Typically, how large are such clusters of hosts with similar performance?
3. How does the throughput observed by a single host vary as a function of time? Over what time scales are observed throughputs relatively stable?

Our analysis yields the following main results:

- Despite the large amount of end-host heterogeneity, we find that we can often express the throughput seen by an end host as a random variable using a log-normal probability distribution. More specifically, the level of confidence is highest for a log-

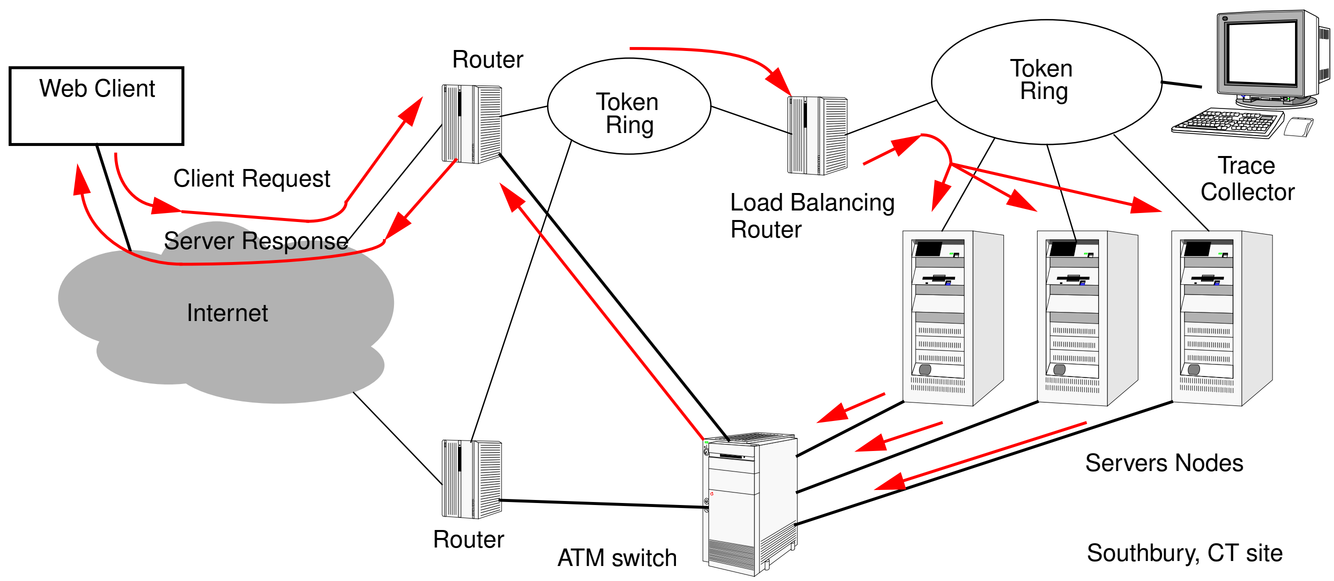

Figure 1. WWW Server and Monitoring Setup

normal distribution compared to other common analytical distributions we considered.

- We find that Internet hosts that are close to each other can often be described by nearly identical probability distributions of throughput. That is, there is no significant statistical difference between the throughputs seen by hosts that are close to each other relative to the server.
- Our analysis of a small number of hosts shows that end-to-end throughputs to hosts often vary by less than a factor of two over time scales on the order of many tens of minutes. We also find that throughputs are piecewise stationary over time scales of this magnitude.

The remainder of this paper is organized as follows. Section 2 describes the content of the traffic traces used and the basic processing we performed on the data to obtain the end-to-end network characteristics. Section 3 describes some aggregate statistics of the traffic traces and highlights some aspects of Internet heterogeneity. Section 4 describes the various statistical methods that we used to analyze these characteristics. Section 5 analyzes the typical performance to an individual client and develops an analytic statistical model for observed performance. Section 6 analyzes the relationship between the observed performance of arbitrary end hosts and their neighbors in the network to characterize the degree of spatial stability. Section 7 examines the variation of a single client's performance over time. Section 8 compares the results of this work with previous efforts in this area. We discuss some current work and future plans in Section 9 and conclude in Section 10.

2. Data Collection

This section describes the details of the trace collection and the postprocessing performed on the packet traces. We also describe the details of the network performance metrics and parameters used in the rest of this paper.

2.1 Data Collection Site and Methodology

The Web traces used in this paper come from the primary Web site for the Atlanta Olympic games (Summer 1996). Located in Southbury, CT, the Web site was maintained and run by IBM (http://www.atlanta.olympic.org). Figure 1 shows the topology of the Web site's network and the hardware used at the site. During the Olympics, the site was connected via T3 links to each of the 4 U.S. Network Access Points (NAPs), located in Chicago (Bellcore and Ameritech), the San Francisco Bay Area (Bellcore and Pacific Bell), New York (Sprint), and Washington, D.C. (MFS Datanet). In addition, there were mirror sites at Cornell (N.Y.), Keio (Japan), Karlsruhe (Germany) and Hursley (U.K.), for which we did not collect data. More details about the site structure and software used at this high-performance server are available from [22].

Requests for data from any client arrive at the Web site routed through the appropriate Internet NAP. These requests are passed to a load-balancing connection router [2, 29], that distributes them across several server nodes. These nodes retrieve the appropriate Web objects and transmit them over an internal ATM network and through the Internet to the clients.

We monitored all the traffic coming into the site and obtained packet-level traces of this traffic by running the tcpdump [18] utility on a machine placed on the token ring connecting the load-balancing router to the server nodes (see Figure 1). This machine was an IBM 150Mhz Pentium Pro PC running BSD/OS 2.1 from BSDI. We extracted and stored the first 350 bytes of every packet destined to TCP port 80 (the HTTP port), compressing this data on-the-fly using the gzip utility. The server also transmitted other data to the clients on other ports, such as streaming audio. We did not capture this data due to limited storage space and the high packet capture rates. However, this does not affect our results because the traffic to port 80 gives us a relatively consistent subsample of traffic to the different clients. Periodically, the packet trace files were dumped onto tape. Limited disk storage and tape drive speed resulted in occasional periods during which the traffic was not monitored. The gray rectangles in Figure 2 show examples of periods during which packet traces were not collected. Table 1 summarizes the aggregate details of the collected traces.

Due to the location of the machine and the topology of the server complex, we could only capture packets coming from the clients *into* the web server complex. These packets include the initial TCP SYN packet from the client, the HTTP request packets, all acknowledgments for data sent from the server to the client, and the TCP FIN or RST packet terminating the connection. In particu-

Trace Statistic	Value
Trace collection start	July 11, 5 pm
Trace collection end	Aug 6, 12 am
Packets captured	1,540,312,422
Packets dropped	7,677,854
Packet drop percentage	0.498%
Distinct client addresses	721,417
Clients reached by traceroute	314,771
Distinct hosts in topology (incl routers)	865,661
Size of packet trace	~189 GB
Number of HTTP requests captured	~60,000,000

TABLE 1. Trace Statistics

lar, this implies that we did not explicitly capture any data packets sent from the server to the client. However, this does not prevent us from reconstructing the events that happened at the server and when packets were transmitted as a function of time, as explained below.

The lack of data packets does not prevent us from determining the packet sequence trace as a function of time, since we can estimate this from the sequence of TCP acknowledgments, and in the absence of losses, predict the times at which the TCP sender transmits packets. In order to accurately reconstruct sender events during loss recovery, we modified the TCP stack on a subset of the servers to transmit the TCP/IP header (together with information on the current sender congestion window size, smoothed round-trip time estimate, and number of duplicate acknowledgments causing the retransmission) of any retransmitted packet on the token ring interface to our collection machine. This retransmission information combined with the packet traces and knowledge of the TCP algorithms at the server allow us to reconstruct the outgoing data stream with high accuracy.

Analyzing the spatial stability of observed performance requires a detailed knowledge of the network topology to each host. Over several weeks following the Olympics, we ran the traceroute [28] utility from the server cluster to each client IP address that appeared in our packet traces. This allows us to record the path that server-side data took from the server to each client in our trace. When the traceroutes were being run, the site was connected to the Internet by a single T3 to the NY Sprint NAP. Clearly, the change in topology and the delay before traceroute was performed could cause the paths recorded by traceroute to be somewhat different from the routes taken by packets during the Olympics [25]. However, the path information obtained still provides a great deal of information about the locality and distribution of the clients on the Internet. Altogether, 44% of the end hosts were reached by the traceroute runs in the subsequent weeks after the Olympics. We believe that the bulk of the unreached hosts may have been temporarily disconnected, behind firewalls, or have been localized network problems.

2.2 Performance Metrics and Parameters

The raw packet traces were post-processed in a variety of ways to calculate various such as connection throughput, connection lengths, request and response sizes, etc. We started our analysis by measuring throughput as the average bandwidth over a single TCP connection from a single client. However, a preliminary analysis of connection behavior showed that retransmissions caused by timeouts added significant randomness to the connection durations and resulting bandwidths, because of the coarseness of the retransmission timer. We found that over 50% of all retransmissions were caused by coarse timeouts rather than by fast retransmissions. Because we were more interested in obtaining an accurate view of the available network bandwidth without being influenced by the specific details of the TCP retransmission timer (which is necessarily conservative), we excluded periods where all of a host's TCP connections were undergoing a retransmission timeout. We did include the time consumed by the fast retransmission and fast recovery mechanisms while obtaining each throughput sample, since this did not result in connection stalls.

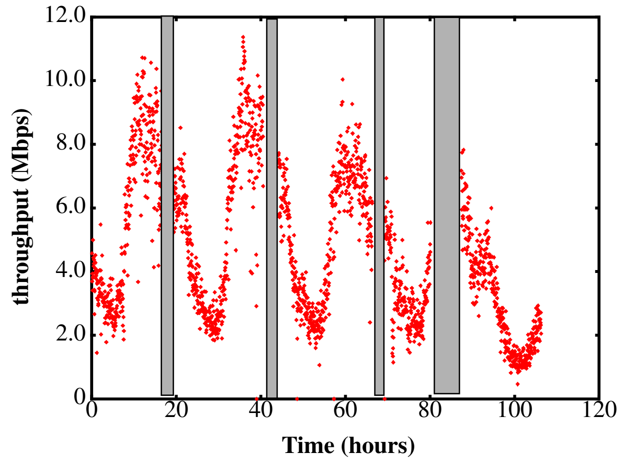

Figure 2. Aggregate throughput leaving the site on port 80 starting at time 12.15 am, EDT, on July 30, 1996.

Another complication arose from the use of multiple concurrent TCP connections by many current Web browsers. We were interested in measuring the aggregate throughput to a client host rather than the throughput across an individual connection, so we aggregated throughput measurements across parallel connections from the same IP address.

The combination of these two considerations leads to the following definition of *throughput*: it is the ratio of the number of bytes transmitted to the duration of transmission, across all active parallel TCP connections from a particular client, excluding intervals when all connections are stalled due to TCP timeouts or when no connections are active from a particular client. Each stall-free interval is used as a single throughput sample. Finally, we performed a log-transformation (base 2) on the throughput data (as described in [23]) to reduce the effect that outliers have on the sample data. All of the analysis is performed in the log domain.

We then used the statistical techniques described in Section 4 to compare the measured samples with various analytical distributions and to compare hosts within a cluster to see if their throughput samples were identically distributed.

3. Aggregate Traffic Statistics

In this section, we first present the details of the trace data we collected. We then discuss some aspects of variability in the Internet, focusing on variations in end-host TCP behavior as well as on observed throughputs for data transfers to different hosts.

Aggregate statistics: Figure 2 shows the throughput of web traffic on port 80 for 106 hours beginning at 12.15 am, EDT, on July 30, 1996. The gaps in the graph correspond to periods during which

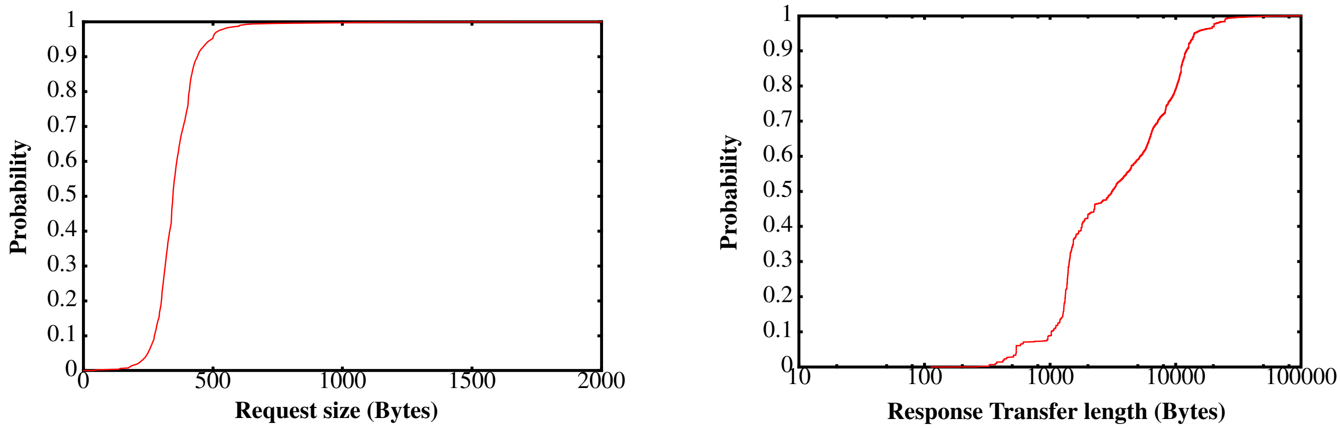

Figure 3. Cumulative Distribution Functions of request and response transfer lengths (on a log scale) a over the 106 hour period described in Figure 2.

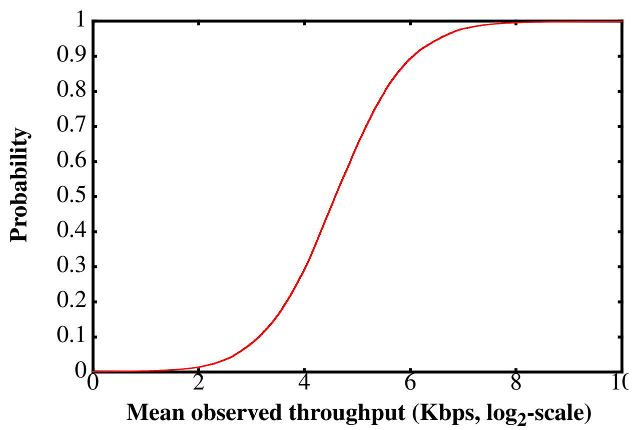

Figure 4. (a) CDF of mean throughputs from the server to 30,000 random hosts over a three hour period. The mean observed throughput varies between 6 and 1000 Kbps.

packet traces were not collected. During the collection period, 189 GB of outgoing data were transferred at an average rate of almost 4 Mbps. The peak throughput sustained over a one second duration from this port was 14.2 Mbps. The distribution of request packet lengths (i.e. the GET request from the client) and response transfer lengths for the same 120 hour period are shown in Figure 3. The mean, median and standard deviation of request sizes are 355, 345, and 94 bytes respectively., and the mean, median, and standard deviation of response sizes are 5543, 3114, and 6203 bytes, respectively.

Heterogeneity and variety: Figure 4

The second aspect of Internet heterogeneity that reveals itself is the end-to-end throughput for data transfers. Figure 4 shows the CDF of mean throughputs observed for transfers to 30000 randomly chosen hosts over one representative three hour period, on a log$_2$-scale. As described previously, performing a log-transformation before analyzing the data eliminates the problems caused by outliers. For this sample dataset measured between 7am and 12pm on a weekday, the observed mean throughput ranged between 6 and 1000 Kbps, with a mean of 44 Kbps and a median of 34 Kbps. This variation confirms that end-to-end network performance varies widely as a function of end-host location, and also emphasizes that the Web server was not the bottleneck in the overall end-to-end performance for most hosts.

The information presented in this section highlights the magnitude of our data set. To our knowledge, this is the largest Web packet trace collected to date. This section also highlights the degree of throughput variation observed for typical Internet transfers. Thus, our challenge is to apply statistics to process this large collection of traces and discern patterns in the probabilistic distributions of throughputs, and explore the ideas of spatial and temporal throughput stability. Sections 5 through 7 develop these ideas further. In the next section, we present some statistical machinery and notation that facilitates this process.

4. Statistical Methodology and Notation

In this section, we present an overview of the different statistical techniques used in the analysis of the data we obtained. We start with some definitions and notation.

4.1 Terminology

We define B_d as the random variable that describes the throughput seen by host d over a short time scale. Individual hosts may be aggregated together into *clusters* of size k where k is the maximum distance (in network hops along the graph formed from the traceroute information in Section 2.1) from any host to any other host. A host may simultaneously be in multiple clusters at a single time for different values of k. For example, a host is usually in a cluster of size 2 which corresponds to its local subnet as well as in clusters of sizes k=*4..6* (which could correspond to an administrative domain).

4.2 Choice of Statistical Method

Recall that two of our main goals are to determine the statistical distributions of throughputs to different sites and to determine if "nearby" hosts can be aggregated together as having identical network performance.

These goals require us (i) to compare and fit a given set of measured samples from a distribution to an appropriate theoretical one in order to characterize the distribution of throughputs, and (ii) to compare two sets of measured samples and decide if the two random variates corresponding to each distribution are identically distributed.

There are several different methods that we considered using to answer these two questions: simple histograms, Chi-square tests, Kolmogorov-Smirnov tests, and quantile-quantile plots ([9], [14] provide excellent descriptions of these techniques). We now describe our reasoning in choosing the method we used.

A histogram is a discretized representation of the PDF using bins, and allows us to graphically represent the distribution of the measured and analytic random variables. The disadvantage of this approach is that it does not lend itself to automatically making millions of comparisons, which made it impossible to use.

A Chi-squared test is an attempt to automate this process by comparing the observed frequency of a measured variable with those obtained from an analytical distribution. One disadvantage of this approach is that the test works well only when the number of items that falls into any particular cell is approximately the same. However, it is relatively difficult to determine the correct cell widths in advance for different measured data sets. Another disadvantage of this approach is that it works well when the number of measured samples is large, as the mean and variance of the hypothesized distribution must be specified in advance. This may be difficult to do accurately for data sets with small number of samples. Many of our measurements consisted of relatively small numbers of samples, and we believed that excluding these samples could bias our data. Moreover, the test requires that the data be divided into specific bins. While this may be appropriate with discrete data which can take on only a small number of values, it is at best an arbitrary process when the values come from a continuous distribution. Since the results of the chi-square test can vary with how the data are divided, this test is not a good alternative when dealing with continuous population distributions.

A Kolmogorov-Smirnov test compares a measured distribution and an analytical distribution by finding the maximum differences between the two variables' cumulative distribution functions. We decided to not use this approach because the Kolmogorov-Smirinov test works best when the number of measured samples is small, and we had many hosts with thousands of throughput samples.

A quantile-quantile plot compares two distributions by plotting the inverse of the cumulative distribution function $F^{-1}(x)$ for two variables and determining the best-fit line (as determined by the minimum mean square error) of the resulting scatter plot. If the coefficient of determination (defined as $R^2 = (1 - SSE/SST)$, where SSE is the mean squared error with respect to the line and SST is the mean squared error with respect to the mean) is high, then the two distributions have the same shape with possibly different parameters (e.g., both random variables are Normal). Furthermore, if the slope of the resulting line is close to unity and the y-intercept of the resulting line is close to 0, then the variables are almost identically distributed (e.g., both random variables are $N(0,1)$) [14]. We decided to use quantile-quantile plots as our primary method in comparing distributions with analytic models and other measured distributions. We chose this method because it easily lends itself to answering parts (i) and (ii) with a single operation, and works well on a variety of sample sizes. We chose to use decile-decile plots (10 quantiles per random variable) in making comparisons between variables. Figure 5 and Figure 6 show this process in more detail. Figure 5 shows the cumulative distribution function of the measured throughput for a single host and the cumulative distribution function of a normal distribution. From these CDFs, we find the values of the inverse of the CDF at values 0.1, ..., 0.9 and plot this parametric function (x(prob),y(prob)) on a single graph. Figure 6 shows this parametric plot and the corresponding best fit line. The error between the parametric curve and the best-fit line is small, which leads to a R^2 value that is close to one. This means that this host's log(throughput) is well-described using a normal distribution. Examining the slope of the best-fit line, we find that the slope is not close to 1, implying that the parameters of this host's normal distribution are different than the parameters of the analytic distribution.

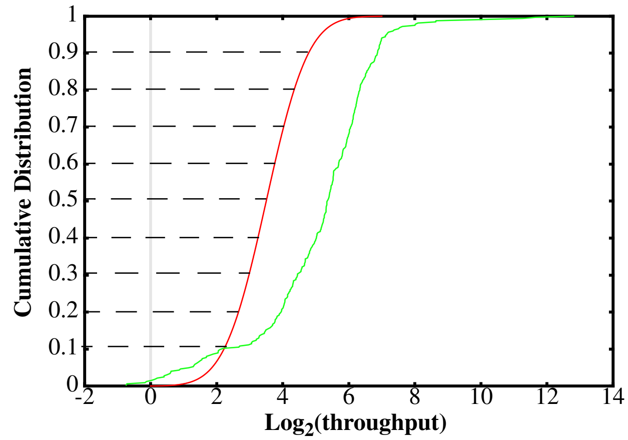

Figure 5. Comparison of analytical and empirical measurements: cdf vs. cdf

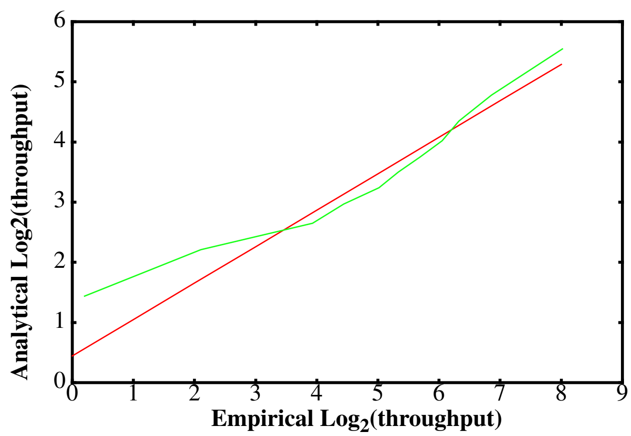

Figure 6. Comparison of analytic and empirical measurements: Q-Q plot.

When comparing a measured variable to an analytical distribution, it is useful to quantify our confidence that the analytic form describes the measured samples. Although the quantile-quantile plot acts as a good *metric* in determining the effectiveness of a fit, it does not allow us to directly express our *confidence* in the goodness of a fit. To determine the degree of confidence, we used the Shapiro-Wilk test [9] to quantify the effectiveness of the fit for a normal distribution. We define the null hypothesis:

H0: the measured samples X1, ..., XN come from a given analytic distribution.

Given the coefficient of determination R^2, we define $Z = n(1 - R^2)$ and use Z, the number of samples, and a level of confidence α to determine whether we must reject the null hypothesis $H0$. By comparing our confidence level in rejecting the null hypothesis for various analytical distributions, we can compare the effectiveness of these distributions to determine which one best fits our measured data. α–levels of significance for these distributions are given in [9].

It is important to notice that it can be difficult to state with high confidence that a particular analytical distribution fits a measured data set. There are potentially an infinite number of curves that we can fit through a finite number of measured data points. As a result, a positive result comes about not by accepting $H0$ but by failing to reject $H0$ with any level of confidence. This is not a major prob-

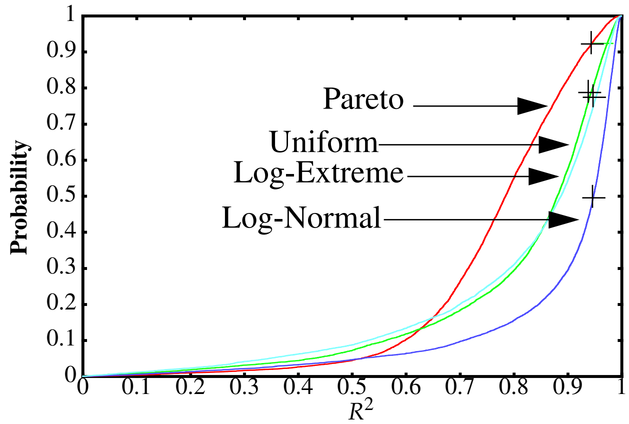

Figure 7. CDF of the values of R^2 for the four compared distributions. The "+" marks show the $\alpha = 0.5$ levels of significance.

lem, however, because we are more interested in the general shape of the curve (normal vs. exponential vs. uniform) rather than showing that a particular analytical model exactly fits our measured data sets.

5. Analyzing and Modeling Individual Host Performance

In this section, we attempt to determine how the random variable B_d behaves for all hosts in the study. We first discuss how we use the statistical methods outlined in Section 4. We then present the results of the analysis and some implications of these results.

5.1 Methodology

Our goal is to determine if the random variable B_d can be characterized by some well-known analytic model. The specific choice of the parameters in the model (for example, the mean and standard deviation for a Gaussian distribution, or the parameters α and β for an Extreme distribution) will obviously vary from host to host — our objective is to determine if B_d follows some consistent "shape" or distribution family for all destinations, d. In order to prevent anomalous samples from skewing our results, we only include hosts with at least 40 samples.

We compared the throughputs in the log domain against four analytic distributions: (i) uniform between the minimum and maximum observations, (ii) normal, (iii) extreme, and (iv) exponential. This corresponds to the original throughputs having distributions (i) uniform, (ii) log-normal, (iii) log-extreme, and (iv) Pareto, parameter 2, respectively.

5.2 Results

We compared measured throughput over several three-hour periods in our trace with the analytical distributions. Figure 7 shows the CDF of the values of R^2 for the four distributions from one of these 3-hour periods. Analysis of the other three-hour periods produced virtually identical results. There are four curves, one for each analytical distribution we measured. Also included for each curve is a "+" mark that shows at the $\alpha = 0.5$ level of significance whether or not the hypothesis H0 can be rejected. The greater that the curve is skewed towards a R^2 of 1, the better that the analytical distribution fits the measured throughputs. For example, the curve labeled "log-normal" indicates that approximately 75 percent of the hosts have a R^2 of 0.9 or greater. We notice that of the four distributions, the log-normal distribution is the most skewed towards high values of R^2. This means that of the distributions we considered, a log-normal distribution does best at modeling an individual host's throughput probability distribution. We also see that H_0 cannot be rejected at $\alpha = 0.5$ (the maximum possible from this test) for about 50% of the observed samples. That means that for 50% of the samples, it is impossible to say that the measured samples are not modeled well by a log-normal distribution. In contrast, the corresponding percentages for the other distributions are 8% (Pareto), 22% (uniform), and 21% (log-extreme). Similar conclusions apply at other levels of significance ranging from 0.25 to 0.01. This analysis suggests that the observed throughputs to many hosts can be analytically modeled as a log-normal distribution.

6. Spatial Stability

In this section, we investigate the degree to which nearby hosts in a cluster see similar throughputs.

6.1 Methodology

To determine the similarity of measured throughputs between nearby hosts, we examined a section of the trace covering 2 successive 3-hour time periods. For each period, we determined if hosts in the same cluster saw similar throughputs from this server. Using the traceroute information described in Section 2, we found all clusters of size *k=2..6* that had at least 3 members. This corresponds to hosts connected to a common router 1, 2, and 3 hops away, respectively. There were approximately 20,000 clusters of size 2, 10,000 clusters of size 4, and 4000 clusters of size 6. The actual number of clusters considered differed from time period to time period. Many clusters did not contain enough hosts with enough samples to make meaningful decile-decile comparisons, so the number of clusters actually considered is smaller than the number of possible clusters. We then used the decile-decile methods described in Section 4 to compare the throughput distribution B_d for each cluster member with each other cluster member. By examining the residual error, slope, and intercept of the best fit line, we can determine whether or not the random variables B_d for that pair of hosts were identically distributed.

To quantitatively show the degree to which nearby hosts share network performance, we define the hypotheses:

HP: The specified pair of hosts have "almost" identical distributions, and

HC: Hosts in the specified cluster have almost identical distributions,

and determine how many pairs of hosts per cluster satisfy *HP* and how many clusters satisfy *HC*. In order for *HP* to hold for a pair of hosts, we require the value of R^2 for the best fit line to be greater than 0.85, the slope of the best fit line to be between 0.6 and 1.4, and for the 90% confidence interval for intercept of the best line to include 0. For *HC* to hold for a specific cluster, the number of host pairs in the cluster where *HP* holds must exceed the number of host pairs where *HP* does not hold.

In order to present an example of how the hypotheses *HC* and *HP* are satisfied, Figure 8 shows the cumulative distribution function of best fit line slopes for a sample cluster of size 2. The vertical lines indicate the minimum and maximum slopes that satisfy *HP*. Notice that the number of pairs between the two lines is much more than one-half the total number of pairs. Therefore, this cluster has slopes that satisfy *HC*. If we also find that more than one-half of the pairs have high values of R^2 and best-fit line intercepts that are close to 0,then we can say that this cluster satisfies *HC*.

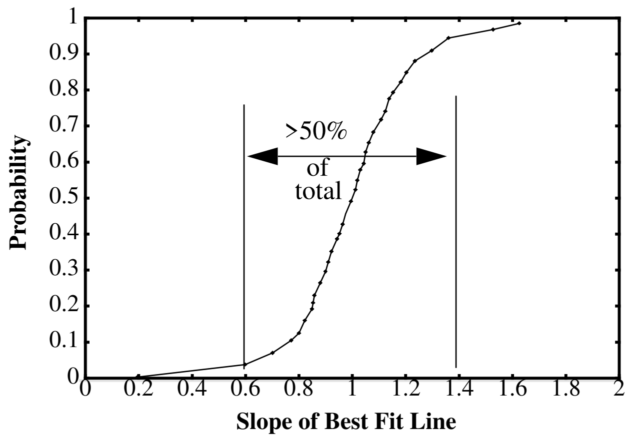

Figure 8. (a) Sample cluster that Satisfies HC

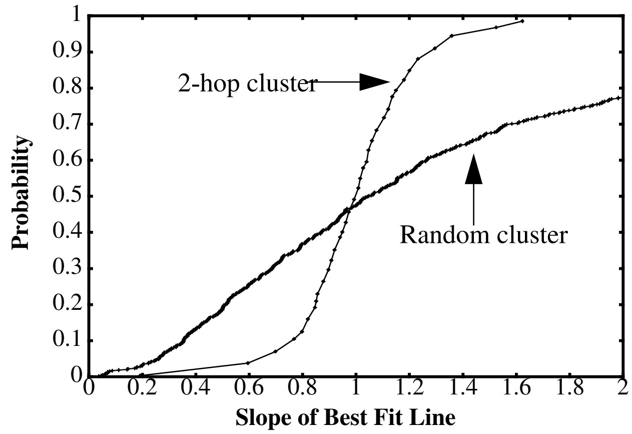

Figure 9. Random cluster vs. 2-hop cluster

Cluster Size	Total # clusters/ clusters considered	Total # HP Satisfied/ Not Satisfied	Total # HC Satisfied/ Not Satisfied
2	20047/2003	15773/ 10063	1238/765
4	9844/3502	109337/ 81028	1941/1462
6	4213/3074	359679/ 185781	1984/1051

TABLE 2. Summary of Clustering Results

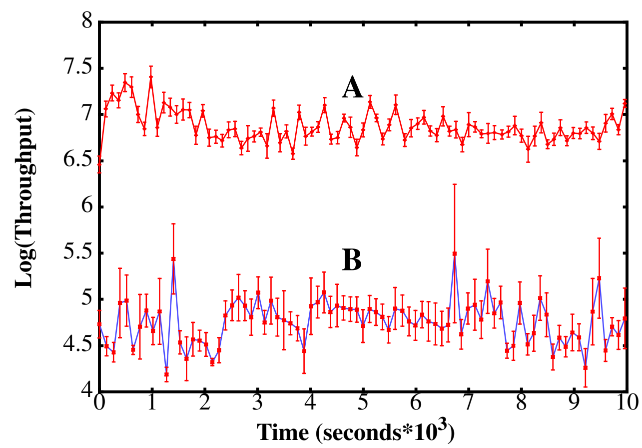

Figure 10. Time series plot of throughputs for two "stable" hosts on a log scale.

6.2 Results

Table 2 shows the number of clusters that satisfied *HC* for each cluster size $k=(2..6)$. The fact that *HC* is satisfied for a majority of clusters indicates that nearby hosts often observe similar or identical throughputs. That *HC* is not satisfied for all clusters is not unexpected — not all clusters consist of hosts on the same network subnet. For example, two nearby routers acting as WWW proxies may be geographically close to each other, but each acts as a representative for a large group of heterogeneous hosts. Also, we note that less than half of the clusters of size 6 satisfied *HC*. This is not surprising, since as the cluster size increases, the probability that hosts with different bottleneck links will be aggregated together into a single cluster also increases, leading to a false attempt at clustering.

To quantify this effect of "false clustering", we chose a set of hosts at random and treated them as a single cluster. This cluster is designed to indicate the worst-case scenario of clustering geographically separated hosts and mistakenly trying to correlate their performance. Not unexpectedly, the random cluster did not satisfy the hypothesis *HC*, as only 86 host pairs satisfied *HP* while 177 pairs failed to satisfy *HP*.

Figure 9 compares this "random" cluster against the sample cluster of size 2 by showing the cumulative distribution functions of slopes for each cluster. We can notice from the figure that the 2-hop cluster has more values close to 1 than the randomly selected cluster, which means that more hosts in the 2-hop cluster have identical throughput characteristics than in the random cluster. However, the distribution of slopes for the random cluster is not completely uniform. This is not unexpected, because geographically separated hosts that are connected via similarly constrained bottleneck links will have B_d distributions that are identical. One example of this phenomenon is ISDN or modem users who are limited by their direct connection to the Internet. Most of these users will have identical throughput distributions B_d.

7. Temporal Stability

In Section 6, we analyzed the degree of spatial stability of observed throughputs to end-hosts and demonstrated that nearby hosts often display the same statistical behavior. We now turn to the analysis of stability of this performance metric over time.

7.1 Two Notions of Temporal Stability

We distinguish between and consider two different notions of temporal stability — *wide-sense stationarity* and *persistence*. A stochastic process $X(t)$ is wide-sense stationary (WSS) if $E[X(t)]$ and $E[X^2(t)]$ are independent of t[1]. Since we have samples of through-

1. We also need the condition that $E[X(t)X(u)]$ depends only on $(u-t)$, but we don't consider this further in this paper.

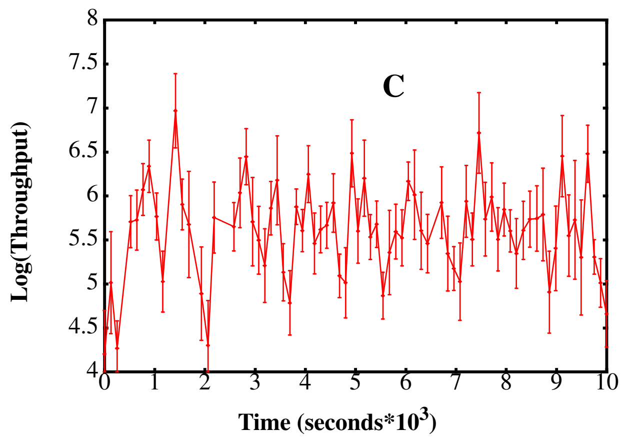

Figure 11. Time series plot of throughput for an "unstable" host on a log scale

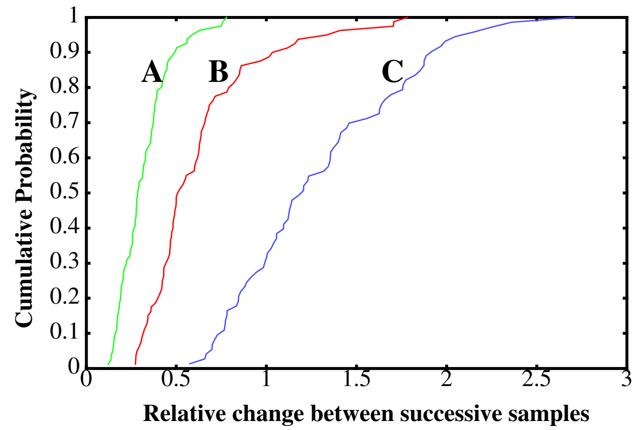

Figure 12. Cumulative distribution functions of differences between successive samples

put, $B(t)$, at discrete instants of time, we consider a sequence of random variables B_1, B_2, ..., B_n drawn from $B(t)$ in order to determine the time scales over which the throughput stochastic process is WSS. In general, $B(t)$ is not WSS over arbitrary time scales, although it is *piecewise WSS* over shorter time scales. Our goal is to understand and quantify this behavior.

In addition to understanding the degree of stationarity[2] in the underlying process, we are also interested in determining the amount by which successive throughput samples change. This is because abrupt changes over short time scales can occur even when $B(t)$ is stationary. For example, the expected values of $B(t)$ could be equal over a time interval T, but the standard deviation of $B(t)$ could span multiple orders of magnitude, making it hard for protocols and applications to adapt to varying conditions in a timely manner. Thus, we define the notion of persistence, which quantifies the degree of change in successive samples.

For example, a random process $X(t)$ with expectation 50 units and standard deviation 50 units for all t is stationary, but is not persistent stable because successive samples could vary by as much as 100 units, with non-trivial probability. These two notions of stability are elaborated upon and quantified in the remainder of this section.

7.2 Methodology

We focus on a 3-hour segment of the trace in order to perform this analysis. For each host, we take successive 2-minute intervals and obtain throughput samples for transfers to the host. These samples are measured the same way as in the previous sections, with care taken to treat coarse retransmission timeouts and concurrent TCP connections. Thus, we obtain a time series plot of throughput samples (in Kbps). These are shown in Figures 10 and 11, where throughputs are plotted as a function of time on a log-scale. Figure 10 shows two examples of hosts with "stable" performance (hosts A and B), while Figure 11 shows an example of a host (host C) with more varying performance. Each sample on the curves corresponds to instances where there were at least 10 different samples in the 10-minute period, and the corresponding errorbars are shown as well. Host A was 14 hops away from the Web server with a round-trip delay of 130 ms. Host B was 20 hops away with the same round-trip delay. Host C was unreachable by our subsequent traceroute probe.

For each sample point x, define $t(x)$ to be the minimum time until a later sample with a completely non-overlapping errorbar. If there is no overlap between errorbars of two samples separated by time τ, then it implies that there is a strong probability that the underlying process has changed and no longer has the same mean, which yields an upper bound on the duration of stationarity. Our analysis of 10 hosts that had samples over most of the three-hour period showed that the distribution of τ was largely independent of the host, even though (as shown in Figures 10 and 11) the time series plots of the different hosts are very different. Thus, it is possible that the throughputs observed by end-hosts are piecewise stationary over time scales on the order of many hundred minutes (i.e., we cannot conclusively show otherwise). This makes it possible to use techniques based on empirical autocorrelation functions to analyze the time-dependence of throughput samples over the time scales over which throughputs are (with high probability) stationary — we are investigating this in ongoing work.

We now consider the notion of persistence, which is a measure of the time duration between changes in observed throughputs. For each pair of successive samples, we quantify the "maximum" likely change in throughputs by finding the difference between the higher error bar of one sample and the lower error bar of the other (or vice versa). We compute the CDF of the amount of change between successive samples and analyze this. This is useful in many practical situations, where protocols and applications are tuned to adapt to and certain maximum variations in network conditions.

7.3 Results

Figure 12 shows the CDF of persistence for the three hosts corresponding to the three hosts whose time series plots are shown in Figures 10 and 11. We find that successive samples for one of the two hosts marked "stable" (host A) vary by less than a factor of 2 over the entire trace period. Over 90% of the successive throughput samples for host B vary by less than a factor of 2. This, coupled with the earlier observation of long-term stationarity indicates that these hosts show a significant amount of temporal persistence.

In contrast, for the host marked "unstable" (host C), about two-thirds of the successive samples differed by more than a factor of two. Thus, this host shows significant short term variations in throughput.

2. Unless otherwise specified, we use "stationary" to mean "wide-sense stationary", according to our definition of the term.

Obviously, one cannot generalize these results to make general statements about the stability of Internet performance, but it is a start. In particular, the fact that performance to certain hosts does not widely vary is promising and this bodes well for measurement-based adaptive approaches to protocols and applications. We intend to complete this analysis and obtain a much better understanding of temporal stability and the implications of these results on protocol design in current future work.

8. Previous Work

Previous work in the area of end-to-end performance characterization and probing have fallen into two major categories (i) passive characterization of wide area traffic patterns and (ii) development of new techniques to probe for network resources.

8.1 Network Characterization

Past work in network characterization has taken traces from web servers, backbone networks and local area networks and developed traffic models from them. Characteristics that have been examined include interarrival patterns, access patterns and transfer size distributions. [20] examines traffic between a busy web server and its clients. In addition, Mogul performed several traceroute, ping and bandwidth tests between the clients and server, in an attempt to correlate these network-level events with receiver bandwidths. [3] examines the workload of several Web servers and identifies several invariant characteristics. These characteristics include mean transfer size (< 21KB), file size distribution (Pareto), inter-reference arrival distribution (exponential distributed and independent), access locality (10% of the domains account for >75% traffic). In contrast to these server-centric traces, [17] captured client-centric WWW traces and characterized such client specific information as the distribution of user think and busy times, the number of items retrieved per web page, the number of web pages retrieved per web browsing "session", and other user specific browsing behavior. [13] measured the packet inter-arrivals patterns for connections on the NSFNET national backbone, showing that the packet arrivals followed a "packet train" rather than a Poisson arrival pattern [16] first showed that network traffic exhibits *self-similar* behavior, displaying identical bursty statistical properties at a variety of time scales. [8] presents evidence indicating the self-similarity of WWW traffic, and explains this behavior by examining the distribution of WWW document sizes, the effects of caching, and the behavior and aggregation of individual clients. [5] analyzes transfer sizes for a variety of applications and uses this data to generate models to synthesize traffic for individual connections. [10] and [11] present empirical models for modeling connection and session arrivals as well as generating artificial workloads for common applications such as TELNET and FTP. [27] focused on FTP connections on the main gateway to the University of Colorado network, measuring packet type (directory query vs file transfer packet) and transfer sizes. [23] examines wide-area traffic traces and derives analytic models that describe the number of bytes sent and received for various types of connections such as ftp, telnet, and nntp. In [26], the author examines connection and session arrival processes for applications such as telnet and ftp and shows that Poisson processes sometimes greatly underestimate the traffic burstiness for these applications.

8.2 Network Probing

Work in network probing algorithms have concentrated on obtaining end-to-end performance estimates more quickly and accurately.

[19] examines the relationship between the instantaneous bandwidth indicated by TCP acknowledgments and the actual bandwidth available to the TCP connection. He shows that on the Internet, TCP acks often significantly overestimates the available bandwidth (known as *ack compression*). In addition, he shows that ack compression can be correlated with the loss of subsequent packets immediately after the ack compression. Packet pair [15] is a technique in which two packets are injected into the network back-to-back. The destination for these packets reflects these packets back to the source. Assuming that the network routers perform fair queuing, the time that separates the packets upon reception at the source gives a good indication of the network congestion along the path. [7] describes a mechanism for estimating network-level performance between hosts in the Internet. The authors describe two tools used to measure network-level performance between clients and servers. *Bprobes* are packet pairs designed to measure the bandwidth of the bottleneck link between two hosts, and *cprobes* are designed to measure the amount of congestion on the path between hosts. Earlier work, [1] and [12], used periodic UDP probe packets to measure transit time and loss patterns for Internet connections between Texas and Maryland. [4] performed similar experiments for the connection between France and Maryland for time scales varying from a few milliseconds to a few minutes.

9. Future Work

TCP Performance Analysis and improvements: We described in Section 2 that we were only able to capture packet acknowledgments coming from clients and not server-side data packets. Unfortunately, this makes it difficult to examine how well current TCP implementations perform congestion control and loss recovery. We are currently implementing a TCP sender-side emulator that takes as input a sequence of acknowledgments and can reconstruct the sender-side TCP state with high accuracy. With this emulator, we plan to quantify the limitations of TCP sender loss recovery and congestion control algorithms in the presence of typical Web transfers (short simultaneous connections from individual hosts). We also plan to investigate several solutions to these limitations, including new loss recovery algorithms and an integrated approach to congestion control and loss recovery.

Caching of Network Parameters: We have shown in Sections 6 and 7 that nearby hosts experience similar or identical throughput performance within a time period measured in minutes. This behavior leads to the idea of *caching* network parameters for a single host across connections as well as across multiple nearby hosts. This could be done at the several levels in the networking stack. At the transport layer, we can exploit the temporal and spatial locality in TCP connections and share useful information across the protocol control blocks of individual connections. At the application layer, we are examining the idea of a "performance database" where clients can query and report on the throughput to distant network clouds.

Our system consists of clients with modified network stacks that perform periodic measurements of network parameters independent of application, or connection. These clients make periodic *performance reports* (similar to RTP receiver reports) to *aggregation servers* that synthesize and combine the performance reports from individual clients and distill the reports into single performance metrics with appropriate confidence intervals. Clients also communicate with *performance servers* when contacting new hosts (for example, when performing a Domain Name System lookup from name to IP address). These performance servers determine the appropriate cluster for the distant and nearby host and return the appropriate aggregated performance metric to the client. We intend to design and implement a prototype system based on these ideas in the CAIRN [6] wide area testbed.

10. Conclusions

In this paper, we have presented a statistical analysis of throughputs of Web transfers. This analysis was performed using a large packet-level trace of traffic at the Web site for the Atlanta Summer Olympic Games. Observed mean throughputs for these transfers vary widely as a function of end-host location and time of day, confirming that a large amount of heterogeneity exists in the Internet. Our analysis showed that despite these wide variations the transfer throughput exhibited significant temporal and spatial stability. In particular, we found that:

1. Throughputs to several individual internet hosts can be modeled using a log-normal distribution regardless of location. More precisely, we found that for over 50% of the hosts in our study, the log-normal hypothesis could not be rejected at any level of significance, and that it characterized observed throughputs better than the other analytic distributions we considered.
2. Nearby Internet hosts often have almost identical probability distributions of observed throughput. The size of the clusters for which performance is identical varies as a function location on the Internet. However, our analyses show that cluster sizes in the range of 2-4 hops work for many regions.
3. Our analysis of a small number of hosts shows that end-to-end throughputs to hosts often vary by less than a factor of two over time scales on the order of many tens of minutes. We also find that throughputs are piecewise stationary over time scales of this magnitude.

Overall, these traffic characteristics indicate that there is great promise in techniques that cache and share such network characteristics amongst nearby hosts as well as within a single host across connections.

11. Acknowledgments

We are extremely grateful to Sean Martin, Ron Woan, Frank Schwithenberg, and others who did an admirable job of running the Atlanta Olympics Web server at the site in Southbury, CT, for their support and help. They provided patient help and advice while under enormous pressures in getting the Web site up and running. We also thank Arvind Krishna and Hamid Ahmadi for their support and encouragement, as well as comments on earlier drafts of this paper. We would also like to thank the anonymous SIGMETRICS reviewers for their detailed comments, suggestions, and criticisms, that led to significant improvements in the quality of this paper.

This work is supported by DARPA contract DAAB07-95-C-D154 and grants from the California MICRO Program, Hughes Aircraft Corporation, and IBM. Hari is partially supported by an Okawa Foundation Fellowship.

12. References

[1] K. Agrawala and D Sanghi. Network Dynamics: An Experimental Study of the Internet. In *Proc. GLOBECOM '92*, December 1992.

[2] IBM AlphaWorks Home Page. http://www.alphaworks.ibm.com, 1996.

[3] M Arlitt and C. L. Williamson. Web Server Workload Characterization: The Search for Invariants. In *Proc. ACM SIGMETRICS '96*, May 1996.

[4] J.C Bolot. End-to-End Packet Delay and Loss Behavior in the Internet. In *Proc. ACM SIGCOMM '93*, San Francisco, CA, Sept 1993.

[5] R Caceres. *Multiplexing Traffic and the Entrance to Wide Area Networks*. PhD thesis, University of California at Berkeley, December 1992.

[6] CAIRN Home Page. http://www.isi.edu/div7/cairn/, 1996.

[7] R. L. Carter and M. E. Crovella. Dynamic server selection using bandwidth probing in wide-area networks. Technical Report BU-CS-96-007, Computer Science Department, Boston University, March 1996.

[8] M. Crovella and A. Bestavros. Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes. In *Proc. ACM SIGMETRICS*, May 1996.

[9] R. B. D'Agostino and M. A. Stephens. *Goodness-of-Fit Techniques*. Marcel Dekker, New York, NY, 1986.

[10] P. Danzig and S. Jamin. tcplib: A Library of TCP Internetwork Traffic Characteristics. Technical Report USC-CS-91-495, University of Southern California, 1991.

[11] P. Danzig, S. Jamin, R. Caceres, D. Mitzel, and D. Estrin. An Empirical Workload Model for Driving Wide-Area TCP/IP Network Simulations. *Internetworking: Research and Experience*, 3(1):1–26, March 1992.

[12] O. Gudmundson, D. Sanghi, and K. Agrawala. Experimental Assessment of End-to-End Behavior on Internet. In *Proc. InfoComm '93*, March 1993.

[13] S.A. Heimlich. Traffic Characterization of the NSFnet National Backbone. In *Proc. SIGMETRICS '90*, May 1990.

[14] R. Jain. *The Art of Computer Systems Performance Analysis*. John Wiley and Sons, 1991.

[15] S. Keshav. Packet-Pair Flow Control. *IEEE/ACM Transactions on Networking*, February 1995.

[16] W. Leland, M. Taqqu, W. Willinger, and D. Wilson. On the Self-Similar Nature of Ethernet Traffic. In *Proc. SIGCOMM '93*, September 1993.

[17] B. A. Mah. An Empirical Model of HTTP Network Traffic. In *Proc. InfoComm '97*, April 1997.

[18] S. McCanne and V. Jacobson. The BSD Packet Filter: A New Architecture for User-Level

Packet Capture. In *Proc. Winter '93 USENIX Conference*, San Diego, CA, January 1993.

[19] J. C. Mogul. Observing TCP Dynamics in Real Networks. Technical Report 92/2, Digital Western Research Lab, April 1992.

[20] J. C. Mogul. Network Behavior of a Busy Web Server and its Clients. Technical Report 95/5, Digital Western Research Lab, October 1995.

[21] ns – LBNL Network Simulator. http://www-nrg.ee.lbl.gov/ns/, 1996.

[22] Official 1996 olympic web site - home page. http://www.atlanta.olympic.org, 1996.

[23] V Paxson. Empirically-Derived Analytic Models of Wide-Area TCP Connections. *IEEE/ACM Transactions on Networking*, 2(4):316–336, August 1994.

[24] V. Paxson. An Analysis of End-to-End Internet Dynamics, Part II, 1996. Ph.D. dissertation in preparation.

[25] V. Paxson. End-to-End Routing Behavior in the Internet. In *Proc. ACM SIGCOMM '96*, August 1996.

[26] V. Paxson and S. Floyd. Wide-Area Traffic: The Failure of Poisson Modeling. *IEEE/ACM Transactions on Networking*, 3(3):226–244, June 1995.

[27] M. Schwartz, D. Ewing, and R. Hall. A Measurement of Internet File Transfer Traffic. Technical Report CU-CS-371-92, University of Colorado, January 1992.

[28] W. R. Stevens. *TCP/IP Illustrated, Volume 1*. Addison-Wesley, Reading, MA, Nov 1994.

[29] WOM Boiler Room. http://www.wom-plex.ibm.com, 1996.